Engineered Science
DOI: https://dx.doi.org/10.30919/es1287

Improving Bipedal Robot Motion via Reinforcement Learning and Tailored Rewards

Abdullah T. Elgammal[1, 2] and Manar Lashin[2, 3,*]

Abstract

This study proposes a novel reward function for deep deterministic policy gradient (DDPG) based bipedal robot walking control. The reward incorporates target reaching and a new term promoting stability and natural gaits via body orientation angles. This design encourages desired behaviors while adapting to diverse robot morphologies. Additionally, action-space noise (Ornstein-Uhlenbeck process) and parameter-space noise (Gaussian noise on stiffness, damping, friction) are introduced to enhance DDPG's exploration efficiency and achieve better policy learning. This combined noise strategy facilitates exploration of diverse terrains and promotes adaptive behavior. The reward function is analyzed for its impact on gait patterns and leg loading, investigating its influence on mimicking human walking and load distribution. Simulations demonstrate the robot's learning capability, achieving coordinated gait, balance, and successful termination. Torque analysis across leg joints and movement axes is conducted. The proposed approach, combining modified rewards and action/parameter space noise, offers a promising solution to mitigate local minima issues in DDPG. The MATLAB®/Simulink Reinforcement Learning toolbox is employed.

Keywords: Reinforcement learning; Bipedal robots; Deep Deterministic Policy Gradient algorithm; Reward function design.
Received: 27 July 2024; Revised: 06 September 2024; Accepted: 18 October 2024.
Article type: Research article.

1. Introduction

Reinforcement Learning (RL) in robotics offers a promising alternative to classical control techniques due to its adaptability and performance. RL, when trained on algorithms like Soft Actor-Critic (SAC),[1] allows agents to learn from experience without direct programming, enhancing adaptability in complex environments. Combining classical control methods with RL can lead to efficient and reliable decision-making systems for real-world robotics. Moreover, RL's capability to control multibody systems, as demonstrated in the context of controlling inverted N-pendulums, showcases its versatility in handling increasingly complex dynamical systems.[2] This synergy enhances adaptability and effectively addresses the complexity of real-world robotic tasks. In another study, RL technology is applied to solve the optimal control problem using an evolutionary algorithm, which finds control solutions that enable a control object to follow different trajectories while maintaining consistent quality criterion values.[3]

Additionally, the root mean square error (RMSE) of a single-degree-of-freedom (DoF) robot was measured using the TD3PG algorithm for reference tracking and rehabilitation exercises, highlighting the precision and effectiveness of RL in specific robotic applications.[4] While Deterministic Policy Gradient (DPG) and Deep Deterministic Policy Gradient (DDPG) are prominent methodologies within RL, their application in controlling nonlinear systems presents challenges. These challenges stem from limitations in adaptability to dynamically evolving obstacles and the potential for becoming trapped in local optima. Researchers are addressing these challenges by enhancing DDPG with features like LSTM network-based encoders for dynamic obstacle avoidance,[5,6] adjusting experience pool capacities for faster convergence, and modifying the Critic network structure to mitigate Q value overestimation and accelerate

[1] *Department of Mechanical Engineering, The British University in Egypt, El-Sherouk City 11837, Egypt.*
[2] *Department of Electrical Engineering, Benha Faculty of Engineering, Benha University, Benha 13511, Egypt.*
[3] *Faculty of Computer Science, Benha National University, ELObour City 11828, Egypt.*
**Email:* manar.lashin@bhit.bu.edu.eg (M. Lashin)

convergence at low learning rates.[7]

RL has been increasingly explored for bipedal locomotion control. Huang *et al.*[8] proposed a Hybrid and Dynamic Policy Gradient (HDPG) method that learns separate value functions for each reward component, resulting in hybrid policy gradients. In the context of biped robots,[9] designed a 6-DoF model and implemented deep RL for the robot to learn efficient straight-line walking, comparing two motion learning approaches. Furthermore,[10] developed a method using RL on a treadmill-like testbed to set trajectory parameters and improve the stability of bipedal walking for humanoid robots in real-world environments.

The critical differences in learning strategies between the DDPG and Twin-Delayed Deep Deterministic Policy Gradient (TD3) for bipedal walking robot control lie in their approaches to optimization and exploration. DDPG utilizes a parallel algorithm with multiple Actor-Critic networks to enhance exploration capability and improve training efficiency.[11] On the other hand, TD3 combines characteristics of different methods to achieve improved results, as demonstrated in locomotion control challenges, showcasing its effectiveness in continuous control tasks.[12] While DDPG focuses on expanding exploration through multiple networks, TD3 leverages a combination of existing methods to enhance performance and achieve competitive results in locomotion challenges.

This study employed the DDPG algorithm to control a bipedal robot model for walking, drawing inspiration from previous works on both the mechanical properties of the robot model[13] and the design of actor and critic networks.[14] While traditional control methods like Zero Moment Point (ZMP) control have been prevalent in bipedal walking control,[15] RL-based approaches offer autonomous learning of optimal walking strategies, thus enhancing adaptability and robustness in dynamic environments.[16] This paper aims to assess the efficacy of DDPG in regulating bipedal robot locomotion, particularly under parameter-space and action-space noise. The study introduces two modified reward functions tailored to regulate bipedal robot locomotion. One of these reward functions is based on the framework proposed by Lillicrap (2015)[14], while the other is designed to prioritize stability and natural walking motion. The latter incorporates additional considerations, such as body orientation angles, to promote smoother and more realistic movement patterns.[8] By developing and evaluating these modified reward functions, the study aims to enhance the performance and adaptability of the RL framework in controlling bipedal robots. Fig. 1 shows the proposed approach, which integrates a tailored reward function with body orientation angles and a combination of action-space and parameter-space exploration noise into the DDPG algorithm. This method significantly enhances the robot's gait coordination and stability, as illustrated in the rest of the paper by comparing undesired and improved walking patterns.

Figure 1 shows the proposed approach, which integrates a tailored reward function with body orientation angles and a combination of action-space and parameter-space exploration noise into the DDPG algorithm. This method significantly enhances the robot's gait coordination and stability, as will be illustrated throughout the paper by comparing undesired and improved walking patterns. The paper is structured into six sections. Section II introduces the dynamic model for biped robots utilizing a 3D linear inverted pendulum approach. Section III explains the application of DDPG in continuous control tasks, while Section IV discusses the design of three reward functions for the bipedal walking robot in the DDPG algorithm. The simulation results are presented in Section V, and Section VI concludes the paper and outlines potential future considerations.

2. Dynamics modeling: simplifying bipedal robots

The cart-table model, a simplified representation of bipedal walking robots, plays a crucial role in balancing and managing complexities. By incorporating various strategies like regulating the center of pressure, angular momentum, and ground reaction force, the model enhances stability.[17] It assumes that the robot's mass moves along a predetermined plane. The 3D-LIPM is a more complex model considering the robot's three-dimensional motion. It is based on the cart-table model but adds additional terms to account for the robot's rotation.[18] It treats its motion as an inverted pendulum with a linear dynamics model as illustrated in Fig. 2, making the control problem more straightforward and less complicated.[19] Another crucial concept in the dynamic modeling of bipedal walking robots is the Zero-Moment Point (ZMP). Controlling the ZMP location enables the generation of stable and efficient gaits for bipedal walking robots.[20,21] Incorporating these approaches provides an effective means of predicting the stability and motion patterns of bipedal walking robots. Furthermore, developing control algorithms based on these models has the potential to achieve stable and efficient bipedal locomotion.[22]

Given a normal vector $(k_x, k_y, -1)$, the constraint plane, as shown in Fig. 2, is defined by its intersection with the z-axis as $z = k_x + k_x + z_c$.[23] When the constraint plane is horizontal (*i.e.*, $k_x = k_y = 0$), the system dynamics under constraint control are described by: $\ddot{y} = \frac{g}{z_c} y - \frac{1}{m z_c} \tau_x$ (1)

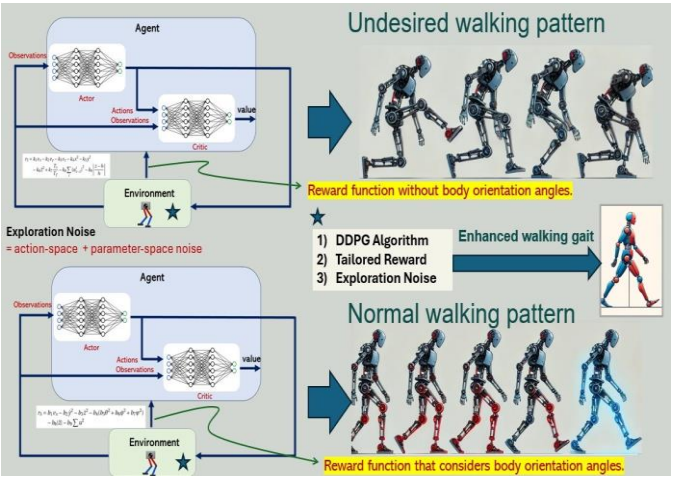


Fig. 1 Workflow of the proposed approach: improved DDPG algorithm through tailored reward functions and combined action-space and parameter-space exploration noise.

$$\ddot{x} = \frac{g}{z_c}x + \frac{1}{mz_c}\tau_y \tag{2}$$

These equations show how a bipedal robot moves like a cart on a table, considering external forces affecting its acceleration in both directions. Where **y** is the vertical position, **x** is the horizontal position, **g** is gravitational acceleration, **m** is cart mass, and $\mathbf{z_c}$ indicates the distance between the Center of Mass (CoM) of the cart and the reference point. While τ_x and τ_y are external torques on the cart. To determine the zero-moment point (ZMP) in 3D-LIPM with a horizontal constraint ($k_x = k_y = 0$), the following equations can be utilized:

$$p_x = -\frac{\tau_y}{mg} \tag{3}$$

$$p_y = -\frac{\tau_x}{mg} \tag{4}$$

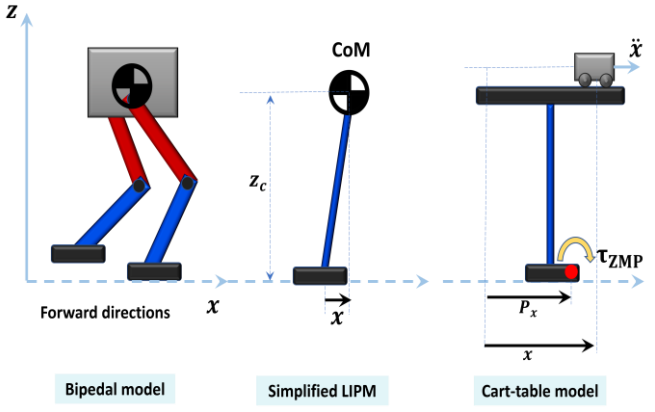


Fig. 2 Bipedal walking robot and the cart-table model.

The coordinates of the zero-moment point (ZMP) on the floor are represented by (p_x, p_y). Upon substituting to obtain the control inputs $\ddot{x}$ and $\ddot{y}$ in terms of the ZMP point, the following results are obtained:

$$\ddot{y} = \frac{g}{z_c}(y - p_y) \tag{5}$$

$$\ddot{x} = \frac{g}{z_c}(x - p_x) \tag{6}$$

Using these equations, researchers can optimize the control inputs and improve bipedal robots' stability and overall performance,[24-26] with various approaches to controller schemes being explored. The effectiveness of these schemes can vary depending on factors such as ease of implementation, energy consumption rates, and their ability to handle inaccuracies and variables. However, most model-based control schemes rely on the 3DLIPM model, which exhibits limitations regarding the robot's fundamental dynamics simulation.

Constructing a bipedal robot that includes all its dynamics and non-linear relationships without simplification poses a complex control task. RL, one of the most widely used artificial intelligence techniques for continuous task applications like control systems, will be applied to address this challenge. RL-based controllers undergo training based on several episodes without using mathematical equations to simulate the model. The effectiveness of the trained agent depends on various factors, including network structure, reward function elements, weight updating mechanisms, training options such as discount factor, and stop training criteria. In our study, we compared the results of an RL-based model that we modified to improve performance with those obtained by [13], using the same environment structure with added uncertainties and parameter-noise incorporated and the same mechanical model found in the MATLAB® program.

3. DDPG for continuous control tasks

DDPG is a popular RL algorithm that can effectively handle continuous action-spaces and learn complex policies. It uses an actor-critic architecture comprising deep neural networks to represent the actor and critic functions as shown in Fig. 3.[27] The actor network, denoted as μ with parameter θ_μ and the critic network, denoted as λ with parameter θ_λ. The DDPG agent uses an action to interact with the environment and receives an observation and a reward. These observations, along with the action taken and the reward received, are stored in a replay memory buffer. During each iteration, a random batch of **N** observations is sampled from the replay memory buffer. The actor and critic networks use these observations to determine the Q-value (a measure of expected future reward) in the next observation and update their predictions accordingly. The actor target network then generates an action (μ) based on the updated Q-value, which is used to determine the Q-value in the following observation.

The main network then calculates a loss value using eq. (7), and this value is used to update the critic network through a process called gradient distribution. The actor network is updated based on the predictions from the critic network using a policy gradient (as shown in eq. (8)). To improve exploration efficiency, noise is added to the action obtained from the actor network using another equation. Therefore, the goal is to optimize the agent's behavior by continually adjusting the actor and critic networks based on their predictions and actions. The loss function for updating the critic network is expressed as follows[28]:

$$L = E[\left(r + \gamma Q\left(S_{i+1}+1, \mu(S_{i+1})\right) - Q(S_i, A_i)\right)^2] \quad (7)$$

E is the expectation over experiences sampled from the replay buffer. **r** is the reward for taking action $\mathbf{A_i}$ in state $\mathbf{S_i}$. $\mathbf{Q(S_i,A_i)}$ is the current estimate of the Q-value function for the given state-action pair, and $\mathbf{Q(S_i+1,}\mu\mathbf{(S_i+1))}$ is the estimated Q-value for the next state $\mathbf{S_i+1}$ with the discount factor γ. Equation (8) provides the method for updating the actor network's policy gradient:

$$\nabla_{\theta_\mu} J = \frac{1}{N}\sum_{i=0}^{N} \nabla_A Q(S_i, \mu(S_i)|\theta_\lambda).\nabla_{\theta_\mu}\mu(S_i, \theta_\mu) \quad (8)$$

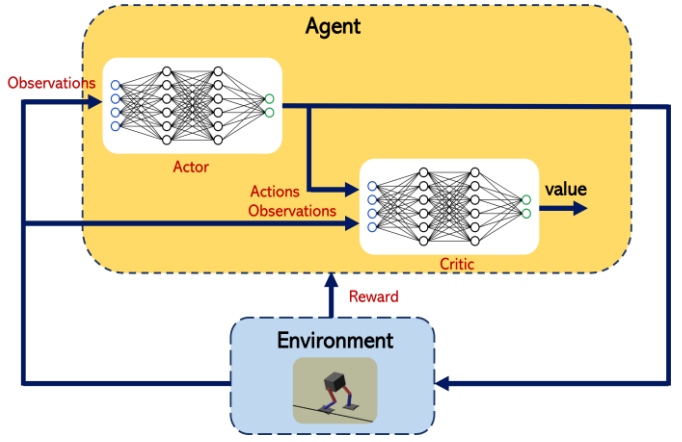


Fig. 3 DDPG for bipedal walking robot.

The stochastic noise is integrated into the action generated by the actor network through eq. 9 to increase exploration efficiency. The DDPG steps are outlined in Table 1.

$$A_t = \mu(S_t) + \epsilon \quad (9)$$

Table.1 DDPG algorithm.

Algorithm 1: DDPG algorithm [14]

Input: Randomly initialize critic network $Q(s,a|\theta_\lambda)$ and actor $\mu(s|\theta_\mu)$ with weights θ_λ and θ_μ. Initialize target network Q_0 and μ_0 with weights $\theta_{\lambda_0} \leftarrow \theta_\lambda$, $\theta_{\mu_0} \leftarrow \theta_\mu$. Initialize replay buffer R.

for *episode* = 1 *to* M **do**
  Initialize a random process ϵ for action exploration;
  Receive initial observation state s_1;
  for $t = 1$ *to* T **do**
    Select action $a_t = \mu(s_t|\theta_\mu) + \epsilon_t$ according to the current policy and exploration noise;
    Execute action a_t and observe reward r_t and observe new state s_{t+1};
    Store transition (s_t, a_t, r_t, s_{t+1}) in R;
    Sample a random minibatch of N transitions (s_i, a_i, r_i, s_{i+1}) from R;
    Set $y_i = r_i + \gamma Q'(s_{i+1}, \mu'(s_{i+1}|\theta_{\mu'})|\theta_{\lambda'})$;
    Update critic by minimizing the loss: $L = \frac{1}{N}\sum_i (y_i - Q(s_i, a_i|\theta_\lambda))^2$;
    Update the actor policy using the sampled policy gradient: $\nabla_{\theta_\mu} J \approx \frac{1}{N}\sum_i \left(\nabla_a Q(s, a|\theta_\lambda)|_{s=s_i, a=\mu(s_i)}\right)\nabla_{\theta_\mu}\mu(s|\theta_\mu)|_{s_i}$;
    Update the target networks: $\theta_{\lambda'} \leftarrow \tau\theta_\lambda + (1-\tau)\theta_{\lambda'}$, $\theta_{\mu'} \leftarrow \tau\theta_\mu + (1-\tau)\theta_{\mu'}$;

There are two types of exploration noise that are commonly used in DDPG: *action-space noise* and *parameter-space noise*. Action-space noise involves adding random perturbations to the actions the actor generates and is the more widely used method. On the other hand, parameter-space noise adds noise directly to the weights of the actor-network or to the parameters of the environment itself. When dealing with nonlinear systems such as bipedal walking robots, or other highly nonlinear and uncertain systems, parameter-space noise may be more reflective of real-world scenarios, as there are many uncertainties in parameters, perturbations, and disturbances that could arise from sensors or model inaccuracies. Incorporating noise into environment parameters can assist in making the trained agent more robust against system uncertainties. Action-space noise allows the agent to explore the state space more diversely and discover new, potentially superior action policies. However, the appropriate noise type and strength choice may vary depending on the task and system dynamics.

Previous studies have investigated the efficacy of exploration noise in DDPG for controlling bipedal walking robots.[29] One commonly used approach for incorporating noise into DDPG involves utilizing an Ornstein-Uhlenbeck (OU) process that generates temporally correlated noise with zero-centered mean-reverting properties, which makes it well-suited for continuous control tasks. In this study, we aim to optimize the performance of the DDPG algorithm by integrating two noise generation approaches. Firstly, we apply the Ornstein-Uhlenbeck (OU) process to introduce noise in the action space, and secondly, we employ Gaussian noise with zero mean and a standard deviation of 0.1 to perturb critical environmental parameters, such as stiffness, damping, and friction coefficients, that affect the contact and friction properties. By combining these two methods, we aim to enhance the exploration efficiency of the DDPG algorithm and achieve better policy learning. This noise encourages the agent's exploration of diverse terrains and adaptive behavior. Additionally, adding noise to leg, foot, and torso dimensions can broaden the range of body configurations and improve overall performance. By utilizing both types of noise, we aim to leverage the strengths of each strategy and achieve a more significant improvement in algorithm performance.

Simulations validate the proposed control approach for bipedal robots, demonstrating its potential for real-world applications. We leveraged MATLAB's deep learning toolbox to optimize key factors influencing walking performance. This involved customizing critic and actor representations using rlOptimizerOptions, allowing control over learning parameters (optimizer, learning rate, *etc*.). As shown in Fig. 4, the critic network (9 layers, inspired by [14]) took both observation and action as input, while the actor network (7 layers) used only observation to generate the control output.

The actor network is designed as a continuous deterministic actor, while the critic network is implemented as a Q-value function. Both networks are optimized using the Adam optimizer with specified learning rates (1e-3) and gradient thresholds. The actor network is trained to provide deterministic continuous actions based on the observation and action spaces, while the critic network estimates the Q-values

by mapping observations and actions. We have also incorporated L2 regularization for both networks to enhance generalization. The DDPG agent's configuration includes essential elements such as mini-batch size, experience buffer, noise options, and discount factor to ensure robust policy learning.

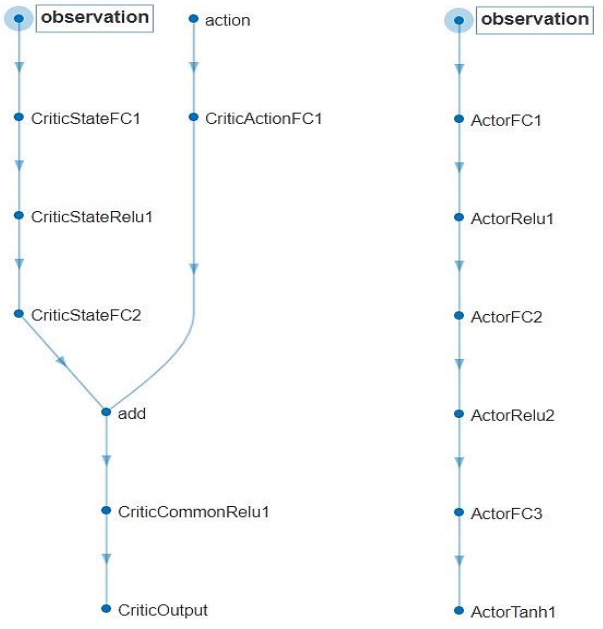


Fig. 4 Actor-critic network structure for bipedal walking robot using MATLAB® deep learning toolbox.

4. Designing reward function for bipedal walking robot in DDPG algorithm

The reward function's design in the DDPG algorithm significantly impacts the agent's learning and performance. Poorly designed reward functions can hinder learning, resulting in suboptimal policies. Overcoming challenges associated with sparse rewards is essential, and techniques like reward shaping or providing dense rewards at intermediate states have been proposed to address this issue.[14] Selecting appropriate scaling factors for different components of the reward function is crucial to balance exploration and exploitation. By manipulating reward signals, bipedal robots can be incentivized to move more naturally and efficiently, prioritizing desired behaviors such as longer strides and balance maintenance. Heess *et al.* (2019) proposed a straightforward reward function for bipedal walking robots using the DDPG algorithm[31]:

$$r_1 = v_x - 3y^2 - 50\hat{z}^2 + 25\frac{T_s}{T_f} - 0.02\sum_i\left(u_{t-1}^i\right)^2 \quad (10)$$

Here, $\mathbf{v_x}$ incentivizes forward motion, $\mathbf{y^2}$ penalizes lateral deviations, $\mathbf{z\hat{}^2}$ encourages balance maintenance, and the last term penalizes excessive control effort. A modified reward function, designed to optimize overall performance, introduces additional terms to address specific challenges:

$$r_2 = k_1 v_x - k_2 v_y - k_3 v_z - k_4 x^2 - k_5 y^2 - k_6 \hat{z}^2 + k_7\frac{T_s}{T_f} - k_8\sum_i\left(u_{t-1}^i\right)^2 - k_9\left|\frac{z-h}{h}\right| \quad (11)$$

This modification includes terms to encourage consistent height, address lateral movement issues, and promote progress towards the goal. However, it still lacks explicit consideration for stability in body orientation and abrupt changes in vertical acceleration, which are critical for bipedal walking. A final proposal to refine the reward function focuses on robot stability:

$$r_3 = b_1 v_x - b_2\dot{y}^2 - b_3\dot{z}^2 - b_4(b_5\theta^2 + b_6\phi^2 + + b_7\psi^2) - b_8|\ddot{z}| - b_9\sum u^2 \quad (12)$$

In addition to promoting forward motion and stability, this reward structure explicitly considers body orientation and changes in vertical acceleration, addressing shortcomings in previous versions. However, achieving optimal performance requires careful tuning of the coefficients and a thorough understanding of system dynamics. Moreover, challenges like tilting, sliding, and jumping still need to be addressed to ensure robust and stable bipedal locomotion.

In Eq. (11), a refined reward structure is presented, introducing terms for vertical velocity ($\mathbf{v_z}$) and height related penalties to enhance the robot's vertical stability. It maintains velocity-based rewards and control effort penalties similar to Eq. (10). However, it lacks explicit penalties for abrupt changes in vertical acceleration and stability in body orientation, potentially limiting its effectiveness in achieving stable bipedal locomotion. Equation 12 stands out by integrating a comprehensive set of rewards and penalties, addressing key aspects of stability and motion control. It incorporates forward velocity ($\mathbf{v_x}$) alongside terms for lateral and vertical stability ($\dot{y}^2$, $\dot{z}^2$), stability in body orientation (θ^2, ϕ^2, ψ^2), penalties for abrupt changes in vertical acceleration ($|\ddot{z}|$), and control effort. This holistic approach ensures the robot moves efficiently while maintaining a stable posture and trajectory. Moreover, including a height-related penalty term further enhances the robot's ability to maintain consistent vertical positioning, which is crucial for bipedal walking. Equation 12 encapsulates a multifaceted reward structure designed for training bipedal walking robots through RL. Each term in this equation serves a specific purpose:

- $b_1 v_x$: Rewards forward velocity, encouraging efficient motion.
- $b_2\dot{y}^2$: Penalizes lateral deviations, ensuring smooth movement.
- $b\,\dot{y}^2$: Penalizes excessive vertical motion, promoting stability.
- $b_4(b_5\theta^2 + b_6\phi^2 + b_7\psi^2)$: Addresses stability in body orientation, fostering robust posture.
- $b_8|\ddot{z}|$: Penalizes abrupt changes in vertical acceleration, promoting smoother motion.
- $b_9\sum u^2$: Penalizes control effort, encouraging energy efficient locomotion.

Equation 12 merges various terms into a unified reward structure, which motivates efficient motion, stability, robust body orientation, smooth acceleration, and minimal control effort. This equation provides a comprehensive framework for training robots to walk stably and effectively by addressing these critical aspects of bipedal locomotion. Encouraging

alternating gaits in bipedal robots requires several strategies. Rewarding swing phase duration promotes alternation, penalizing single stance time discourages static postures, and rewarding horizontal foot velocity stimulates essential leg motion. Minimizing vertical acceleration promotes a stepping gait, while localizing torque rewards to legs emphasizes dynamic leg swinging motions over bracing. These strategies foster alternating gaits, while implementation challenges need careful consideration.

5. Simulation results

This section investigates the performance of RL-based control utilizing the DDPG algorithm with the modified reward functions. All training experiments were conducted on a machine equipped with an 11th Gen Intel(R) Core(TM) i7-1165G7 processor (2.80GHz), 16.0 GB RAM, and a 64-bit operating system. The training was performed over 1000 episodes, with each episode running for a total time of 20 seconds, and the maximum steps per episode determined by the sampling time The model uses **29** observations to learn the behavior of the robot. These observations are:

- **X**, **Y**, and **Z** coordinates of the robot's location in space.
- $\mathbf{V_x}$, $\mathbf{V_y}$, and $\mathbf{V_z}$ velocities of the robot's movement.
- Yaw, pitch, and roll angles of the robot's orientation.
- Angular velocities associated with the robot's orientation.
- Right ankle angle and speed.
- Right knee angle and speed.
- Right hip angle and speed.
- Left ankle angle and speed.
- Left knee angle and speed.
- Left hip angle and speed.
- Previous actions (**6** actions/**3** per leg) of the robot.
- The previous actions provide information about the robot's recent activities and may impact its current behavior. Meanwhile, the model has 6 actions that it can take to control the robot's movement. These actions are 3 joint torques per leg (ankle, knee, hip) for the right/left leg. Once a suitable model has been constructed with the necessary specifications and adjustments using the Deep Learning toolbox, the impact of introducing parameter noise along with action noise can be evaluated. After analyzing the data, it has been determined that specific statistical measures are essential in providing insights into the distribution of rewards within the modified reward function. The mean reward of 16.03 indicates that the agent has received positive feedback for their actions and is performing well in the environment (Fig. 5). However, the median reward of 8.86 is lower than the mean, which suggests the existence of outliers that are causing the mean to skew upward. Despite this, the median is still positive, indicating that most rewards are positive.
- Additionally, the standard deviation of the reward (29.83) is considerably high, which suggests that the rewards are spread across a broad range. This may imply variability in the agent's performance or that the environment may be unpredictable. The minimum reward of -141.24 indicates that the agent has encountered significant negative feedback, while the maximum reward of 150.57 suggests that the agent has also received highly positive rewards. These extreme values indicate that the reward function may be non-linear or the environment may have complicated dynamics(Fig. 6).
- In parallel with this investigation, Fig. 7 displays the torque exerted on the robot's legs during its locomotion. The highest torques are applied at the onset of each step when the robot pushes off from the ground. As the robot moves through the air, the magnitude of the torques decreases, reaching their lowest point at the end of each step when the robot lands on the ground. In Fig. 8, we observe the energy utilized by the robot's motors over time. The peak energy consumption occurs at the beginning of each step when the robot initiates takeoff from the ground. Fig. 9 shows the robot's position in the X, Y, and Z axes over time. The robot was able to maintain a relatively straight trajectory with minimal deviation from its initial path. The robot also maintained a well-balanced CoM throughout its movement.

One of the challenges encountered when training bipedal walking robots using RL is the emergence of suboptimal gait patterns. While the robot may achieve high cumulative and average rewards during training, the resulting walking behavior might deviate significantly from the desired outcome. These suboptimal patterns often manifest as hopping on one leg (either left or right) to avoid falling, or walking at an excessively high speed. Notably, these behaviors might not be readily apparent during the training process itself, only becoming evident upon successful completion. In this work, we investigate the impact of these one-legged motion patterns on joint torques, and average reward (Equation 12).

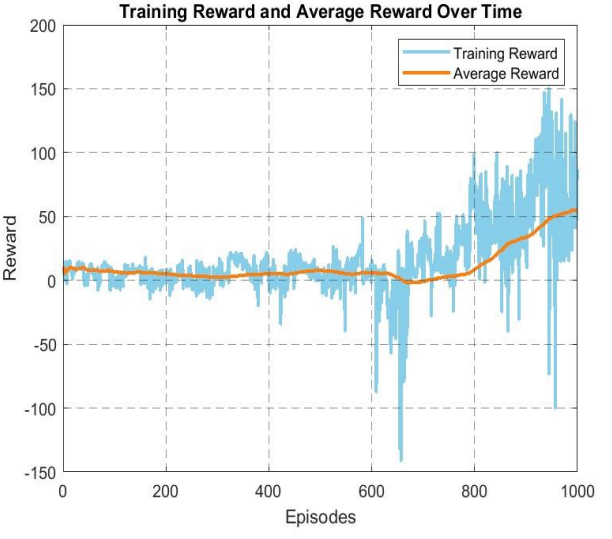
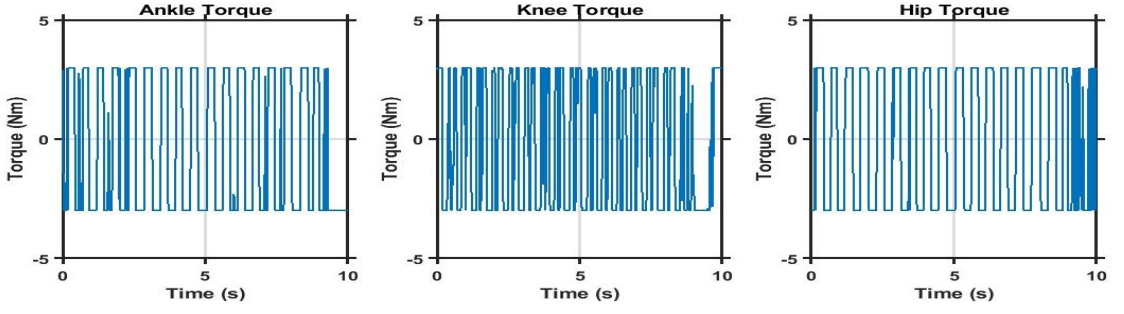


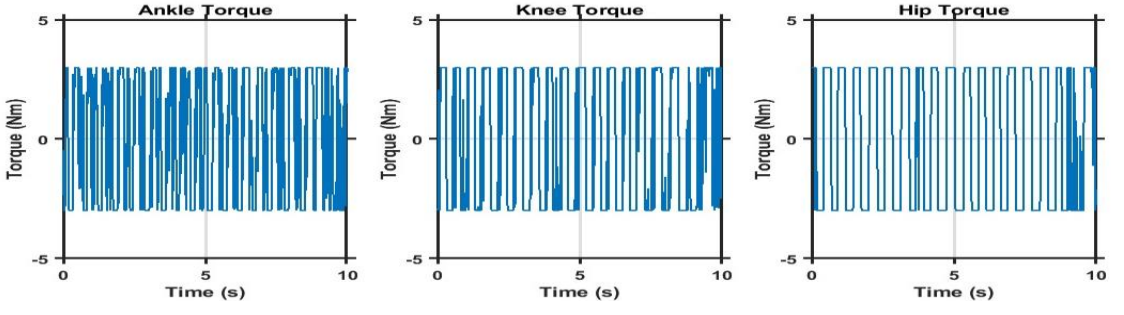
Fig. 5 The reward function VS average reward for DDPG agent.

We categorize the resulting gaits into four groups: **res1**: Robot's left leg significantly precedes the right leg (hopping on the left leg). **res2**: Robot's right leg significantly precedes the left leg (hopping on the right leg). **res3:** Robot utilizes both legs for walking but exhibits an excessively fast gait. **res4:**

Fig. 6 Dynamic interaction between reward function and Q0 in training performance.

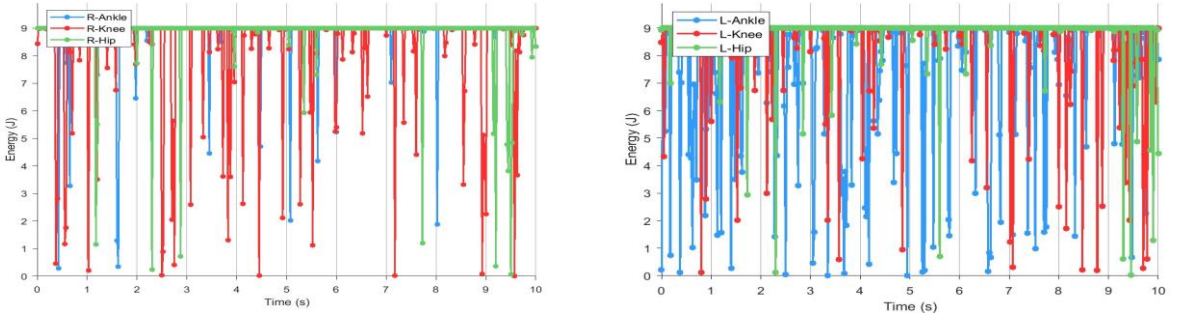


(a) Joint Torques of Right Leg



(b) Joint Torques of Left Leg

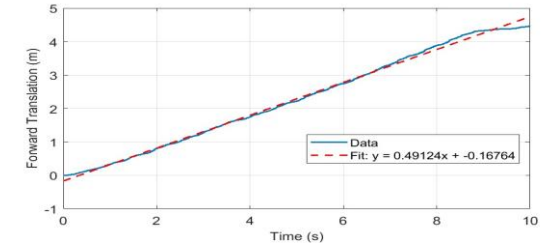
Fig. 7 Exploring the joint torques of right and left legs.



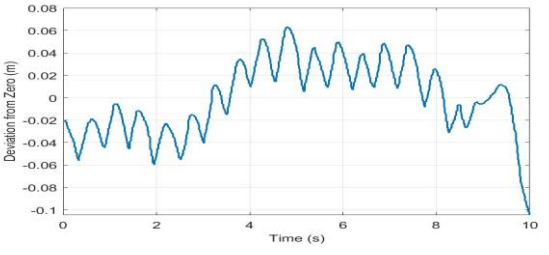
Right torque energy

Left torque energy

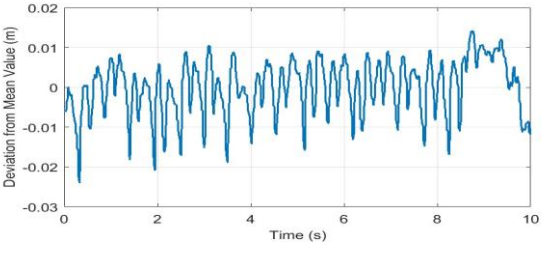
Fig. 8 Comparing right and left torque energies: a visual representation.



(a) Forward Translation x(m)



(b) Lateral Translation y(m)



(c) Vertical Translation z(m)

Fig. 9 Analyzing the XYZ translations: tracking, forward, lateral deviation and vertical movement.

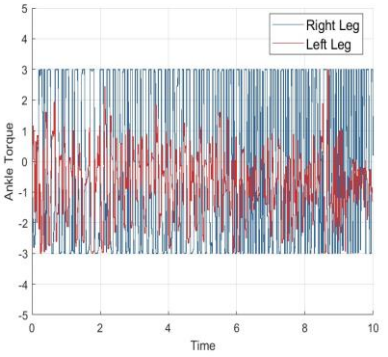
Robot walks satisfactorily on both legs, maintaining a straight-line trajectory without falling(see Fig. 10).

Figure 11 illustrates the episode rewards and the average reward plotted against the episode index for the four presented gait patterns. Fig. 12 illustrates the forward (**x**), lateral (**y**), and vertical (**z**) motion of the robot. The orientation of the body is represented by Fig. 13, depicting the tilting angles ϕ, θ, and ψ. Fig. 12a illustrates the forward and backward movement of the robot's CoM, indicated by the X-axis. A positive value denotes forward motion, while a negative value represents backward movement. Fig. 12b displays the vertical displacement of the CoM over time, denoted by the y-axis. Positive values indicate upward movement relative to a reference point, while negative values signify downward movement. This indicates slight vertical oscillation of the robot's CoM during walking. Fig. 12c confirms the oscillation of the robot's CoM along the z-axis as it walks. This oscillation contributes to stability, especially on uneven terrain, resembling natural human walking patterns. Fig. 13a depicts the side-to side tilting of the robot's body, known as roll. The oscillation of the robot's CoM roll, biased towards negative values initially, indicates a slight lean to the left during walking. Fig. 13b represents the forward and backward tilting of the robot's body, referred to as pitch. The robot's CoM pitch oscillation during walking resembles natural human gait patterns. Fig. 13c illustrates the rotation of the robot's body around a vertical axis, known as yaw. The oscillation of the robot's CoM yaw indicates rotational movement during walking. As discussed, eq (12) introduces a novel reward function incorporating body orientation angles and vertical acceleration. This addition demonstrably improves gait pattern and robot stability compared to prior approaches (eq. 10, 11). Moreover, this reward structure facilitates consistent training convergence without need for terms like ($25\frac{T_s}{T_f}$and $k_7\frac{T_s}{T_f}$), as the training process consistently converges without issues.

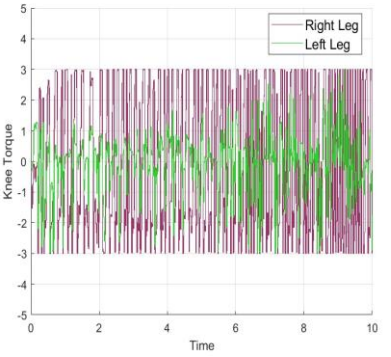
For traditional control methods, such as Zero Moment Point (ZMP) control, studies have demonstrated that energy efficiency can vary depending on the specific implementation. For instance, Vukobratovic *et al.* (2004)[15] highlighted that ZMP-based control strategies for bipedal robots often require continuous adjustments to maintain stability, particularly on uneven terrain, which can result in less energy efficiency. In contrast, reinforcement learning (RL) approaches, such as the DDPG-based method proposed in this study, inherently optimize control effort over time through continuous learning, allowing for more adaptive and efficient control.

In our work, the torque patterns generated by the DDPG agent, as shown in Fig. 11, exhibit smoother and more stable control signals as training progresses (modified reward r3). The DDPG agent stabilizes its control strategy, resulting in more consistent and predictable torque outputs, which correspond to improved balance and smoother locomotion. These patterns indicate that the agent is learning to apply torque more effectively to maintain stability without exerting excessive or unnecessary force on the joints (modified reward r3). While DDPG demonstrates potential for improving energy efficiency through refined torque management, the current results reflect the adaptability of the algorithm in learning efficient control. This lays the groundwork for future comparisons with traditional control methods to quantify energy savings more comprehensively. Additionally, similar results have been observed in other RL applications, such as quadruped robots, where optimized control strategies led to reduced energy consumption.[30] However, challenges remain in ensuring that the DDPG method can adapt to rapidly changing environments, which could require further advancements in RL methodologies.

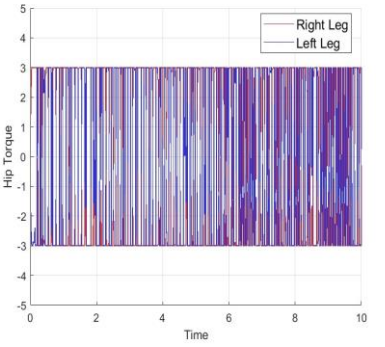
Finally, we compared the performance of the DDPG algorithm with a modified reward function (r2) against the TD3 algorithm using the original reward function proposed by Heess *et al.* (2017).[31] The comparison highlights that while both algorithms demonstrate steady improvement over time, the DDPG algorithm achieves a slightly higher mean reward (16.03) compared to TD3's mean reward of 15.04, as illustrated in Fig. 14.



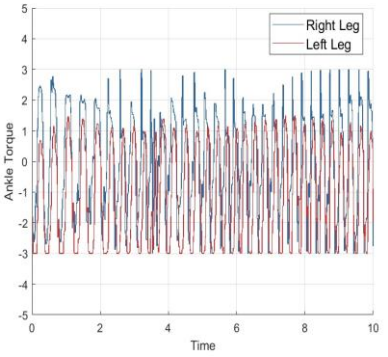
(a) Ankle res1



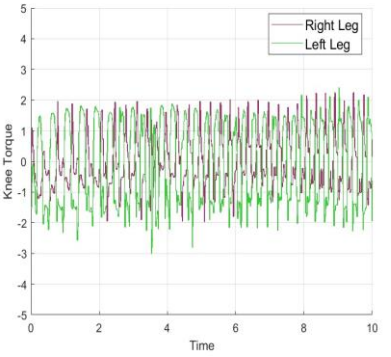
(b) Knee res1



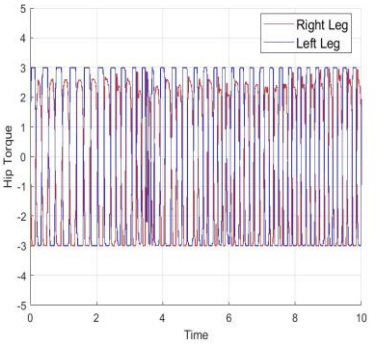
(c) Hip res1



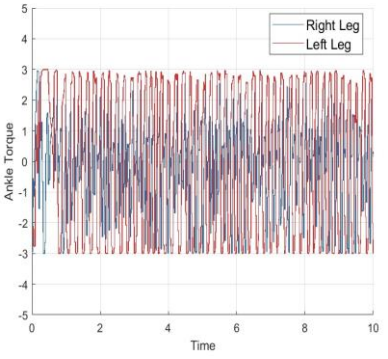
(d) Ankle res2



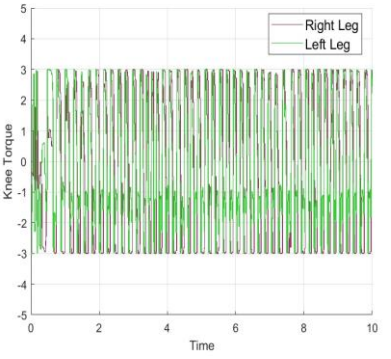
(e) Knee res2



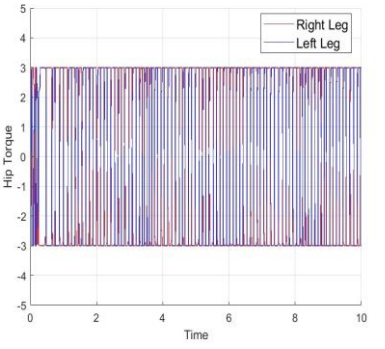
(f) Hip res2



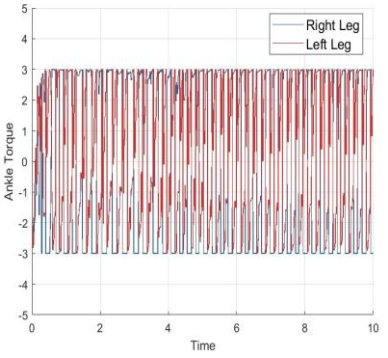
(g) Ankle res3



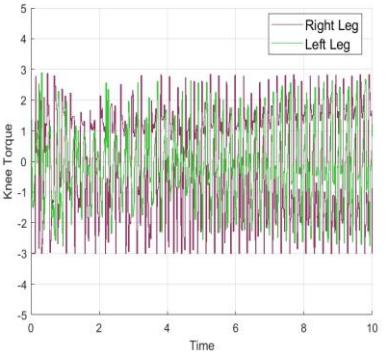
(h) Knee res3



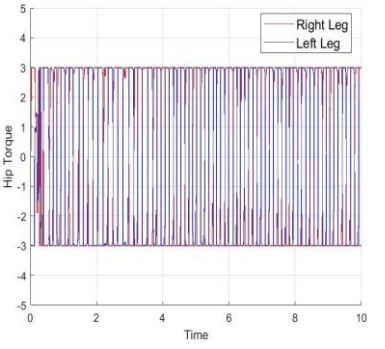
(i) Hip res3



(j) Ankle res4

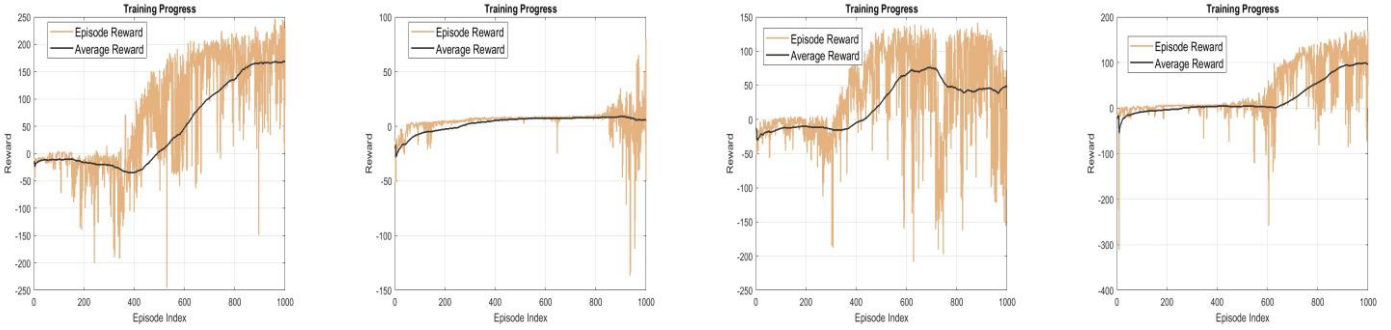


(k) Knee res4



(l) Hip res4

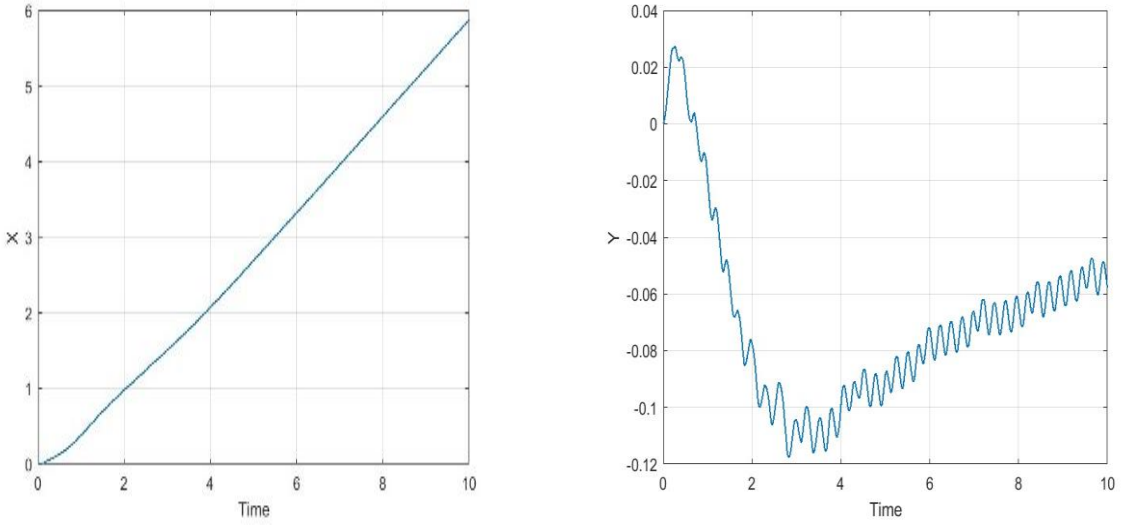
Fig. 10 Comparison of joint torques for different motion patterns.



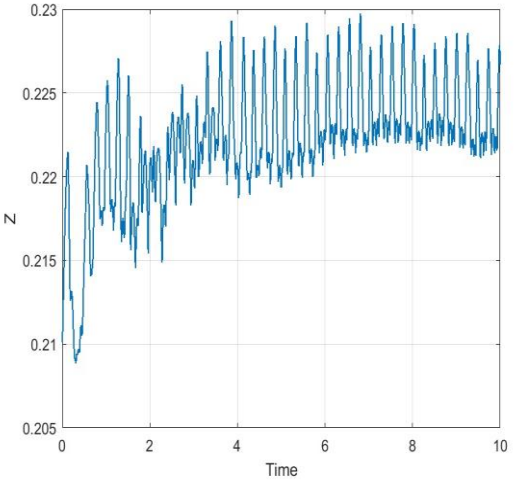
(a) Reward for res1 (b) Reward for res2 (c) Reward for res3 (d) Reward for res4

Fig. 11 Comparison of reward for different result groups.

Furthermore, the critic network in DDPG appears to provide more accurate long-term reward estimates, as indicated by a strong positive correlation between the rewards obtained during training and the Q0 values. This is critical for the agent's ability to learn and maintain a policy that achieves consistent walking behavior across various terrains.

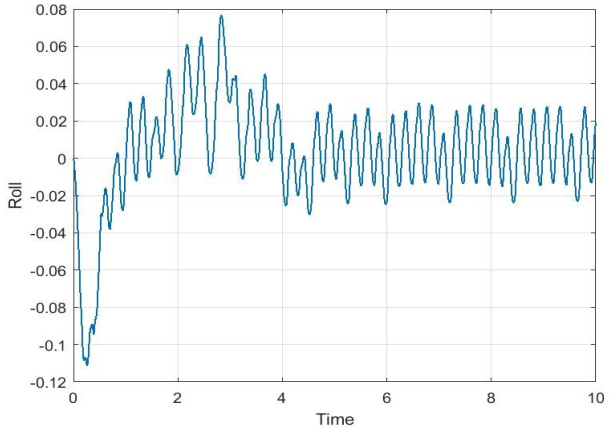


(a) Forward translation x(m) (b) Lateral translation y(m)

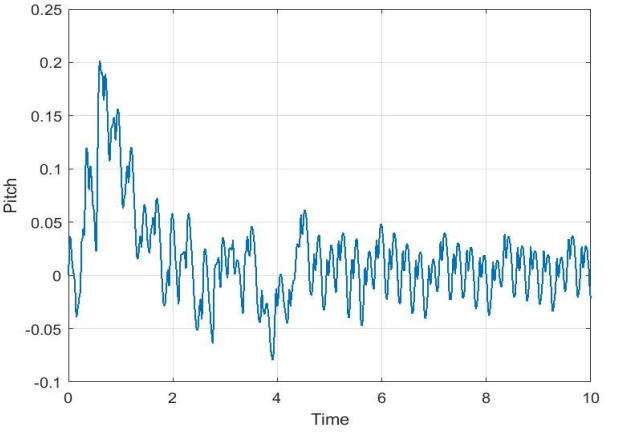


(c) Vertical translation z(m)

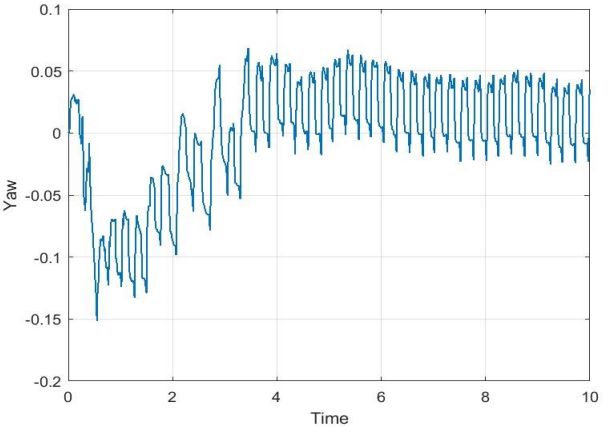
Fig. 12 Analyzing the XYZ translations: tracking forward, lateral deviation and vertical movement.



(a) Tilting from side to side



(b) Tilting forward and backward



(c) Rotation around the axis perpendicular to the ground

Fig. 13 Analyzing the ϕ, θ, and ψ of the robot body continuously learns and adapts, improving walking performance over extended operation.

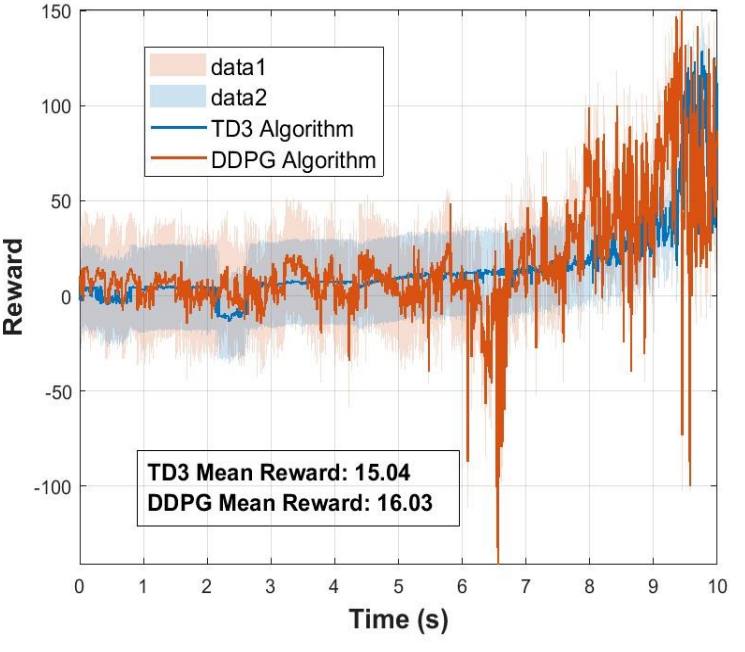


Fig. 14 Comparison of rewards for DDPG and TD3 algorithms.

6. Conclusions and future considerations

This study successfully tested two reward structures using DDPG and was able to achieve robust bipedal walking control. The proposed approach included exploration noise that helped the robots to adapt to diverse morphologies and gaits while mitigating local minima convergence. The learned gait patterns were similar to those of human walking, revealing the potential of RL-based control with adaptive reward functions in achieving natural gaits.

Future research will investigate adaptive learning algorithms to personalize reward functions in real-time based on the robot's performance and environment. Additionally, adaptive noise mechanisms will optimize the exploration-exploitation trade-off, balancing discovery of novel gaits with the refinement of existing ones. Finally, lifelong learning algorithms will enable robots to continuously improve their walking stability and adaptability, allowing them to respond to new environments, changes in terrain, and evolving tasks over time.

Conflict of Interest

There is no conflict of interest.

Supporting Information

Not applicable.

References

[1] D. Chatterjee, R. Roy, A. Sengupta, Comparison of Reinforcement Learning controller with a classical controller for an UAV, 2023 Second International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT). Trichirappalli, India. IEEE, 2023.

[2] P. Manzl, O. Rogov, J. Gerstmayr, A. Mikkola, G. Orzechowski, Reliability evaluation of reinforcement learning methods for mechanical systems with increasing complexity, *Multibody System Dynamics*, 2023, 1–25, doi: 10.1007/s11044-023-09960-2.

[3] A. Diveev, E. Sofronova, S. Konstantinov, V. Moiseenko, Reinforcement learning for solving control problems in robotics, *Engineering Proceedings*, 2023, **33**, 29, doi: 10.3390/engproc2023033029.

[4] B. R. G. Beck, J. Tipper, S. Su, Comparison of constant PID controller and adaptive PID controller via reinforcement learning for a rehabilitation robot, 2022 Australian & New Zealand Control Conference (ANZCC). Gold Coast, Australia. IEEE, 2022.

[5] L. Ye, P. Jiang, Optimization control of the double-capacity water tank-level system using the deep deterministic policy gradient algorithm, *Engineering Reports*, 2023, **5**, e12668, doi: 10.1002/eng2.12668.

[6] H. Hu, Y. Chen, T. Wang, F. Feng, W. Chen, Research on the deep deterministic policy algorithm based on the first-order inverted pendulum, *Applied Sciences*, 2023, **13**, 7594, doi: 10.3390/app13137594.

[7] L. Xi, J. Wu, Y. Xu, H. Sun, Automatic generation control based on multiple neural networks with actor-critic strategy, *IEEE Transactions on Neural Networks and Learning Systems*, 2020, **32**, 2483-2493, doi: 10.1109/TNNLS.2020.3006080.

[8] C. Huang, G. Wang, Z. Zhou, R. Zhang, L. Lin, Reward-Adaptive reinforcement learning: dynamic policy gradient optimization for bipedal Locomotion, *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 2023, **45**, 7686-7695, doi: 10.1109/TPAMI.2022.3223407.

[9] T. Tadić, P. Ćurković, Biped Robot Walking based on Deep Reinforcement Learning, 2023 46th MIPRO ICT and Electronics Convention (MIPRO). Opatija, Croatia. IEEE, 2023.

[10] Y. Chun, J. Choi, I. Min, M. Ahn, J. Han, DDPG reinforcement learning experiment for improving the stability of bipedal walking of humanoid robots, 2023 IEEE/SICE International Symposium on System Integration (SII). Atlanta, GA, USA. IEEE, 2023.

[11] T. Huckell, A. R. Wu, Improved zero step push recovery with a unified reduced order model of standing balance, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). Kyoto, Japan. IEEE, 2022.

[12] C. Tao, J. Xue, Z. Zhang, Z. Gao, Parallel deep reinforcement learning method for gait control of biped robot, *IEEE Transactions on Circuits and Systems II: Express Briefs*, 2022, **69**, 2802–2806, doi: 10.1109/tcsii.2022.3145373.

[13] Y. H. Xu, J. W. **e, Y. G. Zhang, M. Hua, W. Zhou, Reinforcement learning (RL)-based energy efficient resource allocation for energy harvesting-powered wireless body area network, *Sensors*, 2019, **20**, 44, doi: 10.3390/s20010044.

[14] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:2015, 1509.02971.

[15] M. Vukobratović, B. Borovac, Zero-moment point-thirty-five years of its life, *International Journal of Humanoid Robotics*, 2004, **1**, 157-173, doi: 10.1142/S0219843604000083.

[16] J. Kober, J. A. Bagnell, J. Peters, Reinforcement learning in robotics: a survey, *The International Journal of Robotics Research*, 2013, **32**, 1238-1274, doi: 10.1177/0278364913495721.

[17] T. Huckell, A. R. Wu, Improved zero step push recovery with a unified reduced order model of standing balance, 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, 2022.

[18] J. Reher, A. D. Ames, Dynamic walking: toward agile and efficient bipedal robots, *Annual Review of Control, Robotics, and Autonomous Systems*, 2021, **4**, 535-572, doi: 10.1146/annurev-control-071020-045021.

[19] C. Liu, J. Ning, Q. Chen, Dynamic walking control of humanoid robots combining linear inverted pendulum mode with parameter optimization, *International Journal of Advanced Robotic Systems*, 2018, **15**, 1729881417749672, doi:

10.1177/1729881417749672.
[20] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Resolved momentum control: humanoid motion planning based on the linear and angular momentum, Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453). Las Vegas, NV, USA. IEEE, 2003.
[21] M. Vukobratovic, C. Juricic, Contribution to the synthesis of biped gait, *IEEE Transactions on Biomedical Engineering*, 1969, **16**, 1-6, doi: 10.1109/TBME.1969.4502596.
[22] J. J. Kuffner, S. Kagami, K. Nishiwaki, M. Inaba, H. Inoue, Dynamically-stable motion planning for humanoid robots, *Autonomous robots*, 2002, **12**, 105–118, doi: 10.1023/A:1013219111657.
[23] S. Kajita, F. Kanehiro, K. Kaneko, K. Fujiwara, K. Harada, K. Yokoi, H. Hirukawa, Biped walking pattern generation by using preview control of zero-moment point, 2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422). Taipei, China. IEEE, 2003.
[24] T. -H. S. Li, Y. -T. Su, S. -H. Liu, J. -J. Hu, C. -C. Chen, Dynamic balance control for biped robot walking using sensor fusion, kalman filter, and fuzzy logic, *IEEE Transactions on Industrial Electronics*, 2012, **59**, 4394-4408, doi: 10.1109/TIE.2011.2175671.
[25] E. R. Westervelt, J. W. Grizzle, C. Chevallereau, J. H. Choi, B. Morris, Zero dynamics of bipedal locomotion. Feedback Control of Dynamic Bipedal Robot Locomotion, *CRC Press*, 2018, 111-135, doi: 10.1201/9781420053739-5.
[26] D. Katic, M. Vukobratovic, Survey of intelligence control techniques for humanoid robots, *Journal of Intelligence and Robotic Systems*, 2003, **37**, 117-141, doi: 10.1023/A:1024172417914.
[27] Y. Gao, Y. Matsunami, S. Miyata, Y. Akashi, Operational optimization for off-grid renewable building energy system using deep reinforcement learning, *Applied Energy*, 2022, **325**, 119783, doi: 10.1016/j.apenergy.2022.119783.
[28] C. Panjapornpon, P. Chinchalongporn, S. Bardeeniz, R. Makkayatorn, W. Wongpunnawat, Reinforcement learning control with deep deterministic policy gradient algorithm for multivariable pH process, *Processes*, 2022, **10**, 2514, doi: 10.3390/pr10122514.
[29] M. Plappert, R. Houthooft, P. Dhariwal, S. Sidor, R. Y. Chen, X. Chen, T. Asfour, P. Abbeel, M. Andrychowicz, Parameter space noise for exploration, 2017, preprint arXiv:1706.01905, doi: 10.48550/arXiv.1706.01905.
[30] Z. Yan, H. Ji, Q. Chang, Energy consumption minimization of quadruped robot based on reinforcement learning of DDPG algorithm, *Actuators*, 2024, **13**, 18, doi: 10.3390/act13010018.
[31] N. Heess, Dhruva TB, S. Sriram, J. Lemmon, J. Merel, G. Wayne, Y. Tassa, T. Erez, Z. Wang, S. M. Ali Eslami, M. Riedmiller, D. Silver, arXiv, Emergence of locomotion behaviours in rich environments, 2019.