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

by

Jenny L. Zhang

S.B. Electrical Engineering and Computer Science, Massachusetts Institute of Technology (2023)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024



Authored by: Jenny L. Zhang
Department of Electrical Engineering and Computer Science
January 19, 2024

Certified by: Sangbae Kim
Professor of Mechanical Engineering, Thesis Supervisor

Accepted by: Katrina LaCurts
Chair, Master of Engineering Thesis Committee
Department of Electrical Engineering and Computer Science

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

by

Jenny L. Zhang

Submitted to the Department of Electrical Engineering and Computer Science
on January 19, 2024 in partial fulfillment of the requirements for the degree of

MASTER OF ENGINEERING IN ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

## 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 we interpret as an observer feedback gain. The oscillator itself is interpreted 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.

Thesis supervisor: Sangbae Kim
Title: Professor of Mechanical Engineering

# Acknowledgments

To my parents who have always made me feel empowered to chase my improbable dreams: From late-night conversations about aspirations to celebrating even the smallest victories, your love has been a constant source of strength. I am profoundly grateful for the values you've instilled in me and for creating a home filled with warmth, laughter, and the absolute best food. This journey would not have been possible without you, and will continue to be deeply rooted in the foundations of your unwavering support.

To my advisor Professor Sangbae Kim: Thank you for inspiring and guiding my journey in robotics even before I became your student, and for the opportunity to continue exploring and learning in your lab. I hope I will do you proud in the coming adventures.

To Steve: Thank you for being a wonderful mentor, colleague, and friend. You've helped me grow in too many ways to count, and discover parts of myself that I didn't know existed.

To Se Hwan and Adi: Thank you both for the tremendous help with hardware transfer for this project, and for entertaining my random existential conversations.

To all my lab-mates: Thank you for making every day and late night in lab so much fun (both types!). I'll miss the camaraderie, shared passions, and humor I found here in spades. Wishing you all stable feedback and useful gradients always!

To all of my teachers: Thanks for more than just imparting knowledge and making me the learner I am today – thank you for encouraging that introverted kid with the nerdy jokes. Extra credit to the few who were ready with the punchlines.

To my teammates from the Solar Electric Vehicle Team: I love you all to the sun and back.

To my friends, residents of the Sponge, and everyone who made MIT feel like home: I couldn't have truly found paradise without you.

To Quang: For always sharing battery I*V and *light*-hearted laughs with me.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

## 1.1 Motivation

Quadrupedal animals exhibit a variety of gaits, or pattern of footfalls, and the choice of gaits has been linked to energetics, speed of travel, 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, a single gait greatly simplifies the controller design; for conventional model-predictive control, pre-specifying the contact sequence [6], [7] allows a significantly simpler problem to be solved. 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 [8]–[10] or a reward [11]. A periodic clock observation is then necessary to maintain the Markov property of the reference. We postulate that fixing gaits is not required to yield consistent locomotion policies, and that optimal gaits can emerge from a policy rather than being pre-defined during training.

## 1.2 Related Work

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, Fern, *et al.* [12] 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 [13], [14]. 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 [15].

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) [16]. 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, typically desired kinematics. In essence, this is a generalization of time-indexed trajectories. Thus, this yields locomotion behavior that converge to a known limit cycle that was designed for. 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 [17]. Recent works have instead opted to learn the coupling, while fixing the mapping from the phase to robot states [15], [18]. This approach complements the phase-based low-level policies, and CPGs are often interpreted as feed-forward reference generators [19]. An alternative view, recently proposed by Ryu and Kuo [20], interprets the CPG as an observer, with reflex-like sensory feedback playing the role of the observer gain.

## 1.3 Contribution

We present an implementation of decentralized phase oscillators based on the work of Owaki, Kano, Nagasawa, *et al.* [21], and, based on the observer-interpretation of Ryu and Kuo [20], 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, Fern, *et al.* [12] to encourage the policy and phase oscillators to entrain.

The three key signals afforded by our architecture, represented in Fig. 1.1 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* 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.

We combine gait tracking and gait choice in a single policy by mapping the oscillator signals to swing and stance phases and updating oscillator phase values according to dynamics that are dependent on forces felt by each foot. Our approach using all three signals yields a deep reinforcement learning policy using minimal reward shaping that naturally transitions between balanced 4-legged gaits, comes to a standstill without special rewards that toggle on for low velocity commands, and successfully recovers from large impulse disturbances.

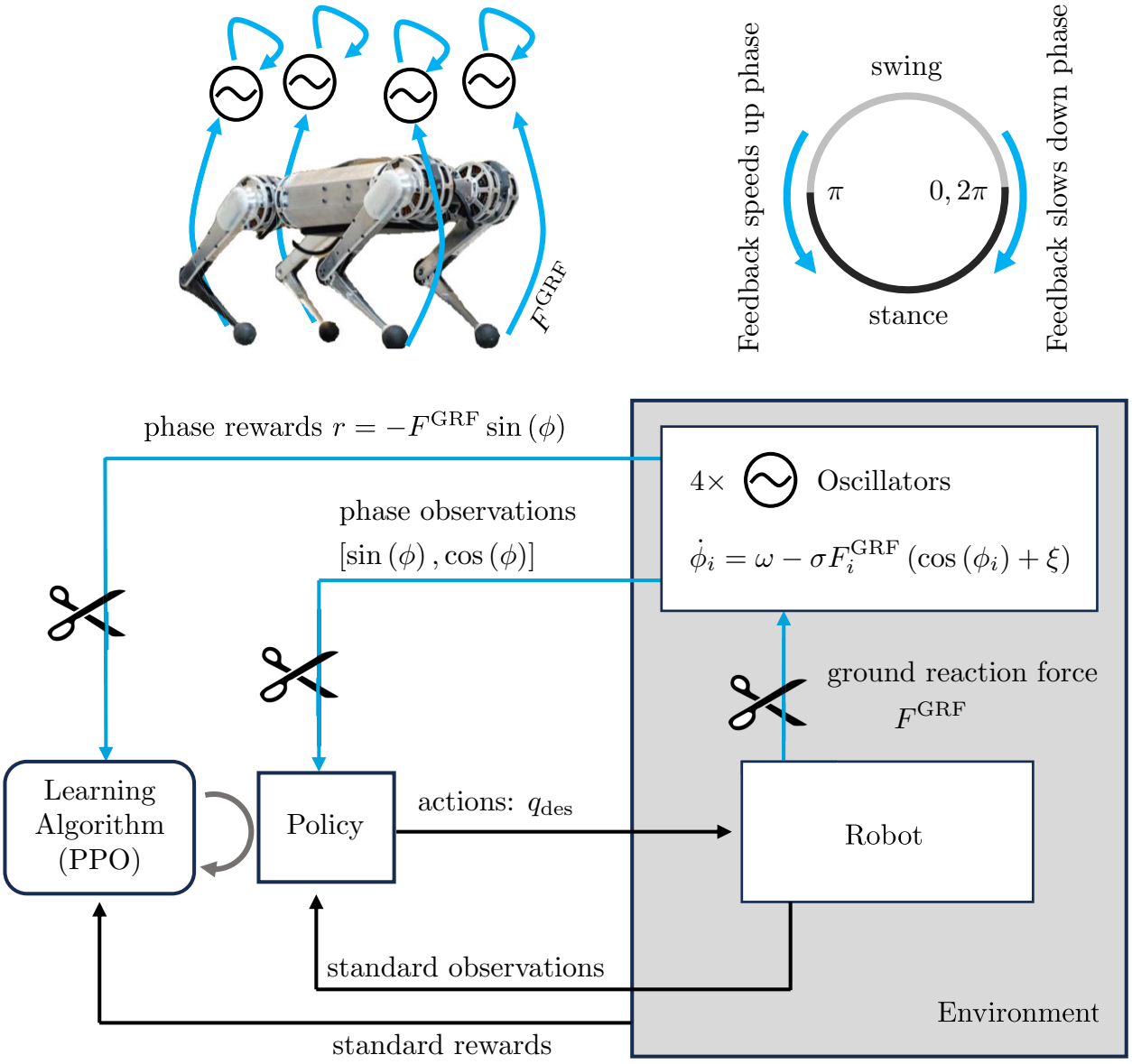


Figure 1.1: We augment the robot state with four decentralized phase oscillators, one per leg. Blue arrows in the diagram indicate the three oscillator-related signals: first, observations of the oscillator phase to the policy to make feed-forward Markov. Second, the phase-based reward encodes the general properties of gaits. Finally, the ground reaction force ($F^{\mathrm{GRF}}$) 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.

# Chapter 2

# Preliminaries

We briefly explain deep reinforcement learning in the context of quadrupedal robot locomotion. We also introduce gaits and pattern generation based on phase oscillators, as key background knowledge specific to our work.

## 2.1 Reinforcement Learning

We use the deep RL Proximal Policy Optimization (PPO) algorithm [22] to optimize two networks in the Asynchronous Advantage Actor Critic (A3C) architecture [23]. The actor network is the policy that is being trained, with the input as a set of state and environment observations, and the output typically as desired joint position setpoints for the legged robotics application. The policy's goal is to output the optimal action that leads to receiving the highest total infinite-horizon reward, with future rewards discounted by some factor. The critic network is used to estimate the value function of expected reward from being in a given observable state, which can contain more information than the actor can observe. The networks are updated by using actual rewards received by many asynchronous agents in parallel environments that take experimental actions informed by the actor in the IsaacGym physics simulator. By recording the states, actions taken, rewards received, and subsequent states in an on-policy fashion, the critic can be updated to minimize the difference between expected value and actual rewards. The actor is then updated, knowing the

"advantage", or how much more reward was actually received by taking certain actions compared to the expected value solely from being in the initial state. For the interested reader, we also recommend [24], [25].

## 2.2 Gaits

A gait is defined as a pattern of movement. 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 $2\pi$. For example, a trotting gait is defined by a diagonally opposed pair of feet contacting the ground at the same time, regardless of the kinematic trajectories the legs take during swing. 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 in Euclidean space. We visualize the ideal symmetric gaits **Trot** $(\pi, \pi, 0)$, **Pace** $(\pi, 0, \pi)$, **Bound** $(0, \pi, \pi)$, **Pronk** $(0, 0, 0)$ in Fig. 4.8.

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 the 3D phase difference space. If the distance to the closest ideal gait is above a threshold of 2, we classify the RPD as being in **Transition**.

## 2.3 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

$$\begin{aligned} \dot{\phi}_i &= \omega + f(\phi_j, x_j),\ j \in C(i) \\ u &= g(\phi_j, x_j) \\ \text{for } i &\in \{1, \ldots, n\},\ \text{and } C(i) \subset \{1, \ldots, n\} \end{aligned} \tag{2.1}$$

where $\omega$ 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.

### 2.3.1 Clocks: $f() = 0$

This degenerate choice for $f()$ reduces the oscillator to a clock with a constant growth rate $\omega$. Though sometimes called CPGs [8], we distinguish this setting as it makes no use of the state and dynamics of the oscillator: most of the burden is placed on designing the map $g()$. This is the minimal form needed to make a cyclic feed-forward pattern Markov. Because of the 1-dimensionality of the clock, a phase-only mapping $g(\phi)$ can index a pre-specified reference trajectory. Siekmann, Godse, Fern, *et al.* [12] 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.

### 2.3.2 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 [17], [26], unencumbered by the physical dynamics of the robot. This can significantly simplify design, especially if a high-level controller can switch 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 $g(\phi)$, retain those properties in the generated reference trajectory. On the other hand, once the phase

has converged to the limit cycle, this setup effectively acts as a clock, as the phase oscillator state cannot be affected by the physical states.

### 2.3.3 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 designing useful dynamics more difficult. Nonetheless, even relatively simple, local feedback has been shown to greatly improve performance [28], [29]. Several studies have relied on RL to learn $f()$ [15], [18]. Ryu and Kuo [20] 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.

### 2.3.4 Decentralized Oscillators: $C(i) = \{\text{FR}, \text{FL}, \text{HR}, \text{HL}\}$

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. Set $C(i)$ for each oscillator is a singleton set indicating direct coupling only to itself. As we are working with a quadruped system, we refer to the set of legs as Front Right (FR), Front Left (FL), Hind Right (HR), and Hind Left (HL). Despite lacking direct coupling between oscillators, such systems can still synchronize due to the indirect coupling through local physical feedback [21], [30]. Our work uses this setup, and in particular a form based on the so-called *Tegotae* feedback model proposed by Owaki, Kano, Nagasawa, *et al.* [21]:

$$f() = -\sigma F_i^{\text{GRF}} \left(\cos\left(\phi_i\right)\right) \tag{2.2}$$

where $\sigma$ is a feedback gain, and $F_i^{\text{GRF}}$ 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. (2.2). This type of decentralized oscillators has been shown to not only converge to gaits, but also switch gaits depending on forward velocity [21], [31].

We use a variation of this oscillator with the interpretation proposed by Ryu and Kuo [20], that each phase is a latent observation of whether the corresponding leg should be in stance or swing, and $\sigma$ in Eq. (2.2) is an observer gain.

# Chapter 3

# Implementation Details

We train policies and collect simulation data in IsaacGym, using a fork of `legged_gym` by Rudin, Hoeller, Reist, *et al.* [24], and proximal policy optimization. Our code can be accessed at **ORCAgym**[1] for implementation details, hyperparameters, and instructions to reproduce our results.

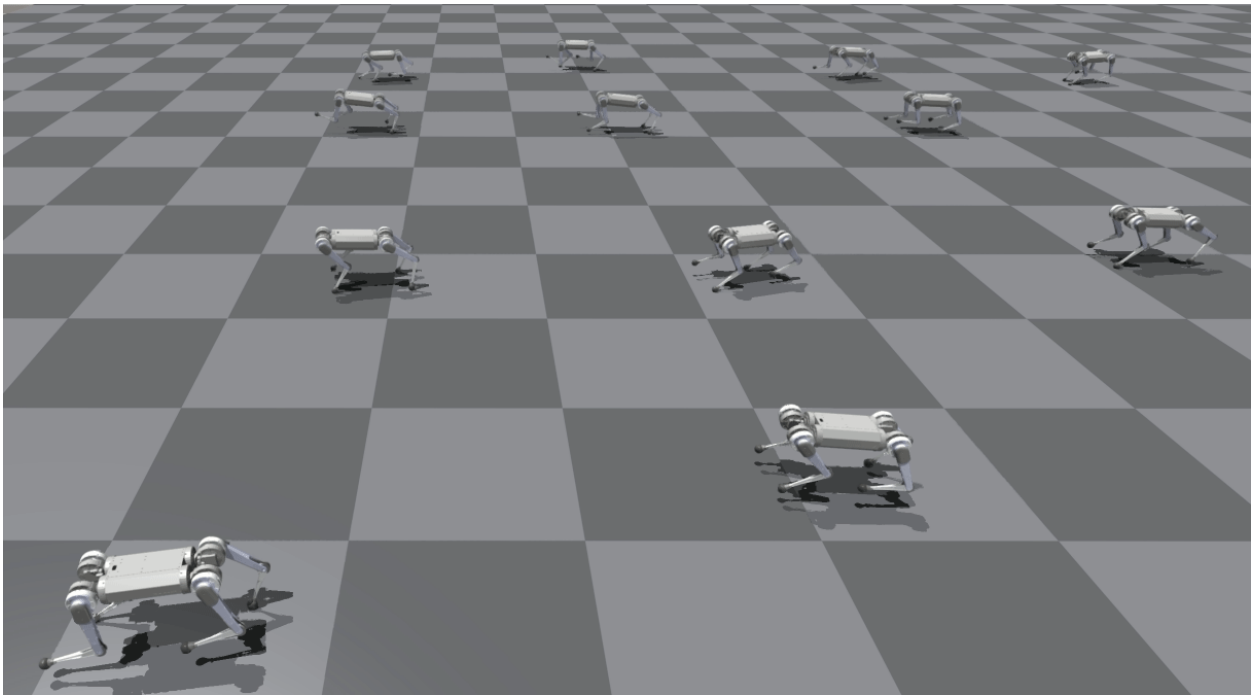

Figure 3.1: IsaacGym training environment with MIT Mini Cheetah models.

[1] https://github.com/mit-biomimetics/ORCAgym

## 3.1 Robot

We use the MIT Mini Cheetah robot [32] in simulation and for hardware transfer. The Mini Cheetah has 12 degrees of freedom, with 3 joints (abduction/adduction, hip, and knee) on each of the 4 legs (Front Right, Front Left, Hind Right, Hind Left). For all results presented in the paper, which rely on large-scale data collection for statistical analysis, we use simulation results. Simulation transfer to Robot Software, an in-house simulator, is done to check the policies before attempting hardware transfer. Preliminary tests on hardware can be seen in the **supplementary video**[2].

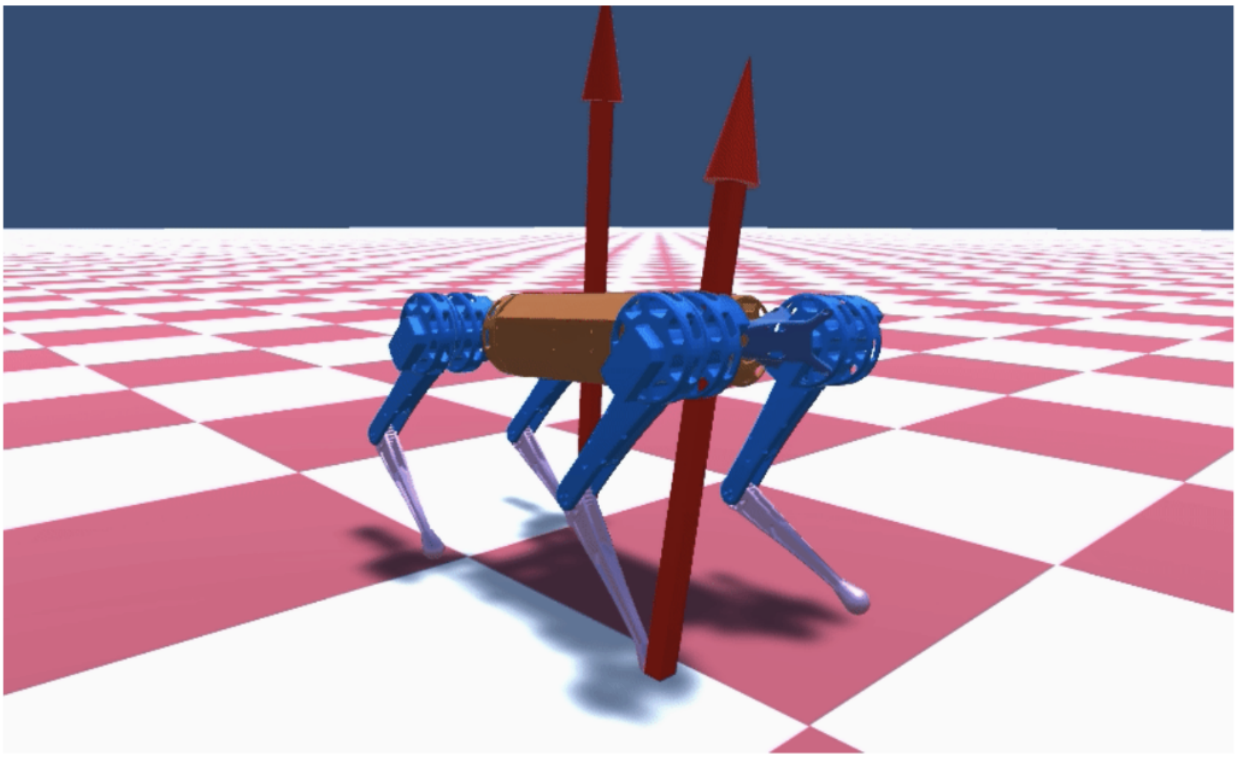

Figure 3.2: Robot Software simulation environment with MIT Mini Cheetah while testing exported deep reinforcement policy controller.

[2]https://youtu.be/RBGQa$_t$5$JV$8

## 3.2 Training

We train each policy on 4096 environments simultaneously for 2000 iterations using the PPO algorithm. Both actor and critic networks have 3 hidden layers of sizes [256, 256, 128], using the ELU activation function. Trained policies are exported to the ONNX format to load and evaluate in real time on hardware.

### 3.2.1 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 $[\sin(\phi_i), \cos(\phi_i)], i \in \{\text{FR}, \text{FL}, \text{HR}, \text{HL}\}$. We provide the sine and cosine of each oscillator value to avoid discontinuous jumps from wrapping the phase around from $2\pi$ to $0$ at each cycle. The critic is additionally given the base height and linear velocity as well as the current oscillator velocity as privileged information.

The commanded velocity for each individual environment is resampled every 3 seconds. A forward velocity is uniformly randomly chosen from an array of values [-3, -1, 0, 1, 3], then a value sampled from a normal distribution centered at 0 with standard deviation 1 is added to the nominal velocity.

### 3.2.2 Action Space

The policy outputs desired joint positions at 100 Hz ($\delta t_{\text{control}} = 0.01\text{s}$), which are fed into a low-gain PD-controller with $K_p = 20$ and $K_d = 0.5$ running at 500 Hz.

Table 3.1: Actor and Critic Observations

| Observation | Dimension | Symbol |
|---|---|---|
| **Actor Observations** | | |
| joint positions | 12 | $\mathbf{x}$ |
| joint velocities | 12 | $\dot{\mathbf{x}}$ |
| body orientation | 3 | $\vec{g}$ |
| body angular velocity | 3 | $\omega_r, \omega_p, \omega_{yaw}$ |
| body linear velocity commands | 2 | $v_x^{des}, v_y^{des}$ |
| body angular velocity command | 1 | $\omega_{yaw}^{des}$ |
| phase oscillators | 8 | $\sin(\phi_i), \cos(\phi_i)$ |
| previous desired joint positions | 12 | $\mathbf{x}^{des}(t=-1)$ |
| **Additional Critic Observations** | | |
| body height | 1 | $h$ |
| body linear velocity | 3 | $v_x, v_y, v_z$ |
| phase oscillator velocities | 4 | $\dot{\phi}_i$ |

### 3.2.3 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

$$r_{\text{gait}}(x) = -F_i^{\text{GRF}} \sin(\phi_i) \tag{3.1}$$

which penalizes foot contact when $\phi_i \in [0, \pi)$ and encourages contact during $\phi_i \in [\pi, 2\pi)$. Due

Table 3.2: Reward Weights and Functions

| **Reward** | **Weight** | **Function** |
|---|---|---|
| phase matched swing + stance | 5 | $-F_i^{\text{GRF}} \sin(\phi_i)$ |
| linear velocity tracking | 4 | $e^{-\left(\left(\frac{v_x - v_x^{des}}{1+\lvert v_x^{des}\rvert}\right)^2 + \left(\frac{v_y - v_y^{des}}{1+\lvert v_y^{des}\rvert}\right)^2\right)/0.25}$ |
| angular velocity tracking | 2 | $e^{-\left(\frac{\omega_{yaw} - \omega_{yaw}^{des}}{5}\right)^2/0.25}$ |
| body orientation | 1 | $e^{-\vec{g}_x^2/0.25} + e^{-\vec{g}_y^2/0.25}$ |
| minimum base height | 1.5 | $e^{-\left(\frac{h - h_{des}}{0.3}\right)^2/0.25}$ |
| hip ab-ad joint regularization | 0.0625 | $-\sum_{j \in \text{ab-ad joints}} \left(\frac{x_j}{0.8}\right)^2$ |
| torques minimization | 5e-7 | $-\sum_{j=1}^{12} \tau_j^2$ |
| first order smoothness | 0.01 | $-\sum_{j=1}^{12} \frac{\left(x_j^{des}(t=-1) - x_j^{des}(t=-2)\right)^2}{(\delta t_{\text{control}})^2}$ |
| second order smoothness | 0.001 | $-\sum_{j=1}^{12} \frac{\left(x_j^{des}(t=-1) - 2*x_j^{des}(t=-2) + x_j^{des}(t=-3)\right)^2}{(\delta t_{\text{control}})^2}$ |

to the dynamics of the phase oscillator, this simple reward encourages two effects: first, to learn a policy with roughly the oscillator nominal frequency $\omega$, and second, to have consistent periods of contact with all feet. We will see that the policy also learns to actively use the $F^{\text{GRF}}$ to 'guide' the oscillators into stable gaits if the coupling is included during training.

**Ground Reaction Force Estimation:** Each $F_i^{\text{GRF}}$ value is normalized by robot mass and clamped between $[0, 1]$, such that $\sigma$ 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. This especially helped us when transferring the policy to hardware, as it caps the largest possible magnitude of the the rate of change of the oscillators, limiting the effect of erroneous ground reaction force estimates while running online.

## 3.3 Decentralized phase oscillators

We add an offset term $\xi$ to the original *Tegotae* feedback model from Eq. (2.2), so our oscillator dynamics become

$$\dot{\phi}_i = 2\pi \left(\omega - \sigma F_i^{\text{GRF}} \left(\cos\left(\phi_i\right) + \xi\right)\right) \tag{3.2}$$

Based on biological observations [33] and trial and error, the parameters $\omega$, $\sigma$, and $\xi$ are set to

$$[\omega,\, \sigma,\, \xi](v_x) = \begin{cases} [1,\, 4,\, 1] & \text{if } |v_x| \leq 0.5 \\ [\min\{1.5 + |v_x|, 4\},\, 1,\, 0] & \text{otherwise} \end{cases} \tag{3.3}$$

where $v_x$ is the commanded forward velocity. An $\omega$ value of 1 should yield an average gait frequency of 1 Hz, where each foot strikes the ground about once a second. Increasing velocity corresponds to increasing gait frequency until it gets capped at 4 Hz. Each phase $\phi_i$ is uniformly randomized at the start of each episode.

During swing phase when the foot is in the air, it feels no ground reaction force and the corresponding oscillator advances at the nominal rate. If the oscillator is at the end of swing phase or beginning of stance phase and feels a force on the corresponding foot, that means it is needed to support the robot's weight and the positive coupling term pushes the phase toward the middle of stance phase faster. If the oscillator is at the end of stance phase or beginning of swing phase and still feels a force on the corresponding foot, it is still important in supporting the robot and the negative coupling term slows down the phase rate of change to keep it in stance for longer.

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 $\sigma$ value, which can destabilize the oscillator by chattering between stance and swing phases. This smoothing effect is illustrated in Fig. 3.4, with the blue line showing the actual

phase velocity vs. phase function used during training for zero velocity commands.

Fig. 3.3 shows an agent transitioning from a trot to a standstill, with all four legs on the ground consistently supporting the weight of the robot.

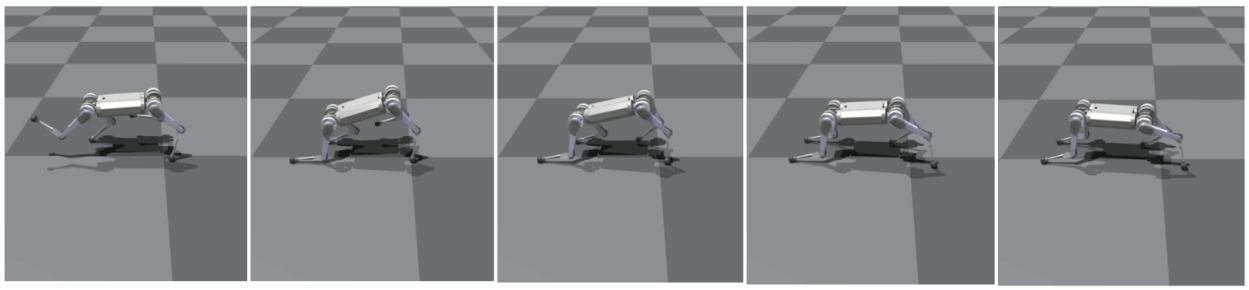

Figure 3.3: Frames showing transition from a trot moving forward to a standstill when commands and $\omega$, $\sigma$, $\xi$ values switch, and the policy outputs actions to get the agent to come to a quick stop.

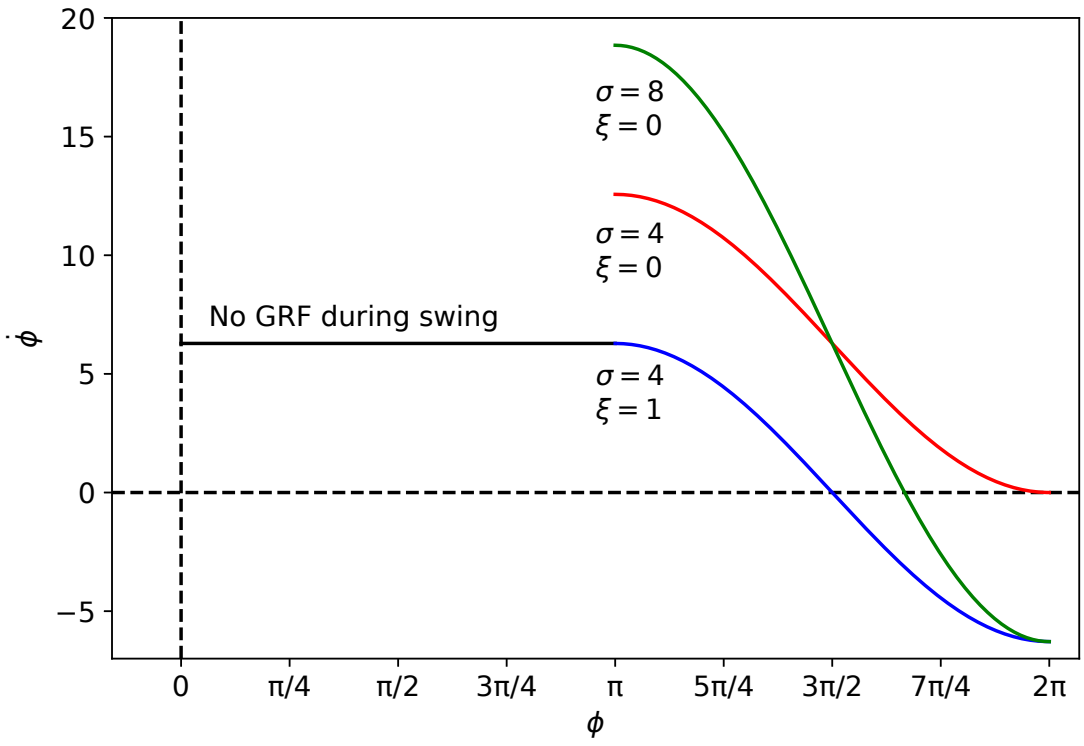


Figure 3.4: For a well-balanced stand, each leg should be supporting about 0.25 of the robot mass, so we set $F^{\text{GRF}} = 0.25$ when graphing the oscillator dynamics in stance phase when $\phi \in [\pi, 2\pi)$. The stable fixed point locations are at the intersections of the colored lines with the x-axis. For the curve with $\sigma = 4$ and $\xi = 0$, the point at $\phi = 2\pi$ is only marginally stable so it would not settle in stance. The limit of the fixed point as $\sigma$ approaches $+\infty$ when $\xi = 0$ is $3\pi/2$, but drastically increasing $\sigma$ alone introduces huge discrete jumps in $\phi$ 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. We achieve stable standing performance with this formulation.

# Chapter 4

# Observation, Reward, Coupling Ablation

We perform an ablation of the phase-based observations, rewards, and coupling by cutting their respective signals during training. 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, Fern, *et al.* [12].

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:

4.1) *Which signals are needed to consistently learn gaits that distribute the load equally across all legs?*

4.2) *How do the signals influence gait emergence?*

4.3) *How does each signal affect overall stability?*

## 4.1 Balanced Leg Use

Ideally, a policy uses all four legs in a consistent and balanced way to propel itself without falling. 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 trained with both oscillator observations and rewards to encourage swing and stance during different ranges of the oscillators have enough information to match phases with 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 $F_i^{\text{GRF}}$ 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. 4.1 shows the distribution of average $F_i^{\text{GRF}}$ experienced by each leg. Since $F_i^{\text{GRF}}$ is normalized by body weight, perfectly balanced leg use yields average $F_i^{\text{GRF}} = 0.25$ for all legs.

We can see that for all ORC configurations without observations and rewards working together, the distributions of $F_i^{\text{GRF}}$ are very wide and have significant clusters around 0 for at least one leg out of four. This corresponds to visual confirmation 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 $F_i^{\text{GRF}}$ 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 $F_i^{\text{GRF}} = 0.25$, 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 $F_i^{\text{GRF}}$ on some legs compared to the others to track, since it cannot converge to a more symmetric gait without coupling.

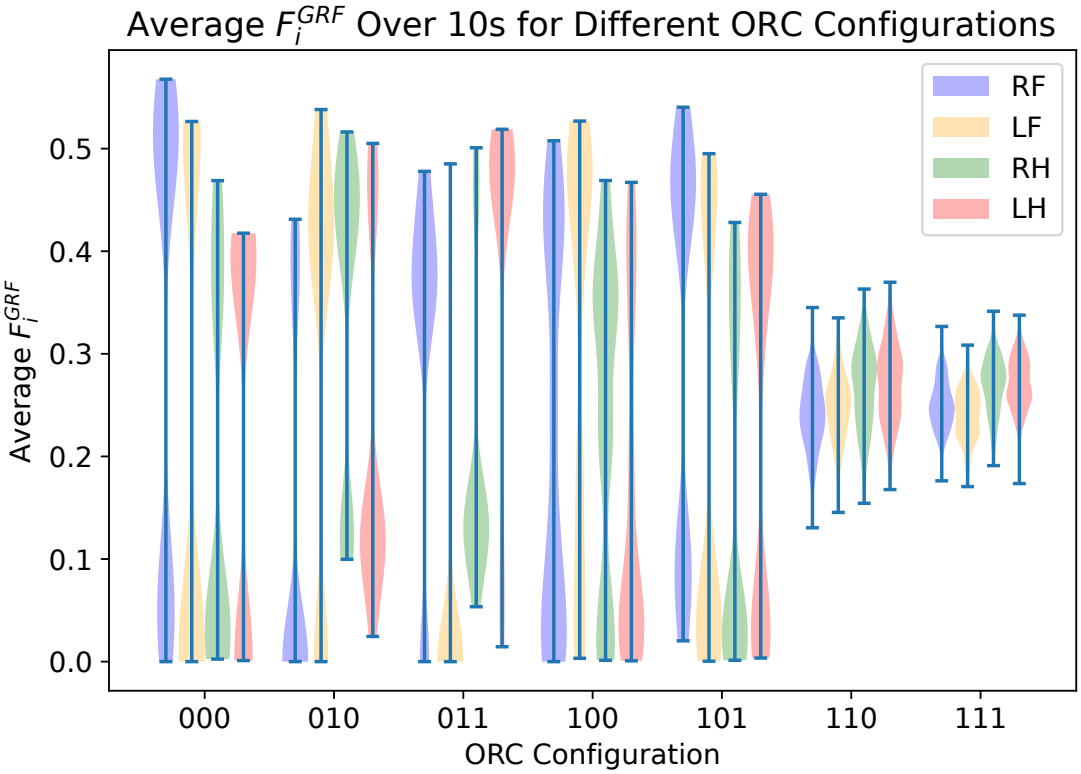


Figure 4.1: Each ORC configuration has 500 agents (50 per re-trained policy), and $F^{\text{GRF}}$ is averaged for each leg across the entire episode. ORC(11x) policies show much more consistent and balanced leg use compared to all other configurations, which tend to exhibit 2 or 3 legged gaits.

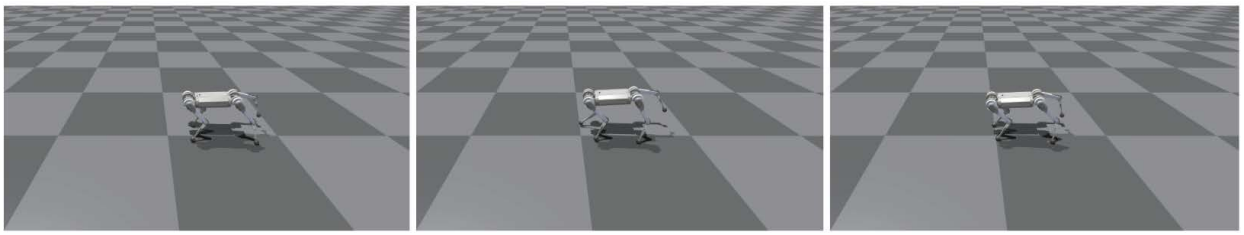

Figure 4.2: Frames from rolling out a ORC(000) policy, showing a 3 legged gait with the hind right leg lifted in the air.

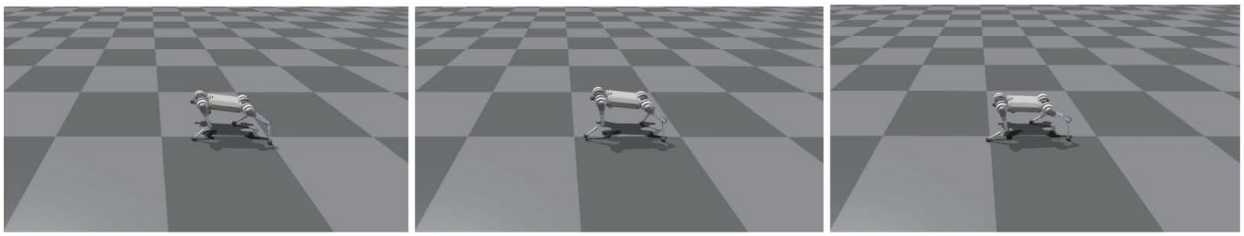

Figure 4.3: Frames from rolling out a ORC(010) policy, showing a 3 legged gait with the front right leg lifted in the air.

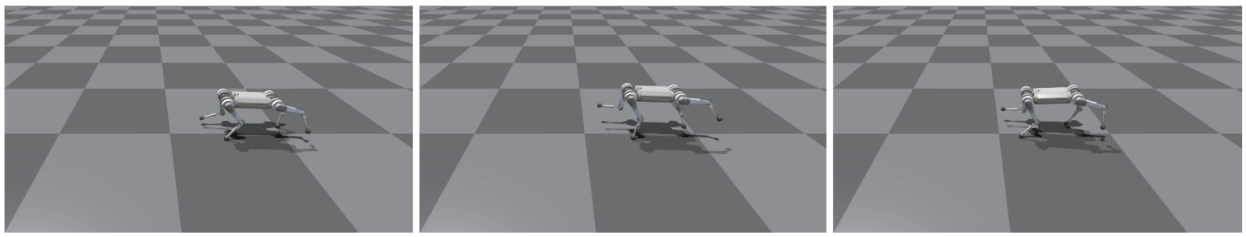

Figure 4.4: Frames from rolling out a ORC(100) policy, showing a 2 legged gait with front right and left hind legs lifted in the air.

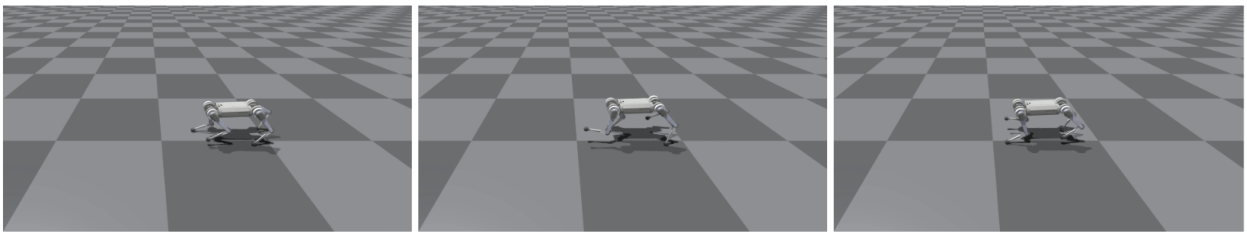

Figure 4.5: Frames from rolling out a ORC(110) policy, showing a regular trotting gait.

## 4.2 Emergence of Gaits

To evaluate gait emergence, we focus only on ORC(110) and ORC(111) policies, since we see from Section 4.1 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 Section 2.2, 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. 4.8a), and ORC(110) policies match their RPD to the oscillator phases when the coupling is set to $\sigma = 0$, as expected. Phase differences are constrained to remain constant during evaluation, so gaits cannot converge. We also test a few commonly known symmetric gaits to verify the policy's ability to track these.

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

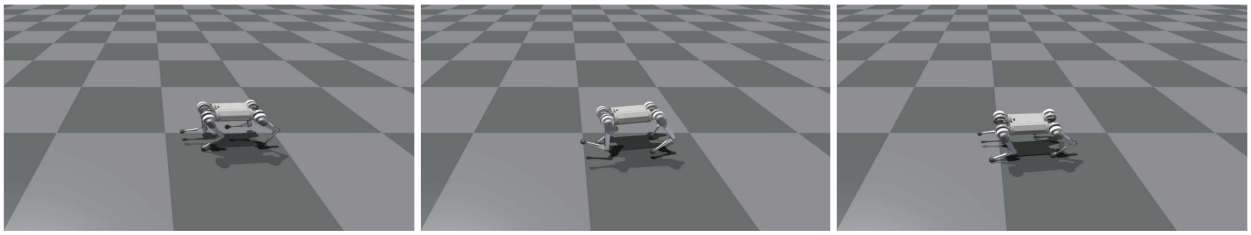

Figure 4.6: Frames from rolling out a ORC(111) policy initialized to pacing with $\sigma = 0$.

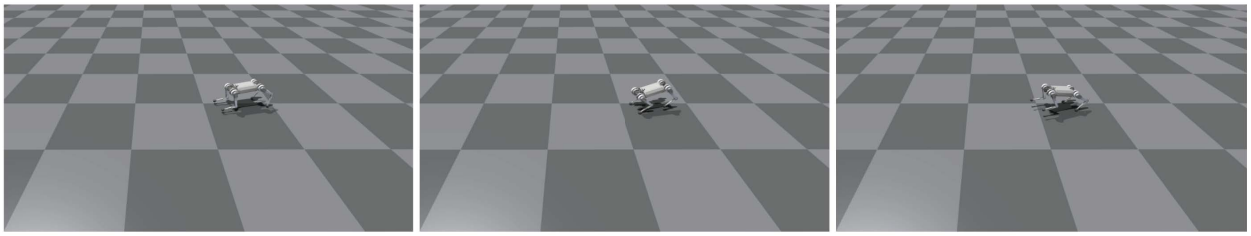

Figure 4.7: Frames from rolling out a ORC(111) policy initialized to bounding with $\sigma = 0$.

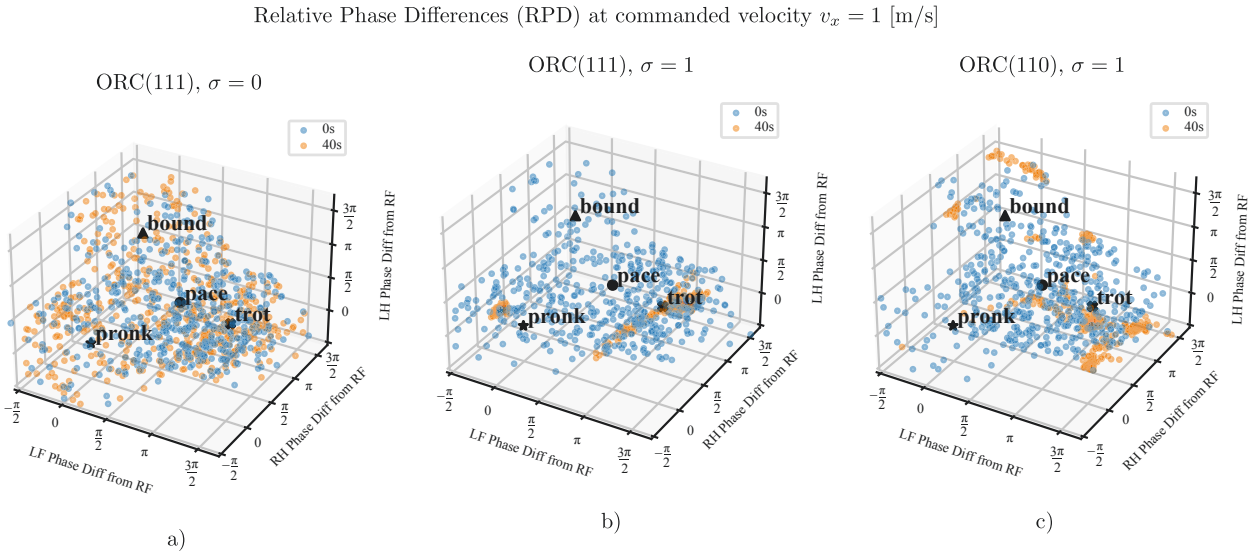


Figure 4.8: Initial and final RPD points are shown for 500 randomly initialized runs in each experiment. ORC(111) evaluated with $\sigma = 0$ tracks the initial phases and cannot converge to any specific gait. ORC(111) evaluated with $\sigma = 1$ exhibits strong convergence to trot and pronk, while ORC(110) evaluated with $\sigma = 1$ exhibits some convergence around trot, but is more spread out compared to the final ORC(111) RPD.

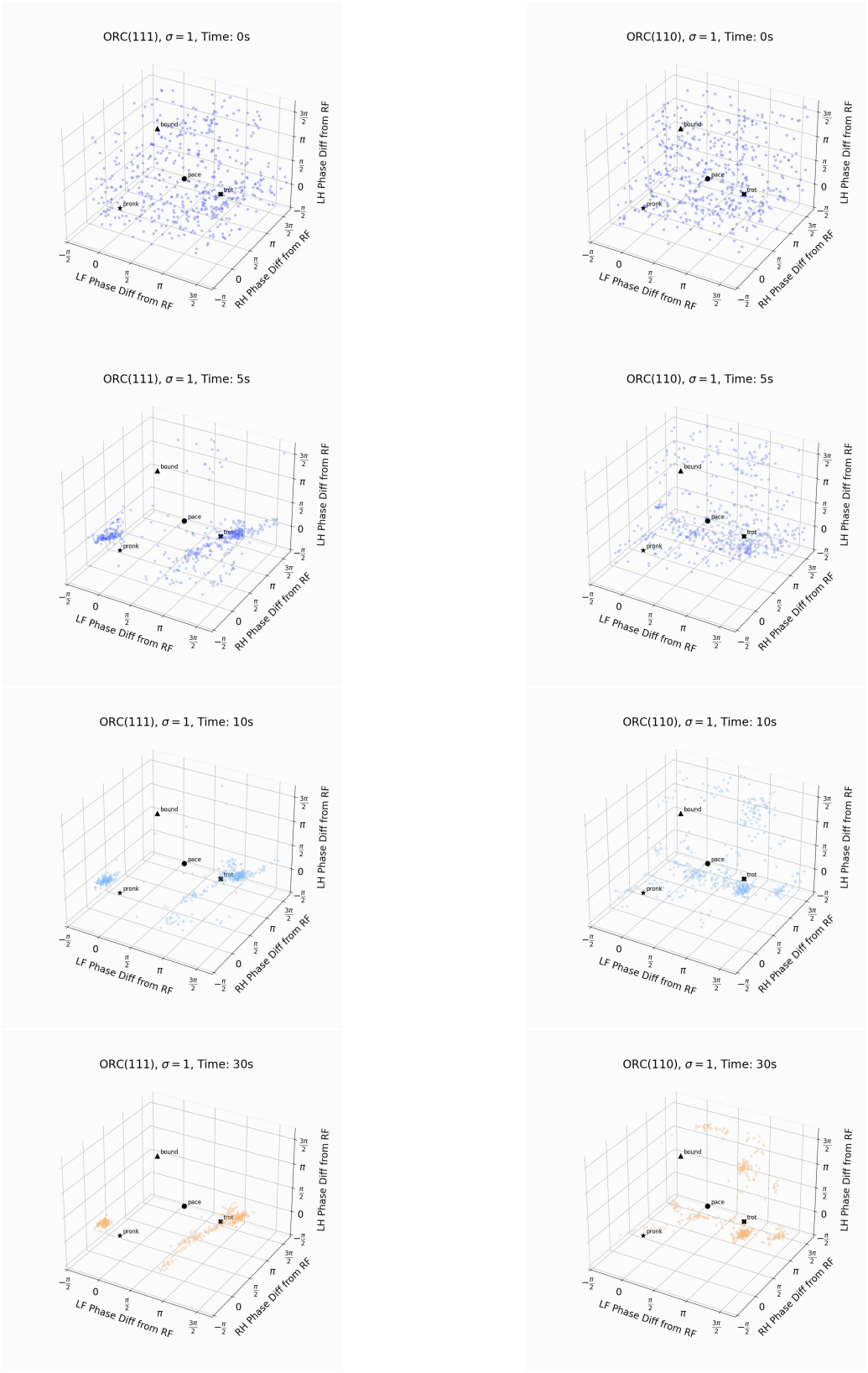


Figure 4.9: Relative Phase Differences at initialization, 5 seconds, 10 seconds, and 30 seconds for ORC(111) and ORC(110) evaluated with $\sigma = 1$, showing more detailed RPD transitions over time.

in tight clusters on Fig. 4.8b). Since the behavior of the policy influences the oscillators through ground contacts, it can learn to manipulate the randomly initialized oscillators and phase lock into desirable gaits faster.

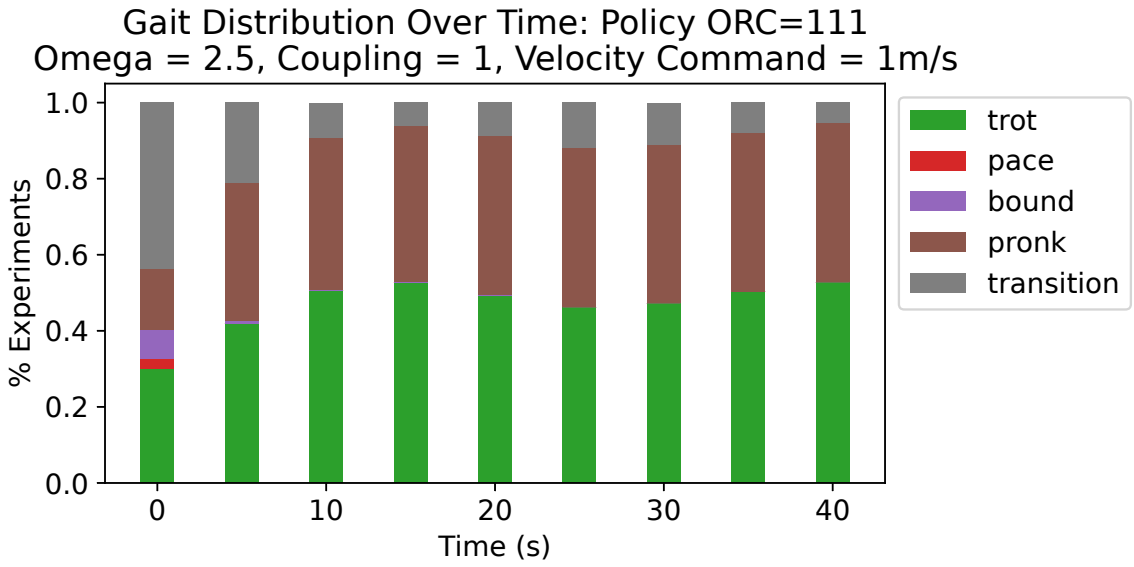


Figure 4.10: The distribution of gaits for 500 runs of ORC(111) with $\sigma = 1$ settles into both trot and bound quickly within 10 seconds.

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. 4.8c), the final gaits are clustered further away from the ideal phase differences of symmetric gaits compared to the ORC(111) policy shown in Fig. 4.8b). We also see in Fig. 4.11 and Fig. 4.10 that the ORC(110) (evaluated with coupling) takes nearly the entire 40 seconds to settle into gait its preferred gait, whereas the ORC(111) policy settles in roughly 10 seconds.

Fig. 4.12 shows a pace to trot transition experienced by one of the runs from Fig. 4.10 by plotting all $F_i^{\text{GRF}}$ from 0-10 seconds with respect to the oscillator phases of the RF reference leg. The thick blue lines show $F_i^{\text{GRF}}$ during the first gait cycle and the thick orange lines show

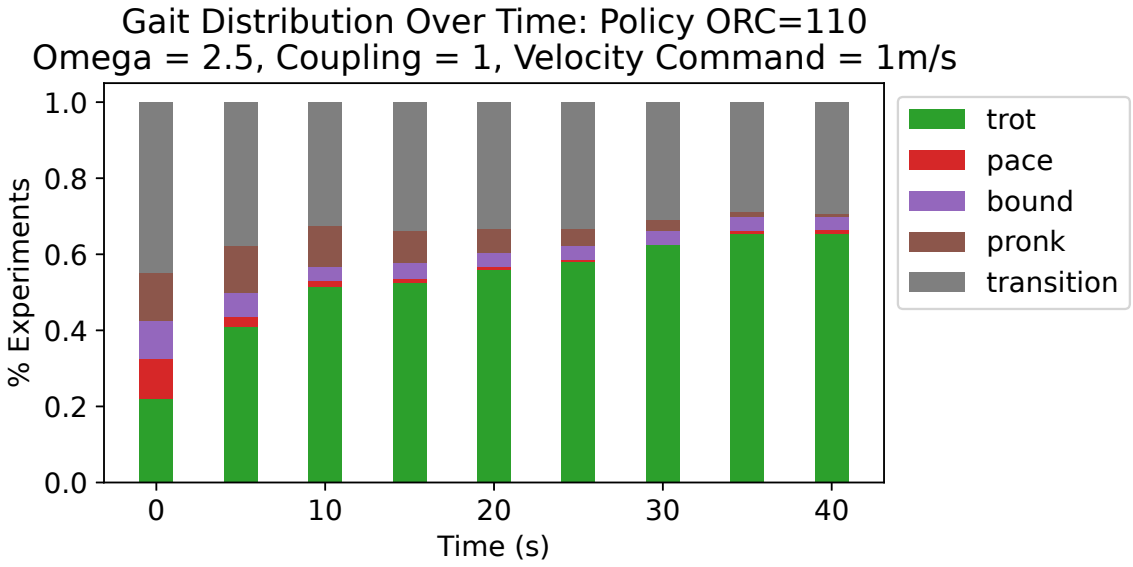


Figure 4.11: The distribution of gaits for 500 runs of ORC(110) with $\sigma = 1$ settles into trot slowly, with more environments transitioning at the beginning but others continuing to slowly converge toward trot as runs with RPD initialized further away become more trot-like over time due to the oscillator dynamics.

$F_i^{\text{GRF}}$ 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 $\pi$ respectively. Those dots do not exactly overlap for the RF leg because of discrete time-step errors. The $F_i^{\text{GRF}}$ of each leg follows its own phases, and over time the gait transitions to diagonal legs being in phase with each other, settling into a trot.

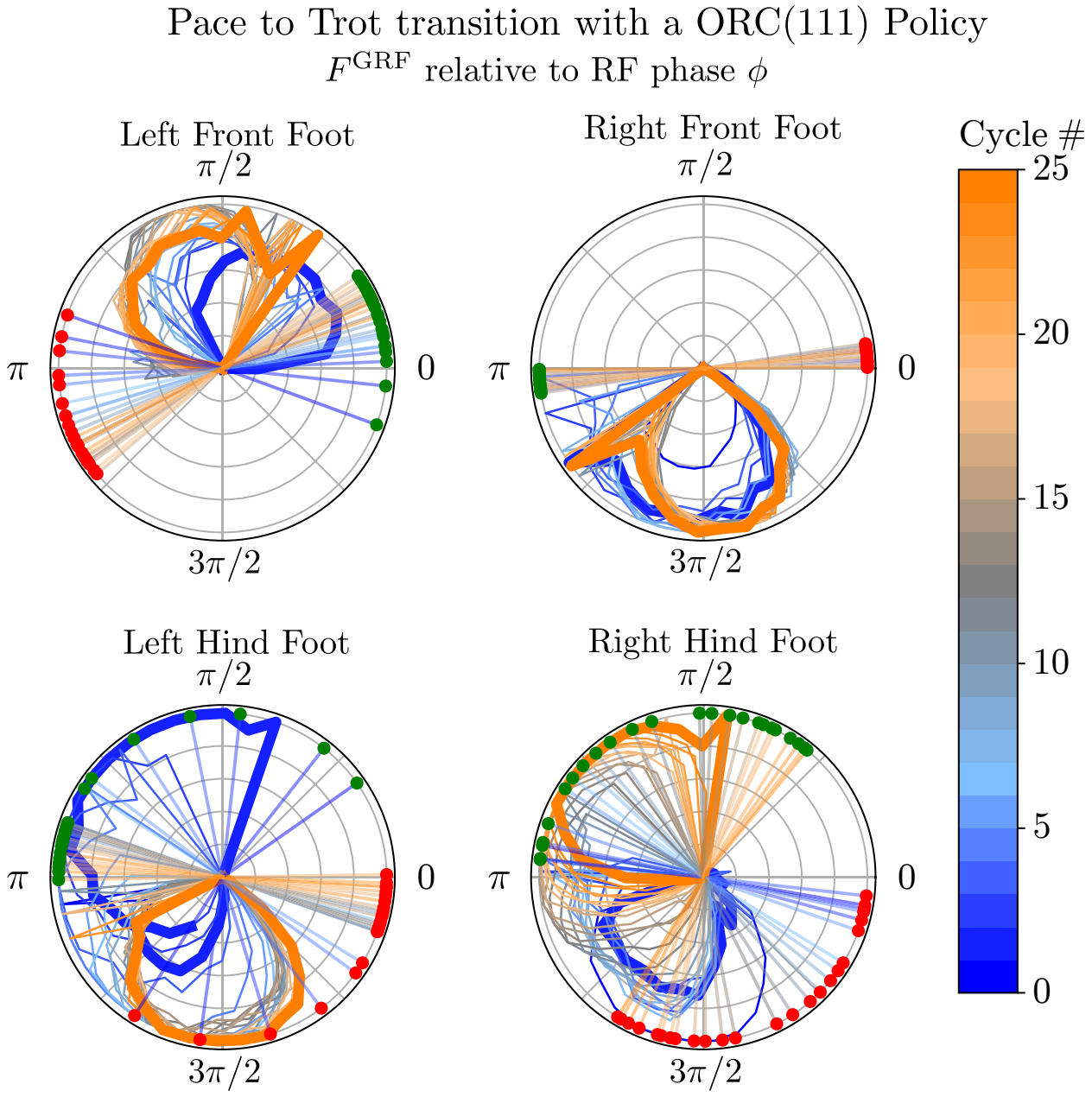


Figure 4.12: The ground reaction force $F_i^{\mathrm{GRF}}$ is plotted for each foot relative to the RF $\phi$, with initial gait cycles in blue and progressing through time to orange. The oscillator 0 and $\pi$ of each leg are shown with red and blue dots, respectively. The bold cycles are the initial 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 $F^{\mathrm{GRF}}$ falling between the green dot $\pi$ crossings and red dot $2\pi$ crossings in every gait cycle.

## 4.3 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 $\mathrm{m/s}$, with 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 impulse perturbations spaced at 10 degree intervals is applied to a different set of 36 robots, over a period of 0.5 seconds.

Fig. 4.13 shows frames of an example of this experiment scaled down to a smaller number of environments for rendering clarity.

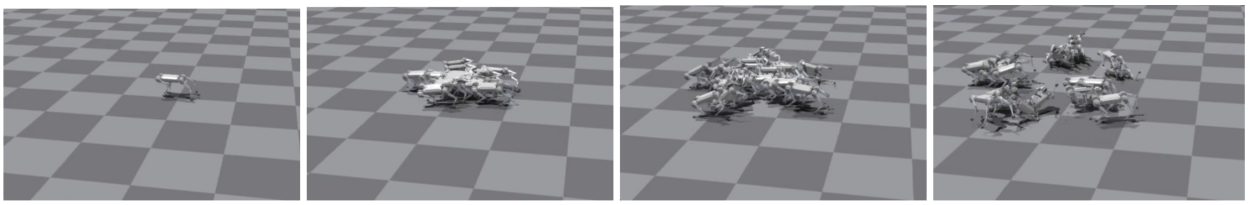

Figure 4.13: Frames from impulse disturbance experiment on ORC(111) policy.

As all policies exhibit a frequency of roughly 4 Hz at this commanded speed, staggering the perturbations ensures that we apply perturbations at different phases of the gait, with different numbers of feet on the ground. We verified this frequency via a fast Fourier transform on the joint positions for policies without phase observations. We then calculate the mean and standard deviation of the failure rate across all ten policies learned for each permutation of ORC(xxx). Failures are counted using terminations where the body of the robot model experiences a collision force from the ground plane.

This entire process is repeated for 5 different impulse magnitudes. The failure rate means and standard deviations, reported in Table 4.1, show that policies trained under ORC(111) consistently have a lower failure rate compared to all other permutations.

Surprisingly, toggling on phase-based rewards seems to have a stronger impact than the phase-based observations, even for cases such as ORC(010) and ORC(011) where the reward is not

Table 4.1: 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** |

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.

# Chapter 5

# Discussion

We presented an augmentation of the quadruped robot state space using one decentralized phase oscillator per leg, 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[1] between the policies, which matches our experience in hardware that local stability is not a useful proxy for legged system 'stability'. Bellegarda and Ijspeert [18] 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. In future work, we hope to quantify the benefit of the phase oscillator observations for sim-to-real.

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, Fern, *et al.* [12]. However, although activating the coupling during evaluation does cause these policies

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

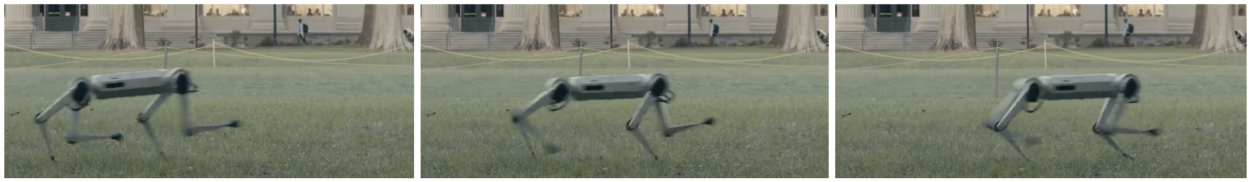

Figure 5.1: Frames from rolling out a ORC(111) policy initialized to trotting with $\sigma = 0$ on MIT Mini Cheetah hardware in grassy environment. The robot was stable in spite of terrain inconsistencies. Further preliminary hardware tests are shown in the supplementary video.

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 [20]: 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 [19] that CPGs may act as both an observer in addition to 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.

## 5.1 Future Work

Anecdotally, before we introduced the offset term in equation (3.2), 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, due to the coupling term, and thus biases training towards pronking gaits. We also observed different gaits to emerge more frequently at different velocities, or with different morphologies.

For example, Fig. 5.3 shows that the gait convergence distribution for the same ORC(110) policy evaluated at a higher commanded velocity (3m/s) and $\omega = 4$, $\sigma = 1$ has a higher proportion of bounding than trotting, which previously dominated the lower velocity trials shown in Fig. 4.11].

Preliminary exploration into morphology effects show that increasing the leg shank length in

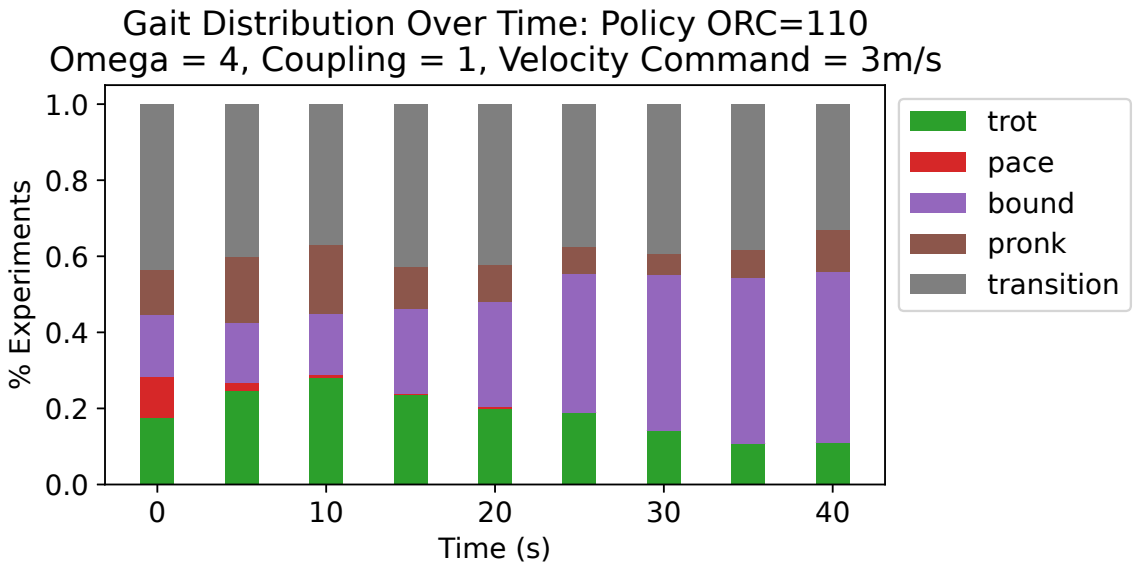


Figure 5.2: The distribution of gaits for 500 randomly initialized runs of ORC(110) with  = 1 evaluated at a higher commanded velocity of 3m/s yields more agents that settle into a bounding gait.

the MIT Mini Cheetah URDF from the original 27cm to 29cm and 32cm during policy evaluation (with all other parameters and initialization kept constant), can yield drastically different gaits post-convergence. Fig. 5.3 shows the elongated collision bodies for the legs in the first row, and in the second row a plot of the phase oscillator differences through time, which form neat limit cycles in blue with some amount of error when the gait settles. The red dots on the 3D plots indicate when the front right leg oscillator passes the same point in the $[0, 2\pi)$ cycle. The last row plots the normalized GRF felt by each leg over time, with the rising edges staggered differently showing that the robots don't converge to the same gait.

Although quantifying the exact convergence and transition patterns is out of the scope of this paper, we recognize the potential for exploring differences in the distribution of preferred stable gaits when these parameters change along with robot morphology. Studying the effects of physical parameters such as center of mass location and leg length on gait emergence for a quadruped robot and verifying convergence patterns on hardware could yield fascinating connections to gait patterns observed in nature for animals of different sizes [33].

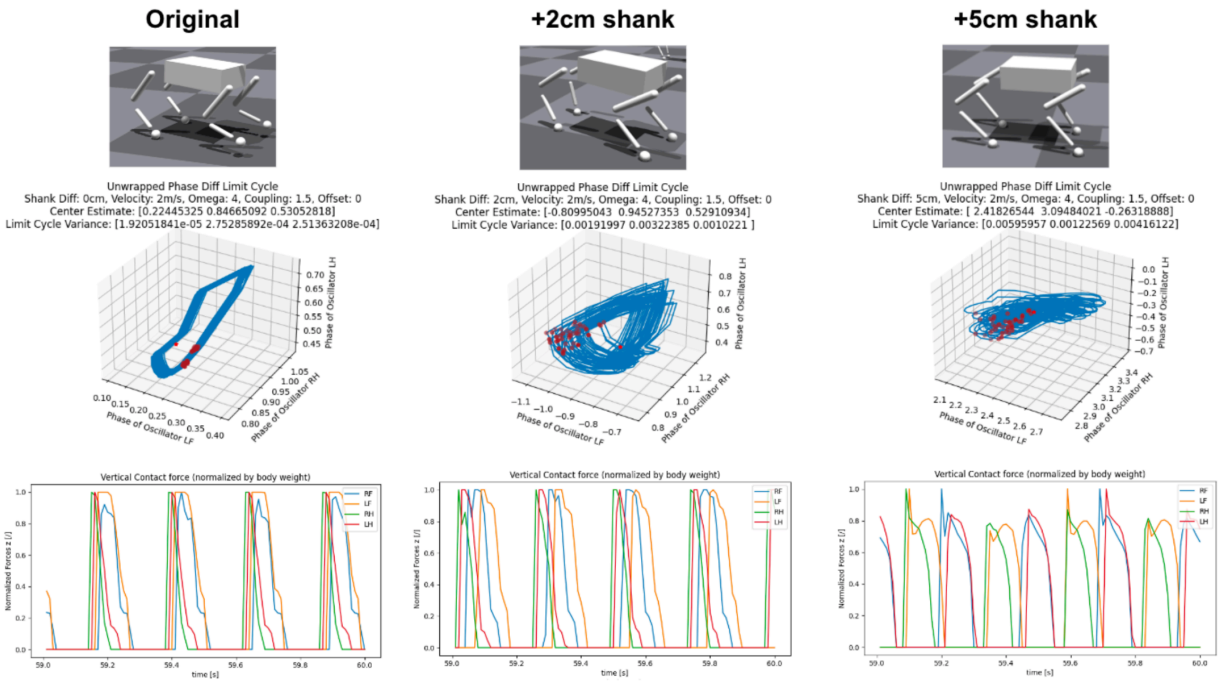


Figure 5.3: All else kept constant, increasing the leg shank length of the robot can drastically change the final gait oscillator phase limit cycle convergence and the corresponding footfall pattern.

Another avenue we find intriguing is the role oscillators may play in a hierarchical RL setting. Higher levels of hierarchy typically reason in both a lower-dimensional space and at a slower timescale; the phase oscillators could be interpreted as a latent state with cyclic dynamics [34]. A latent state space with cyclic dynamics could serve for temporal abstraction, multi-joint coordination and amortized control for cyclic behavior [35], a direction we find very promising.

# References


[1] D. F. Hoyt and C. R. Taylor, "Gait and the energetics of locomotion in horses," *Nature*, vol. 292, no. 5820, pp. 239–240, 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*, vol. 220, no. 10, pp. 1864–1874, 2017.

[3] M. Hildebrand, "Symmetrical gaits of dogs in relation to body build," *Journal of Morphology*, vol. 124, no. 3, pp. 353–359, 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 *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 2245–2252.

[5] W. Xi, Y. Yesilevskiy, and C. D. Remy, "Selecting gaits for economical locomotion of legged robots," *The International Journal of Robotics Research*, vol. 35, no. 9, pp. 1140–1154, 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] 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*, vol. 7, no. 62, eabk2822, 2022.

[9] G. Feng, H. Zhang, Z. Li, X. B. Peng, B. Basireddy, L. Yue, Z. Song, L. Yang, Y. Liu, K. Sreenath, *et al.*, "Genloco: Generalized locomotion controllers for quadrupedal robots," in *Conference on Robot Learning*, PMLR, 2023, pp. 1893–1903.

[10] Z. Xie, X. Da, M. Van de Panne, B. Babich, and A. Garg, "Dynamics randomization revisited: A case study for quadrupedal locomotion," in *2021 IEEE International Conference on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 4955–4961.

[11] Y. Jin, X. Liu, Y. Shao, H. Wang, and W. Yang, "High-speed quadrupedal locomotion by imitation-relaxation reinforcement learning," *Nature Machine Intelligence*, vol. 4, no. 12, pp. 1198–1208, 2022.

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

[13] G. B. Margolis and P. Agrawal, "Walk these ways: Tuning robot control for generalization with multiplicity of behavior," in *Proceedings of The 6th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, PMLR, 2023, pp. 22–31.

[14] 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*, vol. 7, no. 2, pp. 1230–1237, 2021.

[15] Y. Yang, T. Zhang, E. Coumans, J. Tan, and B. Boots, "Fast and efficient locomotion via learned gait transitions," in *Conference on Robot Learning*, PMLR, 2022, pp. 773–783.

[16] 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.

[17] 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*, vol. 95, no. 6, pp. 645–664, 2006.

[18] G. Bellegarda and A. Ijspeert, "Cpg-rl: Learning central pattern generators for quadruped locomotion," *IEEE Robotics and Automation Letters*,

[19] 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*, vol. 226, no. 15, 2023.

[20] H. X. Ryu and A. D. Kuo, "An optimality principle for locomotor central pattern generators," *Scientific Reports*, vol. 11, no. 1, p. 13 140, 2021.

[21] 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.

[22] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.

[23] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. P. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," *CoRR*, vol. abs/1602.01783, 2016.

[24] N. Rudin, D. Hoeller, P. Reist, and M. Hutter, "Learning to walk in minutes using massively parallel deep reinforcement learning," in *Proceedings of the 5th Conference on Robot Learning*, ser. Proceedings of Machine Learning Research, PMLR, 2022.

[25] J. Kober, J. A. Bagnell, and J. Peters, "Reinforcement learning in robotics: A survey," *The International Journal of Robotics Research*, vol. 32, 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*, vol. 263, pp. 41–56, 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*, vol. 315, no. 5817, pp. 1416–1420, 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 *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Ieee, 2013, pp. 3390–3397.

[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*, vol. 10, no. 81, p. 20 120 908, 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*, vol. 16, no. 4, e1007804, 2020.

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

[32] B. Katz, J. Di Carlo, and S. Kim, "Mini cheetah: A platform for pushing the limits of dynamic quadruped control," in *2019 international conference on robotics and automation (ICRA)*, IEEE, 2019, pp. 6295–6301.

[33] N. C. Heglund, C. R. Taylor, and T. A. McMahon, "Scaling stride frequency and gait to animal size: Mice to horses," *Science*, vol. 186, no. 4169, pp. 1112–1113, 1974.

[34] S. Starke, I. Mason, and T. Komura, "Deepphase: Periodic autoencoders for learning motion phase manifolds," *ACM Transactions on Graphics (TOG)*, vol. 41, no. 4, pp. 1–13, 2022.

[35] J. Merel, M. Botvinick, and G. Wayne, "Hierarchical motor control in mammals and machines," *Nature communications*, vol. 10, no. 1, p. 5489, 2019.