Robotica (2026), **44**, pp. 150–168
doi:10.1017/S0263574725103007

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Learning robust bipedal running via structured gait and trajectory guidance

Yunpeng Liang[1], Zhihui Peng[2], Yanzheng Zhao[1] and Weixin Yan[1]

[1]School of Mechanical Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]AgiBot Technology Co. Ltd., Shanghai, China
Corresponding author: Weixin Yan; Email: xiaogu4524@sjtu.edu.cn

Received: 6 April 2025; **Revised:** 29 September 2025; **Accepted:** 5 November 2025; **First published online:** 13 January 2026

Keywords: legged robots; humanoid robots; reinforcement learning; dynamic locomotion; structured gait and trajectory

Abstract

Legged robots have demonstrated remarkable potential for dynamic locomotion and terrain adaptability, making them a prominent focus of research. However, achieving robust and agile bipedal running remains challenging due to the complex dynamics of legged locomotion. In this paper, we propose a reinforcement learning framework for robust bipedal running, incorporating a simple reference trajectory generator and an asymmetric actor-critic architecture. The reference generator, based on kinematics, provides diverse trajectory references while preserving key gait characteristics, facilitating efficient policy exploration. To mitigate the simulation-to-reality gap, we extract latent variables encoding environmental and motion information from dual historical observations. Our method simplifies the trajectory generation process while maintaining effective guidance for learning. Extensive simulation and physical experiments demonstrate that, compared to model-based and learning-based baselines, our approach achieves higher agility, more accurate velocity tracking, and stronger disturbance rejection while preserving gait stability. The resulting controller exhibits spring–mass running dynamics that remain robust on both flat and uneven terrains.

1. Introduction

Legged robots have gained significant popularity over the past two decades, driven by impressive demonstrations of whole-body locomotion and balance by researchers. Beyond their agility and robust locomotion capabilities, a key advantage is their ability to traverse complex and unstructured terrains, including uneven ground, fields, continuous steps, and stairs [1–4]. Their superior terrain adaptability, unmatched by wheeled and tracked robots, makes them invaluable for applications such as field rescue and factory inspection [5]. However, robust and versatile bipedal locomotion presents a significant challenge due to the hybrid and multi-phase dynamics, and factors such as underactuation, unilateral contact constraints, highly nonlinear dynamics, and a large number of degrees of freedom (DoF) contribute to the complexity of modeling and control [6]. Current methodologies for legged robot control can be broadly categorized into two paradigms: model-based and learning-based approaches.

As discussed in ref. [7], model-based methods that use numerical optimization have been extensively reviewed, highlighting the ongoing challenges to achieve agile and robust locomotion. A significant limitation of model-based methods is the difficulty in accurately representing the robot's dynamics in real-world environments and real-time control performance. To achieve dynamic locomotion such as running, Wensing et al. [8] and Englsberger et al. [9] modeled the robot as a Spring-Loaded Inverted Pendulum (SLIP), drawing inspiration from animal locomotion. They achieved highly dynamic bipedal running through offline periodic trajectory generation and online feedback control, but their approach was only validated in simulation. Liang et al. [10] proposed a hierarchical control framework combining

© The Author(s), 2026. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

nonlinear Model Predictive Control (MPC) and Whole-Body Control (WBC) to enhance robustness. A low-frequency MPC optimized the Center of Mass (CoM) trajectory and ground reaction forces, while a high-frequency reduced-dimensional WBC tracked these trajectories. William et al. [11] and Koushil et al. [12] employed a full-body dynamics model and used the Hybrid Zero Dynamics (HZD) method to design periodic gaits offline, with online feedback controllers enforcing virtual constraints [13]. While this approach enabled long-duration running on the Casse robot, its heavy reliance on offline planning and feedback adjustments reduced adaptability to disturbances. Moreover, inherent modeling errors limited the robustness of the resulting controllers, requiring heuristic compensations and extensive manual tuning – a labor-intensive process heavily dependent on expert experience.

With recent advancements in learning-based methods, deep reinforcement learning (RL) has proven highly effective and promising for the dynamic control of underactuated robots, including bipeds [6], quadrupeds [4], and wheeled robots [14]. This success is largely attributed to the powerful function approximation and adaptability of neural networks, along with the availability of abundant training data, allowing RL to find an optimal policy through trial and error. Peng et al. [15] proposed a learning framework that enables bipedal robots to acquire multiple tasks in simulation by mimicking human motion capture data. However, obtaining high-quality mocap data can be challenging. To address this, Xie et al. [16, 17] leveraged motion data generated by manually tuned controllers as a reference for learning. Similarly, Green et al. [18] incorporated SLIP dynamics to encourage the emergence of spring-mass-like locomotion. Additionally, Li et al. [19] utilized an HZD gait library to expand the range of learned behaviors, enabling more diverse and dynamic locomotion.

Although significant progress has been made in the locomotion of bipedal robots, most existing approaches primarily focus on achieving stable omnidirectional walking. However, the ability to seamlessly achieve more dynamic gaits like running remains largely unexplored. Developing learning-based controllers that can generalize various locomotion gaits while maintaining robustness under diverse environmental conditions remains a significant challenge. To achieve versatile bipedal locomotion, Siekmann et al. [20] and Xue et al. [21] introduced a universal reward function adaptable to various gaits, primarily penalizing excessive plantar velocity during the stance phase and excessive plantar force during takeoff. However, as a reference-free approach, it encourages extensive exploration, which can lead to unnatural motion. In contrast, RL with expert reference guidance has proven to be an effective strategy for exploration, which can reduce the complexity of the design of reward functions and disincentivize exploitative behaviors [18]. Additionally, Li et al. [6] developed a robust and versatile controller capable of handling walking, running, and jumping. Their approach integrates skill-specific reference motions derived from multiple sources, including trajectory optimization, human motion capture, and animation, to enhance learning efficiency and adaptability. Despite the success of learning from various reference motions, generating these references remains a time-consuming and labor-intensive process, requiring researchers to master a range of complex techniques. Wu et al. [22] argued that custom sine waves are sufficient for learning bipedal gaits with different styles. While sine waves provide a lightweight alternative for generating gait references, they fail to capture the dynamic characteristics essential for running. Developing a simple yet effective reference trajectory generator that facilitates policy exploration without restricting learning flexibility remains an open challenge.

In this paper, we propose an end-to-end reinforcement learning framework guided by structured gait patterns and trajectory information to achieve robust and agile bipedal running. To enhance learning efficiency, we introduce a simple yet effective reference trajectory generator that provides diverse and meaningful trajectory references while preserving key gait characteristics. Additionally, to enable zero-shot transfer to real-world deployment, we employ an asymmetric actor-critic framework, utilizing dual-history observations to extract key information and mitigate the simulation-to-reality gap. Compared to the work in refs. [6, 22], our method significantly simplifies the trajectory generation process while maintaining effective guidance. Both simulation and real-world experiments show that our approach achieves a running gait with strong robustness, disturbance rejection, and terrain adaptability. Our main contributions are

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

- A simple, kinematics-based trajectory generator for full-body joints that assists policy exploration, captures leg dynamics while coordinating the upper body, and reduces data-generation complexity.
- A robust and effective reinforcement learning training framework, deployable directly on bipedal robots, bridging the gap between simulation and real-world application.
- Validated in both simulation and hardware, the proposed method achieves robust, arm-coordinated bipedal running, improves velocity tracking performance and disturbance rejection, and exhibits effective adaptability to complex terrain.

2. Preliminary and problem description

2.1. Humanoid robot platform

The humanoid robot studied in this paper, *Lingxi-X2*, stands 1.4 m tall and weighs 34 kg, with actuation provided by electric motor-driven joints. It features 32 DoF, distributed as follows: 8 in the arms, 12 in the legs, and 3 in the waist. Figure 1 illustrates its overall appearance and joint configuration. In this study, we focus on controlling all leg joints and the shoulder-pitch joints of the arms.

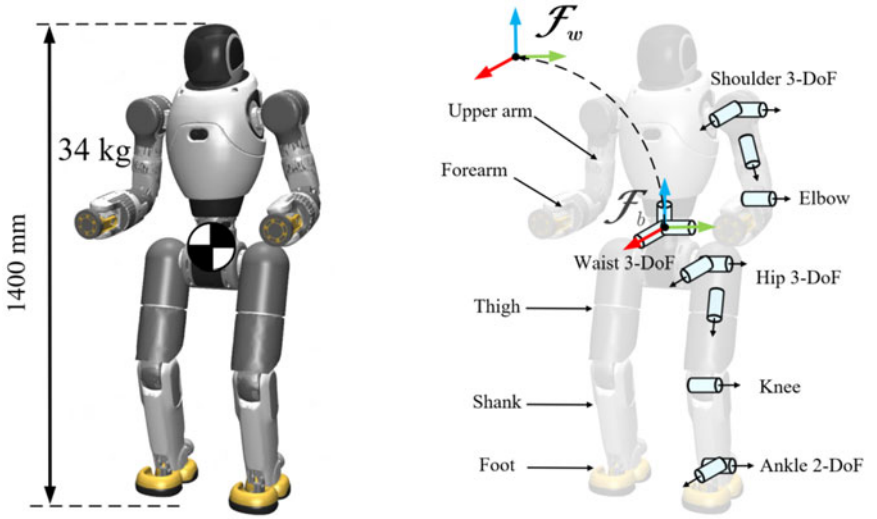


Figure 1. *The appearance and joint configuration of our studied robot.*

2.2. Structured gait and trajectory generation

To provide prior guidance for versatile bipedal locomotion, we employ a simple and effective method to generate reference joint trajectories and contact sequences across diverse gaits. Motivated by ref. [20], we define different gaits using the stepping period T_s, the swing phase duty cycle δ, and the phase offset ψ. In one gait cycle, the contact sequences $\mathbb{1}_{i,\text{contact}}, i = 0, 1$ can be derived from the signals $C_{z,i}(\phi)$ associated with the time-varying periodic phase variable $\phi \in [0, 1]$. These signals $C_{z,i}(\phi) \in [0, 1]$ are polynomial spline curves and can be defined as:

$$C_{z,0}(\phi) = \begin{cases} \sum_{n=0}^{5} \alpha_n \left(\dfrac{\phi + \psi}{2\delta} \right)^n & \phi \in [0, \delta - \psi) \\ 0 & \phi \in [\delta - \psi, \psi) \\ \sum_{n=0}^{5} \alpha_n \left(\dfrac{\phi - \psi}{2\delta} \right)^n & \phi \in [\psi, 1] \end{cases} \tag{1}$$

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

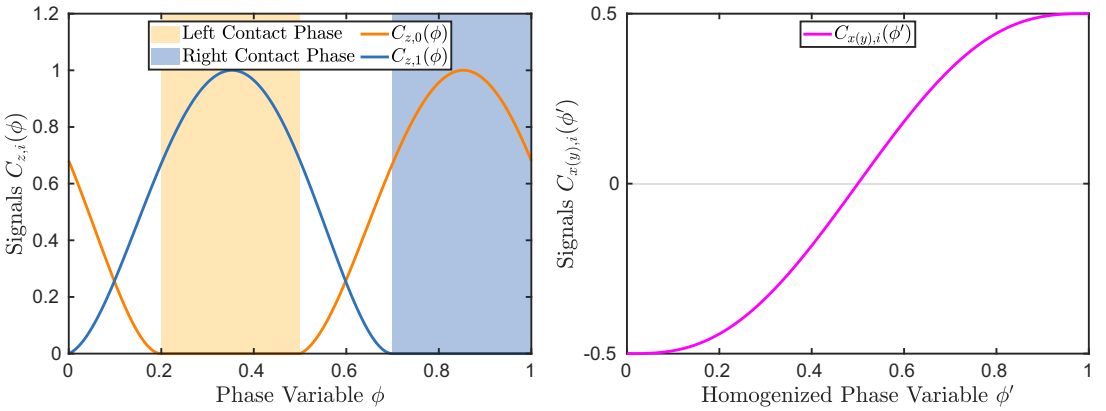


Figure 2. *The variation of the quintic polynomial curves $C_{z,i}(\phi)$ within one running cycle, where $i=0$ and $i=1$ correspond to the left and right legs, respectively, given $\psi=0.5$ and $\delta=0.7$.*

$$C_{z,1}(\phi)=\begin{cases}\sum_{n=0}^{5}\alpha_n\left(\dfrac{\phi}{2\delta}\right)^n & \phi\in[0,\delta)\\ 0 & \phi\in[\delta,1]\end{cases} \tag{2}$$

where the polynomial coefficients are defined as $\boldsymbol{\alpha}=[\alpha_0,\cdots,\alpha_5]^T=[0,0.1,5.0,-18.8,12.0,9.6]^T$ and the phase offset ψ is set to 0.5 for walking or running gaits, and 0 for standing or jumping gaits. Take the running gait as an example, Figure 2 illustrates the variation of the quintic polynomial curves $C_{z,i}(\phi)$ w.r.t the phase variables ϕ, given $\psi=0.5$ and $\delta=0.7$. A signal value greater than zero indicates the swing phase, while a value of zero corresponds to the support phase. The gait stance indicator $\mathbb{1}_{\text{contact}}$ is defined as:

$$\mathbb{1}_{i,\text{contact}}=\begin{cases}0, & \text{if } C_{z,i}(\phi)>0,\\ 1, & \text{otherwise.}\end{cases} \tag{3}$$

Similarly, in the x and y directions, the spline curves $C_{x(y),i}(\phi)\in[-0.5,0.5]$ are employed to map the stride length to the displacement relative to the initial default position on either side, which can be defined as:

$$C_{x(y),i}(\phi)=-6\phi_i^{'5}-15\phi_i^{'4}+10\phi_i^{'3}-0.5 \tag{4}$$

$$\phi_0^{'}=\begin{cases}\dfrac{\phi+\psi}{\delta} & \phi\in[0,\delta-\psi)\\ \dfrac{\phi-\psi+1-\delta}{1-\delta} & \phi\in[\delta-\psi,\psi)\\ \dfrac{\phi-\psi}{\delta} & \phi\in[\psi,1]\end{cases} \tag{5}$$

$$\phi_1^{'}=\begin{cases}\dfrac{\phi}{\delta} & \phi\in[0,\delta)\\ 1-\dfrac{\phi-\delta}{1-\delta} & \phi\in[\delta,1]\end{cases}$$

where $\phi_i^{'}\in[0,1]$ are homogenized phase variables by transforming the original phase variables ϕ using the phase offset ψ and the swing phase duty cycle δ.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

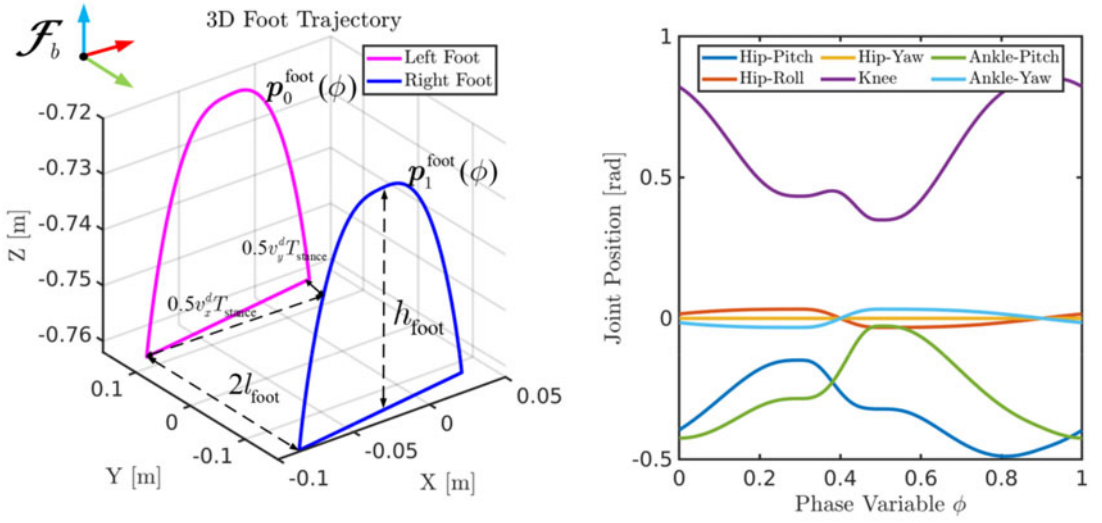


Figure 3. *Foot and joint space trajectories calculated based on the polynomial curves and inverse kinematics, with* $\boldsymbol{v}_x^d = 1.0m/s$ *and* $v_y^d = 0.3m/s$.

Inspired by the Raibert heuristic foothold strategy in ref. [23], given the structured gaits, the trajectory of the leg's end $\boldsymbol{p}_i^{\text{foot}}$ relative to the body frame can be expressed as:

$$\begin{aligned} p_{x,i}^{\text{foot}}(\phi) &= (v_x^d + s_i \omega_z^d l_{\text{foot}}) \frac{(1-\delta)T_s}{2} C_{x,i}(\phi) \\ p_{y,i}^{\text{foot}}(\phi) &= s_i l_{\text{foot}} + v_y^d \frac{(1-\delta)T_s}{2} C_{y,i}(\phi) \\ p_{z,i}^{\text{foot}}(\phi) &= h_{\text{foot}} C_{z,i}(\phi) \end{aligned} \tag{6}$$

where v_x^d, v_y^d, ω_z^d denote the desired linear and angular velocity commands, respectively. The half-distance of foot placement and swing height are represented by l_{foot} and h_{foot}, respectively, as illustrated in Figure 3. A signed variable $\boldsymbol{s} = [1, -1]^T$ distinguishes between the left and right legs.

Subsequently, analytical inverse kinematics is employed to compute the reference joint position $\boldsymbol{q}_{\text{leg}}^{\text{ref}} \in \mathbb{R}^{12}$. Figure 3 illustrates the trajectories of both the leg ends and the joints, with $v_x^d = 1.0m/s$ and $v_y^d = 0.3m/s$. Additionally, the arm reference trajectory can be simply defined as:

$$\begin{aligned} q_{\text{arm},0}^{\text{ref}}(\phi) &= q_{\text{arm}}^{\max} \cdot \sin[2\pi \cdot (\phi - (1-\delta))] \\ q_{\text{arm},1}^{\text{ref}}(\phi) &= q_{\text{arm}}^{\max} \cdot \sin[2\pi \cdot (\phi - (1-\delta) + \psi)] \end{aligned} \tag{7}$$

where $q_{\text{arm}}^{\max}$ denotes the maximum swing range of the shoulder-pitch joints.

In the subsequent RL training, each set of randomized speed commands will correspond to a set of reference joint positions $\boldsymbol{q}^{\text{ref}} = [\boldsymbol{q}_{\text{leg},0}^{\text{ref}}, q_{\text{arm},0}^{\text{ref}}, \boldsymbol{q}_{\text{leg},1}^{\text{ref}}, q_{\text{arm},1}^{\text{ref}}]^T \in \mathbb{R}^{14}$ that guide the learned locomotion, with the corresponding contact masks $\mathbb{1}_{\text{contact}}$.

2.3. RL for bipedal robots

Bipedal robot locomotion control can be formulated as a Partially Observable Markov Decision Process (POMDP). In this framework, at each timestep t, the environment resides in a state $\boldsymbol{s}_t$. The robot, acting as the agent, perceives an observation $\boldsymbol{o}_t$ from the environment, executes an action $\boldsymbol{a}_t$, and consequently influences the transition of the environment to a new state $\boldsymbol{s}_{t+1}$, receiving a reward r_t in response. This interaction sequence continues until the completion of an episode of length T.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

Due to partial observability, the robot cannot access the full environmental state $\boldsymbol{s}_t$. Instead, it relies on observable information $\boldsymbol{o}_t$, which excludes the privileged information such as the base linear velocity and system parameters, as discussed in Section 3.2. The observation o_t can be determined by the observation function $O(\boldsymbol{o}_t|\boldsymbol{s}_t, \boldsymbol{a}_{t-1})$. The reinforcement learning objective is to identify an optimal policy π that selects actions a_t to maximize the expected return, expressed as:

$$J(\pi) = \mathbb{E}_{\tau\sim p(\tau|\pi)}\left[\sum_{t=0}^{T}\gamma^t r_t\right] \tag{8}$$

where $\tau = \{(x_0, a_0, r_0), (x_1, a_1, r_1), \ldots\}$ denotes the trajectory of the agent while following policy π, and $p(\tau|\pi)$ represents the probability of observing this trajectory under the policy π. The discount factor $\gamma \in [0, 1]$ prioritizes immediate rewards over future ones.

3. Methodology

In this section, we introduce our RL training framework, designed to enable the robot to learn a robust locomotion policy. This policy is guided by characteristic gaits and trajectories that capture key aspects of dynamic bipedal running.

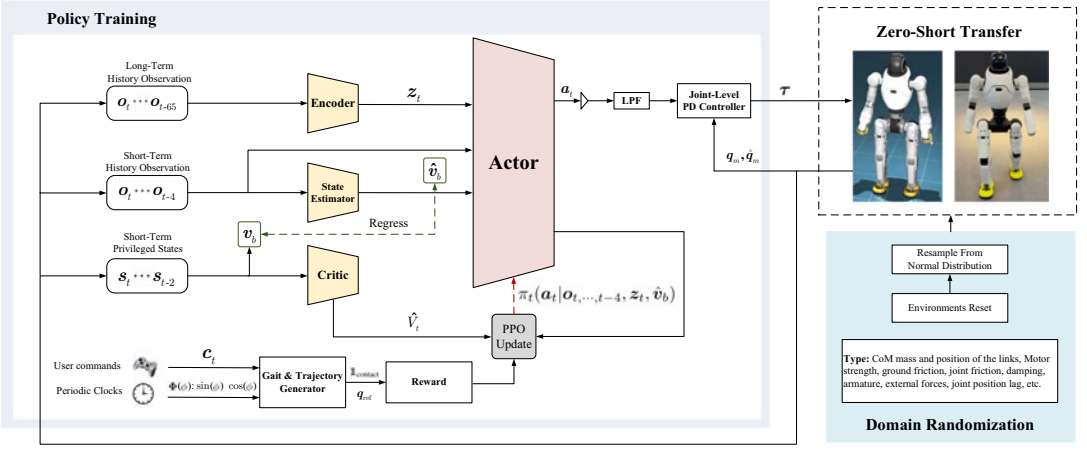


Figure 4. *Our RL control framework utilizes dual-history observations from the robot. The actor policy, running at 100 Hz, processes a 6.6-second history, which is first encoded by a 1D CNN to extract the system's latent representation $\boldsymbol{z}_t$. A short history of four time steps is directly input into the base MLP, along with the estimated velocity $\hat{\boldsymbol{v}}_b$. All networks are trained simultaneously. The actor policy outputs the desired motor positions $\boldsymbol{a}_t$, which are smoothed using an LPF before being transmitted to the joint-level PD controllers to generate motor torques $\boldsymbol{\tau}$.*

3.1. Policy architecture

Figure 4 shows the overview of our control framework. Inspired by ref. [4], we employ an asymmetric actor-critic framework to enable zero-shot transfer from simulation to the real world. This framework integrates a long-term memory encoder and a short-term memory state estimation network [6] to mitigate the sim-to-real gap. The encoder extracts latent information $\boldsymbol{z}_t \in \mathbb{R}^{64}$ (such as terrain, friction, and dynamic parameters) from a 66-frame stack of historical observations. Meanwhile, the state estimation network [24] explicitly estimates the base linear velocity $\hat{\boldsymbol{v}}_b \in \mathbb{R}^3$ using a 3-frame stack of historical observations. Unlike the actor policy input in ref. [6], our approach excludes explicit reference trajectories and does not employ a model-based state estimator. Instead, the structured gait and trajectory implicitly guide the learning process through the reward function, which can facilitate the deployment of policy. All trainable network parameters are provided in Table I.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

Table I. *Network architecture for the actor-critic, encoder, and state estimator networks. The configuration of the 1D encoder CNN's hidden layers is defined as [kernel size, filter size, stride size].*

Module	Inputs	Dim	Hidden Layers	Outputs	Dim
Encoder (CNN)	$\boldsymbol{o}_{t,\cdots,t-65}$	66×53	[6, 32, 3] [4, 16, 2]	$\boldsymbol{z}_t$	64
Estimator (MLP)	$\boldsymbol{o}_{t,\cdots,t-4}$	5×53	[256, 128, 64]	$\hat{\boldsymbol{v}}_b$	3
Actor (MLP)	$[\boldsymbol{o}_{t,\cdots,t-4},\ \hat{\boldsymbol{v}}_b,\ \boldsymbol{z}_t]$	$5 \times 53 + 3 + 64$	[512, 256, 128]	$\boldsymbol{a}_t$	14
Critic (MLP)	$\boldsymbol{s}_{t,\cdots,t-2}$	3×81	[768, 256, 128]	$\hat{V}_t$	1

Convolutional Neural Networks (CNN), Multi-Layer Perceptron (MLP).

3.2. State and action

States $\boldsymbol{s}_t \in \mathbb{R}^{81}$ are divided into two categories: observations and privileged information. Observations, denoted as $\boldsymbol{o}_t \in \mathbb{R}^{53}$, consist of data directly obtainable from the user and the robot, including the user commands $\boldsymbol{c}_t \in \mathbb{R}^3$, periodic clocks $\Phi(\phi) = [sin(\phi),\ cos(\phi)]^T \in \mathbb{R}^2$, and onboard sensor data. Privileged information, represented by $\boldsymbol{p}_t \in \mathbb{R}^{28}$, includes data that is challenging to obtain or prone to significant noise. It is worth noting that no noise is introduced into the critic network to preserve its guiding role and facilitate stable learning during training. Instead, appropriate random noise and time delays are applied to the dual historical observations to enhance the robustness of the actor policy. Table II lists the observations and privileged information in the state space.

Table II. *Composition of the state space.*

Observation	Dim	Privileged State	Dim
Commands ($\boldsymbol{c}_t$)	3	Body mass	1
Periodic clocks ($\Phi(\phi)$)	2	Ground friction	1
Joint position ($\boldsymbol{q}$)	14	Push force	2
Joint velocity ($\dot{\boldsymbol{q}}$)	14	Push torque	3
Last action ($\boldsymbol{a}_{t-1}$)	14	Base linear velocity	3
Base angular velocity	3	Feet contact indicator ($\mathbb{1}_{\text{force}}$)	2
Base Euler angle	3	Gait stance indicator ($\mathbb{1}_{\text{contact}}$)	2
		Joint position diff ($\boldsymbol{q} - \boldsymbol{q}^{\text{ref}}$)	14

The action $\boldsymbol{a}_t \in \mathbb{R}^{14}$ represents the desired joint positions, encompassing both the leg joints and the arm shoulder-pitch joints. A joint-level PD controller subsequently computes the motor torques based on the smoothed action using a low-pass filter (LPF). The policy operates at 100 Hz, while the PD controller runs at 1 kHz.

3.3. Reward

We formulate the reward structure to enforce specific gait locomotion while ensuring stability and adaptability. Given that gait and trajectory guidance are derived from the generator in Section 2.2, we introduce tailored reward terms to incorporate prior knowledge. The overall reward function is formulated as:

$$r = r_{\text{command}} + r_{\text{gait}} + r_{\text{regularization}} \tag{9}$$

where r_{command} denotes the reward for tracking external commands, and the term r_{gait} ensures adherence to the structured locomotion gait. Additionally, to suppress undesirable motions, a set of regularization rewards $r_{\text{regularization}}$ is introduced to assist smooth and stable action.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

3.3.1. Command tracking

Given the desired linear velocity v_x^d, v_x^d, angular velocity ω_z^d, and the base height h_b^d, the tracking rewards are defined as:

$$\begin{aligned} r_{\text{linear}} &= 3 \cdot \exp\left(-8 \cdot \|v_{x,y}^d - v_{x,y}\|^2\right) \\ r_{\text{angular}} &= 1.5 \cdot \exp\left(-8 \cdot \|\omega_z^d - \omega_z\|^2\right) \\ r_{\text{height}} &= 0.2 \cdot \exp\left(-100 \cdot \|h_b^d - h_b\|\right) \end{aligned} \tag{10}$$

To ensure the robot follows the commanded velocity while avoiding excessive deviations, we define a speed tracking reward r_{speed}. Specifically, we penalize low speeds when the robot moves significantly slower than the commanded velocity, assign a neutral reward for excessive speeds, and provide a positive reward when the speed falls within the desired range. Furthermore, since moving in the wrong direction severely affects task performance, we impose a higher penalty for sign mismatches between the actual and commanded velocities. The reward function is defined as:

$$r_{\text{speed}} = \begin{cases} -1.0, & \text{if } |v_x| < 0.5 \cdot |v_x^d| \\ 0.0, & \text{if } |v_x| > 1.2 \cdot |v_x^d| \\ 1.2, & \text{if } 0.5 \cdot |v_x^d| \le |v_x| \le 1.2 \cdot |v_x^d| \\ -2.0, & \text{if } \text{sign}(v_x) \neq \text{sign}(v_x^d) \end{cases} \tag{11}$$

3.3.2. Gait style

The gait reward r_{gait} comprises three components. The first term, r_{contact}, encourages the foot contact pattern to align with the expected structured gait phase, ensuring stable and coordinated locomotion. The second term, r_{swing}, reinforces rewards and penalties for feet in the air phase, i.e., $\mathbb{1}_{\text{contact}} = \boldsymbol{0}$, thereby facilitating diverse and adaptable gait patterns. This reward is crucial for learning more complex gaits, as explained in Section 4.2.2. The third term, r_{joint}, promotes adherence to predefined joint trajectories, minimizing unnecessary deviations during the exploration process. Given the gait stance indicator $\mathbb{1}_{\text{contact}}$ and the joint reference $\boldsymbol{q}^{\text{ref}}$ derived in Section 2.2, the overall gait reward is formulated as follows:

$$\begin{aligned} r_{\text{gait}} &= 1.0 \cdot \sum_{i=0}^{1} r_{\text{i,contact}} + 2.0 \cdot r_{\text{swing}} + 2.2 \cdot r_{\text{joint}} \\ r_{\text{i,contact}} &= \begin{cases} 1.0, & \text{if } \mathbb{1}_{\text{i,force}} = \mathbb{1}_{\text{i,contact}} \\ -0.4, & \text{if } \mathbb{1}_{\text{i,force}} \neq \mathbb{1}_{\text{i,contact}} \end{cases} \\ r_{\text{swing}} &= \begin{cases} 1.5, & \text{if } \mathbb{1}_{\text{force}} = \mathbb{1}_{\text{contact}} = \boldsymbol{0} \\ -0.8, & \text{if } \mathbb{1}_{\text{force}} \neq \mathbb{1}_{\text{contact}} = \boldsymbol{0} \end{cases} \\ r_{\text{joint}} &= \exp\left(-2 \cdot \|\boldsymbol{q}_m - \boldsymbol{q}^{\text{ref}}\|\right) - 0.2 \cdot \|\boldsymbol{q}_m - \boldsymbol{q}^{\text{ref}}\| \end{aligned} \tag{12}$$

where $\mathbb{1}_{i,\text{force}}$ denotes the contact indicator for the i-th foot in the simulator.

3.3.3. Regularization

To suppress undesirable motions, a set of regularization rewards $r_{\text{regularization}}$ is introduced to penalize excessive joint torque, high velocities, unnecessary angular motion, joint regularization, action smoothness, etc. (see Table III). Most of the reward functions are shaped by referring to [25].

3.4. Domain randomization

Domain randomization is a widely adopted technique in RL to improve policy robustness and reduce the gap between simulation and real-world deployment. In our training details, we apply domain randomization to various physical and control parameters. Specifically, we introduce variations in environment

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

Table III. *Regularization rewards.*

Reward	Weight	Expression
Joint torques	−8e-9	$\|\boldsymbol{\tau}\|^2$
Joint velocity	−2e-8	$\|\dot{\boldsymbol{q}}\|^2$
Joint acceleration	−1e-7	$\|(\dot{\boldsymbol{q}}_t - \dot{\boldsymbol{q}}_{t-1})/\Delta t\|^2$
Joint limits	−10	$\max(\|\boldsymbol{q}\| - \boldsymbol{q}^{\max}, 0)$
Joint velocity limits	−1	$\text{clip}(\|\dot{\boldsymbol{q}}\| - 0.9 \cdot \dot{\boldsymbol{q}}^{\max}, 0, 1)$
Joint torque limits	−0.1	$\max(\|\boldsymbol{\tau}\| - 0.9 \cdot \boldsymbol{\tau}^{\max}, 0)$
Contact force	−0.002	$\text{clip}(\|\boldsymbol{f}\| - \boldsymbol{f}^{\max}, 0, 400)$
Action smoothness	−0.03	$\|(\boldsymbol{a}_t - \boldsymbol{a}_{t-1})/\Delta t\|^2 + 0.02 \cdot \|\boldsymbol{a}_t\| + \|(\boldsymbol{a}_t - 2 \cdot \boldsymbol{a}_{t-1} + \boldsymbol{a}_{t-2})/\Delta t\|^2$
Base z-axis velocity	0.8	$\exp(-5 \cdot v_{b,z}^2)$
Base roll-pitch velocity	0.8	$\exp(-10 \cdot (\omega_{b,x}^2 + \omega_{b,y}^2))$
Base orientation	1.0	$\exp(-20 \cdot (g_{b,x}^2 + g_{b,y}^2))$
Hip joint regularization	1.3	$\exp(-100 \cdot \|\boldsymbol{q}_{x,z}^{\text{hip}}\|) - 0.01 \cdot \|\boldsymbol{q}_{x,z}^{\text{hip}}\|$
Foot clearance	1.0	$\sum_{\mathbb{1}_{\text{i,contact}}=0} h_i \in (h_{\text{foot}}, h_{\text{foot}}^{\max})$

parameters (e.g., friction and restitution), external disturbances (e.g., forces and torques), and dynamic properties of the robot (e.g., payload mass, link mass, and CoM position). Additionally, we randomize actuation characteristics, including motor strength, joint properties, and PD gains, as well as sensory defects such as action delay and joint position lag. These parameters are uniformly sampled from the predefined ranges, as detailed in Table IV.

4. Experimental results

In this section, we present sim-to-sim evaluations conducted in MuJoCo [26] alongside hardware experiments on the *Lingxi-X2* robot. These evaluations measure our method's performance in terms of speed tracking accuracy, gait cycle stability and consistency, disturbance rejection, and terrain adaptability across different scenarios.

Table IV. *Domain randomization parameters and their ranges. Additive randomization offsets a parameter within the given range, while scaling applies a multiplicative factor.*

Parameter	Unit	Range	Operator
Friction	–	[0, 1.0]	–
Action delay	–	[0, 0.5]	–
Restitution	–	[0.0, 0.4]	–
External force (*xy*)	N	[0, 150]	–
External torque	Nm	[0, 12]	–
Payload mass	kg	[−3, 3]	Additive
Link mass	kg	[0.9, 1.1]	Additive
CoM position	m	[−0.05, 0.05]	Additive
Motor offset	rad	[−0.035, 0.035]	Additive
Motor strength	%	[90, 110]	Scaling
$\boldsymbol{k}_p, \boldsymbol{k}_d$ Gains	%	[80, 120]	Scaling
Joint properties	%	[90, 110]	Scaling
Joint position lag	s	[0, 40Δt]	–

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

Table V. *PPO hyperparameters.*

Parameter	Value
Horizon (H)	24
Adam learning rate	1×10^{-5}
Number of epochs	2
Number of mini-batches	4
Discount (γ)	0.994
Clipping parameter (ϵ)	0.2
Max gradient norm	1

4.1. Setup

We train our policy using the PPO algorithm [27] in the IsaacGym simulator [28], leveraging parallelization across 4096 environments with input normalization. The PPO hyperparameters are provided in Table V. The training process requires approximately 20,000 episodes to achieve a well-performing policy on an Nvidia GeForce 4090 GPU. For a comprehensive evaluation of dynamic locomotion tracking, including stability, disturbance rejection, and adaptability to complex terrain, we compare our approach against the following settings:

- **Baseline** [22]: The policy was trained using custom sine waves as the reference trajectory, with the velocity command $\boldsymbol{c}_t = \mathbf{0}$ in the gait and trajectory generator set to zero.
- **Asymmetric actor-critic** [29]: This method excludes the dual-history window encoding mechanism and uses the same actor and critic observation stack lengths as reported in [29].
- **Model-based method** [10, 30]: The controller adopts a cascade scheme, including the nonlinear MPC [30] based on the centroidal dynamics and the WBC [10] based on the full-body dynamics.
- **Ours**: The proposed method in this paper.

Additionally, we conduct ablation experiments to assess the contributions of the proposed trajectory generator, gait reward function, upper body motion, and dual-channel network architecture to running performance.

- **Ours w/o reference trajectory**: The proposed method without the joint trajectory guidance, i.e., $r_{\text{joint}} = 0$.
- **Ours w/o structured gait** : The proposed method without the gait style guidance, i.e., $r_{\text{swing}} = 0$.
- **Ours w/o whole-body control**: The proposed method excluding the arm swing control, i.e., $\boldsymbol{a}_t \in \mathbb{R}^{12}$.
- **Ours w/o long-term history**: The proposed method without the 2D-CNN module for extracting the latent representation $\boldsymbol{z}_t$ from long-term observations.
- **Ours w/o short-term history**: The proposed method with a single-frame observation instead of the five-frame stack.

All RL-based methods use identical proportional–derivative joint gains, with proportional gains specified as $\boldsymbol{k}_p = 2 \cdot [60,\ 60,\ 40,\ 80,\ 40,\ 30,\ 40]^T$ and differential gains as $\boldsymbol{k}_d = 2 \cdot [3,\ 3,\ 3,\ 4,\ 2,\ 2,\ 3]^T$. The joint sequence is ordered left to right, beginning with the left leg and left arm, followed by the right leg and right arm.

4.2. Ablation study

4.2.1. Network architecture

We conducted ablation studies on the dual-history window state encoding by selectively removing each module to evaluate its individual contribution. The results, shown in Figure 5, demonstrate that the

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

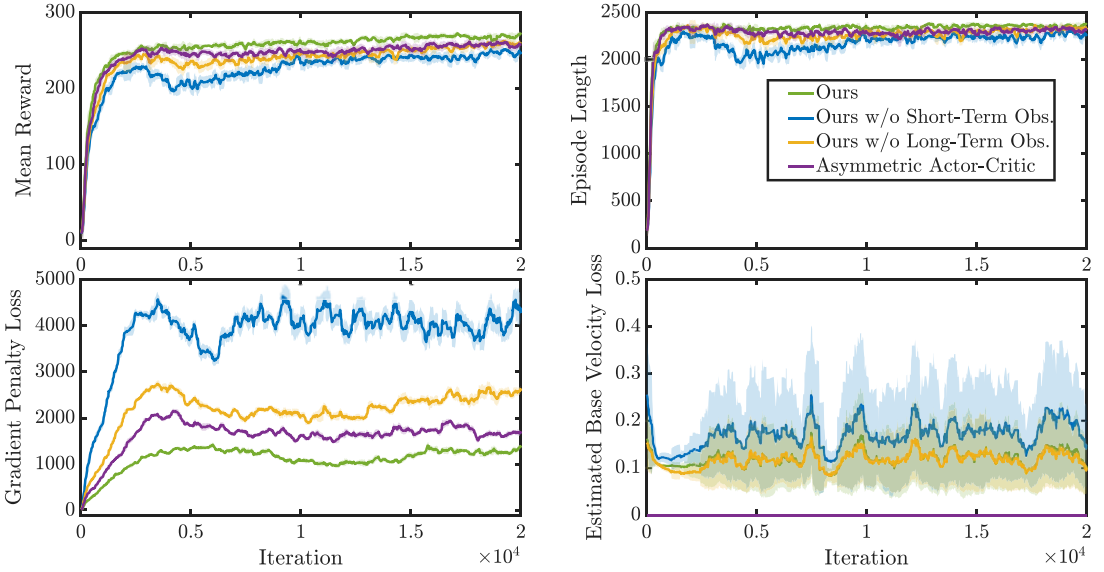


Figure 5. *Ablation study results on the dual-history window state encoding. The plots report the mean reward (top left), episode length (top right), gradient penalty loss (bottom left), and estimated base velocity loss (bottom right). Solid curves represent moving-average filtered data, and shaded regions indicate the standard deviation.*

proposed method consistently outperforms the ablated variants across all evaluation metrics. In particular, our method achieves faster convergence and higher asymptotic performance in terms of mean reward and episode length (Figure 5, top), reflecting improved learning efficiency and task stability. The gradient-penalized loss, which was not included in the overall PPO objective during ablation experiments, is markedly higher for both ablation variants, with the effect especially pronounced when short-term observations are removed (Figure 5, bottom left). Evaluating this loss separately highlights the stabilizing effect of the dual-history window encoding on policy optimization. Moreover, our full method yields lower base velocity loss than the variant without short-term observations (Figure 5, bottom right), while the removal of long-term observations has little impact. This underscores the critical role of short-term history in accurate velocity estimation. Notably, our method also achieves lower gradient penalty loss compared to the configuration in ref. [29], which adopts a simple asymmetric actor–critic network. Furthermore, their approach lacks the capability to estimate real body velocity. Taken together, these results confirm that both short-term and long-term history windows play essential roles in improving the stability and robustness of the learned locomotion policy.

4.2.2. Effect of structured rewards on gait consistency

To evaluate the role of structured gait and trajectory references in maintaining gait consistency, we performed ablation studies on running gaits, which are characterized by a double flight phase. We compared the ground reaction forces (GRFs) of both feet under our proposed method with and without the swing reward r_{swing} and the reference trajectory reward r_{joint}. As shown in the top and middle plots of Figure 6, removing r_{joint} increases apparent alignment with the pre-planned gait schedule, likely due to the increased relative weight of rewards that promote specific motion styles. However, the absence of reference trajectories produces kinematically implausible motions (as illustrated in the supplementary video material), which hinders the learning of robust control policies for highly dynamic tasks such as running. Moreover, excluding r_{swing} results in mismatched gait sequences (Figure 6, bottom). Although the basic contact reward r_{contact} remains, it is insufficient to induce a true running gait: the absence of a double flight phase causes the behavior to degenerate into walking. In contrast, our full method achieves

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

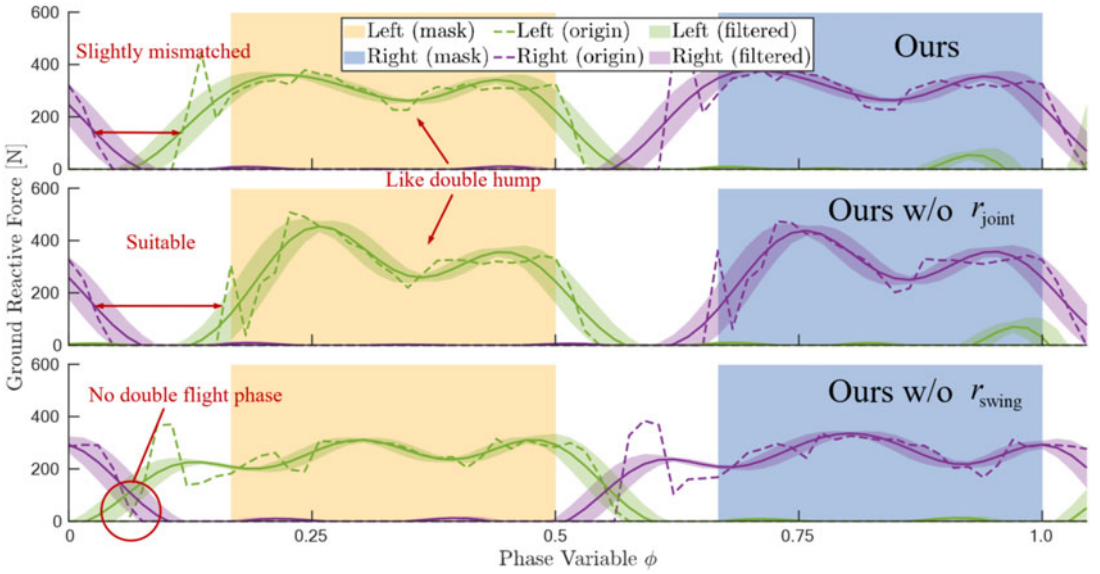


Figure 6. *Ground reaction forces during a* $1.3\,m/s$ *running gait using various method. Solid lines represent low-pass filtered data, while shaded regions indicate data processed with a moving window filter.*

a double-hump GRF pattern purely through kinematic motion planning, consistent with the expected behavior in spring–mass dynamics [18].

4.3. Simulation results

4.3.1. Velocity tracking performance

We evaluate both normal and fast running by starting from rest and ramping to a desired speed and assess tracking performance using the average body height, the average forward velocity, the average velocity tracking reward, the root mean square error (RMSE), and the population standard deviation (PSD) metrics. The results are shown in Figure 7 and Table VI. Given a target speed of $1.3\,m/s$, all methods achieve stable dynamic locomotion, but our method attains the highest average speed and the smallest tracking error. As summarized in Table VI, the maximum achievable speed improves by 9.07% over the baseline. We attribute this gain to the trajectory generator, which supplies reference motions closely aligned with the target speed and thereby guides policy learning, a benefit also reflected in the higher average velocity-tracking reward at convergence.

For a higher target speed of $1.9\,m/s$, the RL-based controller maintains stable locomotion, showing only modest increases in tracking error and body oscillation. In contrast, the model-based controller [10, 30] becomes unstable and often approaches failure (see also the accompanying video). At high running speed, it shows a reduction in body height (Figure 7 middle), a behavior not observed with our

Table VI. *Velocity tracking metrics for different gaits.*

	Running [m/s]			Fast Running [m/s]				h_b [m]
Method	**Avg.**	**RMSE**	**PSD**	**Avg.**	**RMSE**	**PSD**	**Reward**	**RMSE**
NMPC + WBC	1.1861	0.1687	0.1244	/	/	/	/	0.0471
Baseline	1.1719	0.1386	**0.0528**	1.6238	0.2878	**0.0884**	1.2692	0.0145
Ours w/o arm	1.2456	0.0861	0.0667	1.7597	0.1838	0.1188	1.3527	0.0147
Ours	**1.2970**	**0.0581**	0.0558	**1.7710**	**0.1564**	0.0810	**1.3916**	**0.0082**

The bold values indicate that these values outperform others.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

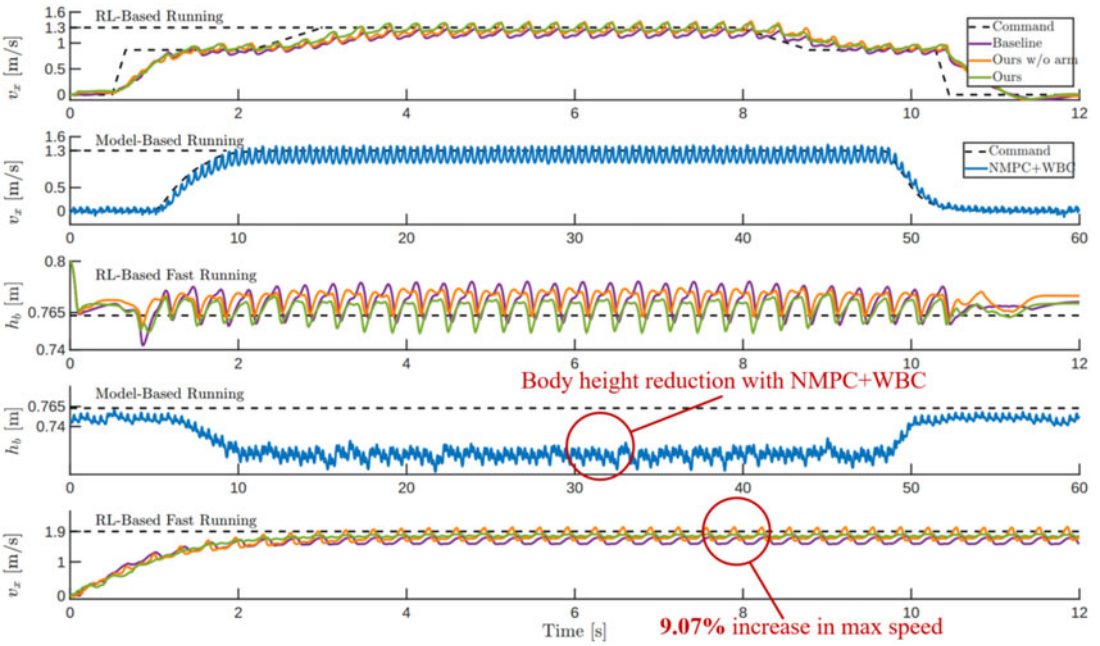


Figure 7. *Forward velocity and body height tracking results. Top two rows: RL-based and model-based controllers under a running gait with desired forward speed $v_x^d = 1.3\,m/s$, gait period $T_s = 0.66\,s$, and duty factor $\delta = 0.65$. Middle two rows: corresponding body-height variations. Bottom row: RL-based controller under fast running with $v_x^d = 1.9\,m/s$ (same T_s and δ).*

policy. These discrepancies reflect an amplified discrepancy between the dynamics model and reality in highly dynamic scenes, which degrades model-based control. Leveraging a neural network's nonlinear capacity and large-scale interaction data, our RL policy achieves greater locomotion stability and robustness. Furthermore, compared with a policy that ignores arm motions, our approach achieves a lower population standard deviation (PSD) of forward velocity. This indicates reduced body oscillations and improved dynamic stability and highlights the importance of whole-body coordination.

4.3.2. Disturbance rejection capability

The contribution of the structured trajectory generator was evaluated through external force perturbation experiments. As illustrated in Figure 8, disturbances were imposed on the robot torso in eight directions ($0^\circ - 360^\circ$ at 45° intervals) and applied for $0.1\,s$. A trial was considered successful if the robot did not fall, and the maximum resistible external force was evaluated across different methods, thereby demonstrating the role of the trajectory generator in robust running locomotion.

Figure 9 (top) shows snapshots where the robot recovers and maintains stability after external disturbances. Figure 9 (bottom) reports the time histories of forward body velocity, body pitch angle, left–hip joint position, and touchdown phase under a horizontal push of $290\,N$ applied at $t = 1\,s$. The model-based controller fails under this disturbance, indicating limited disturbance-rejection capability. In contrast, our method and the baseline exhibit comparable immediate responses in terms of forward velocity and body pitch. However, our method achieves faster recovery and requires fewer phase switches than the baseline, as shown in Figure 9 bottom (see also the supplementary video). This improvement can be attributed to the guidance provided by the joint reference reward r_{joint} during training. Unlike a single sinusoidal reference trajectory, the proposed structured kinematics-based trajectory generator encourages the robot to take larger steps to counter external perturbations, resulting in reactive behaviors more consistent with human locomotion patterns. Figure 8 (left) summarizes a more comprehensive

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

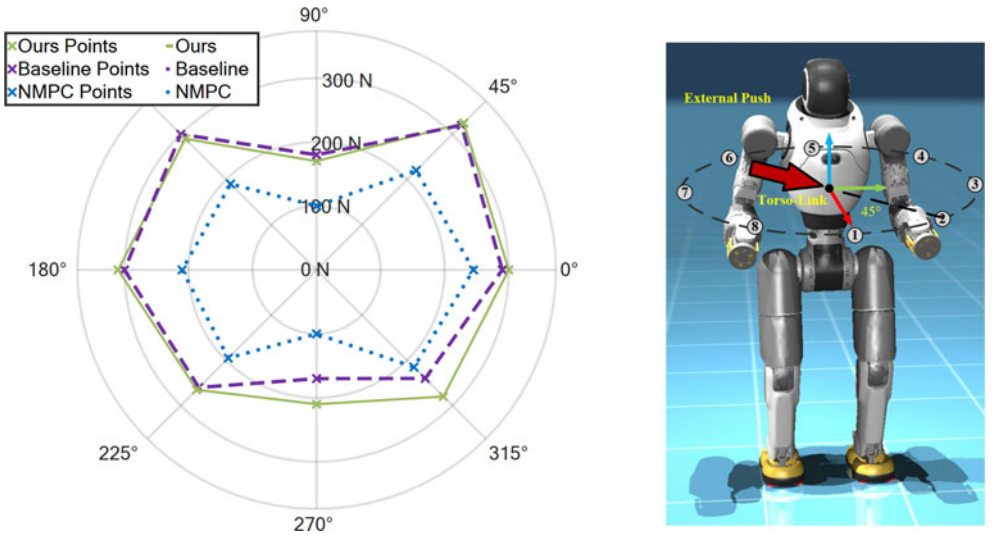


Figure 8. *External perturbation evaluation. The left figure shows the maximum external force that each method can resist, while the right figure depicts the directional distribution of perturbations applied during the experiments.*

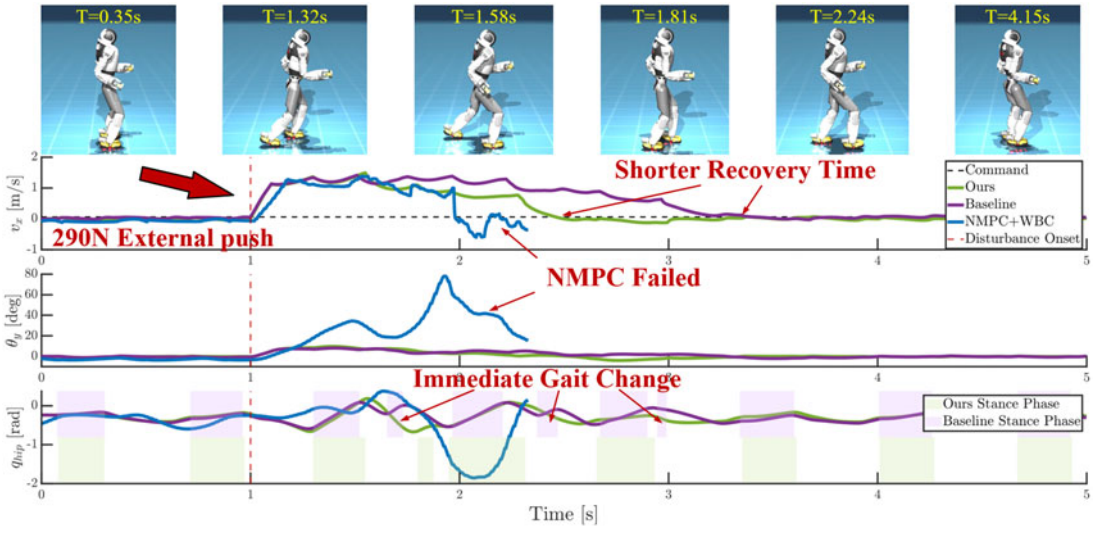


Figure 9. *Recovery behavior under external perturbations. Top: snapshots of the robot's recovery after push disturbances using our method. Bottom: responses of forward velocity, body pitch angle, left-hip joint position, and touchdown phase following the perturbation.*

disturbance–rejection evaluation. Compared with the baseline, our method tolerates larger perturbations for forward, backward, and oblique pushes, and it markedly outperforms the classical model-based controller.

4.3.3. Terrain adaptability

We further constructed continuous, complex terrains to evaluate the proposed method's terrain adaptability relative to the baselines. As illustrated in Figure 10 (top), the environment comprises flat ground, uphill, downhill (8°), uneven ground (random height in [0, 0.05], m), and transitional terrains.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

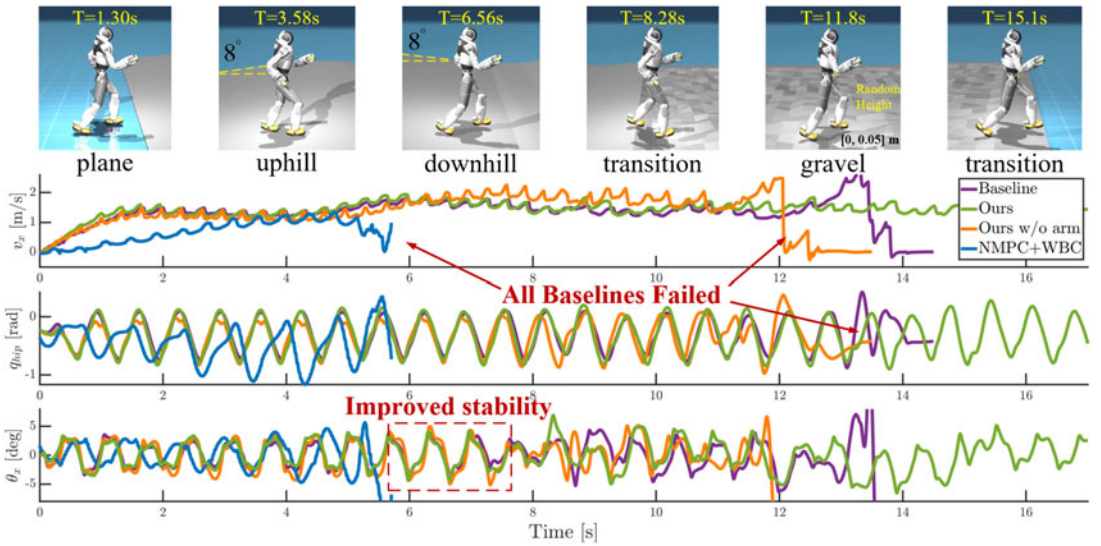


Figure 10. *Body behavior on complex terrains. Top: snapshots from the terrain-adaptation experiment using our method. Bottom: time histories of forward body velocity, left-hip joint position, and body roll angle during terrain traversal.*

The robot traverses the scene at a target speed of 1.3, m/s. Figure 10 (bottom) shows the time histories of forward velocity, left–hip pitch angle, and body roll angle.

Compared with the other baselines, only our method successfully traversed all terrains. On sloped terrain, RL-based methods exhibited comparable performance. However, on irregular terrain, our method achieved higher dynamic stability and better balance, particularly under uncertain ground height at touchdown (see also the supplementary video). An ablation removing arm coordination produced pronounced body-roll fluctuations and more frequent losses of balance, indicating that upper-body coordination is critical for stable dynamic locomotion. Additionally, in contrast to RL-based methods, the model-based controller performed poorly on slopes and irregular ground because it requires explicit terrain height inputs for landing point selection [1]. Notably, our policy was trained without a terrain curriculum yet remained robust across varying terrains, demonstrating superior adaptability and generalization.

4.4. Hardware results

Terrain setup: The physical experiments were conducted on flat indoor ground with a rigid surface.
Sources of disturbance: During the running trials, an attendant accompanied the robot via a safety tether. However, no external force was exerted on the robot through the tether; it was employed solely as a precautionary measure to safeguard against potential accidents.
Perception system configuration: In contrast to model-based controllers that rely on explicit state estimation, such as extended Kalman filtering, our policy uses only raw proprioceptive signals (joint angles, velocities, and IMU data) as input. This eliminates the need for additional estimation modules, reducing system complexity and improving robustness to modeling errors and sensor noise.

4.4.1. Velocity tracking performance

We evaluated the velocity tracking performance of different methods for running gait control on our real robot. The user specified the desired forward velocity using a handheld controller, and the robot's snapshots are shown in Figure 11.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press



Figure 11. *Snapshots of the robot executing running gaits with different methods in the real world.*

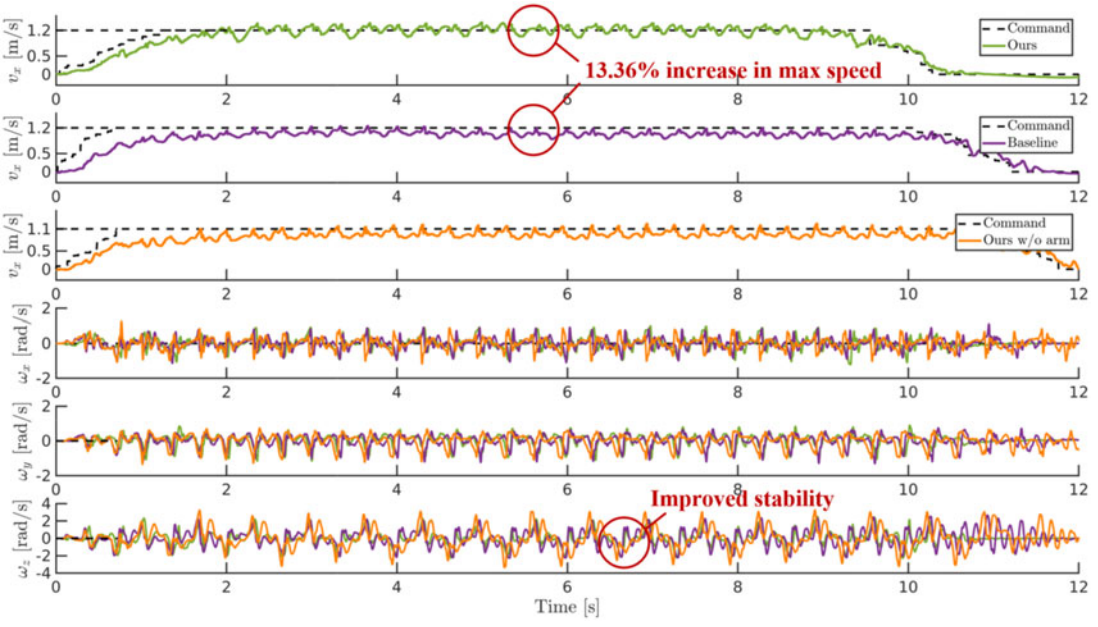


Figure 12. *Base velocity tracking results of different methods in the physical world.*

As further illustrated in Figure 12 and the accompanying videos, our method achieves superior running performance, characterized by a higher average speed and reduced velocity tracking error. As summarized in Table VII, it improves the maximum speed by 13.36% compared to the approach in ref. [22]. This improvement is attributed to the structured gait trajectories, which are generated based on the desired velocity and enable the exploration of more agile and dynamic running behaviors. We also compared the base velocity fluctuations using different methods. Although our method exhibits a slight increase in the standard deviation of the base's linear and angular velocities, which is likely due to the higher running speed, the video results show that the overall base stability remains comparable.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

Table VII. *Velocity tracking metrics for different gaits.*

	Forward [m/s]			Roll [rad/s]		Pitch [rad/s]		Yaw [rad/s]	
Method	**Avg.**	**RMSE**	**PSD**	**RMSE**	**PSD**	**RMSE**	**PSD**	**RMSE**	**PSD**
Baseline	1.0537	0.1617	**0.0725**	**0.3065**	**0.3063**	**0.3319**	**0.3315**	0.8788	0.8787
Ours w/o arm	0.9578	0.1662	0.0862	0.3269	0.3269	0.4038	0.4038	1.2542	1.2542
Ours	**1.1945**	**0.0895**	0.0894	0.3706	0.3706	0.4002	0.4001	**0.8737**	**0.8749**

The bold values indicate that these values outperform others.

Importantly, by incorporating coordinated arm motions, our method effectively reduces base posture fluctuations and enhances overall motion stability compared to cases without arm control. As shown at the bottom of Figures 11 and 12, the absence of arm assistance results in considerable yaw fluctuations, increasing the risk of instability and potential falls.

4.4.2. Periodic stability

Periodic stability is an important indicator of gait robustness. We evaluate it empirically by analyzing joint limit cycles in the running gait. Figure 13 shows the joint limit cycles obtained from real-world experiments under different methods. Although discrepancies between simulation and reality prevent the trajectories from perfectly matching their simulated counterparts, all methods still form relatively closed limit cycles, suggesting that stable periodic patterns are achieved. Our method demonstrates gait stability comparable to the baseline, and both clearly outperform the variant without arm coordination. The latter exhibits more pronounced oscillations in the joint limit cycles – particularly in the right knee – indicating reduced motion consistency and stability.

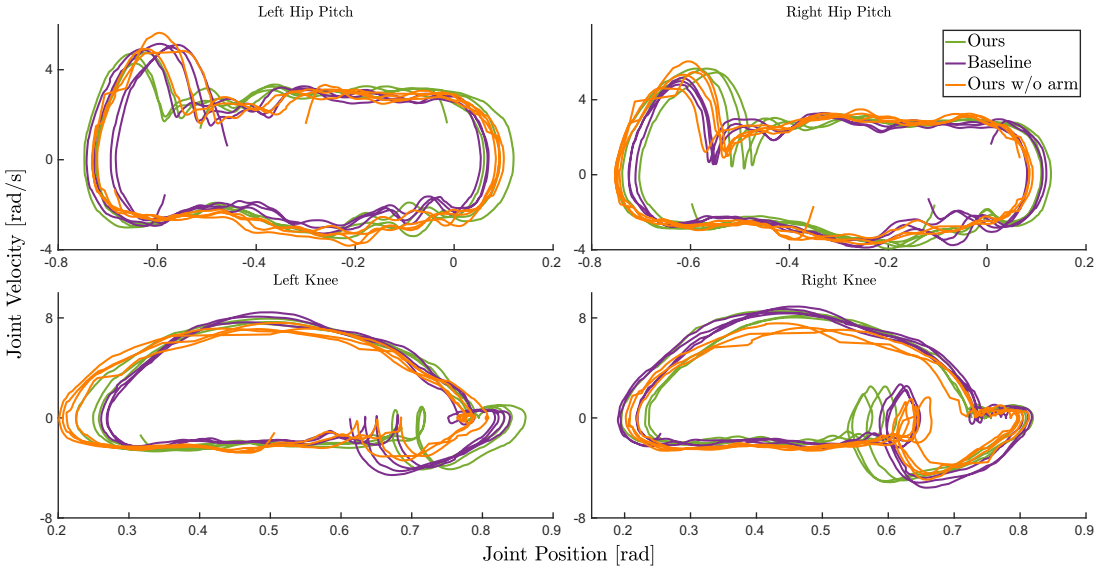


Figure 13. *Limit cycle of the learned policy of running gaits.*

4.5. Discussion

Lightweight trajectory generation: As evidenced by Tables VI, VII, and Figure 9, the proposed control architecture benefits from a kinematics-based gait and trajectory generator, yielding reduced velocity-tracking errors and enabling larger stride lengths both during running and under external disturbances.

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

Unlike prior work [6, 19], our method avoids complex model-based trajectory generators and large externally collected datasets. This simple yet effective generator is sufficient to guide the learning of robust and stable running gaits, thereby streamlining the RL pipeline and improving usability. However, unlike the method in ref. [20], our approach may not handle transitions between diverse gaits.

Terrain adaptability: Although our method is not explicitly trained on certain complex terrains, it still demonstrates strong adaptability and generalization to unseen scenarios, outperforming classical model-based methods that rely on detailed terrain information. Nevertheless, the approach may still fail in more challenging environments, for instance, stair climbing.

Whole-body coordination: Our method seamlessly integrates upper-limb motion within a unified control policy, thereby enhancing the dynamic stability and robustness of locomotion. Compared with [31], our method achieves similar humanoid arm motion without resorting to separate multi-agent training for the upper and lower limbs.

5. Conclusion and future work

In this paper, we propose a reinforcement learning framework for robust bipedal running. The framework incorporates a kinematics-based structured gait generator during training, which facilitates efficient policy learning and exploration. To bridge the sim-to-real gap and enable zero-shot transfer, we introduce two auxiliary networks within an asymmetric actor–critic architecture that infer latent environment state and body velocity from long- and short-horizon observation histories. Extensive experiments in simulation and on hardware demonstrate that our method significantly enhances running agility and achieves larger stride lengths, both in free running and under external disturbances. Moreover, the learned gait maintains periodic consistency on slopes and uneven terrain while naturally capturing the ground reaction force characteristics of spring-mass dynamics without explicitly modeling them.

Our future work will focus on a unified control framework that encompasses multiple locomotion gaits and supports seamless transitions among them. Developing a general strategy for diverse bipedal gaits remains challenging and significant.

Supplementary material. The supplementary material for this article can be found at http://doi.org/10.1017/S0263574725103007.

Author contributions. Conceptualization and methodology: Yunpeng Liang; Software: Yunpeng Liang and Zhihui Peng; Validation: Yunpeng Liang and Zhihui Peng; Writing – original draft preparation: Yunpeng Liang; Writing – review and editing: Yunpeng Liang and Zhihui Peng; Visualization: Yanzheng Zhao; Supervision: Weixin Yan; Project administration: Zhihui Peng and Weixin Yan; Funding acquisition: Zhihui Peng and Weixin Yan. All authors have read and approved the final version of the manuscript for publication.

Financial support. This work was supported by Shanghai AgiBot Technology Co. Ltd.

Competing interest. The authors declare no conflicts of interest exist.

Ethical approval. Not applicable.

References

[1] R. Grandia, F. Jenelten, S. Yang, F. Farshidian and M. Hutter, "Perceptive locomotion through nonlinear model-predictive control," *IEEE Trans. Robot.* **39**(5), 3402–3421 (2023).

[2] Y. Gong, R. Hartley, X. Da, A. Hereid, O. Harib, J.-K. Huang and J. Grizzle, "Feedback Control of a Cassie Bipedal Robot: Walking, Standing, and Riding a Segway," **In:** *2019 American Control Conference (ACC)* (IEEE, 2019) pp. 4559–4566.

[3] J. Siekmann, K. Green, J. Warila, A. Fern and J. Hurst, "Blind bipedal stair traversal via sim-to-real reinforcement learning. arXiv preprint arXiv: 2105.08328 (2021).

[4] J. Lee, J. Hwangbo, L. Wellhausen, V. Koltun and M. Hutter, "Learning quadrupedal locomotion over challenging terrain," *Sci. Robot.* **5**(47), eabc5986 (2020).

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press

[5] D. T. de Paula, E. P. Godoy and M. Becerra-Vargas, "Dynamic modeling and simulation of a torque-controlled spatial quadruped robot," *Robotica* **42**(8), 2761–2780 (2024).
[6] Z. Li, X. B. Peng, P. Abbeel, S. Levine, G. Berseth and K. Sreenath, "Reinforcement learning for versatile, dynamic, and robust bipedal locomotion control," *Int. J. Robot. Res.* **44**(5), 840–888 (2025).
[7] P. M. Wensing, M. Posa, Y. Hu, A. Escande, N. Mansard and A. D. Prete, "Optimization-based control for dynamic legged robots," *IEEE Trans. Robot.* **40**, 43–63 (2023).
[8] P. M. Wensing and D. E. Orin, "High-speed Humanoid Running Through Control with a 3D-Slip Model," **In:** *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems* (IEEE, 2013) pp. 5134–5140.
[9] S. Sovukluk, J. Englsberger and C. Ott, "Highly maneuverable humanoid running via 3D slip+ foot dynamics," *IEEE Robot. Autom. Lett.* **9**(2), 1131–1138 (2023).
[10] Y. Liang, F. Yin, Z. Li, Z. Xiong, Z. Peng, Y. Zhao and W. Yan, "Reduced-dimensional whole-body control based on model simplification for bipedal robots with parallel mechanisms," *IEEE Robot. Autom. Lett.* **10**(2), 1696–1703 (2025).
[11] W. Yang and M. Posa, "Impact-invariant control: Maximizing control authority during impacts," arXiv preprint arXiv: 2303.00817 (2023).
[12] K. Sreenath, H.-W. Park, I. Poulakakis and J. W. Grizzle, "Embedding active force control within the compliant hybrid zero dynamics to achieve stable, fast running on mabel," *Int. J. Robot. Res.* **32**(3), 324–345 (2013).
[13] A. Hereid, C. M. Hubicki, E. A. Cousineau and A. D. Ames, "Dynamic humanoid locomotion: A scalable formulation for hzd gait optimization," *IEEE Trans. Robot.* **34**(2), 370–387 (2018).
[14] J. Lee, M. Bjelonic, A. Reske, L. Wellhausen, T. Miki and M. Hutter, "Learning robust autonomous navigation and locomotion for wheeled-legged robots," *Sci. Robot.* **9**(89), eadi9641 (2024).
[15] X. B. Peng, P. Abbeel, S. Levine and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Trans. Graph. (TOG)* **37**(4), 1–14 (2018).
[16] Z. Xie, G. Berseth, P. Clary, J. Hurst and M. van de Panne, "Feedback Control for Cassie with Deep Reinforcement Learning," **In:** *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, vol. **1241** (2018) pp. 1246.
[17] Z. Xie, P. Clary, J. Dao, P. Morais, J. Hurst and M. Panne, "Learning Locomotion Skills for Cassie: Iterative Design and Sim-to-Real," **In:** *Conference on Robot Learning* (PMLR, 2020) pp. 317–329.
[18] K. Green, Y. Godse, J. Dao, R. L. Hatton, A. Fern and J. Hurst, "Learning spring mass locomotion: Guiding policies with a reduced-order model," *IEEE Robot. Autom. Lett.* **6**(2), 3926–3932 (2021).
[19] Z. Li, X. Cheng, X. B. Peng, P. Abbeel, S. Levine, G. Berseth and K. Sreenath, "Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots," **In:** *2021 IEEE International Conference on Robotics and Automation (ICRA)* (IEEE, 2021) pp. 2811–2817.
[20] 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)* (IEEE, 2021) pp. 7309–7315.
[21] W. Sun, B. Cao, L. Chen, Y. Su, Y. Liu, Z. Xie and H. Liu, "Learning perceptive humanoid locomotion over challenging terrain," arXiv preprint arXiv: 2503.00692 (2025).
[22] Q. Wu, C. Zhang and Y. Liu, "Custom Sine Waves are Enough for Imitation Learning of Bipedal Gaits with Different Styles," **In:** *2022 IEEE International Conference on Mechatronics and Automation (ICMA)* (2022) pp. 499–505.
[23] M. H. Raibert, *Legged Robots that Balance*, MIT Press (Cambridge, MA, 1986).
[24] G. Ji, J. Mun, H. Kim and J. Hwangbo, "Concurrent training of a control policy and a state estimator for dynamic and robust legged locomotion," *IEEE Robot. Autom. Lett.* **7**(2), 4630–4637 (2022).
[25] J. Hwangbo, J. Lee, A. Dosovitskiy, D. Bellicoso, V. Tsounis, V. Koltun and M. Hutter, "Learning agile and dynamic motor skills for legged robots," *Sci. Robot.* **4**(26), eaau5872 (2019).
[26] E. Todorov, T. Erez and Y. Tassa, "MuJoCo: A Physics Engine for Model-based Control," **In:** *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems* (2012) pp. 5026–5033.
[27] J. Schulman, F. Wolski, P. Dhariwal, A. Radford and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347 (2017).
[28] V. Makoviychuk, L. Wawrzyniak, Y. Guo, M. Lu, K. Storey, M. Macklin, D. Hoeller, N. Rudin, A. Allshire, A. Handa and G. State, "Isaac gym: High performance GPU-based physics simulation for robot learning," arXiv: 2108.10470 (2021).
[29] X. Gu, Y.-J. Wang and J. Chen, "Humanoid-gym: Reinforcement learning for humanoid robot with zero-shot sim2real transfer," arXiv preprint arXiv: 2404.05695 (2024).
[30] J.-P. Sleiman, F. Farshidian, M. V. Minniti and M. Hutter, "A unified MPC framework for whole-body dynamic locomotion and manipulation," *IEEE Robot. Autom. Lett.* **6**(3), 4688–4695 (2021).
[31] H. J. Lee, S. H. Jeon and S. Kim, "Learning humanoid arm motion via centroidal momentum regularized multi-agent reinforcement learning," arXiv preprint arXiv: 2507.04140 (2025).

Cite this article: Y. Liang, Z. Peng, Y. Zhao and W. Yan (2026). "Learning robust bipedal running via structured gait and trajectory guidance", Robotica **44**, 150–168. https://doi.org/10.1017/S0263574725103007

https://doi.org/10.1017/S0263574725103007 Published online by Cambridge University Press