Multibody System Dynamics
https://doi.org/10.1007/s11044-024-10035-z

RESEARCH

Modeling and reinforcement learning-based locomotion control for a humanoid robot with kinematic loop closures

Lingling Tang[1] · Dingkun Liang[2,1] · Guang Gao[1] · Xin Wang[1] · Anhuan Xie[1]

Received: 22 March 2024 / Accepted: 18 October 2024

Abstract

Humanoid robots are complex multibody systems, and modeling and locomotion control for them are challenging tasks. In this paper, a rigid multibody model is first built for a homemade humanoid robot with kinematic loop closures. The inverse kinematics solutions based on geometric relationships are then presented for the parallel mechanisms of the knee and ankle joints, and contact detection procedures for foot–ground interactions on flat terrain and collisions between legs are simplified. Based on the above modeling work, a deep reinforcement learning (RL)-based strategy is presented for locomotion control. The reward function in the RL environment is well designed, where the foot periodic cycle penalty is implemented based on the complementary conditions of foot velocity and foot–ground interaction force. A new method is proposed to encourage the symmetric gait by penalizing the differences in the mean values and standard deviations between left and right joint angles, and the whole-body coordination is realized by tracking a pair of reference trajectories of the shoulder pitch degrees of freedom (DoFs). Finally, to verify the effectiveness of the proposed RL-based locomotion control strategy, we present several training cases, each with a separate RL agent, and the goals of foot periodic cycle with a frequency of 2 Hz, gait symmetry, forward speed up to 10 km/h, whole-body coordinated gait, and time-varying velocity command tracking are successfully achieved.

Keywords Humanoid robot · Inverse kinematics · Locomotion control · Reinforcement learning · Gait symmetry · Whole-body coordination

1 Introduction

As complex multibody systems, humanoid robots achieve the bipedal locomotion ability through foot–ground interactions and realize the human-like gaits via whole-body coordinated motions of upper and lower limbs [1]. Typically, humanoid robots are modeled as rigid multibody systems, encompassing forward and inverse kinematics and dynamics, as well as contact detection and contact dynamics. This modeling forms the foundation for locomotion

✉ D. Liang
liangdk@mail.nankai.edu.cn

[1] Zhejiang Lab, Hangzhou, 311100, China

[2] College of Information Engineering, Zhejiang University of Technology, Hangzhou, 310023, China

Published online: 28 October 2024

control. However, the series-parallel hybrid leg mechanism, the underactuated feature of the robot with a floating base, the non-smooth nature of the foot–ground interactions, and the nonlinear response of the actuators make the modeling and locomotion control for humanoid robots challenging.

The current structure designs for humanoid robots tend to employ lightweight materials and put the knee and ankle joint motors away from the corresponding joints [2], thereby reducing the end inertia and achieving higher speed and more agile locomotion. This arrangement means additional parallel mechanisms need to be introduced for motion and force transmissions, which complicates both the forward and inverse kinematics and dynamics solutions. For knee joints driven by planar multi-link mechanisms, this problem can be solved by well-established methods [3]. The same is true for an ankle joint driven by a planar four-bar mechanism, where there is only one revolute joint degree of freedom (DoF) between the foot and the shank, representing the ankle pitch motion, like that in the robots Cassie [4] and BRUCE [5]. However, this problem becomes difficult for an ankle joint driven by a pair of spatial parallel mechanisms, since there are two successive revolute DoFs between the foot and the shank, representing the ankle pitch and roll motions, respectively. Pfeiffer [6] presented a forward kinematics solution for the ankle joint of the humanoid robot Johnnie. Based on the constraint equations of kinematical closed loops, given the spindle positions, the roll and pitch angles of the ankle joint can be solved iteratively using Newton–Raphson algorithm. Buschmann [7] proposed an inverse kinematics solution for the ankle joint of the humanoid robot Lola. The ankle joint is driven by a pair of spatial slider-crank mechanisms, where the linear motions are produced by planetary roller screws. According to geometric relationships, the analytical expression for the roller screw nut displacement as a function of the ankle roll and pitch angles can be derived. For the ankle joint with two DoFs driven by a pair of spatial four-bar mechanisms, the inverse kinematics solution can be obtained by solving algebraic equations derived from geometric relationships [8], but this usually requires a further determination of the correct one from multiple solutions. Therefore, the derivation of analytical solution based purely on geometric relationships is preferred. Nevertheless, to the knowledge of the authors, there were few related studies publicly reported before. For contact modeling of humanoid robots, one primary work is contact detection. If the potential contact points on one body remain almost unchanged, the contact detection problem can be simplified using the master-slave approach [9]. In this study, we focus on the locomotion on flat terrain, and the four corner points on the sole can be chosen as the slave points and represented by mass-less spheres. For the scenario of collision avoidance [10], like that between legs, another simplification can be introduced. Since no collision is allowed in this case, the contact detection can be replaced by the overlap test between bounding volumes, like the oriented bounding boxes, and no overlap between the bounding volumes is a sufficient condition for collision avoidance [11]. Another essential part of contact modeling is solving the contact constraints, including the normal non-penetration condition and the tangential no-slip condition in the static friction phase. As the real time simulation is expected in the locomotion control for humanoid robots, the non-smooth methods [12–14] are usually employed to solve these unilateral constraints.

The aim of locomotion control is to achieve stable bipedal motion of humanoid robots, including standing, walking, and running gaits. For model-based locomotion control methods, in order to realize the above goal, the appropriate gait is usually first planned using trajectory optimization, and a subsequent stabilizer is employed to control the actual trajectory to track the reference trajectory. The gait planning process generally relies on some reduced-order models, such as a linear inverted pendulum model [15] or a single rigid body model [16], to obtain the trajectories of the center of mass and the supporting feet. The trajectories of all joint DoFs are then calculated using inverse kinematics. For joints driven by parallel mechanisms, the rotation angles and torques of the joint motors need to be further determined through inverse kinematics and dynamics, respectively. In comparison, the deep RL-based locomotion control approaches can handle the full dynamics of humanoid robots. They also have the end-to-end advantage of directly generating actions for the joint motors based on the received state observations [17, 18]. This advantage is particularly important for humanoid robots with kinematic loop closures, since the inverse kinematics and dynamics solution process from a series model to a parallel model can be omitted. Furthermore, the RL-based approaches are able to explore more human-like gaits through well-designed reward functions. Therefore, it is more appropriate to adopt the RL-based locomotion control strategy for humanoid robots with kinematic loop closures. The deep RL neural networks can be trained to realize the bipedal walking of humanoid robots even using the simplest reward function. However, these results typically do not include some advanced locomotion abilities such as foot periodic cycle with a prescribed frequency, gait symmetry, and whole-body coordination, which are important for a stable and healthy gait. To achieve these advanced abilities, previous research focused on modifying the deep RL network architecture by introducing recurrent neural networks with memory units [19–21], or by using curriculum learning frameworks [22–25] or adversarial imitation learning frameworks [26–28]. Nevertheless, these modifications are often very complicated, and they are more like solutions proposed by the computer science community than the mechanics or mechanical engineering community. Therefore, how to extract the intrinsic characteristics of bipedal locomotion and achieve the above goals using only well-designed reward functions remains a challenging work. Moreover, little attention has been paid to this aspect before.

The objective of this paper is to model a home-made humanoid robot with kinematic loop closures and achieve the locomotion control goals based on the RL strategy. The remainder of the paper is organized as follows. Section 2 first gives a rigid multibody model of the humanoid robot. The inverse kinematics based on geometric relationships are then presented for the loop closures of the knee and ankle joints, and the contact modeling including contact detection and contact constraint solution is discussed. Based on the above modeling work, we establish an RL environment for locomotion control in Sect. 3, including the fundamental framework, observation space, action space, reward function, and termination conditions. The proposed reward function is well designed to achieve advanced abilities such as the foot periodic cycle, the gait symmetry, and the whole-body coordination. Section 4 presents the RL algorithm and hyperparameters adopted in this paper. In Sect. 5, the inverse kinematics results for parallel mechanisms are first given, which is related to the action space of RL. In order to verify the effectiveness of the proposed RL-based locomotion control strategy, several training cases (each with a separate RL agent) are discussed, including the influence of clock phase observation on RL training, symmetric gaits at speeds of 6 km/h and 10 km/h, whole-body coordinated gaits, and four cases with time-varying velocity commands. Finally, some concluding remarks are summarized in Sect. 6.

2 Multibody model of the humanoid robot

A rigid multibody model of a home-made humanoid robot is presented in this section, including model description, kinematic topology structure, inverse kinematics of parallel mechanisms, and contact modeling.

2.1 Model description

The humanoid robot is an underactuated rigid multibody system with a floating pelvis that can freely move with respect to the global reference frame, as shown in Fig. 1. Since we

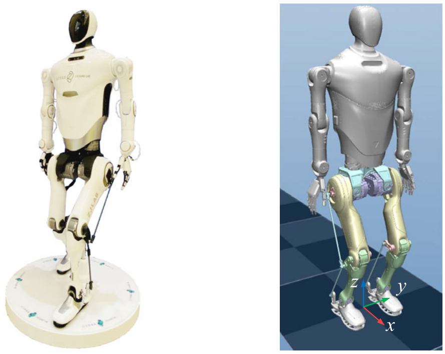


Fig. 1 Physical prototype and simulation model of the humanoid robot

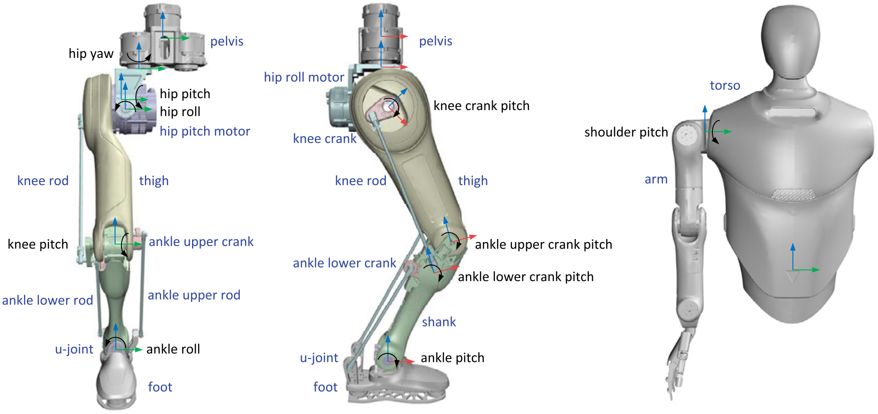


Fig. 2 Kinematic topology structure of the humanoid robot with only right side. Front view of lower limbs (left), right view of lower limbs (middle), and front view of upper limbs (right)

focus on the legged motions of the humanoid robot in this study, a reduced model with 18 DoFs (6 for the pelvis and 6 for each leg) is mainly employed for modeling and locomotion control. Only in the case of whole-body coordinated motions, an improved model with 20 DoFs will be used, which additionally includes the left and right shoulder pitch DoFs.

The global reference frame is established on the ground and defined as follows: the x axis points forward and is perpendicular to the coronal plane, the z axis is along the vertical upward direction, and the y axis is perpendicular to the sagittal plane and points from the right side of the robot to the left. When all joint angles are zero, the reference frame for each body is parallel to the global reference frame.

The dynamic modeling of the humanoid robot is based on the MuJoCo simulation platform [29], where the kinematics and dynamics parameters are described by an XML file format called MJCF. The smooth dynamics is implemented using the composite rigid body algorithm and the recursive Newton–Euler algorithm [30], and the semi-implicit Euler method is employed here for time integration.

2.2 Kinematic topology

Since the structures on the left and right sides are symmetric, without loss of generality, the kinematic topology of the humanoid robot with only right side is shown in Fig. 2.

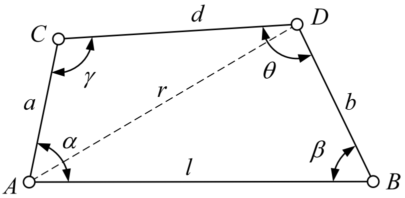


Fig. 3 Planar four-bar mechanism of the knee joint

The pelvis is a floating base moving freely relative to the ground, and the torso is fixed directly to the pelvis. The upper limb can be treated as a whole, that is, the arm. In the case of whole-body coordinated locomotion, it has one shoulder pitch DoF relative to the torso. Otherwise, it is fixed directly to the torso.

The thigh is connected to the pelvis via three revolute joints, and the rotation sequence represents the hip yaw, hip roll, and hip pitch motions, in that order. The shank and the thigh are interconnected through a revolute joint representing the knee pitch DoF. In addition, the foot is connected to the shank by a universal joint body (denoted as a u-joint in Fig. 2), where two rotational DoFs are the ankle pitch motion and a subsequent ankle roll motion.

The knee pitch, the ankle pitch, and the ankle roll motions are driven through parallel mechanisms rather than directly by joint motors like the three hip DoFs. Here, the knee motor is placed on the thigh, and the knee joint is actuated by a planar four-bar mechanism. Similarly, the ankle upper and lower motors are placed on the shank, and two spatial four-bar mechanisms are adopted together to achieve the pitch and roll motions of the ankle joint. Thus, there are three kinematic loop closures for each leg. It should be noted that the rods in the four-bar mechanisms are connected at both ends by spherical joints, and each revolute joint has its physical upper and lower limits.

2.3 Inverse kinematics of parallel mechanisms

The action space of RL is related to the rotation angle limits of joint motors. For parallel mechanisms, these limits are determined through inverse kinematics, which are discussed in this subsection.

2.3.1 Planar four-bar mechanism of the knee joint

The planar four-bar mechanism of the knee joint is shown in Fig. 3, where the line segment AB stands for the thigh, AC represents the knee crank, CD represents the knee rod, and BD is fixed to the shank. Here, the lengths a, d, b, and l are all constant. The inverse kinematics problem is formulated as given the knee pitch angle β, calculate the knee crank pitch angle α.

Denoting the length of line segment AD as $r = \sqrt{b^2 + l^2 - 2bl\cos\beta}$, the knee crank pitch angle α can be calculated as

$$\alpha = \cos^{-1}\left(\frac{r^2 + l^2 - b^2}{2rl}\right) + \cos^{-1}\left(\frac{r^2 + a^2 - d^2}{2ra}\right). \tag{1}$$

This solution is simple and effective. As a comparison, Ref. [3] proposed an algebraic approach for calculating the angle α, which is a bit complicated and requires further determination of the solution from two configurations. It is worth mentioning that the nominal rotation of knee crank is denoted as $\Delta\alpha = \alpha - \alpha_0$, and α_0 is the angle corresponding to the configuration where the nominal knee pitch angle $\Delta\beta$ is zero.

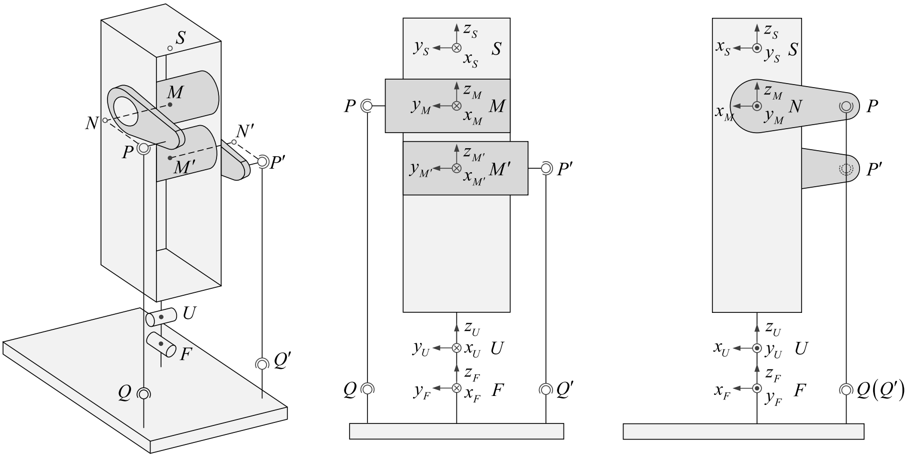


Fig. 4 Configuration of the right ankle joint with all joint angles as zero: isometric view (left), back view (middle), and left view (right)

2.3.2 Spatial four-bar mechanisms of the ankle joint

The spatial four-bar mechanisms of the left and right ankle joints are symmetric about the sagittal plane, as shown in Fig. 6. Without loss of generality, the right one is mainly investigated here. Figure 4 represents the configuration of the right ankle joint with all joint angles as zero, where all reference frames are parallel to each other. In this figure, the shank, the u-joint, the foot, the ankle upper motor, and the ankle lower motor are denoted as symbols S, U, F, M, and M', respectively.

The inverse kinematics problem is stated as given the ankle pitch angle θ and the ankle roll angle ϕ, calculate the ankle upper crank pitch angle α and the ankle lower crank pitch angle α'. We focus on the calculation of α, as the derivation of α' is similar.

The ankle joint first pitches by θ angle, and then rolls by ϕ angle. Thus, the rotation matrix of the reference frame of the foot with respect to that of the shank is

$$\mathbf{A}_{SF} = \mathbf{A}_{SU}\mathbf{A}_{UF}, \quad \mathbf{A}_{SU} = \begin{bmatrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{bmatrix}, \quad \mathbf{A}_{UF} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}. \tag{2}$$

Define an auxiliary point N on the ankle upper motor axis, which has the same y coordinate in the reference frame of the shank as the upper spherical joint point P of the ankle upper rod PQ. The relative position vector between the auxiliary point N and the lower spherical joint point Q of the rod in the reference frame of the shank is written as

$${}_S\mathbf{r}_{NQ} = {}_S\mathbf{r}_{UQ} - {}_S\mathbf{r}_{UN} = \mathbf{A}_{SF}\,{}_F\mathbf{r}_{UQ} - {}_S\mathbf{r}_{UN}, \tag{3}$$

where ${}_F\mathbf{r}_{UQ}$ and ${}_S\mathbf{r}_{UN}$ are both constant vectors. Here, the prefix subscript F or S represents the name of the reference frame, and the suffix subscripts indicate the two points for calculating the relative position vector. Furthermore, the coordinate components of the relative position vector on the x_S, y_S, and z_S axes are denoted as

$$\bar{x} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} {}_S\mathbf{r}_{NQ}, \quad \bar{y} = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix} {}_S\mathbf{r}_{NQ}, \quad \bar{z} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} {}_S\mathbf{r}_{NQ}. \tag{4}$$

Fig. 5 Calculation of the ankle upper crank pitch angle α

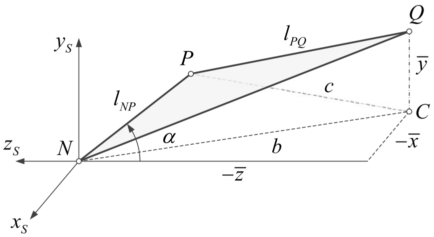
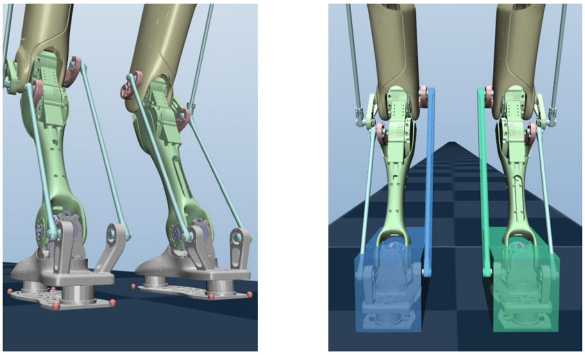


Fig. 6 Contact detection for foot–ground interactions (left) and collisions between legs (right)



The ankle upper crank pitch angle $\alpha \in (0, \pi)$ can be defined as the angle between the line segment NP and the $-z_S$ axis of the shank, as shown in Fig. 5. The projected lengths of line segments NQ and PQ on the $x_S z_S$ plane are

$$b = \sqrt{\bar{x}^2 + \bar{z}^2}, \quad c = \sqrt{l_{PQ}^2 - \bar{y}^2}. \tag{5}$$

Thus, the ankle upper crank pitch angle α can be calculated as

$$\alpha = \tan^{-1}\left(\frac{\bar{x}}{\bar{z}}\right) + \cos^{-1}\left(\frac{l_{NP}^2 + b^2 - c^2}{2 l_{NP} b}\right), \tag{6}$$

where the crank length l_{NP} and the rod length l_{PQ} are both constant scalars.

The nominal rotation of the ankle upper crank $\Delta\alpha = \alpha - \alpha_0$, and α_0 is the angle corresponding to the configuration where the nominal ankle pitch angle $\Delta\theta$ and ankle roll angle $\Delta\phi$ are both zero. The proposed method is simple and effective, and the derivation is purely based on geometric relationships.

2.4 Contact modeling

The contact modeling is a bit complicated but plays an important role in the locomotion of humanoid robots. In this study, we focus on the locomotion on flat terrain, and only foot–ground interactions and collisions between legs are considered in the contact modeling. Details about the contact detection and the contact constraints are presented as follows.

The potential contact points on the foot are usually the four corner points of the sole. The contact detection procedure can be simplified by introducing four mass-less spheres with small radius fixed around the sole, as shown in Fig. 6. Therefore, the foot–ground

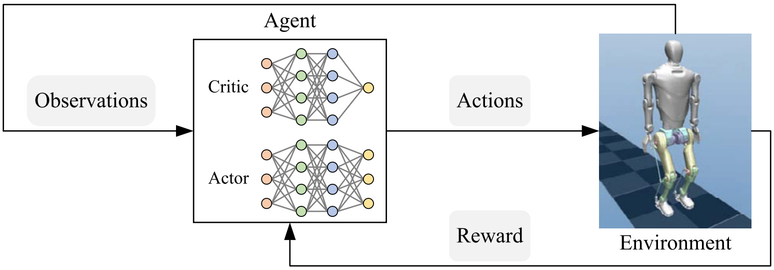


Fig. 7 Fundamental framework of deep RL

interactions can be approximated as the contact between the sole spheres and the flat ground. In order to detect the potential collisions between legs, two bounding boxes are introduced for each leg, which are fixed on the foot and the ankle upper rod, respectively. The collision detection problem can be simplified to an overlap test between the left and right bounding boxes.

The unilateral contact constraints are computed using a non-smooth constraint solver [13] provided by the MuJoCo simulation platform. The constraints are implemented at the acceleration level, and a constraint impedance parameter is introduced to interpolate the constrained acceleration between a reference acceleration and an unconstrained acceleration, where the reference acceleration is expressed as a spring-damper model, similar to the Baumgarte constraint stabilization method [31]. The calculation of the contact constraint forces, including the normal force and the tangential frictional forces, is implemented based on the pyramidal friction cone.

3 RL environment for locomotion control

Based on the modeling work, we start to investigate the locomotion control for the humanoid robot. As mentioned in Sect. 1, the RL-based locomotion control strategy has the end-to-end advantage that it can directly generate actions for the joint motors. This advantage is especially useful for humanoid robots with kinematic loop closures. Therefore, an RL environment for locomotion control is established in this section. We first briefly review the fundamental framework of deep RL and then present the details of the RL environment, including the observation space, action space, reward function, and termination conditions.

3.1 Fundamental framework

The fundamental framework of deep RL is depicted in Fig. 7, which consists of two main components: the agent and the environment. The agent consisting two deep neural networks interacts with the environment and acquires locomotion skills through trial and error [17]. That is, the agent receives observations from the environment and generates actions for the environment. The agent also receives a reward from the environment, and the goal of deep RL training is to find the optimal policy, which is a mapping of observations to actions that maximizes the cumulative reward obtained during the locomotion. In the deep RL, the policy is typically represented by a neural network called actor network, where the tunable policy parameters during training are the weights and bias of the neural network.

Table 1 Observation space

Observation	Expression	Dimension
Pelvis lateral displacement	y	1
Pelvis height change	Δz	1
Pelvis orientation	α, β, γ	3
Pelvis linear velocities	v_x, v_y, v_z	3
Pelvis angular velocities	$\omega_x, \omega_y, \omega_z$	3
Joint angles	$\theta_{L1} \cdots \theta_{L6}, \ \theta_{R1} \cdots \theta_{R6}$	12
Joint angular velocities	$\dot{\theta}_{L1} \cdots \dot{\theta}_{L6}, \ \dot{\theta}_{R1} \cdots \dot{\theta}_{R6}$	12
Previous actions	$\mathbf{a}_{t-1}$	12
Clock phases	$\cos(2\pi t/T), \sin(2\pi t/T)$	2
Velocity command	v_x^{cmd}	1

In this paper, we follow the steps below to set up the RL environment. First, a dynamics model is established based on the MuJoCo 2.0 simulation platform. The simulation time step is set to 1 ms, and the actions are output at a sampling time interval of $T_s = 0.025$ s. Then, the mujoco-py 2.0.2.8 package[1] is adopted, which provides a Python application programming interface to access dynamic parameters and variables in the simulation model. Finally, by inheriting and modifying the environment class template provided by the OpenAI Gym 0.21.0 package [32], the customized RL environment is built.

3.2 Observation space

The RL environment outputs a 50-dimensional observation space, as shown in Table 1, including 11 floating base state variables, 12 joint angles and 12 joint angular velocities, 12 actions of the previous time step, 2 clock phases, as well as 1 forward velocity command. Since we are concerned with the forward motion of the humanoid robot, there is only one velocity command here. The orientation of the pelvis is represented by three Cardan angles α, β, and γ in the sequence of $X_1Y_2Z_3$. The Cardan angles have clear physical meanings and can be directly used to describe the rotations of the pelvis from the upright configuration.

Considering that different observations have different orders of magnitude and different units, observations collected from multiple parallel environments are usually first normalized by subtracting the mean value and dividing by the standard deviation before further calculations [33]. It is worth mentioning that in the case of whole-body coordinated motions, the dimension of observation space increases to 56. This is due to the changes in joint angles, joint angular velocities, and previous actions, since 2 additional shoulder pitch DoFs are included.

3.3 Action space

The RL environment receives a 12-dimensional action space from the agent. The actions $\mathbf{a}$ are defined as the normalized angle targets of 12 joint motors, which has been proven to perform better than directly defining actions as joint motor torques [34]. These normalized angle targets are then converted to the angle targets through scaling and offset as follows:

$$\theta_{di} = \frac{\theta_{di}^H - \theta_{di}^L}{2} a_i + \frac{\theta_{di}^H + \theta_{di}^L}{2}, \tag{7}$$

[1] https://github.com/openai/mujoco-py.

Table 2 Reward function, including baseline rewards (indicated by light gray background) and auxiliary rewards

Reward	Function (r_k)	Weight (w_k)
Pelvis velocity tracking	$\exp(-2(v_x - v_x^{\mathrm{cmd}})^2)$	0.25
Power consumption	$\sum \lvert \tau_i \dot{\theta}_i \rvert$	-0.00002
Alive bonus	T_s	0.2
Pelvis lateral displacement	y^2	-1
Pelvis vertical displacement	Δz^2	-5
Pelvis orientation deviation	$\lvert\alpha\rvert + \lvert\beta\rvert + \lvert\gamma\rvert$	-0.07
Foot periodic cycle	$\Phi_{\mathrm{left\_foot\_cycle}} + \Phi_{\mathrm{right\_foot\_cycle}}$	-0.07
Gait symmetry	$1 - \exp(-20\Delta\mu_i^2) + 1 - \exp(-20\Delta\sigma_i^2)$	-0.05
Whole-body coordination	$(\theta_{\mathrm{left\_shldr}} - \theta_{\mathrm{left\_shldr}}^{\mathrm{ref}})^2 + (\theta_{\mathrm{right\_shldr}} - \theta_{\mathrm{right\_shldr}}^{\mathrm{ref}})^2$	-0.5
Action rate	$\lVert \mathbf{a}_t - \mathbf{a}_{t-1} \rVert^2$	-0.004
Action smoothness	$\lVert \mathbf{a}_t - 2\mathbf{a}_{t-1} + \mathbf{a}_{t-2} \rVert^2$	-0.004

where a_i is the ith action, θ_{di}^L and θ_{di}^H are the lower and upper limits of ith motor joint, respectively. For parallel mechanisms, these limits can be determined using the inverse kinematics solutions of Sect. 2.3.

The torques of the joint motors are computed via proportional-derivative controllers, which track the joint angle targets, that is,

$$\tau_i = K_p(\theta_i - \theta_{di}) + K_d\dot{\theta}_i. \tag{8}$$

Here, θ_i is the actual angle of ith motor joint, $\dot{\theta}_i$ is the actual angular velocity, $K_p = 100$ N m/rad and $K_d = 10$ N m s/rad are the proportional and derivative gains, respectively.

Similar to the observation space, the dimension of action space will increase to 14 when the shoulder pitch DoFs are released in whole-body coordinated motions. In addition, the proportional and derivative gains for the shoulder pitch DoFs are chosen to be $K_p = 20$ N m/rad and $K_d = 2$ N m s/rad since the peak torques of the shoulder pitch motors are much smaller than those of the lower limb motors.

3.4 Reward design

The RL rewards listed in Table 2 are divided into two categories: the baseline rewards and the auxiliary rewards, and the total reward is the weighted sum of all components as $r = \sum w_k r_k$.

Since the forward velocity of the humanoid robot is the main focus of this study, the baseline rewards are designed to accomplish this goal, including the velocity tracking reward, the power consumption penalty, the alive bonus reward, the lateral and vertical displacement penalties, and the orientation deviation penalty. On this basis, additional auxiliary rewards are adopted to achieve advanced abilities such as the foot periodic cycle, the gait symmetry, the whole-body coordination, and the action smoothness. The remainder of this section explains the reward design in detail.

3.4.1 Baseline rewards

The pelvis velocity tracking function has the form similar to the probability density function of normal distribution, which ensures that the reward reaches a maximum when the actual velocity v_x approaches the command v_x^{cmd} and quickly decreases to zero when v_x is away from v_x^{cmd}. It is worth mentioning that this velocity tracking function applies to all RL training cases discussed in this paper, with the exception of instances where the humanoid robot directly accelerates from standstill to 10 km/h. For the latter, three alternative velocity tracking functions are proposed: a coefficient-modified version $\exp(-0.5(v_x - v_x^{\mathrm{cmd}})^2)$ of the original function, a minimum value function $\min(v_x/v_x^{\mathrm{cmd}}, 1))$, and a modified absolute value function $1 - |v_x/v_x^{\mathrm{cmd}} - 1|$. Comparative results for velocity tracking in this special case are presented in Sect. 5.3.

The power consumption penalty can avoid excessive torques or high angular velocities of the joint motors and achieve an energy-efficient gait.

The alive bonus reward here is equal to the sampling time interval T_s, which is a time-invariant constant. It is used to encourage the agent to learn a more stable locomotion policy and avoid triggering termination conditions too early.

The last three penalty terms are related to the position and orientation of the pelvis, ensuring that the humanoid robot walks in an upright posture along a straight line and the center of mass is always maintained at a certain height. It should be noted that these three penalties are consistent with the termination conditions defined in Sect. 3.5.

3.4.2 Foot periodic cycle penalty

During bipedal locomotion of the humanoid robot, the foot–ground interaction status periodically switches between the stance phase and the swing phase. The authors previously proposed a method for foot periodic cycle, which is implemented by penalizing the normal contact force in the form

$$r_k = 1 - \exp(-|f_N/G|), \tag{9}$$

where f_N is the normal contact force of one foot and G is the gravity force of the robot.

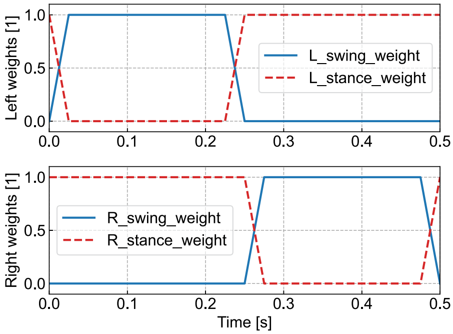
One obvious shortcoming of the above method is the absence of frequency information, and the final gait completely relies on the exploration of the agent. In this section, we adopt an alternative approach inspired by Ref. [21, 35] to achieve the goal of foot periodic cycle with a prescribed frequency.

It is observed that in the stance phase, the foot keeps in contact with the ground without sliding, which indicates the foot velocity is zero while the foot–ground interaction force is non-zero. On the contrary, in the swing phase, the foot can move freely relative to the ground, which means the foot–ground interaction force vanishes while the foot velocity is non-zero. Thus, the reward function is designed based on the complementary conditions of the foot velocity and the foot–ground interaction force. That is, the foot velocity is penalized in the stance phase, and the foot–ground interaction force (especially the normal contact force) is penalized in the swing phase. The corresponding function expressions are as follows:

$$\Phi_{\mathrm{stance}} = \|\mathbf{v}_{\mathrm{foot}}\|, \quad \Phi_{\mathrm{swing}} = |f_N/G|, \tag{10}$$

where $\mathbf{v}_{\mathrm{foot}}$ is the velocity vector of the foot. In the experiments of RL training, it is observed that when the velocity command v_x^{cmd} is less than 6 km/h, the off-ground heights of both feet are often too low. In order to solve this problem, the above swing phase penalty is modified by adding a height penalty term $40(z_{\mathrm{apex}} - z_{\mathrm{foot}})^2$ in this case, where z_{apex} is the desired apex height, and z_{foot} is the actual height [36].

Fig. 8 Weight coefficients for foot periodic cycle over one period



The penalty function for a single foot during the entire period is the weighted sum of the above two components.

$$\Phi_{\text{foot\_cycle}} = w_{\text{stance}} \Phi_{\text{stance}} + w_{\text{swing}} \Phi_{\text{swing}}. \tag{11}$$

The weight coefficients determine when the transition between the stance and swing phases occurs, and they satisfy the relation $w_{\text{stance}} + w_{\text{swing}} = 1$.

A simple but effective way to define the weight coefficients is to use piecewise linear functions [37]. Figure 8 shows the time evolution of the weight coefficients over one period, in which the swing phase accounts for approximately 50% of the entire cycle. The frequency is the reciprocal of the period $T = 0.5$ s, and a 2 Hz frequency is adopted for all RL training cases in this study.

The left and right feet have the same form of penalty function, and the only difference is a phase offset of $T/2$ in the evolution of the weight coefficients. The total periodic cycle penalty function is the sum of left and right components, as shown in Table 2. It is worth mentioning that, according to the weight coefficients in Fig. 8, the left foot will switch from the stance phase to the swing phase at instants of integer multiples of the period T, and simultaneously the right foot will switch from the swing phase to the stance phase. This is verified by the results in Sect. 5.

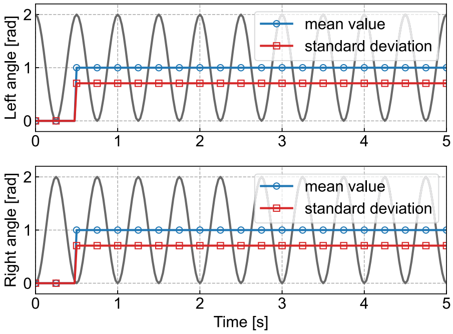
3.4.3 Gait symmetry penalty

The normal locomotion of a humanoid robot is usually accompanied by a symmetric gait. Ideally, the time evolution of the corresponding joint angles on the left and right sides should be the same, with only a phase offset of $T/2$. However, without additional special treatments, abnormal gaits with one leg in front and the other behind are often observed during RL training.

One way to address this problem is to introduce an mirror symmetry loss term in the objective function of RL policy optimization, as shown in Ref. [22]. Several other approaches are similar and require particular choices of policy network structure or special data duplication [38]. The implementations of these methods are a bit complicated, especially when they involve modifications of the neural network.

In this study, we propose a new method based on the reward function design to encourage the symmetric gait. To the knowledge of the authors, no similar work has been publicly reported before. The proposed method can be explained by Fig. 9. When the humanoid robot walks at a constant speed, good periodic characteristics can be found in the joint angles, and statistical variables such as the mean value and the standard deviation over a moving period

Fig. 9 Mean values and standard deviations of left and right joint angles in a symmetric gait



Algorithm 1: Calculation of mean value and standard deviation over a moving period

```
Data: Period $T$, sampling time interval $T_s$, index $i$
Result: Mean value $\mu_i$, standard deviation $\sigma_i$
$n = \mathrm{round}(T/T_s)$   // Number of sampling steps over one period
$\mathbf{\Theta} = \mathbf{0}_{n\times 1}$   // Array for storing the angle $\theta_i$
$c = 0$   // Counter for monitoring steps
for $k \leftarrow 1$ to $k_{\max}$ do
    Simulate the RL environment with actions $\mathbf{a}$
    $\theta_i \leftarrow \mathbf{q}(i)$   // Extract $\theta_i$ from generalized coordinate vector $\mathbf{q}$
    $j = \mathrm{mod}(c, n)$   // Modulo operation returns the remainder
    $\mathbf{\Theta}(j) = \theta_i$   // Update the storage array
    if $c \leq 3n$ then
        $\mu_i = 0$
        $\sigma_i = 0$
    else
        $\mu_i = \mathrm{sum}(\mathbf{\Theta})/n$   // Start calculation after 3 gait cycles
        $\sigma_i = \sqrt{\|\mathbf{\Theta}\|^2/n - \mu_i^2}$
    end
    $c \leftarrow c + 1$   // Update the counter
end
```

are approximately invariant. In the case of symmetric gait, the mean values and standard deviations of left and right joint angles calculated over a moving period should be equal, except for the first few periods. Therefore, the gait symmetry can be achieved by penalizing the differences in the mean values and standard deviations between left and right motor joint angles, as shown in Table 2. It should be noted that, according to the definition of reference frames in Sect. 2.1, the symmetry between left and right hip yaw angles has an additional negative sign, so does the hip roll angles. Algorithm 1 shows how to calculate the mean value μ_i and the standard deviation σ_i of the joint angle θ_i over a moving period in detail, where the gait symmetry penalty is activated after 3 periods (that is, 1.5 s).

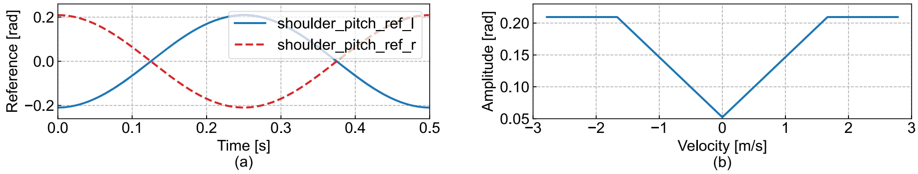


Fig. 10 Reference trajectories of shoulder pitch angles. (a) Evolution over one period, (b) relation between amplitude and forward velocity command

3.4.4 Whole-body coordination penalty

The whole-body coordinated locomotion requires releasing the shoulder pitch DoFs of both arms. Here, the motions of the arms are not directly explored by the agent. Instead, a pair of manually designed periodic reference trajectories are provided for the shoulder pitch angles, as illustrated by Fig. 10(a). In order to be consistent with the periodic alternation of stance and swing phases in Fig. 8, the cosine curves are adopted here to ensure the coordination of upper and lower limb movements.

One important thing to mention is that the shoulder pitching velocity should follow the forward velocity command, rather than remaining constant. Since the frequency is prescribed as 2 Hz in this study, the change in shoulder pitching velocity is reflected in the amplitude of angle. As shown in Fig. 10(b), a piece-wise linear function is employed to represent the relation between the amplitude and the velocity command v_x^{cmd}. Here, the amplitude has a minimum value of $\pi/60$ rad when $v_x^{\mathrm{cmd}} = 0$ km/h (i.e., stepping in place) and a maximum value of $\pi/15$ rad when $|v_x^{\mathrm{cmd}}| \geq 6$ km/h. The choice of maximum amplitude is inspired by observations of whole-body coordinated human gaits.

Similar to the gait symmetry penalty, the whole-body coordination penalty is not fully activated until after three complete periods, and a linear transition of amplitude from zero to its maximum is employed during the first three periods. Otherwise, the initial nonzero values of the reference trajectories may cause the training fail to converge.

The final form of the reference trajectory is defined as follows:

$$\theta^{\mathrm{ref}} = \min\left(\frac{t}{3T}, 1\right) \min\left(\frac{\pi}{60} + \frac{3\pi}{100}|v_x^{\mathrm{cmd}}|, \frac{\pi}{15}\right) \cos\left(\frac{2\pi t}{T}\right). \tag{12}$$

When the humanoid robot moves forward, the reference trajectories of both shoulder pitch DoFs are $\theta^{\mathrm{ref}}_{\mathrm{left\_shldr}} = -\theta^{\mathrm{ref}}$ and $\theta^{\mathrm{ref}}_{\mathrm{right\_shldr}} = \theta^{\mathrm{ref}}$. While in the case of backward locomotion, the result is the opposite, that is, $\theta^{\mathrm{ref}}_{\mathrm{left\_shldr}} = \theta^{\mathrm{ref}}$ and $\theta^{\mathrm{ref}}_{\mathrm{right\_shldr}} = -\theta^{\mathrm{ref}}$.

By penalizing the differences between the actual and the reference angles of shoulder pitch DoFs as in Table 2, the desired whole-body coordinated gait can be achieved.

3.4.5 Action smoothness penalty

The purpose of the action smoothness penalty is to eliminate high-frequency oscillations in joint angles, which are harmful to the motors and may result in an unhealthy gait. The implementations involve penalizing both the difference and the second-order difference between previous and current actions, which correspond to the gradient and the smoothness of the actions, respectively.

Table 3 Hyperparameters of PPO algorithm

Hyperparameter	Variable name	Value
Number of parallel environments	n_envs	32
Mini-batch size	batch_size	256
Number of steps for each environment per update	n_steps	256
Discount factor	gamma	0.95
Learning rate	learning_rate	0.00003
Entropy loss coefficient	ent_coef	0.001
Clipping parameter	clip_range	0.2
Number of epoch when optimizing the loss	n_epochs	5
Maximum value for gradient clipping	max_grad_norm	2

3.5 Termination conditions

In RL, termination conditions are used to end an episode and reset the training when certain observation values become unreasonable, which can avoid further meaningless exploration of actions. Within the scope of this study, three termination conditions are introduced as follows, all of which are related to the position or orientation of the floating base. These conditions include (1) the lateral displacement of the pelvis exceeds an upper limit, that is, $|y| > 1$, (2) the falling height of the pelvis is too large, that is, $\Delta z \leq -0.5$, (3) the orientation of the pelvis is away from the upright posture, that is, $|\alpha| \geq \pi/4$ or $|\beta| \geq \pi/4$ or $|\gamma| \geq \pi/4$.

4 RL algorithm and hyperparameters

The deep RL algorithms can be divided into two categories: off-policy algorithms and on-policy algorithms. The off-policy algorithms, such as the deep deterministic policy gradient algorithm [39] and the twin-delayed deep deterministic policy gradient algorithm [40], employ the experience replay buffer technique and have the advantages of high efficiency and fast convergence. However, sudden decreases in episode reward are often observed during the training, which makes the off-policy algorithms not stable enough. The on-policy algorithms, such as the proximal policy optimization (PPO) algorithm [41], mainly rely on the current training data to update the network, which leads to a low training efficiency. Nevertheless, it has the advantage that the episode reward is stable and monotonically increasing, and higher reward can be obtained. Therefore, the RL training for humanoid robot locomotion control is carried out using the PPO algorithm in this study.

The PPO algorithm is an actor-critic type algorithm that consists of two deep neural networks as shown in Fig. 7: the policy network (also known as actor network), which receives state observations and outputs actions, and the value network (also known as critic network), which is used to estimate the state value function. The policy network and the value network adopt the same network architecture, that is, a fully connected deep neural network based on the PyTorch framework. The network architecture contains two hidden layers, each with 256 units, and the rectified linear unit is chosen as the activation function.

All training in this paper is performed on a desktop workstation with a Ubuntu 20.04 operating system, an Intel Core i9-13900K CPU, 128 GB RAM, and an NVIDIA RTX 4090 GPU. The PPO algorithm is implemented based on the stable-baselines3 1.7.0 package [42], and the hyperparameters are listed in Table 3, where the number of parallel environments

Table 4 Initial values and limits of right knee and ankle joint angles [rad]

Case	Knee joint		Ankle joint			
	$\Delta\beta$	$\Delta\alpha$	$\Delta\theta$	$\Delta\phi$	$\Delta\alpha$	$\Delta\alpha'$
Initial value	0.8727	1.1624	−0.4363	0	−0.6561	−0.6935
Lower limit	0	0	−1.0472	−0.1745	−1.6960	−1.6301
Upper limit	1.3963	2.0346	0.1745	0.1745	0.5950	0.4801

depends on the number of total CPU threads. The other hyperparameters not mentioned here adopt the default values provided by the algorithm. The number of time steps for RL training is 10 million, and the time cost using the above hyperparameters is about 1 hour.

5 Results and discussions

In this section, we first give the inverse kinematics results for parallel mechanisms, which is related to the action space of RL. To verify the effectiveness of the proposed RL-based locomotion control strategy, several training cases (each with a separate RL agent) are then presented to realize the goals including foot periodic cycle with a frequency of 2 Hz, gait symmetry, forward speed up to 10 km/h, whole-body coordinated gait, and time-varying velocity command tracking.

5.1 Inverse kinematics results for parallel mechanisms

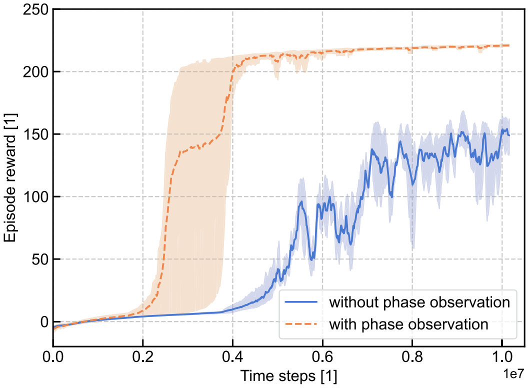
As key parameters of the model, the initial values and limits of the motor joint angles should be first determined before locomotion control, especially the limits that are related to the action space of RL. For four-bar mechanisms of the knee and ankle joints, these values are calculated using the inverse kinematics proposed in Sect. 2.3. Without loss of generality, Table 4 gives the results for the right knee and ankle joints, where all angles are evaluated as nominal values relative to a configuration of all joint angles as zero. The initial values are non-zero due to an initial knee forward configuration shown in Fig. 2.

5.2 Symmetric gait at a speed of 6 km/h

We adopt the well-designed reward function in Table 2 to train the humanoid robot to learn a symmetric gait at 6 km/h (i.e., 1.67 m/s) without a priori defined reference trajectories. Figure 11 compares the training results with and without clock phases in the observation space, where each episode reward shown here is plotted based on the results obtained from three experiments of training. Obviously, the introduction of clock phase observation can significantly improve the training convergence. The corresponding training curve increases monotonically with smaller oscillations, and higher episode rewards can be obtained. This is why the clock phases are included by default in the observation space as shown in Table 1. The training results for other agents in this study are presented in Appendix A.

Figure 12 presents the configurations of the humanoid robot during one gait cycle, where the time instants from left to right are 11.0 s, 11.125 s, 11.25 s, 11. 375 s and 11.5 s, respectively. Attributed to the orientation deviation penalty term in the reward function, the humanoid robot walks in an upright posture, where the pelvis roll, pitch, and yaw angles are all less than 2 degrees during stable locomotion. The gait is also accompanied by a toe-off

Fig. 11 Comparison of training results at 6 km/h with and without clock phase observation



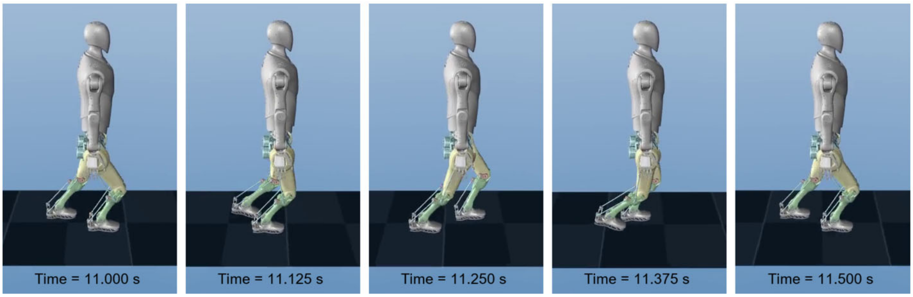


Fig. 12 Configurations of the humanoid robot during one gait cycle at 6 km/h

feature, and no collisions between legs are observed during the locomotion. In addition, it is found that at instants of integer multiples of the period T, the left knee joint is extended and the left foot just switches from the stance phase to the swing phase. At the same time, the right knee joint is flexed and the right foot switches from the swing phase to the stance phase. This is consistent with evolution of the weight coefficients shown in Fig. 8.

The velocity tracking function in Table 2 ensures that when the actual velocity $v_x = 0$, the unweighted reward is almost zero, and when $v_x = 6$ km/h, the reward reaches its maximum value of 1. Figure 13(a) shows the velocity tracking result of the humanoid robot directly accelerating from a stationary state to 6 km/h, where the velocity v_x oscillates around the command with a small amplitude after the initial acceleration process. The normal contact forces in the foot–ground interactions across six gait cycles are depicted in Fig. 13(b), where the force is normalized with respect to the total weight according to Eq. (9). It can be seen that the foot periodic cycle penalty successfully realizes the alternation of stance and swing phases for both feet.

The time evolution of the motor joint angles is illustrated by Fig. 14. It is observed that the joint angles show good periodicity, and the cycle frequencies of both feet are consistent with the prescribed frequency of 2 Hz. More importantly, the symmetric gait is achieved. The evolution of the corresponding joint angles on the left and right is almost the same except for a phase shift of $T/2$, which indicates the proposed method based on the reward function design to encourage gait symmetry is effective here. Furthermore, the hip yaw angles imply a toe-out gait [43], and the toe-out angle is approximately 2 to 3 degrees. This can increase the support polygon area [1] and is similar to that of human beings. The hip

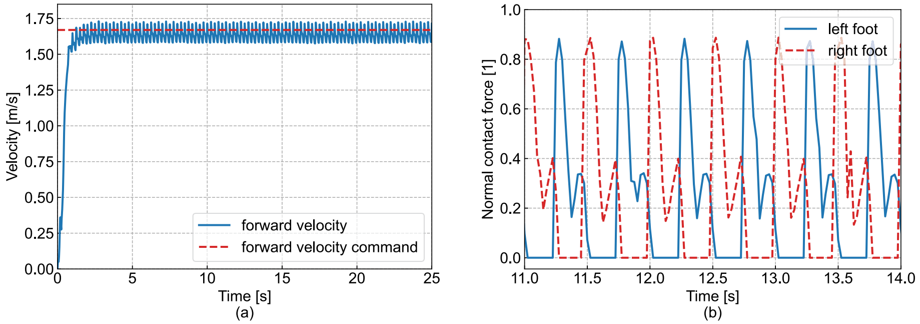


Fig. 13 Results of symmetric gait at 6 km/h. (a) Forward velocity tracking result, (b) normal contact forces in the foot–ground interactions

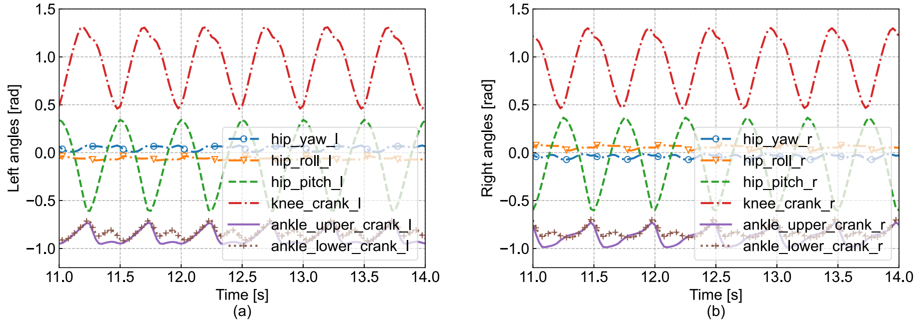


Fig. 14 Evolution of (a) left motor joint angles and (b) right motor joint angles at 6 km/h

roll motions have a characteristics of adduction rather than abduction, which ensures that the ground projection of the center of mass is located within the support polygon during the single support phase.

5.3 Symmetric gait at a speed of 10 km/h

It is difficult for the humanoid robot to accelerate directly from standstill to 10 km/h (i.e., 2.78 m/s), especially when the original velocity tracking function in Table 2 is employed. The value of the original function is close to zero when the velocity v_x is less than 1 m/s as shown in Fig. 15(a), which causes the reward function to encourage the humanoid robot to step in place rather than move forward. This can be verified by the velocity tracking result in Fig. 15(b). To address this problem, three new velocity tracking functions are chosen here as alternatives, including a coefficient-modified version of the original function, a minimum value function, and a modified absolute value function. The values of these alternative functions are all greater than zero when the velocity $v_x \in (0, 1)$ m/s, and increase with the velocity until v_x reaches 2.78 m/s. These three functions encourage the humanoid to move forward to maximize the cumulative reward, and Fig. 15(b) compares their velocity tracking results. It can be seen that the effect of the modified absolute value function is good enough and better than the others, and thus it is adopted in this subsection.

Figure 16 shows the configurations of the humanoid robot during one gait cycle at a speed of 10 km/h, where the human-like walking style with heel-strike and toe-off motion [44] is

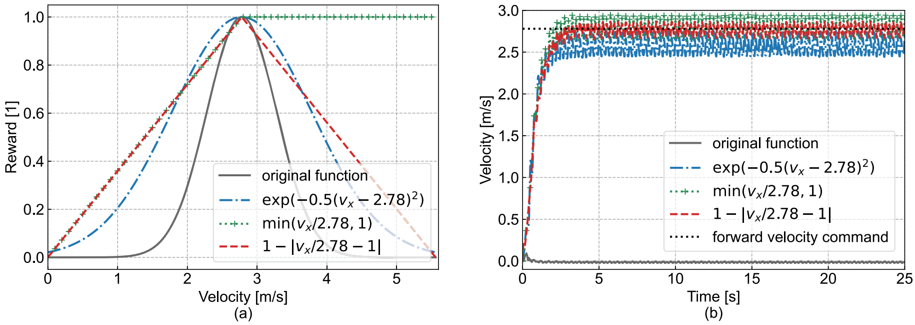


Fig. 15 Forward velocity tracking at 10 km/h. (a) Tracking functions, (b) tracking results

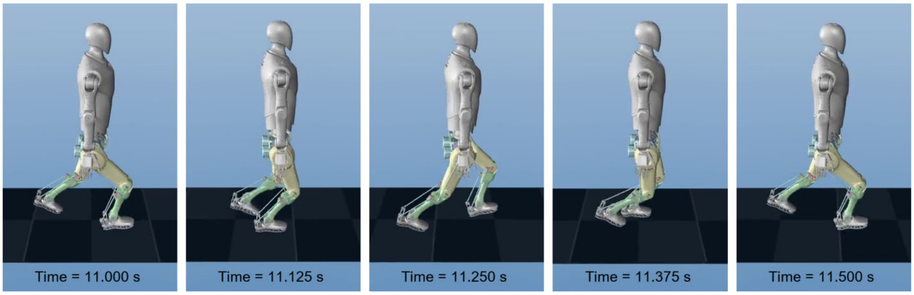


Fig. 16 Configurations of the humanoid robot during one gait cycle at 10 km/h

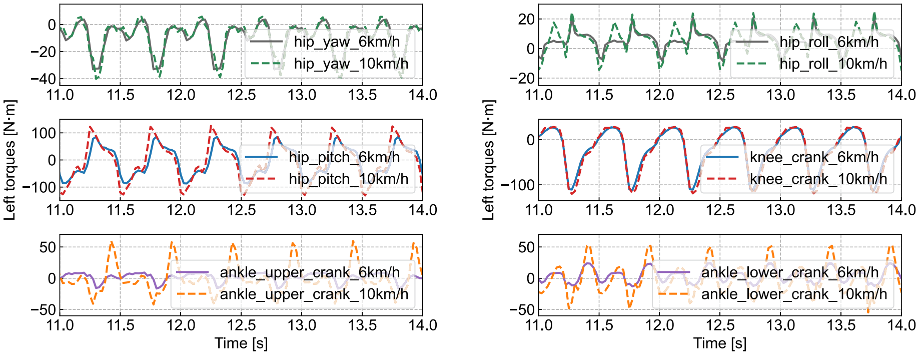


Fig. 17 Comparison of left motor joint torques at 6 km/h and 10 km/h

very obvious. Compared with Fig. 12, it can be seen that since the same stride frequency is employed by all training cases, faster locomotion speed means larger stride length, where the stride length is approximately 1.389 m at 10 km/h and 0.833 m at 6 km/h. Furthermore, the aerial phase where both feet are off the ground is also observed during the locomotion.

The evolution of left motor joint torques is illustrated by Fig. 17, where all motor joint torques are within the limit ranges. Compared with the results in the case of 6 km/h, it is found that faster locomotion speed is mainly achieved through greater hip and ankle motor

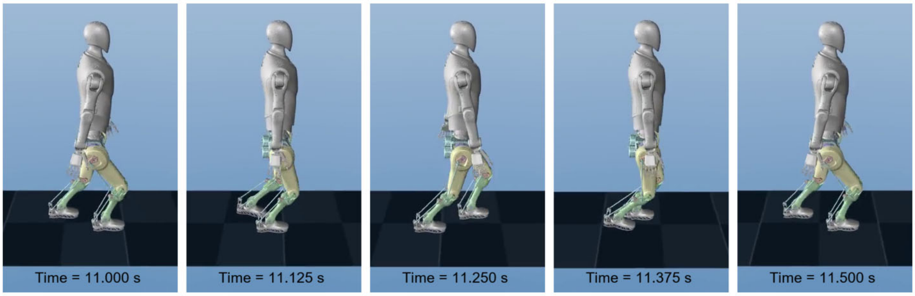


Fig. 18 Whole-body coordinated configurations of humanoid robot during one gait cycle at 6 km/h

joint torques rather than greater knee motor joint torques. Faster speed requires a larger stride length, and the stride length is partially obtained by raising the thigh to a certain height, which needs greater hip motor joint torques, especially the hip pitch torque. The heel-strike and toe-off gait indirectly increases the lengths of lower limbs, which also increases the stride length. In addition, the toe-off motion increases the horizontal component of the ground reaction force, thereby increasing the forward acceleration. However, the heel-strike and toe-off motion will result in a line contact scenario between the supporting foot and the ground, which requires greater ankle motor joint torques to maintain balance during the locomotion.

5.4 Whole-body coordinated gaits

Using the reference trajectory defined in Sect. 3.4.4 for shoulder pitch angles, a whole-body coordinated gait at a speed of 6 km/h can be obtained through RL training. As shown in Fig. 18, when the left foot and the right hand are at the rearmost sides, the right foot and the left hand are at the foremost sides, and vice versa.

Compared with the symmetric gait in Sect. 5.2, the whole-body coordinated gait is more human-like. It is worth mentioning that the time evolution of the left and right shoulder pitch angles is also symmetric and shows good periodicity, and the corresponding joint torques are within the limit ranges. Another benefit of the whole-body coordination is that the angular momentum of shoulder pitching motions can compensate for that of hip pitching motions, thereby reducing the pelvis yaw angle. As illustrated by Fig. 19(a), this compensation effect is not significant in the case of 6 km/h, because the reward function includes an orientation deviation penalty term, which results in a small enough pelvis yaw angle (less than 2 degrees) even without such compensation. However, in the case of 10 km/h, the pelvis yaw motion is obvious due to the large stride length, and the influence of this compensation is noticeable as shown in Fig. 19(b). Furthermore, the whole-body coordination has an indirect effect on reducing the hip yaw angles, since the hip yaw motion is related to the pelvis yaw motion and there is no corresponding penalty term in the reward function.

The velocity tracking results of directly accelerating from standstill to 6 km/h and 10 km/h are good enough in the case of whole-body coordinated gaits, where the velocity tracking functions are the same as those in Sect. 5.2 and Sect. 5.3, respectively. The whole-body coordinated gait at a speed of 10 km/h is shown in Fig. 20. Similar to that in Sect. 5.3, faster locomotion speed implies larger stride length, which leads to a faster foot velocity of 6.5 m/s compared to 4.5 m/s in the case of 6 km/h. Since the whole-body coordination penalty is a soft constraint, the amplitudes of shoulder pitch angles for 10 km/h are a bit larger than those for 6 km/h, so are the corresponding shoulder pitch torques.

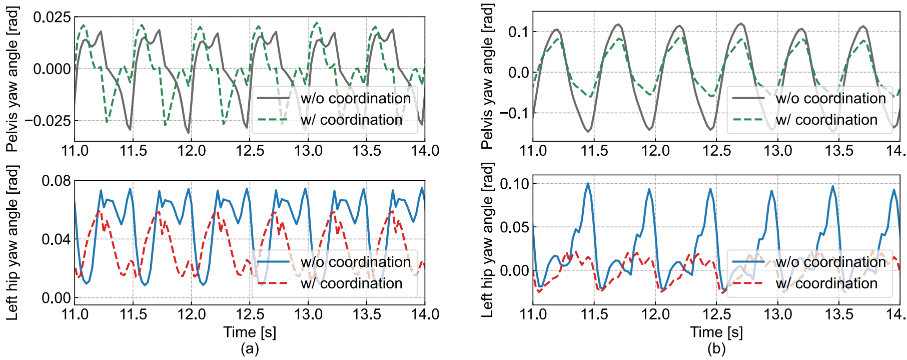


Fig. 19 Comparison of pelvis yaw and hip yaw angles without and with whole-body coordination (a) at 6 km/h and (b) at 10 km/h

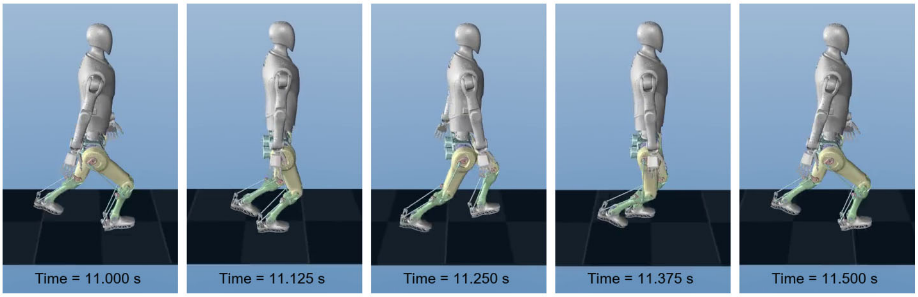


Fig. 20 Whole-body coordinated configurations of humanoid robot during one gait cycle at 10 km/h

5.5 Time-varying velocity command tracking

We investigate four cases of time-varying velocity commands for the humanoid robot with whole-body coordinated gait, including (a) 6/8/10 km/h, (b) 0/2/4/6 km/h, (c) 0/2/4/2/0 km/h, and (d) 0/-2/-4/-2/0 km/h. Here, 0 km/h velocity command represents stepping in place rather than standing still, and the negative velocity command means walking backwards.

The velocity tracking function in Table 2 is employed for RL training, and Fig. 21 illustrates the velocity tracking results of the above four cases, which are good enough for both forward and backward locomotion. It is worth mentioning that when the height penalty term is considered in the swing phase penalty, the problem of the off-ground heights of both feet being too low when the velocity command $v_x^{\mathrm{cmd}} < 6$ km/h is successfully solved.

Among the above four cases, the last one with backward locomotion is a bit special. Figure 22 demonstrates the configurations of the humanoid robot during one gait cycle at a speed of −4 km/h. As can be seen, one distinct feature of backward locomotion is that the thighs keep in front of the torso and the hip pitch angles are always negative. Another obvious feature is that the knee joints are flexed at all times.

Furthermore, we compare the results obtained from the third and last cases, including the evolution of hip, knee, and ankle pitch angles and normal force in forward and backward whole-body coordinated gaits, as shown in Fig. 23. The forward locomotion relies mainly on

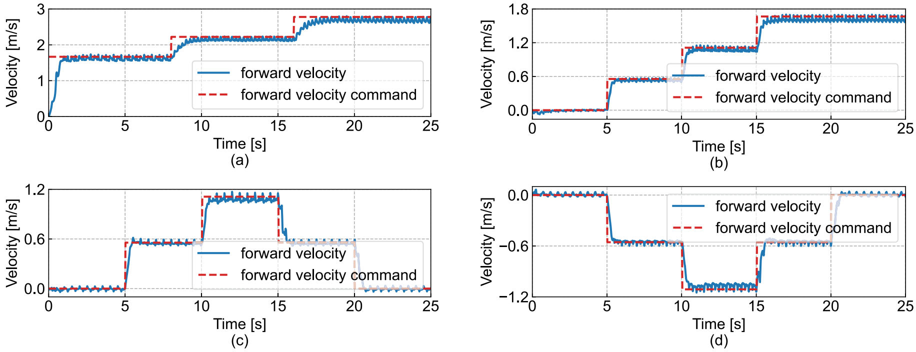


Fig. 21 Time-varying velocity tracking results for whole-body coordinated gaits at (a) 6/8/10 km/h, (b) 0/2/4/6 km/h, (c) 0/2/4/2/0 km/h, (d) 0/-2/-4/-2/0 km/h

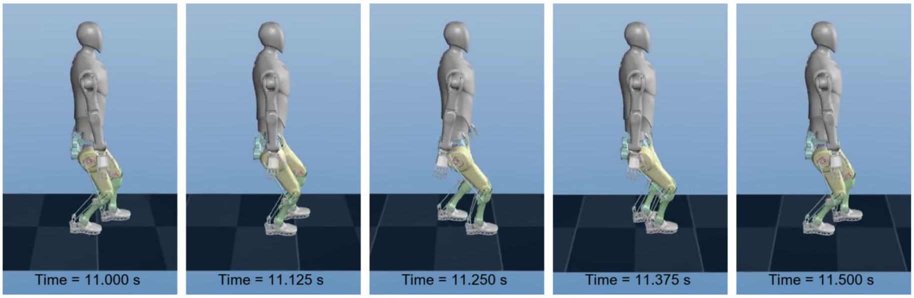


Fig. 22 Whole-body coordinated configurations of humanoid robot during one gait cycle at -4 km/h

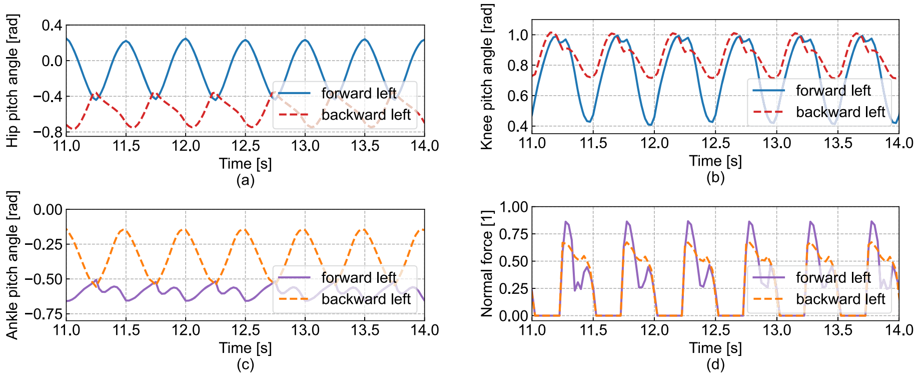


Fig. 23 Comparison of (a) hip pitch angle, (b) knee pitch angle, (c) ankle pitch angle, and (d) normal force in forward (4 km/h) and backward (−4 km/h) whole-body coordinated gaits

the hip pitching and knee pitching motions, and for both hip pitch and knee pitch angles an angular range of about 0.6 rad is found at a speed of 4 km/h. While in the case of backward locomotion at a speed of −4 km/h, only a range of 0.4 rad is observed for the hip pitch angle, and the range of knee pitch angle is just 0.3 rad. In comparison, the ankle pitching motions play an important role in walking backwards. As can be seen from Fig. 23(c), the range of ankle pitch angle at −4 km/h is about three times that when walking forward at 4 km/h. In addition, the peak values of the normal force during backward locomotion are smaller than those during forward locomotion, which is attributed to the lower off-ground heights and smaller vertical velocities of the feet when walking backwards.

From another point of view, the backward locomotion can be treated as a forward locomotion with the knees pointing backwards, like the gait of an ostrich. Ref. [45] presents a similar case study on a planar biped robot with circular feet, where the conclusion for the backward gait is a bit different from this paper. This is due to the absence of ankle pitch DoF in that case, so the push-off velocity for the robot can only be generated from the extension of the knee joint.

6 Conclusions

In this paper, the modeling and RL-based locomotion control for a home-made humanoid robot with kinematic loop closures are investigated. The rigid multibody model is first built for the humanoid robot. The inverse kinematics solutions based on geometric relationships are then presented for the parallel mechanisms, and the contact detection procedures for foot–ground interactions and collisions between legs are simplified by introducing spherical geometries and bounding boxes, respectively.

Based on the modeling work, a deep RL environment for locomotion control is established, including the fundamental framework, observation space, action space, reward function, and termination conditions. The reward function is well designed to achieve the baseline goal of forward velocity tracking, as well as the auxiliary abilities such as the foot periodic cycle, the gait symmetry, and the whole-body coordination. Here, the periodic alternation of stance and swing phases is realized based on the complementary conditions of foot velocity and foot–ground interaction force, and a new method is proposed to encourage the symmetric gait, which is implemented by penalizing the differences in the mean values and standard deviations calculated over a moving period between left and right joint angles.

To verify the effectiveness of the proposed RL-based locomotion control strategy, several training cases (each with a separate RL agent) are carried out using the PPO algorithm, where the key parameters like the initial values and joint limits of the parallel mechanisms are determined by the inverse kinematics. The influence of clock phase observation is first investigated, and the result indicates that it should be included in the observation space to improve the training convergence. We then train the humanoid robot to learn a stable gait at 6 km/h, where the forward velocity tracking performs well and the goals of foot periodic cycle and gait symmetry are achieved. In addition, the hip yaw motions imply a toe-out gait, which is similar to that of human beings. The original velocity tracking function fails in the case of directly accelerating from standstill to 10 km/h, so an alternative modified absolute value function is proposed, and it proves to be better than the others. The human-like walking style with heel-strike and toe-off motion is observed during the locomotion, and it is found that faster locomotion speed means larger stride length and requires greater hip and ankle motor joint torques. After that, the whole-body coordinated gait is obtained by releasing the shoulder pitch DoFs and tracking a pair of manually designed reference trajectories. Compared with the aforementioned results, the whole-body coordinated gait is more human-like, and the angular momentum of shoulder pitching motions can compensate for that of hip pitching motions, thereby reducing the pelvis yaw and the hip yaw angles. Finally, four cases of whole-body coordinated gaits with time-varying velocity commands are studied, and the velocity tracking results are good enough for both forward and backward locomotion. Furthermore, it is found that the forward locomotion relies mainly on the hip pitching and knee pitching motions, whereas in the case of walking backwards, the ankle pitching motions play a dominant role. The backward locomotion can be regarded as a forward locomotion with the knees pointing backwards, which can be used to inspire the design of new humanoid robot structures.

The RL-based locomotion control strategy proposed here can be extended to the case of omni-directional walking, where lateral velocity and yaw angular velocity commands are additionally introduced in the observation space, and the gait symmetry and whole-body coordination terms in the reward function need to be slightly modified. Future work should first focus on the refinement of the humanoid robot model, including an improved multi-body model with waist yaw DoF, an actuator model considering the torque-rotational speed profile of the motor and the influence of the gear ratio, and the uneven terrain environments such as slopes, stairs, and deformable sand or soil terrain. The generalization ability and robustness of the agent will be the next focus, which can be improved by introducing various randomizations, such as observation noise, initial state randomization, random external disturbances, and dynamic domain randomization. In addition, a unified reward function that applies to all cases and a single RL agent that works for various velocity commands will be investigated in the future. The RL training framework will also be revised. The recurrent policy like the long short-term memory neural network will be introduced, and the curriculum learning or imitation learning frameworks will be considered to explore more energy-efficient and agile locomotion gaits for the humanoid robot. Finally, we will focus on the sim-to-real transfer. Some observations and reward components should be replaced by quantities that can be obtained directly from sensors or by simple state estimations. Besides, there are some disadvantages of RL-based approaches, such as lack of interpretability, difficulty in pinpointing issues, high computational cost, and neural network overfitting. These drawbacks affect the transfer and deployment of RL agents from simulation to real-world humanoid robot prototypes. Future work should address these drawbacks.

Appendix A: Training results for agents

The training results for agents in this study are shown below. Here, each episode reward curve is plotted based on the results obtained from three experiments of training, and all training results have reached convergence.

It should be noted that in the cases of gaits at 10 km/h, the modified absolute value function is chosen as the velocity tracking function. Due to this, the final episode rewards in the cases of 10 km/h are larger than those in the cases of 6 km/h.

In the four cases of whole-body coordinated gaits with time-varying velocity commands, the training results converge quickly when the velocity command starts from 0 km/h, that is, the locomotion of the humanoid robot begins with stepping in place. When the velocity command starts from 6 km/h and increases to 8 km/h and then to 10 km/h, the training result converges slowly since this task is a bit challenging. In order to verify the convergence, the time step in this case is extended to 20 million.

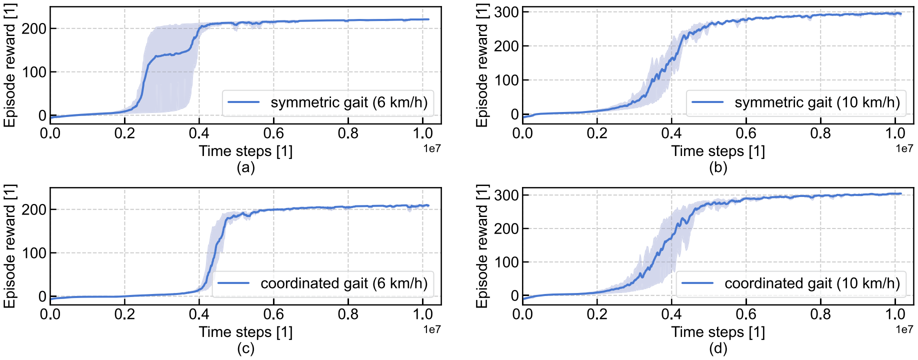


Fig. 24 Training results for agents in the cases of (a) symmetric gait at 6 km/h, (b) symmetric gait at 10 km/h, (c) whole-body coordinated gait at 6 km/h, (d) whole-body coordinated gait at 10 km/h

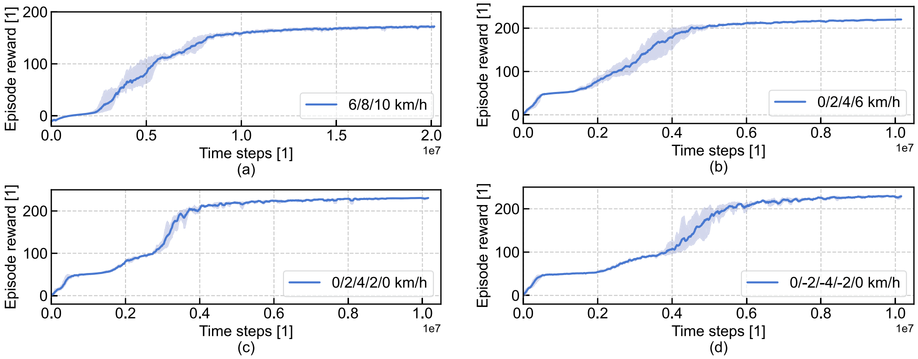


Fig. 25 Training results for agents in the cases of whole-body coordinated gaits at (a) 6/8/10 km/h, (b) 0/2/4/6 km/h, (c) 0/2/4/2/0 km/h, (d) 0/-2/-4/-2/0 km/h

Author contributions Lingling Tang: formal analysis, investigation, methodology, software, visualization and writing the original draft. Dingkun Liang: conceptualization, funding acquisition, project administration and visualization. Guang Gao: methodology and software. Xin Wang: conceptualization, funding acquisition, project administration and review the original draft. Anhuan Xie: conceptualization, funding acquisition and project administration. All authors read and approved the final manuscript.

Funding This research was supported by the Key Research Project of Zhejiang Lab (No. G2021NB0AL03), Zhejiang Provincial Natural Science Foundation of China (No. LQ23F030010), and the High Quality Development Project of the Ministry of Industry and Information Technology of China (No. 2340STCZB1930), for which the authors are grateful.

Data Availability No datasets were generated or analysed during the current study.

Declarations

Competing interests The authors declare no competing interests.

References

1. Kajita, S., Hirukawa, H., Harada, K., Yokoi, K.: Introduction to Humanoid Robotics. Springer, Berlin (2014)
2. Goswami, A., Vadakkepat, P.: Humanoid Robotics: A Reference. Springer, Dordrecht (2019)
3. Géradin, M., Cardona, A.: Flexible Multibody Dynamics: A Finite Element Approach. Wiley, Chichester (2001)
4. Abate, A.M.: Mechanical design for robot locomotion. PhD thesis, Oregon State University, Corvallis (2018)
5. Liu, Y., Shen, J., Zhang, J., Zhang, X., Zhu, T., Hong, D.: Design and control of a miniature bipedal robot with proprioceptive actuation for dynamic behaviors. In: International Conference on Robotics and Automation (ICRA), Philadelphia, pp. 8547–8553 (2022). https://doi.org/10.1109/ICRA46639.2022.9811790
6. Pfeiffer, F.: Mechanical System Dynamics. Springer, Berlin (2008)
7. Buschmann, T.: Simulation and control of biped walking robots. PhD thesis, Technische Universität München, München (2010)
8. Hu, Y., Wu, X., Ding, H., Li, K., Li, J., Pang, J.: Study of series-parallel mechanism used in legs of biped robot. In: International Conference on Control, Automation and Robotics (ICCAR), Singapore, pp. 97–102 (2021). https://doi.org/10.1109/ICCAR52225.2021.9463499
9. Konyukhov, A., Schweizerhof, K.: Computational Contact Mechanics: Geometrically Exact Theory for Arbitrary Shaped Bodies. Springer, Berlin (2013)
10. Schwienbacher, M.: Efficient algorithms for biped robots - simulation, collision avoidance and angular momentum tracking. PhD thesis, Technische Universität München, München (2014)
11. Ericson, C.: Real-Time Collision Detection. Morgan Kaufmann Publishers, Amsterdam (2005)
12. Tasora, A., Anitescu, M.: A fast NCP solver for large rigid-body problems with contacts, friction, and joints. In: Bottasso, C.L. (ed.) Multibody Dynamics: Computational Methods and Applications, pp. 45–55. Springer, Dordrecht (2009)
13. Todorov, E.: Convex and analytically-invertible dynamics with contacts and constraints: theory and implementation in MuJoCo. In: International Conference on Robotics and Automation (ICRA), Hong Kong, pp. 6054–6061 (2014). https://doi.org/10.1109/ICRA.2014.6907751
14. Hwangbo, J., Lee, J., Hutter, M.: Per-contact iteration method for solving contact dynamics. IEEE Robot. Autom. Lett. **3**(2), 895–902 (2018). https://doi.org/10.1109/LRA.2018.2792536
15. Kajita, S., Tani, K.: Study of dynamic biped locomotion on rugged terrain: Derivation and application of the linear inverted pendulum mode. In: International Conference on Robotics and Automation (ICRA), pp. 1405–1411 (1991). https://doi.org/10.1109/ROBOT.1991.131811
16. Ding, Y., Khazoom, C., Chignoli, M., Kim, S.: Orientation-aware model predictive control with footstep adaptation for dynamic humanoid walking. In: International Conference on Humanoid Robots, pp. 299–305 (2022). https://doi.org/10.1109/Humanoids53995.2022.10000244
17. Sutton, R.S., Barto, A.G.: Reinforcement Learning: An Introduction, 2nd edn. The MIT Press, Cambridge (2018)
18. Manzl, P., Rogov, O., Gerstmayr, J., Mikkola, A., Orzechowski, G.: Reliability evaluation of reinforcement learning methods for mechanical systems with increasing complexity. Multibody Syst. Dyn., 1–25 (2023). https://doi.org/10.1007/s11044-023-09960-2
19. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997). https://doi.org/10.1162/neco.1997.9.8.1735
20. Siekmann, J., Valluri, S., Dao, J., Bermillo, F., Duan, H., Fern, A., Hurst, J.: Learning memory-based control for human-scale bipedal locomotion. In: Robotics: Science and Systems, Corvalis, pp. 31:1–31:8 (2020). https://doi.org/10.15607/RSS.2020.XVI.031
21. Siekmann, J., Green, K., Warila, J., Fern, A., Hurst, J.: Blind bipedal stair traversal via sim-to-real reinforcement learning. In: Robotics: Science and Systems, Virtual, pp. 61:1–61:9 (2021). https://doi.org/10.15607/RSS.2021.XVII.061
22. Yu, W., Turk, G., Liu, C.K.: Learning symmetric and low-energy locomotion. ACM Trans. Graph. **37**(4), 144:1–144:12 (2018). https://doi.org/10.1145/3197517.3201397
23. Lee, J., Hwangbo, J., Wellhausen, L., Koltun, V., Hutter, M.: Learning quadrupedal locomotion over challenging terrain. Sci. Robot. **5**(47), 5986 (2020). https://doi.org/10.1126/scirobotics.abc5986
24. Xie, Z., Ling, H.Y., Kim, N.H., van de Panne, M.: ALLSTEPS: curriculum-driven learning of stepping stone skills. Comput. Graph. Forum **39**(8), 213–224 (2020). https://doi.org/10.1111/cgf.14115
25. Rudin, N., Hoeller, D., Reist, P., Hutter, M.: Learning to walk in minutes using massively parallel deep reinforcement learning. In: Conference on Robot Learning (CoRL), London, pp. 91–100 (2021). https://doi.org/10.48550/arXiv.2109.11978

26. Peng, X.B., Abbeel, P., Levine, S., van de Panne, M.: DeepMimic: example-guided deep reinforcement learning of physics-based character skills. ACM Trans. Graph. **37**(4), 143:1–143:14 (2018). https://doi.org/10.1145/3197517.3201311
27. Peng, X.B., Ma, Z., Abbeel, P., Levine, S., Kanazawa, A.: AMP: adversarial motion priors for stylized physics-based character control. ACM Trans. Graph. **40**(4), 144:1–144:20 (2021). https://doi.org/10.1145/3450626.3459670
28. Peng, X.B., Guo, Y., Halper, L., Levine, S., Fidler, S.: ASE: large-scale reusable adversarial skill embeddings for physically simulated characters. ACM Trans. Graph. **41**(4), 94:1–94:17 (2022). https://doi.org/10.1145/3528223.3530110
29. Todorov, E., Erez, T., Tassa, Y.: MuJoCo: A physics engine for model-based control. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Algarve, pp. 5026–5033 (2012). https://doi.org/10.1109/IROS.2012.6386109
30. Featherstone, R.: Rigid Body Dynamics Algorithms. Springer, New York (2008)
31. Baumgarte, J.: Stabilization of constraints and integrals of motion in dynamical systems. Comput. Methods Appl. Mech. Eng. **1**(1), 1–16 (1972). https://doi.org/10.1016/0045-7825(72)90018-7
32. Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., Zaremba, W.: OpenAI Gym, pp. 1–4 (2016). https://doi.org/10.48550/arXiv.1606.01540
33. Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., Tassa, Y., Erez, T., Wang, Z., Eslami, S.M.A., Riedmiller, M., Silver, D.: Emergence of locomotion behaviours in rich environments, pp. 1–14 (2017). https://doi.org/10.48550/arXiv.1707.02286
34. Peng, X.B., van de Panne, M.: Learning locomotion skills using DeepRL: does the choice of action space matter? In: Proceedings of the ACM SIGGRAPH / Eurographics Symposium on Computer Animation, Los Angeles, pp. 1–13 (2017). https://doi.org/10.1145/3099564.3099567
35. Siekmann, J., Godse, Y., Fern, A., Hurst, J.: Sim-to-real learning of all common bipedal gaits via periodic reward composition. In: International Conference on Robotics and Automation (ICRA), Xi'an, pp. 7309–7315 (2021). https://doi.org/10.1109/ICRA48506.2021.9561814
36. Dao, J.: Practical reinforcement learning for bipedal locomotion. Master's thesis, Oregon State University, Corvallis (2021)
37. Dao, J., Green, K., Duan, H., Fern, A., Hurst, J.: Sim-to-real learning for bipedal locomotion under unsensed dynamic loads. In: International Conference on Robotics and Automation (ICRA), Philadelphia, pp. 10449–10455 (2022). https://doi.org/10.1109/ICRA46639.2022.9811783
38. Abdolhosseini, F., Ling, H.Y., Xie, Z., Peng, X.B., van de Panne, M.: On learning symmetric locomotion. In: 12th ACM SIGGRAPH Conference on Motion, Interaction and Games, Newcastle, pp. 19:1–19:10 (2019). https://doi.org/10.1145/3359566.3360070
39. Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., Wierstra, D.: Continuous control with deep reinforcement learning. In: International Conference on Learning Representations (ICLR), San Juan, Puerto Rico, pp. 1–10 (2016). https://doi.org/10.48550/arXiv.1509.02971
40. Fujimoto, S., Hoof, H., Meger, D.: Addressing function approximation error in actor-critic methods. In: International Conference on Machine Learning (ICML), Stockholm, pp. 1587–1596 (2018). https://doi.org/10.48550/arXiv.1802.09477
41. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms, pp. 1–12 (2017). https://doi.org/10.48550/arXiv.1707.06347
42. Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., Dormann, N.: Stable-baselines3: reliable reinforcement learning implementations. J. Mach. Learn. Res. **22**(268), 1–8 (2021)
43. Whittle, M.W.: Gait Analysis: An Introduction, 4th edn. Butterworth-Heinemann, Oxford (2007)
44. Ogura, Y., Shimomura, K., Kondo, H., Morishima, A., Okubo, T., Momoki, S., Lim, H.-o., Takanishi, A.: Human-like walking with knee stretched, heel-contact and toe-off motion by a humanoid robot. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, pp. 3976–3981 (2006). https://doi.org/10.1109/IROS.2006.281834
45. Smit-Anseeuw, N., Gleason, R., Vasudevan, R., Remy, C.D.: The energetic benefit of robotic gait selection: A case study on the robot RAMone. IEEE Robot. Autom. Lett. **2**(2), 1124–1131 (2017). https://doi.org/10.1109/LRA.2017.2661801

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.