Journal of Zhejiang University-SCIENCE A 2025 26(12):1163-1178
www.jzus.zju.edu.cn; www.springer.com/journal/11582
E-mail: jzus_a@zju.edu.cn

Research Article
https://doi.org/10.1631/jzus.A2500142

Check for updates

Stable and continuous vertical jumping control of hydraulic legged robots through reinforcement learning

Junhui ZHANG, Pengyuan JI, Lizhou FANG, Jinyuan LIU, Dandan WANG, Jikun AI, Huaizhi ZONG✉, Bing XU

State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310058, China

Abstract: Hydraulic legged robots have potential for high-dynamic motion due to their large power-to-weight ratios. However, it is challenging to ensure both stability and continuity in the motion of such robots. In this study, we propose a jumping motion control framework based on deep reinforcement learning that enables hydraulic limb leg units to perform stable and continuous jumping motions. First, to accurately represent the performance of a physical prototype, a quasi-realistic model incorporating physical feasibility constraints is constructed. This model is informed by analysis of the relevant fluid dynamics, and incorporates a trajectory generator and a motion tracking controller. To achieve stable and continuous jumping performance, a deep reinforcement learning algorithm is developed, which jointly optimizes the trajectory generator and the motion tracking controller. Through validation on the physical prototype, we demonstrate that the proposed method reduces the maximum deviation and the average deviation by over 47% and 60%, respectively, and improves landing compliance by up to 7.7% compared to a baseline optimization algorithm, the non-dominated sorting genetic algorithm (NSGA-II). The proposed control framework may serve as a reference for high-dynamic motion control of legged robots and multi-objective optimization across several decision variables.

Key words: Legged robot; Deep reinforcement learning; Quasi-realistic modelling; Hydraulic system; Jumping control

1 Introduction

Hydraulic legged robots have been a popular research focus in robotics due to their ability to execute high-dynamic and agile motions (Semini et al., 2017; Li et al., 2024; Xiang et al., 2025; Zhang JH et al., 2025). In field traversal tasks, legged robots are often required to perform stable and continuous jumping, where the limb leg unit (LLU) governs all leg movements throughout the jump cycle and directly determines the jumping performance (Zong et al., 2024). During such motions, the LLU frequently switches between swing and stance phases, which generates substantial ground contact forces (Zong et al., 2025). Meanwhile, the strong nonlinearities of hydraulic systems lead to increased complexities in their associated control methods.

Current research has primarily focused on improving jumping performance through foot-end trajectory generators and motion tracking controllers (Ba et al., 2025; Zhang K et al., 2025). The trajectory generator ensures smooth transitions in velocity and acceleration during switches in the motion phase (Gao et al., 2019; Bjelonic et al., 2020; Ahn and Cho, 2022). A well-designed trajectory contributes to smoother motion and reduces energy consumption, which improves overall performance. For example, Zhu et al. (2024) proposed a convergent iterative linear quadratic regulator (iLQR)-based trajectory optimization method that explicitly minimizes the worst-case perturbation growth along hybrid trajectories, which improved the stability of legged robots executing hybrid and underactuated motions. Also, Sun et al. (2021) developed a foot-end trajectory based on piecewise cubic spline interpolation during the swing phase of a quadruped robot, which reduced the energy consumption of the hydraulic actuators and enabled higher dynamic performance. Since the interaction between the robot's LLU and the ground during motion generates significant and uncertain external

✉ Huaizhi ZONG, hzzong@zju.edu.cn
Junhui ZHANG, https://orcid.org/0000-0002-2603-2065
Pengyuan JI, https://orcid.org/0009-0006-3435-5776
Lizhou FANG, https://orcid.org/0009-0009-4467-7236
Jinyuan LIU, https://orcid.org/0009-0003-1613-5823
Dandan WANG, https://orcid.org/0009-0007-6925-6273
Jikun AI, https://orcid.org/0000-0002-3416-1764
Bing XU, https://orcid.org/0000-0003-0236-7896

Received Apr. 23, 2025; Revision accepted July 29, 2025; Crosschecked Dec. 2, 2025

© Zhejiang University Press 2025

forces on the joints, the motion tracking controller is often employed to ensure precise tracking of the LLU's target position and force trajectories while ensuring smooth phase transitions; accordingly, stable and coordinated locomotion can be achieved (Boaventura et al., 2013; Han et al., 2023; Ba et al., 2024). Gu and Yuan (2020) proposed an adaptive and robust control strategy combining multiple Lyapunov functions to improve the trajectory tracking performance in the presence of uncertainties. Shao et al. (2023) defined the control parameters for a hydraulic quadruped robot and employed the self-growing Lévy-flight salp swarm algorithm to minimize controller signal tracking errors. For high-dynamic motion tasks, such as continuous jumping, the parameters of both the trajectory generator and the motion tracking controller jointly influence system performance. Therefore, designing them independently fails to capture their coupled dynamics, often resulting in suboptimal parameter combinations.

In recent years, deep reinforcement learning (DRL) has demonstrated significant potential in robotic control applications (Hoeller et al., 2024), providing solutions for multi-objective optimization problems involving coupled system dynamics. DRL-trained policies in stochastic environments can accomplish tasks stably and be directly transferred to physical robots. Common approaches to improve the success rate of deploying trained policies on physical prototypes include domain randomization (Exarchos et al., 2021) and dynamics randomization (Xie et al., 2021). However, most existing studies of DRL have focused on electric-actuated robots, while relevant applications on hydraulic robots have mainly involved excavators and hydraulic manipulators performing relatively low-speed motions, demonstrating the great potential of DRL in addressing complex nonlinear control problems (Egli and Hutter, 2022; Spinelli et al., 2024; Yao et al., 2024). To date, there has been no reported work applying DRL to hydraulic legged robots, which exhibit high-dynamic motion potential. The inherent strong nonlinearity of hydraulic systems requires precise dynamic modeling and controller parameter tuning, thereby increasing the difficulty of deploying DRL polices on physical prototypes. Moreover, recent studies on the application of DRL have primarily emphasized gait learning and adaptive locomotion over complex terrains (Hwangbo et al., 2019; Choi et al., 2023; Han et al., 2024). However, for high-dynamic motions such as continuous jumping, DRL implementation remains challenging due to stringent requirements for controller robustness and model accuracy, as well as the inherent sparsity of training rewards. To date, no satisfactory implementation has been reported.

To address the aforementioned challenges, a DRL-based jumping motion control framework for hydraulic LLUs is proposed. In order to bridge the gap between DRL simulation training and real-world deployment, a quasi-realistic model of the LLU incorporating hydraulic system dynamics, physical constraints, and dynamic randomization is established. The proposed DRL algorithm jointly optimizes the parameters of the trajectory generator and the motion tracking controller, ensuring expected explosive power while enhancing landing compliance. Finally, the trained policy is deployed on a physical prototype, and the motion performance and robustness are tested. The main contributions of this work are summarized as follows:

1. A quasi-realistic model incorporating physical feasibility constraints for the hydraulic system is established, reflecting the performance of the physical prototype and providing a reliable foundation for simulation training and deployment of DRL.

2. A DRL-based jumping motion control framework for legged robots is proposed, in which the DRL algorithm jointly optimizes the trajectory generator and the motion tracking controller to achieve stable continuous jumping motion in simulation training.

3. The DRL-trained policy is deployed on a physical prototype using the proposed control framework, and the target-height tracking and landing compliance in continuous jumping are compared to a commonly used optimization method.

The remainder of this paper is organized as follows. Section 2 describes the components of the quasi-realistic simulation model. Section 3 presents the DRL-based jumping motion control framework and details the algorithm for continuous target-height jumping tasks. The simulation training and deployment experiments, along with analysis of the results, are provided in Section 4. Finally, conclusions are given in Section 5.

2 Quasi-realistic LLU model

In this study, the control object is the LLU of the Spurlos hydraulic quadruped robot, which was designed

and fabricated in our earlier studies (Zhang K et al., 2025). To facilitate successful simulation training and deployment on the physical prototype, a quasi-realistic model is developed. This section introduces the structure of the proposed model, which is shown in Fig. 1. The trajectory generator provides the desired foot-end trajectory to the motion tracking controller, which computes the desired forces for the two joint hydraulic actuators. The forces are then input to the direct adaptive robust controller (DARC), which outputs the corresponding servo valve current signals to achieve the expected motion control of the LLU.

2.1 Kinematic and dynamic model

The LLU of the Spurlos hydraulic quadruped robot includes a hip flexor extensor joint (HFE) and a knee flexor extensor joint (KFE), both of which are driven by linear hydraulic cylinders arranged in the same plane. The hydraulic cylinder has a maximum stroke of 60 mm, with the diameters of the rodless and rod-side chambers being 25 and 12 mm, respectively. The structure of the LLU is shown in Fig. 2, and the structural parameters are listed in Tables 1 and 2.

The kinematics of the leg are formulated as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} L_1 \sin q_1 + L_2 \sin(q_1+q_2) \\ -L_1 \cos q_1 - L_2 \cos(q_1+q_2) \end{bmatrix}, \tag{1}$$

where x and y are the horizontal and vertical positions of the foot end, respectively, q_1 is the HFE angle, q_2 is the KFE angle, and L_1 and L_2 are the thigh and calf lengths, respectively.

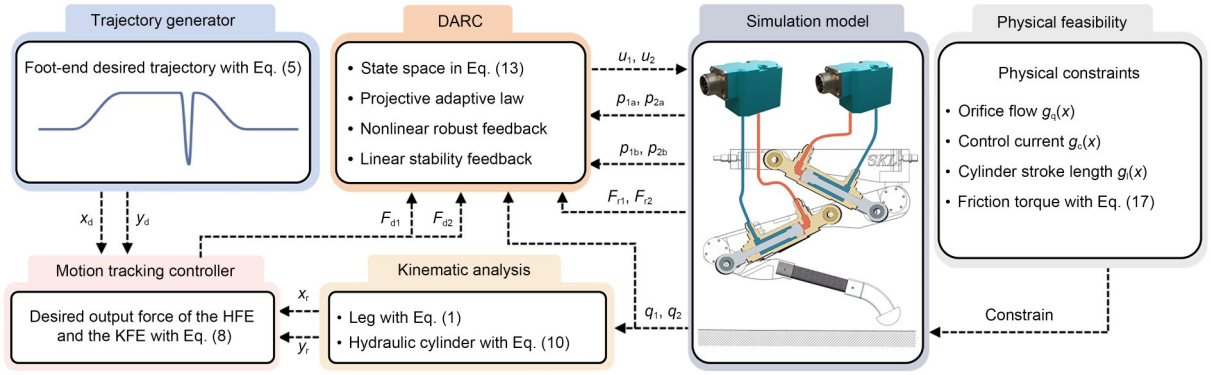

Fig. 1 Quasi-realistic model of the LLU. u_1 and u_2 are the control signals of the servo valves for the two joints, respectively; p_{1a} and p_{2a} are the pressures in the two chambers of the hip joint's hydraulic cylinder, respectively; p_{1b} and p_{2b} are the pressures in the two chambers of the knee joint's hydraulic cylinder, respectively; F_{r1} and F_{r2} represent the actual output forces of the two joints, respectively; F_{d1} and F_{d2} represent the desired output forces of the two joints, respectively; x_d and y_d represent the desired displacements of the foot-end; x_r and y_r represent the actual displacements of the foot-end. The other parameters are explained in the text

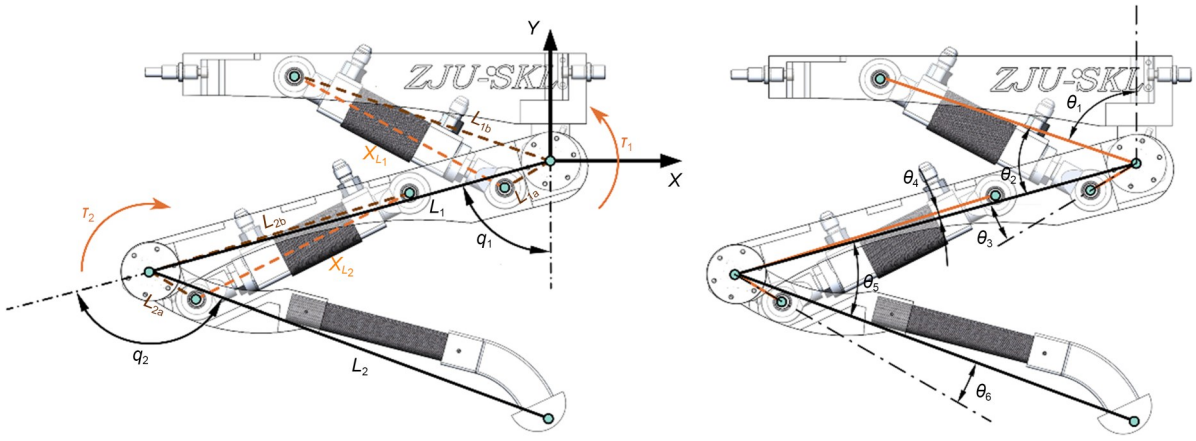

Fig. 2 Structure of the LLU

Table 1 Structural parameters of the LLU shown in Fig. 2

Parameter	Value	Parameter	Value
L_1 (mm)	350	L_{1b} (mm)	228
L_2 (mm)	350	L_{2b} (mm)	228
L_{1a} (mm)	45	X_{L_1} (mm)	201–261
L_{2a} (mm)	45	X_{L_2} (mm)	201–261

Table 2 Joint angles of the LLU shown in Fig. 2

Parameter	Value	Parameter	Value
θ_1 (°)	71.5	θ_4 (°)	1.5
θ_2 (°)	30.0–115.0	θ_5 (°)	33.5–118.5
θ_3 (°)	17.0	θ_6 (°)	15.0

Then, the inverse kinematics can be formulated as:

$$\boldsymbol{q}=\begin{bmatrix}q_1\\q_2\end{bmatrix}=\begin{bmatrix}-\left[\arctan\left(\dfrac{x}{y}\right)-\arccos\left(\dfrac{L_1^2+x^2+y^2-L_2^2}{2L_1\sqrt{x^2+y^2}}\right)\right]\\ \pi-\arccos\left(\dfrac{L_1^2+L_2^2-x^2-y^2}{2L_1L_2}\right)\end{bmatrix}. \tag{2}$$

To resolve the foot-end forces and velocities of the LLU into joint angular velocities and output torques, we construct the Jacobian matrix $\boldsymbol{J}$:

$$\boldsymbol{J}(\boldsymbol{q})=\begin{bmatrix}L_1\cos(q_1)+L_2\cos(q_1+q_2) & L_2\cos(q_1+q_2)\\ L_1\sin(q_1)+L_2\sin(q_1+q_2) & L_2\sin(q_1+q_2)\end{bmatrix}. \tag{3}$$

The joint torques of HFE and KFE, i.e., τ_1 and τ_2, respectively, can be obtained from the Lagrange equation. The dynamics of the LLU can be generally formulated as:

$$\boldsymbol{\tau}=\begin{bmatrix}\tau_1\\\tau_2\end{bmatrix}=\boldsymbol{M}(\boldsymbol{q})\ddot{\boldsymbol{q}}+\boldsymbol{C}(\boldsymbol{q},\dot{\boldsymbol{q}})\dot{\boldsymbol{q}}+\boldsymbol{G}(\boldsymbol{q}), \tag{4}$$

where $\boldsymbol{\tau}\in\mathbb{R}^2$ is the vector of the external generalized torques, $\boldsymbol{M}\in\mathbb{R}^{2\times2}$ is the mass inertial term, $\boldsymbol{C}\in\mathbb{R}^2$ is the Coriolis force and centrifugal force vector, and $\boldsymbol{G}\in\mathbb{R}^2$ is the gravitational torque vector.

2.2 Trajectory generator

The jumping process of the LLU consists of four phases: the standing stage, squatting stage, jumping stage, and buffering stage. The foot-end motion of the LLU in the fuselage coordinate system is shown in Fig. 3.

In order to make the position of the foot-end during the jumping motion satisfy continuity of displacement, velocity, and acceleration, the trajectory of the foot-end is designed by the quintic polynomial (Nansai et al., 2015):

$$\begin{cases}y(t)=c_0+c_1t+c_2t^2+c_3t^3+c_4t^4+c_5t^5,\\ \begin{bmatrix}c_0\\c_1\\c_2\\c_3\\c_4\\c_5\end{bmatrix}=\begin{bmatrix}1&t_s&t_s^2&t_s^3&t_s^4&t_s^5\\1&t_e&t_e^2&t_e^3&t_e^4&t_e^5\\0&1&2t_s&3t_s^2&4t_s^3&5t_s^4\\0&1&2t_e&3t_e^2&4t_e^3&5t_e^4\\0&0&2&6t_s&12t_s^2&20t_s^3\\0&0&2&6t_e&12t_e^2&20t_e^3\end{bmatrix}^{-1}\begin{bmatrix}y_s\\y_e\\v_s\\v_e\\a_s\\a_e\end{bmatrix},\end{cases} \tag{5}$$

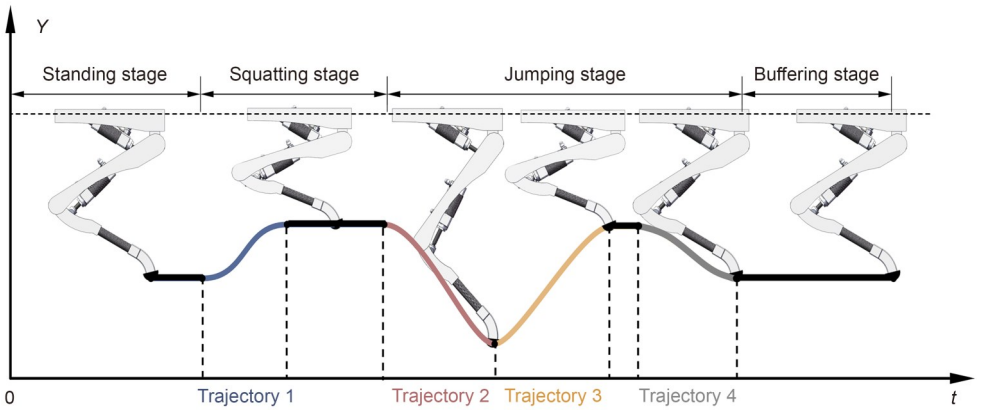

Fig. 3 Stages and trajectories in the jumping process

where t_s is the initial time, t_e is the end time, y_s is the initial displacement, y_e is the end displacement, v_s is the initial velocity, v_e is the end velocity, a_s is the initial acceleration, a_e is the end acceleration, and $[c_0, c_1, c_2, c_3, c_4, c_5]^T$ indicates the quintic polynomial coefficients.

2.3 Motion tracking controller

Since the quintic polynomial trajectory needs to be converted into the control signals of each servo valve to control the motion of the LLU, the foot contact force of the LLU is considered as a 2D spring-damping system. Based on the foot-end position error and velocity error, the virtual forces of the foot-end can be calculated as follows:

$$F_C = \boldsymbol{K}_P(\boldsymbol{p}_d - \boldsymbol{p}_r) + \boldsymbol{K}_D(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}_r), \tag{6}$$

where F_C denotes the virtual forces of the foot-end, $\boldsymbol{p}_r=[x_r, y_r]^T$ is the actual position calculated by the positive kinematics, $\boldsymbol{p}_d=[x_d, y_d]^T$ is the desired position, $\dot{\boldsymbol{p}}_r$ and $\dot{\boldsymbol{p}}_d$ are the actual velocity of motion and the desired velocity of motion at the endpoint of the foot, respectively, and $\boldsymbol{K}_P\in\mathbb{R}^{1\times 2}$ and $\boldsymbol{K}_D\in\mathbb{R}^{1\times 2}$ are the desired stiffness and damping values in the forms of diagonal positive definite matrixes, respectively.

The effects of $\boldsymbol{M}(\boldsymbol{q})$ and $\boldsymbol{C}(\boldsymbol{q}, \dot{\boldsymbol{q}})$ are often ignored because of the influence of noise on the angular velocities and accelerations. Thus, the dynamics of the LLU in Eq. (4) can be reformulated as:

$$\tau = \boldsymbol{G}(\boldsymbol{q}) + \boldsymbol{J}(\boldsymbol{q})^T(\boldsymbol{K}_p(\boldsymbol{p}_d - \boldsymbol{p}_r) + \boldsymbol{K}_d(\dot{\boldsymbol{p}}_d - \dot{\boldsymbol{p}}_r)). \tag{7}$$

The output torque of the joint needs to be converted into the output force of the hydraulic cylinder in the LLU. Based on the structural parameters, the interrelationships are:

$$\begin{cases} F_1 = \tau_1 \dfrac{\sqrt{L_{1b}^2 + L_{1a}^2 - 2L_{1a}L_{1b}\sin\phi_1}}{L_{1a}L_{1b}\sin\phi_1}, \\ F_2 = \tau_2 \dfrac{\sqrt{L_{2b}^2 + L_{2a}^2 - 2L_{2a}L_{2b}\sin\phi_2}}{L_{2a}L_{2b}\sin\phi_2}, \end{cases} \tag{8}$$

where F_1 and F_2 are the output forces of the HFE and the KFE, respectively, ϕ_1 is the angle between L_{1a} and L_{1b}, and ϕ_2 is the angle between L_{2a} and L_{2b}.

2.4 Valve-controlled cylinder system

The output force of any linear hydraulic cylinder in a hydraulic system can be described as:

$$F = P_1A_1 - P_2A_2 - b\dot{x}_L - F_{fc} + d, \tag{9}$$

where P_1 and P_2 are the pressures in the two chambers of the hydraulic cylinder, respectively, A_1 and A_2 are the actual actuating areas of the two chambers, respectively, x_L and $\dot{x}_L$ are the output displacement and velocity of the hydraulic cylinder, respectively, b is the friction combination factor, F_{fc} is the Coulomb friction, and d is the lumped modelling error.

The dynamics of the hydraulic cylinder are formulated as:

$$\begin{cases} \dfrac{V_1}{\beta_e}\dot{P}_1 = -A_1\dot{x}_L + Q_1, \\ \dfrac{V_2}{\beta_e}\dot{P}_2 = A_2\dot{x}_L - Q_2, \end{cases} \tag{10}$$

where β_e is the bulk modulus of the hydraulic fluid, V_1 and V_2 are the actual volumes of the head-end and rod-end chambers of the hydraulic cylinders, respectively, and Q_1 and Q_2 are the inflow and outflow of the oil, respectively (Jaiswal et al., 2021):

$$\begin{cases} Q_1 = k_{q_1}u\sqrt{\Delta P_1}, \quad \Delta P_1 = \begin{cases} P_s - P_1, & x_v > 0, \\ P_1 - P_r, & x_v < 0, \end{cases} \\ Q_2 = k_{q_2}u\sqrt{\Delta P_2}, \quad \Delta P_2 = \begin{cases} P_s - P_2, & x_v > 0, \\ P_2 - P_r, & x_v < 0, \end{cases} \end{cases} \tag{11}$$

where $k_{qi}=C_d w k_i\sqrt{2/\rho}$ (i=1, 2) represents the flow gain coefficient of the servo valve, C_d is the discharge coefficient, w is the gradient of the slide valve area, k_i is the positive gain factor, ρ is the fluid density, u is the control signal, x_v represents the displacement of the servo valve spool, P_s is the supply pressure of the hydraulic fluid, and P_r is the drained pressure of the hydraulic fluid.

Therefore, the kinematics of the joint force control can be obtained as follows:

$$\dot{F} = \left(\frac{A_1}{V_1}\sqrt{\Delta P_1} + \frac{A_2}{V_2}\sqrt{\Delta P_2}\right)\beta_e k_{qi}u - \left(\frac{A_1^2}{V_1} + \frac{A_2^2}{V_2}\right)\beta_e\dot{x}_L - b\ddot{x}_L - A_f\dot{S}_f(\dot{x}_L) + \dot{d}, \tag{12}$$

where $A_f S_f$ is the approximated nonlinear Coulomb friction, in which the amplitude A_f is unknown but the continuous shape function S_f is known.

2.5 Direct adaptive robust controller

For hydraulic legged robots, the upper control algorithm transfers the required output force from each motion joint of the single leg to the underlying valve-controlled hydraulic cylinder control system, acting as a tracking target for the output force of the joints. However, there are many time-varying parameters in the output force control model that cannot be obtained directly. To eliminate the discrepancy between the output force and the desired force, and to enable the controller to estimate relevant dynamic parameters based on system feedback errors and other measurements, the DARC method is adopted for joint force control (Yao, 2009). By setting $x=F$ and $\boldsymbol{\theta}=[\theta_1,\theta_2,\theta_3,\theta_4,\theta_5]^{\mathrm{T}}=[\beta_e k_{qi},\beta_e,b,A_f,d_n]^{\mathrm{T}}$ (d_n represents the nominal value of the lumped uncertain nonlinearity d), the state space can be formulated as:

$$\dot{x}=\theta_1 a_1 u-\theta_2 a_2-\theta_3\ddot{x}_L-\theta_4\dot{S}_f(\dot{x}_L)+\theta_5+\Delta d, \quad (13)$$

where $\Delta d=\dot{d}-d_n$ is the value of the nonlinearity, and the nonlinear equations a_1 and a_2 are defined as:

$$\begin{cases} a_1(P_1,P_2,x_L)=\dfrac{A_1}{V_1}\sqrt{\Delta P_1}+\dfrac{A_2}{V_2}\sqrt{\Delta P_2}, \\ a_2(x_L,\dot{x}_L)=\left(\dfrac{A_1^2}{V_1}+\dfrac{A_2^2}{V_2}\right)\dot{x}_L. \end{cases} \quad (14)$$

The resulting controller law of the DARC is designed as follows:

$$\begin{cases} u_a=\dfrac{1}{\hat{\theta}_1 a_1}\left[\hat{\theta}_2 a_2+\hat{\theta}_3\ddot{x}_L+\hat{\theta}_4 S_f(x_L)-\hat{\theta}_5\dot{F}_d\right], \\ u_s=\dfrac{-kz-k_s z}{\hat{\theta}_1 a_1}, \\ u=u_a+u_s, \end{cases} \quad (15)$$

where $\hat{\theta}_i$ is the estimation of θ_i, F_d is the desired force, u is the nonlinear adaptive robust control law, u_a is the adaptive model compensation term, u_s is the robust control term, z is the tracking error of the joint output force, k is the positive feedback gain, and k_s is the positive nonlinear gain (Fang et al., 2025).

The DARC controller parameters were kept constant across all experimental trials, as listed below. The feedback gain $K=k+k_s=2\times10^{-6}$, and the bounds of uncertainty ranges and the initial estimates of the parameter variations were chosen as:

$$\begin{cases} \boldsymbol{\theta}_{\min}=[6,\ 1.0\times10^8,\ 60,\ 60,\ -1.0\times10^{-3}]^{\mathrm{T}}, \\ \boldsymbol{\theta}_{\max}=[10,\ 3.0\times10^8,\ 90,\ 100,\ 1.0\times10^{-3}]^{\mathrm{T}}, \\ \boldsymbol{\theta}(0)=[8,\ 2.0\times10^8,\ 60,\ 60,\ 0]^{\mathrm{T}}. \end{cases} \quad (16)$$

2.6 Physical constraints

To construct a simulation model that accurately represents the physical prototype, three key constraints of the LLU were considered: the orifice flow $g_q(x)$, control current $g_c(x)$, and cylinder stroke length $g_l(x)$.

$$\begin{cases} g_{q\min}\leqslant g_q(x)\leqslant g_{q\max}, \\ g_{c\min}\leqslant g_c(x)\leqslant g_{c\max}, \\ g_{l\min}\leqslant g_l(x)\leqslant g_{l\max}, \end{cases} \quad (17)$$

where $(\cdot)_{\min}$ and $(\cdot)_{\max}$ are the minimum and maximum values of the constraint. Frictional torque is considered to make the model more accurately reflect real movement:

$$\begin{cases} \tau^i=\tau_c^i-K_\tau\omega^i, \quad i=1,2, \\ F_p=K_G v_L, \end{cases} \quad (18)$$

where τ^i is the actual joint friction torque, τ_c^i is the torque supplied by the hydraulic cylinder, K_τ is the joint friction coefficient, ω^i is the angular velocity, F_p is the friction with the platform, K_G is the guideway friction coefficient, v_L is the speed of the LLU, and $i=1,2$ denotes the two joints.

3 Motion learning control framework

In this section, a DRL algorithm is introduced to jointly optimize the primary parameters of the trajectory generator and motion tracking controller that were described in Section 2. The control policy is trained in a simulation environment to achieve stable, continuous jumps at the target height and enhance landing compliance. The proposed jumping motion learning control framework is shown in Fig. 4. In the DRL environment, the policy outputs actions based on the current

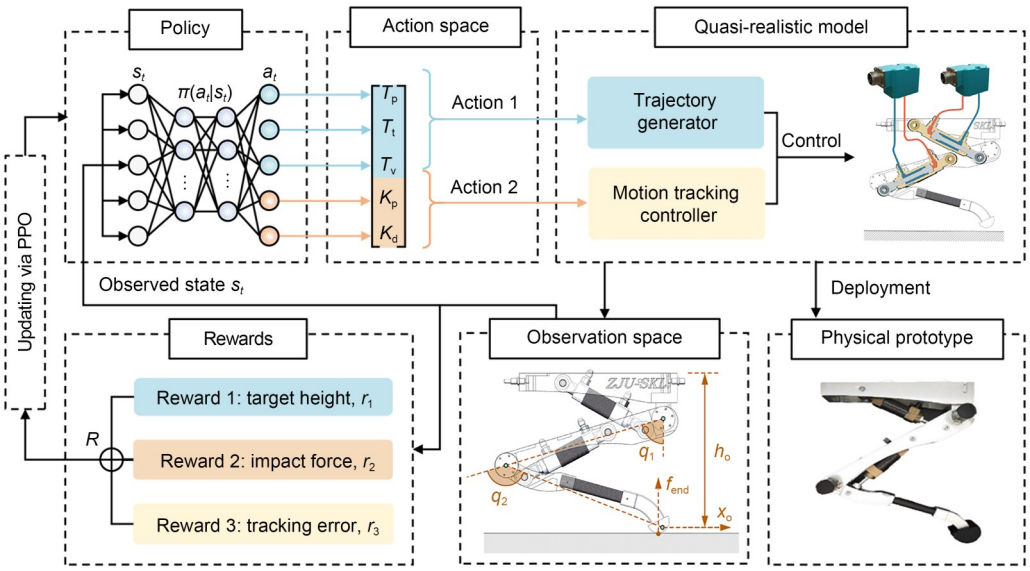

Fig. 4 Jumping motion learning control framework. PPO is the proximal policy optimization; π is the current policy of taking action a in state s, and the other parameters are explained in the text

observation space. The action space is designed to consist of key parameters of the trajectory generator and motion tracking controller, which directly determine the performance of the next jump of the LLU. The policy is updated by maximizing the cumulative reward, and can be further deployed on the physical prototype using the proposed control framework.

3.1 Deep reinforcement learning

Reinforcement learning (RL) is used to formulate the control problem as a Markov decision process, which is typically represented by the tuple $(\mathcal{S}, \mathcal{A}, p, r, \lambda)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ denotes the set of actions that the agent can execute, p denotes the state transition function that defines the probability of transitioning to the next state s_{t+1} when the agent takes action a_t in state s_t according to the policy, r denotes the reward function, which is usually set artificially based on the designed task objectives, and $\lambda \in (0,1)$ is the discount factor that determines the weight assigned to future rewards. The larger the value of the discount factor, the more the agent pays attention to the long-term cumulative reward. The agent aims to maximize the expectation of cumulative rewards obtained over multiple rounds of interactions. The expected reward of the policy can be expressed as follows:

$$Q(s,a)=\mathbb{E}_t\left\{\sum_{j=0}^{\infty}\lambda^j r_{t+j+1}\,|\,s_t=s,a_t=a\right\}, \tag{19}$$

where $Q(s, a)$ is the action-value function of taking action a in state s, and $\mathbb{E}_t$ denotes the expectation operator over timesteps.

DRL integrates deep learning with RL, leveraging the ability of deep neural networks to approximate functions in order to extend RL to high-dimensional problems. In recent years, DRL has achieved significant progress in the field of robotic control. The proximal policy optimization (PPO) strikes an excellent balance among training stability, ease of implementation, and convergence efficiency, making it a popular algorithm for applications of legged robotics (Schulman et al., 2017). Its distinctive clipped surrogate objective constrains the policy update within a trusted region, which enhances the efficiency and stability of training. Additionally, it balances sample efficiency and performance through training involving multiple epochs of minibatch updates on each collected batch of data. These advantages make PPO an ideal choice for addressing the complex, coupled parameter control problems encountered in this study. The advantage function in the PPO algorithm is calculated as:

$$A(s,a)=Q(s,a)-V(s), \tag{20}$$

where $A(s, a)$ is an estimator of the advantage function in state s_t, and $V(s)$ is the value function of state s, which represents the expected return of taking any action in state s_t.

The clipped surrogate objective is given by:

$$\begin{cases} \mathcal{L}^{\text{clip}}(\delta) = \\ \quad \mathbb{E}_t\left[\min\left(r(\delta),\ \text{clip}\left(r(\delta),\ 1-\varepsilon,\ 1+\varepsilon\right)\right) A(s,\ a)\right], \\ r(\delta) = \dfrac{\pi(a|s)}{\pi_{\text{old}}(a|s)}, \end{cases} \tag{21}$$

where δ is the parameter of the policy network, $r(\delta)$ is the ratio between the new policy $\pi(a|s)$ and the old policy $\pi_{\text{old}}(a|s)$, and ε is a hyperparameter that limits the update magnitude.

During the manual tuning of the jump control model parameters described in Section 2, high-dynamic tasks such as continuous target-height jumping present significant limitations. This parameter tuning process is labor-intensive and inefficient, and the controller cannot adjust parameters in response to dynamic changes in the environment, thereby compromising motion robustness. Directly controlling the jumping motion through DRL presents a significant challenge compared to locomotion tasks, as the rewards for jumping are highly sparse. This sparsity results in lower training efficiency and reduced controller robustness. Instead, integrating DRL with a preset trajectory-based controller eliminates the tedious process of manually tuning the controller parameters, and also enhances the efficiency of training for the desired jumping task. Moreover, this approach improves the robustness of the controller. The discrepancies between the simulation and reality, caused by modeling errors and the limitations of the simulation physics engine, can be effectively minimized by incorporating dynamic parameter randomization, domain randomization, and correlated noise.

3.2 Action space

To achieve stable and continuous jumps at target heights while improving landing compliance, and to enhance training efficiency by reducing ineffective exploration, multiple key parameters from both the trajectory generator and the motion tracking controller are selected as the action space.

In the trajectory generator, the trajectory parameters of the swing phase at takeoff directly influence the jump height. Specifically, the trajectory starting position T_{p}, the time to reach the designation point T_{t}, and the slope of the designation point T_{v} determine the coefficients of the quintic polynomial for this phase. In the motion tracking controller, the stiffness coefficient K_{p} and damping coefficient K_{d} of the virtual spring-damping system are the primary parameters of the desired force generator. K_{p} determines the system's response speed and K_{d} helps to suppress oscillations. In summary, the action space is $A=(T_{\text{p}},\ T_{\text{t}},\ T_{\text{v}},\ K_{\text{p}},\ K_{\text{d}})$. Furthermore, to reduce unnecessary and ineffective exploration by the agent, we manually set the range of variation for each action based on prior engineering experience, for which the values are detailed in Table 3.

Table 3 Range of the action variations

Action	Value range
T_{p}	[−0.05, 0.05]
T_{t}	[0.1, 0.2]
T_{v}	[0.5, 2.5]
K_{p}	[1500, 5000]
K_{d}	[50, 170]

3.3 Observation space

For continuous jumping tasks with high dynamicity, the key performance indicators are observed from multiple perspectives to further accelerate the convergence of the reward function. The observation space is $S=(x_{\text{o}},\ h_{\text{o}},\ f_{\text{end}},\ q_1,\ q_2,\ q_{1\text{d}},\ q_{2\text{d}})$, where x_{o} denotes the horizontal position of the foot relative to the body frame, h_{o} is the vertical height of the body relative to the ground, f_{end} denotes the contact force between the foot-end and the ground (which can be measured and relayed in real-time using force sensors), q_1 and q_2 denote the rotation angles of the LLU knee and hip joints, respectively, and $q_{1\text{d}}$ and $q_{2\text{d}}$ are the desired rotational angles for the respective joints.

3.4 Reward function

The reward function is designed to achieve stable and continuous jumping at the target height. Specifically, it encourages each jump to reach the target height while penalizing behaviors that compromise motion stability, such as excessive landing impact forces and large motion tracking errors. The reward function is comprised of the following components:

(1) Target height r_1: The agent is encouraged to reach the predefined target height in each jump, while undershooting and overshooting the target are both penalized. The corresponding term is calculated as:

$$r_1 = |h_t - h_m|, \tag{22}$$

where h_t denotes the predefined target height of the LLU's body, and h_m is the maximum height attained by the body during each jump, which is reset after the completion of each jump cycle.

(2) Impact force r_2: To reduce the large impact force generated during LLU landing, which can adversely affect motion stability, a landing compliance index ω is introduced to encourage the agent to improve landing compliance. A lower value of ω indicates better compliance upon landing. These quantities are defined as:

$$r_2 = \frac{1}{\omega}, \quad \omega = \frac{F_e}{F_b}, \tag{23}$$

where F_b and F_e denote the maximum foot contact forces during takeoff and landing, respectively.

(3) Tracking error r_3: The agent is encouraged to accurately track the desired trajectories to achieve the expected motion performance. A penalty is applied to the motion tracking errors of the two joint angles, which is formulated as follows:

$$r_3 = (\varepsilon_1 + \varepsilon_2)^2, \tag{24}$$

where $\varepsilon_1 = |q_{1d} - q_1|$, $\varepsilon_2 = |q_{2d} - q_2|$.

The final expected reward R can then be determined by:

$$R = \omega_1 r_1 + \omega_2 r_2 + \omega_3 r_3. \tag{25}$$

Note that the weights used during training are $\omega_1 = -60$, $\omega_2 = 10$, and $\omega_3 = -100$.

3.5 Simulation to reality transfer

Bridging the gap between simulation and prototype deployment is a fundamental challenge in DRL, and the nonlinear characteristics of hydraulic actuation systems impose greater demands on policy robustness. To further improve the success rate of sim-to-real transfer, a series of dynamic randomizations is incorporated into the control framework of the simulation environment.

First, due to the delay between the desired force issuance and the execution of the hydraulic actuator, a random delay of 1–5 ms is introduced into the desired force output within the simulation training environment. The friction between the body and the fixed guide rail has a non-negligible impact on actual jumping performance. To compensate for the negative effects of this friction, a disturbance force opposite to the direction of motion is applied in the simulation environment. Also, to account for the inherent variability in foot-ground friction, the ground friction coefficient is randomized at the beginning of each training episode within a predefined range to enhance the robustness of the policy. Furthermore, to compensate for inaccuracies in real-world sensors, Gaussian noise with a standard deviation of 0.05 is introduced into the observation space.

3.6 Training in simulation

Our simulation is conducted within the Matlab/Simulink® simulation environment. The physics engine in the environment is configured with an iteration step size of 1 ms. The actor-critic networks in the algorithm both feature two fully connected hidden layers, with ReLU (rectified linear unit) used as the activation function. The first and second layers contain 256 and 128 neurons, respectively. The other hyperparameters of the algorithm are detailed in Table 4.

Table 4 Hyperparameters of the algorithm

Hyperparameter	Value
Learning rate	10^{-4}
Horizon	512
Minibatch size	128
Number of epochs	4
Discount	0.99
GAE factor	0.95
Clip factor	0.2

Three groups of continuous jumping tasks were designed, with the target body heights set to 0.64, 0.68, and 0.72 m, respectively. Each group begins from an initial body height of 0.44 m, and requires the robot to perform three consecutive jumps to reach the corresponding target height. The objective is to consistently reach the target height in each jump, while enhancing the landing compliance. The reward evolution during training is illustrated in Fig. 5. Each group was trained five times with different random seeds, and the shaded regions in this figure indicate 95% confidence intervals. During the exploration process in the three training tasks, the agent's mean reward gradually increased

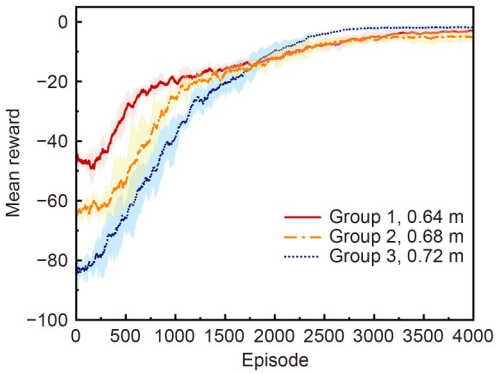

Fig. 5 Learning curve for the continuous jumping tasks

and eventually converged, indicating successful and efficient training.

4 Simulation results and discussion

In this section, we present the deployment of the DRL-trained policy from the simulation environment to the physical prototype, and compare its motion performance with a baseline optimized using the fast and elitist non-dominated sorting genetic algorithm (NSGA-II). The goal is to evaluate the improvement in performance from the proposed method.

4.1 Experimental platform

The trained policy is deployed on the experimental platform depicted in Fig. 6, which includes a hydraulic pump station, an LLU platform, internal sensors, a control board, and external sensors.

In this experiment, the hydraulic pump station operates at a working pressure of 15 MPa and utilizes YH-10 hydraulic oil. The LLU platform consists of an integrated frame, vertical guide rails, a limb, and a leg unit base. The linear hydraulic cylinders are actuated by Domin S4 Pro servo valves, which feature a frequency response exceeding 300 Hz (±25%), so as to meet the high-frequency response requirements of the robotic system. Each driven joint of the LLU is equipped with an absolute angular encoder with a resolution of 32768 P/R (pulses per revolution). The selected hydraulic sensors are custom-made and have an accuracy of 0.5% FS (full scale). The external sensors include a draw-wire sensor for measuring jumping height, and a 3-axis force sensor for quantifying contact forces.

The host computer in this experiment is a laptop equipped with an Intel i7-14650HX CPU and a GeForce RTX 4060 GPU. The experimental procedure involves running the trained policy on the host computer, which sends the control commands to the servo valves via the control board in order to actuate the LLU. The real-time sensor feedback from the legged unit is transmitted back to the host computer, which then computes and outputs the next action in a continuous loop until the task is completed.

4.2 Sim-to-real validation

To validate the effectiveness of the proposed method in sim-to-real transfer, the trained policies for the three task groups are deployed onto the physical prototype and are compared with the results from the simulation environment, as shown in Fig. 7 and Table 5.

Here, h_s denotes the average jumping height over three trials in the simulation environment, while h_r represents the corresponding average height measured

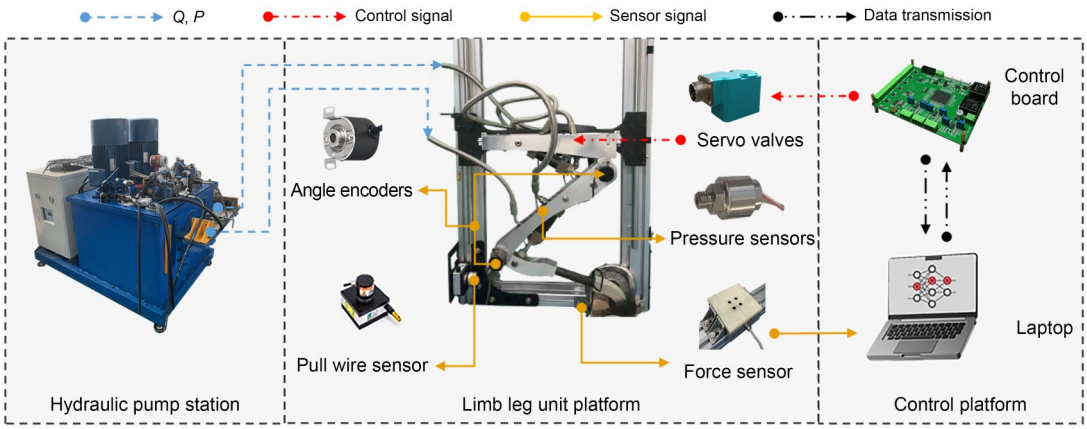

Fig. 6 LLU jumping test platform

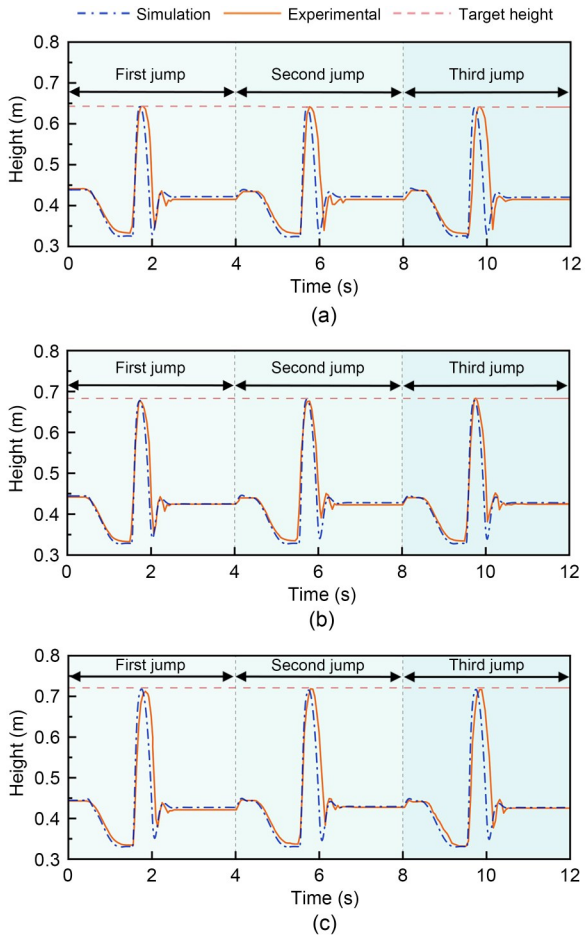

Fig. 7 Simulation and experimental results for continuous jumping at different target heights: (a) Group 1; (b) Group 2; (c) Group 3

Table 5 Height errors between the simulation and experiment

Group	Target height (m)	h_s (m)	h_r (m)	ε_a (%)	ε_m (%)
Group 1	0.64	0.64090	0.64222	0.21	0.37
Group 2	0.68	0.67918	0.67720	0.40	0.90
Group 3	0.72	0.71794	0.71587	0.29	0.83

in the real-world experiments. ε_a is defined as the mean absolute deviation between the simulation and physical prototype jump heights across the three groups. The maximum error ε_m refers to the largest deviation rate in a single jump.

The deviations observed between the actual and target jump heights in the experiments can be attributed to multiple factors, such as the increased response lag caused by the hydraulic transmission lines, the instability of the hydraulic oil's properties due to pressure increases and temperature variations, and the influence of noise from the test platform and the environment. The reduction of deviations reflects an improvement in both the accuracy and robustness of the controller.

The results indicate that the jump height errors remain under 1% for both the simulation and experimental scenarios. This demonstrates the effectiveness of the control framework in bridging the sim-to-real gap and confirms the accuracy of the proposed simulation model.

4.3 Experimental results

As a predominant algorithm in the field of multi-objective optimization, the NSGA-II (Ma et al., 2023; Duan et al., 2024) has the ability to handle the nonlinear and strongly coupled optimization problems addressed in this study. Therefore, we select the NSGA-II as the baseline method to compare with in subsequent experiments. To intuitively demonstrate the effectiveness of DRL in managing continuous jumping of legged robots, we compare the motion performance of the DRL-trained policy with the NSGA-II group, which employs the NSGA-II to optimize the same set of parameters; the desired jump height and improved landing compliance are targeted. To evaluate the performance, both algorithms are deployed on the physical prototype to perform jumping experiments across three target-height tasks. The resulting jump heights and foot contact force profiles are shown in Fig. 8.

As illustrated in Fig. 8, each jump executed using the proposed DRL algorithm closely reaches the target height, and the amplitude and frequency of the LLU's oscillations upon landing are significantly reduced compared to the NSGA-II group. This improvement is attributed to the optimized parameter combinations that were determined through extensive training with DRL, which effectively enhance the stability of continuous jumping. Moreover, by adaptively adjusting the impedance controller parameters according to the LLU's dynamic state, the DRL algorithm improves the landing compliance. A quantitative comparison of the results is presented in Table 6 and Fig. 9.

Specifically, h_r denotes the average height of the three consecutive jumps for each target height, Δ is the absolute deviation between the average jumping height and the target height, σ_a represents the mean deviation rate between each jump height and the corresponding target height, and σ_m is the largest deviation rate

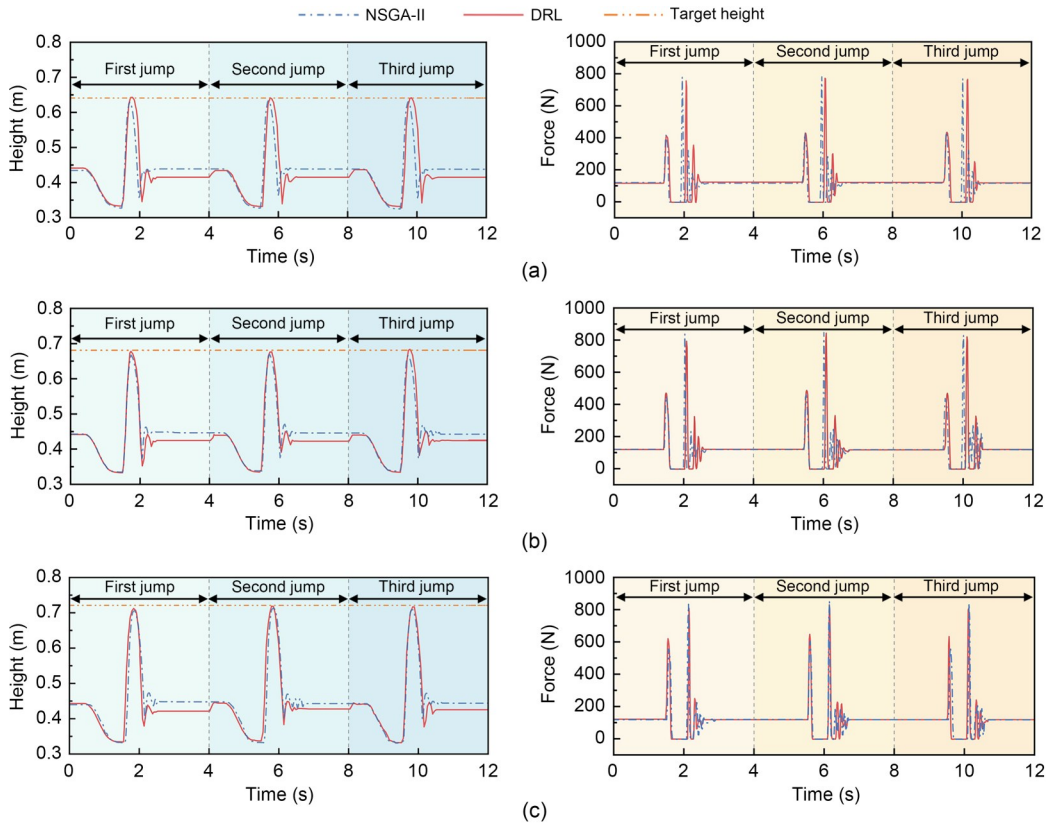

Fig. 8 Comparison of continuous jumping results regarding height and contact force: (a) Group 1; (b) Group 2; (c) Group 3

Table 6 Comparison of performance between different algorithms

Group	Algorithm	h_r (m)	Δ (m)	σ_a (%)	σ_m (%)	ω
Group 1 (0.64 m)	NSGA-II	0.63126	0.00874	1.37	1.62	1.39456
	DRL	0.64222	0.00222	0.35	0.48	1.28698
Group 2 (0.68 m)	NSGA-II	0.66699	0.01301	1.91	2.36	1.86820
	DRL	0.67720	0.00280	0.76	1.21	1.72931
Group 3 (0.72 m)	NSGA-II	0.70910	0.01090	1.51	2.00	1.82173
	DRL	0.71587	0.00413	0.57	1.06	1.79915

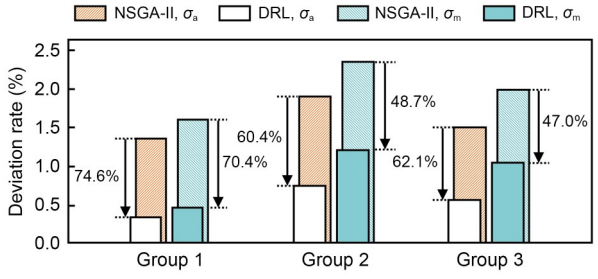

Fig. 9 Comparison of target-height deviation rate between different methods

observed in a single jump. Additionally, ω is the landing compliance index and a lower value of ω indicates improved landing compliance.

As shown in Table 5 and Fig. 9, in comparison to NSGA-II, the proposed DRL algorithm achieves significant improvements in both average error σ_a and maximum single-jump error σ_m. Specifically, for each group, σ_m is reduced by 70.4%, 48.7%, and 47.0%, respectively. σ_a is reduced by over 60% in all groups, with a reduction of up to 74.6% for the group with a target height of 0.64 m. Furthermore, the landing compliance index ω achieved by the DRL algorithm is consistently lower, indicating greater mitigation of impact upon landing, with the maximum improvement exceeding 7.7%. These results demonstrate that the proposed DRL algorithm effectively improves both the stability of motion and the landing compliance of the LLU.

4.4 Discussion of transferability

To investigate the generalizability of the proposed control framework and its bolstering of performance in other high-dynamic motion tasks, we selected forward horizontal jumping as a representative task for additional evaluation and discussion. The entire policy transfer process required only basic, task-specific adjustments, which are discussed below.

Compared to vertical jumping, horizontal jumping introduces an additional degree of freedom along the x-axis. Therefore, an additional trajectory parameter representing the x-axis displacement T_f over a specified time interval is introduced to the action space, while the other optimization parameters are kept the same. Accordingly, the term that previously encouraged reaching the vertical target height was modified to instead encourage horizontal jumping distance in the reward function, as detailed below:

$$\begin{cases} r_{1h}=\omega_1|x_d|, \\ r_{2h}=\omega_2\dfrac{1}{\omega}, \\ r_{3h}=\omega_3(\varepsilon_1+\varepsilon_2)^2, \\ R_h=r_{1h}+r_{2h}+r_{3h}, \end{cases} \tag{26}$$

where r_{1h} denotes the reward for forward distance, r_{2h} denotes the reward for landing compliance, and r_{3h} represents the penalty for the trajectory tracking error. R_h denotes the final expected reward, and x_d is the x-axis displacement of the foot relative to the ground during forward jumping. The weights used during training are ω_1=6, ω_2=8, and ω_3=−80. The specific training parameter settings are the same as in the vertical jumping task. The trained policy is deployed on the forward jumping platform, which is reconstructed based on the vertical jumping platform. The experimental results and relevant comparisons are presented in Fig. 10.

Compared to the baseline method, the horizontal distance the LLU jumped increased from 591 to 783 mm using the DRL method, representing a 32.5% improvement. Meanwhile, the landing compliance index was lower, specifically corresponding to 9.5% greater landing compliance. These results showcase the excellent performance and generalizability of the proposed DRL optimization method for multi-objective tasks.

To further assess the robustness of the proposed method under unknown disturbances, we introduced oil to alter the friction and placed a 20 cm-tall soft platform at the landing site. Without any retraining, the LLU successfully completed the jump, as shown in Fig. 11.

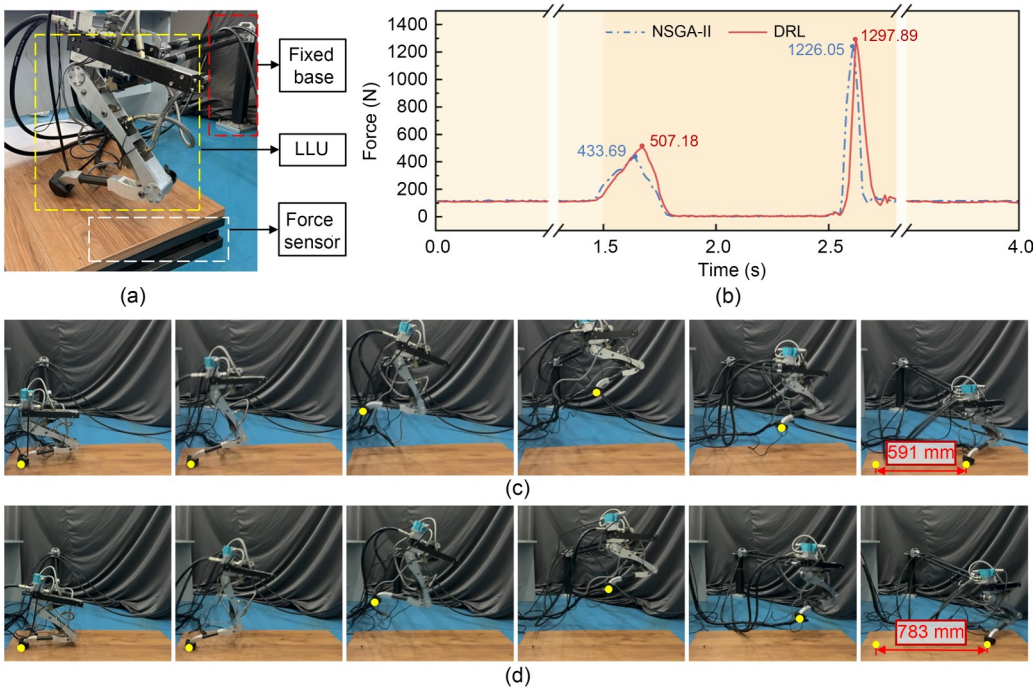

Fig. 10 Results and comparison of the forward jumping experiment: (a) forward jumping platform; (b) contact force results during forward jumping with different methods; (c) snapshots of the forward jumping experiment using the baseline method; (d) snapshots of the forward jumping experiment using the DRL method

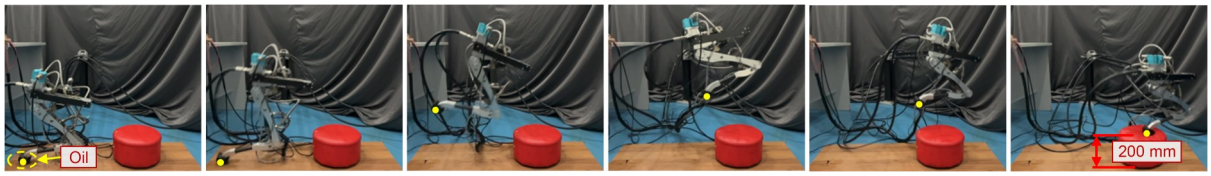

Fig. 11 Snapshots of the forward jumping experiment under external disturbances

This result highlights the method's strong control robustness, which is likely a consequence of the randomized DRL training.

5 Conclusions

A novel jumping motion control framework based on DRL was proposed for hydraulic LLU robots, with the goal of improving jumping performance. Experiments on a physical prototype demonstrated that the trained policies enhanced both the stability and landing compliance in continuous jumping motions. To the best of our knowledge, it is the first successful deployment of DRL on a hydraulic legged robot.

1. The proposed quasi-physical model accurately reflects the performance of the physical prototype, and enables the successful deployment of DRL-trained policies from simulations to real-world systems.

2. The DRL algorithm jointly optimizes the parameters of the trajectory generator and the motion tracking controller, enabling the LLU model to achieve continuous and stable jumping in the simulations following training.

3. The trained policy is successfully deployed on the physical prototype, achieving continuous jumping motions at the target height. In comparison to the NSGA-II algorithm, the height tracking error is reduced by over 60%, and the landing compliance is improved by up to 7.7%.

The proposed control framework can be extended to more complex tasks, such as the continuous traversal of obstacles with varying heights, or stair climbing. We deployed the fine-tuned trained policy on a forward jumping task with additional degrees of freedom, where it showed strong performance across multiple objectives. Additionally, this study provides a valuable reference regarding optimization and control of high-dynamic motions, and may prove useful for other types of legged robots.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (No. U24B2049) and the Scientific Research Fund of Zhejiang Provincial Education Department (No. Y202457047), China.

Author contributions

Junhui ZHANG: conceptualization, methodology, resources, writing–review & editing, and supervision. Pengyuan JI: conceptualization, formal analysis, writing–original draft, and data curation. Lizhou FANG: investigation, writing–review & editing, and visualization. Jinyuan LIU: investigation, data curation, and visualization. Dandan WANG: investigation, writing–original draft, and visualization. Jikun AI: investigation, data curation, and funding acquisition. Huaizhi ZONG: conceptualization, methodology, writing–review & editing, supervision, and funding acquisition. Bing XU: supervision and writing–review & editing.

Conflict of interest

Junhui ZHANG, Pengyuan JI, Lizhou FANG, Jinyuan LIU, Dandan WANG, Jikun AI, Huaizhi ZONG, and Bing XU declare that they have no conflict of interest.

References

Ahn D, Cho BK, 2022. Online jumping motion generation via model predictive control. *IEEE Transactions on Industrial Electronics*, 69(5):4957-4965. https://doi.org/10.1109/tie.2021.3078396

Ba KX, Chen CH, Ma GL, et al., 2024. A compensation strategy of end-effector pose precision based on the virtual constraints for serial robots with RDOFs. *Fundamental Research*, in press. https://doi.org/10.1016/j.fmre.2024.06.005

Ba KX, She JB, Xu B, et al., 2025. Matrix sensitivity-based adaptive iterative feedback control of leg hydraulic drive system of legged robot. *Control Engineering Practice*, 165:106557. https://doi.org/10.1016/j.conengprac.2025.106557

Bjelonic M, Sankar PK, Bellicoso CD, et al., 2020. Rolling in the deep-hybrid locomotion for wheeled-legged robots using online trajectory optimization. *IEEE Robotics and Automation Letters*, 5(2):3626-3633. https://doi.org/10.1109/lra.2020.2979661

Boaventura T, Medrano-Cerda GA, Semini C, et al., 2013. Stability and performance of the compliance controller of the quadruped robot HyQ. Proceedings of the IEEE/

RSJ International Conference on Intelligent Robots and Systems, p.1458-1464. https://doi.org/10.1109/IROS.2013.6696541

Choi S, Ji G, Park J, et al., 2023. Learning quadrupedal locomotion on deformable terrain. *Science Robotics*, 8(74): eade2256. https://doi.org/10.1126/scirobotics.ade2256

Duan P, Yu ZN, Gao KZ, et al., 2024. Solving the multi-objective path planning problem for mobile robot using an improved NSGA-II algorithm. *Swarm and Evolutionary Computation*, 87:101576. https://doi.org/10.1016/j.swevo.2024.101576

Egli P, Hutter M, 2022. A general approach for the automation of hydraulic excavator arms using reinforcement learning. *IEEE Robotics and Automation Letters*, 7(2):5679-5686. https://doi.org/10.1109/LRA.2022.3152865

Exarchos I, Jiang YF, Yu WH, et al., 2021. Policy transfer via kinematic domain randomization and adaptation. Proceedings of the IEEE International Conference on Robotics and Automation, p.45-51. https://doi.org/10.1109/ICRA48506.2021.9561982

Fang LZ, Zhang K, Lu ZY, et al., 2025. Adaptive robust joint force control for rapid motion of hydraulic limb leg units. *Control Engineering Practice*, 165:106596. https://doi.org/10.1016/j.conengprac.2025.106596

Gao HB, Liu YF, Ding L, et al., 2019. Low impact force and energy consumption motion planning for hexapod robot with passive compliant ankles. *Journal of Intelligent & Robotic Systems*, 94(2):349-370. https://doi.org/10.1007/s10846-018-0828-2

Gu Y, Yuan CZ, 2020. Adaptive robust trajectory tracking control of fully actuated bipedal robotic walking. Proceedings of the IEEE/ASME International Conference on Advanced Intelligent Mechatronics, p.1310-1315. https://doi.org/10.1109/AIM43001.2020.9158814

Han L, Zhu QX, Sheng JP, et al., 2024. Lifelike agility and play in quadrupedal robots using reinforcement learning and generative pre-trained models. *Nature Machine Intelligence*, 6(7):787-798. https://doi.org/10.1038/s42256-024-00861-3

Han YY, Liu GP, Lu ZY, et al., 2023. A stability locomotion-control strategy for quadruped robots with center-of-mass dynamic planning. *Journal of Zhejiang University-SCIENCE A*, 24(6):516-530. https://doi.org/10.1631/jzus.A2200310

Hoeller D, Rudin N, Sako D, et al., 2024. Anymal parkour: learning agile navigation for quadrupedal robots. *Science Robotics*, 9(88):eadi7566. https://doi.org/10.1126/scirobotics.adi7566

Hwangbo J, Lee J, Dosovitskiy A, et al., 2019. Learning agile and dynamic motor skills for legged robots. *Science Robotics*, 4(26):eaau5872. https://doi.org/10.1126/scirobotics.aau5872

Jaiswal S, Sopanen J, Mikkola A, 2021. Efficiency comparison of various friction models of a hydraulic cylinder in the framework of multibody system dynamics. *Nonlinear Dynamics*, 104(4):3497-3515. https://doi.org/10.1007/s11071-021-06526-9

Li X, Yu HY, Zong HZ, et al., 2024. Light weight design and integrated method for manufacturing hydraulic wheel-legged robots. *Journal of Zhejiang University-SCIENCE A*, 25(9):701-715. https://doi.org/10.1631/jzus.A2300343

Ma HP, Zhang YJ, Sun SY, et al., 2023. A comprehensive survey on NSGA-II for multi-objective optimization and applications. *Artificial Intelligence Review*, 56(12):15217-15270. https://doi.org/10.1007/s10462-023-10526-z

Nansai S, Rojas N, Elara MR, et al., 2015. A novel approach to gait synchronization and transition for reconfigurable walking platforms. *Digital Communications and Networks*, 1(2):141-151. https://doi.org/10.1016/j.dcan.2015.04.003

Schulman J, Wolski F, Dhariwal P, et al., 2017. Proximal policy optimization algorithms. arXiv:1707.06347. https://doi.org/10.48550/arXiv.1707.06347

Semini C, Barasuol V, Goldsmith J, et al., 2017. Design of the hydraulically actuated, torque-controlled quadruped robot HyQ2Max. *IEEE/ASME Transactions on Mechatronics*, 22(2):635-646. https://doi.org/10.1109/tmech.2016.2616284

Shao X, Fan YQ, Shao JP, et al., 2023. Improved active disturbance rejection control with the optimization algorithm for the leg joint control of a hydraulic quadruped robot. *Measurement and Control*, 56(7-8):1359-1376. https://doi.org/10.1177/00202940221100298

Spinelli FA, Egli P, Nubert J, et al., 2024. Reinforcement learning control for autonomous hydraulic material handling machines with underactuated tools. Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, p.12694-12701. https://doi.org/10.1109/IROS58592.2024.10802199

Sun YR, Hua ZS, Li YB, et al., 2021. Modeling and analysis on low energy consumption foot trajectory for hydraulic actuated quadruped robot. *International Journal of Advanced Robotic Systems*, 2021(11):1-12. https://doi.org/10.1177/17298814211062006

Xiang PJ, Yan L, Liu XS, et al., 2025. Structural topology design for electromagnetic performance enhancement of permanent-magnet machines. *Chinese Journal of Mechanical Engineering*, 38(1):26. https://doi.org/10.1186/s10033-025-01184-z

Xie ZM, Da XY, van de Panne M, et al., 2021. Dynamics randomization revisited: a case study for quadrupedal locomotion. Proceedings of the IEEE International Conference on Robotics and Automation, p.4955-4961. https://doi.org/10.1109/ICRA48506.2021.9560837

Yao B, 2009. Desired compensation adaptive robust control. *Journal of Dynamic Systems, Measurement, and Control*, 131(6):061001. https://doi.org/10.1115/1.3211087

Yao ZK, Xu FY, Jiang GP, et al., 2024. Data-driven control of hydraulic manipulators by reinforcement learning. *IEEE/ASME Transactions on Mechatronics*, 29(4):2673-2684. https://doi.org/10.1109/TMECH.2023.3336070

Zhang JH, Liu JY, Zong HZ, et al., 2025. Bridging the gap to

bionic motion: challenges in legged robot limb unit design, modeling, and control. *Cyborg and Bionic Systems*, 6:0365. https://doi.org/10.34133/cbsystems.0365

Zhang K, Zhang JH, Zong HZ, et al., 2025. High dynamic position control for a typical hydraulic quadruped robot leg based on virtual decomposition control. *IEEE/ASME Transactions on Mechatronics*, 30(4):2473-2484. https://doi.org/10.1109/tmech.2024.3446840

Zhu J, Payne JJ, Johnson AM, 2024. Convergent iLQR for safe trajectory planning and control of legged robots. Proceedings of the IEEE International Conference on Robotics and Automation, p.8051-8057. https://doi.org/10.1109/ICRA57147.2024.10611641

Zong HZ, Zhang JH, Jiang L, et al., 2024. Bionic lightweight design of limb leg units for hydraulic quadruped robots by additive manufacturing and topology optimization. *Bio-Design and Manufacturing*, 7(1):1-13. https://doi.org/10.1007/s42242-023-00256-0

Zong HZ, Lou B, Yuan HH, et al., 2025. Integrating kinematic and dynamic factors with generative design for high-performance additive manufacturing structures. *Virtual and Physical Prototyping*, 20(1):e2501383. https://doi.org/10.1080/17452759.2025.2501383