
Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2025 SCPE. Volume 26, Issues 5, pp. 1964–1973, DOI 10.12694/scpe.v26i5.4790

MULTI PATH GAIT CONTROL METHOD FOR BIPEDAL ROBOTS BASED ON DEEP REINFORCEMENT LEARNING

LEHUI LIN *AND PINGLI LV †

Abstract. We propose a multipath gait control strategy based on deep reinforcement learning (DRL) for bipedal robot motion planning on diverse and challenging terrains. Traditional control methods, such as PID controllers and model-based motion planning, often struggle in complex environments. These approaches typically underperform because they rely on precise mathematical models or predefined rules, making them ill-suited for nonlinear, uncertain, and dynamic settings. Conventional techniques also have difficulty adapting their control strategies in unpredictable and fluctuating terrains, where robots may encounter unforeseen disturbances, leading to instability or failure.

Deep reinforcement learning is able to independently acquire optimal control methods from environmental feedback without requiring a precise model since it combines deep learning and reinforcement learning. In this work, we leverage deep reinforcement learning algorithms (DDPG, TRPO, PPO, A3C, SAC, etc.) based on actor-critic (AC) architectures to enable reliable gait control of bipedal robots in a continuous motion environment. The issue that traditional approaches have in challenging to converge complicated environments is solved by DRL, which, when compared to traditional methods, can effectively cope with the high nonlinearity of complex terrain and adaptively alter the strategy through continuous contact with the environment.

Using goal-conditional techniques, we created a motion planning model and tested it on the actual hardware platform Cassie. According to the experimental results, the approach successfully transfers the simulation strategy to the actual environment, and the robot can accurately complete the goal task without global location feedback. It can also perform a variety of complex tasks, like jumping on discontinuous and flat terrain. Furthermore, the method exhibits significant robustness and adaptability through multithreaded asynchronous training and randomized strategy selection, which solves the shortcomings of conventional motion planning methods in hyperparameter tuning and strategy convergence.

Key words: Reinforcement learning; Bipedal robot; Multi path gait control; Actor critic; Robot robustness

1. Introduction. The potential use of bipedal robots in a variety of complicated contexts has steadily gained significant attention as a result of the rapid advancement of robotics technology [1]. Nevertheless, bipedal robot motion control remains extremely difficult, particularly in dynamic, uneven, or uncharted territory [2]. Robust gait control techniques are essential for bipedal robots to be able to move independently and adapt to a variety of situations [3].

Traditional robot control approaches, such as pre-defined motion trajectories, reveal difficulties when coping with complicated environmental changes. This is because the robot might not be able to finish the task if the environment changes or if there are inaccuracies in the sensor data, as this approach usually depends on exact sensor data and correct environment modeling [4]. Consequently, in an effort to attain more autonomous gait control, an increasing number of researchers have focused on deep reinforcement learning (DRL) and other reinforcement learning-based control techniques in recent years.

Among various DRL algorithms, the actor critic framework based algorithm has received widespread attention due to its superior performance in continuous action space. Deep Deterministic Policy Gradient (DDPG), Trust Region Policy Optimization (TRPO), Near End Policy Optimization (PPO), Asynchronous Advantageous Actor Critic (A3C), and Soft Actor Critic (SAC) are a few examples of this kind of technique [5]. These algorithms each have their own characteristics and are suitable for different task scenarios. For example, DDPG has been successfully applied in continuous action planning for mobile robots by using two neural networks to estimate the policy function and value function separately [6]. However, DDPG is prone to poor convergence in practical applications [7]. TRPO improves the convergence of the algorithm by optimizing the step size of policy updates, but its high computational complexity limits its application scope [8]. PPO simplifies the calcu-

*Dongbei University of Finance and Economics, Dalian 116025, China (linlehui@dufe.edu.cn).
†Xuzhou Industrial Vocational and Technical College, Xuzhou 221140, China.

lation process of TRPO, although it improves computational efficiency, there are still shortcomings in terms of exploration [9]. A3C improves training efficiency through multi-threaded parallel training, but requires a large amount of training data, which increases the cost in practical applications [10]. SAC improves the sensitivity of the policy gradient algorithm to hyperparameters and enhances the robustness of the algorithm by introducing the maximum entropy strategy [11].

While these DRL algorithms work well for controlling robot motion, there are still significant obstacles when trying to directly apply them to controlling the gait of bipedal robots. First of all, it is challenging to directly use known algorithms in real-world applications due to the high dimensionality and complexity of bipedal robot motion control. Second, there is an urgent need to find a solution to the issue of how to guarantee the algorithm's robustness and universality while transferring between various jobs and contexts. Furthermore, present algorithms still perform inadequately when it comes to handling the impacts of uncertainty and randomness in the motion of bipedal robots in dynamic situations.

This research suggests a deep reinforcement learning-based multi-path gait control technique for bipedal robots as a solution to these problems. This method aims to enhance the adaptability and stability of bipedal robots in different environments by introducing a strategy that combines multi-path planning and DRL. The main contributions of this article include the following aspects:

The multi-path gait control method proposed in this article not only considers the stability of the robot on a single path, but also enhances the adaptability of the robot in complex environments through multi-path planning. Multi path planning allows robots to quickly select the optimal path when facing environmental changes, thereby improving the robustness of their motion.

By combining DRL and multi-path planning, this paper constructs an adaptive control framework that enables robots to flexibly switch between different tasks and environments. This framework not only considers various possible environmental changes during the training process, but also has good generalization ability in practical applications.

This research carried out numerous experiments on both real hardware platforms and simulated environments to confirm the efficacy of the suggested approach. The experimental findings demonstrate that the suggested multi-path gait control approach outperforms existing DRL algorithms and conventional techniques in complicated situations. Simultaneously, the approach showed its promise in real-world applications by performing better in various tasks and situations.

2. Actor critic-based DRL motion planning. In the task of robot motion planning in continuous action space, the output of the robot is no longer a finite set of discrete actions, but the execution probability of each action in the continuous action domain [12]. The speed and acceleration changes of wheeled robots acting on the active wheels, and the output of force and torque of legged robots acting on each joint. For this type of task, the actor critic based DRL motion planning method has stronger applicability, and commonly used algorithms include DDPG, TRPO, PPO, A3C, SAC, etc., which will be introduced one by one below.

In practice, multipath control methods have shown significant advantages in several areas, especially in robot motion control in complex environments. The following are some typical application cases to demonstrate how the method can solve real-world motion control problems and its potential application areas:

Rescue robots: After natural disasters or accidents, rescue robots are often required to perform tasks in complex terrains, such as rubble piles, mountain slopes, or forest environments. Traditional motion control methods often perform poorly in such uncertain and changing environments. In contrast, deep reinforcement learning-based multipath control methods are able to adapt to various complex terrains through continuous learning and adjustment. Through training and strategy optimization in different terrains, the rescue robot can achieve more stable and flexible motion control, easily cross obstacles, quickly reach the disaster site, and improve rescue efficiency.

Industrial robots: In modern production lines, industrial robots often need to perform high-precision motion planning in limited space, such as handling heavy objects and assembling parts. Multipath control methods in such scenarios allow for more flexible and efficient operations based on real-time changes in the environment, such as avoiding obstacles or quickly adjusting paths. In addition, when the robot is performing multiple tasks, the algorithm is able to effectively respond to the switching of complex tasks through multipath planning, ensuring the efficiency and continuity of industrial production.

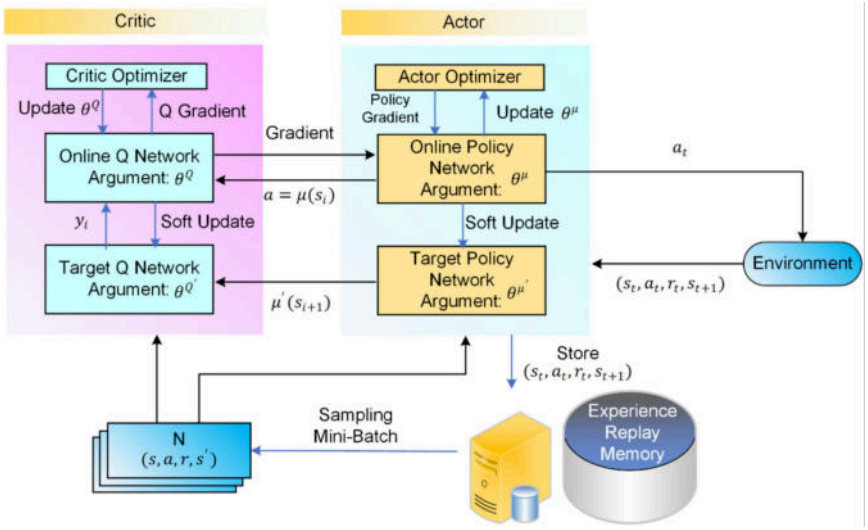

Fig. 2.1: Motion Planning Method Based on DDPG Algorithm.

Medical robots: During surgery, medical robots require extreme precision and flexibility, especially when operating complex surgeries. Motion planning methods based on multipath control can simulate and learn from the anatomical structures and surgical requirements of different patients to provide optimal operation paths for surgical robots while avoiding errors. The application of this technology can significantly improve the success rate of surgery and reduce the operation time.

Driverless vehicles: Driverless technology also faces the challenge of complex environments, especially path planning in dynamic traffic scenarios. Multipath control methods can help driverless vehicles adaptively adjust driving paths in complex urban roads, safely avoid pedestrians and obstacles, and optimize driving routes through learning to improve driving efficiency and safety. These typical application cases show that the multipath control method has obvious practicality and advantages in solving robot motion control problems in complex environments, and its potential application areas cover many key fields such as rescue, industry, medical care, and autonomous driving.

2.1. DDPG. A model free deep deterministic policy gradient algorithm (DDPG) based on the actor critic (AC) algorithm framework is proposed for motion planning of mobile robots in continuous action spaces, which has the ability of continuous action planning for mobile robots [13].The DDPG algorithm constructs four neural networks for mobile robots, while estimating the policy function and value function, including the target network of actor and critic and the current network, respectively. The actor's current network is responsible for iteratively updating the robot strategy network parameter θ, and the parameter θ' in the actor's target network is copied from the current network based on a fixed step size; The current network of critic is responsible for iteratively updating the value network parameter w and calculating the current Q value. The target network parameter w' of critic is periodically copied from w. The motion planning method for mobile robots is shown in Fig.2.1.

DDPG uses memory replay units for random selection, breaking the correlation of data and improving algorithm efficiency. It adopts a deterministic strategy to directly output the action corresponding to the maximum value function of the mobile robot, and defines the optimization objective function of DDPG algorithm as:

$$J(\theta^\mu) = E_{\theta^\mu}\left[r_1 + \gamma r_2 + \gamma^2 r_3 + \cdots\right] \tag{2.1}$$

where μ is the parameter of the strategy network that generates deterministic actions.

With the use of a dual network structure and priority experience replay mechanism, DDPG has successfully

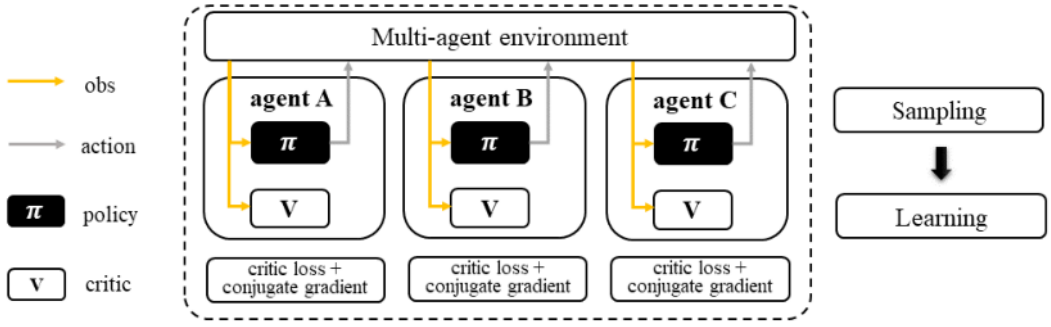

Fig. 2.2: Motion Planning Method Based on TRPO Algorithm.

tackled the problems of actor critic convergence and continuous control of mobile robots [14].

2.2. TRPO. In the optimization process of DDPG strategy gradient, the update step size will directly determine whether the mobile robot can quickly and accurately reach the target point [15]. Some researchers have suggested a reinforcement learning algorithm (TRPO) based on trust domain strategy optimization to determine the proper step size [16]. TRPO is based on calculating the KL dispersion range between the new strategy and the old strategy, with the goal of maximizing the difference between the action value function and the state value function, that is, taking the advantage function as the objective. The objective function is defined as:

$$J_{TRPO}^{\theta'}(\theta) = E_{\pi_\theta}\left[\pi_\theta / \pi_{\theta'} A^{\theta'}(s_t, a_t)\right] \tag{2.2}$$

Among them, θ and θ' are the network parameters of the new and old strategies, respectively.

A motion planning technique for a mobile robot based on the TRPO algorithm is shown in Fig.2.2. TRPO outputs an action probability distribution, and the mobile robot can optimize its motion parameters based on this probability distribution.

TRPO (Trust Region Policy Optimization) addresses the challenge of determining the appropriate update step size in policy gradient methods. It does so by using a trust region to ensure stable updates, which prevents large, destabilizing changes to the policy. However, in practice, the implementation of TRPO involves certain approximations that can complicate the process. These approximations often lead to labor-intensive calculations and, in some cases, substantial inaccuracies, particularly when the trust region is not well-calibrated. Despite its theoretical advantages, the computational complexity and potential for approximation errors make TRPO challenging to apply in real-time systems or environments with high-dimensional state spaces. These limitations highlight the need for further optimization of the algorithm to improve both accuracy and computational efficiency.

2.3. PPO. In response to the problem of excessive complexity in the calculation process of TRPO algorithm, which leads to deviation between the motion planning path and the optimal path of mobile robots, some scholars have proposed a reinforcement learning algorithm (PPO) for proximal strategy optimization, which enables mobile robots to have better exploratory ability in action selection [17]. The PPO algorithm directly uses the $KL(\theta, \theta')$ divergence of the old and new strategies as the penalty term, and its objective function is updated to:

$$J_{PPO}^{\theta}(\theta) = E_{\pi_j}\left[\frac{\pi_\theta}{\pi_\theta} A^{\theta}(s_t, a_t)\right] - \beta KL(\theta, \theta') \tag{2.3}$$

where β is the parameter for updating dispersion. Compared with the TRPO reinforcement learning algorithm, the PPO algorithm significantly simplifies the calculation steps of TRPO while retaining the exploration method of random strategies. In the case where the sampled samples satisfy the maximum likelihood probability, the robot's motion planning method will have better exploration and robustness [18].

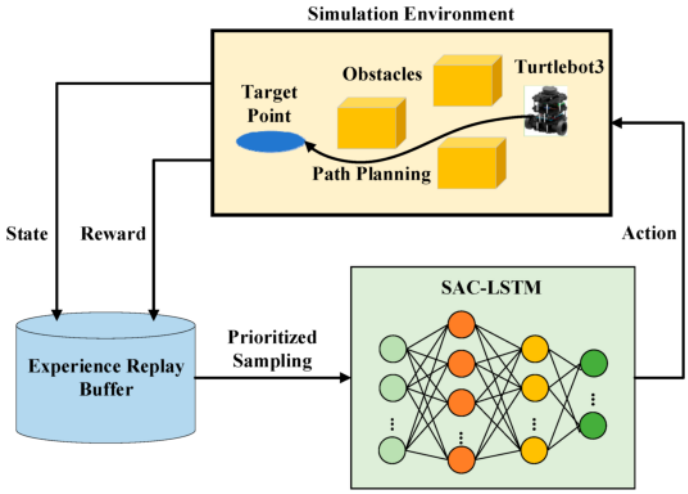

Fig. 2.3: SAC Algorithm-Based Robot Motion Planning Principle.

2.4. SAC. SAC is an offline learning reinforcement learning method based on maximum entropy, and mobile robots also use random strategies to select actions, which can learn from past experiences as well as other task experiences. As shown in Fig.2.3, the reinforcement learning motion planning method based on SAC maximizes the objective function through the maximum entropy method, where the objective function based on maximum entropy is:

$$J(\theta) = E_{\pi}\left[\sum Q(s_t, a_t) - \hat{\alpha} log \pi(a_t|s_t)\right] \tag{2.4}$$

Among them, $\hat{\alpha}$ are entropy regularization coefficients, which contain dynamic parameters from the real robot system.

The SAC motion planning algorithm uses a strong learning framework to reduce the need for hyperparameter change under the entropy constraint. In order to maximize expected return on the mobile robot's objective function, it updates the action selection strategy network of the robots using regularization coefficients rather than hyperparameters. This increases the policy gradient method's sensitivity and weak convergence to hyperparameters, improving the stability and applicability of the SAC motion planning algorithm in real-world settings. For legged and other mobile robots, it can quickly learn and plan walking skills.

3. Experiments. We have successfully deployed simulated target condition strategies and flat ground strategies for different target tasks on Cassie hardware, covering tasks such as jumping to different positions and turning directions, as well as discrete terrain strategies for variable positions and heights.

This section aims to verify two hypotheses: 1) whether the simulated training strategy can perform the same tasks in the real world; 2) Can the target condition strategy stably utilize the learned tasks to control the robot after being transferred to reality. The robot did not receive global position feedback in any of the studies, so once it moves, it will not be able to determine how far it is from the target landing location or the earth.

1. Validation of Hypothesis 1: Whether a simulation-trained strategy can perform the same task in reality.

Experimental Goal: To verify whether the strategies trained in the simulation environment can be directly applied to real robot systems to ensure that the robots perform well when performing the same tasks in the real environment.

Experimental Design:

Experimental conditions: In the simulated environment, the robot will be trained on multiple tasks, including operations such as walking, jumping and steering on different terrains. After the training is completed, the same tasks will be executed directly on the real hardware platform using the

same strategy, and the simulated training environment matches the dynamic characteristics of the real hardware environment.

Experimental Tasks:

Task 1: Jumping to a fixed target point. The robot needs to jump from its starting position to a fixed target point 1 meter away, the goal of the task is to test its accuracy in completing the task in both simulated and real environments.

Task 2: Backward jump. The robot needs to jump backward 0.3 meters from the starting position and accurately land on the ground marking point to verify whether the strategy can be stably executed in different directions of motion.

Task 3: In-situ steering and jumping. The robot first turns -60 degrees in place and then jumps back to the starting position to test its adaptability in complex motions.

Experimental Steps:

Simulation training phase: complete training of the robot on all tasks in a simulated environment, using deep reinforcement learning strategies for optimization until the robot demonstrates efficient action planning and execution in the simulated environment.

Realistic task execution phase: the trained strategies are applied to the real robot platform, and the above three tasks are repeated under the same physical conditions (same start position and goal position). Record the robot's movement trajectory, jumping distance and landing point, and compare the accuracy, time and stability of task completion in the simulated and real environments.

Validation metrics:

Task success rate: whether the robot successfully completed the tasks in the real environment.

Jumping accuracy: measure the deviation of the robot from the target point in the real environment.

Motion Stability: Observe whether the robot is unstable or deviates from the original strategy during the actual task execution.

2. Validate Hypothesis 2: Whether the goal-conditional strategy can control the robot stably after transferring to reality.

Experimental Goal: To verify whether the robot trained by the goal-conditional strategy can stably execute the learned task in the real environment, especially when facing complex terrain.

Experimental Design:

Experimental conditions: The goal-conditioned strategy will be trained on a variety of complex tasks in a simulated environment that contains irregular terrain and different task requirements, such as jumping to target points at different heights or crossing obstacles at different heights.

Experimental Tasks:

Task 1: Jumping to a 1-meter distant target point. The robot is required to jump 1 meter from the starting point to a target point on flat ground to verify its stability under target conditions.

Task 2: Jump to 1.4 meter away target point. Increase the jumping distance to verify whether the target condition strategy can maintain accuracy and stability under increasing distance.

Task 3: Jump to target points at different heights. The robot needs to jump to a 0.44-meter-high platform to test its adaptability under different height conditions.

Experiment Steps:

Training in the simulated environment: the robot is trained using goal-conditional strategies to ensure that the robot learns the corresponding strategies under different tasks and executes them accurately at different target locations and heights.

Realistic task execution phase: transfer the strategies in the simulated environment to the actual hardware platform, repeat the same jumping task, record the robot's trajectory and completion, and test its stability and accuracy in different tasks.

Validation metrics:

Task completion rate: whether the robot can complete the task in the real environment.

Target accuracy: measure the accuracy of the robot in different jumping distance and height tasks.

Stability and Adaptability: assess the robot's ability to adapt in the real world by observing its performance in different complex environments.

Analysis of experimental results : By comparing the results of the robot's task execution in the simulated and real-world environments, the following points can be verified: Whether the simulation training strategy is able to achieve seamless transfer in reality and the stability of task execution.
Whether the target conditioning strategy is able to maintain high accuracy and stability under complex terrain conditions, especially when the jumping distance and height change. These experiments are designed to verify the performance and stability of the simulated strategies in real-world environments, thus better explaining the two hypotheses in the article.

3.1. Execution of Realistic Tasks. Initially, we conducted three challenges to evaluate the robot's flat ground strategy: Jumping forward to a target point one meter ahead, jumping 0.3 meters backward, and jumping in place after turning negative sixty degrees.

These tests highlight two major benefits of the suggested approach: Initially, it can adapt to different system dynamics and transition from simulation to real environments; Secondly, it can flexibly deviate from the reference motion trajectory and complete predetermined tasks through multiple touchpoints.

Next, we validated the performance of the discrete terrain strategy in three tasks: jumping 1 meter forward, jumping 1.4 meters forward, and jumping to a target 0.88 meters ahead with a height of 0.44 meters. This tactic illustrates the capacity to accurately manipulate the robot to land on a specific target,. The robot took off and landed later during the 1.4-meter jump compared to the 1-meter jump, suggesting that the robot needs longer flight phases and higher takeoff speeds to land at further places.Furthermore, to ensure accuracy in height, the robot made a more vertical jump and raised its leg position when jumping to a platform 0.44 meters high . This method can still stabilize the robot's performance at various landing places even with an early landing time.

These experimental results indicate that the proposed strategy can flexibly adapt to the dynamic characteristics of robot hardware and achieve precise target landing by adjusting takeoff attitude and acceleration. This ability is particularly critical when dealing with ballistic movements during flight, as small errors during takeoff can lead to significant deviations in landing position.

3.2. Semi sealed gap. To further reveal the difficulty of the successful jumping experiment of the robot in Fig. 3.1, we conducted a thorough analysis of the gap between simulation and the real world. Fig.3.1 shows the joint position changes when the robot performs a turning task . Comparing the recorded data, it was found that there is a significant deviation between the joint positions of the robot in the simulated environment (blue curve) and the actual positions in the real world (red curve).

For example, the maximum deviation of the tarsal joint connection position q_6 exceeds 0.35 radians. Considering that this joint is not driven by a leaf spring with a nominal stiffness of 1250 Nm/rad, this deviation has a significant impact on the dynamic performance of the robot. Similar deviations also occur in other key joints, such as the rotation joint q_2, thigh joint q_3, and knee joint q_4, which play crucial roles in jumping and turning processes. In addition, similar differences were observed in other experiments using discrete terrain strategies. These differences emphasize the huge gap between simulation and the real world, but also demonstrate that despite such significant differences.

As shown in Fig.3.1, when the robot executes the command to jump and rotate -60 °, its joint positions show significant differences between simulated and real environments. For example, the tarsal joint driven by passive leaf springs exhibits significant deviations during the transition from simulation to actual operation. In addition, the flight phase in the real environment is significantly delayed compared to the simulated world.

3.3. Diversified and Strong Strategies under Target Condition Policies. To further overcome the limitations of the proposed control strategy, we conducted more complex dynamic jump experiments, as shown in Fig. 3.3. In these multi axis jumps, robots exhibit more complex motion behaviors. For example, in Fig. 3.3, the robot tilts laterally while jumping forward, and rotates -45 ° after landing, accurately landing at the target position 0.5 meters in front of the starting point and 0.2 meters to the left. In some challenging tasks, as shown in Fig. 3.3, the robot adjusts its body posture through small jumps after unstable landing to maintain balance.

In addition, to verify the robustness of the strategy, we applied a backward perturbation force at the pelvic vertex of the robot and tested its performance in dynamic environments, as shown in Fig.3.3. After

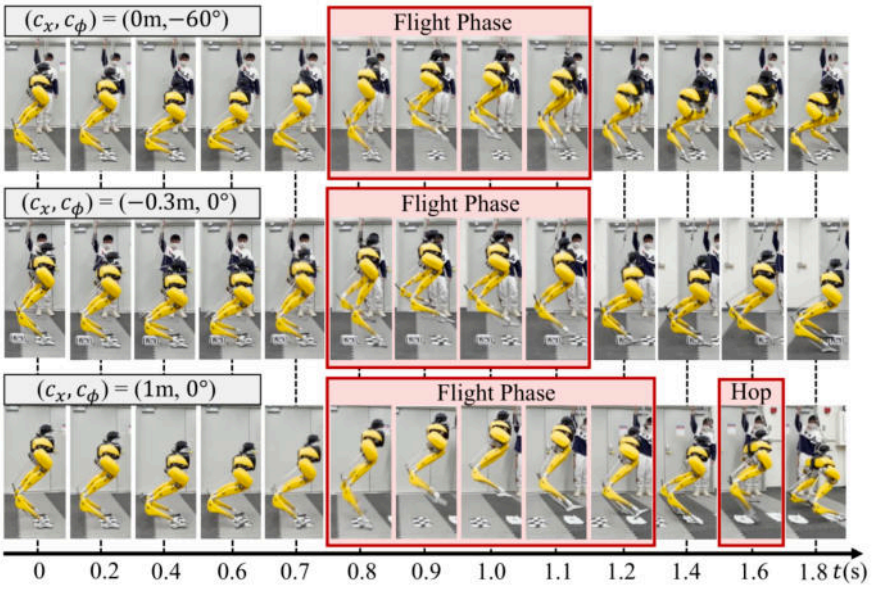

Fig. 3.1: Simulation differences in robot jumping and rotation -60 °.

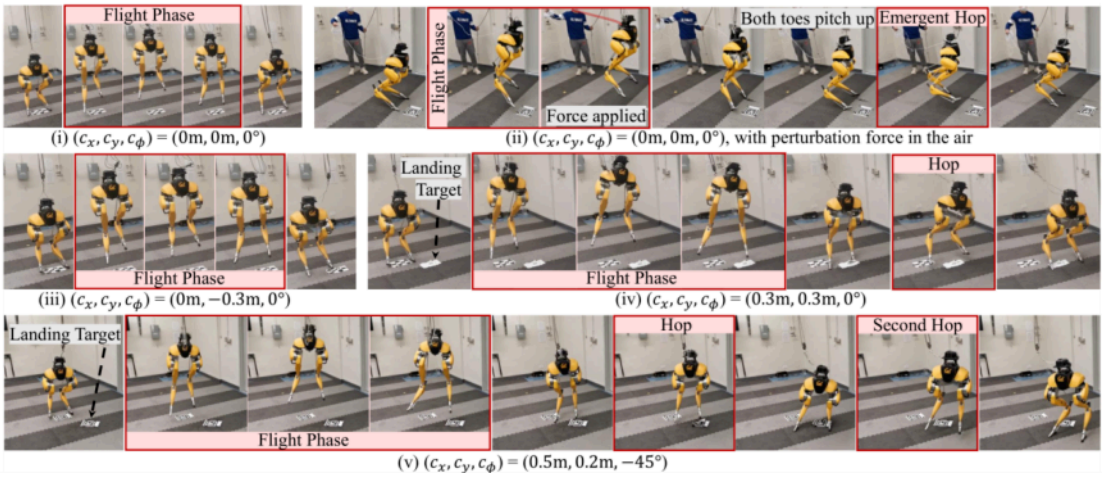

Fig. 3.2: Different jumps using flat ground strategy.

being disturbed, the robot tilted backwards during descent and caused its toes to tilt upwards during landing, resulting in insufficient driving force at the contact point. However, robots quickly adjust themselves by jumping backwards, a response learned in multi-objective training. During this jumping process, the robot is able to readjust its posture during the flight phase, achieving stability upon landing. The original goal of this test was to jump in place, but interestingly, the robot deviated from its initial task to avoid falling.

4. Conclusion. In this study, we propose a deep reinforcement learning (DRL) based multipath gait control technique for bipedal robots to overcome the difficulties of motion planning over complex terrain. The irregularity and variety of the terrain present a challenge to established control approaches, and the methods' limitations in terms of resilience and adaptation in continuous action space are just two of the many challenges

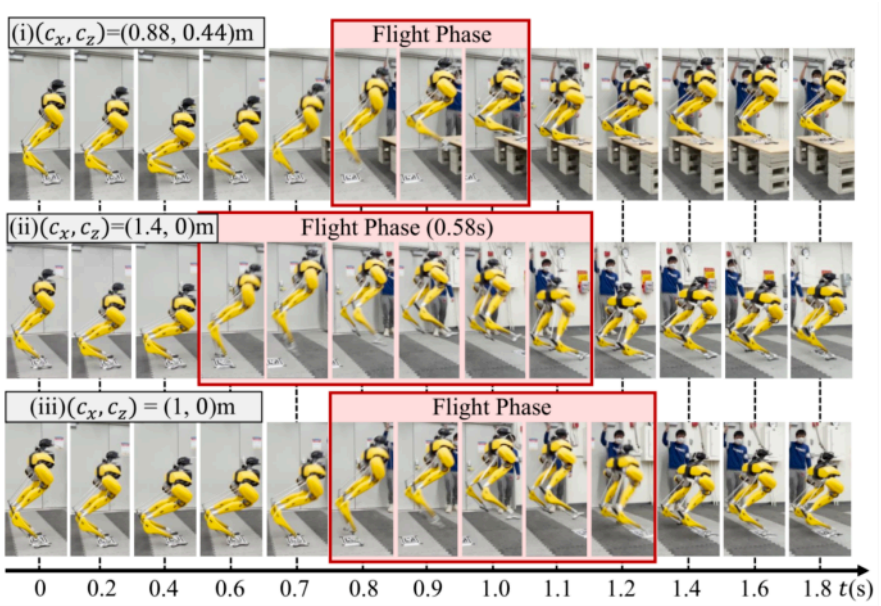

Fig. 3.3: Different jumps using discrete terrain strategy.

that come with motion planning in complex terrain. These factors hinder the robot's ability to walk steadily and accurately. We achieve accurate gait control of robots in continuous motion space by methodically examining various deep reinforcement learning algorithms based on actor-critic structure (e.g., DDPG, TRPO, PPO, A3C, SAC, etc.). The method's superior adaptability and robustness in a variety of complex tasks and environments, along with its ability to migrate strategies trained in a simulated environment to the real hardware platform Cassie, are demonstrated by the experimental results. This improves the performance of traditional methods in complex terrains to a significant degree.

Data Availability. The experimental data used to support the findings of this study are available from the corresponding author upon request.

Funding Statement. Xuzhou Industrial Vocational and Technical College High level Talent Research Launch Project Project name: Research on Deep Reinforcement Learning Method Based on Value Function Exploration Strategy Project Number: XGY2021EG02

REFERENCES

[1] Roka, S., Diwakar, M., Singh, P., & Singh, P. *Anomaly behavior detection analysis in video surveillance: a critical review.* Journal of Electronic Imaging, (2023), 32(4), 042106-042106.

[2] Wang, Y., Ren, Y., & Yang, J. *Multi-Subject 3D Human Mesh Construction Using Commodity WiFi.* Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, (2024), 8(1), 1-25.

[3] Shen, L. H., Feng, K. T., & Hanzo, L. *Five facets of 6G: Research challenges and opportunities.* ACM Computing Surveys, (2023), 55(11), 1-39.

[4] C. Zhang, M. Li and D. Wu. *"Federated Multidomain Learning With Graph Ensemble Autoencoder GMM for Emotion Recognition".* IEEE Transactions on Intelligent Transportation Systems, vol. 24, no. 7, pp. 7631-7641, July 2023, doi: 10.1109/TITS.2022.3203800.

[5] Rinchi, O., Ghazzai, H., Alsharoa, A., & Massoud, Y. *Lidar technology for human activity recognition: Outlooks and challenges.* IEEE Internet of Things Magazine, (2023), 6(2), 143-150.

[6] Abuhoureyah, F., Wong, Y. C., Isira, A. S. B. M., & Al-Andoli, M. N. *CSI-Based Location Independent Human Activity Recognition Using Deep Learning.* Human-Centric Intelligent Systems, (2023), 3(4), 537-557.

[7] Jamil, H., & Kim, D. H. *Optimal fusion-based localization method for tracking of smartphone user in tall complex buildings.* CAAI Transactions on Intelligence Technology, (2023), 8(4), 1104-1123.

[8] Lian, J., Ren, W., Li, L., Zhou, Y., & Zhou, B. *Ptp-stgcn: pedestrian trajectory prediction based on a spatio-temporal graph convolutional neural network*. Applied Intelligence, (2023), 53(3), 2862-2878.

[9] Bian, S., Liu, M., Zhou, B., Lukowicz, P., & Magno, M. *Body-Area Capacitive or Electric Field Sensing for Human Activity Recognition and Human-Computer Interaction: A Comprehensive Survey*. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, (2023), 8(1), 1-49.

[10] Zoghlami, C., Kacimi, R., & Dhaou, R. *5G-enabled V2X communications for vulnerable road users safety applications: a review*. Wireless Networks, (2023), 29(3), 1237-1267.

[11] Rizk, H., Yamaguchi, H., Youssef, M., & Higashino, T. *Laser range scanners for enabling zero-overhead wifi-based indoor localization system*. ACM Transactions on Spatial Algorithms and Systems, (2023), 9(1), 1-25.

[12] Prateek, K., Ojha, N. K., Altaf, F., & Maity, S.*Quantum secured 6G technology-based applications in Internet of Everything*. Telecommunication Systems, (2023), 82(2), 315-344.

[13] Liu, Y., Yang, D., Wang, Y., Liu, J., Liu, J., Boukerche, A., ... & Song, L. *Generalized video anomaly event detection: Systematic taxonomy and comparison of deep models*. ACM Computing Surveys, (2024), 56(7), 1-38.

[14] Yang, J., Chen, X., Zou, H., Wang, D., & Xie, L. *Autofi: Toward automatic wi-fi human sensing via geometric self-supervised learning*. IEEE Internet of Things Journal, (2022), 10(8), 7416-7425.

[15] Shastri, A., Valecha, N., Bashirov, E., Tataria, H., Lentmaier, M., Tufvesson, F., ... & Casari, P.. *A review of millimeter wave device-based localization and device-free sensing technologies and applications*. IEEE Communications Surveys & Tutorials, (2022), 24(3), 1708-1749.

[16] Wu, Y.*Exploration of the Integration and Application of the Modern New Chinese Style Interior Design*. International Journal for Housing Science and Its Applications, (2024), 45(2), 28-36.

[17] Wang, W.*Esg Performance on the Financing Cost of A-Share Listed Companies and an Empirical Study*. International Journal for Housing Science and Its Applications, (2024), 45(2), 1-7.

[18] Z. Guo, K. Yu, N. Kumar, W. Wei, S. Mumtaz and M. Guizani.*"Deep-Distributed-Learning-Based POI Recommendation Under Mobile-Edge Networks"*. IEEE Internet of Things Journal, vol. 10, no. 1, pp. 303-317, 1 Jan.1, 2023.

Edited by: Ashish Bagwari
Special issue on: Adaptive AI-ML Technique for 6G/ Emerging Wireless Networks
Received: Aug 30, 2024
Accepted: Apr 21, 2025