

# Joint-Space CPG for Safe Foothold Planning and Body Pose Control during Locomotion and Climbing

Ge Sun<sup>1,†</sup> and Guillaume Sartoretti<sup>1</sup>

**Abstract**—From insects to larger mammals, legged animals can be seen easily traversing a wide variety of challenging environments, by carefully selecting, reaching, and exploiting high-quality contacts with the terrain. In contrast, existing robotic foothold planning methods remain computationally expensive, often relying on exhaustive search and/or (often offline) optimization methods, thus limiting their adoption for real-life robotic deployments. In this work, we propose a low-cost, bio-inspired foothold planning method for legged robots, which replicates the mechanism of the central nervous system of legged mammals. We develop a low-level joint-space CPG model along with a high-level vision-based controller that can inexpensively predict future foothold locations and locally optimize them based on a potential field based approach. Specifically, by reasoning about the quality of ground contacts and the robot’s stability through the high-level vision-based controller, our CPG model smoothly and iteratively updates relevant locomotion parameters to both optimize foothold locations and body pose, directly in the joint space of the robot for easier implementation. We experimentally validate our control model on a modular hexapod on various locomotion tasks in obstacle-rich environments as well as on stair climbing. Our results show that our method enables stabler and steadier locomotion than an open-loop CPG controller, yielding higher-quality feedback from onboard sensors by minimizing the effect of slippage and unexpected impacts.

**Index Terms**—Biologically-Inspired Robots, Legged Robots

## I. INTRODUCTION

LEGGED animals possess advanced neural control systems and body structures from millions of years of evolution. In particular, they can seamlessly generate coordinated and flexible gaits with feedback from their different senses to walk on various complicated terrains. Inspired by these biological systems, there is a broad interest in developing better locomotive skills based on feedback from various onboard sensors for legged robots, especially towards autonomous real-life deployments such as search and rescue, mapping, or exploration of human-denied areas [1]. Inspired by the locomotion mechanisms of animals [2], [3], this work addresses safe foothold planning for locomotion and climbing over general unknown terrain, based on onboard visual feedback.

Manuscript received: February, 24, 2022; Revised: May, 30, 2022; Accepted: July, 9, 2022.

This paper was recommended for publication by Editor Xinyu Liu upon evaluation of the Associate Editor and Reviewers’ comments.

<sup>†</sup> Corresponding author, to whom correspondence should be addressed.

<sup>1</sup>G. Sun and G. Sartoretti are with the department of Mechanical Engineering, National University of Singapore, 117575 Singapore. [sunge@u.nus.edu](mailto:sunge@u.nus.edu), [mpegas@nus.edu.sg](mailto:mpegas@nus.edu.sg)

Digital Object Identifier (DOI): see top of this page.

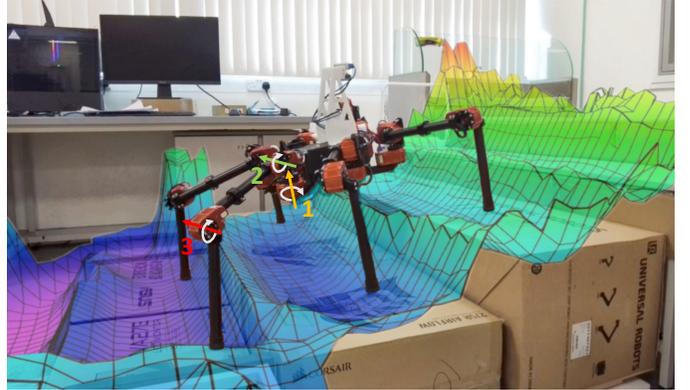


Fig. 1. Hexapod robot HEBI Daisy climbs stairs with onboard visual sensors (Intel Realsense D435i for local terrain mapping, and T265 for visual odometry). The robot can optimize the footholds based on the gradient of the potential field map shown above. In our work, the outputs of the joint-space CPG model are directly commanded to the horizontal and vertical shoulder joints (1 and 2) without any modification. The elbow joint 3 is handled via (closed-form) inverse kinematics. The rotation axes of each joint are labeled with arrows. Our simulation code is available online at [https://github.com/marmotlab/CPG\\_Foothold\\_Planning](https://github.com/marmotlab/CPG_Foothold_Planning).

We develop a computationally low-cost, bio-inspired control framework that allows a legged robot to correct its predicted foothold locations and exploit safe contacts with its environment, while ensuring stable locomotion, by integrating sensed information in a manner that replicates the main mechanisms behind stable, legged locomotion of vertebrates over rough terrain.

The rhythmic locomotion of vertebrates is usually generated by biological neural motor circuits in their spinal cord, referred to as Central Pattern Generators (CPGs) [4]. The CPGs of animals can produce rhythmic behaviors such as the ones needed to generate walking, swimming, flying, or breathing in the absence of sensory input from the higher levels of the central nervous system [5]. However, the current behavior of CPGs can be adapted based on high-level input from the brain (e.g., change step height/length, or initiate gait transitions), or based on low-level feedback from limbs (e.g., early contact with the ground or slippage) [6]. Notably, experimental results have demonstrated the existence of a *short-latency visuomotor pathway* during vertebrate locomotion over challenging terrain [3], [7]. In this pathway, the central nervous system is involved in adjusting foot trajectories reactively during the legs’ swing phase, to identify and leverage good contacts with the environment while ensuring balance [2]. Specifically, the brain integrates visual feedback with proprioceptive informa-

tion, to then output the relevant signal to the premotor circuitry in the spinal cord, adapting the CPGs' behaviors to reach corrected footholds [8].

In this work, we develop a closed-loop CPG-based framework that mimics these key mechanisms to endow a legged robot with the ability to predict, identify, correct, and leverage useful contacts with the environment during locomotion over irregular terrain. Our control framework is divided into two parts: 1) a low-level joint-space CPG model (akin to the neuronal motor circuits in the spinal cord), which directly controls the leg movements, and 2) a high-level vision-based controller (replicating the necessary brain mechanisms), which integrates the internal state information of the robot with environmental feedback from onboard sensors. Our high-level controller can easily play forward the current behavior of the CPGs to predict foothold locations without internal physics simulation. These locations are overlaid on the local map of the robot's surrounding, to assess their quality. There, we propose the use of a potential field to inexpensively and iteratively correct the footholds towards safer areas. These iterative correction signals from the high-level controller are used by the low-level joint-space CPG model, to update the relevant CPG parameters and smoothly reach these optimized footholds, while ensuring stable locomotion by adapting the body pose. Compared to existing foothold planning methods [9]–[13], which often rely on exhaustive search and/or expensive model-based optimization, our CPG model is directly defined in the joint space of the robot, closer to the biological inspiration for this mechanism, leading to simpler and computationally inexpensive implementation.

We conduct experiments to validate our approach on a hexapod robot through walking and climbing tasks. There, we show that by fully integrating our high-level vision-based controller and our low-level joint-space CPG model, the robot can carefully select and adapt footholds in real-time. Our experiments on steeper stairs, in particular, highlight that careful foothold correction is paramount, as our method allows the robot to climb significantly more effectively than an open-loop CPG controller, maximizing success rate and forward velocity.

The structure of this article is as follows: Section II presents some related works about CPG-based controller and foothold control for legged robots. Section III introduces our basic CPG model. Section IV, introduces the potential field map as well as foothold prediction and correction and body pose control. In Section V, experiments are conducted to validate our model on a Hexapod robot on different terrains. Finally, Section VI summarizes and concludes our work.

## II. PRIOR WORKS

### A. Stable Legged Locomotion

Stable legged locomotion and climbing over irregular terrain requires two key capabilities: (1) safe foothold planning, to allow the robot to identify, reach, and exploit safe contacts that aid locomotion, and (2) Body pose control, to ensure stable locomotion. Foothold planning is often cast as an optimization or search problem, with the goal to select safe and usable

contacts in the immediate neighborhood of the robot [10]–[12]. Specifically, early works have looked at mixed-integer convex optimization to plan footholds for bipedal robots [14], or relied on polynomial-based approximations to build a decision surface with ground surface characteristics for foothold selection [12]. Some works also proposed learning-based methods to compute foothold cost maps of the terrain, which can then be used by legged robots for foothold selection [15]. For hexapod robots, Tian et al. recently proposed a method based on such a foothold cost map, generated by analysis of the terrain features and the relationship between robot and terrain [13]. In order to plan more complex motions over fully-mapped environments, Winkler et al. [10] proposed an NLP solver based method to generate feasible, highly dynamic motions that optimized body trajectory along with footholds.

Existing body pose control methods can be broken down into two main categories: proprioceptive sensor-based methods [16], [17], and exteroceptive sensor-based methods [9], [18]. The former class adapts the body pose reactively, often based on feedback from onboard inertial measurement units and/or force/torque sensors [19], [20]. The latter class senses the environment first, usually using vision sensors, before taking a correcting action (i.e., more akin to feedforward control). These methods generally plan the motion or trajectory of the center of mass based on foreknowledge of the environment and of the robot's trajectory. In unknown environments, exteroceptive sensor-based approaches are generally more effective and robust than proprioceptive sensor-based ones, as they can make predictive corrections and do not only react to losses of balance. Our work, like legged animals, combines both methods.

### B. CPG

Inspired by animals, CPG-based motion control has been widely studied and used for generating gaits for legged robots [21]–[23]. Many CPG models to date rely on nonlinear equations [24], artificial neural networks [25], or topological models [26]. For dynamic system based CPGs, on which this work focuses, there are two commonly used methods of CPG for legged robots, task-space and joint-space CPGs. Task-space CPGs generate leg trajectories in Cartesian space [27], and then obtain joint angles via inverse kinematics. On the other hand, each output of a joint-space CPG represents one of the key degrees-of-freedom (DoFs) of the robot [20]. In doing so, joint-space CPGs remain closer to their biological inspiration, where outputs of a CPG are neural signals directly transmitted to individual muscles to produce motion. However, compared to model-based control or dynamic motion primitives [28], CPGs are often less effective on systems with strong dynamics like bipedal and quadrupedal robots [29]. Nevertheless, CPGs remain an enticing framework to produce low-cost, closed-loop locomotion controllers for multi-legged systems such as hexapod robots, which have the potential to remain statically-stable during locomotion and climbing [20], [30]. There are, however, a few papers on CPG-based foothold planning and control for legged robots [31]–[34]. In particular, we note the work of Asadi et al. [32] that proposed controlling footholds

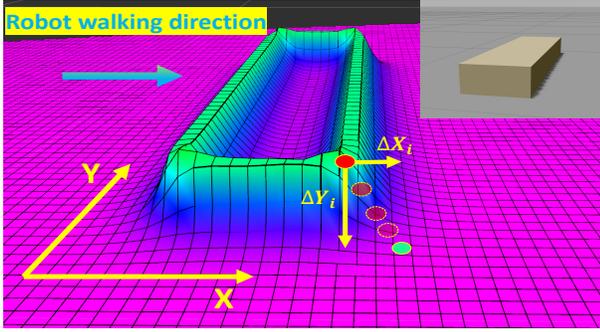


Fig. 2. Potential field map over a block obstacle (shown at the top right), and its use to iteratively correct predicted foothold locations. The red dots indicate iteratively predicted footholds, and the green point the adjusted foothold based on the local gradient of the potential field, which now lies further from the dangerous area (defined here as the edges of the obstacle).

by adjusting the main frequency of their CPG model, which was limited to avoiding simple obstacles on flat ground. Sapatra *et al.* [34] integrated a dynamic-density topological map with an affordance mechanism to deal with sudden obstacles within a CPG model. Nonetheless, their work does not actively control the robot's body pose for stability.

### III. LOW-LEVEL JOINT-SPACE CPG MODEL

In this work, we rely on a dynamical system approach to CPGs, in which the central pattern generator is expressed as a set of coupled oscillators. The mathematical CPG model we use builds upon and extends previous works [20], [27]. Since we consider a joint-space CPG, the output values of the oscillators directly represent joint angles of key articulations of the robot. Specifically for our robot,  $\theta_1(t) = [\theta_{1,1}(t), \theta_{1,2}(t), \dots, \theta_{1,6}(t)]$  was used to present the angles of the first, horizontal shoulder joints (joint 1 in Fig. 1), and  $\theta_2(t) = [\theta_{2,1}(t), \theta_{2,2}(t), \dots, \theta_{2,6}(t)]$  the angles of the second, vertical shoulder joints (joint 2 in Fig. 1).

$$\begin{cases} \dot{\theta}_{1,i}(t) = -\frac{a_i b_i}{2} \cdot \omega_i \cdot \partial H_{\rho\theta_{2,i}} + \alpha_i (\mu^2 - H_{\rho}(\theta_{1,i}, \theta_{2,i})) \cdot \partial H_{\rho\theta_{1,i}} \\ \dot{\theta}_{2,i}(t) = +\frac{a_i b_i}{2} \cdot \omega_i \cdot \partial H_{\rho\theta_{1,i}} + \beta_i (\mu^2 - H_{\rho}(\theta_{1,i}, \theta_{2,i})) \cdot \partial H_{\rho\theta_{2,i}} \\ \dot{c}_{x,i}(t) = f_{cx}(\Delta X_i(t), s, d_i(t) + \Delta Y_i(t), c_{x,i}) \\ \dot{a}_i(t) = f_a(c_{x,i}(t), s, d_i(t) + \Delta Y_i(t), a_i), \end{cases} \quad (1)$$

where  $\partial H_{\rho\xi} = \frac{\partial H_{\rho}}{\partial \xi}(\theta_{1,i}(t), \theta_{2,i}(t))$ ,  $\mu$  denotes the ‘‘radius’’ of the limit cycle,  $\omega_i$  the oscillation frequency (thus defining the overall locomotive velocity),  $\alpha_i$ , and  $\beta_i$  the forcing strength in the horizontal axis and vertical axis respectively,  $s$  the desired stride length of legs on the ground, and  $d_i$  the desired distance between the trajectory of grounded legs and the side of the robot's body (shown in Fig. 4).  $\Delta X_i(t)$  and  $\Delta Y_i(t)$  denote the desired correction between predicted and desired footholds along the  $X$  and  $Y$  axis respectively (as illustrated in Fig. 2 and discussed below in Section IV-C). In this work, we rely on an elliptic limit cycle [20], for which the Hamiltonian function reads:

$$H_{\rho}(\theta_{1,i}, \theta_{2,i}) = \left| \frac{\theta_{1,i} - c_{x,i}}{a_i} \right|^2 + \left| \frac{\theta_{2,i} - c_{y,i}}{b_i} \right|^2, \quad (2)$$

with  $\rho = (a_i, c_{x,i})$ , because the foothold of leg  $i$  is mainly adapted by changing the values of  $a_i$  and  $c_{x,i}$ .  $a$  is the vector of semi-major axes (thus impacting the stride length of each leg), and  $b$  the vector of semi-minor axes (impacting step heights).  $c_x$  and  $c_y$  are the vectors of  $x$ - and  $y$ -coordinates of the origin of each leg's limit cycle in its joint space. In this work, to implement our foothold correction method, the key CPG parameters  $a$  and  $c_x$  are updated with time (as shown in Fig. 3). The associated foothold adjustment functions  $f_{cx}$  and  $f_a$  in Eq.(1) are detailed in Section IV-C below. Finally, we note that, especially in our experiments on robot, we further leverage our previous work [20] to control the robot body pose with inertial feedback, by adapting  $c_y$  with time based on feedback from the onboard IMU towards a desired body pose (this is detailed further in Section IV-D).

To ensure stable, steady locomotion, the coupling between the six oscillators (legs) is crucial and defines the gait of the robot. This work relies on the *Kuramoto* model from [35], [36] to define the gait coupling term of our CPG model, which updates the angular frequency of each oscillator to keep the legs coupled:

$$\omega_i = \omega_0 + \sum_{j=1}^N \lambda_{ij} \sin(\phi_j - \phi_i - \Delta\phi_{ij}), \quad (3)$$

where  $\phi_i(t) = \arctan\left(\frac{\theta_{2,i}(t) - c_{y,i}(t)}{\theta_{1,i}(t) - c_{x,i}(t)}\right)$  is the current phase of the  $i$ th oscillator, and  $\omega_0$  is the nominal angular velocity (same constant value for all oscillators). The angular velocity of the  $i$ th oscillator  $\omega_i$  is affected by the current  $(\phi_j - \phi_i)$  and the desired  $(\Delta\phi_{ij})$  phase difference between every pair of oscillator  $i$  and  $j$ .  $\lambda_{ij}$  denotes the coupling strength between two oscillators (constant for all leg pairs in practice). In this work, we rely on the alternate-tripod gait, for which the desired phase matrix reads:

$$\Delta\phi_{ij} = \begin{cases} 0, & \text{if } i, j \in \mathbb{A} \text{ or } i, j \in \mathbb{B} \\ \pi, & \text{if } i \in \mathbb{A}, j \in \mathbb{B} \text{ or } i \in \mathbb{B}, j \in \mathbb{A} \end{cases}, \quad (4)$$

where  $\mathbb{A} = \{1, 4, 5\}$  and  $\mathbb{B} = \{2, 3, 6\}$  are sets of three legs defining the first and second tripods respectively (legs are numbered in Fig. 4).

Finally, to make the ground trajectory of the end-effectors (feet) straight in the world frame, and parallel to the forward direction of the robot (refer to Fig. 4), the angle of the third joint of each leg can be expressed via inverse kinematics (closed-form for our 3-DoF-per-leg robot):

$$\theta_{3,i}(t) = \frac{\pi}{2} + \arcsin\left(\frac{d_i(t)}{L_3 \cdot \cos\theta_{1,i}(t)} - \frac{L_2 \cos\theta_{2,i}(t)}{L_3}\right) - \theta_{2,i}(t) \quad (5)$$

where  $L_2$  and  $L_3$  are the lengths of the second link and third link for each leg.

It is worth mentioning that the same framework could easily be extended to use a task-space CPG [27], e.g., to allow more complex combined motion such as omnidirectional translations, or turning walks. However, such a change would come at the cost of losing some of the bio-inspiration, and would likely increase the overall computational cost due to the heavy reliance on inverse kinematics calls.

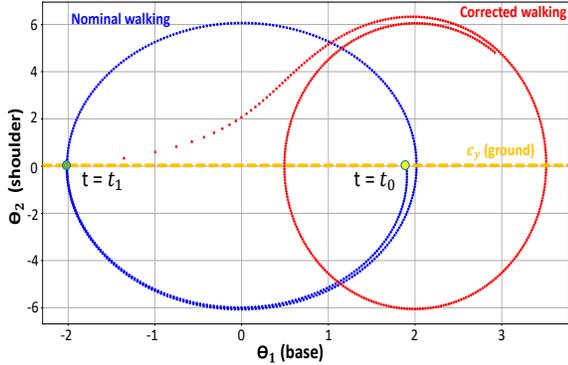


Fig. 3. Example foothold correction via the change of  $c_x$  and  $a$ . Initial values of the CPG parameters are  $a = 2$ ,  $b = 6$ ,  $c_x = 0$ , and the initial position ( $\theta_1 = 1.9$   $\theta_2 = 0$ ) is shown as the yellow dot with green outline ( $t = t_0$ ). The CPG is first allowed to converge to the first limit cycle (blue circle). At  $t = t_1$  (solid green dot), the CPG parameters suddenly change to  $a = 1.5$ ,  $b = 6$ ,  $c_x = 2$ , and we can see the oscillator converge to the new limit cycle (red circle) rapidly and stably. This mimics the change of parameters associated with foothold correction (here, the leg associated with this oscillator would be placed further forward along the X axis of the world); note that  $c_y$  indicates the ground level.

#### IV. HIGH-LEVEL VISION-BASED CONTROLLER

This section describes our high-level vision-based controller for foothold planning and body pose optimization.

##### A. Foothold Potential Field Map

A precise map of the environment is required to plan the motion and foothold adaption for legged robots on irregular terrain. Real-time map building with onboard sensors is a challenging issue for legged robots. Odometry drift might occur due to some unexpected events like foot slippages and vibrations from unexpected impacts with the ground [19]. Recently, Fankhauser et al. [37] proposed an efficient mapping method to build and update an *elevation map* based on the uncertainty estimates of the robot's pose estimation. In this work, we use a similar approach to build a local elevation map from onboard visual sensors. In their later work on foothold planning and motion control for quadrupedal robots [9], Fankhauser et al. then proposed to evaluate the foothold score of each cell in the elevation map, based on terrain features (slope, curvature and roughness). This approach works very well, but requires significant computation for this exhaustive search step. Here, we propose instead to rely on a potential field defined over the local elevation map, to iteratively compute corrections to the predicted foothold that can optimize future foothold locations at a low cost, and can be feedback directly into our joint-space CPG model. Furthermore, our potential field map can be customized according to different scenarios and demands, i.e., to avoid specific types of hazards.

The robot-centric elevation map is expressed as a 2D grid map, with its origin at the (projected downwards) position of the robot's body. The height of each corresponding terrain area/cell is stored in this 2D grid map as  $h(X, Y)$ . We then construct a potential field function  $P(X, Y)$  over the elevation map, which will be used for foothold correction.

Sudden changes in the elevation usually signify an edge in the world, such as the edge of stairs or other sharper object,

which are often poor foothold locations that can lead to slippage. Therefore, as a first example, we detail the construction of a potential field function that will discourage the placement of feet at regions of high gradient of the elevation map. We later detail another potential field constructed to avoid objects of a specific height of the ground.

We first define the gradient map of the elevation map as:

$$g(X, Y) = \sqrt{\frac{\partial h}{\partial X}(X, Y)^2 + \frac{\partial h}{\partial Y}(X, Y)^2}. \quad (6)$$

However, the gradient map is often noisy, especially when the elevation map is the result of onboard mapping on a real robot, and cannot directly be used as a good potential field for foothold planning. Therefore, we propose to rely on a simple thresholding operation to identify dangerous points  $(x, y)$ , around which we place a Gaussian distribution  $\mathcal{N}\left(\begin{bmatrix} x \\ y \end{bmatrix}, \frac{\eta}{g(x, y)}\right)$ , with  $\eta$  the covariance. We further propose to make the covariance a function of the magnitude of the gradient, to create a more peaked potential field around areas of high danger (i.e., larger gradient magnitude, thus stronger repulsion). The potential field  $P(X, Y)$  is finally constructed as a sum of these Gaussian functions:

$$P(X, Y) = \sum_{i=1}^N \frac{1}{2\pi\sigma_i^2} e^{-[(X-\mu_{xi})^2 + (Y-\mu_{yi})^2]/(2\sigma_i^2)}, \quad (7)$$

where  $(\mu_{xi}, \mu_{yi})$  are all the dangerous areas identified by our simple thresholding operation, and  $\sigma_i^2 = \frac{\eta}{g(\mu_{xi}, \mu_{yi})}$ .

##### B. Foothold Prediction

To predict the footholds of the swing legs, we first predict their foot trajectories in the joint space by forward-solving our CPG model using the Euler method, while keeping the current parameters constant (by selecting the appropriate  $\alpha$  and  $\beta$  values in Eq.(1), we assume that the oscillators of each leg have already rapidly converged to the current CPG parameters). To do this efficiently, we transform the known limit cycle of the CPG into polar coordinates, and predict the outputs of the CPG as:

$$\begin{cases} \theta_{1,ip} = a_i \cos \psi_i + c_{x,i} \\ \theta_{2,ip} = b_i \sin \psi_i + c_{y,i} \\ \theta_{3,ip} = f_{IK}(\theta_{1,ip}, \theta_{2,ip}, d_i), \end{cases} \quad (8)$$

where  $\psi_i(t) = \arctan\left(\frac{\theta_{2,ip}(t) - c_{y,i}(t)}{\theta_{1,ip}(t) - c_{x,i}(t)}\right)$  is the predicted phase of the swing leg's oscillator. We discretize  $\psi$  within  $[0, T_\psi]$  (with  $T_\psi$  the remainder of the flight phase duration) in the swing phase of each leg and substitute it into Eq.(8) to express the predicted foot trajectories (as illustrated in Fig. 7, (c)).  $\theta_{1,ip}$ ,  $\theta_{2,ip}$ ,  $\theta_{3,ip}$  denotes the predicted first, second, and third joint angles of leg  $i$ . The predicted foot trajectories are calculated using forward kinematics, while taking into account future body poses by assuming the robot's body moves at a constant linear speed. That is, in our work, the robot is assumed to be quasi-static and purely kinematic. Finally, the intersection point between each of these discretized trajectories and the elevation map, which can be obtained very inexpensively, yields the predicted next-step foothold  $(X_{p,i}, Y_{p,i})$  for each leg.

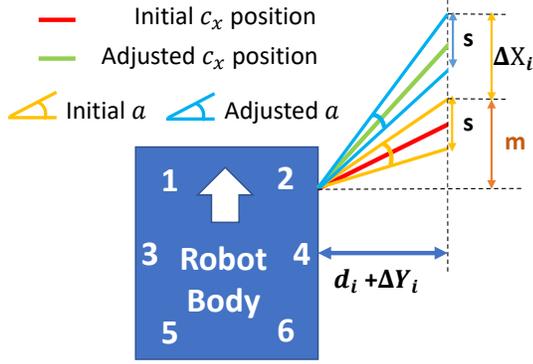


Fig. 4. Geometric top view of the relevant quantities at play when updating  $c_x$  and  $a$  in the joint space, based on the  $s$  and  $d_i$  values given in the world frame, based on the kinematics of the robot. Solid colored line for  $c_x$  and  $a$  instead represent their angular value from the horizontal dotted line, for easier visualization (legs are numbered with white numbers).

### C. Foothold Correction

In our work, the potential field map (Section IV-A) is used to iteratively correct the predicted footholds, pushing them away from undesired areas along the X and/or Y dimensions of the elevation/potential field map. Specifically, we rely on the local gradient of the potential field map at the predicted foothold position of each leg, to calculate the correct steps  $\Delta X_i$  and  $\Delta Y_i$ :

$$\begin{cases} \Delta X_i = \zeta \frac{\partial P}{\partial X}(X_{p,i}, Y_{p,i}), \\ \Delta Y_i = \zeta \frac{\partial P}{\partial Y}(X_{p,i}, Y_{p,i}), \end{cases} \quad (9)$$

where  $\zeta$  is the step size of the gradient update.

In this work, the key idea behind performing repeated such gradient-based corrections, as opposed to performing a single foothold optimization/correction, is to distribute the associated computation over time. Our foothold prediction and correction process is inexpensive, and can still yield near-optimal (locally-optimized) footholds after enough gradient updates. Thus, compared to model-based optimization algorithms, our method can be easily applied to multi-legged systems such as hexapod robots. Performing numerous small foothold corrections also allows us to keep running the main CPG loop at relatively high frequency, smoothly updating the key CPG parameters and thus keeping the oscillators (and thus the robot's motion) stable.

To enact the foothold corrections Eq.(9), we update the relevant parameters of the low-level CPG model with time. Regarding  $\Delta X$ , we note that the parameters  $a$  and  $c_x$  are the main ones driving the placement of the end effector along the X axis of the world. That is, intuitively, the foothold's offset in x direction is mainly affected by  $c_x$ , the joint-space offset of the planar base joint of the robot. However, the stride length of the different legs in the world should remain constant to make sure the robot is walking forward in a fixed direction with a constant speed. Hence, we also need to update  $a$  along with  $c_x$  to keep the stride length of the legs unchanged.

Given the robot's morphology, we can calculate the new  $a$  and  $c_x$  values for each leg, which allow the placement of the foot at the corrected foothold location ( $X_{p,i} + \Delta X_i$ ) while keeping  $s$  and  $d_i$  unchanged, which we express as:

$$\begin{aligned} c_{x,i}(t+1) &= \kappa(\Delta X_i(t), s, d_i(t) + \Delta Y_i(t)), \\ a_i(t+1) &= \phi(c_{x,i}(t), s, d_i(t) + \Delta Y_i(t)), \end{aligned} \quad (10)$$

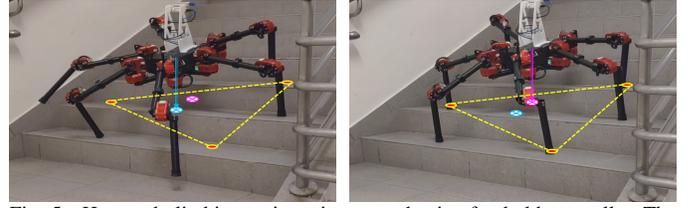


Fig. 5. Hexapod climbing stairs using our adaptive foothold controller. The robot aims at avoiding to step on the edge of each step while optimizing its body pose for balance. The new support polygon (yellow dotted line) is defined using the predicted footholds (yellow circles filled with red). The blue circle and pink circle are the downward-projection (along gravity) of the current and predicted CoM of the robot, respectively.

Specifically,  $c_x$  is updated using the closed-form equation below, which is explained geometrically in Fig. 4.  $a$  values are updated using a similar geometric argument to ensure  $s$  and  $d$  are preserved, after  $c_x$  is updated.

$$\begin{aligned} c_{x,i}(t+1) &= \frac{1}{2} \left[ \arctan \left( \frac{m + \Delta X_i(t)}{d_i(t) + \Delta Y_i(t)} \right) \right. \\ &\quad \left. + \arctan \left( \frac{m + \Delta X_i(t) - s}{d_i(t) + \Delta Y_i(t)} \right) \right] \end{aligned} \quad (11)$$

Differently, adjusting the foothold with  $\Delta Y$  along the Y direction in the world is achieved by updating the third joint of each leg. There, Eq.(5) is modified to:

$$\theta_{3,i}(t) = \frac{\pi}{2} + \arcsin \left( \frac{d_i(t) + \Delta Y_i(t)}{L_3 \cdot \cos \theta_{1,i}(t)} - \frac{L_2 \cos \theta_{2,i}(t)}{L_3} \right) - \theta_{2,i}(t) \quad (12)$$

Finally, the functions  $f_{c_x}$  and  $f_a$  needed for the main CPG equation Eq.(1) can be expressed as:

$$f_{c_x}(\cdot) = \frac{\kappa(\cdot) - c_{x,i}(t)}{\Delta t}, \quad f_a(\cdot) = \frac{\phi(\cdot) - a_i(t)}{\Delta t}. \quad (13)$$

We note that, to best utilize the robot's workspace and avoid singular configurations, the adjustment range of the values of  $a$  and  $c_x$  are limited; furthermore, we reset the values of  $a$  and  $c_x$  for legs about to step off the ground (i.e., resets happen at the beginning of each flight phase). Finally, to ensure the leg swings smoothly to the next desired foothold without being affected by the reset of  $a$  and  $c_x$ , we set the value of  $\beta$  larger than  $\alpha$ , thus pushing the oscillators to preferentially converge to the new limit cycle along the vertical axis.

### D. Body Pose Correction

One of the concerns with adapting footholds is that of negatively impacting the stability of the robot's body, as the support polygon changes when footholds are modified. In our previous work [20], the robot's body orientation was controlled by correcting the  $c_y$  values from inertial feedback. In this work, we extend this idea and propose to further correct the  $c_x$  and  $a$  after each foothold correction step to improve stability. We update all  $c_x$  and  $a$  values to translate the robot's body forward/backward with respect to the legs, in order to keep the robot's center of mass (CoM) over its new support polygon without affecting the footfall planning.

To this end, we calculate both the current and predicted downward-projection (along gravity) of the robot's CoM onto the new support polygon, which is calculated on the basis of predicted footholds (Fig. 5). In doing so, we ignore the mass of the robot's leg (including the actuator of the third joint of each



Fig. 6. Screenshots of our flat-ground, obstacle-rich experiment. Boxes, acting as obstacles that should not be stepped on (the color of each box is meaningless here), have been randomly placed on the ground. Here, note that the potential field is built from the *height* of cells in the elevation map, and not the local gradient. In this scenario, the robot is seen adapting its foothold in both the  $X$  and  $Y$  directions to avoid stepping on the obstacles, as highlighted for the front left leg in the two right-most images. The robot's traveling direction is indicated by the orange arrow in the first frame.

leg) and assume that the robot's CoM coincides with the center point of its body. If the projected CoM position falls within the new support polygon, the robot is considered statically stable, and no correction is made to  $c_x$  and  $a$ . However, if this projected point lies outside the support polygon, we calculate a translation vector  $\vec{p} = (\Delta R, 0, 0)$  to translate the body forward/backward with respect to the grounded leg frame. The  $c_{x,i}$  and  $a_i$  are further updated as:

$$\begin{aligned} c_{x,i}(t+1) &= \kappa(-\Delta R, s, d_i(t) + \Delta Y_i(t)), \\ a_i(t+1) &= \phi(c_{x,i}(t), s, d_i(t) + \Delta Y_i(t)). \end{aligned} \quad (14)$$

By using the robot's kinematics to translate this translation vector to an offset in the joint space of the first shoulder joint, the CoM will be moved to fall within the new support polygon. It should be noted that body position correction occurs once foothold planning has found the local-optimized footholds after sufficient iterations. As described, we updated the  $c_x$  and  $a$  for both grounded legs and swing legs at the same time, to ensure that the corrected foothold positions are left unchanged in the world frame.

## V. EXPERIMENTS

We experimentally validate and evaluate our CPG-based foothold controller on a modular hexapod robot (HEBI Daisy, illustrated in Fig. 1). The robot is equipped with an onboard tracking camera for visual odometry (Realsense T265) and an RGB-D camera for local terrain mapping (Realsense D435i). The depth range of the D435i camera is between 0.3 m and 3 m, and its field of view is  $87^\circ \times 58^\circ (\pm 3^\circ)$ . The robot-centric map is a 3m x 3m discretized elevation map that is built incrementally over time, while taking into account the robot's motion. The radius of the robot is 0.5 m, and the radius of the robot's feet is 0.025 m. Therefore, the elevation map and potential field map resolution are set to 0.05 m, which is precise enough for foothold prediction and adjustment at low computational cost. The frequency of foothold prediction and correction is set at 10 Hz. Moreover, the minimum update frequency of the elevation map and the potential field map is 2Hz, which we have found to be sufficient to handle reasonable dynamic changes in the environment.

Based on the robot's morphology, we define the initial CPG parameters as:

$$\begin{aligned} c_{x_0} &= \begin{bmatrix} -\frac{\pi}{6} & -\frac{\pi}{6} & 0 & 0 & \frac{\pi}{12} & \frac{\pi}{12} \end{bmatrix}, \\ c_{y_0} &= \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (15)$$

The shoulder joint angles of the robot are directly set by the CPG output  $\theta_{1,i}$  and  $\theta_{2,i}$  ( $\theta_{2,i}(t) = \max(\theta_{2,i}(t), c_{y,i})$  account for ground contact), while elbow joint angles are calculated using Eq.(5).

$b_i$	0.8	$\omega_0$	1
$\alpha_i$	0.3	$\zeta$	0.25
$\beta_i$	150	$\lambda_{i,j}$	3
$\mu$	1	$\eta$	0.05
$s$	0.13 m	$L_2, L_3$	0.325 m

TABLE I

CONTROLLER SETTINGS AND ROBOT DATA

We discuss the results of two sets of experiments. First, we place box obstacles randomly on flat ground (as shown in Fig. 6). The length, width, and height of the box are 16 cm, 9.5 cm, and 8 cm, respectively. In this scenario, we assume all the boxes are dangerous areas (i.e., undesired foothold), and build the potential field map based on the height of cells in the elevation map (all grids with a height of more than 0.03 m above ground are considered as dangerous area). Fig. 6 illustrates how the robot adjusts the foothold with our controller when walking on such irregular terrain, by adapting foothold locations along the  $X$  and  $Y$  dimensions in the world, by updating the relevant CPG parameters online. We report the traversing time and the number of failed footholds for the open-loop CPG controller and our adaptive foothold controller in Table II. We note that our foothold controller can achieve the same control frequency as the open loop controller, when both controllers run map building, but the open loop controller does not run the foothold prediction and correction mechanisms.

Control method	Time to Traverse	Failed foothold	Frequency
Open-loop Controller	83.2s	10	50Hz
Adaptive Foothold Controller	86.6s	0	50Hz

TABLE II

Real robot results in our flat-ground, obstacle-rich experiment.

Stairs' Average Slope	Control method	Climbing Time (std)	Success Rate
17°	Open-loop Controller	151.2s (4.58)	100%
	Adaptive Foothold Controller	<b>117.4s</b> (8.89)	100%
26°	Open-loop Controller	140.3s (30.62)	60%
	Adaptive Foothold Controller	<b>109.2s</b> (16.87)	100%

Instantaneous Acceleration Magnitude (lower is better)	Average (std)	Peak
Open-loop Controller	0.0698m/s <sup>2</sup> (0.0181)	1.0632m/s <sup>2</sup>
Adaptive Foothold Controller	<b>0.0611m/s<sup>2</sup></b> (0.0132)	<b>0.6803m/s<sup>2</sup></b>

TABLE III

Real robot results in our stair climbing experiment.

Second, we perform experiments on a mockup stairs made of 5 irregular steps, as shown in Fig. 7. The first three steps are 14.5 cm high, while the latter two are 16 cm. The width of each step varies from 35 cm to 52 cm. The thresholding operation here is based on the gradients of the elevation map: all grids with a gradient of more than 0.03 in magnitude are considered as dangerous area. In this experiment, the robot needs to precisely control its footholds to fully climb up to the highest platform. We perform 5 trials with our adaptive

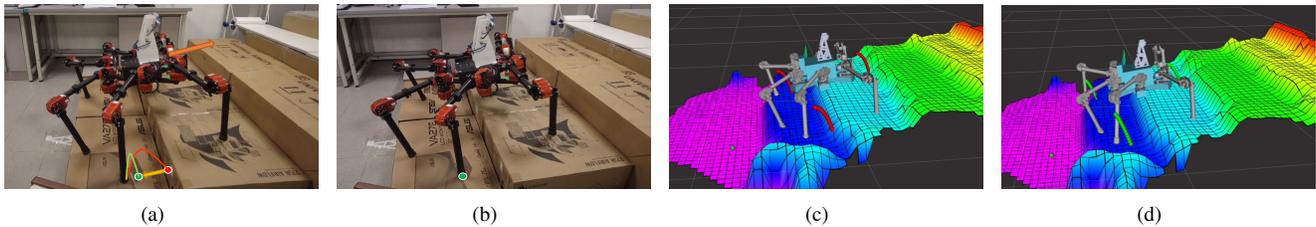


Fig. 7. (a) The predicted foothold of hind right leg lies at the edge of the stairs (red point), which is an unsafe area (high chance of slippage). (b) Therefore, the foothold is adjusted to a safe area based on the gradient of the potential field map at that predicted foothold location. (c)(d) Visualization of the elevation map/potential field, foothold prediction and foothold correction at the same instants as (a) and (b). The red line (before foothold correction) in (c) and green line (after foothold correction) in (d) is the predicted trajectory of the foot. In our experiments, all controllers run directly on the robot, on an onboard Intel NUC i5 with 8GB RAM.

foothold controller, and 5 using a simple open-loop controller. In this open-loop controller, we disable foothold correction and fix the body-pose correction to match the stairs’ average slope. In Table III, top, we report the average time to climb the first ( $17^\circ$  slope) portion of the stairs, as well as the second ( $26^\circ$  slope) portion, as well as the success rate. In the lower part of the same table, we also report the average and peak magnitude of instantaneous acceleration measured at the robot’s body (sampled at 50Hz frequency).

From these results, we note that our foothold controller significantly helps maintain the average velocity of the robot during climbing, by avoiding poor ground contacts that result in slippage (and thus loss of propulsion) that we observed in our open-loop experiments. The corrected footholds are guaranteed to be at least locally optimal after enough gradient updates (footholds can converge within the first half of the leg swing phase). Compared to the open-loop controller, our foothold controller can effectively eliminate large vibrations of the robot body, hence improving the locomotion stability (refer to Table III bottom). Second, we further note that, in experiments on the steeper stairs, the success rate of the open-loop controller is only around 60%, as the poor ground contacts often lead to loss of balance and/or to significant changes in the robot’s direction of travel (which forced us to stop the experiment for safety). Fig. 7 shows a breakdown of the main foothold prediction, correction, and action loop of our CPG-based controller. First, by combining trajectory prediction and local elevation map, the controller obtains the predicted foothold in Fig. 7 (c). There, the local gradient of the potential field map is calculated at that predicted foothold location, which in this case falls in a potentially dangerous area (i.e., large gradient/repulsion). This repulsion force iteratively pushes the predicted foothold away over the next update cycles of the CPG, finally settling on a corrected foothold over the safe area shown in Fig. 7 (d).

## VI. CONCLUSION AND FUTURE WORK

In this work, we developed a low-cost, bio-inspired CPG framework that predicts, iteratively corrects, and exploits safe footholds on challenging terrain, for the locomotion and climbing of legged robots based on visual feedback. Our framework replicates the main mechanism at play in the *short-latency visuo-motor pathways* documented in legged vertebrates [3] [7]. The mapping and potential field construction may be linked to the manner in which animals aggregate visual information about their surroundings and add/embed

semantic information (regarding the safety of various areas for the purpose of foothold planning). Additionally, our joint-space CPG controller replicates the function of the spinal cord which can alter foot trajectories directly and quickly. Our controller can distribute the foothold and body correction processes in time and keep computation low, while improving overall stability by keeping changes small and smooth. In particular, we note that the potential field used by our high-level vision-based control can be easily customized to fit various scenarios, as shown in our experiments that consider obstacle avoidance on flat ground as well as safe foothold and body pose correction for efficient stair climbing. Unlike model-based optimization methods, our approach is model-free and can adapt to sudden changes in the real-time map, as we predict and correct footholds continually during the swing phase. Furthermore, we note that our controller can be directly extended to other hexapedal gaits. To generalize to various robot morphologies (e.g., quadrupeds), our CPG model can be adjusted according to the joint-to-joint relationship, as proposed in [38]. Incorporating foot contact information (e.g., force/torque sensor) could allow our method to also adapt to dynamic gaits, as in [39], while remaining close to the biological inspiration. We believe that our work may provide an avenue to further our understanding of the key mechanisms at play in animal locomotion, via their replication on robots and further analysis.

However, mammals in nature most likely reason about the quality of ground contacts in their surrounding by relying more on experience and memory, especially when traversing terrains they have not seen before. Therefore, future work will investigate the use of deep learning methods (including memory-enabling networks), to improve the assessment of potential contacts and further improve performance across tasks and environments, as investigated in [40]. Our method may face difficulties when dealing with complex scenarios, such as unstructured terrains (gravel, etc.) and dynamical scenarios where footholds might be dynamically safe (i.e., can be moved by the robot, e.g., rocks). Our future work will also evaluate and improve our method in such cases by developing a mechanism to determine whether the corrected foothold falls within a local-optimal solution and fall back on other means of optimizing footholds that can circumvent this issue.

## ACKNOWLEDGMENTS

This work was supported by the Singapore Ministry of Education Academic Research Fund Tier 1. Authors would

like to thank Qun Cao for her assistance with the experimental setup and supplementary movie, as well as Kathryn Daltorio for her feedback on earlier versions of this manuscript.

## REFERENCES

- [1] J. Ackerman and J. Seipel, "Energetics of bio-inspired legged robot locomotion with elastically-suspended loads," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 203–208.
- [2] J.-H. Courjon, E. Olivier, and D. Péliisson, "Direct evidence for the contribution of the superior colliculus in the control of visually guided reaching movements in the cat," *The Journal of physiology*, vol. 556, no. 3, pp. 675–681, 2004.
- [3] R. Reynolds and B. Day, "Rapid visuo-motor processes drive the leg regardless of balance constraints," *Current Biology*, vol. 15, no. 2, pp. R48–R49, 2005.
- [4] S. Grillner and P. Wallen, "Central pattern generators for locomotion, with special reference to vertebrates," *Annual review of neuroscience*, vol. 8, no. 1, pp. 233–261, 1985.
- [5] E. Marder and D. Bucher, "Central pattern generators and the control of rhythmic movements," *Current biology*, vol. 11, no. 23, pp. R986–R996, 2001.
- [6] T. Iwasaki and M. Zheng, "Sensory feedback mechanism underlying entrainment of central pattern generator to mechanical resonance," *Biological cybernetics*, vol. 94, no. 4, pp. 245–261, 2006.
- [7] B. L. Day and P. Brown, "Evidence for subcortical involvement in the visual control of human reaching," *Brain*, vol. 124, no. 9, pp. 1832–1840, 2001.
- [8] S. E. Boehnke and D. P. Munoz, "On the importance of the transient visual response in the superior colliculus," *Current opinion in neurobiology*, vol. 18, no. 6, pp. 544–551, 2008.
- [9] P. Fankhauser, M. Bjelonic, C. D. Bellicoso, T. Miki, and M. Hutter, "Robust rough-terrain locomotion with a quadrupedal robot," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 5761–5768.
- [10] A. W. Winkler, C. D. Bellicoso, M. Hutter, and J. Buchli, "Gait and trajectory optimization for legged systems through phase-based end-effector parameterization," *IEEE Robotics and Automation Letters*, vol. 3, no. 3, pp. 1560–1567, 2018.
- [11] R. Deits and R. Tedrake, "Footstep planning on uneven terrain with mixed-integer convex optimization," in *2014 IEEE-RAS international conference on humanoid robots*. IEEE, 2014, pp. 279–286.
- [12] D. Belter, P. Łabecki, and P. Skrzypczyński, "Map-based adaptive foothold planning for unstructured terrain walking," in *2010 IEEE International Conference on Robotics and Automation*. IEEE, 2010, pp. 5256–5261.
- [13] Y. Tian and F. Gao, "Efficient motion generation for a six-legged robot walking on irregular terrain via integrated foothold selection and optimization-based whole-body planning," *Robotica*, vol. 36, no. 3, pp. 333–352, 2018.
- [14] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, and S. Schaal, "Learning, planning, and control for quadruped locomotion over challenging terrain," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 236–258, 2011.
- [15] J. Zico Kolter and A. Y. Ng, "The stanford littledog: A learning and rapid replanning approach to quadruped locomotion," *The International Journal of Robotics Research*, vol. 30, no. 2, pp. 150–174, 2011.
- [16] R. Grandia, F. Farshidian, R. Ranftl, and M. Hutter, "Feedback mpc for torque-controlled legged robots," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 4730–4737.
- [17] M. Bjelonic, N. Kottege, and P. Beckerle, "Proprioceptive control of an over-actuated hexapod robot in unstructured terrain," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2016, pp. 2042–2049.
- [18] A. Roennau, G. Heppner, M. Nowicki, J. M. Zöllner, and R. Dillmann, "Reactive posture behaviors for stable legged locomotion over steep inclines and large obstacles," in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2014, pp. 4888–4894.
- [19] D. Belter and M. R. Nowicki, "Optimization-based legged odometry and sensor fusion for legged robot continuous localization," *Robotics and Autonomous Systems*, vol. 111, pp. 110–124, 2019.
- [20] 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*, 2018, pp. 1–5.
- [21] 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.
- [22] A. Sprowitz, L. Kuechler, A. Tuleu, M. Ajallooeian, M. D'Haene, R. Moeckel, and A. Ijspeert, "Oncilla robot: a light-weight bio-inspired quadruped robot for fast locomotion in rough terrain," in *5th International Symposium on Adaptive Motion of Animals and Machines*, no. CONF, 2011.
- [23] P. Arena, L. Fortuna, M. Frasca, and G. Sicurella, "An adaptive, self-organizing dynamical system for hierarchical control of bio-inspired locomotion," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 4, pp. 1823–1837, 2004.
- [24] Y. Fukuoaka, H. Kimura, and A. H. Cohen, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," *The International Journal of Robotics Research*, vol. 22, no. 3-4, pp. 187–202, 2003.
- [25] P. Arena and L. Fortuna, "Collective behaviour in cellular neural networks to model the central pattern generator," *International Journal of Systems Science*, vol. 31, no. 7, pp. 827–841, 2000.
- [26] M. Felipe, F. Yang, and Z. Yang, "Building artificial cpgs with asymmetric hopfield networks," in *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 4. IEEE, 2000, pp. 290–295.
- [27] S. Shaw, G. Sartoretti, J. Olkin, W. Paivine, and H. Choset, "Workspace cpg with body pose control for stable, directed vision during omnidirectional locomotion," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 6316–6322.
- [28] G. Xin, H.-C. Lin, J. Smith, O. Cebe, and M. Mistry, "A model-based hierarchical controller for legged systems subject to external disturbances," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 4375–4382.
- [29] J. Yu, M. Tan, J. Chen, and J. Zhang, "A survey on cpg-inspired control models and system implementation," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 3, pp. 441–456, 2013.
- [30] A. Skaburskytė, M. Luneckas, T. Luneckas, J. Kriaučiūnas, and D. Udris, "Hexapod robot gait stability investigation," in *2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE)*. IEEE, 2016, pp. 1–4.
- [31] 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 *International Conference on Artificial Neural Networks*. Springer, 2019, pp. 698–710.
- [32] F. Asadi, M. Khorram, and S. A. A. Moosavian, "Cpg-based gait planning of a quadruped robot for crossing obstacles," in *2015 3rd RSI International Conference on Robotics and Mechatronics (ICROM)*. IEEE, 2015, pp. 216–222.
- [33] —, "Bio-inspired central pattern generator for development of cartesian motor skills for a quadruped robot," *World Journal of Advanced Research and Reviews*, vol. 12, no. 1, pp. 066–085, 2021.
- [34] A. A. Saputra, N. Takesue, K. Wada, A. J. Ijspeert, and N. Kubota, "Aquro: A cat-like adaptive quadruped robot with novel bio-inspired capabilities," *Frontiers in Robotics and AI*, vol. 8, p. 35, 2021.
- [35] J. A. Acebrón, L. L. Bonilla, C. J. P. Vicente, F. Ritort, and R. Spigler, "The kuramoto model: A simple paradigm for synchronization phenomena," *Reviews of modern physics*, vol. 77, no. 1, p. 137, 2005.
- [36] C. Liu, Y. Chen, J. Zhang, and Q. Chen, "Cpg driven locomotion control of quadruped robot," in *2009 IEEE International Conference on Systems, Man and Cybernetics*. IEEE, 2009, pp. 2368–2373.
- [37] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *Mobile Service Robotics*. World Scientific, 2014, pp. 433–440.
- [38] C. Liu, Q. Chen, and G. Wang, "Adaptive walking control of quadruped robots based on central pattern generator (cpg) and reflex," *Journal of Control Theory and Applications*, vol. 11, no. 3, pp. 386–392, 2013.
- [39] A. A. Saputra, A. J. Ijspeert, and N. Kubota, "A neural primitive model with sensorimotor coordination for dynamic quadruped locomotion with malfunction compensation," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 3783–3788.
- [40] M. Calandra, L. Patanè, T. Sun, P. Arena, and P. Manoonpong, "Echo state networks for estimating exteroceptive conditions from proprioceptive states in quadruped robots," *Frontiers in neurobotics*, p. 118, 2021.