AI-CPG: Adaptive Imitated Central Pattern Generators for Bipedal Locomotion Learned through Reinforced Reflex Neural Networks

Guanda Li[1], Auke Ijspeert[2], and Mitsuhiro Hayashibe[1]

***Abstract*—Humans have many redundancies in their bodies and can make effective use of them to adapt to changes in the environment while walking. They can also vary their walking speed in a wide range. Human-like walking in simulation or by robots can be achieved through imitation learning. However, the walking speed is typically limited to a scale similar to the examples used for imitation. Achieving efficient and adaptable locomotion controllers for a wide range from walking to running is quite challenging. We propose a novel approach named adaptive imitated central pattern generators (AI-CPG) that combines central pattern generators (CPGs) and deep reinforcement learning (DRL) to enhance humanoid locomotion. Our method involves training a CPG-like controller through imitation learning, generating rhythmic feedforward activity patterns. DRL is not used for CPG parameter tuning; instead, it is applied in forming a reflex neural network, which can adjust feedforward patterns based on sensory feedback, enabling the stable body balancing to adapt to environmental or target velocity changes. Experiments with a 28-degree-of-freedom humanoid in a simulated environment demonstrated that our approach outperformed existing methods in terms of adaptability, balancing ability, and energy efficiency even for uneven surfaces. This study contributes to develop versatile humanoid locomotions in diverse environments.**

***Index Terms*—Bioinspired robot learning, legged robots, machine learning for robot control.**

I. Introduction

Humanoid robots have been a topic of interest for researchers because of their potential to revolutionize various fields, such as healthcare, industry, and entertainment [1]. These robots are designed to mimic human behavior, movement, and communication, making them more approachable and relatable to humans [2]. Humanoid robots can walk on both legs similar to humans, in a feature known as bipedal locomotion. Therefore the humanoid robots can navigate and interact with environments designed for humans, making them more versatile and useful for various applications [3], [4].

Despite the potential advantages of humanoid robots with bipedal locomotion, adaptively controlling humanoid robots is challenging because of their complex dynamics and overly redundant degrees of freedom (DoFs) [4]. One of the current issues that should be addressed is related to the gait coordination of humanoid robots, which can significantly affect their overall performance. Some existing humanoid robots, including advanced models like Atlas from Boston Dynamics, have made significant progress, yet there is still room for improvement to enhance their adaptability to complex and unknown environments. Furthermore, current humanoid robots often struggle to adapt to changes in their surroundings, which makes them less efficient and effective [5], [6]. To overcome these challenges, the gait of humanoid robots should be improved with a focus on enhancing energy efficiency, increasing flexibility, and improving adaptability to complex environments through learning.

Deep reinforcement learning (DRL), a machine learning algorithm has gained significant attention in recent years owing to its potential to solve complex problems in various fields, including robotics [7], [8]. DRL involves training an agent to learn the optimal behavior through trial-and-error interactions with its environment, using a reward signal to guide its actions. In robotics, DRL has been used to improve the performance of various tasks, such as grasping [9], locomotion [10], especially on the quadrupedal robots [11], [12]. However, one of the current obstacles in applying DRL to humanoid robots is the large dimensional space that should be explored and the imbalance of biped locomotion. This makes it challenging to learn a desirable gait directly because there are significantly many possible combinations of movements to consider and many lead to falls. Currently, the application of DRL in humanoid locomotion necessitates intricate reward functions and high computational costs [13], or a reduction in the robot's DoFs [14]. Therefore, new methods and techniques to address this challenge and enable DRL to effectively control humanoids by managing high dimensionality, are desirable.

Inspired by neuroscience, central pattern generators (CPGs) are another promising approach for improving legged robots locomotion [15]. CPGs are neural circuits located in the spinal cord that generate rhythmic patterns of muscle activity, such as those used during walking and running [16]. Using CPGs, robots can achieve more natural and stable movements, similar to those of living organisms [17]–[19]. The CPG mechanism involves a network of interconnected neurons that generate oscillatory signals that are transmitted to the muscles responsible for movement. In animals, the reflex circuit usually works together with CPGs as a feedback control [20]. Computational

Manuscript received: November, 25, 2023; Revised February, 22, 2024; Accepted April, 3, 2024. This paper was recommended for publication by Editor Abderrahmane Kheddar upon evaluation of the Associate Editor and Reviewers' comments. This work was funded by JSPS 20KK0256 (Fostering Joint International Research A) and 22H04764 (Hyper-adaptability project). (Corresponding authors: Guanda Li and Mitsuhiro Hayashibe.)

[1] G. Li and M. Hayashibe are with the Neuro-Robotics Lab, Department of Robotics, Graduate School of Engineering, Tohoku University, Sendai, Japan li.guanda.s5@dc.tohoku.ac.jp

[2] A. Ijspeert is with the Biorobotics Laboratory, the École polytechnique fédérale de Lausanne (EPFL), Lausanne, Switzerland

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

models were used to investigate the merging of CPGs with sensory feedback [21]–[23]. However, the question of how to effectively integrate and apply them to humanoid locomotion control remains unresolved [20], since CPGs can potentially constrain the control space and help decrease dimensionality but to be adaptive and flexible for different environments, it should be well supported by reflex networks.

Our study aims to enhance learning-based algorithms for humanoid locomotion using CPGs with a sensory feedback mechanism. We trained a CPG controller using imitation learning and then trained a reflex neural network using DRL. Unlike other algorithms that use reinforcement learning for imitation purposes [24], [25], our training objective was not only to make the agent behave similarly to the collected human motion data; we used imitation learning to train a CPG-like controller to form feedforward control. The CPG-like controller was designed to generate rhythmic patterns of joint torques, similar to those generated by CPGs in living organisms. We use imitation learning for training pattern formation of CPG to avoid the complex calculations and tuning required by other nonlinear functions, such as Hopf and Matsuoka Oscillators [26]. The reflex neural network was then trained with DRL to adjust the movements generated by the CPG-like network based on sensory feedback, allowing the robot to adapt to changes in the environment. Regarding the combination of CPG and RL, CPG-RL was recently proposed for learning and modulating oscillator parameters of CPG [18]. In this research, reinforcement learning is used for forming a reflex neural network to support CPG rather than forming CPG itself.

The contribution of this study is that we propose a new learning-based control framework for the locomotion task of a legged system inspired by CPG with a reinforced reflex neural circuit mechanism, without reducing the robot's DoFs. Our method employs a CPG as a feedforward controller, which is trained by imitation learning, and another reflex neural network as a feedback controller, which is trained using DRL. Then we verify the performance of the proposed framework, which can adapt to environmental changes, and demonstrate its performance through a comparative study with existing learning methods. Our approach is based on bio-inspired mechanisms, which help us better understand the potential mechanisms of human locomotion and develop more sophisticated and versatile humanoid robots with improved locomotion capabilities in a variety of environments.

II. Method

A. CPG-Learning

Our control framework, adaptive imitated CPG (AI-CPG), consists of rhythm generator $\mathbb{G}$, which defines the rhythm of motor activities; pattern formation layer $\mathbb{S}$, which shapes the rhythmic timing signals to the target joint angles of the robot; PD controller, which outputs the motor commands based on the error between current joint angles and target joint angles; and reflex neural network controller $\mathbb{R}$ based on sensory feedback, as shown in Fig. 1 (b).

During the process of controlling the robot locomotion, the speed command in Fig. 1 (b) modulates the robot's speed by altering the frequency of $\mathbb{G}$ and $\mathbb{S}$. This corresponds to a similar mechanism in Fig. 1 (a), where the brain adjusts the human motor pattern by descending modulation to the spinal network. Previous research has shown how descending modulation adjusts the activity of the CPG [23], collaborates with a sensory-driven model [27], [28], and facilitates walk-run transitions [29]. $\mathbb{G}$ and $\mathbb{S}$ served as feedforward CPG controllers that reduce the dimensionality of the action space of the robot using prior knowledge. Contrarily, $\mathbb{R}$ serves as a feedback controller responsible for maintaining the balance of the robot and adapting to the given physical environment.

1) CPG controller: The rhythm generator $\mathbb{G}$ is defined by

$$\mathbb{G}(\mathbf{T_k}) = \sin(2\pi f \mathbf{T_k}), \tag{1}$$

$$\mathbf{T_k} = [t_k,\ t_{k+1},\ t_{k+2},\ \cdots,\ t_{k+i}]. \tag{2}$$

The input of $\mathbb{G}$ is a set of sine wave phase oscillators starting from different timesteps. i is the number of phase oscillators and f is the adjustable frequency. The output of $\mathbb{G}$ is called the fundamental timing signal $\boldsymbol{w}$ and is the input to $\mathbb{S}$. The output $\boldsymbol{\theta_t}$ of $\mathbb{S}(\boldsymbol{w})$ is the target angles of the robot joints.

As shown in Fig. 2 (a) and (b), We trained $\mathbb{S}$ through imitation learning using human motion data from the CMU motion capture database [30], which consisted of a set of gait data both for walking and running. We used Fast Fourier Transform (FFT) to obtain the motion frequencies f_w and f_r for the two sets of data. Based on the motion frequencies and $\mathbb{G}$, we calculated the input features used for training. After mapping the input features to the real motion data in the time series, we obtained a training dataset that was used to train $\mathbb{S}$ by supervised learning. By varying the frequency f of the input sine wave signal to $\mathbb{S}(t, f)$, we could generate the joint angles and torques of the humanoid robot corresponding to different movement speeds.

In our study, $\mathbb{S}$ was a multilayer perceptron (MLP) with an input layer of size 50 (the same as the value of i in Equation 2), a hidden layer of size [128, 128], and an output layer of size 28. The activation function is ReLU. Using a PD controller, we obtained the joint torque $\boldsymbol{\tau_g}$ of the robot from the target joint angles $\boldsymbol{\theta_t}(t)$ and actual joint angles $\boldsymbol{\theta}(t)$.

$$\boldsymbol{e}(t) = \boldsymbol{\theta_t}(t) - \boldsymbol{\theta}(t), \tag{3}$$

$$\boldsymbol{\tau_g}(t) = K_p \boldsymbol{e}(t) + K_d \frac{d}{dt}\boldsymbol{e}(t), \tag{4}$$

where for the three joints on the torso K_p is 750, and for the remaining joints K_p is 250. K_d for all joints is 1.

2) Reflex neural network: The reflex neural network controller was trained using proximal policy optimization (PPO) [31]. During the training process, $\mathbb{S}$ produced rhythmic control signals to the robot. We used 8,192 agents in parallel to interact with the environment and collect data. Each agent was assigned a different frequency f such that $\mathbb{R}$ could simultaneously learn how to keep the humanoid stable under the influence of $\mathbb{S}$ at different motion frequencies.

After the training, changing the input frequency f in both $\mathbb{G}(t, f)$ and $\mathbb{R}(t, f)$ enabled the humanoid to move at different speeds and both for walking and running.

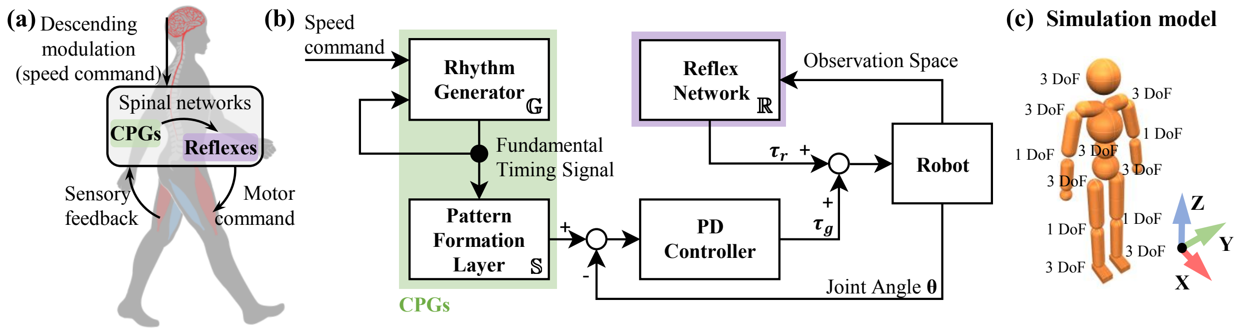

Fig. 1. (a) Schematic of the central pattern generator mechanism in human locomotion. CPG is designed to combine motor rhythm with sensory feedback to achieve a bipedal gait. (b) The control framework of our study comprises feedforward and feedback controllers. The feedforward controller is the generative shaping network output of the joint angles θ of the robot, while the PD controller calculates the output torque $\boldsymbol{\tau_g}$ based on the input target angles $\boldsymbol{\theta}$. The feedback controller consists of a reflex neural network trained by DRL, which takes environmental information as the input and outputs the joint torque $\boldsymbol{\tau_r}$. (c) Humanoid model in the simulation environment with DoFs distribution.

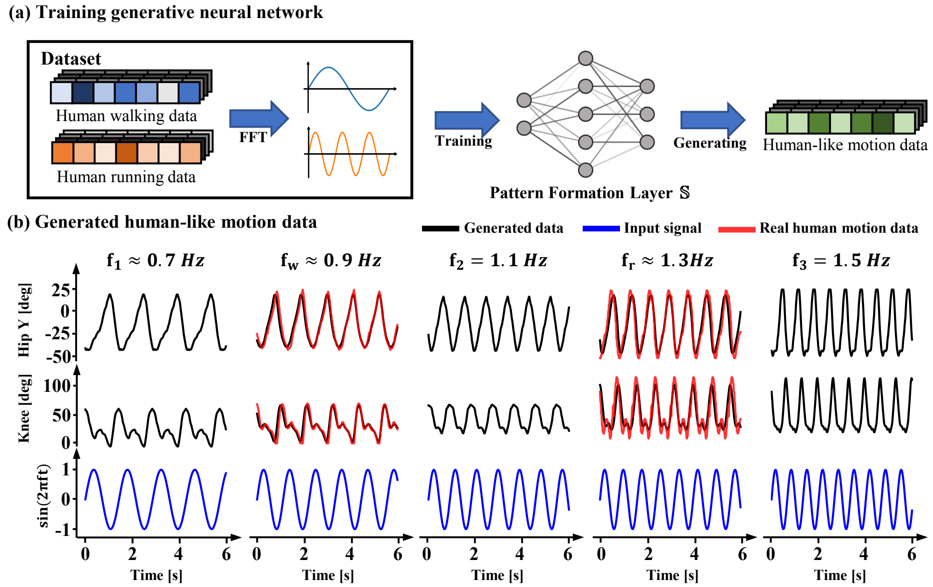

Fig. 2. Training approach of a pattern formation layer. (a) Dataset for training generative shaping network with fixed-length segments of sine signals as input features and motion data obtained from human subjects as the corresponding output labels. The frequency of the sine signals was calculated using FFT from the real motion data. The generative model was trained using supervised learning. (b) Generating different motion data by varying the input sine signal frequency from the generative shaping network within six seconds. Owing to the periodicity of the input sine signal, the generated motion data exhibited basic rhythmic activity patterns similar to the neural signals outputted by the CPG. f_w and f_r are the frequencies of the real human walking and running data.

The reward function R for training $\mathbb{R}$ is

$$R = R_h - \alpha R_e + R_a + R_s + R_d + \beta R_g. \tag{5}$$

R_h is the height of the humanoid head, which helps the robot learn to stand. R_e is used to limit the energy consumption of $\mathbb{R}$ with an α value of 0.5.

$$R_e = \sum_J |\tau_j(t)\omega_j(t)| + \tau_j^2(t). \tag{6}$$

If the robot falls, we set the total reward R to zero and reset the environment. Contrarily, if the robot does not fall, it gets an accumulating survival reward $R_a = 1$ in each timestep. R_s is equal to $\boldsymbol{z}_{global} \cdot \boldsymbol{z}_{pelvis}$, which is used to optimize the orientation of the pelvis and improve the robot's balance. R_s equals 1 when the pelvis's z-axis is perpendicular to the ground, and R_s equals 0 when it is parallel to the ground. R_d is used to teach robots to move in a target direction and is equivalent to the velocity of the robot in the target direction. The higher the velocity towards the target, the greater the reward R_d received by the robot. Note that our reward function does not directly specify the movement speed of the robot. The robot can move at different speeds is only influenced by changes in the input frequency f. R_g is used to reduce $\boldsymbol{e}(t)$ in Eq. 3, which is

$$e = \left| \sum_J \boldsymbol{e}(t) \right| \tag{7}$$

$$R_g = \begin{cases} 1 - e/b, e \leq b \\ 0, e > b \end{cases}, \tag{8}$$

where b is 2.5, and β is 5.0 in Eq. 5.

Compared with other studies that use DRL to train humanoid robots [13], [14], our reward function does not include terms related to tracking velocity and motion trajectory.

$\mathbb{R}$ is an MLP with an input layer of size 192 corresponding to the size of the observation space. The observation space includes the robot's joint angles, angular velocities, foot pressures, spatial orientation, target angles $\boldsymbol{\theta_t}(t)$, and output torques $\boldsymbol{\tau_g}$ of the CPG controller. The output layer has a size of 28, which is the same as the number of joints in the robot. The output of $\mathbb{R}$ is $\boldsymbol{\tau_r}$. The hidden layer sizes are [512, 256, 128], and the activation function is ReLU. The joint torque $\boldsymbol{\tau}$ applied to the robot is the sum of $\boldsymbol{\tau_g}$ and $\boldsymbol{\tau_r}$.

B. Simulation Method

The simulation software used in our study is Isaac Gym, which stores all computational data as tensors in the graphics processing unit (GPU), enabling the DRL algorithm to collect data for training from thousands of actors through a parallel training framework [32]. This approach takes full advantage of the computing power of the GPU and eliminates the need to transfer data from the central processing unit (CPU) to the GPU during simulation and training, significantly improving the training speed of DRL.

We used a humanoid robot with a height of 1.6 m and weight of 48.9 kg, 28 DoFs, and 13 links throughout its body, as shown in Fig. 1 (c), for both of training and evaluating the performance of the proposed approach. First, we compared our method with other learning methods for controlling the humanoid gait. Second, we performed a transition task from walking to running and analyzed the changes in gait. Third, we trained and tested the performance of the humanoid on uneven terrain, where the robot has to adjust its gait to maintain postural balance.

C. Evaluation Index

The following indexes are used to evaluate the performance of the humanoid robot after training:

1) Symmetry Index: Symmetric gait is considered normal in human walking; therefore, we used the symmetry index to determine the similarity of a robot's gait with that of a human. The symmetry index refers to the extent to which movement patterns are similar between the left and right sides of the body and is calculated by dividing the difference between the left and right parameters by the sum of the left and right parameters. A typical equation [33] used to calculate the symmetry index is

$$SI = \frac{(X_R - X_L)}{0.5\,(X_R + X_L)}, \tag{9}$$

where X_R and X_L could be the angles, angular velocities, or torques of the joints produced by the left and right limbs. A value of SI close to zero indicates a symmetric gait, whereas far from zero indicates an asymmetric gait.

In our study, the equation used to calculate the symmetry index is

$$SI = \frac{1}{T \cdot J} \sum_{t=0}^{T} \sum_{J} \frac{|X_R(t,j)) - X_L(t,j)|}{0.5\,(X_R(t,j) + X_L(t,j))}, \tag{10}$$

where T is the number of timesteps in one test trial, J is the joint number of the robot, and $X_R(t,j)$ and $X_L(t,j)$ are the joint angles of the limbs on the left and right sides of the robot, respectively.

2) Froude Number: The Froude number (Fr) is a dimensionless quantity used to determine whether a person is walking, running, or performing other forms of locomotion [34]. Humans tend to transition from walking to running at Fr between 0.4 and 0.6, with individual variations depending on factors, such as age, fitness level, and body proportion [35].

Fr is used to determine the gait pattern of the robot. When the Froude number is closer to zero, the gait of the robot tends to be more stable and suitable for slow movements. Conversely, when the Froude number is large, the gait tends to be more dynamic and unstable and is suitable for fast movement. When the Froude number was approximately 0.5, the gait was in a transitional state. Fr is expressed as

$$Fr = \frac{v^2}{gL}, \tag{11}$$

where v is the characteristic velocity, g is the gravitational acceleration, and L is the characteristic length. In this study, v is the average velocity of the center of mass of the robot on the x-axis, g was 9.81 m/s^2, and L is the total leg length of the humanoid robot (0.855 m).

3) Average Velocity and Cost of Transport: We employed the following equations to compute the average velocity v and cost of transport (CoT) of the robot during movement:

$$CoT = \frac{p}{mgv} = \frac{\sum_{t=0}^{T} \sum_{J} |\tau(t,j)\omega(t,j)|}{mg \sum_{t=0}^{T} v_t(t)} \tag{12}$$

where v_t represents the velocity of the robot's CoM in the target direction, $\boldsymbol{\tau}$ is the torque of the robot's joints, $\boldsymbol{\omega}$ is the angular velocity of the robot's joints, T is the number of time steps, J is the number of robot joints, m is the mass of the robot, and g is 9.81 m/s^2.

4) Balance: We assessed the robot's balance in motion by measuring the offset between the pelvis (lower part of the trunk) and the ground on the z-axis directional vectors. The equation to calculate the balance index BI is

$$BI = \frac{1}{T} \sum_{0}^{T} \boldsymbol{z}_{global} \cdot \boldsymbol{z}_{pelvis}, \tag{13}$$

where $\boldsymbol{z}_{global}$ and $\boldsymbol{z}_{pelvis}$ are the direction vectors of the ground and robot torso on the z-axis, respectively. A value of BI close to one indicates better balancing ability.

III. Results

A. Comparison

To showcase the features of our new control framework, we trained a humanoid agent to perform the task of moving along a straight line at different speeds on flat ground. We used two algorithms, PPO [31] and Adversarial Motion Priors (AMP) [25] for comparison. PPO is one of the most commonly used DRL methods in robotics. Its advantage is to handle high dimensional and continuous state and action spaces with stable training performance. AMP is a new algorithm for animation generation and robot control that combines imitation learning, adversarial learning, and DRL. It is an efficient method for imitating natural and lifelike behaviors from real motion data without requiring the artifical design of reward functions.

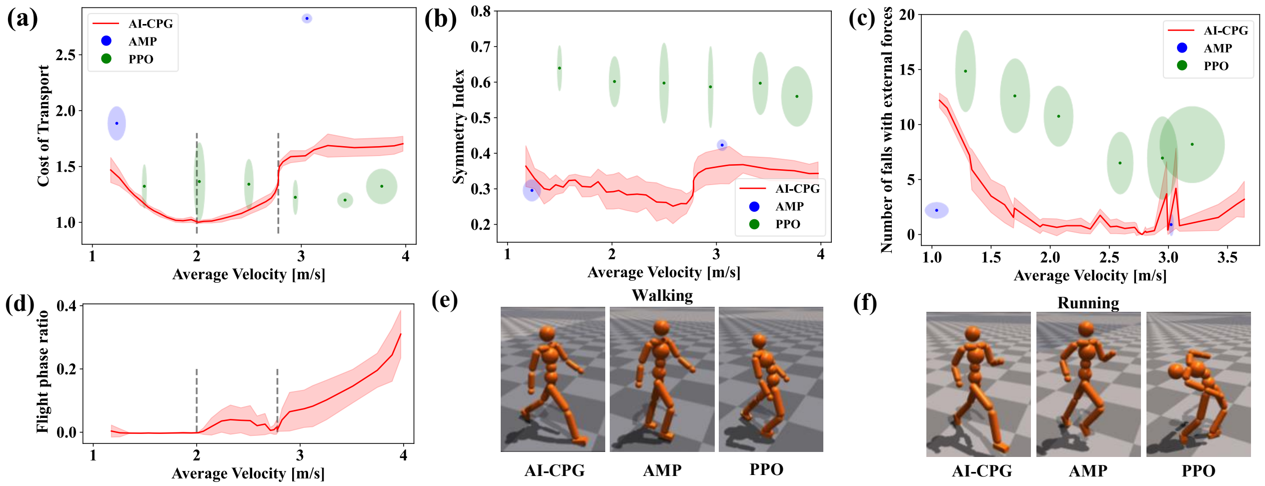

Fig. 3. Performance comparison of humanoid agents trained using AI-CPG, AMP, and PPO. (a) Cost of transport at different speeds. (b) Symmetry indices at various speeds. (c) Number of the robot falls within one minute under random disturbances. Red lines indicate the results of AI-CPG, and blue and green dots indicate the results of AMP and PPO. The red shaded area represents the standard deviation of the corresponding points; the blue and green shaded areas represent the standard deviation of corresponding points on the x- and y-axises. (d) The variation in the flight phase ratio of AI-CPG result at different speeds. The black dashed line indicates the points of gait transition. The stable walking gait ends at 2.0 m/s, and the stable running gait begins at 2.8 m/s. (e) and (f) Motion examples of three algorithms at walking and running. The motion pattern results could be referred at the accompanying video of the paper.

To ensure that the experimental conditions were as similar as possible during training, the number of epochs for each of the three algorithms was 3,000, with each epoch lasting for 1,000 iterations. The number of actors trained in parallel was 8,192 and the neural networks used were MLP with hidden layers of sizes [512, 256, 128] for all three algorithms. The two sets of real-motion data used to train the AMP were identical to those used to train the AI-CPG.

The reward function of the PPO algorithm is similar to AI-CPG. The only difference is we use R_t instead of R_g in the reward function (see Eq. 5). R_t is used to set the target velocity for the humanoid robot. For the AMP algorithm, we do not explicitly set a reward function. The style reward in AMP is derived through an automatic learning process from a dataset of reference motion clips. The observation space of AMP and PPO is similar to that of AI-CPG but does not contain the information in the CPG controller part of AI-CPG.

For the PPO algorithm, we set the target velocities in the reward function to 1.5, 2.0, 2.5, 3.0, 3.5, and 4.0 [m/s]. The AMP algorithm learns two different speeds based on the walking and running gaits of human motion data it uses. For AI-CPG, the range of input frequency of the CPG generator is [0.8, 1.4] for training and [0.7, 1.4] for testing. After every 100 training epochs, the neural network was saved as a checkpoint. We selected the checkpoints with the highest average velocity in each round of training as the convergence results after training. We trained each algorithm with five random seeds, and the results are shown in Fig. 3.

By comparing the CoT at different average velocities in Fig. 3 (a), we can notice that the agent trained with the AI-CPG could adjust its movement speed using only one neural network controller even for a wide range of speeds. In addition, the U-shaped CoT-velocity relationship in walking and the linear CoT-v relationship in running is very similar to the actual human case relationship [36]. However, the movement speed of the robot trained using the PPO algorithm was limited to the design of its reward function. The AMP algorithm focuses excessively on making the robot's movements similar to the real motion data, which makes it hard to flexibly adjust the robot's movement speed. Additionally, compared to the AMP algorithm, the PPO and AI-CPG algorithms optimized the energy efficiency of the robot during movement by following the energy consideration at the reward function.

In Fig. 3 (b), we compared the symmetry index of the robot at different speeds, and in Fig. 3 (e) and (f), we visually display the gait of the robot moving in the simulator. We found that because the PPO algorithm did not reference any real motion data during training, its gait symmetry was far worse (i.e. with higher values) than that of the AMP and AI-CPG. These abnormal gaits limit the application of the PPO method in real-world robots.

As shown in Fig. 3 (c), we verified the robustness of the gait controller by applying a certain high random disturbance force from -200 N to 200 N for all axes to the torso of the trained agent during gait. The external force was applied for a duration of 0.1 s for every second. The results are the averages of the last five checkpoint test results for each random seed training. During the 60-second test, it was observed that for robots controlled by the same algorithm, the low-speed motion state is more unstable than the high-speed state. Moreover, the AI-CPG and AMP experienced fewer instances of falls compared with the PPO over most of the speed ranges. This suggests that the human-like gait, which AMP and AI-CPG learn from human motion data, is more robust against external disturbances compared to the abnormal gait of PPO.

In Fig. 3 (d), we use the ”flight phase ratio” to determine the agent's gait. The flight phase refers to the phase in which both feet are not in contact with the ground during one complete gait cycle. The ”flight phase ratio” indicates the proportion of time during the gait cycles when it is in the flight phases. According to the flight phase ratio, the moving speed controlled by AI-CPG is divided into three periods by

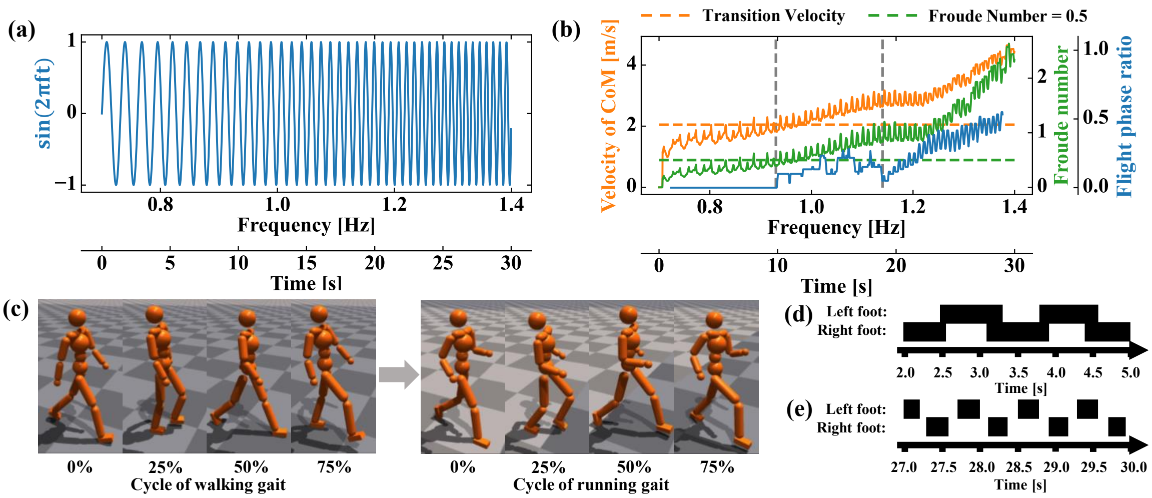

Fig. 4. Analysis of the transition process from walking to running gait in humanoid robot trained and controlled using the AI-CPG method. (a) Sinusoidal signals with increasing frequency were used as inputs to the rhythm generator of the CPG controller. The x-axis represents the frequency increase over time and the y-axis represents the amplitude of the input signal. The total duration was 30 seconds, and the rate of frequency increase was 0.023 Hz/s. (b) Correspondence between robot center of mass velocity, Froude number, flight phase ratio, input frequency of AI-CPG, and motion time. The transition velocity was 2.05 m/s when the Froude number was 0.5. The black dashed line indicates where the gait of the robot changes. When the frequency is less than 0.93 Hz, the robot has a stable walking gait, and when the frequency is greater than 1.14 Hz, the robot has a stable running gait. (c) Transition schematic from walking to running gait. (d) Time diagram of walking gait cycles from 2.0 to 5.0 s. (e) Time diagram of running gait cycles from 27.0 to 30.0 s.

the black dashed line in Fig. 3 (a) and (d). The leftmost period represents the stable walking gait without a flight phase. The middle range represents a transition gait where the flight phase fluctuates. The rightmost period represents the stable running gait, where the float phase ratio is greater than zero and steadily increases with speed. The result confirms that AI-CPG lets the neural network learn different gaits including transitions up to running successfully.

Based on a comparison of the three methods above, we can conclude that AI-CPG combines the advantages of imitation learning and DRL. Through the shaping of the reward function, DRL assists the robot in learning to maintain balance and optimize energy efficiency. The human-like gait learned from human motion data enables AI-CPG to better resist external disturbances. Additionally, the feedforward control mechanism in the CPG part of AI-CPG enables it to consistently handle a wide range of moving speeds and different gaits.

B. Transition from Walking to Running

By adjusting the value of f at the checkpoint trained in Section III-A, we achieved a smooth transition in the robot's gait from walking to running. The relationship between f and time t is $f(t) = 0.7 + 0.023t$. As shown in Fig. 4 (a), an increase in f causes the input sine signal to gradually become denser in the AI-CPG, leading to an adjustable dynamic gait. Meanwhile, in Fig. 4 (b), we observed the gait transition in the different stages of the result. A time window of 60 time steps (1 second) is used to calculate the flight phase ratio. When both t and f are small, the robot moves slowly in a walking gait. The flight phase ratio, Fr, and the gait diagram in Fig. 4 (d) confirm this observation. As t and f increased, the speed of the robot also increased, and the flight phase ratio gradually increased and fluctuated, indicating a transition gait of the robot. When the frequency is greater than 1.14 Hz, The robot transitioned to a stable running gait with further increases in t and f, as shown in Fig. 4 (e).

One advantage of the AI-CPG that can be observed from this result is its ability to adjust the speed of the robot during its movement simply with tonic input through f. This special feature enables the humanoid to operate more efficiently and effectively in real-world applications, making it a versatile and flexible solution for various scenarios.

C. Locomotion on Uneven Terrain

We retrained and tested the locomotion task of the humanoid on an uneven terrain using the PPO, AI-CPG, and AMP algorithms with the same training parameters as those described in Section III-A. The uneven terrain consisted of a triangular mesh and exhibited a height variation range of 10 cm. Fig. 5 (a) shows the changes in the motion trajectory of the robot as the training iterations increased at different target velocities. The results indicate that PPO and AI-CPG were successful in controlling the movement of the robot on uneven terrain, whereas AMP failed to learn the task.

In the early stages of training, both the PPO and AI-CPG had a disordered velocity vector (indicated by the dark-colored arrow in the figure) that clustered around the origin, making the agent unable to move effectively. As the number of training iterations increased, the velocity vector gradually aligned with the positive x-axis and shifted toward the right, indicating that the agent learned to move in the desired direction. Conversely, AMP's trajectory and velocity vectors were always disorganized and haphazard.

In Fig. 5 (b), we compared the experimental results of PPO and AI-CPG and found that AI-CPG outperformed PPO in terms of the symmetry index, balance index, and cost of transport at two different moving speeds. It is important to note that the AI-CPG case is with the same neural network both

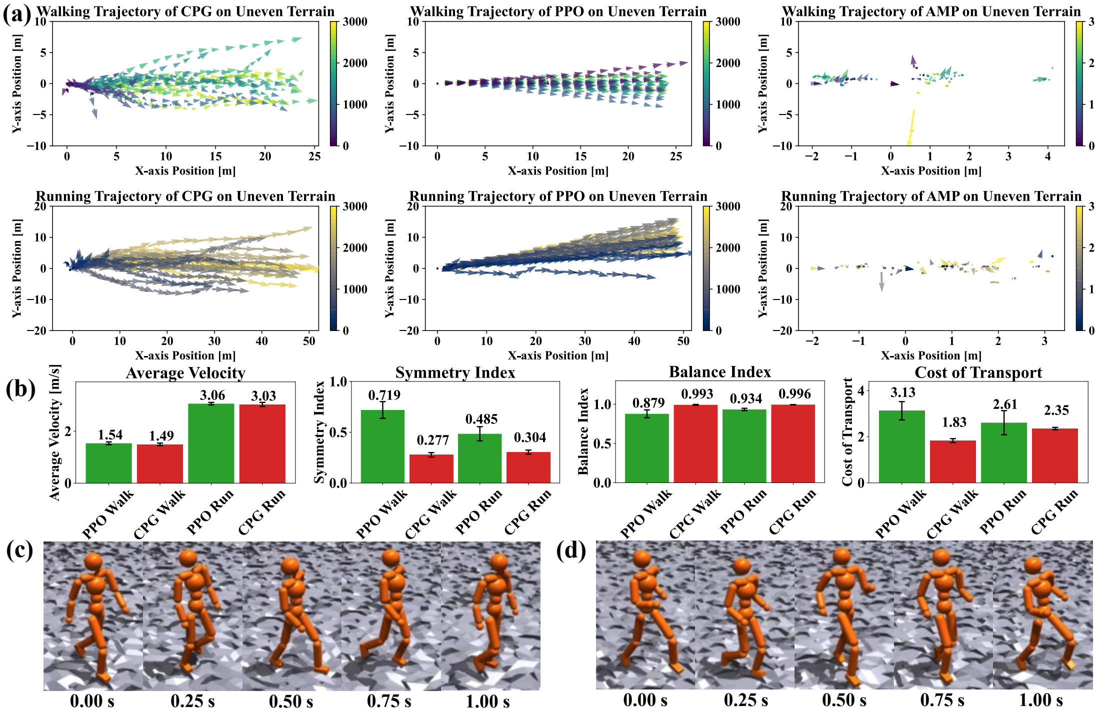

Fig. 5. Training and testing results of humanoid locomotion task on uneven terrain. (a) Changes in the motion trajectory of the agent as the training epoch progresses at walking speed (1.5 m/s) and running speed (3.0 m/s) for the three algorithms. The position of the arrow corresponds to the position of the robot, and the direction and magnitude of the arrow represent the direction and magnitude of the velocity vector of the robot, respectively. The color bar represents the number of epochs for which the robot was trained. Lighter arrows indicate an increased number of training epochs. (b) Bar chart used to compare the mean and standard deviation of the velocity, symmetry index, balance index, and cost of transport of the PPO and AI-CPG methods for walking and running. (c)-(d) Agent moving on uneven terrain using (c) walking gait and (d) running gait controlled by the same AI-CPG controller.

for walking and running, whereas PPO needs a different neural network for walking and running, respectively. Furthermore, the standard deviation of the AI-CPG results was smaller, indicating a more stable learning process.

Finally, transition from walking to running was tested on uneven terrain, which is quite challenging task. Similar to the flat surface result described in Section III-B, the AI-CPG could manage to implement a speed transition from walking to running on uneven terrain, as shown in Fig. 6.

IV. Discussion

In this study, we propose a learning framework that combines a generative imitative neural network and a reinforced reflex neural network controller to achieve stable, energy-efficient, and natural control of humanoid bipedal locomotion. The proposed framework can control the motion speed both for walking and running, and the direction of the robot and can adapt to different environments and terrains, as demonstrated by the successful locomotion of the humanoid under different gait patterns and speeds. The generative neural network generated periodic control signals based on real human motion data, making humanoid locomotion more natural and intuitive. Moreover, the reflex neural network could simultaneously learn to keep the high DoFs humanoid stable under the influence of a generative neural network at different frequencies, which can finally create Adaptive Imitated Central Pattern Generators, keeping a good balance of human motion imitation and adaptive capabilities by reinforcing reflex networks. It demonstrated the advantages of energy efficiency, postural balance coordination, and natural symmetry indexes.

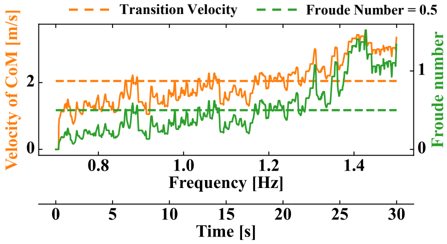

Fig. 6. Transition process from walking to running on uneven terrain. The variation for the humanoid's center of mass velocity and the Froude number.

One limitation of our study is that we tested the framework only in a simulated environment as a first-step evaluation. Further testing with a real robot is necessary to evaluate the effectiveness of the framework in the real world. Another limitation is that we tested only the locomotion task in a straight direction. It would be interesting to investigate the application of the proposed framework to other motor tasks.

In this work, reflex neural network was formed by using DRL, however the model-based approach for parameter-tuning reflex circuit is also an important research direction for understanding its internal mechanism of adaptive gait control [37].

V. Conclusions

Inspired by the roles of CPGs and reflex neural circuits in controlling legged locomotion in human, we proposed a novel control framework for humanoid locomotion based on imitation learning and reinforced reflex networks. Our AI-CPG control framework combines the advantages of feedforward and feedback control and improves the utilization efficiency of real motion data extending its speed variation. To evaluate the effectiveness of our control framework, we compared our controller with other learning-based controllers widely used for robot gait control. The results demonstrate that our controller outperforms other state-of-the-art deep reinforcement and imitation learning controllers in terms of energy efficiency, balancing ability, and adaptability for a wide range of moving speeds even on uneven surface. Our controller has the benefits of flexible gait speed adjustment during humanoid locomotion with only a single training session and one neural network, which can complete the gait transition from walking to running, at different speeds and on uneven terrains.

References

[1] O. Stasse and T. Flayols, "An overview of humanoid robots technologies," *Biomechanics of Anthropomorphic Systems*, pp. 281–310, 2019.

[2] S. Shigemi, A. Goswami, and P. Vadakkepat, "Asimo and humanoid robot research at honda," *Humanoid robotics: A reference*, pp. 55–90, 2018.

[3] T. Mikolajczyk, E. Mikołajewska, H. F. Al-Shuka, T. Malinowski, A. Kłodowski, D. Y. Pimenov, T. Paczkowski, F. Hu, K. Giasin, D. Mikołajewski *et al.*, "Recent advances in bipedal walking robots: Review of gait, drive, sensors and control systems," *Sensors*, vol. 22, no. 12, p. 4440, 2022.

[4] J. Reher and A. D. Ames, "Dynamic walking: Toward agile and efficient bipedal robots," *Annual Review of Control, Robotics, and Autonomous Systems*, vol. 4, pp. 535–572, 2021.

[5] A.-C. Hildebrandt, M. Klischat, D. Wahrmann, R. Wittmann, F. Sygulla, P. Seiwald, D. Rixen, and T. Buschmann, "Real-time path planning in unknown environments for bipedal robots," *IEEE Robotics and Automation Letters*, vol. 2, no. 4, pp. 1856–1863, 2017.

[6] C. G. Hobart, A. Mazumdar, S. J. Spencer, M. Quigley, J. P. Smith, S. Bertrand, J. Pratt, M. Kuehl, and S. P. Buerger, "Achieving versatile energy efficiency with the wanderer biped robot," *IEEE Transactions on Robotics*, vol. 36, no. 3, pp. 959–966, 2020.

[7] W. Zhao, J. P. Queralta, and T. Westerlund, "Sim-to-real transfer in deep reinforcement learning for robotics: a survey," in *2020 IEEE symposium series on computational intelligence (SSCI)*. IEEE, 2020, pp. 737–744.

[8] J. Chai and M. Hayashibe, "Motor synergy development in high-performing deep reinforcement learning algorithms," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 1271–1278, 2020.

[9] B. S. Homberg, R. K. Katzschmann, M. R. Dogar, and D. Rus, "Robust proprioceptive grasping with a soft robot hand," *Autonomous Robots*, vol. 43, pp. 681–696, 2019.

[10] S. Choi, G. Ji, J. Park, H. Kim, J. Mun, J. H. Lee, and J. Hwangbo, "Learning quadrupedal locomotion on deformable terrain," *Science Robotics*, vol. 8, no. 74, p. eade2256, 2023.

[11] 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, p. eabk2822, 2022.

[12] S. Koseki, K. Kutsuzawa, D. Owaki, and M. Hayashibe, "Multimodal bipedal locomotion generation with passive dynamics via deep reinforcement learning," *Frontiers in Neurorobotics*, 2023.

[13] I. Radosavovic, T. Xiao, B. Zhang, T. Darrell, J. Malik, and K. Sreenath, "Learning humanoid locomotion with transformers," *arXiv preprint arXiv:2303.03381*, 2023.

[14] R. P. Singh, M. Benallegue, M. Morisawa, R. Cisneros, and F. Kanehiro, "Learning bipedal walking on planned footsteps for humanoid robots," in *2022 IEEE-RAS 21st International Conference on Humanoid Robots (Humanoids)*. IEEE, 2022, pp. 686–693.

[15] G. Cheng, S. K. Ehrlich, M. Lebedev, and M. A. Nicolelis, "Neuroengineering challenges of fusing robotics and neuroscience," *Science Robotics*, vol. 5, no. 49, p. eabd1911, 2020.

[16] I. Steuer and P. A. Guertin, "Central pattern generators in the brainstem and spinal cord: an overview of basic principles, similarities and differences," *Reviews in the Neurosciences*, vol. 30, no. 2, pp. 107–164, 2019.

[17] 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.

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

[19] J. Baladron, J. Vitay, T. Fietzek, and F. H. Hamker, "The contribution of the basal ganglia and cerebellum to motor learning: A neuro-computational approach," *PLoS computational biology*, vol. 19, no. 4, p. e1011024, 2023.

[20] H. X. Ryu and A. D. Kuo, "An optimality principle for locomotor central pattern generators," *Scientific Reports*, vol. 11, no. 1, pp. 1–18, 2021.

[21] J. Nassour, T. D. Hoa, P. Atoofi, and F. Hamker, "Concrete action representation model: from neuroscience to robotics," *IEEE Transactions on Cognitive and Developmental Systems*, vol. 12, no. 2, pp. 272–284, 2019.

[22] J. Duysens and A. Forner-Cordero, "A controller perspective on biological gait control: Reflexes and central pattern generators," *Annual Reviews in Control*, vol. 48, pp. 392–400, 2019.

[23] 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, p. jeb245784, 2023.

[24] X. B. Peng, P. Abbeel, S. Levine, and M. Van de Panne, "Deepmimic: Example-guided deep reinforcement learning of physics-based character skills," *ACM Transactions On Graphics (TOG)*, vol. 37, no. 4, pp. 1–14, 2018.

[25] X. B. Peng, Z. Ma, P. Abbeel, S. Levine, and A. Kanazawa, "Amp: Adversarial motion priors for stylized physics-based character control," *ACM Transactions on Graphics (TOG)*, vol. 40, no. 4, pp. 1–20, 2021.

[26] B. W. Verdaasdonk, H. F. Koopman, and F. C. Van Der Helm, "Energy efficient and robust rhythmic limb movement by central pattern generators," *Neural Networks*, vol. 19, no. 4, pp. 388–400, 2006.

[27] H. Geyer and H. Herr, "A muscle-reflex model that encodes principles of legged mechanics produces human walking dynamics and muscle activities," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 18, no. 3, pp. 263–273, 2010.

[28] R. Ramadan, H. Geyer, J. Jeka, G. Schöner, and H. Reimann, "A neuromuscular model of human locomotion combines spinal reflex circuits with voluntary movements," *Scientific Reports*, vol. 12, no. 1, p. 8189, 2022.

[29] J. M. Wang, S. R. Hamner, S. L. Delp, and V. Koltun, "Optimizing locomotion controllers using biologically-based actuators and objectives," *ACM Transactions on Graphics (TOG)*, vol. 31, no. 4, pp. 1–11, 2012.

[30] CMU Graphics Lab. CMU graphics lab motion capture database. Available: http://mocap.cs.cmu.edu/. Accessed on June 14, 2023.

[31] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[32] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937.

[33] R. Robinson, W. Herzog, and B. M. Nigg, "Use of force platform variables to quantify the effects of chiropractic manipulation on gait symmetry." *Journal of manipulative and physiological therapeutics*, vol. 10, no. 4, pp. 172–176, 1987.

[34] R. Alexander, "Optimization and gaits in the locomotion of vertebrates," *Physiological reviews*, vol. 69, no. 4, pp. 1199–1227, 1989.

[35] S. Gatesy and A. Biewener, "Bipedal locomotion: effects of speed, size and limb posture in birds and humans," *Journal of Zoology*, vol. 224, no. 1, pp. 127–147, 1991.

[36] D. M. Bramble and D. E. Lieberman, "Endurance running and the evolution of homo," *nature*, vol. 432, no. 7015, pp. 345–352, 2004.

[37] S. Koseki, M. Hayashibe, and D. Owaki, "Identifying essential factors for energy-efficient walking control across a wide range of velocities in reflex-based musculoskeletal systems," *PLOS Computational Biology*, vol. 20, no. 1, p. e1011771, 2024.