GRAB: GRAdient-Based Shape-Adaptive Locomotion Control

Sujet Phodapol[1], Thirawat Chuthong[1], Binggwong Leung[1], Arthicha Srisuchinnawong[1], Poramate Manoonpong[1,2] and Nat Dilokthanakul[1]

***Abstract*—Adaptive systems enable legged robots to cope with a wide range of environmental settings and unforeseen events. Existing reactive methods adapt either the walking frequency or the amplitude to only simple perturbations. This paper proposes an adaptive mechanism for central pattern generator (CPG)-based locomotion control that online-reacts to both internal and external soft constraints by adapting both the frequency and amplitude of driving signals. Our approach, namely GRAdient-Based shape adaptive control (GRAB), utilises real-time sensory signals for adapting the dynamics of the CPG. GRAB reacts to locomotion soft constraints given in a loss function. It can quickly adapt CPG's dynamics variables to reduce such a loss, with a gradient-descent-like update step. The update perturbs the shape of the driving signal, which implicitly changes both frequency and amplitude of the robot locomotion pattern. We test the GRAB mechanism on a hexapod robot and its simulation, where we demonstrate its several benefits over a state-of-the-art adaptive control baseline. First, we show that it can be used for reducing the tracking error by simultaneously changing the walking amplitude and frequency. Also, GRAB can be used for limiting the maximum torque/current, preventing motor damage from unexpected perturbations. Finally, we demonstrate how GRAB can be utilised to naturally adjust the robot's walking speed while taking into account multiple constraints, including target walking speed, external weight perturbations, and the robot's physical limit.**

***Index Terms*—Central Pattern Generators, Adaptive Control, Fail-safe Protection, Gradient Descent**

I. INTRODUCTION

THE challenge of legged-locomotion control comes with the large space of complex interaction dynamics between robots and their environments. Recent deep learning approaches tackle this problem using larger and more complex policies [1]. However, the robustness of these policies depends heavily on the quality of the training data. In contrast, bio-inspired locomotion control tackles this problem using simple control rules with *online adaptive mechanisms* to handle the changing dynamics of the test problem. These rules are hand-crafted and derived from biological knowledge and real-world observations, which are customised to the problem at hand. This makes them rather brittle to a new type of robots and constraints. In this work, we aim to create a more general framework for implementing a more general bio-inspired adaptive mechanism.

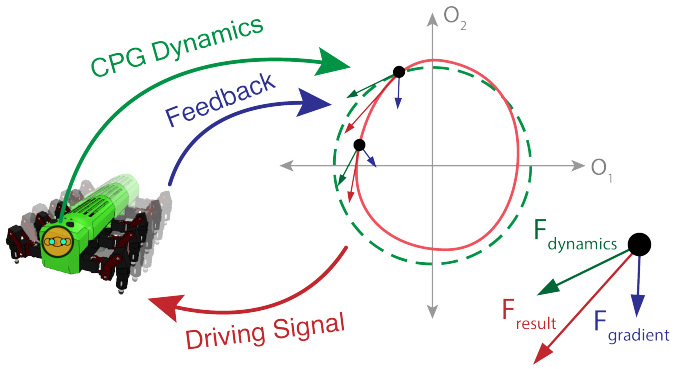


Fig. 1: GRAB controls the locomotion of the legged robot by perturbing the CPG dynamics in a direction that minimises a loss function. At each step, the state is pulled by the force from both the CPG dynamical surface (shown in a green arrow) and the force from the gradient of the loss (shown in a blue arrow), resulting in a resultant force (shown in a red arrow). As a result, the new trajectory complies with the pre-specified soft constraints as well as the original walking pattern.

There are also model-based [2], [3] and data-driven control approaches [4] that can flexibly and efficiently adapt robot's walking patterns to external perturbations [3], [5]. However, these approaches also need very complex controllers, which are either computationally expensive or very difficult to construct as they require a large number of high-quality data samples or faithful models of the robots and their environments. Here, we develop our adaptive controller from the bio-inspired perspective, where the focus is on the simplicity of the mechanisms allowing it to be easily implemented on the robots with varying computing power and morphology.

Notable bio-inspired methods such as the central pattern generator (CPG; [6], [7]) and the dynamic movement primitive (DMP; [8]) can be used to create complex walking patterns with relatively simple control mechanisms. The most successful studies use rule-based adaptation methods that are tailored to the type of perturbation, and the morphology of the robots [9], [10], [11]. While, the more generic adaptive rules, e.g.

Manuscript received: August 31, 2021; Revised November 10, 2021; Accepted December 4, 2021.

This paper was recommended for publication by Editor Clement Gosselin upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the PTT Exploration and Production Public Co. Ltd. under the development of Advanced Robot Motion Control for Freelander Inspection Robots (FREELANDER) project [Grant no. 3450024742] and the startup grant on Bio-inspired Robotics (P.M., the project PI) of Vidyasirimedhi Institute of Science and Technology.

[1]S.P., T.C., B.L., A.S., P.M. and N.D. is with Bio-inspired Robotics and Neural Engineering Laboratory, School of Information Science and Technology, Vidyasirimedhi Institute of Science and Technology (VISTEC), Rayong, Thailand. (Corresponding author: Nat Dilokthanakul. natd_pro@vistec.ac.th)

[2]P.M. is also with the Embodied AI and Neurorobotics Lab, SDU Biorobotics, and the Mærsk Mc-Kinney Møller Institute, University of Southern Denmark, Odense, Denmark. poramate.m@vistec.ac.th

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

reactive DMP [12], [13], can only be used to stop or slow down the control execution without much consideration to the desired behaviours and external dynamics. There are also several works that add customised reactive feedback terms into the DMP [12], [13]. However, these reactive DMP are tailored to be specific to the robot and the task at hand.

Our work extends the line of research in CPG-based controllers [6], [7], which is not online-adaptive in its original form. Other extensions to the CPG-based control that allows it to be online-adaptive can be categorised into two groups: (i) frequency adaption [14], [9], [15], [16], [17] and (ii) amplitude adaptation [18], [19]. Frequency adaptation is useful for adapting the speed of the locomotion and for energy efficiency. Amplitude adaptation, on the other hand, can be used as a safety mechanism as well as to improve traversal on rough terrain.

Our recent work, Dynamical State Forcing CPG (DSF-CPG) [20], is the first CPG-based reactive controller that can simultaneously adapt both frequency and amplitude in an online fashion. Although, there are other works that change both frequency and amplitude with their *learning* ability[1]. These learning-based methods are not reactive and need a longer time-scale to perform the learning-based adaptation [21].

This ability to reactively and simultaneously adapt both frequency and amplitude is done by directly modifying the control signal at the level of the CPG dynamics. DSF-CPG periodically resets the dynamical state of the CPG to the position of a weighted average between the command signal and the actual robot reading signal in order to reduce the tracking error. The key weakness of DSF-CPG is that it requires an inverse mapping between the dynamical space and the robot space. Also, it can only be used to reduce tracking error, whereas we would like it to be more widely applicable.

To remove these limitations of DSF-CPG, we propose a novel adaptive mechanism, namely GRAdient-Based shape adaptive control (GRAB), which is intended to be a generic adaptation rule that can flexibly online adapt the shape of the driving signal towards desired behaviours (see Fig. 1). Inspired by the deep learning framework, we take a top-down approach by putting an abstract description of the desired behaviour at the core of the adaptive mechanism via a given loss function. The adaptation works by continuously perturbing the CPG driving signal in a direction that reduces the loss. As a mechanism, we propose to use the gradient of the loss as a perturbation force that can be combined with the force in the attractor dynamics of the CPG. This combination draws out a new trajectory, which creates a driving signal that complies with both the walking pattern (as created by the CPG attractor) and the desired behaviour (as specified via the loss function).

The contributions of this work can be summarised as follow:

- We propose a novel online-adaptive mechanism that can spontaneously adapt the *shape* of the driving signal via the combination of the gradient perturbation and the CPG attractor dynamics. While this work could be seen as an extension to our previous work (DSF-CPG), GRAB has an entirely different implementation than DSF-CPG, and thus offers a considerably broader range of applications.
- We show that a simple loss function of tracking error reduction can be used to create adaptive walking behaviours that comply with external perturbations to the robot.
- We demonstrate that the GRAB mechanism can be used to limit joint torque, which can help prevent motor failure in a situation where heavy force is applied on the robot.
- We demonstrate that the GRAB mechanism can handle an additional speed soft constraint. This shows GRAB's ability to comply with both internal and external constraints[2], which is useful for controlling the robot.

II. METHOD

A. Central Pattern Generator

Central Pattern Generator (CPG) is a mathematical model of coupled neural oscillators, which can generate a periodic signal and can be used as a target pattern for locomotion control [6]. In this work, we use a type of CPG called the SO(2)-network [22]. The SO(2) network can be described as two coupled neurons. The activities of both neurons (a_1, a_2) and their outputs (o_1, o_2) are described by the following update equations:

$$\begin{bmatrix} o_1 \\ o_2 \end{bmatrix} \leftarrow \tanh \begin{bmatrix} a_1 \\ a_2 \end{bmatrix} = \tanh \begin{bmatrix} w_{11}o_1 + w_{12}o_2 \\ w_{21}o_1 + w_{22}o_2 \end{bmatrix} \quad (1)$$

where the weight matrix for the SO(2)-network depends on two variables ϕ and α:

$$W = \begin{bmatrix} w_{11} & w_{12} \\ w_{21} & w_{22} \end{bmatrix} = \alpha \cdot \begin{bmatrix} cos(\phi) & sin(\phi) \\ -sin(\phi) & cos(\phi) \end{bmatrix} \quad (2)$$

α is set to a value larger than 1 to create a limit cycle dynamics. ϕ is a parameter that can determine the frequency of the oscillation. To generate a desired trajectory, we take the oscillating outputs o_1, o_2 and transform them through a post-processing function into a locomotion pattern. In this work, we use simple linear functions ($y_i = mo_i + c$) as our post-processing functions, which are hand-tuned to create a suitable locomotion pattern. We then use this pattern as a target driving signal for a low-level position controller (e.g., a PID controller).

B. GRAdient-Based Shape-Adaptive Control (GRAB)

In order to incorporate adaptive behaviour into a CPG, one could implement a higher-level mechanism that uses sensory feedback to either modulate the oscillation frequency via the parameter ϕ [9], [15], [16], [17] or the driving amplitude via the post-processing function [18], [19]. Alternately, we propose a novel adaptive CPG, namely *GRAB*, which incorporate the adaptive mechanism directly into the dynamics of the activity of the CPG's neurons.

[1]The term "adaptation" is sometimes used to refer to long-term changes in the dynamics (i.e., learning). In this paper, we use the term adaptation to refer to changes to the dynamics both with and without learning.

[2]We use the term constraint to refer to a soft constraint, as opposed to a strict condition, throughout the paper.

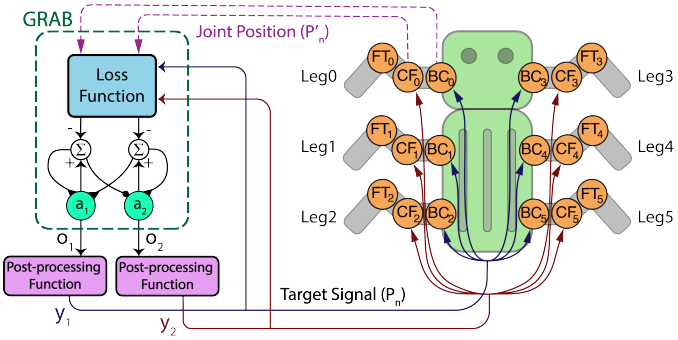


Fig. 2: CPG with GRAB adaptive mechanism can be represented as a closed-loop system between the neurons and the robot system. As shown in the blue and red lines, the target signals from o_1 and o_2 are processed using the post-processing function and used as a driving signal for the BC and CF joints. The joint sensory feedback will be monitored from joint BC_0 and CF_0, as shown in the dashed purple line. The joint sensory feedback is then used to calculate the loss function in order to change the neural dynamics of the CPG.

GRAB's ability to perturb the dynamics directly allows it to be reactive and, therefore, adapt quickly to any instantaneous perturbation. It can also modulate the shape of the driving signal (as illustrated in Fig. 1). The key component of GRAB is the use of a given loss function and its gradient with respect to the dynamic variables. The loss function can be understood as an abstracted description of *how* we would like the pattern to be adaptive. Inspired by the stochastic gradient descent method [23], the gradient of the loss is used to move the oscillating dynamic closer to the desired behaviour. Equation (3) describes GRAB as an additional update equation of the SO(2) dynamics.

$$\begin{bmatrix} o_1 \\ o_2 \end{bmatrix} \leftarrow \begin{bmatrix} o_1 \\ o_2 \end{bmatrix} - \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \odot \begin{bmatrix} \frac{\partial L}{\partial o_1} \\ \frac{\partial L}{\partial o_2} \end{bmatrix} \tag{3}$$

where γ_1, γ_2 are hyper-parameters of the updating gains (or step-sizes). $\odot$ is an element-wise product operator. L is a pre-defined loss function. For example, L can be defined with a constraint such as a measure of mismatch between the desired behaviour and the current state of the robot. The combined update equations can also be represented with a discrete-time dynamics equation as:

$$\begin{bmatrix} o_1' \\ o_2' \end{bmatrix} = \tanh \begin{bmatrix} w_{11}o_1 + w_{12}o_2 \\ w_{21}o_1 + w_{22}o_2 \end{bmatrix} - \begin{bmatrix} \gamma_1 \\ \gamma_2 \end{bmatrix} \odot \begin{bmatrix} \frac{\partial L}{\partial o_1} \\ \frac{\partial L}{\partial o_2} \end{bmatrix} \tag{4}$$

where o' is one time-step ahead of o.

The gradients of the loss force the outputs of the neurons to the direction that reduces such a mismatch. Fig. 1 illustrates the adaptive mechanism of GRAB. The state of a particle in the phase space is driven by two forces: the force due to the dynamical surface of the SO(2) and the additional force from the gradient descent to the direction that minimises the loss function, resulting in a new trajectory.

As a side note, the gradient-based perturbation mechanism used in GRAB has a striking similarity to the Tagotae method introduced by Owaki et al. [24]. The Tagotae method is used for adapting the phases of the leg controllers, which are needed to be coordinated with each other in the multi-leg decentralised locomotion control. Despite the similarity in the mechanism, GRAB aims to be a generic adaptive mechanism that can react to both external and internal perturbations. Therefore, the shape-adaptive ability is an important property of GRAB; this has not been explored in the Tagotae method, which only considers phase adaptation in a system of phase oscillators.

C. Tracking Error Reduction

The loss function is central to our method as it determines the desired adaptive behaviour of the locomotion trajectory. First, we explore a simple loss that attempts to reduce the tracking error, which is an important source of undesired motion in a robot's locomotion [9]. We define the tracking error as the square difference between the driving target position and the actual robot's joint position:

$$L = \sum_{i=1}^{n} \frac{1}{2}(P_n - P_n')^2 \tag{5}$$

where P_n is the target position and P'_n is the actual robot position. n represents the n^{th} driving joint of the robot.

D. Speed Modulation

It is also possible to add multiple constraints to the trajectory dynamics. In our experiments, we investigate whether we can modulate the speed of the locomotion through GRAB mechanism. This is particularly useful when we want to perform semi-automatic teleoperation, while still enabling the robot to adapt to the external and internal perturbations. We cannot naively set ϕ to adjust the walking frequency because each robot can only support a range of ϕ due to the hardware limitation [25]; this could lead to an undesired walking behaviour if the hardware cannot keep up to the control target [9].

We define the speed modulation loss as the mismatch between our desired speed and the robot driving speed. In addition, we regularise the dynamics with the tracking-error reduction term. Intuitively, with a speed-up command, the tracking-error reduction would implicitly decrease the walking frequency to reduce the error, while the speed modulation loss increases the frequency to speed up the robot. Therefore, the robot's walking frequency would reach a certain value that cannot be increased any further due to the tug-of-war between the two. The loss function and the dynamic update are defined as:

$$L = \frac{1}{2}(V_{target} - V_{driving})^2 + \sum_{i=1}^{n} \frac{1}{2}(P_n - P_n')^2 \tag{6}$$

$$\begin{bmatrix} o_1 \\ o_2 \\ \phi \end{bmatrix} \leftarrow \begin{bmatrix} o_1 \\ o_2 \\ \phi \end{bmatrix} - \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix} \odot \begin{bmatrix} \frac{\partial L}{\partial o_1} \\ \frac{\partial L}{\partial o_2} \\ \frac{\partial L}{\partial \phi} \end{bmatrix} \tag{7}$$

where V_{target} is a velocity target value set by the robot's operator; this can be seen as an internal speed-constraint of the robot. This desired speed can be realised by increasing the parameter ϕ, which controls CPG's default oscillating frequency. Thus, we estimate the robot's driving speed with

$V_{driving} = k_v\phi$. It is important to note that the robot's walking frequency depends on both driving force ϕ and the gradient from tracking error loss.

E. Bounding the CPG Dynamics

An important property of SO(2) is the bounded periodic dynamics that is generated from its stable limit cycle property. By perturbing the signal with the gradient term, we run into the risk of driving the dynamics into an unstable region resulting in a divergence behaviour.

To reconcile this problem, we must bound the perturbation to a certain value. We hypothesise that if the force acting on the particle is dominated, to a certain degree, by the dynamic force, $|F_{dynamics}| > |F_{gradient}|$, then the trajectory stays bounded. One possible method to reduce the size of $|F_{gradient}|$ is to use the method of gradient clipping, where $|\frac{\partial L}{\partial \phi}|$ or $|\frac{\partial L}{\partial o}|$ are capped below a specific maximum value, Ω, to avoid the gradient term becoming too large. Empirically, in our experiments, the dynamics are always bounded even without the capping. However, we find that it is useful to cap the gradient $|\frac{\partial L}{\partial \phi}|$ in the speed modulation mechanism because it simplifies the parameter-tuning process. We discuss about the value selection of this parameter in Sec. III.D.

In a typical CPG network, ϕ is a fixed parameter and, therefore, does not have a default dynamics (i.e., $|F_{dynamics}| = 0$). In our speed adaptation mechanism, this fact leads to the domination of the gradient term $F_{gradient}$, which create a divergence behaviour. To this end, we define the default dynamics for ϕ to be a point attractor dynamics pointing to a default value ϕ_0,

$$\phi \leftarrow \phi - k_\phi(\phi - \phi_0). \tag{8}$$

As one can see, GRAB can be used to adapt a CPG parameter by turning it into a bounded dynamic variable. As a side note, it is advisable to avoid adapting α because α has a non-trivial effect on the shape of the SO(2); there is a range of α that creates unstable SO(2) dynamics.

III. Simulations and Experiments

We investigate three questions in our experiments:

- Q1: How does GRAB adapt the walking signal of a robot when a load is added to the robot?
- Q2: How good is GRAB at reacting to an external force that can potentially damage the robot's motor?
- Q3: Can we generalise GRAB mechanism to be reactive to other internal constraints, such as the speed of the robot?

By studying these questions, we can understand more about the situations where we would prefer GRAB adaptive mechanism over other mechanisms, such as the reactive DMP [12], [13]. Also, we want to understand potential applications that GRAB could be used for in future work. [3]

We evaluate our method using the MORF robot (Fig. 3, [26]). MORF is a hexapod with six legs; following a morphology of an insect, each leg of the robot consists of three actuated joints: body-coxa (BC), coxa-femur (CF) and femur-tibia (FT) joints (see Fig. 2). The driving signals from the SO(2) neurons are transformed through linear (post-processing) functions, F_1 and F_2, to create suitable joint angles, y_1 and y_2. These signals are used to drive BC and CF joints, respectively. BC performs a horizontal leg swing motion, while CF performs a lifting motion. The FT joint is fixed at a specific position. There is a phase-shift of $\frac{\pi}{2}$ radian between y_1 and y_2 to create an appropriate intra-leg coordination. [4]

(a) MORF robot in real world (b) MORF robot in simulation

Fig. 3: MORF: Hexapod Robot Platform developed by Thor *et al* [26]. The robot has a weight of approximately 4 kg.

To investigate the adaptive mechanism, we compare GRAB with a periodic dynamic movement primitive (pDMP) [27], which is a state-of-the-art method for adaptive control, and the non-adaptive SO(2) [22] as our baselines.

pDMP[27] can be described as follows:

$$\dot{z} = \Omega(\alpha(\beta(-y) - z) + f(\phi)) \tag{9}$$

$$\dot{y} = \Omega z(1 + \alpha_r(L))^{-1} \tag{10}$$

$$\dot{\phi} = \Omega \tag{11}$$

where z is an auxiliary variable, y is the joint angle, ϕ and Ω are the phase and frequency, respectively. $\alpha = 8$ and $\beta = 2$ are the positive gain, f is the forcing function, which create the desired trajectory (see [27] for more details). For easy comparison, DMP error function, L, is defined as (5) with a suitable coupling gain for this error function $\alpha_r = 1000$. For the implementation on the robot, GRAB (green box) is replaced with pDMP, in Fig. 2, where the connections between target signals and the robot remain the same.

To achieve similar default robot locomotion patterns, before the perturbation, we manually tune the hyper-parameter of each method. First, we set $\phi = \frac{\pi}{12}, \alpha = 1.4, \gamma_1 = \gamma_2 = 2.0$ for GRAB, which are chosen such that it create a stable walking pattern. Then, we use the joint signal from GRAB as a reference trajectory for pDMP. The hyper-parameters of SO(2) are chosen to be $\phi = 0.137, \alpha = 1.23$. These are chosen by manually adjusting the amplitude and the frequency to be as close to GRAB's trajectory as possible.

A. Experiment 1: adaptation to an external load

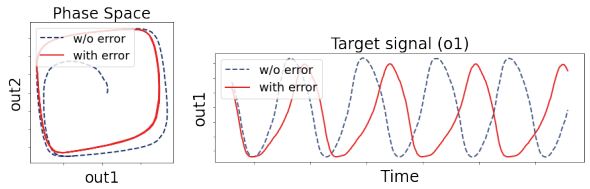
The first experiment is set out to investigate GRAB's ability to online manipulate the driving signal under an external perturbation (Q1). To this end, we investigate MORF's walking behaviours under three situations: (i) MORF walks freely (without load), (ii) a load of 8 kg is added on the back of MORF and (iii) the load is lifted off (see Fig. 5).

[3]The experiment video can be viewed at www.manoonpong.com/GRAB/SupplementaryVideo.mp4

[4]The GRAB code is available at www.gitlab.com/BRAIN_Lab/public/grab

Method	Symbols	Values	Description
GRAB	ϕ	$\pi/12$	Frequency parameter
	α	1.4	CPG shape parameter
	γ_1	2.0	Gradient gain
	γ_2	2.0	Gradient gain
	γ_3	0.05	Gradient gain
	k_v	5	Frequency-to-speed gain
	k_ϕ	0.01	Point attractor gain
	ϕ_0	0	Fixed point position
	Ω	0.1	Gradient cap
pDMP	α	8	Positive gain
	β	2	Positive gain
	Ω	1	Time constant
	α_r	1000	Coupling gain
SO(2)	ϕ	0.137	Frequency parameter
	α	1.23	CPG shape parameter

TABLE I: Parameters used in the experiments. These parameters are set such that the default walking behaviours of the three methods have the same amplitude and frequency.



(a) Phase-space (b) GRAB with and without perturbation

Fig. 4: Comparing the GRAB output signals with and without weight perturbations. (a) The phase space shows that the GRAB trajectory (dotted blue line) starts from an initial position and converges to a limit cycle behaviour. During the perturbation (solid red line), the shape of the limit cycle has been distorted from the gradient term. (b) The time plot shows that the perturbed trajectory also has a lower frequency with wider troughs than the crests and a slightly smaller amplitude than the unperturbed trajectory.

Fig. 4 shows the phase-space plot and the driving signal comparing GRAB with the weight perturbation and without the perturbation. We can see in Fig. 5, 6 that, when the load is added to the robot, GRAB immediately adapts the shape, amplitude and frequency of the driving signal. The frequency becomes lower, and the amplitude becomes slightly smaller. Interestingly, we can see that the periodic driving signal becomes asymmetric, with the crest being narrower than the bottom trough. This is because, around the crest, the robot is lifting the leg up and, therefore, there is a smaller external force acting on the joints. Fig. 7 shows that GRAB uses the smallest maximum torque compared to pDMP and SO(2) controllers.

B. Experiment 2: torque limitation

In this experiment, we investigate GRAB's ability to prevent motor failure from external forces (Q2). We divide the experiment into two parts.

a) Leg blocking: In the first part, we investigate the adaptation to a horizontal force. This is done by attaching MORF robot to a holder, allowing the legs to move in the air freely. Then, we block the leg with our hands. Fig. 9 shows that GRAB can quickly stop the target signal within 1s to prevent damage. From Fig. 8, we can see that pDMP adapt the target signal by only reducing the frequency of the signal. SO(2) does not change the target signal. As a consequence,

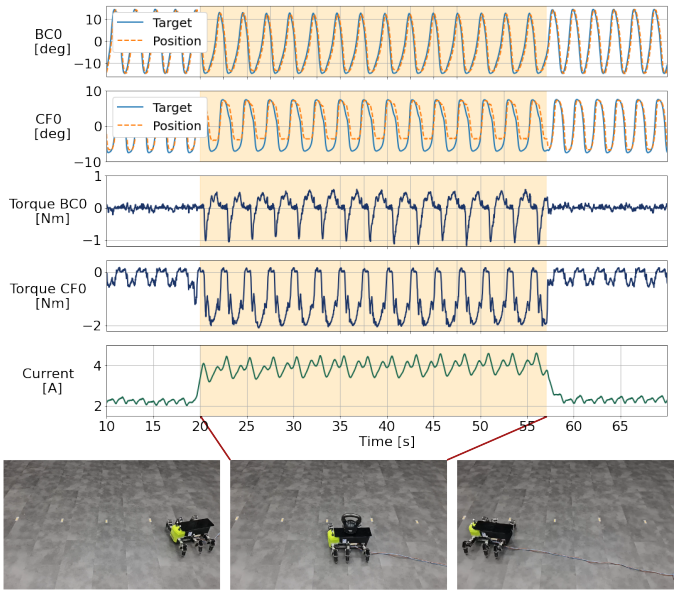


Fig. 5: This figure shows the setting of Experiment 1. First, the robot walks normally for 20 seconds. Then, an 8 kg load is added to the robot. The graphs show that the robot has to exert more force and uses more current to keep it walking. GRAB can quickly adapt the robot locomotion pattern to deal with the added load and to maintain its locomotion behaviour.

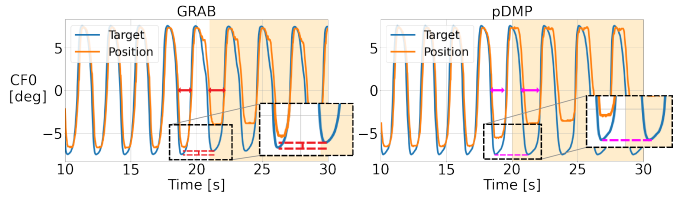


Fig. 6: Comparison between GRAB and pDMP in Experiment 1 shows that GRAB manipulates both the frequency and amplitude of the signal, while pDMP cannot change the amplitude of the signal. See also Fig. 4b, where the target signals, before and after the perturbation, are overlaid.

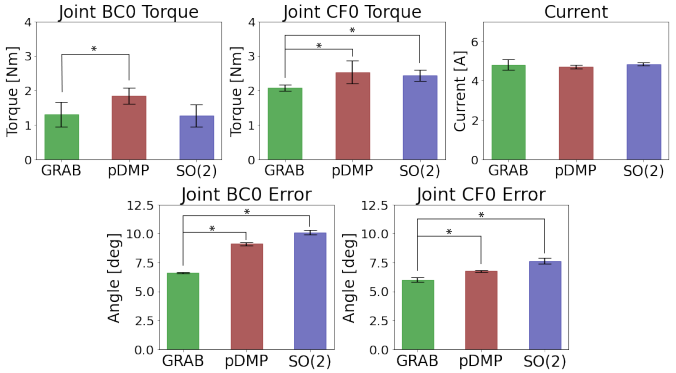


Fig. 7: GRAB is capable of reducing the tracking error of the joint position, leading to the smallest value of maximum torque in joint CF_0, which is the main joint that carries the weight. The error bar is the standard deviation from the results of five trials. (* denotes a significant difference with p-value < 0.05, Kruskal-Wallis's ANOVA test, Mann-Whitney-Wilcoxon's post-hoc test)

the maximum current from pDMP and SO(2) controllers are higher than GRAB by 37.5% and 54.4%, respectively.

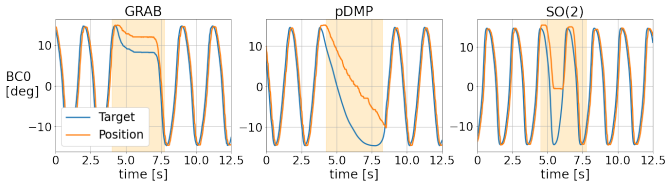


Fig. 8: GRAB, pDMP and SO(2) in Experiment 2a (leg blocking). GRAB can quickly stop the target signal when the leg is blocked, whereas pDMP can only prolong it, and the SO(2) signal remains the same.

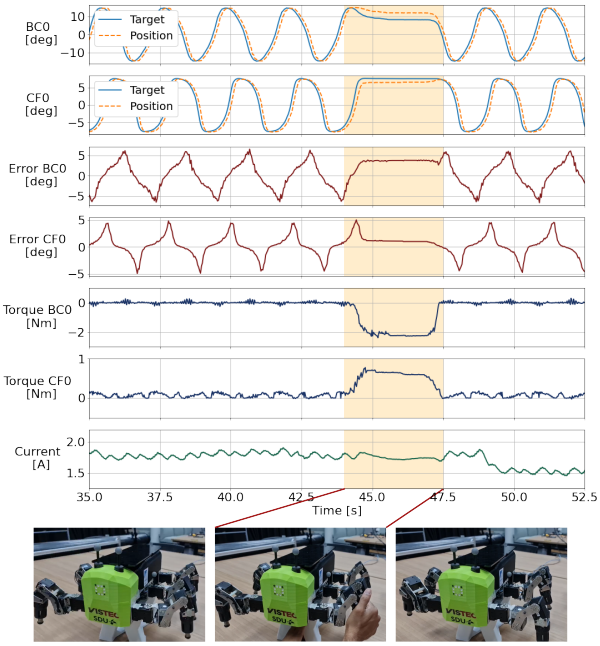


Fig. 9: GRAB is adapted to leg blocking, resulting in the driving signal comes to a halt, as shown in the solid blue line. GRAB also immediately reduces tracking error to small values.

b) Heavy Weight: In the second part, similar to Experiment 1, we investigate the adaptation to weight during a walking motion. However, this time, we put in a 16 kg weight (approximately four times robot weight), which is heavy enough to prevent the robot from moving ahead. First, we allow it to walk for 50 cm before adding the weight. Then, after 10 seconds of carrying the weight, we lift the weight up. According to the result shown in Fig. 10, the robot drives with GRAB can stop the driving signal and limit the maximum torque output.

The result in Fig. 11 shows that our method provides the least maximum torque and current, allowing the motors to continue to move after removing the additional weight. On the contrary, with pDMP and vanilla SO(2) methods, some motors are shut down, leading them to stop moving (see the video in the supplementary). This is because Dynamixel motors have a built-in mechanism that shut the motor down when it is overloaded.

C. Experiment 3 (simulation): speed modulation

In this experiment, we investigate GRAB's ability to adapt to an additional internal velocity constraint (Q3). GRAB's mechanism uses in this experiment is outlined in Sec. II-D.

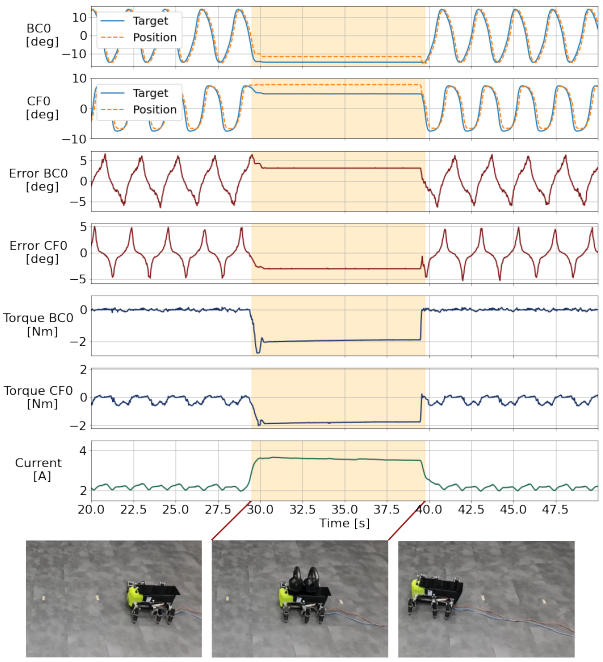


Fig. 10: GRAB is used to adapt the driving target to additional weight. The robot is walking normally at first; then, a heavy load is applied to stop the robot from moving. As a result, the robot quickly adapts to a halt state, which prevents the motors from shutting down and getting damage.

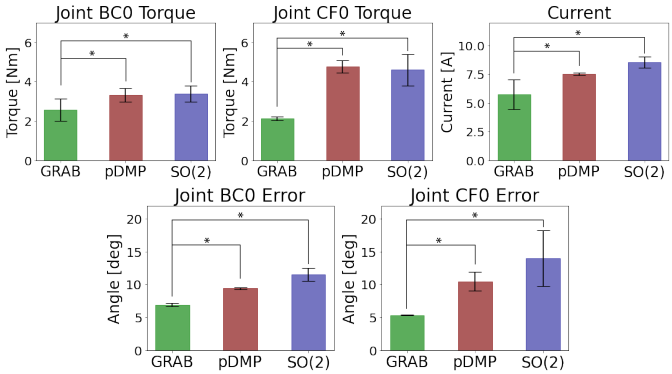


Fig. 11: The results from Experiment 2b demonstrate that GRAB is capable of preventing the motors from breaking down by reducing the tracking error to small values, resulting in the lowest maximum torque and current among the baselines. (* denotes a significant difference with p-value < 0.05, Kruskal-Wallis's ANOVA test, Mann-Whitney-Wilcoxon's post-hoc test)

Here, we implement it in a simulated MORF. We use the following parameters: $\gamma_3 = 0.05, k_v = 5$, $k_\phi = 0.01$, $\phi_0 = 0$ and $\Omega = 0.1$. As the robot walks, we vary V_{target} and put a small weight and a large weight on the robot to investigate its walking behaviour.

We can see in Fig. 12 that the speed of the robot, as can be observed from the frequency of the signal, varies according to the target velocity constraint, V_{target}. As we set V_{target} to be higher, GRAB gradually increases the parameter ϕ, which is bounded by the point attractor force.

We can see that there is a drop in the walking frequency when the robot carries a small weight. It completely stops when it carries a large weight. This shows that GRAB can naturally adapt to the robot's internal and physical constraints.

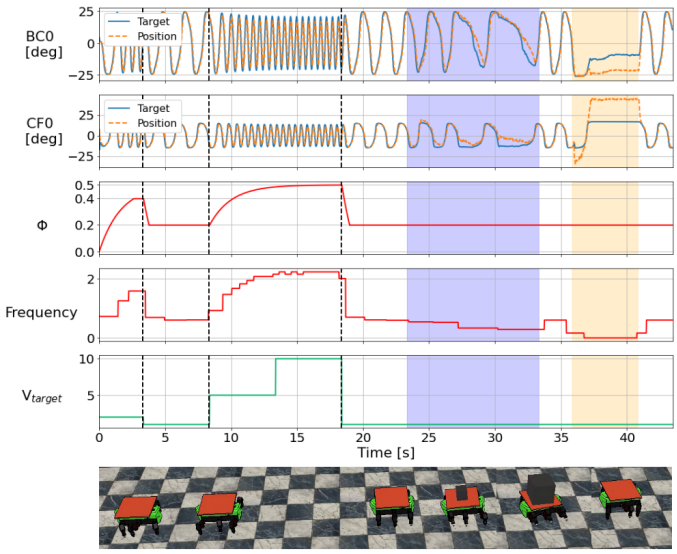
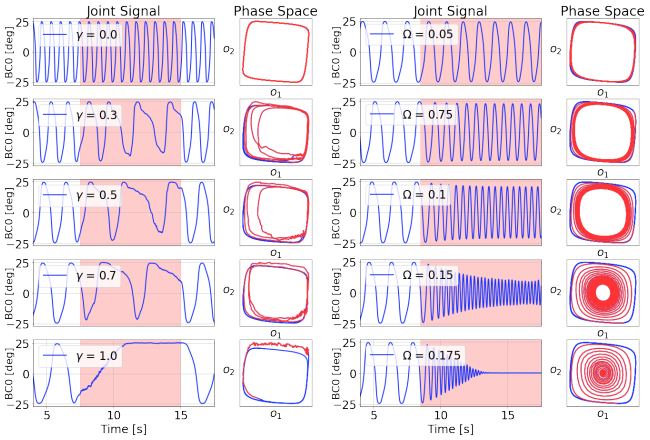


Fig. 12: GRAB with internal and external constraints. By incorporating V_{target} as an internal constraint, GRAB can be used to modulate the parameter ϕ such that the walking frequency, as well as the driving signal, comply with both internal constraints (robot's physical limit and V_{target}) and external constraint (weight perturbation). During $t = 24$s to $t = 34$s, a light load is put on the robot. A heavy load is placed on the robot during $t = 36$s to $t = 41$s. The frequency is estimated from the BC_0 position signal.



(a) Effect of γ after an external load is added.

(b) Effect of Ω during an increase in V_{target}.

Fig. 13: The adaptation behaviours during the increase in external loss and internal loss. γ adjusts GRAB sensitivity to tracking-error loss due to external perturbations. Ω adjusts GRAB sensitivity to tracking-error loss due to the internal speed constraint.

Lastly, GRAB can also simultaneously adapt to an external weight perturbation and revert back to its previous dynamical state.

D. Experiment 4 (simulation): GRAB parameters

In this section, we discuss the sensitivity and applicability of parameters required by the GRAB mechanism. Firstly, ϕ and α are inherited from the SO(2) mechanism. The value selection of these parameters is the same as SO(2), which is to choose ϕ that matches the robot's locomotion speed requirement, i.e., by testing with a low frequency ϕ and increases the value until the observed locomotion speed and tracking error are at suitable levels, and to select α to be slightly more than one to avoid the unstable region.

Next, the gradient gain, γ, specifies the sensitivity of GRAB to the error term. Fig. 13a shows a result of a simple experiment of dropping a 20 kg box on a simulated robot with different values of $\gamma = \gamma_1 = \gamma_2$. During the adaptation, we can see that, with small γ, the robot would only slightly adapt to the load. However, the robot quickly comes to a halt if the γ_1, γ_2 are set to as high as 1.0.

In the speed modulation mechanism, k_v and k_ϕ adjust the sensitivity to the internal speed constraint (V_{target}). Since V_{target} is uncalibrated and can be arbitrarily chosen, we do not need to properly tune these values. However, we need to tune either $\Omega = |\frac{\partial L}{\partial \phi}|$ or γ_3 which has the effect of adjusting the balance between the internal and external constraints. We found that Ω is easier to tune because it is not directly coupled to γ_1 and γ_2. A high value of Ω means that GRAB is more sensitive to external constraint, which results in the robot trying to stop/slow down the locomotion to reduce the tracking-error loss. While, a low value of Ω means that GRAB is more sensitive to the internal requirement and choose to speed up ϕ to match V_{target} while allowing the external loss to be increased. Fig. 13b shows how BC0 joint target adjusts to the abrupt changes in the internal constraint from $V_{target} = 1$ to $V_{target} = 5$. We can then select the value of Ω that keeps the robot walking forward, while still reasonably reacting to external perturbations. This shows that the suitable range of γ and Ω depend on the potential size of the perturbations. A future work could incorporate a meta mechanism that adjusts these variables as the robot observes its environment.

IV. Discussion and Conclusion

In this paper, we propose a novel adaptation mechanism, GRAB, that works by continuously perturbing the CPG dynamics in the direction that reduces a pre-specified loss function. The key benefit of GRAB is the ability to handle generic pre-specified soft constraints in the loss function, which allows it to be used in a versatile manner with fast adaptation.

In the experiments, GRAB is able to manipulate the shape, amplitude and frequency of the CPG driving signal. The results show that, by adapting the signal, GRAB can lower maximum torque, tracking error and current used by the robot under external perturbations. This point to several potential applications such as a motor protection mechanism and a fail-safe mechanism.

The final experiment highlights the potential impact of GRAB. By describing speed as an internal constraint, GRAB can modulate the walking pattern towards the constraint while simultaneously handling external perturbations. The speed modulation mechanism can be used for the semi-automatic teleoperation of the robot. By describing desired speed in the loss function, we can avoid a direct setting of the walking frequency, which can go wrong if the driving frequency becomes too high. GRAB naturally adjust for an appropriate frequency by considering both the internal dynamic and the external perturbation. In future work, other constraints, such as the

balancing constraint, the energy constraint, the robot's height, or the robot's stride width, could potentially be expressed in the loss function as well.

The key weakness of GRAB lies in the locality of the loss gradient. In other words, the gradient only reacts to the instantaneous sensory perceptions. Without the ability to plan forwards into the future nor predict the long-term result of its action, GRAB is still limited to only a few types of perturbations. An interesting direction to explore is how to combine the foresight ability of the high-level model-based method with the low-level adaptive ability of GRAB such that the robot could handle a wider range of environments and perturbations. Furthermore, applying GRAB to an unstable system, like a biped robot, or to robot navigation in cluttered environments, the loss function and constraints need to be appropriately specified. For example, in the future work, one could incorporate a balancing constraint (e.g., stability index [28]) into the loss function to keep the robot balance and, potentially, a speed constraint that keeps the robot moving, while adapting its head direction to avoid obstacles by using exteroceptive sensory feedback with adaptive neural sensory processing [29].

ACKNOWLEDGMENT

This work was supported by the PTT Exploration and Production Public Co. Ltd. under the development of Advanced Robot Motion Control for Freelander Inspection Robots (FREELANDER) project [Grant no. 3450024742] and the startup grant on Bio-inspired Robotics (P.M., the project PI) of Vidyasirimedhi Institute of Science and Technology.

The authors would like to thank Potiwat Ngamkajornwiwat, Jettanan Homchanthanakul and Kongkiat Rothomphiwat for their useful comments. Mathias Thor for his support with the MORF robot and its simulation. Lastly, we would like to thank the anonymous reviewers and editor of RA-L for their thoughtful comments.

REFERENCES

[1] C. Yang, K. Yuan, Q. Zhu, W. Yu, and Z. Li, "Multi-expert learning of adaptive legged locomotion," *Science Robotics*, vol. 5, no. 49, 2020. [Online]. Available: https://robotics.sciencemag.org/content/5/49/eabb2174

[2] E. Camacho, C. Bordons, and C. Alba, *Model Predictive Control*, ser. Advanced Textbooks in Control and Signal Processing. Springer London, 2004. [Online]. Available: https://books.google.co.th/books?id=Sc1H3f3E8CQC

[3] J. Di Carlo, P. M. Wensing, B. Katz, G. Bledt, and 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)*. IEEE, 2018, pp. 1–9.

[4] R. S. Sutton and A. G. Barto, *Reinforcement learning: An introduction*. MIT press, 2018.

[5] D. Kim, J. Di Carlo, B. Katz, G. Bledt, and S. Kim, "Highly dynamic quadruped locomotion via whole-body impulse control and model predictive control," *arXiv preprint arXiv:1909.06586*, 2019.

[6] 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.

[7] 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.

[8] S. Schaal, "Dynamic movement primitives-a framework for motor control in humans and humanoid robotics," in *Adaptive motion of animals and machines*. Springer, 2006, pp. 261–280.

[9] M. Thor and P. Manoonpong, "Error-based learning mechanism for fast online adaptation in robot motor control," *IEEE transactions on neural networks and learning systems*, vol. 31, no. 6, pp. 2042–2051, 2019.

[10] C. Akkawutvanich, F. I. Knudsen, A. F. Riis, J. C. Larsen, and P. Manoonpong, "Adaptive parallel reflex- and decoupled CPG-based control for complex bipedal locomotion," *Robotics and Autonomous Systems*, p. 103663, 2020.

[11] T. Sun, Z. Dai, and P. Manoonpong, "Distributed-force-feedback-based reflex with online learning for adaptive quadruped motor control," *Neural Networks*, 2021.

[12] H. Tan, E. Erdemir, K. Kawamura, and Q. Du, "A potential field method-based extension of the dynamic movement primitive algorithm for imitation learning with obstacle avoidance," in *2011 IEEE International Conference on Mechatronics and Automation*. IEEE, 2011, pp. 525–530.

[13] T. Petrič, A. Gams, L. Žlajpah, A. Ude, and J. Morimoto, "Online approach for altering robot behaviors based on human in the loop coaching gestures," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2014, pp. 4770–4776.

[14] L. Righetti, J. Buchli, and A. J. Ijspeert, "Slow-fast dynamics of strongly coupled adaptive frequency oscillators," 2021.

[15] ——, "Dynamic hebbian learning in adaptive frequency oscillators," *Physica D: Nonlinear Phenomena*, vol. 216, no. 2, pp. 269–281, 2006.

[16] M. Khoramshahi, R. Nasiri, M. Shushtari, A. J. Ijspeert, and M. N. Ahmadabadi, "Adaptive natural oscillator to exploit natural dynamics for energy efficiency," *Robotics and Autonomous Systems*, vol. 97, pp. 51–60, 2017.

[17] T. Nachstedt, C. Tetzlaff, and P. Manoonpong, "Fast dynamical coupling enhances frequency adaptation of oscillators for robotic locomotion control," *Frontiers in neurorobotics*, vol. 11, p. 14, 2017.

[18] Y. Hu, W. Tian, J. Liang, and T. Wang, "Learning fish-like swimming with a cpg-based locomotion controller," in *2011 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2011, pp. 1863–1868.

[19] Y. Hu, J. Liang, and T. Wang, "Parameter synthesis of coupled nonlinear oscillators for cpg-based robotic locomotion," *IEEE Transactions on Industrial Electronics*, vol. 61, no. 11, pp. 6183–6191, 2014.

[20] T. Chuthong, B. Leung, K. Tiraborisute, P. Ngamkajornwiwat, P. Manoonpong, and N. Dilokthanakul, "Dynamical State Forcing on Central Pattern Generators for Efficient Robot Locomotion Control," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 12533 LNCS, pp. 799–810, 2020.

[21] J. Buchli, L. Righetti, and A. J. Ijspeert, "Engineering entrainment and adaptation in limit cycle systems," *Biological Cybernetics*, vol. 95, no. 6, p. 645, 2006.

[22] F. Pasemann, M. Hild, and K. Zahedi, "SO(2)-Networks as Neural Oscillators," in *Computational Methods in Neural Modeling*, J. Mira and J. R. Álvarez, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 144–151.

[23] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, *Deep learning*. MIT press Cambridge, 2016, vol. 1, no. 2.

[24] D. Owaki and A. Ishiguro, "A Quadruped Robot Exhibiting Spontaneous Gait Transitions from Walking to Trotting to Galloping," *Scientific Reports*, vol. 7, no. 1, p. 277, 2017. [Online]. Available: https://doi.org/10.1038/s41598-017-00348-9

[25] P. Manoonpong, U. Parlitz, and F. Wörgötter, "Neural control and adaptive neural forward models for insect-like, energy-efficient, and adaptable locomotion of walking machines," *Frontiers in neural circuits*, vol. 7, p. 12, 2013.

[26] M. Thor, J. Larsen, and P. Manoonpong, "Morf—modular robot framework," in *Proc. 2nd Int. Youth Conf. Bionic Eng.(IYCBE)*, 2018, pp. 21–23.

[27] M. Saveriano, F. J. Abu-Dakka, A. Kramberger, and L. Peternel, "Dynamic movement primitives in robotics: A tutorial survey," 2021.

[28] P. Ngamkajornwiwat, J. Homchanthanakul, P. Teerakittikul, and P. Manoonpong, "Bio-inspired adaptive locomotion control system for online adaptation of a walking robot on complex terrains," *IEEE Access*, vol. 8, pp. 91 587–91 602, 2020.

[29] E. Grinke, C. Tetzlaff, F. Wörgötter, and P. Manoonpong, "Synaptic plasticity in a recurrent neural network for versatile and adaptive behaviors of a walking robot," *Frontiers in neurorobotics*, vol. 9, p. 11, 2015.