Learning Emergent Gaits with Decentralized Phase Oscillators: on the role of Observations, Rewards, and Feedback

Jenny Zhang, Steve Heim, Se Hwan Jeon, Sangbae Kim

Abstract— We present a minimal phase oscillator model for learning quadrupedal locomotion. Each of the four oscillators is coupled only to itself and its corresponding leg through local feedback of the ground reaction force, which can be interpreted as an observer feedback gain. We interpret the oscillator itself as a latent contact state-estimator. Through a systematic ablation study, we show that the combination of phase observations, simple phase-based rewards, and the local feedback dynamics induces policies that exhibit emergent gait preferences, while using a reduced set of simple rewards, and without prescribing a specific gait. The code is open-source, and a video synopsis available at **https://youtu.be/1NKQ0rSV3jU**.

I. INTRODUCTION

Quadrupedal animals exhibit a variety of gaits, or pattern of footfalls, and the choice of gaits has been linked to energetics, speed, morphology, etc. [1]–[3]. Quadrupedal robot controllers, on the other hand, are typically designed around a fixed contact sequence, for a number of reasons. First, the energetic difference between gaits for robots has been shown to be inconsequential [4], [5]. Perhaps more importantly, using a single gait greatly simplifies the controller design; for conventional model-predictive control, pre-specifying the contact sequence allows a significantly simpler problem to be solved [6]–[8]. Model-free reinforcement learning (RL) side-steps the computational burden of reasoning over different contact sequences. Nonetheless, RL typically requires extensive reward shaping and regularization, which is often encoded in a time-indexed reference trajectory based on a fixed gait, either as a nominal trajectory [9]–[11] or a reward [12]. A periodic clock observation is then necessary to maintain the Markov property of the reference.

It is generally difficult to design shaping rewards that capture a general high-level notion, in our context "locomote with a regular gait", without over-specifying the solution. Siekmann, Godse, *et al.* [13] proposed a simple phase-based reward to encourage stance or swing at any phase difference for their bipedal robot Cassie, and demonstrated this policy can then track any desired gait by simply adjusting the phase difference between the legs accordingly. Similar work has applied this type of reward to quadrupeds as well [14], [15]. In these cases, the actual gait choice is removed from the policy, which is essentially treated as a low-level policy. Instead, the phase difference needs to be specified by the user or a high-level policy [16].
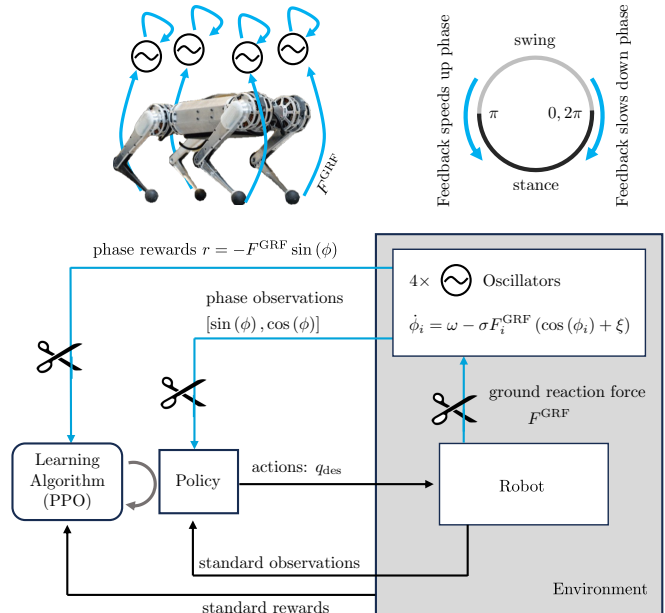
Fig. 1: We augment the robot state with four decentralized phase oscillators, one per leg. Blue arrows in the diagram indicate the three oscillator-related signals: first, observations of the oscillator phase to the policy to make feed-forward Markov. Second, the phase-based reward encodes the general properties of gaits. Finally, the ground reaction force (F^{GRF}) is used as feedback, which we view as the observer feedback that allows us to interpret the phase oscillators as a state observer of whether each foot should be in stance or swing. The scissors represent our ablation study.

A popular approach to achieving specific phase differences is to augment the state space with a network of coupled phase oscillators called central pattern generators (CPGs) [17]. In its simplest form, the dynamics of the oscillators are designed to exhibit stable limit cycles, and the resulting phases are mapped to values relevant to the robot controller, such as desired kinematics. In essence, this is a generalization of time-indexed trajectories. Despite the often simpler and lower-dimensional design space of the phase oscillators (compared to the space of reference trajectories in joint space), it can still be difficult to design the oscillator dynamics and mapping [18]. Recent works have instead opted to learn the coupling, while fixing the mapping from the phase to robot states [16], [19]. This approach complements the phase-based low-level policies, and CPGs are often interpreted as feed-forward reference generators [20], [21]. An alternative view, recently proposed by Ryu and Kuo [22], interprets the CPG as an observer, with reflex-like sensory feedback playing the role of the observer gain.

We present an implementation of *decentralized* phase oscillators based on the work of Owaki, Kano, *et al.* [23], and, based on the observer-interpretation of Ryu and Kuo [22],

treat the oscillator phase as a loose estimate of whether each leg should be in swing or stance. Based on this interpretation, we use a reward similar to Siekmann, Godse, *et al.* [13] to encourage the policy and phase oscillators to entrain. We further enrich the oscillator dynamics with an offset term to entrain the phases for standing still while retaining the same reward structure.

The three key signals afforded by our architecture, represented in Fig. 1 with blue arrows, are the *phase observation* which renders feed-forward policies Markov, the *phase rewards* that encode the high-level gait properties of duty factor and nominal frequency, and the *local feedback coupling* through the ground reaction force coupling the oscillator dynamics and the policy. We will show the importance of each of these through a systematic ablation.

II. PRELIMINARIES

We introduce gaits and pattern generation based on phase oscillators. We assume working knowledge of reinforcement learning and quadrupedal robots; for the interested reader, we recommend [24], [25].

A. Gaits

For the scope of this paper, we restrict ourselves to quadrupedal gaits, which are defined by the pattern of stance phases [3]. In the rest of this paper, we will define gaits by the relative phase difference (RPD); using the right front (RF) foot as the reference, we first define the gait cycle length as the time between consecutive RF foot touchdowns, normalized to 2π. We then calculate the RPD as a 3D vector, composed of the time difference between touchdowns of the left front (LF), right hind (RH), and left hind (LH) feet to the RF reference foot, normalized by the gait cycle length. Each gait is fully defined by these three values. The ideal symmetric gaits **Trot** $(\pi, \pi, 0)$, **Pace** $(\pi, 0, \pi)$, **Bound** $(0, \pi, \pi)$, **Pronk** $(0, 0, 0)$ are marked in Fig. 4.

We classify gaits by averaging the RPD over two gait cycles every 5 seconds, then taking the closest ideal gait within the set of symmetric quadruped gaits by Euclidean distance in RPD space. If the distance to the closest ideal gait exceeds 2, we classify the RPD as being in **Transition**.

B. Phase-based Pattern Generation

We will distinguish between clocks, central pattern generators (CPGs) without feedback, CPGs with feedback, and decentralized oscillators (which are driven by feedback by definition). Each of these is a special case of a system of oscillators with state $\phi \in [0, 2\pi)^n$, where $n \in \mathbb{N}$ is the number of oscillators, and

$$\dot{\phi}_i = \omega + f(\phi_j, x_j), \; j \in C(i)$$
$$u = g(\phi_j, x_j) \tag{1}$$
$$\text{for } i \in \{1, \ldots, n\}, \text{ and } C(i) \subseteq \{1, \ldots, n\}$$

where ω is a nominal frequency, x is the system state, $f()$ is a function that determines the dynamics properties, $g()$ is a mapping function to some control-relevant input u, and the set $C()$ indicates which oscillators are directly coupled. We

will generally consider the case where there is one oscillator per leg, that is $n = 4$ for a quadruped.

Clocks: $f() = 0$
This degenerate choice for $f()$ reduces the oscillator to a clock with a constant growth rate ω. Though sometimes called CPGs [9], we distinguish this setting as it makes no use of the state and dynamics of the oscillator: the burden is placed on designing the map $g()$. This is the minimal form needed to make a cyclic feed-forward pattern Markov. Siekmann, Godse, *et al.* [13] use this setup, and learn the mapping function $g(\phi, x_j)$, using the clock as both an observation and to design a simple reward function.

CPGs without feedback: $f() = f(\phi_j)$
This form provides a pure feed-forward pattern, and allows the engineer to design the oscillator dynamics, such as limit cycles and convergence properties [18], [26], unencumbered by the physical dynamics of the robot. This can significantly simplify the design of a high-level controller that switches between multiple $f()$ [27]: the engineer can ensure smooth transitions by simply enforcing the desired properties in the phase oscillator space, and given a smooth mapping $g(\phi)$, those properties are retained in the generated reference trajectory. However, once the phase has converged to the limit cycle, this setup effectively acts as a clock, as the phase oscillator state cannot be perturbed off the limit cycle.

CPGs with feedback: $f() = f(\phi_j, x_j)$
Coupling the phase oscillator and physical states fully exploits the dynamics properties of the phase oscillator, but also makes it more difficult to design useful dynamics. Nonetheless, even relatively simple, local feedback has been shown to greatly improve performance [28], [29]. Several studies have relied on RL to learn $f()$ [16], [19]. Ryu and Kuo [22] also use this form, but re-interpret it as a state estimator, with the state feedback acting as the observer gain. We rely heavily on this interpretation.

Decentralized Oscillators: $C(i) = \{i\}$
This is a special case where $f() = f(\phi_i, x_i)$: each oscillator is only affected by feedback from itself and sensory information from its corresponding leg. Despite lacking direct coupling between oscillators, such systems can still synchronize due to the indirect feedback coupling through local physical interaction [23], [30]. Our work uses this setup, and in particular a form based on the so-called *Tegotae* feedback model proposed by Owaki, Kano, *et al.* [23]:

$$f() = -\sigma F_i^{\text{GRF}} \left(\cos\left(\phi_i\right) \right) \tag{2}$$

where σ is a feedback gain, and F_i^{GRF} is the ground reaction force felt at foot i.

This model captures the essence of a gait (frequency, swing/stance phases) without specifying any details. Indeed, it would be difficult to predict any specific gait when looking at Eq. (2). This type of decentralized oscillators has been shown to not only converge to gaits, but also switch gaits depending on forward velocity [23], [31].

We use a variation of this oscillator with the interpretation proposed by Ryu and Kuo [22], that each phase is a latent

observation of whether the corresponding leg should be in stance or swing, and σ in Eq. (2) is an observer gain.

III. IMPLEMENTATION DETAILS

We train policies and collect simulation data in IsaacGym, using a fork of `legged_gym` by Rudin, Hoeller, *et al.* [24], and proximal policy optimization. Our code is open-sourced [32] and can be referenced for implementation details, hyperparameters, and instructions to reproduce our results.

A. Robot

We use the MIT Mini Cheetah robot [33] in simulation and for hardware transfer. For all results presented in the paper, which rely on large-scale data collection for statistical analysis, we use simulation results. Preliminary tests on hardware can be seen in the supplementary video.

Observation space: The policy is given standard state observations composed of the base angular velocity, projected gravity vector, joint positions and velocities, the previous actions, and the desired base linear and angular velocity commands. When not ablated, the observations also include the phase oscillator observations $[\sin(\phi_i), \cos(\phi_i)], i \in \{FR, FL, HR, HL\}$. The critic is additionally given the current oscillator velocity as privileged information.

Action space: The policy outputs desired joint positions at 100 Hz, which are fed into a low-gain PD-controller with $K_p = 20$ and $K_d = 0.5$ running at 500 Hz.

Rewards: Following standard conventions, we use positive rewards for the command tracking error, orientation error, and a minimum base-height error, all passed through a squared exponential function. In addition, we regularize with negative rewards on the square of the torques, first and second-order action smoothness, and the hip abduction/adduction joints deviating from the resting position. Finally, body collisions with the ground terminate the episode and incur a flat penalty.

When not ablated, we also add the phase-based reward

$$r_{\text{gait}}(x) = -F_i^{\text{GRF}} \sin(\phi_i) \quad (3)$$

which penalizes foot contact when $\phi_i \in [0, \pi)$ and encourages contact during $\phi_i \in [\pi, 2\pi)$. Due to the dynamics of the phase oscillator, this simple reward encourages two effects: first, to learn a policy with roughly the oscillator nominal frequency ω, and second, to have consistent periods of contact with all feet. We will see that the policy also learns to actively use the F^{GRF} to 'guide' the oscillators into stable gaits if the coupling is included during training.

Ground Reaction Force Estimation: Each F_i^{GRF} value is normalized by robot mass and clamped between $[0, 1]$, such that σ values can be kept consistent for robots of different sizes. We also found this mitigated issues caused by the highly inaccurate contact force estimates in IsaacGym.

B. Decentralized phase oscillators

We add an offset term ξ to the original *Tegotae* feedback model from Eq. (2), so our oscillator dynamics become

$$\dot{\phi}_i = 2\pi \left(\omega - \sigma F_i^{\text{GRF}} (\cos(\phi_i) + \xi) \right) \quad (4)$$

Based on biological observations [34] and trial and error, the parameters ω, σ, and ξ are set to

$$[\omega, \sigma, \xi](v_x) = \begin{cases} [1, 4, 1] & \text{if } |v_x| \le 0.5 \\ [\min\{1.5 + |v_x|, 4\}, 1, 0] & \text{otherwise} \end{cases} \quad (5)$$

where v_x is the commanded forward velocity. Each phase ϕ_i is uniformly randomized at the start of each episode.

Standing still is a special case for which we want all oscillators to settle to a stable point in stance. Rather than activating special rewards to minimize joint velocities when the commanded velocity is low, we change the oscillator parameters to smoothly introduce a stable fixed point. This allows the fixed point to be closer to the middle of the stance range $[\pi, 2\pi)$ without using an excessively large σ value, which can destabilize the oscillator by chattering between stance and swing phases. This smoothing effect is illustrated in Fig. 2, with the blue line showing the actual phase velocity vs. phase function used during training for zero velocity commands.
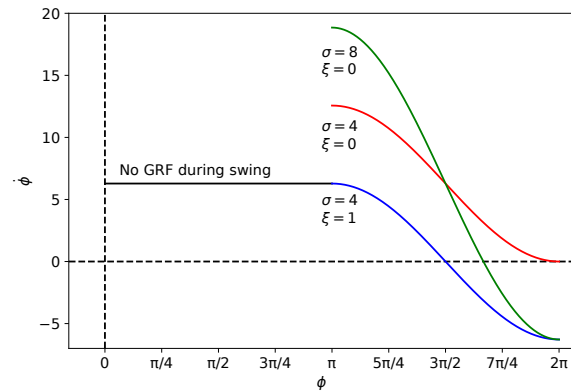

Fig. 2: To illustrate the oscillator fixed points (where $\dot{\phi} = 0$), we graph the oscillator dynamics with $F^{\text{GRF}} = 0.25$, where each leg supports a quarter of the body weight. For the curve with coupling $\sigma = 4$ and offset $\xi = 0$, the point at phase $\phi = 2\pi$ is only marginally stable, and would not settle in stance. When $\xi = 0$, the limit of the fixed point as σ approaches $+\infty$ is $3\pi/2$, but drastically increasing σ alone introduces discrete jumps in ϕ that are destabilizing. Setting $\xi = 1$ with $\sigma = 4$ caps $\dot{\phi}$ at the nominal $2\pi\omega$, and places the fixed point directly in the middle of the stance phase. This formulation helps to achieve standing without stepping in place.

IV. OBSERVATION, REWARD, COUPLING ABLATION

We perform an ablation of the phase-based observations, rewards, and coupling by cutting their respective signals during training, see Fig. 1. We will refer to each permutation as ORC(xxx), where each entry is a boolean indicating whether the corresponding signal is present or not during training. For example, ORC(110) indicates a policy that was trained with the phase observations and phase-based rewards, but with the coupling term $\sigma = 0$. Note that ORC(110) is essentially the setting used by Siekmann, Godse, *et al.* [13].

We skip the permutation ORC(001) as it is equivalent to ORC(000). For all other permutations, we train and statistically analyze 10 policies each to answer three questions:

IV-A) *How do the signals influence leg-load distribution?*
IV-B) *How do the signals influence gait emergence?*
IV-C) *How does each signal affect overall stability?*

A. Balanced Leg Use

Ideally, a policy should use all four legs in a consistent and balanced manner. ORC(x0x) policies either don't have oscillator observations or are not encouraged to use them in any particular way. ORC(01x) policies are non-Markov as it appears to receive inconsistent rewards for performing the same action given its observable state. ORC(11x) policies are trained with both oscillator observations and rewards to encourage swing and stance during specific ranges of the oscillators, and have enough information to match phases with rewards related to ground contact. We expect to see more consistent leg use with the policies trained using ORC(11x), and more variability in the other policies.

For each policy, 50 robots are initialized with random oscillator phases and rolled out in simulation with 1 m/s forward velocity command. We calculate the average F_i^{GRF} over the entire 10 second episode for each leg separately. Randomly initialized oscillator phases result in different behavior in the 50 agents for ORC(1xx) policies. Results from all 500 runs belonging to each ORC configuration are aggregated into the same dataset. Each violin plot in Fig. 3 shows the distribution of average F_i^{GRF} experienced by each leg. Since F_i^{GRF} is normalized by body weight, perfectly balanced leg use yields average $F_i^{\mathrm{GRF}} = 0.25$ for all legs.
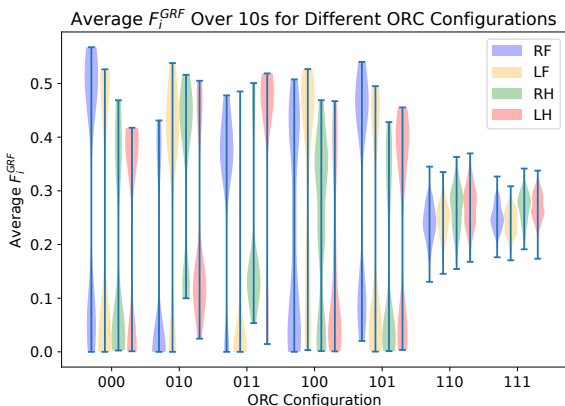

Fig. 3: Each ORC configuration has 500 agents (50 per re-trained policy), and F^{GRF} is averaged for each leg across the entire episode. ORC(11x) policies show much more consistent and balanced leg use compared to all other configurations, which tend to exhibit 2 or 3 legged gaits.

We can see that for all ORC configurations without observations and rewards working together, the distributions of F_i^{GRF} are very wide and have significant clusters around 0 for at least one leg out of four. This agrees with our observation that the policies trained with those ORC configurations often result in 2 or 3 legged gaits, where some feet are dragging or always kept in the air, and therefore experiencing little to no F_i^{GRF} over the roll-out. The pattern of two clusters around the extremities of the distributions for each leg arises from inconsistent policies after repeated training with the same setting. In some policies, the RF foot might always be in the air, while in others the LF foot might always be in the air.

The phase observations and rewards strongly encourage ORC(11x) policies to use all four legs cyclically without specifying the exact desired gait. Both configurations

yield distributions that are roughly centered around average $F_i^{\mathrm{GRF}} = 0.25$, showing that the leg use is well balanced over all trials and policies.

ORC(110) experiments have slightly larger range compared to ORC(111), which may be attributed to some randomly initialized asymmetric gaits requiring higher F_i^{GRF} on some legs compared to the others to track, since it cannot converge to a more symmetric gait without coupling.

B. Emergence of Gaits

To evaluate gait emergence, we focus only on ORC(110) and ORC(111) policies, since we see from Subsection IV-A that other permutations rarely yield well-defined gaits. Each experiment evaluates a single policy chosen at random, and includes 500 runs with randomized initial oscillator values, rolled out over 40 seconds with a 1 m/s forward velocity command. We calculate the RPD as described in Subsection II-A, and assess gait preference by evaluating the RPD distribution at the end of the roll-out. We verify that ORC(111) policies don't ignore the phase observation by deactivating the coupling at execution time. We further probe the role of the feedback coupling by activating it for an ORC(110) policy, which was trained with no coupling.

Both ORC(111), visualized in Fig. 4a), and ORC(110) policies match their RPD to the oscillator phases when the coupling is set to $\sigma = 0$, as expected.

When evaluating ORC(111) with coupling $\sigma = 1$ (same as during training), the gaits converge within 10 seconds to their final preferred state, and we observe mostly trotting and pronking gaits in tight clusters on Fig. 4b). Since the policy's actions influences the oscillators through ground contacts, it can learn to manipulate the randomly initialized oscillators and phase lock into desirable gaits faster.

When evaluating ORC(110) with $\sigma = 1$, the policy continues to track the oscillators, which are now also modulated by the feedback term that was not seen during training. The gaits that emerge from this experiment have regions of attraction dictated by the oscillator dynamics, which highlights the role of the oscillator in determining the preferred gait. However, as shown in Fig. 4c), the final gaits are clustered further away from the ideal phase differences of symmetric gaits compared to the ORC(111) policy shown in Fig. 4b). We also see in Fig. 5 and Fig. 6 that the ORC(110) (evaluated with coupling) takes nearly 40 seconds to settle into its preferred gait, whereas the ORC(111) policy settles in roughly 10 seconds.

Fig. 7 shows a pace to trot transition of one of the runs from Fig. 5 by plotting all F_i^{GRF} from 0-10 seconds with respect to the oscillator phases of the RF reference leg. The thick blue lines show F_i^{GRF} during the first gait cycle and the thick orange lines show F_i^{GRF} during the last gait cycle. As more gait cycles occur, the phase differences between the leg oscillators change, indicated by the red and green dots showing the RF phase value at the time-step when the corresponding leg's oscillator crosses 0 and π respectively. Those dots do not exactly overlap for the RF leg because of discrete time-step errors. The F_i^{GRF} of each leg follows its

Relative Phase Differences (RPD) at commanded velocity $v_x = 1$ [m/s]

ORC(111), $\sigma = 0$

ORC(111), $\sigma = 1$
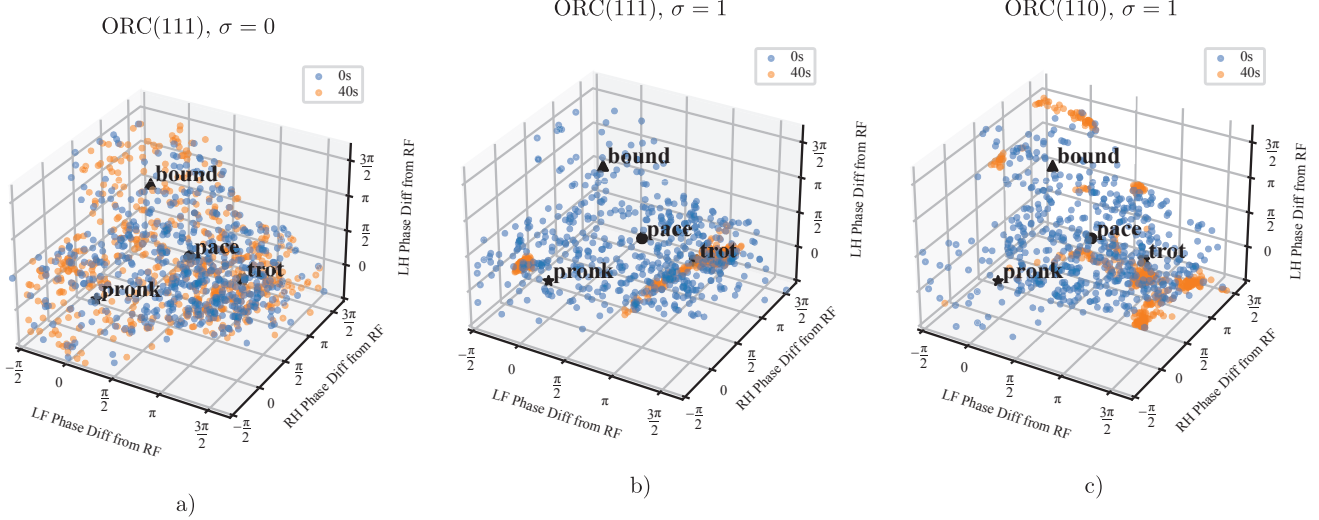
ORC(110), $\sigma = 1$

a)   b)   c)

Fig. 4: Initial and final relative phase difference RPD points are shown for 500 randomly initialized runs in each experiment. ORC(111) evaluated with coupling $\sigma = 0$ tracks the initial phases and cannot converge to any specific gait. ORC(111) evaluated with $\sigma = 1$ exhibits strong convergence to trot and pronk, while ORC(110) evaluated with $\sigma = 1$ exhibits some convergence around trot, but is more spread out compared to the final ORC(111) RPD.
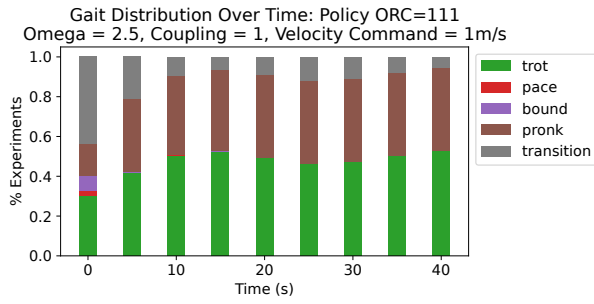

Fig. 5: The distribution of gaits for 500 runs of ORC(111) with $\sigma = 1$ settles into both trot and bound quickly within 10 seconds.
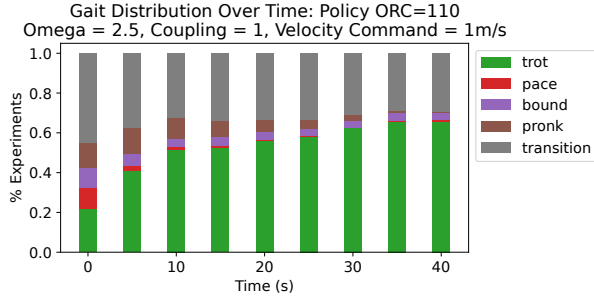

Fig. 6: The distribution of gaits for 500 runs of ORC(110) with coupling $\sigma = 1$ settles into trot slowly, with more environments transitioning at the beginning but others continuing to slowly converge toward trot as runs with RPD initialized further away become more trot-like over time due to the oscillator dynamics.

own phases, and over time settles into a trotting gait, with diagonal legs being in phase.

C. Disturbance Rejection

To compare the overall effect on stability, we test how well each policy can reject a planar velocity impulse applied to the body while being commanded to run at 3 m/s, with
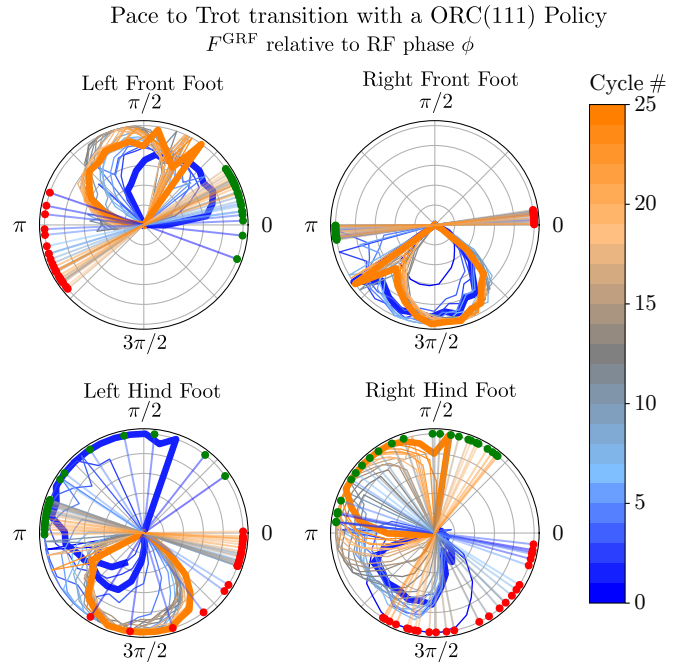

Fig. 7: The ground reaction force F_i^{GRF} is plotted for each foot relative to ϕ_{RF}, with initial gait cycles in blue and progressing through time to orange. The phase 0 and π crossings of each leg are shown with red and green dots, respectively. The bold cycles are the initial and final cycles, which show this run starting in pace with lateral feet in phase and ending in trot with diagonal feet in phase. All swing and stance behavior obeys each leg's respective oscillator phase well, with non-zero F^{GRF} falling between the green dot π crossings and red dot 0 crossings in every gait cycle.

oscillators being initialized at random and allowed a 5 second settling period. To ensure that we apply perturbations at all phases of the gait, we run 1800 trials per policy, and stagger the perturbations: every 0.01 seconds, a ball of perturbations spaced at 10 degree intervals is applied to a different set of 36 robots, over a period of 0.5 seconds. As all policies exhibit a frequency of roughly 4 Hz at this commanded

TABLE I: Failure rate after velocity impulse disturbance (All values in %)

ORC			000	100	101	010	110	011	111
Impulse magnitude [m/s]		1.5	**0.0 ± 0.0**	**0.0 ± 0.0**	0.1 ± 0.2	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**	**0.0 ± 0.0**
		2.0	1.7 ± 2.5	4.1 ± 5.2	0.9 ± 1.2	**0.0 ± 0.0**	0.0 ± 0.1	**0.0 ± 0.0**	0.0 ± 0.1
		2.5	11.0 ± 13.0	10.6 ± 9.1	7.2 ± 6.3	0.7 ± 0.7	0.7 ± 1.0	**0.3 ± 0.4**	0.4 ± 0.4
		3.0	16.3 ± 12.4	17.5 ± 8.9	16.0 ± 10.2	3.2 ± 3.0	4.1 ± 3.6	3.2 ± 3.1	**2.2 ± 1.7**
		3.5	25.0 ± 13.5	29.7 ± 12.7	24.0 ± 17.1	10.6 ± 8.0	13.8 ± 7.8	13.1 ± 9.2	**7.4 ± 4.0**

speed[1], staggering the perturbations ensures that we apply perturbations at different phases of the gait, with different numbers of feet on the ground. We then calculate the mean and standard deviation of the failure rate across all ten policies learned for each permutation of ORC(xxx).

The failure rate means and standard deviations, reported in Table I, show that policies trained under ORC(111) consistently have a lower failure rate compared to all other permutations. Surprisingly, toggling on phase-based rewards has a stronger impact than the phase-based observations, even for cases such as ORC(010) and ORC(011) where the reward is not Markov. We conjecture that, when the reward is not Markov, it should be viewed as a stochastic reward that still discourages chattering contacts or dragging feet, despite not signaling any specific gait schedule.

V. DISCUSSION

We presented an augmentation of the quadruped robot state space using one decentralized phase oscillator per leg, with a simple feedback coupling to the ground reaction force of the corresponding leg, which can be interpreted as an observer gain. Through a systematic ablation study, we investigated the importance of each phase-related signal: observations of the oscillator phase, phase-based rewards to encourage distinct swing and stance phases, and feedback coupling.

Overall, ORC(111) policies trained with all three signals demonstrated the fastest convergence to well-defined gaits, and were consistently the most robust to large impulse perturbations. We did not find significant differences in local stability[2] between the policies, which matches our experience in hardware that local stability is not a useful proxy for legged system 'stability'. Nonetheless, preliminary sim-to-real trials (see supplementary video) show significantly better transfer with the observation signal. Bellegarda and Ijspeert [19] also reported more reliable sim-to-real transfer when learning a CPG with feedback, although in their case the phase is directly mapped to desired kinematics with a pre-designed mapping.

ORC(110) policies trained with phase observations and rewards but no coupling showed only slightly worse performance to those with coupling, while tracking the gait defined by whichever phase-difference the oscillators are initialized in, similar to the results of Siekmann, Godse, *et al.* [13]. However, although activating the coupling during evaluation does cause these policies to converge toward symmetric gaits, the convergence time is nearly four times slower. This observation suggests that the dual roles of control and estimation are not fully separated between the oscillators and the policy, as it is in the linearized model presented by Ryu and Kuo [22]: the policy is affected by the oscillator state, but can also learn to actively drive it towards a more stable gait if trained with the coupling active. This also matches the observation of Ijspeert and Daley [20] that CPGs may act as both an observer and a pattern generator.

Surprisingly, we found that the reward signal has a stronger effect on stability than the phase observations, despite being non-Markov in some ablations. Nonetheless, only when both observation and reward signals were present during training, did policies consistently train to exhibit gaits with balanced load distribution among the legs.

Anecdotally, before we introduced the offset term in equation (4), policies did not settle into standing as well, but did appear to favor gaits other than pronking more compared to the results presented. We conjecture that frequent standing causes the oscillators to all sync to stance, and thus biases training data towards pronking gaits. We also observed different gaits to emerge more frequently at different velocities, or with different morphologies. Although quantifying the exact convergence and transition patterns is out of the scope of this paper, future work studying the effects of changing oscillator parameters along with physical parameters such as center of mass location and leg length on gait emergence for a quadruped robot could yield connections to gait patterns observed in nature for animals of different sizes [34].

Another avenue we find intriguing is the role oscillators may play in a hierarchical RL setting. Higher levels of hierarchy typically reason about task-level objectives in both a lower-dimensional space and at a slower timescale; the phase oscillators could be interpreted as a latent state with cyclic dynamics [35]. A latent state space with cyclic dynamics could serve for temporal abstraction, multi-joint coordination and amortized control for cyclic behavior [36], a direction we find very promising.

---

[1]We verified this frequency via a fast Fourier transform on the joint positions for policies without phase observations.

[2]Analysis of Floquet multipliers and rate of entropy decay were evaluated.

REFERENCES

[1] D. F. Hoyt and C. R. Taylor, "Gait and the energetics of locomotion in horses," *Nature*, 1981.

[2] S. Wilshin, M. A. Reeve, G. C. Haynes, S. Revzen, D. E. Koditschek, and A. J. Spence, "Longitudinal quasi-static stability predicts changes in dog gait on rough terrain," *Journal of Experimental Biology*, 2017.

[3] M. Hildebrand, "Symmetrical gaits of dogs in relation to body build," *Journal of Morphology*, 1968.

[4] G. Bledt, M. J. Powell, B. Katz, J. Di Carlo, P. M. Wensing, and S. Kim, "Mit cheetah 3: Design and control of a robust, dynamic quadruped robot," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[5] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *The International Journal of Robotics Research*, 2016.

[6] 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.

[7] H. Chen, Z. Hong, S. Yang, P. M. Wensing, and W. Zhang, "Quadruped capturability and push recovery via a switched-systems characterization of dynamic balance," *IEEE Transactions on Robotics*, 2023.

[8] N. J. Kong, C. Li, G. Council, and A. M. Johnson, "Hybrid ilqr model predictive control for contact implicit stabilization on legged robots," *IEEE Transactions on Robotics*, 2023.

[9] 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.

[10] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath, and S. Levine, "Genloco: Generalized locomotion controllers for quadrupedal robots," in *Conference on Robot Learning (CoRL)*, 2023.

[11] Z. Xie, X. Da, M. Van de Panne, B. Babich, and A. Garg, "Dynamics randomization revisited: A case study for quadrupedal locomotion," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[12] Y. Jin, X. Liu, Y. Shao, H. Wang, and W. Yang, "High-speed quadrupedal locomotion by imitation-relaxation reinforcement learning," *Nature Machine Intelligence*, 2022.

[13] J. Siekmann, Y. Godse, A. Fern, and J. Hurst, "Sim-to-real learning of all common bipedal gaits via periodic reward composition," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2021.

[14] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Conference on Robot Learning (CoRL)*, 2023.

[15] Y. Shao, Y. Jin, X. Liu, W. He, H. Wang, and W. Yang, "Learning free gait transition for quadruped robots via phase-guided controller," *IEEE Robotics and Automation Letters*, 2021.

[16] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conference on Robot Learning (CoRL)*, 2022.

[17] A. J. Ijspeert, "Central pattern generators for locomotion control in animals and robots: A review," *Neural Networks*, 2008.

[18] J. Buchli, L. Righetti, and A. J. Ijspeert, "Engineering entrainment and adaptation in limit cycle systems: From biological inspiration to applications in robotics," *Biological Cybernetics*, 2006.

[19] G. Bellegarda and A. Ijspeert, "CPG-RL: Learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*, 2022.

[20] A. J. Ijspeert and M. A. Daley, "Integration of feedforward and feedback control in the neuromechanics of vertebrate locomotion: A review of experimental, simulation and robotic studies," *Journal of Experimental Biology*, 2023.

[21] A. Raffin, O. Sigaud, J. Kober, A. Albu-Schäffer, J. Silvério, and F. Stulp, "A simple open-loop baseline for reinforcement learning locomotion tasks," in *International Conference on Learning Representations*, PMLR, 2024.

[22] H. X. Ryu and A. D. Kuo, "An optimality principle for locomotor central pattern generators," *Scientific Reports*, 2021.

[23] D. Owaki, T. Kano, K. Nagasawa, A. Tero, and A. Ishiguro, "Simple robot suggests physical interlimb communication is essential for quadruped walking," *Journal of The Royal Society Interface*, vol. 10, no. 78, 2013.

[24] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Conference on Robot Learning (CoRL)*, PMLR, 2022.

[25] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, 2013.

[26] M. Ajallooeian, J. van den Kieboom, A. Mukovskiy, M. A. Giese, and A. J. Ijspeert, "A general family of morphed nonlinear phase oscillators with arbitrary limit cycle shape," *Physica D: Nonlinear Phenomena*, 2013.

[27] A. J. Ijspeert, A. Crespi, D. Ryczko, and J.-M. Cabelguen, "From swimming to walking with a salamander robot driven by a spinal cord model," *Science*, 2007.

[28] M. Ajallooeian, S. Gay, A. Tuleu, A. Spröwitz, and A. J. Ijspeert, "Modular control of limit cycle locomotion over unperceived rough terrain," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[29] S. Aoi, D. Katayama, S. Fujiki, N. Tomita, T. Funato, T. Yamashita, K. Senda, and K. Tsuchiya, "A stability-based mechanism for hysteresis in the walk–trot transition in quadruped locomotion," *Journal of The Royal Society Interface*, 2013.

[30] M. Schilling and H. Cruse, "Decentralized control of insect walking: A simple neural network explains a wide range of behavioral and neurophysiological results," *PLoS Computational Biology*, 2020.

[31] D. Owaki and A. Ishiguro, "A quadruped robot exhibiting spontaneous gait transitions from walking to trotting to galloping," *Scientific Reports*, 2017.

[32] MIT Biomimetics, *ORCAgym*, `https://github.com/mit-biomimetics/ORCAgym`, 2023.

[33] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *IEEE International Conference on Robotics and Automation (ICRA)*, 2019.

[34] N. C. Heglund, C. R. Taylor, and T. A. McMahon, "Scaling stride frequency and gait to animal size: Mice to horses," *Science*, 1974.

[35] S. Starke, I. Mason, and T. Komura, "Deepphase: Periodic autoencoders for learning motion phase manifolds," *ACM Transactions on Graphics (TOG)*, 2022.

[36] J. Merel, M. Botvinick, and G. Wayne, "Hierarchical motor control in mammals and machines," *Nature Communications*, 2019.