2024 IEEE International Conference on Robotics and Automation (ICRA 2024)
May 13-17, 2024. Yokohama, Japan

Visual CPG-RL: Learning Central Pattern Generators for Visually-Guided Quadruped Locomotion

Guillaume Bellegarda, Milad Shafiee, Auke Ijspeert

***Abstract*— We present a framework for learning visually-guided quadruped locomotion by integrating exteroceptive sensing and central pattern generators (CPGs), i.e. systems of coupled oscillators, into the deep reinforcement learning (DRL) framework. Through both exteroceptive and proprioceptive sensing, the agent learns to coordinate rhythmic behavior among different oscillators to track velocity commands, while at the same time override these commands to avoid collisions with the environment. We investigate several open robotics and neuroscience questions: 1) What is the role of explicit interoscillator couplings between oscillators, and can such coupling improve sim-to-real transfer for navigation robustness? 2) What are the effects of using a memory-enabled vs. a memory-free policy network with respect to robustness, energy-efficiency, and tracking performance in sim-to-real navigation tasks? 3) How do animals manage to tolerate high sensorimotor delays, yet still produce smooth and robust gaits? To answer these questions, we train our perceptive locomotion policies in simulation and perform sim-to-real transfers to the Unitree Go1 quadruped, where we observe robust navigation in a variety of scenarios. Our results show that the CPG, explicit interoscillator couplings, and memory-enabled policy representations are all beneficial for energy efficiency, robustness to noise and sensory delays of 90 ms, and tracking performance for successful sim-to-real transfer for navigation tasks.**

I. Introduction

Animals perform complex navigation tasks over variable terrain in the search for prey or to escape predators. In order to plan and execute such agile behaviors in unknown environments, exteroceptive sensing is necessary for both planning and control purposes (i.e. anticipatory behavior vs. reactive behavior from "blind" walking). For both animals and robots, adding exteroception is both an opportunity (anticipation, planning) and a challenge (higher dimensional measurements, noise) for a control architecture. On the other hand, animals process such high-dimensional information very quickly, and have internal mechanisms that share control between the spinal cord and higher control centers (e.g. motor cortex and cerebellum). This avoids the concept that all motor commands come from higher control centers (i.e. the biological parallel of current optimal control methods and learning-based policies). Towards a better biological parallel, in this work we represent higher control centers with an artificial neural network, which sends modulation signals to the Central Pattern Generator (CPG) in the spinal cord. The CPG is represented as a system of oscillators, and its states are modulated through feedback from both exteroceptive (i.e. visual features) and proprioceptive (i.e. base velocities, contact feedback) sensing to produce robust navigation policies. This framework enables us to investigate several scientific questions, and in particular we study 1) the role of interoscillator couplings in the spinal cord, 2) the role of a memory-enabled vs. a memory-free neural network as a higher control center, and 3) the effects of sensory delays on motor performance. The answers to these questions have not yet been confirmed from biology for animal navigation, nor addressed in robotics for robot obstacle avoidance. Here, we train and deploy 12 different navigation policies in over 250 sim-to-real experiments. Our results suggest that CPGs are useful intermediary control layers compared to direct joint control, that memory-enabled networks perform better than memory-free networks, and that when vision is included, interoscillator couplings are useful vs. no couplings.

Fig. 1: Visual CPG-RL perceptive locomotion on Unitree Go1.

A. Related Work

1) Biology-Inspired Control: Central Pattern Generators are neural networks found in the spinal cord and brainstem of vertebrates which are capable of generating rhythmic patterns of muscle activity, such as those involved in walking, swimming, and flying [1]. For quadrupedal robots, CPGs have been used as a tool to both replicate and understand locomotion from several points of view, including the role of sensory input, reflexes, and mechanical design inspired from biology [2], [3]. While most works integrating sensory feedback with CPGs have focused on using proprioceptive sensing, incorporating cameras and other external sensory feedback have demonstrated walking over varying terrains [4], and responding to sudden obstacle appearances [5], [6].

This research is supported by the Swiss National Science Foundation (SNSF) as part of project No.197237. The authors are with the BioRobotics Laboratory, Ecole Polytechnique Federale de Lausanne (EPFL). {firstname.lastname}@epfl.ch

2) Model-Based Control: Model Predictive Control (MPC) approaches have demonstrated that using simplified dynamics models with filtered perception data as constraints during the optimization process can produce robust locomotion skills [7]–[10]. These works build an elevation map of the environment through one or more depth cameras [11], [12], and the resulting grid map [13] is postprocessed for smoothing, inpainting, or plane segmentation, depending on the application.

3) Learning-Based Control: Learning-based control has also demonstrated impressive real-world capabilities, even for "blind" policies which have access only to proprioceptive sensing [14]–[23]. Similarly to recent MPC approaches, integrating perception into such learning-based control approaches can lead to even more robust locomotion. Demonstrative applications include combining occupancy maps with proprioceptive sensing for navigation [24], obstacle avoidance with end-to-end training directly from pixels [25], [26], local navigation for challenging short time horizon tasks [27], and crossing rough terrain dynamically by sampling height maps extracted from 3D Lidar [28], [29]. Other works have decoupled footstep planning processes from the control policy, either each trained with DRL [30], or leveraging MPC to track the footstep plan [31]. Learning a state representation of the environment from depth images and sending high-level commands to a separately trained DRL control policy also allows navigating cluttered environments [32]. Additional works have used vision to demonstrate variable gap crossing capabilities [33]–[37].

In our previous work, we proposed CPG-RL [38], a framework for using deep reinforcement learning to directly learn the time-varying oscillator intrinsic amplitude and frequency for each oscillator which together forms a central pattern generator. We implemented the CPG network with one oscillator per limb, but without explicit couplings between oscillators as they did not prove beneficial for locomotion with only proprioceptive sensing.

B. Contribution

In this paper, we present a framework for incorporating terrain-awareness as exteroceptive feedback to the policy network which in turn modulates the CPG, enabling our quadruped to navigate efficiently in cluttered terrains. With our framework, we center our scientific investigation around three fundamental robotics and neuroscience questions:

1. What is the role of explicit interoscillator couplings between oscillators, and can such coupling improve the sim-to-real transfer for perceptive locomotion tasks?
2. What are the effects of using a memory-enabled vs. a memory-free policy network with respect to robustness, energy-efficiency, and tracking performance in the sim-to-real navigation task?
3. Does the CPG (with/without coupling) increase robustness with respect to baselines when sensory delays are present?

For question (1), we train perceptive locomotion policies with varying explicit coupling factors between oscillators in the dynamics equations. While such couplings between oscillators are known to exist in biological CPGs, recent work has shown that they might not be as strong as previously thought [39], [40], and that sensory feedback and descending modulation might play an important role in interoscillator synchronization. For the sim-to-real transfer, in contrast with our previous work on "blind" locomotion [38], we find that explicit coupling improves policy robustness when concatenating high-dimensional (and noisy) exteroceptive inputs with the lower-dimensional proprioceptive sensing.

For question (2), we test different neural network architectures, specifically purely feedforward networks (MLPs) as well as memory-enabled networks (LSTMs), and find that the memory-enabled networks produce more robust and energy-efficient policies for sim-to-real transfers.

For question (3), we compare and evaluate our method with a joint-space policy baseline and find that our architecture, and the CPG in particular, is able to maintain robustness to large sensory delays, comparable to those known to exist in similarly sized mammals [41], [42].

The rest of this paper is organized as follows. In Section II we present Visual CPG-RL, including our design choices and integration of Central Pattern Generators and exteroceptive sensing into the deep reinforcement learning framework. In Section III we discuss results and analysis from learning our controller and sim-to-real transfers for varying neural network architectures, specified interoscillator couplings, and sensory delays, and we give a brief conclusion in Section IV.

II. Learning Central Pattern Generators for Visually-Guided Locomotion

In this section we describe our CPG-integrated deep reinforcement learning framework and design decisions for learning visually-guided locomotion controllers for quadruped robots. The agent receives exteroceptive and proprioceptive sensing measurements and the current CPG state as input, and learns to modulate and coordinate the CPG parameters for each limb to track velocity commands while avoiding collisions. A high-level control diagram is illustrated in Figure 2, and we explain all components below.

A. Central Pattern Generators and Action Space

Based on our previous work [38], in this work we propose the following amplitude-controlled phase oscillators to coordinate locomotion between each limb i:

$$\ddot{r}_{x_i} = a_x\left(\frac{a_x}{4}(\mu_{x_i} - r_{x_i}) - \dot{r}_{x_i}\right) \tag{1}$$

$$\ddot{r}_{y_i} = a_y\left(\frac{a_y}{4}(\mu_{y_i} - r_{y_i}) - \dot{r}_{y_i}\right) \tag{2}$$

$$\dot{\theta}_i = \omega_i + \frac{1}{2}\sum_j (r_{x_j} + r_{y_j}) w_{ij} \sin(\theta_j - \theta_i - \phi_{ij}) \tag{3}$$

where r_{x_i} and r_{y_i} are the current amplitudes of the oscillator, θ_i is the phase of the oscillator, μ_i and ω_i are the intrinsic amplitude and frequency, a is a positive constant representing the convergence factor. Couplings between oscillators are defined by the weights w_{ij} and phase biases ϕ_{ij}.

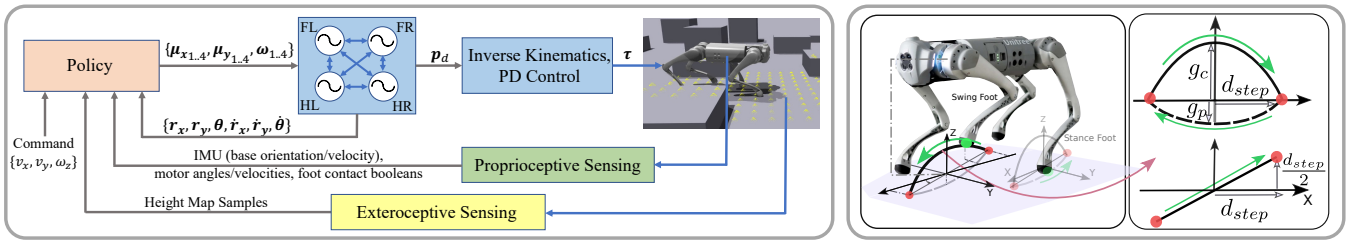

Fig. 2: **(a)**: Control architecture for learning central pattern generators for visually-guided quadruped locomotion. The observation consists of velocity commands, exteroceptive measurements, proprioceptive measurements, and the current CPG states, which the policy network uses to select CPG parameters μ_x, μ_y, and ω for each leg i (Front Left (FL), Front Right (FR), Hind Left (HL), Hind Right (HR)). The resulting CPG states are mapped to desired foot positions $\boldsymbol{p}_d$, which are then converted to desired joint angles with inverse kinematics, and finally tracked with joint PD control to produce torques $\boldsymbol{\tau}$. The control policy selects actions at 100 Hz, and all other blocks operate at 1 kHz. **(b)**: Mapping CPG states to Cartesian foot positions. Left: feet path during swing and stance phases. Top right in the (vertical) XZ-plane: ground clearance (g_c), ground penetration (g_p), max step length (d_{step}) are design parameters, whereas CPG states r_x and θ control amplitude and phase. Bottom right in the (horizontal) XY-plane: coordinating omnidirectional motion in the leg frame (arrow shows swing phase motion) with converged amplitude set points $\mu_x = 2$, $\mu_y = 1.25$, representing the full d_{step} and $\frac{1}{2}d_{step}$, respectively.

Compared with previous work that typically uses a single amplitude and phase variable, here we have a third state variable to represent amplitude in the y direction. The agent can then learn to coordinate both amplitudes for omnidirectional locomotion (see Figure 2-b), and we update the coupling to reflect both x and y components. In particular, this mapping is different from our previous work [38], where we had added an extra phase variable to orient the foot, instead of the new proposed amplitude. We find this new mapping is able to achieve higher returns during training.

Our action space provides an interface for the agent to directly modulate the intrinsic oscillator amplitudes and phases, by learning to modulate μ_{x_i}, μ_{y_i}, and ω_i for each leg. This allows the agent to adapt each of these states online in real-time depending on sensory inputs, in contrast with the more traditional CPG approach of optimizing for a specific fixed open-loop gait. Thus, for the omnidirectional perceptive locomotion task, our action space can be summarized as $\mathbf{a} = [\boldsymbol{\mu}_x, \boldsymbol{\mu}_y, \boldsymbol{\omega}] \in \mathbb{R}^{12}$. The agent selects these parameters at 100 Hz, and we use the following action space ranges during training: $\mu_x, \mu_y \in [1,2]$, $\omega \in [0, 4.5]$ Hz, and $a_x = a_y = 50$.

The oscillator states are mapped to joint commands by first computing corresponding desired foot positions, and then calculating the desired motor angles with inverse kinematics. This is an approximation of the typical two layers found in mammalian CPGs, with one rhythm generating layer, and one pattern formation layer [43], with here the pattern formation layer implementing the inverse kinematics. The desired foot position coordinates are computed as:

$$x_{i,\text{foot}} = -d_{step} f(r_{x_i}) \cos(\theta_i) \tag{4}$$

$$y_{i,\text{foot}} = \quad d_{step} f(r_{y_i}) \cos(\theta_i) \tag{5}$$

$$z_{i,\text{foot}} = \begin{cases} -h + g_c \sin(\theta_i) & \text{if } \sin(\theta_i) > 0 \\ -h + g_p \sin(\theta_i) & \text{otherwise} \end{cases} \tag{6}$$

where d_{step} is the maximum step length, h is the robot height, g_c is the max ground clearance during swing, g_p is the max ground penetration during stance, and $f(r) = 2\frac{(r - \mu_{\min})}{(\mu_{\max} - \mu_{\min})} - 1$. In this mapping, since r_{x_i} and r_{y_i} each vary between $\mu_{\min} = 1$ and $\mu_{\max} = 2$, the foot can vary within $\pm d_{step}$ in both x and y directions in each leg frame.

Figure 2-b illustrates the foot trajectories for a set of these parameters, which greatly simplifies specifying behaviors that are challenging to learn when directly learning joint commands. As in [38], we sample h, g_c, and g_p during training so the agent can learn to locomote with varying base heights, swing foot ground clearances, and stance foot ground penetrations. The agent does not receive any explicit observation of these parameters, and the user can specify each of these parameters during deployment.

B. Observation Space

Our observation space consists of velocity commands and measurements available with proprioceptive sensing as in [38], as well as exteroceptive sensing. The exteroceptive measurements consist of querying a terrain height map at a 17×11 grid spaced at intervals of 0.1 m around the robot base (as shown by the yellow dots in Figure 2-a). In simulation, the ground truth terrain height data is known, and on hardware such a grid can be estimated, for example by using depth cameras to build an elevation map [11], [12], and then querying the resulting postprocessed grid map [13].

The proprioceptive sensing includes the body state (orientation, linear and angular velocities), joint state (positions, velocities), and foot contact booleans. The last action chosen by the policy network and CPG states $\{\boldsymbol{r}_x, \dot{\boldsymbol{r}}_x, \boldsymbol{r}_y, \dot{\boldsymbol{r}}_y, \boldsymbol{\theta}, \dot{\boldsymbol{\theta}}\}$ are concatenated to the exteroceptive and proprioceptive measurements. While the exteroceptive and proprioceptive sensing are subject to measurement noise from onboard sensors, the CPG states are always known both in simulation and during hardware experiments, easing the sim-to-real transfer. Sensorimotor delays are present in both animals and robots, and the exteroceptive (vision) and proprioceptive sensing loops operate at different frequencies. We investigate the effects of sensory delays in Section III-C.

C. Reward Function

Our reward function primarily rewards tracking body linear and angular velocity commands in the base frame (x and y directions, as well as yaw rate $\omega^*_{b,z}$). We also add terms to minimize other undesired base velocities (vertical oscillations in the base z direction, and base roll and pitch rates). To minimize energy consumption, we penalize the total power. The terms and respective weights

TABLE I: Reward function terms. $(\cdot)^*$ represents a desired command, and $f(x) := \exp(-\frac{||x||^2}{0.25})$. $dt = 0.01$ is the control policy time step.

Name	Formula	Weight
Linear velocity tracking $v^*_{b,x}$	$f(v^*_{b,x} - v_{b,x})$	$3dt$
Linear velocity tracking $v^*_{b,y}$	$f(v^*_{b,y} - v_{b,y})$	$0.75dt$
Angular velocity tracking $\omega^*_{b,z}$	$f(\omega^*_{b,z} - \omega_{b,z})$	$0.5dt$
Linear velocity penalty $v_{b,z}$	$-v^2_{b,z}$	$2dt$
Angular velocity penalty $\boldsymbol{\omega}_{b,xy}$	$-\|\|\boldsymbol{\omega}_{b,xy}\|\|^2$	$0.05dt$
Power	$-\|\boldsymbol{\tau} \cdot \dot{\boldsymbol{q}}\|$	$0.001dt$

are summarized in Table I. Similarly to [38], we do not need to add any reward terms beyond those fully specifying the base motion behavior. Due to the variable height terrain in the training environment, however, it is not possible to track all velocity commands at all times (i.e. if there are obstacles in the way). As shown in the Table, we more heavily weight the forward velocity reward term $v^*_{b,x}$ so the policy learns to prefer deviations to the other velocity commands/penalties (i.e. to turn if approaching an obstacle head-on).

D. Neural Network Architectures

To address our question about the importace of memory (question 2), we consider two different neural network architectures to map the concatenated proprioceptive and exteroceptive sensing observation to actions modulating the intrinsic oscillator amplitudes and phases. The first is a purely feedforward network, or multi-layer perceptron (MLP), consisting of three hidden layers of [512, 256, 128] hidden units per layer. The second architecture is a memory-enabled network, consisting of a Long Short-Term Memory (LSTM) layer of 512 hidden units, followed by two fully connected layers of [256, 128] hidden units. The memory-enabled network has a better biological parallel, and we also anticipate this will provide better robustness for the sim-to-real transfer in the event of noisy measurements and latency.

E. Training Details

We use Isaac Gym and PhysX as our training environment and physics engine [29], [44], and the Unitree Go1 quadruped [45]. This framework has high throughput, enabling us to simulate 4096 Go1s in parallel on a single NVIDIA RTX 3090 GPU. We use the Proximal Policy Optimization (PPO) algorithm [46] to train the policy, with the same hyperparameters as in [38]. Similarly to [29], [38], this framework allows us to learn control policies within minutes.

During training, we reset the environment for an agent if the base or a thigh comes in contact with the terrain (i.e. with either a box or the ground, so the agent learns to avoid collisions), or if the episode length reaches 20 seconds. We employ a terrain curriculum starting from flat terrain, to random boxes of varying widths (0.4 to 2 $\mathtt{m}$) and heights (0.1 to 1 $\mathtt{m}$) so the agent can learn to avoid obstacles in a variety of scenarios. With each reset, we sample new parameters h and g_c for mapping the oscillator states to motor commands, allowing the agent to learn continuous locomotion behavior at varying body heights and step heights. New velocity commands $\{v^*_{b,x}, v^*_{b,y}, \omega^*_{b,z}\}$ are sampled every 5 seconds, though as explained in Section II-C, these cannot be perfectly tracked due to the presence of obstacles, which forces the agent to learn to deviate from commands to avoid terminating the episode. We also apply domain randomization on the physical mass properties and coefficient of friction, as summarized in Table II. An external push of up to 0.5 $\mathtt{m/s}$ is applied in a random direction to the base every 15 seconds. While no noise is added to the proprioceptive measurements, we add Gaussian noise to the exteroceptive measurements with a standard deviation of 0.1.

TABLE II: Randomized parameters during training and their ranges.

Parameter	Lower Bound	Upper Bound	Units
$v^*_{b,x}$	-0.6	0.6	$\mathtt{m/s}$
$v^*_{b,y}$	-0.4	0.4	$\mathtt{m/s}$
$\omega^*_{b,z}$	-0.8	0.8	$\mathtt{rad/s}$
Joint Gain K_p	55	100	-
Joint Gain K_d	0.7	2.5	-
Mass (each body link)	70	130	%
Added base mass	0	5	kg
Coefficient of friction	0.3	1	-

The policy network outputs modulation signals at 100 Hz, and the torques computed from the mapped desired joint positions are updated at 1 kHz. The equations for each of the oscillators (Equations 1-3) are thus also integrated at 1 kHz. During training we re-sample joint PD controller gains at each environment reset as described in Table II, but during deployment we use $K_p = 100$, $K_d = 2$.

F. Sim-to-Real Transfer

As in [7], we mount two Intel RealSense cameras (D435i and T265) on the Unitree Go1 quadruped [45]. The D435i is mounted forward-facing to provide point clouds to elevation mapping software [11], [12] to construct a map of the area surrounding the robot. The T265 provides high accuracy localization. To run the policy, we query this map at the same 17×11 points around the robot as seen during training (i.e. yellow dots in Figure 2), and concatenate the proprioceptive sensing measurements as read by the Unitree sensors, along with the CPG states and previous actions.

III. Experimental Results and Discussion

In this section we report and discuss results from learning visually-guided locomotion controllers with Visual CPG-RL. Sample snapshots of one of our perceptive locomotion policy deployments are shown in Figure 1, and the reader is encouraged to watch the supplementary video for clear visualizations of the discussed experiments. Our experiments are designed to investigate the three questions detailed in the introduction.

A. Role of Policy Architecture and Interoscillator Coupling

We train policies with both neural network architectures (MLP and LSTM) described in Section II-D. For each architecture, we train separate policies with increasingly strong interoscillator coupling weights, namely $w_{i,j} = \{0, 0.2, 0.4, 0.6, 0.8, 1.0\}$ in Equation 3, for a total of 6 MLP policies and 6 LSTM policies. We train several policies with different random seeds within each of these categories, and select the one achieving the highest return for deployment. We use a trot gait coupling matrix, since without coupling (i.e. $w_{i,j} = 0$), all policies learn an approximate trot gait.

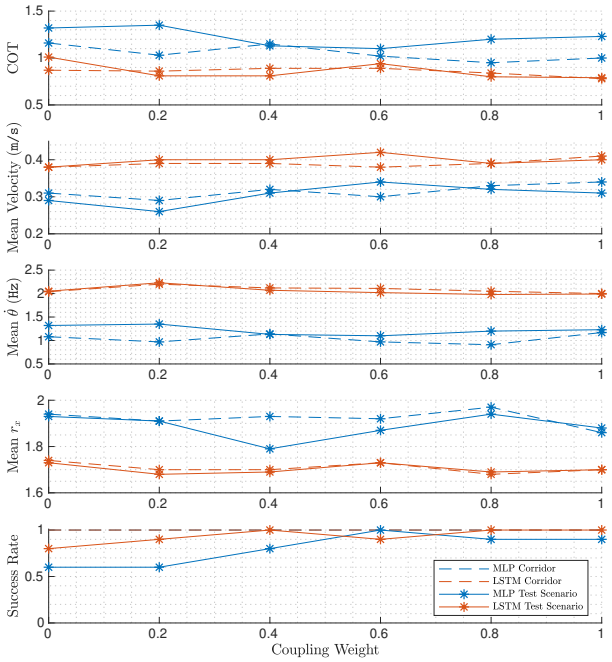

Fig. 3: Sim-to-real tracking performance of $v^*_{b,x} = 0.35$ m/s in a wide corridor (dashed line) and on a test navigation environment involving both left and right turns (solid lines), with both MLP and LSTM policies trained with varying coupling weights ($w_{i,j}$ in Equation 3). Each data point represents the mean of 10 trials. From top to bottom, we present the mean Cost of Transport (COT), the quadruped base mean velocity, the mean frequency across all oscillators ($\dot{\theta}$), the mean amplitude r_x correlated with the mean step length, and the success rate denoting avoidance of obstacle collisions or falls.

1) Corridor Test: We first test all policies in a 1.7 m wide corridor and command a forward velocity of $v^*_{b,x} = 0.35$ m/s for 10 seconds. This test is done to ensure successful sim-to-real transfers on (mostly) flat terrain, and mean results from 10 rollouts of each policy are shown by the dashed lines in Figure 3. All policies successfully locomote with 100% success rate for both MLP and LSTM architectures, and with all varying coupling strengths. However, the LSTM policies have a lower Cost of Transport for all couplings. When comparing the CPG state mean amplitudes and frequencies, the LSTM policies locomote with a higher leg frequency but lower step length than the MLP policies. Due to having memory, the LSTM policies are able to better optimize quantities like stride length and stride frequency, leading to better energy-efficiency. The LSTM policies also track the commanded velocities slightly more closely, and interestingly overshoot the command, compared with the MLP policies which are consistently slower.

2) Navigation Test: We next test all policies in a navigation environment which requires both left and right turns in order to avoid obstacles. We define a failure as the robot colliding with an obstacle or falling down. We again command a forward velocity of $v^*_{b,x} = 0.35$ m/s for 10 seconds, and mean results from 10 rollouts of each policy are summarized by the solid lines in Figure 3. In order to avoid the obstacles, the agent is forced to violate at least one of the 0 velocity commands for $v^*_{b,y}$ and $\omega^*_{b,z}$. In these tests, for both the MLP and LSTM policies, we observe that coupling is beneficial for successful navigation of the terrain. The explicit coupling helps to reject noise and latency associated with the high-dimensional (relative to the proprioceptive) exteroceptive measurements. The LSTM policies also prove to be overall more robust than the MLP policies.

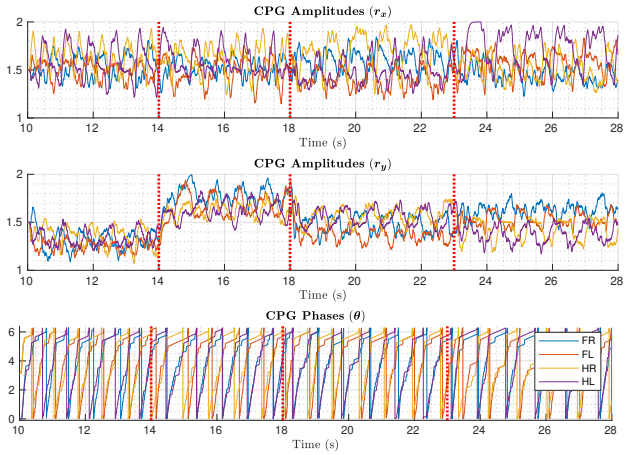

Fig. 4: CPG states during omnidirectional commands: $v^*_{b,y} = 0.4$ m/s from 10-14 s, $v^*_{b,y} = -0.4$ m/s from 14-18 s, $\omega^*_{b,z} = 0.7$ rad/s from 18-23 s, and $\omega^*_{b,z} = -0.7$ rad/s from 23-28 s. The y amplitudes r_y produce locomotion for the system for the lateral commands. For turning in place, we observe coordination between both x and y amplitudes.

B. CPG State Modulation for Omnidirectional Locomotion

We investigate how the agent modulates the CPG states to produce omnidirectional locomotion on flat terrain. Figure 4 shows the CPG states when commanded to move laterally in the body y directions, $v^*_{b,y} = \pm 0.4$ m/s, and then turn in place in both directions, $\omega^*_{b,z} = \pm 0.7$ rad/s. The video shows that the gait is smooth and consistent for the entirety of the motion, at a much lower frequency than the default Go1 controller. The lateral motions show the amplitudes y selecting the expected directions, and coordination can be seen between both x and y amplitudes for turning in place. An approximate trot gait can be observed throughout all commands. For turning left, for example, the hind right foot has the largest amplitude in the x direction, which combined with the y amplitude moving mostly right in the body frame, turns the system in the expected direction. The reverse is true for turning right, where the hind left foot has the largest amplitude in the x direction, and combines with the y amplitude now moving mostly left in the body frame.

In the video, we also show how the CPG states are modulated during a navigation test involving both left and right turns through exteroceptive feedback as discussed in Section III-A.2.

C. Sensory Delay Study

While in simulation it is possible to query any sensory information at any rate, real-world sensing will inevitably have latency issues, especially for vision-based systems. Although the control policy network is queried at 100 Hz, the terrain map on the hardware is built and updated at a lower frequency, meaning inevitable sensory delays and latency compared with in simulation. Several biological studies have shown that animals locomote with sensorimotor delays on the order of tens of milliseconds, increasing for

TABLE III: Success rate, mean body angular velocity $\bar{\omega}$, and mean joint accelerations $\bar{\ddot{\boldsymbol{q}}}$ for 100 quadrupeds tracking $v^*_{b,x} = 0.35$ m/s for 40 seconds with increasing sensory delays on the test environment of Figure 5. High angular velocities and joint accelerations correspond to shaky and non-optimal locomotion.

			Sensory Delay (s)											
			0			0.03			0.06			0.09		
Method	NN	w_{ij}	Success Rate	$\bar{\omega}$	$\bar{\ddot{\boldsymbol{q}}}$	Success Rate	$\bar{\omega}$	$\bar{\ddot{\boldsymbol{q}}}$	Success Rate	$\bar{\omega}$	$\bar{\ddot{\boldsymbol{q}}}$	Success Rate	$\bar{\omega}$	$\bar{\ddot{\boldsymbol{q}}}$
Visual CPG-RL	MLP	0	**1**	0.26	0.85	**1**	0.26	0.84	**1**	**0.36**	**0.86**	0.55	**0.56**	**0.92**
		1	**1**	**0.25**	**0.83**	**1**	**0.25**	**0.83**	**1**	0.38	**0.86**	**0.71**	**0.57**	**0.97**
	LSTM	0	**1**	0.28	0.89	**1**	0.28	0.90	**1**	0.46	1.04	0	0.55	1.25
		1	**1**	0.28	0.85	**1**	0.28	0.85	**1**	0.47	1.02	0	0.77	1.23
Joint PD	MLP	-	0.96	**0.25**	0.90	0.97	**0.25**	0.90	0.98	0.43	3.98	0.08	0.88	2.26
	LSTM	-	0.97	**0.25**	1.02	0.98	**0.25**	1.01	0.98	0.43	1.25	0	0.92	2.20

Fig. 5: Simulation test environment involving both left and right turns, as well as turning around an obstacle, as in the hardware experiments.

larger mammals [41]. Using the same scaling equation for the Go1 quadruped, which has a mass of 12 kg, would give an expected sensorimotor delay of $31 * 12^{0.21} = 52.24$ ms between stimulus onset and peak muscle force.

In this section, we compare the effects of introducing such sensory delays on both the proprioceptive and exteroceptive measurements with our hierarchical biology-inspired architecture. As a baseline and to study the usefulness of integrating the CPG that exists in vertebrates with different neural networks and coupling weights, we train joint PD policies with a similar reward function to [29]. We note that the reward function for the joint PD baseline is much more complex and has to be significantly tuned, in contrast with Visual CPG-RL, in order to get similarly natural-looking navigation policies. The policies are all trained in the same environment described in Section II-E.

We evaluate Visual CPG-RL and the joint PD baseline on a test environment consisting of left and right turns, as well as an obstacle to move around, where the agent is free to choose to go left or right before coming back the way it came, as shown in Figure 5. We let 100 agents for each method navigate in this terrain, and define a success as tracking the desired velocity of $v^*_{b,x} = 0.35$ m/s for 40 seconds without falling or colliding with the environment.

Table III summarizes results of testing the trained Visual CPG-RL policies with both neural network architectures (MLP and LSTM) and coupling weights $w_{i,j} = \{0,1\}$ (in Equation 3), as well as the joint PD baseline with both NN architectures. We can observe that all policies perform well without any sensory delays. Notably, the Visual CPG-RL architecture performs almost identically through delays of 60 ms, which is greater than the expected animal-estimated sensorimotor delay of 52.24 ms. As the observations are increasingly delayed to 90 ms, the joint PD policies cannot produce any successful locomotion, and behave erratically with high body angular velocities and large joint angular accelerations (red values in the Table). In contrast, we note that Visual CPG-RL MLP policies perform well, and much better than the LSTM policies, whose internal states are mismatched with the new observation latency compared with the training environment. We also observe that coupling appears to help the MLP policies perform better with increasing sensory delays. We note that the Visual CPG-RL policies are consistently smoother, as evidenced by the mean body angular velocity and joint accelerations being lower than the joint PD baseline, and are qualitatively more natural with increasing delays.

IV. Conclusion

In this work we have presented Visual CPG-RL, a framework for learning perceptive quadruped locomotion by integrating central pattern generators and exteroceptive sensing into the deep reinforcement learning framework. The agent learns to modulate the intrinsic oscillator amplitudes and frequencies to coordinate rhythmic behavior among limbs to track omnidirectional velocity commands, while also learning to deviate from the commands in order to avoid collisions with obstacles. We represented higher control centers in the brain with an artifical neural network (ANN), and showed that memory-enabled networks (i.e. LSTMs) provide higher robustness and encode more energy-efficient policies than feedforward networks (MLPs) for real-world navigation tasks which may have high-dimensional inputs, noise, and latency concerns (question 2). The ANN sends modulation signals to a system of oscillators in the spinal cord (CPG), and explicit couplings within the CPG dynamics equations proved to be beneficial in terms of robustness and stability for the sim-to-real navigation task (question 1). Compared with our previous work which did not investigate interoscillator couplings for "blind" locomotion [38], these new results suggest the reason direct coupling is present in animals, which rely heavily on vision, and gives robots the ability to learn and deploy robust, adaptive, and efficient policies when subject to noisy high-dimensional inputs. From a robotics perspective, coupling between oscillators improves stability, at the cost of potential less flexibility (i.e. it may be more difficult to recover from a push or other disturbance with strong and/or speed-dependent coupling [47]). Moreover, we studied the effects of sensory delays on policy robustness of each of the proposed neural network architectures and coupling weights, and found that the CPG is beneficial for suppressing sensory latency, and the performance was inline with estimated sensorimotor delays for a comparably sized animal [41] (question 3). Future work will focus on adding additional states to the CPG to account for rough terrain, as well as compare with joint-space CPGs, towards a better understanding of the biological parallels in learning legged locomotion.

REFERENCES

[1] 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, robotics and Neuroscience.

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

[3] A. Spröwitz, A. Tuleu, M. Vespignani, M. Ajallooeian, E. Badri, and A. J. Ijspeert, "Towards dynamic trot gait locomotion: Design, control, and experiments with cheetah-cub, a compliant quadruped robot," *The International Journal of Robotics Research*, vol. 32, no. 8, pp. 932–950, 2013.

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

[5] A. A. Saputra, N. Takesue, K. Wada, A. J. Ijspeert, and N. Kubota, "Aquro: A cat-like adaptive quadruped robot with novel bio-inspired capabilities," *Frontiers in Robotics and AI*, vol. 8, p. 562524, 2021.

[6] A. A. Saputra, J. Botzheim, A. J. Ijspeert, and N. Kubota, "Combining reflexes and external sensory information in a neuromusculoskeletal model to control a quadruped robot," *IEEE Transactions on Cybernetics*, 2021.

[7] D. Kim, D. Carballo, J. Di Carlo, B. Katz, G. Bledt, B. Lim, and S. Kim, "Vision aided dynamic exploration of unstructured terrain with a small-scale quadruped robot," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 2464–2470.

[8] R. Grandia, F. Jenelten, S. Yang, F. Farshidian, and M. Hutter, "Perceptive locomotion through nonlinear model predictive control," *arXiv preprint arXiv:2208.08373*, 2022.

[9] F. Jenelten, R. Grandia, F. Farshidian, and M. Hutter, "Tamols: Terrain-aware motion optimization for legged systems," *IEEE Transactions on Robotics*, 2022.

[10] A. Agrawal, S. Chen, A. Rai, and K. Sreenath, "Vision-aided dynamic quadrupedal locomotion on discrete terrain using motion libraries," in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 4708–4714.

[11] P. Fankhauser, M. Bloesch, and M. Hutter, "Probabilistic terrain mapping for mobile robots with uncertain localization," *IEEE Robotics and Automation Letters (RA-L)*, vol. 3, no. 4, pp. 3019–3026, 2018.

[12] P. Fankhauser, M. Bloesch, C. Gehring, M. Hutter, and R. Siegwart, "Robot-centric elevation mapping with uncertainty estimates," in *International Conference on Climbing and Walking Robots (CLAWAR)*, 2014.

[13] P. Fankhauser and M. Hutter, "A Universal Grid Map Library: Implementation and Use Case for Rough Terrain Navigation," in *Robot Operating System (ROS) – The Complete Reference (Volume 1)*, A. Koubaa, Ed. Springer, 2016, ch. 5.

[14] J. Tan, T. Zhang, E. Coumans, A. Iscen, Y. Bai, D. Hafner, S. Bohez, and V. Vanhoucke, "Sim-to-real: Learning agile locomotion for quadruped robots." in *Robotics: Science and Systems*, 2018.

[15] A. Iscen, K. Caluwaerts, J. Tan, T. Zhang, E. Coumans, V. Sindhwani, and V. Vanhoucke, "Policies modulating trajectory generators," in *Conference on Robot Learning*. PMLR, 2018, pp. 916–926.

[16] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun, and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Science Robotics*, vol. 4, no. 26, 2019.

[17] A. Kumar, Z. Fu, D. Pathak, and J. Malik, "Rma: Rapid motor adaptation for legged robots," *arXiv preprint arXiv:2107.04034*, 2021.

[18] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Science Robotics*, vol. 5, no. 47, 2020.

[19] S. Chen, B. Zhang, M. W. Mueller, A. Rai, and K. Sreenath, "Learning torque control for quadrupedal locomotion," *arXiv preprint arXiv:2203.05194*, 2022.

[20] G. Bellegarda and K. Byl, "Training in task space to speed up and guide reinforcement learning," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2019, pp. 2693–2699.

[21] G. Bellegarda and Q. Nguyen, "Robust quadruped jumping via deep reinforcement learning," *arXiv preprint arXiv:2011.07089*, 2020.

[22] G. Bellegarda, Y. Chen, Z. Liu, and Q. Nguyen, "Robust high-speed running for quadruped robots via deep reinforcement learning," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2022, pp. 10 364–10 370.

[23] M. Shafiee, G. Bellegarda, and A. Ijspeert, "Manyquadrupeds: Learning a single locomotion policy for diverse quadruped robots," *arXiv preprint arXiv:2310.10486*, 2023.

[24] Z. Fu, A. Kumar, A. Agarwal, H. Qi, J. Malik, and D. Pathak, "Coupling vision and proprioception for navigation of legged robots," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 17 273–17 283.

[25] R. Yang, M. Zhang, N. Hansen, H. Xu, and X. Wang, "Learning vision-guided quadrupedal locomotion end-to-end with cross-modal transformers," in *International Conference on Learning Representations*, 2021.

[26] C. S. Imai, M. Zhang, Y. Zhang, M. Kierebinski, R. Yang, Y. Qin, and X. Wang, "Vision-guided quadrupedal locomotion in the wild with multi-modal delay randomization," *arXiv preprint arXiv:2109.14549*, 2021.

[27] N. Rudin, D. Hoeller, M. Bjelonic, and M. Hutter, "Advanced skills by learning locomotion and local navigation end-to-end," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 2497–2503.

[28] T. Miki, J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun, and M. Hutter, "Learning robust perceptive locomotion for quadrupedal robots in the wild," *Science Robotics*, 2022.

[29] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *5th Annual Conference on Robot Learning*, 2021.

[30] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3699–3706, 2020.

[31] S. Gangapurwala, M. Geisert, R. Orsolino, M. Fallon, and I. Havoutis, "Rloc: Terrain-aware legged locomotion using reinforcement learning and optimal control," *IEEE Transactions on Robotics*, 2022.

[32] D. Hoeller, L. Wellhausen, F. Farshidian, and M. Hutter, "Learning a state representation and navigation in cluttered and dynamic environments," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 5081–5088, 2021.

[33] W. Yu, D. Jain, A. Escontrela, A. Iscen, P. Xu, E. Coumans, S. Ha, J. Tan, and T. Zhang, "Visual-locomotion: Learning to walk on complex terrains with vision," in *5th Annual Conference on Robot Learning*, 2021.

[34] K.-H. Lee, O. Nachum, T. Zhang, S. Guadarrama, J. Tan, and W. Yu, "Pi-ars: Accelerating evolution-learned visual-locomotion with predictive information representations," in *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2022, pp. 1447–1454.

[35] G. B. Margolis, T. Chen, K. Paigwar, X. Fu, D. Kim, S. bae Kim, and P. Agrawal, "Learning to jump from pixels," in *5th Annual Conference on Robot Learning*, 2021.

[36] M. Shafiee, G. Bellegarda, and A. Ijspeert, "Puppeteer and marionette: Learning anticipatory quadrupedal locomotion based on interactions of a central pattern generator and supraspinal drive," in *2023 IEEE International Conference on Robotics and Automation (ICRA)*, 2023, pp. 1112–1119.

[37] M. Shafiee, G. Bellegarda, and A. Ijspeert, "Deeptransition: Viability leads to the emergence of gait transitions in learning anticipatory quadrupedal locomotion skills," *arXiv preprint arXiv:2306.07419*, 2023.

[38] G. Bellegarda and A. Ijspeert, "CPG-RL: Learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 12 547–12 554, 2022.

[39] D. Owaki and A. Ishiguro, "A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping," *Scientific reports*, vol. 7, no. 1, pp. 1–10, 2017.

[40] R. Thandiackal, K. Melo, L. Paez, J. Herault, T. Kano, K. Akiyama, F. Boyer, D. Ryczko, A. Ishiguro, and A. J. Ijspeert, "Emergence of robust self-organized undulatory swimming based on local hydrodynamic force sensing," *Science Robotics*, vol. 6, no. 57, 2021.

[41] H. L. More and J. M. Donelan, "Scaling of sensorimotor delays in terrestrial mammals," *Proceedings of the Royal Society B*, vol. 285, no. 1885, p. 20180613, 2018.

[42] M. S. Ashtiani, A. Aghamaleki Sarvestani, and A. Badri-Spröwitz, "Hybrid parallel compliance allows robots to operate with

sensorimotor delays and low control frequencies," *Frontiers in Robotics and AI*, vol. 8, p. 645748, 2021.
[43] D. A. McCrea and I. A. Rybak, "Organization of mammalian locomotor rhythm and pattern generation," *Brain research reviews*, vol. 57, no. 1, pp. 134–146, 2008.
[44] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa *et al.*, "Isaac gym: High performance gpu-based physics simulation for robot learning," *arXiv preprint arXiv:2108.10470*, 2021.
[45] Unitree Robotics. Go1. https://www.unitree.com/products/go1/.
[46] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.
[47] V. Berendes, S. N. Zill, A. Büschges, and T. Bockemühl, "Speed-dependent interplay between local pattern-generating activity and sensory signals during walking in drosophila," *Journal of Experimental Biology*, vol. 219, no. 23, pp. 3781–3793, 2016.