Multi-Objective Reinforcement Learning for Efficient and Robust Gait Planning in Bipedal Robot

Haolin Ma, Zijia He, Haomiao Qiu, Kaibo Yang, Jintao Mo and Jian Fu*

Abstract—**Reinforcement learning (RL) has achieved outstanding outcomes in the field of robust locomotion of bipedal robots. Typically, a meticulously calibrated optimal reward function is used to improve the generalizability of the robot's performance in different environments. However, the metrics are often in conflict with each other, such as the trade-off between environmental robustness and energy efficiency, which is difficult to solve through traditional RL. To address this challenge, this paper proposed a multi-objective reinforcement learning (MORL) framework to train and optimize a solution set of weights corresponding to the varied metrics required for effective human-robot interactions. To reduce the increased training costs associated with expanding the dimensionality of the reward space, a reference trajectory based on the Linear Inverted Pendulum Model (LIPM) is integrated into the training program as a priori knowledge. Our method was rigorously tested through simulations in MuJoCo's humanoid environment and in our bipedal robot environment. The results demonstration that a significant improvement in performance metrics, convergence speed and loss function compared to traditional RL method. The training and evaluation scripts are available online for further exploration and application.** †

I. Introduction

Bipedal robots have distinct advantages compared to other robot forms due to their human-like motion ability. Their bipedal mobility enables efficient movement in human world, while their advanced manipulation and dexterity allows them to perform complex tasks [1], [2]. Reinforcement learning (RL) is recognized as a viable method for solving robust walking gait planning problems for bipedal robots. RL enhances locomotion capabilities by learning from data and experience. Many researchers have contributed to this field in recent years [3], [4], [5]. Designing a good and efficient reward function is essential for locomotion performance [6]. In previous work, the reward function for the robot's distance traveled was set, and several penalty terms were applied, including the center of mass (CoM) height (with larger weights) and the sum of moments at each joint (with smaller weights). However, in various applications, bipedal robots require different and sometimes conflicting metrics in different task environments, such as walking efficiency and environmental robustness. For instance, robots in the rescue and relief category prioritize travel speed over energy consumption efficiency. On the other hand, border patrol robots or space exploration robots prioritize energy efficiency as they require long endurance, and the importance of travel rate efficiency is not particularly significant. Researchers often include penalty terms in reward functions and apply different weights to them to design reward functions [7]. However, this strategy lacks scientific rigor and heavily relies on the level of tuning parameterization of reinforcement learning engineers.

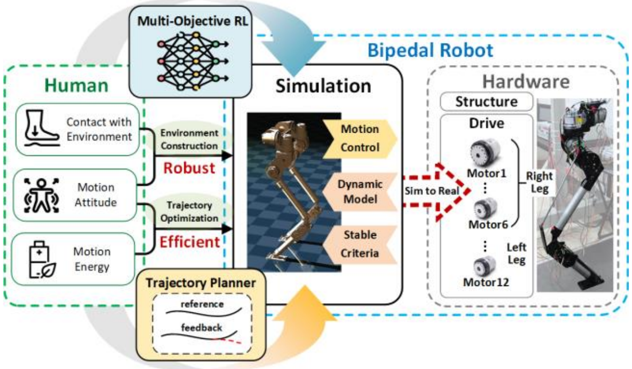


Fig. 1. We're developing a bipedal robot in MuJoCo, leveraging the MORL method with trajectory planner for gait planning. Our ongoing hardware development will facilitate sim-to-real transition.

Multi-objective reinforcement learning (MORL) approach is used to solve problems with multiple competing or mutually independent objectives by representing different performance metrics [8]. It is important to note that some researchers argue that modelling problems as multi-objective (MO) is unnecessary and that all rewards can be represented as a single scalar signal. Hayes et al. [8] propose six motivating scenarios for the application of bipedal robots in the future. Two of these scenarios are the decision support scenario and the dynamic utility function scenario. The decision support scenario is relevant when the user's preferences are unknown or difficult to determine. The dynamic utility function scenario is relevant when the user's preferences for certain objectives change over time.

In addition, RL often includes numerous interference tests during training to ensure result robustness. However, this approach can lead to high training costs and slow convergence. Furthermore, the high-dimensional controllers used in sim-to-real exacerbate hardware computational pressure [9]. Therefore, this paper adopts model-based reinforcement learning (MBRL), while introducing linear inverted pendulum model (LIPM) as a reference model, which accelerates the learning process, improves the sampling efficiency, and achieves the small number of interactions between robot and environment and faster convergence to the optimal solution [10].

This paper applies MORL to gait planning for bipedal robots, considering both energy efficiency and robustness requirements in the application environments, which is compared to single-objective reinforcement learning (SORL).

All authors are with the School of Mechanical Engineering and Automation, Beihang University, 100191, Beijing, China. (ma207185099@gmail.com)

† https://github.com/sm-1z/BH-Dynamic-RL

Additionally, the paper introduces the robot's dynamics model and LIPM as prior knowledge to reduce the learning cost and obtain feasible results. The method was validated in the MuJoCo environment for humanoid-v4 and a self-developed bipedal robot as shown in Fig. 1.

II. Related Work

A. Multi-objective Reinforcement Learning in Robotics

MORL approach has been shown to be effective in addressing these problems. Sandy H. Huang et al. [11] combined three reward signals: extrinsic rewards from the task, intrinsic rewards from impact, and curiosity. The method was applied to ensure that the robot hand interacts gently with fragile objects through simulation and experimentation. However, this research did not introduce any MORL algorithms, but rather utilized the ideas of MORL. Several MORL algorithms have been developed for continuous robotics control, including envelop MORL (EMORL) [12], prediction-guided MORL (PGMORL) [13], and evolutionary multi-objective game intelligence (EMOGI) [14], etc. Additionally, new robotics control frameworks have been proposed, such as the constrained MORL algorithm for personalized end-to-end robotic control with continuous actions by Xiangkun He et al. [15]. Tran et al. [16] developed the MOPDERL framework, which is a novel MO proximal distilled evolutionary reinforcement learning framework consisting of a warm-up stage and an evolution stage. However, the verification of this framework was only conducted using simple benchmarks supported by Gymnasium, which may not accurately reflect the algorithm's performance on realistic robot platforms.

B. Model-based Reinforcement Learning in Robotics

Environmental adaptation problems in robotics can often be solved using RL algorithms. MBRL appears to be the most effective approach for such tasks, as it requires less interaction with the environment compared to model-free RL (MFRL) by utilizing transition models. This reduction in interaction also minimizes the risk of accidents and damage to the robot [10]. MBRL approaches have been used for bipedal walking in simplified robots [17] or small-size humanoid robots [18]. However, these approaches cannot reflect the whole-actuated bipedal robot. Cheng-Yu Kuo et al. [19] presented a probabilistic MBRL for learning the energy exchange dynamics of a spring-loaded biped robot. The results showed successful on-site walking acquisition with a compact nine-dimension dynamics model, 40 Hz real-time planning, and on-site learning within a few minutes.

III. Preliminaries

In this section, we outline the fundamental architecture and dynamic framework of our bipedal robot. Subsequently, we employ the 3D-LIPM to establish a reference gait. This approach significantly streamlines the training duration and reduces associated expenses for the forthcoming MORL.

A. Description of Robot

Based on the human biological structure, a series-parallel hybrid configuration of lower limb structure of humanoid robots was designed as shown in Fig. 1. The robot mainly includes three parts: hip, knee and ankle, with 12 active degrees of freedom (DOF) and 4 passive DOF. We adopt the passive DOF driven by the elastic element to reduce the driving torque when the foot hits the ground, which introduces the passive flexibility to the robot. And we use the multi-bar mechanism for transmission to reduce the moment of inertia of the robot. The hip joint has three active rotation DOF [q_{hroll}, q_{hyaw}, q_{hpitch}], which can drive the whole leg to rotate in three directions. The knee joint adopts parallelogram-like multi-bar mechanism, which has 1 active DOF [q_{kpitch}] and 2 passive DOF [q_{kp1}, q_{kp2}], designed to drive the lower leg pitch. And the ankle joint adopts a parallel rod mechanism to realize it has 2 active DOF [q_{apitch}, q_{aroll}], which drive the robot's foot to pitch and roll around the universal joint of the cross axis.

B. Dynamic Model

1) Forward Kinematics

Based on the structure of our bipedal robot model, the kinematic model is constructed using the Denavit-Hartenberg (D-H) method, as illustrated in Fig. 2. Since the leg configuration involves parallel mechanisms and passive joints. Treating the knee joint q_{kp2} as an active driving joint. The seven pose transformation matrices with added constraint relationships are as follows:

$$\begin{gathered} {}^4_5\boldsymbol{T} = {}^4_5\boldsymbol{T}_0\,{}^5_6\boldsymbol{T}_0\,{}^6_7\boldsymbol{T}_0 \quad {}^5_6\boldsymbol{T} = {}^7_8\boldsymbol{T}_0 \quad {}^6_7\boldsymbol{T} = {}^8_9\boldsymbol{T}_0 \\ {}^1_7\boldsymbol{T} = {}^1_2\boldsymbol{T}\,{}^2_3\boldsymbol{T}\,{}^3_4\boldsymbol{T}\,{}^4_5\boldsymbol{T}\,{}^5_6\boldsymbol{T}\,{}^6_7\boldsymbol{T} \end{gathered} \tag{1}$$

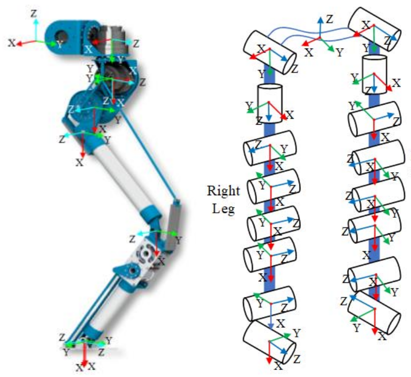


Fig. 2. D-H modeling structure for humanoid biped robot

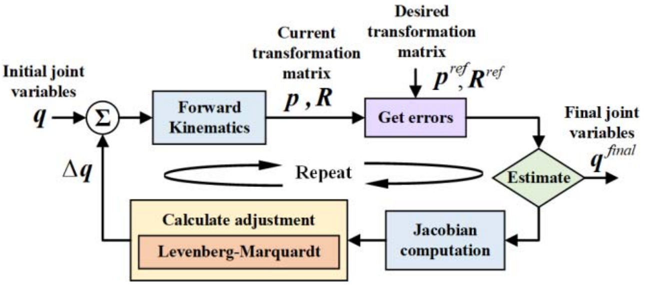


Fig. 3. Inverse kinematics numerical solution schematic diagram

2) Inverse Kinematics

The inverse kinematics are solved using numerical methods. The iterative process involves incorporating the damped Gauss-Newton method (Levenberg-Marquardt method) for result refinement as shown in Fig. 3.

C. *Stable Criteria*

The LIPM is a commonly used gait planning model for humanoid robots. It models walking as an inverted pendulum's oscillation around a pivot, with stability maintained through control of the zero moment point (ZMP).

3) *Centroid calculation*

According to D-H parameters and the CoM of the link, the centroid of the robot is calculated. The total CoM **c** can be obtained by dividing the sum of the moment by the total mass:

$$\boldsymbol{c} = \sum_{j=1}^{N} m_j \boldsymbol{c}_j / M \tag{2}$$

4) *3D-linear inverted pendulum*

The gait of the robot can be simplified to the 3D-LIPM during the one-legged support period. When a height constraint plane for the CoM is given, the robot's CoM can move within the X-Y plane during gait motion. A schematic diagram of the 3D-LIPM is shown in Fig. 4.

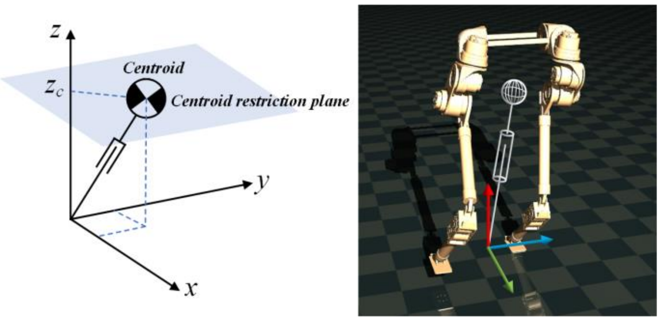


Fig. 4. Simplified model of 3D-LIPM

To maintain stability and prevent the robot from falling over during motion, the equation of motion for a 3D-LIPM can be derived as:

$$\ddot{x} = (g/z_c)x \quad \ddot{y} = (g/z_c)y \tag{3}$$

By solving the differential equation through Laplace transform and combining the equation of the constraint surface, the trajectory of the CoM can be determined:

$$\begin{cases} x(t) = x(0)\cosh(t/T_c) + T_c\dot{x}(0)\sinh(t/T_c) \\ y(t) = y(0)\cosh(t/T_c) + T_c\dot{y}(0)\sinh(t/T_c) \\ z = k_x x + k_y y + z_c \end{cases} \tag{4}$$

Where $T_c \equiv \sqrt{z/g}$ is a constant of the ratio of gravity acceleration to the height of the CoM;

A biped support phase with a long period of T_{dbl} is inserted when the support feet are switched. The position of the CoM is described by forth order polynomials [20]:

$$x(t) = \sum_{i=0}^{4} a_i t^i \tag{5}$$

Where the coefficients a_i are determined by position, velocity and acceleration of the CoM at the instant of support exchange.

IV. Multi-Objective Reinforcement Learning Methods

In this section, we detail the MORL algorithms utilized in our study and elucidate the construction of our learning framework. Subsequently, we conduct a comprehensive performance evaluation of the MORL to ascertain its efficacy and robustness within the context of our research.

A. *CAPQL Algorithm*

In the SORL domain, algorithms such as twin delayed DDPG (TD3)[21] and soft actor-critic (SAC)[22] are all predicated under the actor-critic framework. Building upon this foundation, TD3 introduces twin Q-networks and delayed policy updates to mitigate overestimation biases and enhance the stability of learning. SAC integrates the concept of maximum entropy into the actor-critic architecture, optimizing not only the expected reward but also the diversity of exploration.

The MORL problem, however, due to varying preferences among different objectives, there exist multiple solutions to the same problem. CAPQL is one of the MORL methods, which learns policy and Q-networks conditioned on weight vectors, extending the SAC to the realm of MORL. It is trained based on the theory of the Pareto front, enabling optimization and solution finding for various weight vectors. The algorithmic workflow of CAPQL is illustrated in Algorithm 1

Algorithm 1: Concave-Augmented Pareto Q-Learning

```
Input: $\lambda, \alpha, \gamma, \tau, L, N, D_\phi$
Initialize Network parameter vectors $\theta_1, \theta_2, \bar{\theta}_1, \bar{\theta}_2, \psi$ , $\bar{\theta}_i \leftarrow \theta_i$ for $i \in 1,2$
Initialize Replaybuffer $D$
for each episode do
    for each step do
        if warm-up do
            sample $w \sim D_\psi$
            $a_t \sim \pi_\psi(s_t, w)$
            $s_{t+1} \sim P(a_t, s_t)$
            $D \leftarrow D \bigcup (s_t, a_t, R(a_t, s_t), s_{t+1}, w)$
        elif training do
            $S$ ←sample $N$ transitions from $D$
            $\theta_i \leftarrow \theta_i - \lambda_Q \nabla_{\theta_i} (\frac{1}{2} \mathbb{E}_S \| \hat{Q}(s_j, a_j, w) -$
                $Q_{\theta_i}(s_j, a_j, w) \|_2^2)$ for $i \in 1,2$
              where $\hat{Q}(s_j, a_j, w) = R(a_j, s_j) +$
                    $\gamma(min_{i\in\{1,2\}} Q_{\bar{\theta}_i}(s_{j+1}, a_{j+1}, w) -$
                    $\alpha log \pi_\varphi(a_{j+1}, s_{j+1}, w)$
              and $a_{j+1} \sim \pi_\psi(s_{j+1}, w)$
            $\psi \leftarrow \psi - \lambda_\pi \nabla_\psi \mathbb{E}_S (D_{KL}(\pi_\psi(\cdot, sj, w) \|$
                    $\frac{exp(w^T min_{i\in\{1,2\}} Q_{\theta_i}(s_j, \cdot, w) / \alpha)}{Z(s_j, w)}))$
              with $Z(s_j, w) = \int_A exp(w^T min_{i\in\{1,2\}} Q_{\theta_i}(s_j, a, w) / \alpha)\, da$
            $\bar{\theta}_i \leftarrow \tau\theta_i + (1-\tau)\theta_i$ for $i \in \{1,2\}$
    end
end
```

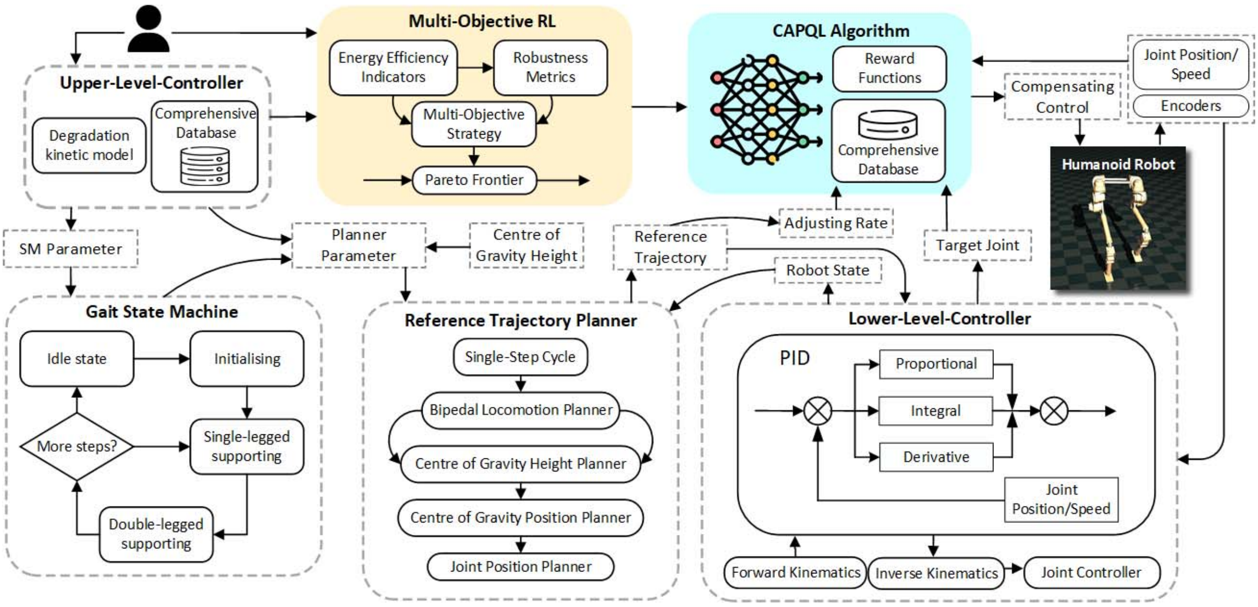


Fig. 5. MORL framework for the locomotion of bipedal robot

B. Learning Framework

Our learning framework initiate from deducing the robot's dynamical model to extract key parameters for the Finite State Machine (FSM). Leveraging these insights, we devise strategies for the robot's leg support, which are then applied to compute reference trajectories for the robot's CoM, foot placements, and joint angles, contingent upon a predefined CoM height.

Contrasting with traditional control methodologies, which rely on feeding computed reference trajectories into a lower-level controller using PID and similar techniques for precise joint regulation, our approach innovatively integrates these trajectories and the target control values into a MORL algorithm. This integration serves as a reward signal, allowing the algorithm to adaptively refine the robot's movement trajectories. The result is an enhanced gait, characterized by improved fluidity and stability, which is pivotal for the robot's effective navigation through complex and dynamic environments.

V. Results

In this section, the MORL training environment of our bipedal robot is introduced. Critical comparing environments are conducted based on this platform.

A. Gymnasium Environment Building

We used Gym API interface and MuJoCo platform to create the MORL training environment. The configuration of training environment includes the creation of robot physical environment and the realization of RL algorithm.

Utilizing SORL/MORL algorithms such as TD3, SAC, and CAPQL, we conducted these tests on two unique humanoid robot models: *humanoids-v4 (mo-humanoids-v4)*, a MuJoCo's tutorial model, and *bh-dyn-v0 (mo-bh-dyn-v0)*, our bipedal robot model, which adapted for both SORL and MORL. The information of tests is illustrated in TABLE I. In addition, the experimental results are computed on the computer with Intel Core i7-12700H @ 4.7GHz with 14 cores and NVIDIA GeForce RTX 3060 GPU.

TABLE I. The information of experiments

Index	Env	Algo	Type	Desc
Exp. 1	*humanoids-v4*	TD3	SORL	To find the effects of action noise
Exp. 2	*bh-dyn-v0*	SAC	SORL	To explore the effects of robot trace
Exp. 3	*humanoids-v4 / mo-humanoids-v4*	SAC/CAPQL	SORL/MORL	To compare SORL and MORL algorithm
Exp. 4	*bh-dyn-v0/ mo-bh-dyn-v0*	SAC/CAPQL	SORL/MORL	To explore conflict resolution capabilities of SO/MO in robustness and energy efficiency

The observation space, action space, and reward space are the basis of RL. Information of the fundamental space for the *humanoids-v4* environment can be obtained from the official Gym website [23]. In contrast, the specifics for the *bh-dyn-v0* environment are as depicted in the TABLE II. Each reward is calculated using the following method:

$$\begin{cases} R_f = K_f \times v_y \\ R_h = K_h \times \operatorname{sgn}(H) \\ R_r = -\dfrac{K_r}{2} \times \{(P_{center} - P^*_{center})^2 + \\ \qquad (P_{foots} - P^*_{foots})^2 + (P_{joints} - P^*_{joints})^2\} \\ R_e = \dfrac{K_e}{(\sum T \times w) / (d_x{}^2 + d_y{}^2)} \end{cases} \tag{6}$$

In the equation, K_f, K_h, K_r, K_e represents the weight coefficient, v_y is the velocity of the robot in the y-axis; H denotes the robot's survival status: $H=0$ if the robot is down, and $H=1$

otherwise; P signifies the current position, with $P*$ being the target position; T and w correspond to the torque and rotational speed of the joints, respectively, while d_x and d_y indicate the distance the robot moves during the current action.

TABLE II. The fundamental space of *bh-dyn-v0*

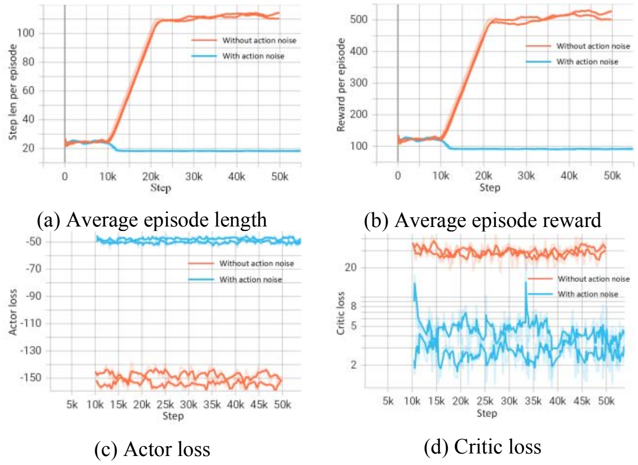
<table>
<tr><th colspan="3">Observation Space</th></tr>
<tr><th>Component</th><th>Count</th><th>Description</th></tr>
<tr><td>Waist and Joint Positions</td><td>21</td><td>Positions of the waist and each joint</td></tr>
<tr><td>Joint Velocities</td><td>20</td><td>Velocities of the joints</td></tr>
<tr><td>Rigid Body CoM and Inertia</td><td>18x10</td><td>CoM and inertia of each rigid body</td></tr>
<tr><td>Rigid Body Velocities Relative to CoM</td><td>18x6</td><td>Velocities of each rigid body relative to its CoM</td></tr>
<tr><td>Joint Torques</td><td>20</td><td>Torques applied to each joint</td></tr>
<tr><td>External Forces on Rigid Bodies</td><td>18x6</td><td>External forces acting on each rigid body based on its CoM</td></tr>
<tr><th colspan="3">Action Space</th></tr>
<tr><th>Component</th><th>Range(Nm)</th><th>Description</th></tr>
<tr><td>Hip Joint (roll, yaw, pitch)</td><td>(-40, -25, -25) ~ (40, 25, 25)</td><td>The hip is the joint that connects the thigh to the waist</td></tr>
<tr><td>Knee Joint (Active, Passive)</td><td>(-25, -25) ~ (25, 25)</td><td>The knee is the joint that connects the thigh to the shank</td></tr>
<tr><td>Ankle Joint (roll, pitch)</td><td>(-15, -15) ~ (15, 15)</td><td>The ankle is the joint that connects the shank to the foot</td></tr>
<tr><th colspan="3">Reward Space</th></tr>
<tr><th>Component</th><th colspan="2">Description</th></tr>
<tr><td><i>Forward progress reward</i></td><td colspan="2">Encourages the robot to move forward effectively.</td></tr>
<tr><td><i>Healthy reward</i></td><td colspan="2">Promotes the maintenance of the robot's operational health.</td></tr>
<tr><td><i>Robustness reward</i></td><td colspan="2">Tracking of the robot's CoM, bipedal stance, and joint positions to the given trajectory.</td></tr>
<tr><td><i>Energy efficiency reward</i></td><td colspan="2">Power consumption per unit distance traveled by the robot.</td></tr>
</table>

B. *Effects of Action Space Noise*

In Exp.1, TD3 is employed for training the *humanoid-v4* to examine the influence of action space noise on the efficacy of RL training. Throughout smoothing the result data, the following phenomena is observed as shown in Fig. 6.

In the experiments without noise, the average episode length and reward stabilized at approximately 110 steps and 500 points as shown in Fig. 6(a) and Fig. 6(b), respectively. This indicates that, in the absence of external noise, the algorithm was able to learn consistently and achieve commendable performance. Conversely, when noise was introduced, there was a significant decline in both the average episode length and reward as shown in Fig. 6(a) and Fig. 6(b), which were maintained at around 20 steps and 100 points, respectively. This outcome highlights the detrimental impact of noise on RL training outcomes, likely due to the increased uncertainty in the environment, making it more challenging for the algorithm to identify effective strategies.

Furthermore, the loss function values are noted, as depicted in Fig. 6(c) and Fig. 6(d), are lower than another. A possible explanation for this impunity is that the introduction of noise enhances the stochasticity of learning, potentially aiding the algorithm in escaping local optima and thereby improving the performance of the loss functions to some extent. In other words, while noise may lead to a decrease in performance metrics, it may also promote a broader exploration of the strategy space by the algorithm, preventing premature convergence to suboptimal strategies.



(a) Average episode length

(b) Average episode reward

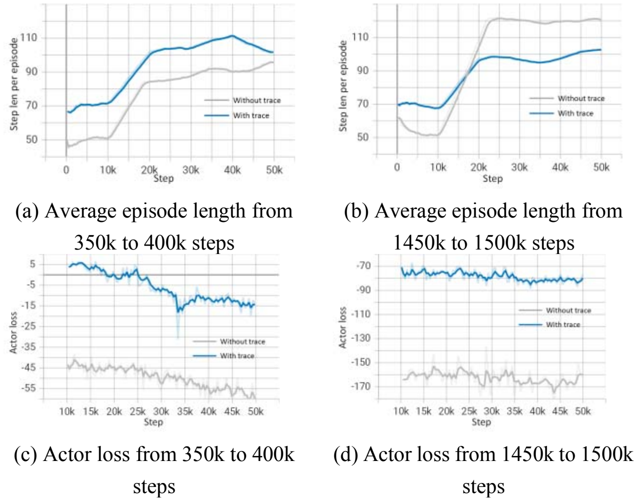
(c) Actor loss

(d) Critic loss

Fig. 6. Performance of TD3 within *humanoid-v4* from 400k to 450 steps

C. *Effects of Rewards for Robustness and Energy Efficiency*

In Exp.2, SAC is used to train the *bh-dyn-v0* model. We incorporate robustness and energy efficiency rewards into the reward evaluation framework for robotic motion trajectory optimization, aiming to assess and guide the performance of multi-objective reinforcement learning algorithms.



(a) Average episode length from 350k to 400k steps

(b) Average episode length from 1450k to 1500k steps

(c) Actor loss from 350k to 400k steps

(d) Actor loss from 1450k to 1500k steps

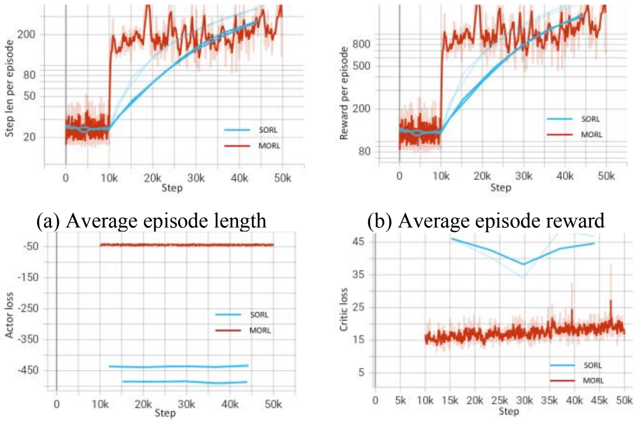
Fig. 7. Performance of SAC within *bh-dyn-v0* environment

During the initial phase of training, we observed that the trajectory-informed training exhibited a rapid increase in the average number of steps per episode, surpassing the training without trajectory incorporation as illustrated in Fig. 7(a). However, as training progressed, the episodes without trajectory integration gradually outperformed those with trajectory inclusion in terms of average steps per episode, as illustrated in Fig. 7(b). This phenomenon may be attributed to the complexity added by the introduction of robustness rewards in conjunction with energy efficiency rewards. SORL struggle to resolve the inherent conflict between robustness and energy efficiency, leading to degraded learning outcomes over time due to the persistent trade-off between these two objectives. Notably, the Fig. 7(c) and Fig. 7(d) show action loss function for the trajectory-informed training consistently remained lower than that of the trajectory-free training.

D. *Effects of SORL and MORL*

In Exp.3, we conduct a comparative analysis of the SAC within the *bh-dyn-v0* and the CAPQL within the *mo-bh-dyn-v0* (the walking process trained by CAPQL is as shown in Fig. 10(a)). The findings indicated that MORL exhibited a swift increase in performance levels at the onset of training, reflected in the metrics of average episode steps and average episodic rewards.

Notably, following the sampling phase, MORL achieved an average episode step count of about 250 and an average episodic reward nearing 1000 points. Highlighting the algorithm's effectiveness and robustness in handling complex tasks as shown in Fig. 8(a) and Fig. 8(b). As training continued, however, the trend of performance enhancement for MORL was observed to plateau, with a slower rate of improvement compared to that of SORL. This observation may point to potential areas for optimization in MORL during prolonged learning phases. Despite this, the initial rapid performance surge of MORL still underscores its potential in MO optimization tasks, especially in contexts that demand quick adaptation and response.



(a) Average episode length

(b) Average episode reward

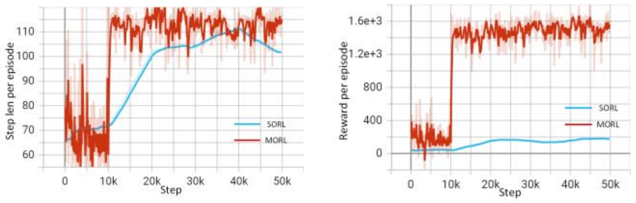
(c) Actor loss

(d) Critic loss

Fig. 8. Performance of SAC within *humanoid-v4* (from 1150k to 1200k steps) and CAPQL within *mo-humanoid-v4* (from 250k to 300k steps)

During the training process, the performance enhancement trend for CAPQL was observed to plateau, with a slower rate of improvement compared to SORL. This could indicate potential areas for optimization in MORL during extended learning phases. Nonetheless, the initial rapid surge in performance for MORL still highlights its potential for MO optimization tasks, particularly in scenarios requiring swift adaptation and response. As illustrated in Fig. 8(c) and Fig. 8 (d), MORL's loss function value diminished rapidly in the early training stages, while SORL showed comparatively higher values. The loss function value for MORL continued to decline throughout the training, remaining at a lower level for the majority of the steps. This suggests that MORL demonstrates superior performance in learning and predictive tasks, efficiently identifying and refining effective strategies, and ensuring stability throughout the learning process.

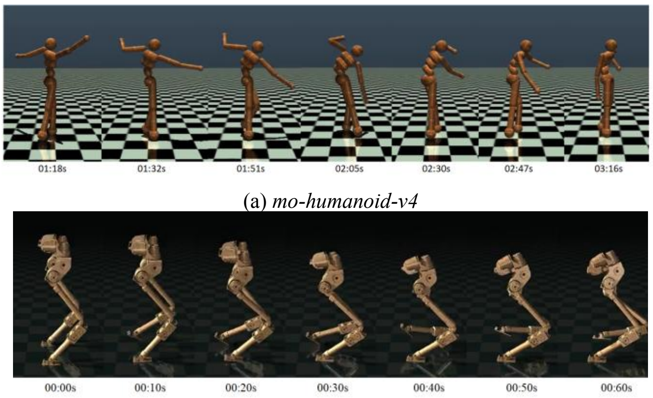
In Exp.4, the conflict resolution capabilities between robustness and energy efficiency in SORL versus MORL were investigated without the use of action noise, as depicted in Fig. 9. The CAPQL algorithm achieved an average step count of 120 and an average reward of 1400 per episode shortly after the sampling phase concludes. This rapid attainment showed that CAPQL outperforms SAC in terms of learning speed and effectiveness, which enabled the robot to achieve greater forward progress rewards.



(a) Average episode length

(b) Average episode reward

Fig. 9. Performance of SAC within *bh-dyn-v0* (from 1150k to 1200k steps) and CAPQL within *mo-bh-dyn-v0* (from 250k to 300k steps)



(a) *mo-humanoid-v4*

(b) *mo-bh-dyn-v0*

Fig. 10 The walking process trained by CAPQL algorithm

VI. Conclusion

In this paper, we present a comprehensive study on the application of MORL for efficient and robust gait planning in bipedal robots. The results show that the introduction of action space noise in RL increases training variability and reveals potentially innovative solutions, while the integration of prior knowledge can both increase the robustness of trajectories and limit the exploration of a wider solution space. Compared to SORL, MORL algorithm CAPQL effectively balances the considerations of robustness and energy efficiency for gait of bipedal robots, which can quickly achieve the stability of the reward function. The results show that MORL can provides a customized solution set that adapts to more complex RL application contexts. Future work will focus on refining the design of the reward function, exploring additional MORL scenarios, and extending the research to real-world robotic platforms to validate the practical applicability of our proposed methods.

ACKNOWLEDGMENT

We express our gratitude to the OpenAI Gym team for providing the Gymnasium platform, which served as an essential tool for developing and testing the algorithms presented in this study.

REFERENCES

[1] M. S. Khan and R. K. Mandava, “A review on gait generation of the biped robot on various terrains,” Robotica, vol. 41, no. 6, pp. 1888–1930, Jun. 2023.

[2] Y. Tong, H. Liu, and Z. Zhang, “Advancements in Humanoid Robots: A Comprehensive Review and Future Prospects,” IEEECAA J. Autom. Sin., vol. 11, no. 2, pp. 301–328, Feb. 2024.

[3] Z. Li et al., “Reinforcement Learning for Robust Parameterized Locomotion Control of Bipedal Robots,” in 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi’an, China: IEEE, May 2021, pp. 2811–2817.

[4] M. Kasaei, M. Abreu, N. Lau, A. Pereira, and L. P. Reis, “Robust biped locomotion using deep reinforcement learning on top of an analytical control approach,” Robot. Auton. Syst., vol. 146, p. 103900, Dec. 2021.

[5] B. Singh, R. Kumar, and V. P. Singh, “Reinforcement learning in robotic applications: a comprehensive survey,” Artif. Intell. Rev., vol. 55, no. 2, pp. 945–990, Feb. 2022.

[6] J. Kober, J. A. Bagnell, and J. Peters, “Reinforcement learning in robotics: a survey”.

[7] C. Huang, G. Wang, Z. Zhou, R. Zhang, and L. Lin, “Reward-Adaptive Reinforcement Learning: Dynamic Policy Gradient Optimization for Bipedal Locomotion,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 45, no. 6, pp. 7686–7695, Jun. 2023.

[8] C. F. Hayes et al., “A practical guide to multi-objective reinforcement learning and planning,” Auton. Agents Multi-Agent Syst., vol. 36, no. 1, p. 26, Apr. 2022.

[9] H. Duan et al., “Sim-to-Real Learning of Footstep-Constrained Bipedal Dynamic Walking,” in 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA: IEEE, May 2022, pp. 10428–10434.

[10] A. S. Polydoros and L. Nalpantidis, “Survey of Model-Based Reinforcement Learning: Applications on Robotics,” J. Intell. Robot. Syst., vol. 86, no. 2, pp. 153–173, May 2017.

[11] S. H. Huang et al., “Learning Gentle Object Manipulation with Curiosity-Driven Deep Reinforcement Learning.” arXiv, Mar. 20, 2019. Accessed: Mar. 03, 2024.

[12] R. Yang, X. Sun, and K. Narasimhan, “A Generalized Algorithm for Multi-Objective Reinforcement Learning and Policy Adaptation”.

[13] J. Xu, Y. Tian, P. Ma, D. Rus, S. Sueda, and W. Matusik, “Prediction-Guided Multi-Objective Reinforcement Learning for Continuous Robot Control”.

[14] R. Shen et al., “Generating Behavior-Diverse Game AIs with Evolutionary Multi-Objective Deep Reinforcement Learning,” in Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, Yokohama, Japan: International Joint Conferences on Artificial Intelligence Organization, Jul. 2020, pp. 3371–3377.

[15] X. He, Z. Hu, H. Yang, and C. Lv, “Personalized robotic control via constrained multi-objective reinforcement learning,” Neurocomputing, vol. 565, p. 126986, Jan. 2024.

[16] H.-L. Tran, L. Doan, N. H. Luong, and H. T. T. Binh, “A Two-Stage Multi-Objective Evolutionary Reinforcement Learning Framework for Continuous Robot Control,” in Proceedings of the Genetic and Evolutionary Computation Conference, Lisbon Portugal: ACM, Jul. 2023, pp. 577–585.

[17] S. Levine and V. Koltun, “Learning Complex Neural Network Policies with Trajectory Optimization”. International Conference on Machine Learning. PMLR, 2014.

[18] M. Frank, J. Leitner, M. Stollenga, A. Förster, and J. Schmidhuber, “Curiosity driven reinforcement learning for motion planning on humanoids,” Front. Neurorobotics, vol. 7, 2014.

[19] C.-Y. Kuo, H. Shin, and T. Matsubara, “Reinforcement Learning With Energy-Exchange Dynamics for Spring-Loaded Biped Robot Walking,” IEEE Robot. Autom. Lett., vol. 8, no. 10, pp. 6243–6250, Oct. 2023.

[20] Kajita, Shuuji, et al. Introduction to humanoid robotics. Vol. 101. Springer Berlin Heidelberg, 2014.

[21] S. Fujimoto, H. Hoof, and D. Meger, “Addressing Function Approximation Error in Actor-Critic Methods,” in Proceedings of the 35th International Conference on Machine Learning, PMLR, Jul. 2018, pp. 1587–1596. Accessed: Mar. 13, 2024.

[22] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor,” in Proceedings of the 35th International Conference on Machine Learning, PMLR, Jul. 2018, pp. 1861–1870. Accessed: Mar. 13, 2024.

[23] Towers, Mark and Terry, Jordan K., “Humanoid,” Gymnasium, https://gymnasium.farama.org/environments/mujoco/humanoid/, Mar.2023(accessed Mar. 2024)