Summary of replies to the editor and reviewer comments

Tao Sun, Zhendong Dai, Poramate Manoonpong

RC: ***Reviewer Comment***, **AR:** ***Author Response***

We would like to thank the editor and reviewers for their thorough review and comments, as well as for their substantial praise of the work. Here follows our replies to editor's and all reviewers' comments. The editor's and reviewers' comments are shown in blue. Our answers are shown in black. The (modified and added content) in the revised manuscript is shown in red.

1. Editor

The paper received three review reports. On one hand, the reviewers found that the paper presents some interesting results that are publishable. On the other hand, however, the reviewers made a number of critical comments on the paper. Therefore, based on the reviewers' comments, the authors are encouraged to examine carefully the comments of the reviews and to consider making a major revision of the paper that can be submitted for further considerations in journal. In addition, the following comments should be also considered by authors:

1) the readability and presentation of the work should be improved.

AR: We have now improved the readability and presentation of the work in the revised manuscript.

2) the main contributions of the work should be clearly explained in both Theoretical and Practical aspects,

AR: We have now explained the main contributions of the work clearly in the revised manuscript as:

"The main contributions of this work are as follows:

1. A novel bio-inspired reflex mechanism with fast online learning (i.e., DFRL), which provides CPG-based control with an offset adaptation function, for adaptive body posture corresponding to diverse slopes. Compared with the quadruped locomotion control based on classical techniques [1, 2, 3], the DFRL with CPG-based control does not require any robot kinematics and environmental model. Thus, it is more practical. In principle, the proposed DFRL based on biological mechanisms is characterized by the independence of specific robots and CPG-based control. It provides a generic offset adaptation method which can be integrated with different CPG models (e.g., SO(2) CPG and dynamical movement primitives (DMPs)) for controlling different sized and weighted quadruped robots.
2. A demonstration involving quadruped robots with the proposed reflex for adaptive body posture to navigate ascending and descending steep slopes (i.e., $\pm 35^{\circ}$ for a small robot and 50° and -45° for a larger one), as well as a complex terrain with multiple slopes using a trot gait (see Fig. 1). This in our knowledge is an advanced achievement in quadruped slope locomotion based on reflex mechanisms.
3. A comparison of the performance between the traditional vestibular reflex (using the telescoping strut strategy) and the proposed DFRL (using the lever mechanics strategy) for adaptive quadruped locomotion on various slopes.

4. A possible option for integration and interaction of CPGs, sensory feedback, reflexes, and motor learning. This contributes to better understanding of biological locomotion mechanisms and development of the reflex-based quadruped locomotion control with great adaptability."

3) the dissemination of the results should be further explained and compared with existing results.

AR: We have now added more information about the dissemination of our results in the discussion section of the revised manuscript as:

"We developed the control software based on a standard language (C++) with the robot operating system (ROS) interface. Thus it can be applied to ROS-based quadruped robots. To maximize the contribution and dissemination of our study to the research community, we also provide our control method as open-source software which can be accessed at https://gitlab.com/neutron-nuaa/dfrl."

We have also discussed our results with existing ones as:

" Alongside the flexibility and adaptability realized by the DIL, the DFRL is also intrinsically modular and independent of the CPG-model format or the size and weight of the robot. The DFRL can be incorporated with SO(2)-based CPGs (see Fig. 5) or with dynamical movement primitives-based CPGs [4, 5] (see Supplementary material), suggesting that the DFRL is generic and applicable to different CPG models. Thus, it is readily integrable with diverse CPG-based controls and also facilitates generic adaptive quadruped motor control with CPG or joint offset adaptation for different quadruped robots.

. . . . . .

Vestibular reflexes have been proposed in several works; these facilitate adaptive dynamic running over unperceived slopes, where the slopes are regarded as disturbances [6, 7, 8, 9, 10, 11]. More specifically, the vestibular reflex is used to shift the offsets of joint movement commands (originally produced by CPGs), thereby extending or flexing the legs to maintain the robot body parallel to the horizontal [see Fig. A.3 (b)]. In addition, Zhao et al. applied a vestibular reflex to adjust the frequency and amplitude of the joint movement commands [12]. This allows the quadruped robot AIBO to trot steadily over slight slopes using a high step frequency and short step length. Despite the successful implementation of vestibular reflexes in these cases, all were limited to gentle slopes (below $20°$, see Table 1).

In contrast to vestibular reflexes, the DFFB reflex can translate the ZMP coordination into a proper position along the slope direction. Thus, the robot body's orientation is aligned to the slope when the stability margin is increased [see Fig. A.3 (c)]. As a result, the front and hind legs are in a far more natural position to distance themselves from singular configurations or joint limits. Thus, these robots can adapt to steeper slopes. The two distinct strategies (body orientation parallel to the horizontal and parallel to the slope) are referred to in biomechanics as the "telescoping strut" and "lever mechanics", respectively [13]. Unfortunately, the vestibular reflex cannot realize the lever mechanics strategy because it requires a body orientation parallel to the slope rather than the horizontal. The DFFB reflex is based on the lever mechanics strategy and enables quadruped robots to trot on steep-sloped terrains (e.g., $35°$ for Lilibot and $-45°$ and $50°$ for Laikago).

Table 1: The maximum slope to which robots can adapt based on vestibular reflex modulation.

Works	Year of publication	Max. slope [degree]
Xiuli Zhang et. al. [7]	2008	Around 11.2
Mostafa Ajallooeian et. al. [9]	2013	11.85
Duc Trong Tran et. al. [14]	2014	Around 11
Chengju Liu et. al. [6]	2018	Around 12

”

2. Reviewer #1

This paper represents a new distributed force feedback-based reflex with online learning, called DFRL. A new bio-inspired reflex mechanism with fast online learning is proposed, which provides CPG-based control with an offset adaptation function, for adaptive body posture corresponding to diverse slopes. The performed three experiments on a small-sized quadruped robot. The video of the experiments was interesting. http://www.manoonpong.com DFFB/video1.mp4 . The results are good, I have some comments as below:

RC: Figure 1, “distributed force feedback-based reflex with online learning” is not reflected in this figure.

AR: We have now improved the figure and added more accurate information to its caption, as shown in the new figure (Fig. 1). We have now inserted this new figure (Fig. 1) into the revised manuscript.

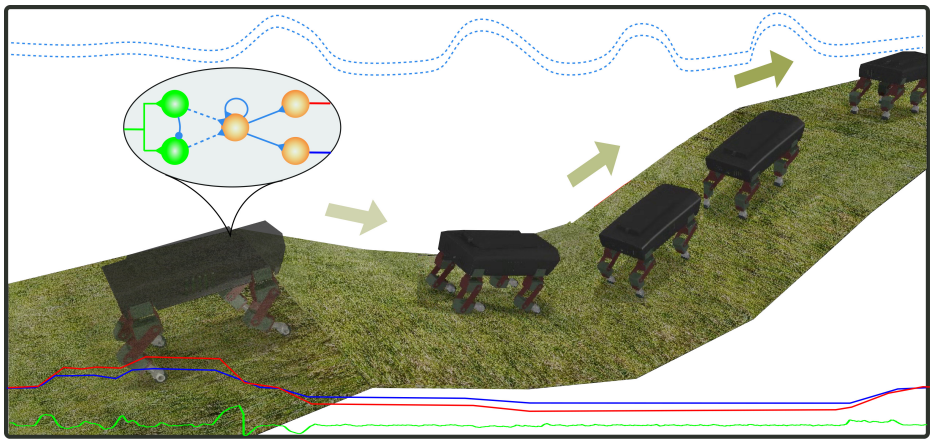

Figure 1: A quadruped robot successfully trots on a complex terrain consisting of multiple slopes. It is driven by the distributed force feedback-based reflex with online learning represented by a simplified neural diagram (see circle). The reflex circuit has two plastic synaptic weights (dashed lines projecting from the two input neurons to the hidden neurons in the diagram). The weights are adapted online by a fast learning mechanism. The reflex is stimulated by the ground reaction force (GRF) distribution (green line projecting to the two input neurons in the diagram). Its output neurons generate adaptive knee and hip joint commands (red and blue lines in the diagram, respectively). The adaptive commands enable the robot to trot stably on the terrain. The blue dashed lines above the robot and the green, red and blue lines below the robot describe the weight adaptations, the change of the GRF, the change of the knee joint offset, and the change of the hip joint offset, respectively, during walking on the terrain.

RC: Line 83, why the recurrent neural network organized in three layers? Is this an optimized parameter of the RNN

AR: Our answer relates to the question in two ways, enumerated as follows:

1. Network structure design: The recurrent neural network (the DFFB network, see Fig. 5 below or in the revised manuscript) was designed to map the difference between the actual and desired GRF distributions ($\bar{\gamma}^a - \bar{\gamma}^d$) to the desired joint command offsets (β). To achieve this, the DFFB network must perform a minimum of three necessary functions: (i) it must calculate the difference between the actual and desired GRF distributions, (ii) it must calculate the desired joint command offsets according to this difference, and (iii) it must generate the proper offsets at the knee and hip joints. The functions must be executed sequentially. Thus, they are realized independently in three layers (Layer 1, Layer 2, and Layer 3).

   More specifically, Layer 1: The N1 neuron of the input layer calculates the difference between the actual and desired GRF distributions ($\bar{\gamma}^a$ and $\bar{\gamma}^d$, respectively). In our setup, the desired value is 1.1 [see Fig. 4 (b) in the revised manuscript], as defined by the bias term. Accordingly, the difference equation is given by $\Delta\bar{\gamma}(n) = \bar{\gamma}^a(n) - 1.1$. The N2 neuron of Layer 1 calculates the second-order difference. Its formula is $\Delta\bar{\gamma}'(n) = \Delta\bar{\gamma}(n) - \Delta\bar{\gamma}(n-1)$. Thus, the synaptic weight projection from N1 to N2 is set to -1. The use of the second-order difference can partially compensate for the delay effect on $\bar{\gamma}$ produced by the low-pass filters (see L1, L2, and L3 in Fig. 5).

Layer 2: The N3 neuron of the hidden layer adds two differences ($\Delta\bar{\gamma}(n)$ and $\Delta\bar{\gamma}'(n)$) via the two plastic synapses (w_1 and w_2), and it accumulates them using a recurrent connection (i.e., 1.0).

Layer 3: The N4 and N5 neurons of the output layer properly re-scale the offsets to the knee and hip joints of the leg. To this end, the network is manually designed with the minimum structure required to meet the desired function.

2. Network control parameters: The two key synapse parameters (w_1 and w_2, Fig. 5) are online-adapted through a dual integral learner (DIL); this can identify appropriate values of the parameters, thereby maximizing the performance of the DFRL [see the plots (with and without DIL) in Fig. 6 of the revised manuscript]. The value of w_3 is approximately twice that of w_4 (e.g., w_3=2, w_4=1). The network input and output weights are set to 1.0, to directly receive input information ($\bar{\gamma}^a$) and directly transfer outputs to the motor neurons. The output magnitudes were indirectly tuned by the plastic synapses (w_1 and w_2). Therefore, the DIL can online-optimize the overall network performance.

We have now improved Fig. 5 and added this information to the revised paper, as follows:

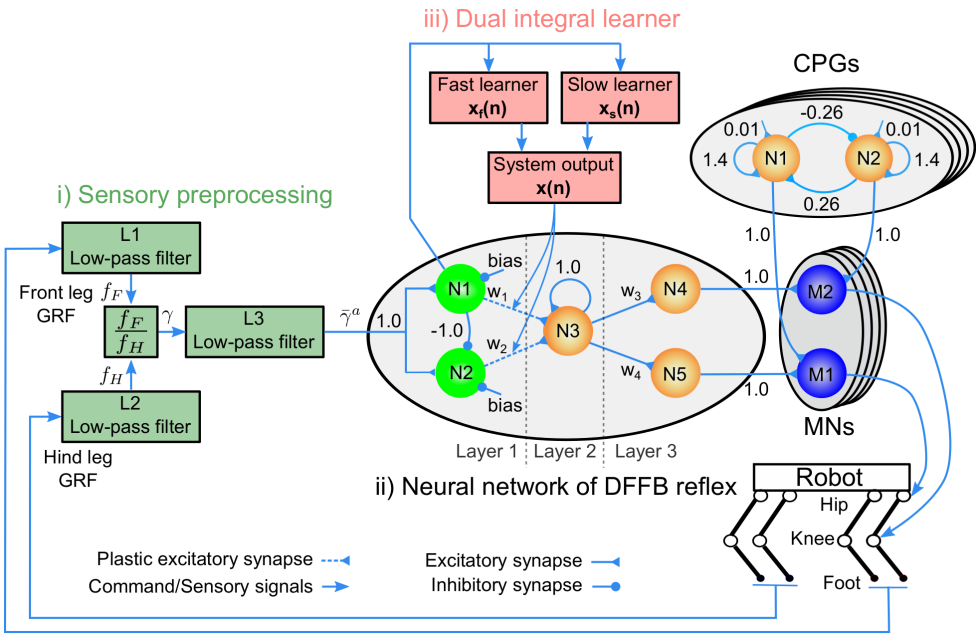

Figure 5: Schematic diagram of the DFRL, featuring three main components: (i) sensory preprocessing, (ii) DFFB reflex network, and (iii) DIL. Sensory preprocessing calculates the actual GRF distribution (γ) using the front and hind GRFs, which are filtered by two infinite impulse response low-pass filters (L1 and L2 low-pass filters); then, it smooths γ using a moving average filter (L3 low-pass filter) to acquire the actual smoothed GRF distribution $\bar{\gamma}^a$, which triggers the reflex network. The DFFB reflex network is organized into three layers (Layers 1, 2, and 3) with five neurons (N1, N2, N3, N4, and N5). It can automatically adapt the knee and hip-joint offsets at MNs. Two plastic synapses (w_1 and w_2) of the network are online-modulated by the DIL. The biases of the Layer 1 neurons (N1 and N2) represent the desired GRF distribution ($\bar{\gamma}^d$). In the following experiments, they are set to 1.1, depending on the particular step length and period. The synapse (w_3 and w_4) projections from the Layer 2 neuron (N3) to the Layer 3 neurons (N4 and N5) were set to 2.0 and 1.0, respectively. Note: w_3 is here set to twice that of w_4 because the foot displacement moved by the hip joint will be approximately twice that moved by the knee joint if the two joints receive the same command values. Therefore, we compensate for this by setting the knee-joint's command value twice that of the hip joint, by setting the w_3 value to double the w_4 value.

"The DFFB reflex neural network employs the smoothed GRF distribution ($\bar{\gamma}^a$) as its sensory input, using it to trigger neural network activation. The network consists of three layers: input (Layer 1), hidden (Layer 2), and output (Layer 3). As shown in Fig. 5, the N1 neuron in Layer 1 calculates the difference between the actual and desired GRF distributions ($\bar{\gamma}^a$ and $\bar{\gamma}^d$, respectively). For our setup, the desired value is 1.1, as defined by the bias term [see Fig. 4 (b)]. Accordingly, the difference equation is given by $\Delta\bar{\gamma}(n) = \bar{\gamma}^a(n) - 1.1$. The N2 neuron calculates the second-order difference. Its formula is $\Delta\bar{\gamma}'(n) = \Delta\bar{\gamma}(n) - \Delta\bar{\gamma}(n-1)$. Thus, the synaptic weight projection from N1 to N2 was set to -1. The use of the second-order difference can partially compensate for the delay effect on $\bar{\gamma}$ produced by the low-pass filters [see L1, L2, and L3 in Fig. 5]. This is because the second-order difference—which reflects the change tendencies of the difference—can adjust the DFFB outputs in advance. In Layer 2, the N3 neuron adds two differences via the two plastic synapses (w_1 and w_2) and accumulates them via a recurrent connection (i.e., 1.0). In Layer 3, the N4 and N5 neurons

properly re-scale the offsets to the knee and hip joints of the leg. Thus, the outputs of N4 and N5 (o_4^{dffb} and o_5^{dffb}, respectively) are combined with the CPG outputs at the M2 and M1 neurons. o_4^{dffb} and o_5^{dffb} are set to the joint command offsets [$\boldsymbol{\beta}$; see Eq. (7)]."

RC: Line 84: If the weights in the network are plasticity weights, if also the leading algorithm time-dependent plasticity?

AR: We follow the neuroscientific terminology "plastic weights/synapses" and "synaptic plasticity," which describe the ability of synapses to strengthen or weaken over time [15]. However, the learning mechanism we employ is not based on a time-dependent learning rule [i.e., spike-timing-dependent plasticity (STDP)]. Instead, it relies on an error function (i.e., the difference between the actual and desired GRF distributions).

To this end, we treat the weights (w_1 and w_2) in the network as plastic weights (i.e., with synaptic plasticity) that are adapted or changed over time, rather than as static ones. However, they do not feature time-dependent plasticity (in terms of STDP) because their changes rely on an error function of the DIL rather than relative spike timings.

We have now clarified and supplemented this information in the revised paper, as follows (Line 84):

"The DFFB reflex is implemented through a simple recurrent neural network organized in three layers. The key synaptic weights in the network are plastic and adapted or changed over time by a fast online learning mechanism, called "dual integral learne" (DIL) [16]. . . . . It is important to note that the plastic weights here are not spike-timing-dependent plasticity (STDP) because their changes rely on an error function of the DIL rather than relative spike timings"

RC: Line 134: what is the computational model of the "time-discrete neuron", how the neuron deals with input temporal data? Is this based on activation function of action potential?

AR: 1) Here, "time-discrete neuron" means that the CPG neuron is modeled using discrete-time dynamics [17], which are expressed via the difference equations [see Eqs. (1) and (2)]. 2) (Temporal) Input data to the neurons are transmitted thereto as a discrete time series and inputted at each time step (n). n denotes a discrete time series with an update frequency of 60 Hz. 3) Each neuron has a nonlinear activation function [i.e., a hyperbolic tangent (tanh) transfer function]. At each time step, the neuron inputs are multiplied by the input weights ($\mathbf{w}$) and summed to obtain the neuron activations ($\mathbf{a}$). Next, the activations are transformed to produce the outputs ($\mathbf{o}$) via the activation function ($\mathbf{o} = \tanh(\mathbf{a}) \in [-1, 1]$). This process is updated in each iteration or step.

We have now inserted this information into the revised paper, as follows:

"As shown in Fig. 3 (b), the neural SO(2)-based CPG is a recurrent neural network. It consists of two fully connected neurons (N1 and N2). The neurons are modeled in time-discrete dynamics using difference equations [17]. Their activation function is a hyperbolic tangent (tanh) function, expressed as follows:
. . .
Thus, the neurons' input data consist of a discrete-time series provided to them at each time step (n). Here, n denotes discrete time with an update frequency of 60 Hz. Each neuron features a nonlinear activation function [i.e., a hyperbolic tangent (tanh) transfer function]. At each time step, the neuron inputs are multiplied by the input weights ($\mathbf{w}$) and summed to obtain the neuron activations ($\mathbf{a}$). Next, the activations are transformed to produce the outputs ($\mathbf{o}$) via the activation function ($\mathbf{o} = \tanh(\mathbf{a}) \in [-1, 1]$). This process is updated at each iteration or step."

RC: Line 148: what is the range of b?

AR: Here, **b** denotes the biases of the SO(2)-based CPG neurons; it is a constant matrix that initializes CPG activity with a periodic pattern (i.e., the neural dynamics of the CPG are those of a quasi-periodic attractor). To obtain periodic dynamics, it must be set within a small range of -0.085 to 0.085. Setting it to a larger positive or negative value will drive the neural dynamics to those of, for example, a fixed point attractor, which results in constant CPG behavior.

We have now included this information in the revised paper, as follows:

"The biases are constant, which generates the initial activity of the CPG neurons in accordance with a periodic pattern (i.e., the neural dynamics of the CPG are those of a quasi-periodic attractor); their values can be set within the small range of -0.085–0.085. Note: setting them to a larger positive or negative value will drive the neural dynamics to, for example, a fixed point attractor, resulting in constant CPG activities."

RC: Figure 3: are the input and output neurons directly connected? How the model was initialized? More information about the learning algorithm is required.

AR: In our control architecture (see also Fig. 2), the input neurons are "sensory input neurons" and the output neurons are "motor output neurons." Thus, the input and output neurons are not directly connected. The sensory inputs are projected, to control the motor neurons through the DFFB reflex network. Alongside this, a CPG-based control with four identical SO(2)-based CPGs transmits its periodic signals to the motor neurons. The outputs of the motor neurons are finally applied to control the knee and hip joints of the robot. Thus, whilst the CPGs make the joints move periodically, the sensory inputs in the DFFB reflex network control the joint offsets, facilitating stable locomotion on different slopes.

In this control setup, each CPG model features four synaptic weights (w_{12}, w_{21}, w_{11}, and w_{22}) and two bias terms [b_1 and b_2, Fig. 3 (b)]. The weights and bias terms are empirically set such that the CPG generates stable periodic signals. The connections between the CPGs are also predefined according to a phase-locked function of distributed oscillators [18], to obtain a trot gait (described in brief below). This type of function is widely employed for interactions (e.g., phase locking) in oscillator network systems [18, 19, 20, 21, 22]. Here, the inputs a CPG neuron receives from other CPG neurons are modeled and described by the term $\mathbf{g} = (g_{il}) \in \mathbb{R}^{2\times 4}$ in the CPG model [see Eqs. (1) and (5) in the manuscript]. g_{il} is given by

$$g_{il}(n) = \xi \sum_{k=1}^{4} \left(\sin\left(o_{il}(n) - o_{ik}(n) - \phi_{lk}\right)\right), \tag{5}$$

where $o_{il}(n)$ and $o_{ik}(n)$ are the outputs of the i-th neurons in CPG l and CPG k. ξ is a communication gain that is empirically set to 0.01. ϕ_{lk} is the desired relative phase of CPG k with respect to CPG l. For instance, if we set the right-front leg's CPG (CPG 1) in anti-phase to the right-hind leg's CPG (CPG 2), then $\phi_{21} = \pi$ and $\phi_{12} = -\pi$. If they are in phase, then $\phi_{12} = 0$ and $\phi_{21} = 0$. For a trot gait, the diagonal legs move in phase while in anti-phase with the other legs. Thus, the relative phases between the CPGs for the trot gait are set as

$$\mathbf{\Phi} = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} \end{pmatrix} = \begin{pmatrix} 0.0 & -\pi & -\pi & 0.0 \\ \pi & 0.0 & 0.0 & \pi \\ \pi & 0.0 & 0.0 & \pi \\ 0.0 & -\pi & -\pi & 0.0 \end{pmatrix}. \tag{6}$$

Note: all the parameters of the CPG-based control are held constant after initialization; that is, no learning

algorithm is applied to the CPG-based control.

This work focuses on the development of the distributed-force-feedback-based reflex with online learning (DFRL), and the only learning algorithm used here is the DIL, which has already been described in detail in the revised manuscript [see Section 2.2.3 and Eq. (13)].

We have clarified Fig. 3 and inserted the above information in the revised paper, as follows:

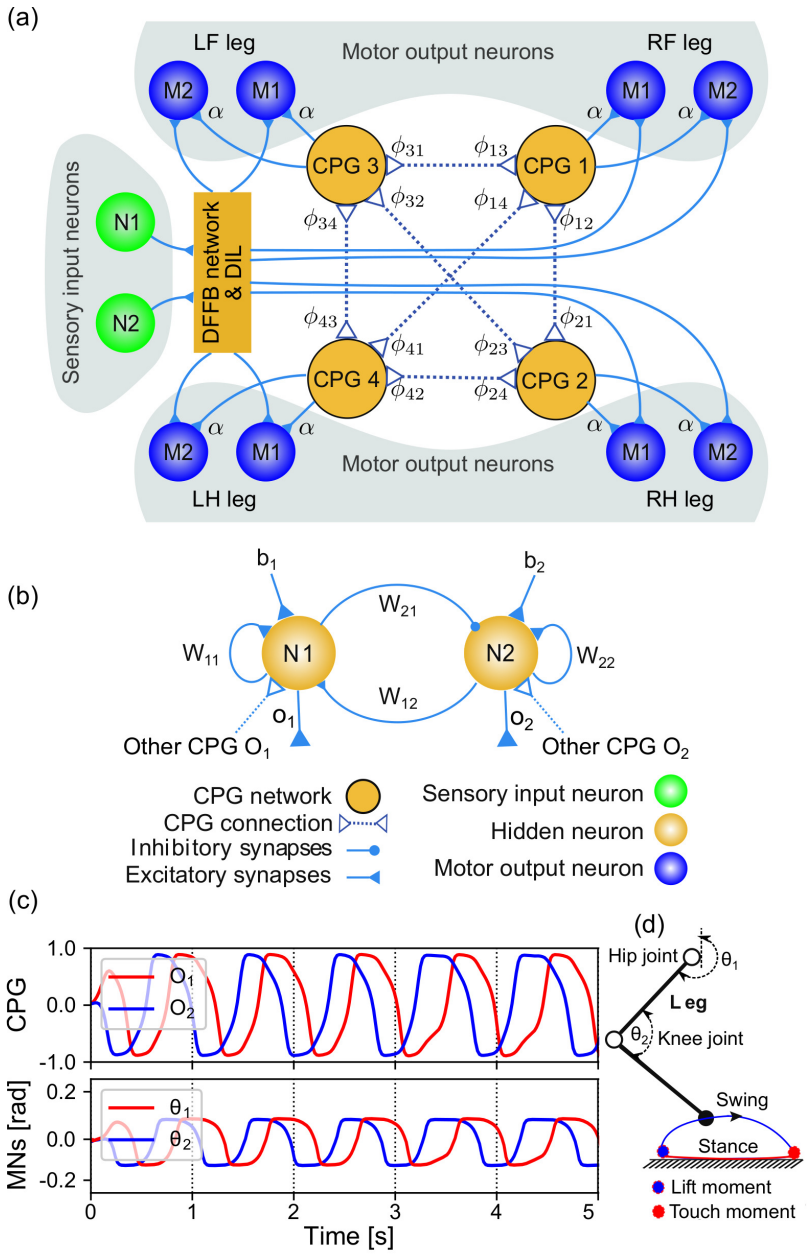

Figure 3: CPG-based control. (a) Four identical CPGs are fully connected via the coupling parameters ϕ_{ij}, which represent CPG phase relationships. Their outputs are sent to four groups of motor neurons (M1 and M2) through the synaptic weights α, to control the four legs' movements. The motor neurons integrate the DFFB network outputs, triggered by sensory feedback and CPG outputs. (b) The CPG is based on the SO(2) neural oscillator, which is a recurrent neural network with two neurons (N1 and N2); these neurons feature internal neural connections (i.e., w_{ij}) and receive inputs from the corresponding neurons of the other CPGs (dashed lines). Each CPG produces two outputs (o_1 and o_2) with a fixed phase shift $\pi/2$. (c) The outputs of the CPG and MNs of a leg. The MN outputs (θ_1 and θ_2) send the frequency, waveform, phase, and offset of the CPG outputs (o_1 and o_2) to the robot joints. The amplitudes of the MN outputs are scaled by α-projecting from the CPGs to MNs (e.g., 0.16 for Lilibot[1] and 0.12 for Laikago[3]) to determine a particular step length. (d) Foot trajectory formed under MN outputs. Ideally, the trajectory has four states: lift moment, touch moment, swing phase, and stance phase.

"More specifically, the DFFB reflex network projects the sensory inputs to control the motor neurons. Simultaneously, the CPG-based control (featuring four identical SO(2)-based CPGs) transmits periodic signals to the motor neurons. Finally, the outputs of the motor neurons control the robot's knee and hip joints. Thus, whilst the CPGs make the joints move periodically, the sensory inputs and DFFB reflex network control the joint offsets, to facilitate stable locomotion over different slopes.

...

As shown in Fig. 3 (b), the neural SO(2)-based CPG is a recurrent neural network. It consists of two fully connected neurons (N1 and N2). The neurons are modeled in time-discrete dynamics using difference equations [17]. Their activation function is a hyperbolic tangent (tanh) function, expressed as follows:

$$\mathbf{a}(n+1) = \mathbf{w} \cdot \mathbf{o}(n) + \mathbf{b} + \mathbf{g}(n), \tag{1}$$

$$\mathbf{o} = \tanh(\mathbf{a}). \tag{2}$$

Here, $\mathbf{a}$, $\mathbf{o}$, and $\mathbf{b} \in \mathbb{R}^{2\times 4}$ represent the activations, outputs, and biases of the CPG neurons, respectively. Note: each column of the matrices represents the state variables of a CPG. The biases are constant, which generates the initial activity of the CPG neurons in accordance with a periodic pattern (i.e., the neural dynamics of the CPG are those of a quasi-periodic attractor); their values can be set within the small range of -0.085–0.085. Note: setting them to a larger positive or negative value will drive the neural dynamics to, for example, a fixed point attractor, resulting in constant CPG activities. $\mathbf{w} \in \mathbb{R}^{2\times 2}$ denotes the synaptic weights between the two neurons of the neural SO(2)-based CPG. Each CPG model has four synaptic weights (w_{12}, w_{21}, w_{11}, and w_{22}) and two bias terms [b_1 and b_2, Fig. 3 (b)]. The weights and bias terms are empirically set such that the CPG generates two stable periodic signals [o_1 and o_2, Fig. 3 (b)]. The two outputs have a stable phase shift $\pi/2$ between them, thereby realizing the intralimb coordination of a leg (i.e., the joints of the leg move coordinately) [23].

...

In the following experiments, the biases, weights, and coupling terms are set according to the neural dynamics described in [17]:

$$\mathbf{b} = \begin{pmatrix} 0.01 & 0.01 & 0.01 & 0.01 \\ 0.01 & 0.01 & 0.01 & 0.01 \end{pmatrix}, \tag{3}$$

$$\mathbf{w} = \begin{pmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{pmatrix} = \begin{pmatrix} 1.4 & 2.6 \\ -2.6 & 1.4 \end{pmatrix}. \tag{4}$$

The four neural SO(2)-based CPGs are fully connected via parameters ϕ_{ij}, which represent CPG phase relationships. To generate a trot gait, the inter-CPG connections are also predefined based on a phase-locked function of distributed oscillators [18]; such functions are widely employed for interactions (e.g., phase locking) in oscillator network systems [18, 19, 20, 21, 22]. Here, the inputs delivered to one CPG neuron from other CPG neurons are modeled and described via the term $\mathbf{g} = (g_{il}) \in \mathbb{R}^{2\times 4}$ in the CPG model [see Eqs. (1) and (5)]. g_{il} is given by

$$g_{il}(n) = \xi \sum_{k=1}^{4} \left(\sin\left(o_{il}(n) - o_{ik}(n) - \phi_{lk}\right)\right), \tag{5}$$

[2]Lilibot is a small-sized quadruped robot used in this study (see Fig. 7 and Appendix).
[3]Laikago is a large quadruped robot used in this study (see Fig. 7 and Appendix).

where $o_{il}(n)$ and $o_{ik}(n)$ are the outputs of the i-th neuron in CPG l and CPG k, respectively. ξ is a communication gain that is empirically set to 0.01. ϕ_{lk} is the desired relative phase of CPG k with respect to CPG l. For instance, if we set the right-front leg's CPG (CPG 1) in anti-phase to the right-hind leg's CPG (CPG 2), then $\phi_{21} = \pi$ and $\phi_{12} = -\pi$; if they are in phase, then $\phi_{12} = 0$ and $\phi_{21} = 0$. For a trot gait, the diagonal legs move in phase but in anti-phase with the other legs. Thus, the relative phases between the CPGs under the trot gait are set as

$$\mathbf{\Phi} = \begin{pmatrix} \phi_{11} & \phi_{12} & \phi_{13} & \phi_{14} \\ \phi_{21} & \phi_{22} & \phi_{23} & \phi_{24} \\ \phi_{31} & \phi_{32} & \phi_{33} & \phi_{34} \\ \phi_{41} & \phi_{42} & \phi_{43} & \phi_{44} \end{pmatrix} = \begin{pmatrix} 0.0 & -\pi & -\pi & 0.0 \\ \pi & 0.0 & 0.0 & \pi \\ \pi & 0.0 & 0.0 & \pi \\ 0.0 & -\pi & -\pi & 0.0 \end{pmatrix}. \tag{6}$$

ϕ_{lk} determines the interlimb coordination or walking pattern. Optimizing this parameter can allow the robot to realize different gaits, which can enhance its performance under certain conditions. However, in this study, the parameter was predefined to produce a trot gait without further gait optimization; this design was based on a phase-lock function for coupling oscillators."

RC: Line 153: How parameter optimization can be useful here?

AR: In Line 153, two parameters are discussed: Φ and β. We explain them separately, as follows:

1. Φ represents the CPG phase relationships, which determine the interlimb coordination or walking pattern. Optimizing this parameter can allow the robot to perform different gaits, which could enhance its performance under certain conditions. However, in this study, this parameter was predefined to produce a trot gait without further gait optimization. It was designed based on a phase-lock function for coupled oscillators.

2. β represents the joint command offset. In this work, we optimize this parameter using the proposed reflex (distributed-force-feedback-based reflex, see Fig. 5). Optimizing the parameter β provides the robot with proper body posture for stable locomotion on various slopes, as shown in the experiments.

We have now added the explanation in the revised manuscript, as follows:

"ϕ_{lk} determines the interlimb coordination or walking pattern. Optimizing this parameter can allow the robot to realize different gaits, which can enhance its performance under certain conditions. However, in this study, the parameter was predefined to produce a trot gait without further gait optimization; this design was based on a phase-lock function for coupling oscillators."

"$\boldsymbol{\beta}$ denotes the command offsets and determines the robot-joint offsets required to set the robot body posture. In this work, we optimize this parameter using the proposed DFRL (see Fig. 5). Optimizing the parameter β provides the robot with a proper body posture for stable locomotion over various slopes."

RC: Lines 175-180: what are h and f?

AR: The subscript h and f denote the hind and front legs, respectively. Therefore, in Lines 175—180, F_h and F_f represent the hind and front leg ground reaction forces (GRFs), respectively. This is indicated in Line 172 and illustrated in Fig. 4.

To express this more clearly, we have added the relevant information to the revised manuscript, as follows:

"Here, F_f and F_h are the GRFs of the front (f) and hind (h) legs, respectively. By combining Eqs. (8) and (9), γ can be conveniently calculated online from the GRF signals measured by the foot force sensors, as follows: "

RC: Line 255: I don't think equation 14 is showing the plasticity aspect of the weights.

AR: Eq. (14) is $\Delta w_i(n) = x(n), i = 1, 2$. $\Delta w_i(n)$ is the change in the synaptic weight w_i of the distributed-force-feedback-based (DFFB) reflex network. $x(n)$ is the output of the dual integral learner (DIL), which is set directly to change the weights. It is derived from Eq. (13) of the revised manuscript.

The plasticity is based on the error $e(n)$, which is the N1 neuron's output in the DFFB reflex network. The N1 neuron output expresses the difference between the actual and desired smoothed GRF distributions ($\bar{\gamma}^a$ and $\bar{\gamma}^d$, respectively). As shown in Fig. 6 (b) of the manuscript, the synaptic weights (w_1 and w_2) are adapted over time according to the difference between $\bar{\gamma}^a$ and $\bar{\gamma}^d$. When the actual distribution $\bar{\gamma}^a$ gradually converges to the desired $\bar{\gamma}^d$ (i.e., 1.1), the changes in the synaptic weights become small and are stabilized to some extent.

Here, we use the neuroscientific terminology "plastic weights/synapses" and "synaptic plasticity," which refers to the ability of synapses to strengthen or weaken (change) over time [15]. However, the learning mechanism we employ is not based on a time-dependent learning rule [i.e., spike-timing-dependent plasticity (STDP)]. Instead, it relies on an error function (i.e., the difference between the actual and desired GRF distributions, as described above).

To this end, we treat the weights (w_1 and w_2) in the network as plastic weights (i.e., synaptic plasticity) that are adapted or changed over time, rather than as static ones. However, they do not feature time-dependent plasticity in terms of STDP because their changes rely on the error function of the DIL rather than relative spike timings.

We have now clarified and inserted this information in the Introduction of the revised paper, as follows:

"The DFFB reflex is implemented through a simple recurrent neural network organized in three layers. The key synaptic weights in the network are plastic and adapted or changed over time by a fast online learning mechanism, called dual integral learner (DIL) [16]. ..... It is important to note that the plastic weights here are not spike-timing-dependent plasticity (STDP) since their changes rely on an error function of the DIL rather than relative spike timings."

RC: Line 306 and Figure 9: any justification of the setting different C values? Are there any other conditions that you did not include?

AR: The C values represent different conditions ($C_1\ C_2\ C_3 \ldots C_7$) in which the initial joint offsets ($\beta_{1,2}(0)$) differ ($\beta_{1,2}(0) = -0.3$, $\beta_{1,2}(0) = -0.2 \ldots \beta_{1,2}(0) = 0.3$) (see Fig. 9). We set these different initial joint offsets to evaluate the posture stability of the robot when implementing the DFRL (see Fig. 5 in the paper). To conduct a full evaluation, the initial values of $\beta_{1,2}$ were set within the range of -0.3 to 0.3. This range covers all possible joint movement ranges. If the initial offsets lie outside this range, the robot will fall down. For example, if the value exceeds 0.3, the robot will lean too far forward; if the value is smaller than -0.3, the robot will lean too far backward. In both extreme cases, the center of mass of the robot will shift to outside the supporting area of the robot's legs, thereby causing it to fall down. In the range [-0.3, 0.3], we chose seven values that are sufficient to validate and demonstrate the performance of our control method. We have now added this information to the figure caption, as follows:

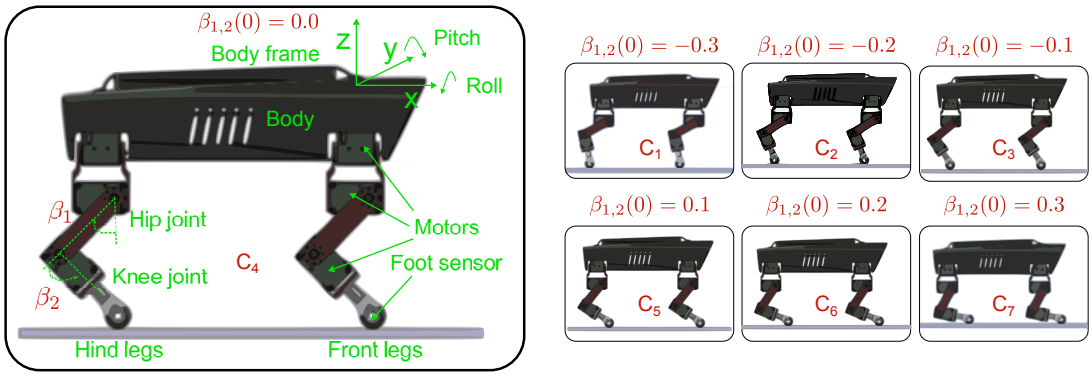

Figure 9: Lilibot configuration and its posture initializations with corresponding joint offsets ($\beta_{1,2}$) in seven conditions (C_1, C_2, C_3, C_4, C_5, C_6, and C_7). The initial values of $\beta_{1,2}(0)$ were set in the range -0.3–0.3, which covers all possible joint-movement ranges. If the initial offsets ($\beta_{1,2}(0)$) lie outside this range, the robot will fall down. For example, if the value exceeds 0.3, the robot will lean too far forward; if it is smaller than -0.3, the robot will lean too far backward. In both extreme cases, the robot's center of mass will be outside the supporting area of the robot legs, causing it to fall.

3. Reviewer #2

In this paper the authors present an adaptive CPG based locomotion control system. They compare the results of their reflexive controller to others on slope-traversing tasks, showing the improvements in performance they achieve. I enjoyed reading this paper, it was well written and structured well. Throughout the paper there are occasional grammar issues that make it difficult to parse, so it should be reviewed and have grammar issues fixed before publication. I have highlighted some of these examples, but just a subset.

AR: We have now employed a proofreading service to review the paper and check for grammar issues.

Main comments:

RC: I would like to see comparison with Dynamic Movement Primitives that use sensory feedback control, although non-adaptive I think it would be helpful background discussion.

AR: We have now included a comparison between the DFRL [with the SO(2)-based control] and dynamical movement primitives [4, 5] (DMPs)-based controls. In this comparison, the DMPs serve as CPGs, replacing the SO(2)-based CPGs used in this study. The DMPs are trained by local weight regression to match the SO(2)-based CPGs. Subsequently, the DMP-based control incorporating the DFRL was implemented in a quadrupedal robot walking on level ground. Experimental results show that the DMP-based DFRL control can also obtain appropriate joint offsets for robot-posture stabilization. This suggests that the DFRL is generic and can be applied to different CPGs or oscillators [e.g., SO(2) and DMPs]. In principle, the DFRL operates as a neural module that can be directly used in different types of CPGs to adjust a quadruped robot's body posture and allow it to stably walk on different slopes.

We have now provided the results of the DMP-based DFRL control experiment in the Supplementary material.

Furthermore, we have provided the following information in the Discussion section of the revised manuscript:

"Alongside the flexibility and adaptability realized by the DIL, the DFRL is also intrinsically modular and independent of the CPG-model format or the size and weight of the robot. The DFRL can be incorporated with SO(2)-based CPGs (see Fig. 5) or with dynamical movement primitives-based CPGs [4, 5] (see Supplementary material), suggesting that the DFRL is generic and applicable to different CPG models. Thus, it is readily integrable with diverse CPG-based controls and also facilitates generic adaptive quadruped motor control with CPG or joint offset adaptation for different quadruped robots."

RC: It seems like (comparing part 1 and 2 of video 4) that for steeper terrain the quad is using smaller step sizes at the beginning (before it gets to the steep terrain). In reading the last paragraph of the discussion it seems like this may be caused by the bias of the joints being near saturation points, but I think more discussion of this in analyzing the experiments would be helpful / of interest.

AR: We have now expanded the discussion of this point in the revised manuscript, as follows:

"The stability margin for quadruped locomotion can be defined as the minimum distance between the ZMP and support-polygon boundaries [24, 25]. The stability margin of locomotion on level ground is relatively larger than that on slopes. This is because the ZMP approaches one set of feet when traversing slopes (e.g., the ZMP approaches the hind feet on uphill surfaces and the front feet on downhill ones, see Figs. 4 and A.1). Locomotion stability is affected not only by the joint offsets but also the step length and period (see Fig. 4). The step length determines the motion distance of the ZMP during one step period. In some cases (e.g., when trotting on steep slopes), a small step length is necessary to prevent the ZMP moving outside of the support polygon. In the proposed CPG-based control, the MNs can scale the joint command amplitudes to reduce the step length and height on slopes when the MN offsets are shifted. For instance, when robots trot on a steep slope, the MN offsets are shifted close to the saturation zones of the MN transfer function [i.e., Eq. (7)] owing to the DFFB reflex. As a result, the joint command amplitudes will be decreased, and the robot's step length and height will be reduced. This strategy allows the robot to stabilize its posture during locomotion over slopes (see Figs. 17 and 24). This accords with biomechanical investigations into animal locomotion [26]. "

RC: Last sentence page 3 rework grammar

AR: We have corrected this as follows: "To obtain the robot posture balance, its legs must almost fully extend or flex. This leads to leg joint movements near their singular configuration or joint limits [27], particularly on steep slopes (e.g., 30°)."

RC: 65 - 2nd sentence grammar

AR: We have corrected this as follows: "Although the framework has the ability to realize adaptive offsets of CPG signals implicitly, it still requires several training sessions (up to 100 min)."

RC: 75 - why is '(data-driven) machine learning' specified?

AR: Here, "(data-driven) machine learning" refers to machine learning algorithms (e.g., data-driven reinforcement learning [28] and imitation learning [29]), which have been employed in locomotion control [28, 30, 31]; these use large datasets and trials to make models converge over a number of training hours or days.

However, to avoid confusion, we have now removed the term "data-driven". The sentence has been rewritten as follows:

"To overcome the limitations of the aforementioned control techniques (classical engineering, bio-inspired CPGs with reflexes, and machine learning), ..."

RC: 80 - last sentence grammar

AR: We have corrected this as follows: "This strategy is called "lever mechanics" in biomechanics [13]."

RC: 85 - "The employed learning mechanism endows the reflex with fast adaptation capability due to the induced synaptic plasticity of the reflex neural network." This reads to me as 'the learning network can adapt quickly', but is there other information to take from this sentence?

AR: This sentence in the manuscript is a little obscure. We have now rewritten the sentence as follows: " The DFFB reflex has fast adaptability owing to the learning mechanism (i.e., DIL). This is because the DIL can appropriately modulate the synaptic weights of the DFFB reflex network online with respect to sensory feedback."

RC: 90 - 'prevents' -> 'does not require'

AR: We have corrected this.

RC: 90 - 'To access the performance' - not clear what this means. Improve the performance?

AR: We have now modified this sentence as follows: "To evaluate the performance of the DFRL, it was integrated with a simple CPG-based control."

RC: 140 - "can be scaled to meet the joint movement range" - this sentence is hard to parse, did not understand before looking at Fig 3

AR: We have now simplified this sentence as follows: " The amplitudes of the MN outputs can be set via the synaptic weight projection from the CPGs to the MNs [α, see Fig. 3 (a)]. As a result, the CPG outputs are properly mapped to the joint angles, and the robot feet will exhibit alternating stance and swing motion states when the robot is balanced at the lift/touch moment [Fig. 3 (d)]"

RC: Similar grammar issues throughout, I stopped making note of them.

AR: We appreciate your kindness and patience for commenting upon the language issues. Furthermore, we apologize for the uncomfortable reading experiences produced by these language issues. We have now carefully checked the manuscript and employed a proofreading service from a professional institute, to improve the fluency of the text.

RC: The videos are great, it might be informative for the viewer if the $\bar{\gamma}$ value was displayed in a text overlay.

AR: We have now added the information for $\bar{\gamma}$ to the videos. The new videos have now been uploaded to the following links, which have also been included in the revised manuscript.

1. http://www.manoonpong.com/DFFB/video1.mp4
2. http://www.manoonpong.com/DFFB/video2.mp4
3. http://www.manoonpong.com/DFFB/video3.mp4
4. http://www.manoonpong.com/DFFB/video4.mp4

4. Reviewer #3

RC: I have read the manuscript. In this manuscript, experimental results on different quadruped robots have shown that the Distributed Force Feedback-Based Reflex with Online Learning (DFRL). It can automatically and

quickly adapt the Central Pattern Generators (CPG) patterns (motor commands) to the robots, enabling them to perform appropriate body posture during locomotion, and enable the robots to effectively accommodate various slope terrains including steep ones.

Authors have done quality work. So, this manuscript is suitable for publication in this journal.

AR: We greatly appreciate your careful review and your agreement to the publication of this manuscript.

References

[1] M. Kalakrishnan, J. Buchli, P. Pastor, M. Mistry, S. Schaal, Learning, planning, and control for quadruped locomotion over challenging terrain, The International Journal of Robotics Research 30 (2) (2011) 236–258.

[2] M. Hutter, C. Gehring, A. Lauber, F. Gunther, C. D. Bellicoso, V. Tsounis, P. Fankhauser, R. Diethelm, S. Bachmann, M. Blösch, et al., Anymal-toward legged robots for harsh environments, Advanced Robotics 31 (17) (2017) 918–931.

[3] J. Carlo, P. Wensing, B. Katz, G. Bledt, S. Kim, Dynamic locomotion in the mit cheetah 3 through convex model-predictive control, in: 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2018, pp. 1–9. doi:10.1109/IROS.2018.8594448.

[4] J. Nakanishi, J. Morimoto, G. Endo, G. Cheng, S. Schaal, M. Kawato, A framework for learning biped locomotion with dynamical movement primitives, in: 4th IEEE/RAS International Conference on Humanoid Robots, 2004., Vol. 2, IEEE, 2004, pp. 925–940.

[5] A. J. Ijspeert, J. Nakanishi, H. Hoffmann, P. Pastor, S. Schaal, Dynamical movement primitives: learning attractor models for motor behaviors, Neural Computation 25 (2) (2013) 328–373.

[6] C. Liu, L. Xia, C. Zhang, Q. Chen, Multi-layered cpg for adaptive walking of quadruped robots, Journal of Bionic Engineering 15 (2) (2018) 341–355.

[7] X. Zhang, H. Zheng, Walking up and down hill with a biologically-inspired postural reflex in a quadrupedal robot, Autonomous Robots 25 (1-2) (2008) 15–24.

[8] J. Sousa, V. Matos, C. P. dos Santos, A bio-inspired postural control for a quadruped robot: an attractor-based dynamics, in: 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE, 2010, pp. 5329–5334.

[9] M. Ajallooeian, S. Pouya, A. Sproewitz, A. J. Ijspeert, Central pattern generators augmented with virtual model control for quadruped rough terrain locomotion, in: 2013 IEEE international conference on robotics and automation, IEEE, 2013, pp. 3321–3328.

[10] Y. Fukuoka, H. Kimura, A. H. Cohen, Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts, The International Journal of Robotics Research 22 (3-4) (2003) 187–202.

[11] T. Sun, X. Xiong, Z. Dai, P. Manoonpong, Small-sized reconfigurable quadruped robot with multiple sensory feedback for studying adaptive and versatile behaviors, Frontiers in Neurorobotics 14 (2020) 14.

[12] X. Zhao, J. Zhang, C. Qi, Cpg and reflexes combined adaptive walking control for aibo, in: 2012 11th International Conference on Machine Learning and Applications, Vol. 1, IEEE, 2012, pp. 448–453.

[13] D. V. Lee, Effects of grade and mass distribution on the mechanics of trotting in dogs, Journal of Experimental Biology 214 (3) (2011) 402–411.

[14] D. T. Tran, I. M. Koo, Y. H. Lee, H. Moon, S. Park, J. C. Koo, H. R. Choi, Central pattern generator based reflexive control of quadruped walking robots using a recurrent neural network, Robotics and Autonomous Systems 62 (10) (2014) 1497–1516.

[15] J. R. Hughes, Post-tetanic potentiation, Physiological Reviews 38 (1) (1958) 91–113, pMID: 13505117. arXiv:https://doi.org/10.1152/physrev.1958.38.1.91, doi:10.1152/physrev.1958.38.1.91.
URL https://doi.org/10.1152/physrev.1958.38.1.91

[16] M. Thor, P. Manoonpong, Error-based learning mechanism for fast online adaptation in robot motor control, IEEE Transactions on Neural Networks and Learning Systems 31 (6) (2020) 2042–2051.

[17] F. Pasemann, M. Hild, K. Zahedi, So (2)-networks as neural oscillators, in: International Work-Conference on Artificial Neural Networks, Springer, 2003, pp. 144–151.

[18] S. Aoi, T. Yamashita, K. Tsuchiya, Hysteresis in the gait transition of a quadruped investigated using simple body mechanical and oscillator network models, Physical Review E 83 (6) (2011) 061909.

[19] T. Ichinomiya, Frequency synchronization in a random oscillator network, Phys. Rev. E 70 (2004) 026116. doi:10.1103/PhysRevE.70.026116.
URL https://link.aps.org/doi/10.1103/PhysRevE.70.026116

[20] G. Schöner, W. Jiang, J. Kelso, A synergetic theory of quadrupedal gaits and gait transitions, Journal of Theoretical Biology 142 (3) (1990) 359 – 391. doi:10.1016/S0022-5193(05)80558-2.
URL http://www.sciencedirect.com/science/article/pii/S0022519305805582

[21] T. Aoki, T. Aoyagi, Co-evolution of phases and connection strengths in a network of phase oscillators, Phys. Rev. Lett. 102 (2009) 034101. doi:10.1103/PhysRevLett.102.034101.
URL https://link.aps.org/doi/10.1103/PhysRevLett.102.034101

[22] I. Z. Kiss, Y. Zhai, J. L. Hudson, Emerging coherence in a population of chemical oscillators, Science 296 (5573) (2002) 1676–1678. arXiv:https://science.sciencemag.org/content/296/5573/1676.full.pdf, doi:10.1126/science.1070757.
URL https://science.sciencemag.org/content/296/5573/1676

[23] T. Sun, D. Shao, Z. Dai, P. Manoonpong, Adaptive neural control for self-organized locomotion and obstacle negotiation of quadruped robots, in: 2018 27th IEEE International Symposium on Robot and Human Interactive Communication (RO-MAN), IEEE, 2018, pp. 1081–1086.

[24] Y. Jia, X. Luo, B. Han, G. Liang, J. Zhao, Y. Zhao, Stability criterion for dynamic gaits of quadruped robot, Applied Sciences 8 (12) (2018) 2381.

[25] H. Hirukawa, S. Hattori, K. Harada, S. Kajita, K. Kaneko, F. Kanehiro, K. Fujiwara, M. Morisawa, A universal stability criterion of the foot contact of legged robots-adios zmp, in: Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006., IEEE, 2006, pp. 1976–1983.

[26] Z.-Y. Wang, A.-H. Ji, T. Endlein, W. Li, D. Samuel, Z.-D. Dai, Locomotor kinematics of the gecko (tokay gecko) upon challenge with various inclines, Chinese Science Bulletin 59 (33) (2014) 4568–4577.

[27] C. Gehring, C. D. Bellicoso, S. Coros, M. Bloesch, P. Fankhauser, M. Hutter, R. Siegwart, Dynamic trotting on slopes for quadrupedal robots, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2015, pp. 5129–5135.

[28] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, M. Hutter, Learning agile and dynamic motor skills for legged robots, Science Robotics 4 (26).

[29] X. B. Peng, E. Coumans, T. Zhang, T.-W. Lee, J. Tan, S. Levine, Learning agile robotic locomotion skills by imitating animals, arXiv preprint arXiv:2004.00784.

[30] W. Jones, T. Blum, K. Yoshida, Adaptive slope locomotion with deep reinforcement learning, in: 2020 IEEE/SICE International Symposium on System Integration (SII), IEEE, 2020, pp. 546–550.

[31] M. Thor, T. Kulvicius, P. Manoonpong, Generic neural locomotion control framework for legged robots, IEEE Transactions on Neural Networks and Learning Systems (2020) 1–13doi:10.1109/TNNLS.2020.3016523.