# Deep Deterministic Policy Gradient with Constraints for Gait Optimisation of Biped Robots

/

Xingyang Liu<sup>a</sup>, Haina Rong<sup>1 a</sup>, Ferrante Neri<sup>b,e,1</sup>, Peng Yue<sup>c</sup>, Gexiang Zhang<sup>d</sup>

<sup>a</sup> School of Electrical Engineering, Southwest Jiaotong University, Chengdu, 610031, China

<sup>b</sup>Nature Inspired Computing and Engineering Research Group, School of Computer Science and Electronic Engineering, University of Surrey, Guildford, Surrey GU2 7XH, United Kingdom

<sup>c</sup> State Key Laboratory of Synthetical Automation for Process Industries, Northeastern University, Shenyang 110819, China

<sup>d</sup> School of Control Engineering, Chengdu University of Information Technology, 610225 Chengdu, China

<sup>e</sup> School of Software, Nanjing University of Information Science and Technology, Jiangsu, China.

Abstract. In this paper, we propose a novel Reinforcement Learning (RL) algorithm for robotic motion control, that is, a constrained Deep Deterministic Policy Gradient (DDPG) deviation learning strategy to assist biped robots in walking safely and accurately. The previous research on this topic highlighted the limitations in the controller's ability to accurately track foot placement on discrete terrains and the lack of consideration for safety concerns. In this study, we address these challenges by focusing on ensuring the overall system's safety. To begin with, we tackle the inverse kinematics problem by introducing constraints to the damping least squares method. This enhancement not only addresses singularity issues but also guarantees safe ranges for joint angles, thus ensuring the stability and reliability of the system. Based on this, we propose the adoption of the constrained DDPG method to correct controller deviations. In constrained DDPG, we incorporate a constraint layer into the Actor network, incorporating joint deviations as state inputs. By conducting offline training within the range of safe angles, it serves as a deviation corrector. Lastly, we validate the effectiveness of our proposed approach by conducting dynamic simulations using the CRANE biped robot. Through comprehensive assessments, including singularity analysis, constraint effectiveness evaluation, and walking experiments on discrete terrains, we demonstrate the superiority and practicality of our approach in enhancing walking performance while ensuring safety. Overall, our research contributes to the advancement of biped robot locomotion by addressing gait optimisation from multiple perspectives, including singularity handling, safety constraints, and deviation learning.

Keywords. constraints' handling, Deep Deterministic Policy Gradient, deviation learning, Reinforcement Learning, gait optimisation

# 1. Introduction

Biped robots, with their high flexibility and mobility, have vast potential for real-world applications. For example, Atlas excels in challenging terrains, making it suitable for search and rescue missions [1]. Asimo serves as a household assistant and can also showcase dance performances and other acts [2]. HRP-4 is suitable for various industrial applications [3]. However, in the control of biped robot motion, ensuring the safety of system states poses a significant challenge, thus making the control difficult [4,5,6,7]. Furthermore, gait deviations may result from factors such as inaccurate modeling and unstable environments, leading to unexpected situations during robot motion and posing safety risks for both the robot and its surroundings [8,9,10,11,12]. Therefore, precise gait control and correction are vital to ensuring the accuracy and stability of a robot's motion trajectory and execution results, and are crucial for efficient robot operation and human-robot safety.

The locomotion of a robot can be divided into two key components: planning and execution [13,14,15,

<sup>&</sup>lt;sup>1</sup>Corresponding Author: Haina Rong, Email: ronghaina@126.com and Ferrante Neri, Email: f.neri@surrey.ac.uk

16]. The planning phase addresses the mapping problem from the desired pose or position of the robot's end effector to the corresponding joint angles through inverse kinematics [17,18,19,20,21]. The solution of inverse kinematics plays a crucial role in achieving accurate motion. In the execution phase, the controller takes centre stage as it determines whether the control objectives can be effectively realized. In the case of biped robot gait control, ensuring both accuracy and safety is of utmost importance. Therefore, incorporating joint angle constraints in both the planning and execution phases is vital for effective control of biped robots [22,23,24,25]. These constraints serve as crucial guidelines by limiting the range of joint angles that the robot can attain. In the control of biped robots, joint angles play a fundamental role as basic parameters governing the robot's motion. By applying constraints to the joint angles, we can effectively prevent the robot from exceeding its range of motion, thereby enhancing both the stability and safety of its movements.

In our recent research, we have made significant progress in enabling the dynamic walking of robots through two essential steps [26]. Firstly, we implemented an interpolation-based method to plan the trajectory of footstep locations. By employing a direct approach to solve the inverse kinematics problem, we successfully obtained the desired joint trajectory. Secondly, to ensure smooth and stable robot motion, we incorporated a human-simulated fuzzy (HF) controller to accurately track the target joint trajectory. By implementing this controller, we achieve precise control over the robot's motion, ensuring balance and stability during walking actions. This substantial improvement enhances the overall quality of robot locomotion. In this research, it is crucial to consider several essential factors for safe operation, such as singular value issues and angle constraints, in order to prevent unforeseen and unpredictable situations. Additionally, when dealing with high-precision control over uneven terrains, we face challenges in parameter tuning. Addressing these issues holds great significance for the practical application of robots.

To meet the high precision requirements of discrete terrains, correcting deviations in the controller is an effective and viable approach. Available options include feedforward control, model predictive control (MPC), and reinforcement learning (RL) [27,28,29], combination control [30,31,32,33,34], among others [35,36,37,38,39,40]. While feedforward control and MPC rely on accurate modeling and complex computational techniques, RL does not require precise system modeling. RL is a field of machine learning [41,42,43,44,45,46,47,48] that involves interacting with an environment and adjusting control policies based on reward signals to achieve optimization objectives.

RL achieves optimisation objectives by interacting with the environment and adjusting control strategies based on reward signals [49,50,51]. It excels in learning complex control tasks without prior knowledge and exhibits the ability to adapt to changing environments. Consequently, it has emerged as a powerful technique widely applied in the field of robotics. Its applications encompass gait training [52,53], motion strategy learning [54], trajectory optimisation [55], automatic residual learning [56,57], and optimisation control [58], among others. Through the use of reinforcement learning, robots can continuously learn and improve their control strategies and behaviors by actively interacting with the environment.

In the field of RL, DDPG (Deep Deterministic Policy Gradient) combines the advantages of deep learning [59,60,61,62] and policy gradient methods to address reinforcement learning problems in continuous action spaces, making it highly favored by researchers [63,64,65,66,67]. Tao et al. focused on utilizing parallel DDPG strategies for gait control. However, in their Markov Decision Process model, the primary aim was to address the issue of sparse rewards without explicitly defining safety-related states or reward signals [63]. Gao et al. proposed an improved DDPG algorithm that used a Long Short-Term Memory (LSTM) network encoder to achieve dynamic obstacle avoidance for mobile robots. Nevertheless, their approach did not explicitly consider the safety constraints of the robot's motion, such as rotational constraints, among other issues [66]. In the context of multi-robot navigation tasks in positional environments, Chang et al. designed a learning network based on DDPG to solve the global navigation task for a single robot. However, the use of DDPG in their work was primarily focused on global navigation tasks and did not explicitly address constraints related to the robot's own motion, such as rotation and velocity constraints [68]. In summary, while these methods each have their unique focuses and contributions, they did not directly or comprehensively consider safety constraint issues in the use of DDPG for motion control [69,70].

This paper fills these gaps in the field by addressing the most important aspects of safety and precision in gait optimisation and delving into comprehensive research in both the planning and execution phases. In the planning phase, we tackle inverse kinematics by incorporating constraints into the damped least squares method [71,72,73,19,74]. During the execution phase, we place special emphasis on rectifying deviations in the HF controller. To achieve this, we introduce a constrained DDPG deviation learner.

The key contributions of this article are as follows:

1) In the process of solving inverse kinematics, we introduce angle constraints based on the damped least squares method to avoid singular value issues and ensure safety constraints. This improvement aims to restrict the range of joint angles during the optimisation process, ensuring system stability and reliability.

2) A constrained DDPG model is designed, and an angle deviation learner is trained offline. The joint angle deviation is utilised as the input state for the Actor network, and a constraint layer is added to restrict the joint angles within a safe range and achieve a target deviation of zero.

3) By incorporating the angle deviation learner into the HF controller and conducting simulation experiments on discrete terrains, the safety and accuracy of walking on such terrains are ensured. This innovative approach combines deviation learning and HF control techniques, providing a better solution for enhancing the walking capabilities of robots on discrete terrains.

The remaining sections of this article display the following content. Section 2 describes the kinematic model of a robot that walks on two legs and the statement of the associated gait problem; section 3 presents the design of gait optimisation; section 4 displays ad comments the experimental results of this study; finally, section 5 summarises the conclusions of this work.

### 2. Robot Model and Preliminaries

### 2.1. Dynamic Model of Biped Robot

The model of a walking biped robot with *n*-degree of freedom (*n*-DOF) is here represented by a hybrid dynamic model. This hybrid model includes a continuous single-leg support phase (SSP) and discrete impact events. In this model, one degree of freedom is removed from the robot's model. If we consider gravity, the dynamic equation of a joint can be formulated by exploiting Lagrange's method [75]:

$$D(q)\ddot{q} + C(q,\dot{q})\dot{q} + G(q) = \tau \tag{1}$$

where  $\tau \in \mathbb{R}^n$  is the torque applied to each robot's joint;  $q, \dot{q}, \ddot{q} \in \mathbb{R}^n$  represent position, velocity, and acceleration of the joint (they are vectors as each joint is in a 3D space);  $D(q) \in \mathbb{R}^{n \times n}$  and  $C(q, \dot{q}) \in \mathbb{R}^{n \times n}$  denote the inertia matrix and the Coriolis and centripetal forces; at last G(q) formalises the effect of gravity on the robot.

The Lagrangian model presents a discontinuity when the robot's leg touches the ground. Let us consider a function H describing the height of the end of the swing leg above the ground. S = $\{(q, \dot{q}) | H(q) = 0, \nabla H(q)\dot{q} < 0\}$  is the set of states  $(q, \dot{q})$  occurring when the swinging leg touches the ground.

The impact model from [75] is given by:

$$\begin{cases} q(t^+) = q(t^-) \\ \dot{q}(t^+) = Z(q(t^-))\dot{q}(t^-), (q(t^-), \dot{q}(t^-)) \in S \end{cases} (2)$$

(-, +) and Z indicate, respectively, the state values and transition matrix before and after the impact.

The complete dynamic model of the robot under consideration is composed of (1) and (2).

### 2.2. Motion Planning Challenges and Preliminaries

#### 2.2.1. Problem Statement

In the walking planning phase for biped robots, the primary challenges involve designing appropriate foot placement and computing target joint angles utilising inverse kinematics. During the inverse kinematics solution process, singular poses may arise, resulting in control failure and the robot's inability to walk. Moreover, joint angle constraints are of vital importance for maintaining control safety. As such, circumventing singular poses while solving inverse kinematics under constraint conditions is a pressing issue that must be addressed. The damped least squares method is a classic approach for solving inverse kinematics, which can effectively avoid singularities [19]. However, it does not account for constraints. To address this issue, our study will adopt a strategy that incorporates joint angle constraints into the solution process.

# 2.2.2. Damped Least Squares with Constraints for Inverse Kinematics

The damped least squares method is a common numerical technique used to solve the inverse kinematics problems of robots. It achieves the desired endeffector pose by adjusting the joint angles while effectively avoiding singularity issues through the introduction of damping terms. In this paper, we introduce angle constraints within this method, imposing limitations on the robot's joint angles. This ensures that the robot consistently operates within a safe range of joint motion, which is crucial for preventing damage or unexpected accidents, particularly in extreme scenarios. Additionally, it enhances stability and mitigates singularity issues.

Inverse kinematics refers to the process of computing robot joint angles to achieve a desired endeffector position and orientation. In biped robot posture control, inverse kinematics is a critical issue that requires accurate solutions for stable walking and balancing. The inverse kinematics problem involves finding the joint angle vector q, such that the robot's endeffector reaches the target position and orientation. The robot's motion equation can be represented as follows:

$$x = f(q) \tag{3}$$

where, x represents the position and orientation of the end effector, q denotes the joint angle vector, and f is a nonlinear mapping function.

In order to address the inverse kinematics problem, we utilise the Jacobian matrix, which signifies the rate at which the end effector's position and orientation change with respect to the joint angles. The Jacobian matrix is delineated as follows:

$$J = \partial f(q) / \partial q \tag{4}$$

In the damping and constraint-based least-squares method, we aim to solve for the joint angle increment  $\Delta q$ , which minimises the following objective function:

$$\min \|J\Delta q - e\|^2 + \lambda^2 \|\Delta q\|^2 \tag{5}$$

where  $\Delta q$  is the increment of joint angles, *e* represents the error in the end-effector's position and orientation, and  $\lambda$  is the damping coefficient. To solve this problem, we can compute the joint angle increment  $\Delta q$  as follows:

$$\Delta q = (J^T J + \lambda^2 I)^{-1} J^T e \tag{6}$$

In practical applications, robot joints often need to satisfy certain constraints, such as joint angle and velocity limits as well as collision avoidance. For example, joint angle constraints:  $q_{min} \le q \le q_{max}$ , where  $q_{min}$  and  $q_{max}$  represent the minimum and maximum limits of the joint angles, respectively. When joint angle constraints are added to the damped least squares method, the optimisation problem becomes a constrained optimisation problem. The formulation can be expressed as follows:

$$\min |J\Delta q - e|^2 + \lambda^2 |\Delta q|^2$$
s.t.  $q_{\min} \le q + \Delta q \le q_{\max}$ 
(7)

To solve the inverse kinematics problem with constraints, we can introduce these constraints into the least squares problem, forming a constrained optimisation problem. By solving this optimisation problem, we can obtain the joint angle increments  $\Delta q$  that satisfy the constraints, thus achieving the inverse kinematics solution.

# 2.3. Walking Control and Preliminaries

### 2.3.1. Problem Statement

After determining the target foot placement and target joint angles during the planning phase, the controller enables biped robot walking by adjusting joint angles. In this stage, the main challenges lie in ensuring control safety and achieving control precision. To maintain safety, constraints are generally introduced at the control level, while a well-designed controller is crucial for control precision. In our prior research, we primarily employed an HF controller (see Fig. 1), which is a model-free control method. To further enhance the accuracy of robot gait control, we aim to develop a constrained DDPG algorithm to correct the robot's gait, building upon our previous controller. This method integrates the constraints with model-free reinforcement learning, facilitating seamless integration with the previously designed controller.

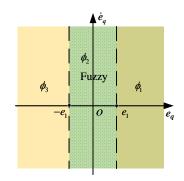


Figure 1. Human-simulated Fuzzy Controller.

In [26], the HF controller for gait control of a biped robot was introduced. The HF controller addressed the jitter issue during mode switching in human-simulated intelligent controllers using fuzzy algorithms. As illustrated in Fig. 1 and Table. 1, this controller employs PD (Proportional-Derivative) control when the value is greater than a certain threshold of  $|e_1|$  and switches to fuzzy control when the value is less than  $|e_1|$ .

Table 1. Control Law of HF

Feature State	Control Law	Remark
$\phi_1: e_q \ge e_1$	$K_p e_q + K_d \dot{e}_q$	PD
$\phi_2: -e_1 \leq e_q < e_1$	fuzzy control	
$\phi_3: e_q \leq -e_1$	$K_p e_q + K_d \dot{e}_q$	PD

# 2.3.2. DDPG with constraints

DDPG algorithm is a method that combines deep learning and reinforcement learning to solve the problem of continuous action space. DDPG is an extension of deterministic policy gradient algorithm by introducing deep neural networks [76,77,78,79,80,81,82,83] as approximators of actors (policy) and critics (value functions). It mainly includes the following parts:

1) Network component

a.Actor Network: It is a deep neural network that is responsible for learning strategies for choosing the optimal action in a given state. It receives the current state of the environment of s, as input and outputs a deterministic action of a.

b.Actor Target Network: It is a copy of the Actor network, which is used to calculate the target value during training to improve the stability of training. Its weights are not updated as frequently as the Actor network, but track the Actor network in smaller steps during training.

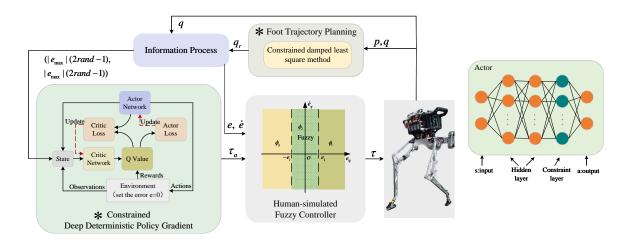
c.Critic Network: It is a deep neural network, which is responsible for learning the action value function Q(s,a) for a given state and action. It receives the current state of the environment *s* and the action *a* actually performed as input, and outputs an estimate of the action value function Q(s,a).

d. Critic Target Network: It is a copy of the critic network that is used to calculate the current value Q' during training. Its weights are not updated as frequently as the critic network, but track the critic network in smaller steps during training.

2) Experience Relay Buffer

DDPG uses an experiential replay buffer to store the experiences  $(s_t, a_t, r_t, s_{t+1})$  generated during the agent's interaction with the environment. During the training process, the agent randomly selects a batch of experiences from the buffer zone for learning, which helps to break the time correlation between experiences and improve the stability of training. 3) Policy Gradient Optimisation

DDPG uses a deterministic policy gradient method to optimise Actor network. The critic network is trained by minimising the error between the predicted value and the actual value. Meanwhile, the Actor network is trained by maximising the value function of the action predicted by the network of critics. By integrating the target value network with the target policy



/

**Figure 2.** The left diagram is the control framework. \* This represents the innovation points: 1) Obtaining joint target angles  $q_r$  using constrained damped least squares; 2) Processing the maximum error in joint angles,  $e_{max}$ , as the state input to CDDPG (Constrained Deterministic Deep Policy Gradient), enabling offline training to derive a deviation correction model. The right diagram illustrates the addition of a constraint layer to the Actor network in CDDPG.

network, the Q value of the next state is calculated. It is updated by minimising the mean square error (MSE) loss *L*:

$$L = \frac{1}{N} \sum_{i} \left( y_i - Q\left( s_i, a_i | \theta^Q \right) \right)^2 \tag{8}$$

The policy loss is updated as:

$$\nabla_{\theta^{\mu}} J \approx \frac{1}{N} \sum_{i} \nabla_{a} Q\left(s, a | \theta^{Q}\right) |_{s=s_{i}, a=\mu(s_{i})} \nabla_{\theta^{\mu}} \mu\left(s | \theta^{\mu}\right) |_{s_{i}}$$
(9)

The only adjustable coefficient in Equations 8 and 9 is the number of samples, denoted as N. In Eq. 8, changing N impacts the MSE loss function. Increasing N typically decreases MSE, enhancing stability and robustness in model training. However, it also escalates computational costs and training time. Conversely, reducing N raises MSE, leading to a less stable loss function. In Eq. 9, increasing N improves gradient estimate accuracy but at the expense of complexity and computation time. Choosing an appropriate N depends on the specific problem and available computing resources.

DDPG algorithm is a reinforcement learning algorithm in continuous action spaces, aiming to learn a mapping from a continuous state space to a continuous action space so that the agent can perform appropriate actions in the environment to maximise the cumulative reward. However, in practical applications, the agent often needs to satisfy some constraints on states or actions, such as limiting the joint angles of a robot while performing actions. In order to achieve state or action constraints without affecting the performance of the algorithm, this paper proposes a new method of adding a constraint layer in DDPG algorithm to achieve action constraints.

In the DDPG algorithm, the Actor network learns the mapping from state to action, while the Critic network learns the value function of state and action. To impose constraints on actions, we can augment the Actor network with a constraint layer. This layer serves to restrict the output actions within the designated constraint boundaries, thereby ensuring adherence to the defined limits. Specifically, the constraint layer of the Actor network limits the action increments within a certain range and also limits the joint angles within the predetermined constraint range after executing the action increments.

### 3. Gait Optimisation Design

# 3.1. Control scheme framework

In this section, we describe the workflow of the proposed method of this paper. Combined with Fig. 2, firstly, the target joint angle  $q_r$  ( $q_{r_{new}}$ ) within the constraint angle range was obtained through the constrained damping least square method, and the angle error information  $e(\Delta q)$ ,  $\dot{e}$  was obtained and input to the HF controller to control the robot motion. After obtaining the motion data, we can get the extreme value  $e_{max}$  of the two joint errors of the swinging leg, which is processed as  $(|e_{max}|(2rand-1), |e_{max}|(2rand-1))$ and input to the constrained DDPG for off-line training. This model can learn how to make the joint move from the current error to the target value 0. The trained model is combined with the HF controller so that the motion deviation caused by the HF controller can be compensated. Two of the main points are as follows:

1) The motion control of a biped robot can be divided into two stages: foot trajectory planning and control execution. As shown in Fig. 1, during the trajectory planning stage, a cubic interpolation is utilised to calculate the swing foot's movement trajectory, resulting in the desired foot placement position  $p_r$ . Then, the position error  $\Delta p$  is calculated based on the actual position p of the swing foot and the desired position  $p_r$ . By employing constrained damping least squares, the target joint angles  $q_r$ , are derived based on  $\Delta p$ .  $q_r$ , along with the robot's actual joint angles q, are then input to the information processing module, enabling the acquisition of angle error e and its derivative  $\dot{e}$ . Finally, these inputs are passed to the HF controller, which serves as the fundamental controller responsible for executing motion control.

2) Furthermore, the composition of constrained DDPG has been described in 2.3.2, and the absolute values of the extreme errors in the hip and knee joints are mapped to random values in four quadrants, serving as state inputs for the Actor network in the constrained DDPG. As depicted in Fig.2, the Actor network of the constrained DDPG includes a constraint layer inserted before the output layer. This constraint layer plays a vital role in applying constraints to the generated output actions, ensuring their adherence to a safe range of angular motion. By incorporating the

constraint layer, the system guarantees that the generated actions operate within an acceptable angle range, thereby enhancing both safety and reliability. Notably, the purpose of deviation correction is to achieve zero error in control. Therefore, the constrained DDPG utilises a target value of 0 for training, which is performed offline. The trained network is then integrated with the angle deviation corrector, working in tandem with the HF controller. This integration allows for the realization of biped robot locomotion by leveraging the combined functionality of the angle deviation corrector and HF controller.

# 3.2. Inverse kinematics of a biped robot based on least square method with damping and constraints

This section describes in detail the process of using the damped least square method to solve the target joint angles in the angle constraint range of 90°-180°, which can ensure that the singularity problem is avoided in the safe angle range. First, we need to set the target foot placement  $p_d$  of the robot, then add angle constraints to the joints, and finally apply a constrained damped least square method to compute the target joint angles.

(1) Set the foot placement  $p_d$  for the biped robot: The foot placement  $p_d$  is a predetermined end effector position that can be set based on the robot's motion requirements and terrain features. In this paper, to fully validate the control accuracy, we designed a discrete point terrain device and fixed foot placement,where  $p_d = T_s/2 * v$ ,  $T_s$  is the single step time, and v is the target velocity of the biped robot.

(2) Add angle constraints to the joints: The joint angles q of a biped robot are often restricted by physical limitations, such as maximum and minimum limits. To ensure the robot's safe and normal operation, it is necessary to consider these constraints when solving the inverse kinematics problem. In this paper, we added angle constraints of 90° to 180° to the joints. This means that the joint angles solved must fall within this range.

(3) Calculate target joint angles using a constrained damped least squares method: To begin with, we calculate the difference between the current end effector position p and the target end effector position  $p_d$ :  $\Delta p = p_d - p$ . Here,  $p_d$  is the predetermined foot placement, and p is the current end effector position. Next, we use the damping least squares method (DLS) to calculate the variation of joint angle deviation,  $\Delta q$ :

$$\Delta q = (J^T J + \lambda^2 I + \Phi(q))^{-1} J^T \Delta p \qquad (10)$$

where, *J* is the Jacobian matrix,  $\lambda$  is the damping coefficient, *I* is the identity matrix, and  $\Phi(q)$  is the joint angle constraint matrix, which contains information about the upper and lower limits of the joint angles.

For a planar robot with n joints, the Jacobian matrix J is a 2n matrix, where each column represents the Jacobian elements of a joint. Specifically, we can express it as:

$$J = [J_1, J_2, \dots, J_n]$$
(11)

where,  $J_i$  is the Jacobian element of the *i*th joint, which can be represented as:

$$J_i = \left[\frac{\partial p_x}{\partial q_i}, \frac{\partial p_z}{\partial q_i}\right]$$
(12)

where,  $p_x$  and  $p_z$  represent the position of the end effector on the x and z axes, respectively, and  $q_i$  is the angle of the *i*th joint. The Jacobian matrix J can be used to describe the relationship between the change in joint angles and the change in end effector position for the robot.

 $\Phi(q)$  represents the joint angle constraint matrix, which is typically a diagonal matrix with diagonal elements used to apply constraints to the angle of each joint. For a robot with *n* joints,  $\Phi(q)$  is an *n*×*n* matrix, expressed as follows:

$$\Phi(q) = diag(\phi_1(q_1), \phi_2(q_2), \cdots, \phi_n(q_n))$$
(13)

where, diag represents a diagonal matrix,  $\phi_i(q_i)$  is the constraint function for the *i*th joint, which can be defined based on the limits of the joint angle. In this study we restrict the joint angle within a certain range as  $[q_{min}, q_{max}]$ , the constraint function can be expressed as:

$$\phi_i(q_i) = k(q_i - q_{min})(q_{max} - q_i) \tag{14}$$

where, k is a weighting coefficient used to control the strength of the constraint,  $q_{max} = 180^{\circ}$  and  $q_{min} = 90^{\circ}$ . As the joint angle approaches the boundary of the limit, the value of the constraint function increases, thereby strongly maintaining the joint angle within the allowable range during inverse kinematics computation. Finally, we update the current joint angle q:

$$q_{r_{new}} = q + \Delta q \tag{15}$$

Through this process, we can obtain the target joint angles  $q_{r_{new}}$  that satisfy the constraint conditions. These angles can be used to control the biped robot's walking during the planning phase.

# 3.3. Application of Constrained DDPG Algorithm for Gait Optimisation of Biped Robot

To apply the constrained DDPG algorithm for gait optimisation, we first use the least square method with damping and constraints to calculate the target joint angles. Then, we use an HF controller to complete the basic posture control, followed by using the constrained DDPG algorithm to compensate for joint errors. In this way, we can achieve precise control of the posture of the swinging leg of the biped robot while ensuring that the actual constraints of joint angles are satisfied during its movement. In our research, the environment is a simulation that represents the physical system of the robot, including two joint angles (hip and knee). We utilized the DDPG algorithm to train a model correcting joint angle deviations by interacting with the environment. The DDPG agent interacts through time steps, adjusting movements to minimize deviations and maximize long-term rewards, optimizing performance toward zero joint angle deviation.

In Fig.2, the joint angle error of the swing leg is processed as  $|e_{max}|(2rand - 1)$  and then input into the Constrained DDPG for training as the initial state. The design process of the DDPG algorithm with added constraint layer is as follows:

1) Define the Actor and Critic network structure;

2) Add a constraint layer to the Actor network structure. In the constraint layer, action increments are calculated based on input state *s*, and then the action increments are limited to a preset range,  $\Delta q$  is the

joints angle error, and  $\Delta q_{max}$  is the maximum joints angle error.

$$\Delta q = max(-\Delta q_{max}, min(\Delta q_{max}, \Delta q)) \qquad (16)$$

3) Calculate the post-execution joint angle based on the current state and the post-execution action increment and limit it to a preset constraint.

$$s' = max(q_{min}, min(q_{max}, s + \Delta q))$$
(17)

4) The joint angle after execution is regarded as the state of the next moment, and the main process of DDPG algorithm is continued.

Algorithm 1 Constrained DDPG

- 1: Define Actor with constraint layer; Define Critic.
- 2: Initialize  $Q(s,a|\theta^Q), \mu(s,a|\theta^{\mu})$  with random weights.
- 3: Initialize target  $Q', \mu'$  with  $\theta^{Q'} \leftarrow \theta^Q, \theta^{\mu'} \leftarrow \theta^{\mu}$ .
- 4: Initialize relay buffer *R*.
- 5: for episode = 1 to M do
- 6: Reset environment *s*; Initialize cumulative reward  $r_m$ .
- 7: **for** step = 1 to maxSteps **do**
- 8:  $a_t = \mu(s_t | \theta^{\mu}) + N_t \quad \triangleright \text{ Action selection}$ with noise
- 9: Execute  $a_t$ , get reward  $r_t$ , observe new state  $s_{t+1}$ .
- 10: Store  $(s_t, a_t, r_t, s_{t+1})$  in R. 11: **if** size of  $R \ge$  miniBatch **then**
- 12: Sample minibatch from *R*.
- 13: Compute targetAction using Q', targetQ using  $\mu'$ .
- 14:Update Critic and Actor networks.15:Soft update:  $\theta^{\mu'} \leftarrow \tau \theta^{\mu} + (1 \tau) \theta^{\mu'}$ , $\theta^{Q'} \leftarrow \tau \theta^Q + (1 \tau) \theta^{Q'}$ .16:end if17:end for

18: end for

Algorithm 1 presents the pseudo-code for the training of Constrained DDPG algorithm. A customized constraint layer is integrated into the Actor network, enabling training of the neural network within the specified angle boundaries, that is 90°-180°. This model not only considers the movement towards

the target position (the joints angle errors move from the initial state to the target state 0) but also incorporates decision-making related to safety constraints. By incorporating the constraint layer, the agent's exploration range in action spaces is restricted, thereby reducing algorithmic uncertainty and volatility. This mechanism facilitates the agent in learning reasonable and viable strategies more effectively, leading to accelerated learning. Moreover, it ensures that the agent's actions remain within the designated safe range, minimising potential risks and losses.

### 4. Experimental Results

To verify the control effectiveness of the scheme proposed in this paper, based on the biped robot Crane dynamics model, we used Matlab R2019a to construct the model for numerical simulation. Parameters are shown in Table 2: to verify the effectiveness of the innovation subpart and the overall scheme, the verification is carried out in four parts: 1) constraint damping least square method verification; 2) constrained DDPG training results; 3) experiments of a biped robot up and down stairs; 4) experiments of a biped robot walking on discrete terrains. In experiments 3) and 4), our primary focus is to benchmark our approach against the HF control method, which is considered an advanced solution for addressing the specific problem under investigation [26] since it recently displayed a superior performance with respect to state-of-the-art methods, e.g., [75,84,85,86]. By comparing our method with HF, we aimed to demonstrate the effectiveness of our proposed enhancements and validate their impact on improving the existing state-of-the-art solution.

Table 2. Parameters of Biped Robot Crane

		1	
Biped	Mass (kg)	Length (m)	Inertia $(kg \cdot m^2)$
Torso	5.580	0.257	0.043320
Thigh	0.548	0.401	0.000680
Shank	0.813	0.300	0.000034

The experimental model has 4 active joints, and the corresponding control joint angles are  $q_1$ ,  $q_2$ ,  $q_3$ ,  $q_4$ , corresponding to the hip joint of the knee joint of the left leg, and the hip joint and knee joint of the right leg. The controller's input is the error of the joint angle  $e_i$  (i = 1, 2, 3, 4) and error derivative  $\dot{e}_i$  (i = 1, 2, 3, 4). The output is joint torque  $\tau_i$ .

The extrema of joint angle errors were used as the state input for constrained DDPG, with a training objective of achieving zero error. Then, offline training was conducted, and the trained network was utilised to correct the angle deviations based on the fundamental HF control implementation.

# *4.1. Verifications of constrained damping least squares method*

To validate the effectiveness of the constrained damping least squares method, we performed two verification steps. Firstly, we examined the efficacy of the damping least squares method in addressing singular values. Subsequently, we assessed the effectiveness of the constraint conditions.

1) In order to validate the effectiveness of this method, considering that singular values can lead to control failures, a specific target singular state was intentionally established in this study. This involved adjusting the knee joint of the swinging leg to  $0^{\circ}$ , and subsequently observing the control results from the initial state to the target state. Fig. 3 depicts the time-varying curves of joint angular velocities under the direct inverse kinematics method. According to the

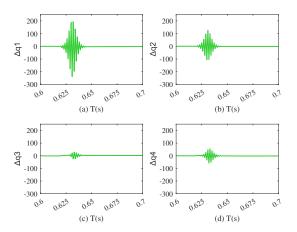


Figure 3. Joint angular velocities under the direct inverse kinematics method.

figure, at 0.63 seconds, there is a sudden change in

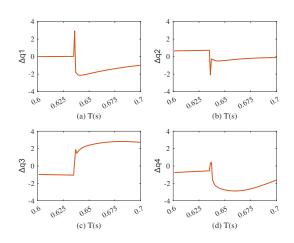


Figure 4. Joint angular velocities under constrained damped least square method.

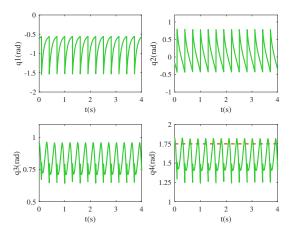


Figure 5. The target joint angle without constraints.

joint angular velocity, accompanied by severe shaking, leading to instability and control failure. Fig. 4 illustrates the time-varying curves of joint angular velocities when employing the method proposed in this study. After introducing damping, the joint angular velocities are significantly constrained within the singularity region of the Jacobian matrix. The curves of joint angular velocities exhibit smoother behavior. Through this validation, it is demonstrated that this method effectively avoids the occurrence of singular values.

2) In this study, a safety constraint angle range of  $[0^{\circ} - 100^{\circ}]$  was established. Without the inclusion of constraints, Fig. 5 indicates that the computed target joint angle q4 exceeds the imposed limit of 100 degrees. Such an occurrence poses a significant risk of

**Table 3.** Joint Angle Error Performance of Up and Down Stairs. This table shows the error data of joint angle in experiments with different stair heights (0.01 m, 0.02 m, 0.03 m), that is, the Mean Error (ME)  $\pm$  Standard Deviation (SD). SAME is the sum of absolute values of ME for each joint.

		HF			HFCDDPG	
	0.01	0.02	0.03	0.01	0.02	0.03
$q_1$	$-0.0865 \pm 0.1362$	$-0.0973 \pm 0.1405$	-0.1121±0.1521	$-0.0775 \pm 0.1148$	$-0.0910 \pm 0.1006$	-0.1008±0.1348
$q_2$	$0.1342{\pm}0.1604$	$0.1200{\pm}0.1662$	$-0.2403 {\pm} 0.1855$	$-0.1280{\pm}0.1449$	$0.0899 {\pm} 0.1526$	$-0.1730{\pm}0.1692$
$q_3$	$0.0081 {\pm} 0.2750$	$0.1580 {\pm} 0.2853$	$0.2589 {\pm} 0.2967$	$-0.0017 \pm 0.2083$	$0.1072{\pm}0.2112$	$0.1161 {\pm} 0.2238$
$q_4$	$0.2124{\pm}0.2492$	$0.4743 {\pm} 0.2538$	$0.2965{\pm}0.2598$	$0.1508 {\pm} 0.2328$	$0.4062{\pm}0.2371$	$0.2852{\pm}0.2531$
SAME	0.4412	0.8496	0.9078	0.3580	0.6943	0.6751

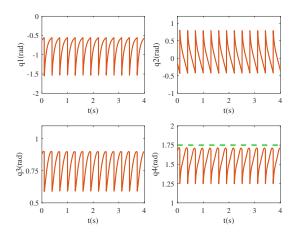


Figure 6. The target joint angle with constraints.

instability and potential robot falls. To ensure system stability and safety, the introduction of constraint conditions was necessary.

Fig. 6 illustrates the target joint angles after the incorporation of constraints, effectively constraining them within the designated safety range. This constraint implementation serves to protect the robot from excessive joint angles, mitigating unsafe movements.

### 4.2. Constrained DDPG training

In this section, we conducted a comparison between the common angle constraint method and the method proposed in this study. We utilised the angle deviations of the knee and hip joints of the swinging leg as the state inputs for the Actor network. To ensure a comprehensive assessment of the method's effectiveness, we randomly selected four sets of state values from the four quadrants of these deviations for training. In the common constraint method, training is halted and restarted with a new training session whenever the constraints are violated. In contrast, this study incorporates a constraint layer into the Actor network to ensure compliance with the constraint conditions. By comparing Fig. 7, it is evident that the proposed method achieves the maximum reward value more rapidly and attains a higher maximum reward compared to the conventional approach. Additionally, the state transition plot in Figs. 8 and 9 clearly illustrate that, although both methods operate within the safety constraints, the proposed method converges faster and demonstrates greater stability throughout the convergence process.

By incorporating the constraint layer into the training process, the proposed method in this study not only complies with the constraints but also brings advantages in two aspects. Firstly, it enables faster attainment of the maximum reward value, specifically, the convergence rate ratio between CDDPG and DDPG in Fig. 7(a) is 3/2, that of Fig. 7(b) is 2.5, and that of Figs. 7(c) and (d) is 2. Therefore, these results show that the method of CDDPG can effectively learn control strategies. Secondly, the increase in the maximum reward value signifies that the system exhibits improved execution of desired actions during training, thereby enhancing control accuracy and stability.

### 4.3. Up and down stairs

In the up and down stairs experiments (see Fig. 10), our aim is to evaluate the performance of differ-

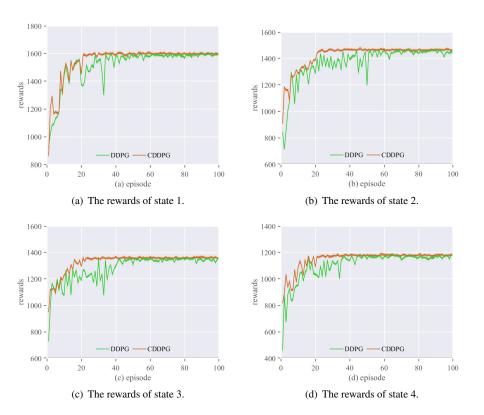


Figure 7. Rewards for different states.

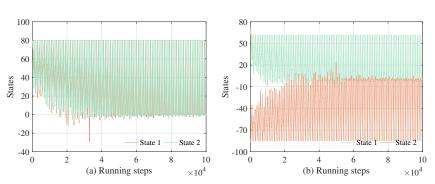
ent control methods, particularly comparing the HF and HFCDDPG control methods. Three different stair heights (0.01 m, 0.02 m, and 0.03 m) provided a diverse set of challenges, requiring the robot to execute tasks with stability and precision across varying heights.

First, let's focus on the performance of the HF control method. As seen in Fig.11, while the robot is capable of navigating the stairs, significant issues become apparent. Firstly, the foot placement and height of the first step are not accurately maintained, potentially affecting the precision of the robot's stride. In Figs.11 (a) and (b), the sixth and seventh steps display sudden increases in stride, resulting in an uneven gait that impacts the robot's balance. Most notably, in Fig.11 (c), the fourth step exhibits two landing points, indicating an inconsistency in the robot's stride during this phase. Additionally, the sixth step also displays two landing points, attributed to the robot's backward motion at the final step, a clear sign of task failure.

Particularly at a stair height of 0.03 m, the robot's performance is notably inferior, highlighting the limitations of the HF control method under challenging conditions.

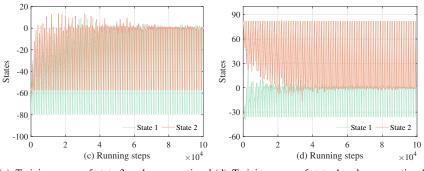
In contrast, the HFCDDPG control method excels in the task of up and down stairs. As seen in Fig. 12, regardless of the stair height, the robot can smoothly complete the task. Unlike HF, the HFCDDPG method does not exhibit multi-step actions, sudden increases in stride, or noticeable backward movements. Foot placement data indicates overall uniformity and stability, demonstrating that HFCDDPG can maintain excellent control precision across varying stair heights.

Table. 3 presents the average and standard deviation of joint errors for the three stair heights. Furthermore, the Sum of Absolute values of Mean Errors (SAME) over all the joints has been calculated as an aggregated metric of the accuracy of the controller. Observing Table. 3, we can clearly see the significant advantage of the HFCDDPG control method in terms



/

(a) Training curve of state 1 under conventional (b) Training curve of state 2 under conventional constraints.



(c) Training curve of state 3 under conventional (d) Training curve of state 4 under conventional constraints.

Figure 8. Training curve of different states under conventional constraints.

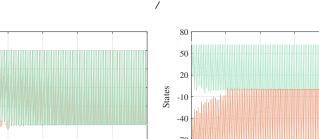
of joint errors. Not only the average error is smaller, but the standard deviation is also lower, indicating that HFCDDPG can maintain more consistent joint posture control across different stair heights. This result is of paramount importance, as stable joint control is crucial for the safety and successful completion of robotic tasks. In contrast, the HF method exhibits greater variability in terms of joint errors, with larger average errors and higher standard deviations. This implies that under HF control, the robot may demonstrate significant variations across different trials, which could be deemed unacceptable for practical applications. Furthermore, this observation corroborates the previously noted irregular and unstable gait in Fig.11, suggesting a potential connection between the variability in joint errors and these gait issues.

In summary, the HFCDDPG control method showcases outstanding stability and precision in the task of ascending and descending stairs. Compared to the HF method, it demonstrates a higher degree of adaptability when facing challenges posed by varying stair heights.

### 4.4. Walking on discrete terrains

To verify the effectiveness of the entire control scheme, this study conducted tests on two types of discrete terrains: uniform (Fig. 13) and non-uniform (Fig. 15).

In the uniform terrain, three different lengths of discrete points were set. In the top of Fig. 14, the step length is 0.2 m, in the middle of Fig. 14, the step length is 0.23 m, and in the bottom of Fig. 14, the step length is 0.3 m. In the figures, the blue color represents the target values, the green color represents the control effect of HF method, and the red color represents the effect of the method proposed in this study. Through comparison, it is evident that the proposed method ensures the precision of the foot placement, enabling stable walking on discrete terrains. On the other hand, the HF method cannot guarantee that all foot placements



(a) Training curve of state 1 with constrained (b) Training curve of state 2 with constrained layer. layer.

10

 $\times 10^4$ 

-100

0

2

4

(b) Running steps

6

8

10

 $\times 10^4$ 

State 2

8

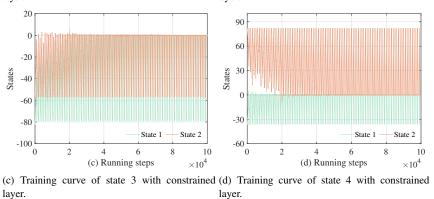


Figure 9. Training curve of different states with constrained layer.

Table 4. Joint Angle Error Performance of Uniform Discrete Terrain. This table shows the error data of joint angle in experiments with different discrete terrain lengths (0.2 m, 0.3 m, 0.4 m), that is, the Mean Error (ME) ± Standard Deviation (SD). SAME is the sum of absolute values of ME for each joint.

		HF			HFCDDPG	
	0.2	0.3	0.4	0.2	0.3	0.4
$q_1$	$-0.0780 {\pm} 0.0879$	$-0.1328 \pm 0.1532$	$-0.1558 {\pm} 0.1818$	$-0.0641 {\pm}\ 0.0662$	$-0.0967 \pm 0.1031$	-0.1397±0.1648
$q_2$	$-0.0036 \pm 0.0160$	$-0.0060 \pm 0.0262$	$-0.0069 \pm 0.0304$	$-0.0029 \pm 0.0103$	$-0.0043 \pm 0.0134$	$-0.0055 \pm 0.0172$
$q_3$	$-0.0814 \pm 0.2180$	$-0.1848 {\pm} 0.3069$	$-0.2287 \pm 0.3434$	$0.0189 {\pm} 0.1419$	$-0.0248 \pm 0.1960$	$-0.1156 {\pm} 0.3381$
$q_4$	$0.0763 {\pm} 0.1742$	$0.0602{\pm}0.1738$	$0.0524{\pm}0.1843$	$0.0655 {\pm} 0.2036$	$0.0387 {\pm} 0.1910$	$0.0073 {\pm} 0.1812$
SAME	0.2393	0.3838	0.4438	0.1514	0.1645	0.2681

reach their target positions, thus leading to unsuccessful locomotion on discrete terrains. Table. 4 shows the performance of joint error in the experiment of three uniform discrete terrains (0.2 m, 0.3 m, 0.4 m). Compared with the HF data, it can be seen that the error and standard deviation of HFCDDPG are smaller. HFCD-DPG control is more accurate than HF.

100

80

60

40 States

20

0 -20

-40 0

2

Δ

(b) Running steps

6

In the non-uniform discrete terrain, we intro-

duced discrete points with varying heights and lengths. Similarly, the proposed method demonstrated successful locomotion, while the HF method failed to guarantee successful walking. On both, uniform and nonuniform discrete terrains, in Fig.s 14 and 16, we may observe that the HF-controlled robot misplaced the foot on average in the 30% of the steps, thus being in practice unusable. Conversely, HFCDDPG appears to land

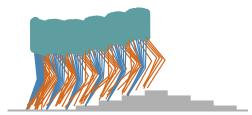


Figure 10. Uniform discrete terrain walking.

the foot in close proximity to the target points, thus displaying much better stability performance.

These results confirm the effectiveness and superiority of the proposed method. By utilising the constraint DDPG control strategy, stable and precise walking on discrete terrains was successfully achieved. In comparison to the traditional HF approach, the proposed method overcomes the difficulties encountered during walking on discrete terrains, resulting in higher success rates and improved accuracy.

In conclusion, the control scheme proposed in this study has achieved satisfactory results on discrete terrains. This achievement holds significant implications for the walking capability and stability of robots in real-world scenarios.

# 5. Conclusion

In this study, ensuring safety throughout the entire operation process was a paramount concern. To address the singularity issue within a defined safe range of angles, we incorporated the constraintdamping least squares method during the planning phase. During the control process, we initially implemented an HF controller to establish basic control and subsequently integrated a designed constraint DDPG learner to rectify joint angle deviations. Constraint DDPG combines the ideas of Reinforcement Learning and security constraint. The neural network within the constraint DDPG learner aimed to maximise longterm rewards by minimising target deviations within the safe angle range. This approach facilitated safe learning during the training process. By constraining the range of actions through the constraint layer, we maintained system stability while ensuring safety, ultimately enhancing training efficiency.

We conducted a comprehensive evaluation of the proposed method's effectiveness, assessing its performance in handling singular states, the efficacy of the constraints, and its ability to achieve stable up and down stairs and walking on discrete terrains. Encouraging results were obtained across all evaluations. Looking ahead, our future plans involve implementing this control framework on a physical robot to further validate its practical applicability.

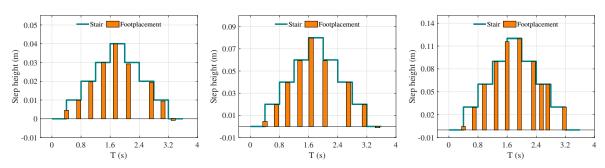
### Acknowledgments

This work was supported by the National Natural Science Foundation of China (61972324), Beijing Advanced Innovation Center for Intelligent Robots and Systems (2019IRS14), and Jiangsu Distinguished Professor programme.

### References

- P. F. Escobar and T. Falcone, Atlas of Robotic, Conventional, and Single-port Laparoscopy: A Practical Approach in Gynecology. Springer, 2022.
- [2] M. K. Strait, C. Aguillon, V. Contreras, and N. Garcia, "The public's perception of humanlike robots: Online social commentary reflects an appearance-based uncanny valley, a general fear of a "technology takeover", and the unabashed sexualization of female-gendered robots," in 2017 26th IEEE international symposium on robot and human interactive communication (RO-MAN). IEEE, 2017, pp. 1418–1423.
- [3] S. Caron, A. Kheddar, and O. Tempier, "Stair climbing stabilization of the hrp-4 humanoid robot using whole-body admittance control," in 2019 International conference on robotics and automation (ICRA). IEEE, 2019, pp. 277–283.
- [4] D. Yao, L. Yang, X. Xiao, and M. Zhou, "Velocity-based gait planning for underactuated bipedal robot on uneven and compliant terrain," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 11, pp. 11414–11424, 2021.
- [5] Q. Huang, C. Dong, Z. Yu, X. Chen, Q. Li, H. Chen, and H. Liu, "Resistant compliance control for biped robot inspired by humanlike behavior," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 5, pp. 3463–3473, 2022.
- [6] O. Zahra, D. Navarro-Alarcon, and S. Tolu, "A neurorobotic embodiment for exploring the dynamical interactions of a spiking cerebellar model and a robot arm during vision-based manipulation tasks," *International Journal of Neural Systems*, vol. 32, no. 08, p. 2150028, 2022.
- [7] J. M. Pimentel, R. C. Moioli, M. F. De Araujo, and P. A. Vargas, "An integrated neurorobotics model of the cerebellarbasal ganglia circuitry," *International Journal of Neural Systems*, 2023.

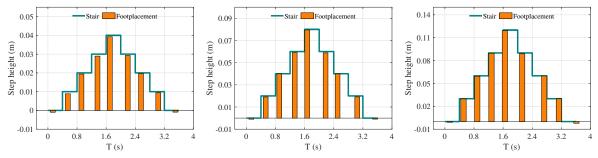
/



/

(a) Foot placement on a 0.01 m high stair with (b) Foot placement on a 0.02 m high stair with (c) Foot placement on a 0.03 m high stair with HF. HF.

Figure 11. Foot placement up and down stairs with HF.



(a) Foot placement on a 0.01 m high stair with (b) Foot placement on a 0.02 m high stair with (c) Foot placement on a 0.03 m high stair with HFCDDPG. HFCDDPG.

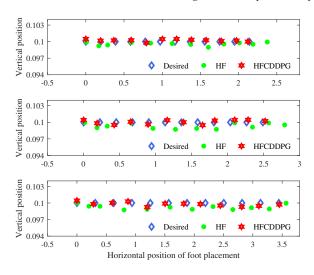


Figure 12. Foot placement up and down stairs with HFCDDPG.

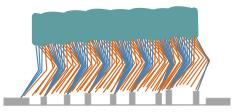


Figure 15. Non-uniform discrete terrain walking.

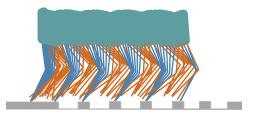


Figure 13. Uniform discrete terrain walking.

[8] J.-K. Huang and J. W. Grizzle, "Efficient anytime clf reactive planning system for a bipedal robot on undulating terrain,"

Figure 14. The foot placement of uniform discrete terrain.

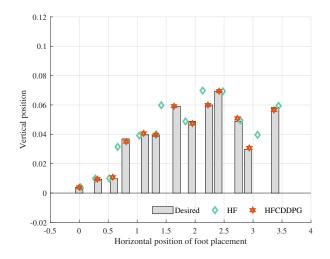


Figure 16. The foot placement of non-uniform discrete terrain.

IEEE Transactions on Robotics, 2023.

- [9] V. Klemm, A. Morra, L. Gulich, D. Mannhart, D. Rohr, M. Kamel, Y. de Viragh, and R. Siegwart, "Lqr-assisted whole-body control of a wheeled bipedal robot with kinematic loops," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3745–3752, 2020.
- [10] K. S. Narkhede, A. M. Kulkarni, D. A. Thanki, and I. Poulakakis, "A sequential mpc approach to reactive planning for bipedal robots using safe corridors in highly cluttered environments," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11831–11838, 2022.
- [11] B. Ren, X. Luo, H. Li, J. Chen, and Y. Wang, "Gait trajectorybased interactive controller for lower limb exoskeletons for construction workers," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 5, pp. 558–572, 2022.
- [12] D. Hu and S. Li, "Recognizing object surface materials to adapt robotic disinfection in infrastructure facilities," *Computer-Aided Civil and Infrastructure Engineering*, vol. 37, no. 12, pp. 1521–1546, 2022.
- [13] Y. Lee, H. Lee, J. Lee, and J. Park, "Toward reactive walking: Control of biped robots exploiting an event-based fsm," *IEEE Transactions on Robotics*, vol. 38, no. 2, pp. 683–698, 2021.
- [14] J. R. Guadarrama-Olvera, S. Kajita, and G. Cheng, "Preemptive foot compliance to lower impact during biped robot walking over unknown terrain," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 8006–8011, 2022.
- [15] S. Makris, Virtual Reality for Programming Cooperating Robots Based on Human Motion Mimicking. Cham: Springer International Publishing, 2021, pp. 339–353. [Online]. Available: https://doi.org/10.1007/ 978-3-030-51591-1\_18
- [16] M. S. Khan and R. K. Mandava, "A review on gait generation of the biped robot on various terrains," *Robotica*, vol. 41, no. 6, p. 1888–1930, 2023.
- [17] M. Giamou, F. Marić, D. M. Rosen, V. Peretroukhin, N. Roy, I. Petrović, and J. Kelly, "Convex iteration for distance-

geometric inverse kinematics," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1952–1959, 2022.

- [18] B. Ames, J. Morgan, and G. Konidaris, "Ikflow: Generating diverse inverse kinematics solutions," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 7177–7184, 2022.
- [19] M. Safeea, R. Bearee, and P. Neto, "A modified dls scheme with controlled cyclic solution for inverse kinematics in redundant robots," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 12, pp. 8014–8023, 2021.
- [20] H. Ye, D. Wang, J. Wu, Y. Yue, and Y. Zhou, "Forward and inverse kinematics of a 5-dof hybrid robot for composite material machining," *Robotics and Computer-Integrated Manufacturing*, vol. 65, p. 101961, 2020.
- [21] J. Hernández-Barragán, C. López-Franco, N. Arana-Daniel, A. Y. Alanis, and A. Lopez-Franco, "A modified firefly algorithm for the inverse kinematics solutions of robotic manipulators," *Integrated Computer-Aided Engineering*, vol. 28, no. 3, pp. 257–275, 2021.
- [22] H. Su, W. Qi, J. Chen, and D. Zhang, "Fuzzy approximationbased task-space control of robot manipulators with remote center of motion constraint," *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 6, pp. 1564–1573, 2022.
- [23] W. He, C. Xue, X. Yu, Z. Li, and C. Yang, "Admittancebased controller design for physical human–robot interaction in the constrained task space," *IEEE Transactions on Automation Science and Engineering*, vol. 17, no. 4, pp. 1937–1949, 2020.
- [24] Z. Chen, Y. Liu, W. He, H. Qiao, and H. Ji, "Adaptiveneural-network-based trajectory tracking control for a nonholonomic wheeled mobile robot with velocity constraints," *IEEE Transactions on Industrial Electronics*, vol. 68, no. 6, pp. 5057–5067, 2020.
- [25] X. Yu, B. Li, W. He, Y. Feng, L. Cheng, and C. Silvestre, "Adaptive-constrained impedance control for human–robot co-transportation," *IEEE transactions on cybernetics*, vol. 52, no. 12, pp. 13 237–13 249, 2021.
- [26] X. Liu, G. Zhang, M. S. Mastoi, F. Neri, and Y. Pu, "A humansimulated fuzzy membrane approach for the joint controller of walking biped robots," *Integrated Computer-Aided Engineering*, vol. 30, no. 2, pp. 105–120, 2023.
- [27] A. Daranda and G. Dzemyda, "Reinforcement learning strategies for vessel navigation," *Integrated Computer-Aided Engineering*, vol. 30, no. 1, pp. 53–66, 2023.
- [28] R. A. Sørensen, M. Nielsen, and H. Karstoft, "Routing in congested baggage handling systems using deep reinforcement learning," *Integrated Computer-Aided Engineering*, vol. 27, no. 2, pp. 139–152, 2020.
- [29] B. Küçükoglu, B. Rueckauer, N. Ahmad, J. de Ruyter van Steveninck, U. Güçlü, and M. van Gerven, "Optimization of neuroprosthetic vision via end-to-end deep reinforcement learning," *International Journal of Neural Systems*, vol. 32, no. 11, pp. 2 250 052:1–2 250 052:16, 2022.
- [30] S. Javadinasab Hormozabad, M. Gutierrez Soto and H. Adeli, "Integrating structural control, health monitoring, and energy harvesting for smart cities," *Expert Systems*, vol. 38, no. 8, pp. 12845–12869, 2021.

- [31] M. Gutierrez Soto and H. Adeli, "Semi-active vibration control of smart isolated highway bridge structures using replicator dynamics," *Engineering Structures*, vol. 186, pp. 536– 552, 2019.
- [32] Z. Li and H. Adeli, "Control methodologies for vibration control of smart civil and mechanical structures," *Expert systems*, vol. 35, no. 6, pp. 12 354–12 374, 2018.
- [33] M. Gutierrez Soto and H. Adeli, "Vibration control of smart base-isolated irregular buildings using neural dynamic optimization model and replicator dynamics," *Engineering Structures*, vol. 156, pp. 322–336, 2018.
- [34] A. Morrone, W. Anderson, and S. J. Simske, "Occluded image function: A novel measure for evaluating machine learning classifiers for biometrics," *Journal of Imaging Science* and Technology, 2021.
- [35] N. Wang and H. Adeli, "Self-constructing wavelet neural network algorithm for nonlinear control of large structures," *Engineering Applications of Artificial Intelligence*, vol. 41, pp. 249–258, 2015.
- [36] Z. Li and H. Adeli, "New discrete-time robust h2/h∞ algorithm for vibration control of smart structures using linear matrix inequalities," *Engineering Applications of Artificial Intelligence*, vol. 55, pp. 47–57, 2016.
- [37] K. Ghaedi, Z. Ibrahim, H. Adeli, and A. Javanmardi, "Invited review: Recent developments in vibration control of building and bridge structures," *Journal of Vibroengineering*, vol. 19, no. 5, pp. 3564–3580, 2017.
- [38] M. Gutierrez Soto and H. Adeli, "Recent advances in control algorithms for smart structures and machines," *Expert Systems*, vol. 34, no. 2, p. e12205, 2017.
- [39] M. G. Soto and H. Adeli, "Multi-agent replicator controller for sustainable vibration control of smart structures," *Journal* of Vibroengineering, vol. 19, no. 6, pp. 4300–4322, 2017.
- [40] M. Gutierrez Soto and H. Adeli, "Many-objective control optimization of high-rise building structures using replicator dynamics and neural dynamics model," *Structural and Multidisciplinary Optimization*, vol. 56, pp. 1521–1537, 2017.
- [41] H. Adeli and S.-L. Hung, "A concurrent adaptive conjugate gradient learning algorithm on mimd shared-memory machines," *The International Journal of Supercomputing Applications*, vol. 7, no. 2, pp. 155–166, 1993.
- [42] S.-L. Hung and H. Adeli, "Parallel backpropagation learning algorithms on cray y-mp8/864 supercomputer," *Neurocomputing*, vol. 5, no. 6, pp. 287–302, 1993.
- [43] S.-L. Hung and H. Adeli, "A parallel genetic/neural network learning algorithm for mimd shared memory machines," *IEEE Transactions on neural networks*, vol. 5, no. 6, pp. 900– 909, 1994.
- [44] H. Adeli and S.-L. Hung, "An adaptive conjugate gradient learning algorithm for efficient training of neural networks," *Applied Mathematics and Computation*, vol. 62, no. 1, pp. 81–102, 1994.
- [45] H. Adeli and S.-L. Hung, Machine learning: neural networks, genetic algorithms, and fuzzy systems. John Wiley & Sons, Inc., 1994.
- [46] A. Olamat, P. Ozel, and S. Atasever, "Deep learning methods for multi-channel eeg-based emotion recognition," *Interna-*

tional Journal of Neural Systems, vol. 32, no. 05, p. 2250021, 2022.

- [47] Y. Dong, S. Patil, B. van Arem, and H. Farah, "A hybrid spatial-temporal deep learning architecture for lane detection," *Computer-Aided Civil and Infrastructure Engineering*, vol. 38, no. 1, pp. 67–86, 2023.
- [48] A. Hassanpour, M. Moradikia, H. Adeli, S. R. Khayami, and P. Shamsinejadbabaki, "A novel end-to-end deep learning scheme for classifying multi-class motor imagery electroencephalography signals," *Expert Systems*, vol. 36, no. 6, p. e12494, 2019.
- [49] L. Gan, J. W. Grizzle, R. M. Eustice, and M. Ghaffari, "Energy-based legged robots terrain traversability modeling via deep inverse reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 8807–8814, 2022.
- [50] N. Rudin, H. Kolvenbach, V. Tsounis, and M. Hutter, "Catlike jumping and landing of legged robots in low gravity using deep reinforcement learning," *IEEE Transactions on Robotics*, vol. 38, no. 1, pp. 317–328, 2021.
- [51] V. Tsounis, M. Alge, J. Lee, F. Farshidian, and M. Hutter, "Deepgait: Planning and control of quadrupedal gaits using deep reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 3699–3706, 2020.
- [52] A. S. Lele, Y. Fang, J. Ting, and A. Raychowdhury, "Learning to walk: Bio-mimetic hexapod locomotion via reinforcementbased spiking central pattern generation," *IEEE Journal* on Emerging and Selected Topics in Circuits and Systems, vol. 10, no. 4, pp. 536–545, 2020.
- [53] B. Acosta, W. Yang, and M. Posa, "Validating robotics simulators on real-world impacts," *IEEE Robotics and Automation Letters*, vol. 7, no. 3, pp. 6471–6478, 2022.
- [54] B. Tidd, N. Hudson, A. Cosgun, and J. Leitner, "Learning setup policies: Reliable transition between locomotion behaviours," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 11 958–11 965, 2022.
- [55] B. Sharma, B. M. Pillai, and J. Suthakorn, "Biomechanical trajectory optimization of human sit-to-stand motion with stochastic motion planning framework," *IEEE Transactions* on Medical Robotics and Bionics, vol. 4, no. 4, pp. 1022– 1033, 2022.
- [56] C. Yu and A. Rosendo, "Multi-modal legged locomotion framework with automated residual reinforcement learning," *IEEE Robotics and Automation Letters*, vol. 7, no. 4, pp. 10312–10319, 2022.
- [57] J. Ahn, J. Lee, and L. Sentis, "Data-efficient and safe learning for humanoid locomotion aided by a dynamic balancing model," *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4376–4383, 2020.
- [58] C. Huang, G. Wang, Z. Zhou, R. Zhang, and L. Lin, "Rewardadaptive reinforcement learning: Dynamic policy gradient optimization for bipedal locomotion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 45, no. 6, pp. 7686–7695, 2022.
- [59] J. Urdiales, D. Martín, and J. M. Armingol, "An improved deep learning architecture for multi-object tracking systems," *Integrated Computer-Aided Engineering*, vol. 30, no. 2, pp.

121–134, 2023.

- [60] E. Macias-Garcia, D. G. Pérez, J. Medrano-Hermosillo, and E. Bayro-Corrochano, "Multi-stage deep learning perception system for mobile robots," *Integrated Computer-Aided Engineering*, vol. 28, no. 2, pp. 191–205, 2021.
- [61] J. Gasienica-Józkowy, M. Knapik, and B. Cyganek, "An ensemble deep learning method with optimized weights for drone-based water rescue and surveillance," *Integrated Computer-Aided Engineering*, vol. 28, no. 3, pp. 221–235, 2021.
- [62] M. J. Gómez-Silva, A. de la Escalera, and J. M. Armingol, "Back-propagation of the mahalanobis istance through a deep triplet learning model for person re-identification," *Integrated Computer-Aided Engineering*, vol. 28, no. 3, pp. 277–294, 2021.
- [63] C. Tao, J. Xue, Z. Zhang, and Z. Gao, "Parallel deep reinforcement learning method for gait control of biped robot," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 69, no. 6, pp. 2802–2806, 2022.
- [64] J. Xu, Z. Hou, W. Wang, B. Xu, K. Zhang, and K. Chen, "Feedback deep deterministic policy gradient with fuzzy reward for robotic multiple peg-in-hole assembly tasks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 3, pp. 1658–1667, 2018.
- [65] G. Hao, Z. Fu, X. Feng, Z. Gong, P. Chen, D. Wang, W. Wang, and Y. Si, "A deep deterministic policy gradient approach for vehicle speed tracking control with a robotic driver," *IEEE Transactions on Automation Science and Engineering*, vol. 19, no. 3, pp. 2514–2525, 2021.
- [66] X. Gao, L. Yan, Z. Li, G. Wang, and I.-M. Chen, "Improved deep deterministic policy gradient for dynamic obstacle avoidance of mobile robot," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 53, no. 6, pp. 3675–3682, 2023.
- [67] F. Ying, H. Liu, R. Jiang, and X. Yin, "Trajectory generation for multiprocess robotic tasks based on nested dual-memory deep deterministic policy gradient," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 6, pp. 4643–4653, 2022.
- [68] L. Chang, L. Shan, W. Zhang, and Y. Dai, "Hierarchical multi-robot navigation and formation in unknown environments via deep reinforcement learning and distributed optimization," *Robotics and Computer-Integrated Manufacturing*, vol. 83, p. 102570, 2023.
- [69] B. Bordel, R. Alcarria, D. Martín, and D. S. de Rivera, "An agent-based method for trust graph calculation in resource constrained environments," *Integrated Computer-Aided En*gineering, vol. 27, no. 1, pp. 37–56, 2020.
- [70] A. Canito, A. Nobre, J. Neves, J. M. Corchado, and G. Marreiros, "Using sensor data to detect time-constraints in ontology evolution," *Integrated Computer-Aided Engineering*, vol. 30, no. 2, pp. 169–184, 2023.
- [71] K. Pfeiffer, A. Escande, P. Gergondet, and A. Kheddar, "The hierarchical newton's method for numerically stable prioritized dynamic control," *IEEE Transactions on Control Systems Technology*, no. 99, pp. 1–14, 2023.
- [72] W.-S. Yang, W.-C. Lu, and P.-C. Lin, "Legged robot running using a physics-data hybrid motion template," *IEEE Transac*-

tions on Robotics, vol. 37, no. 5, pp. 1680-1695, 2021.

- [73] S. Wang, Z. Chen, J. Li, J. Wang, J. Li, and J. Zhao, "Flexible motion framework of the six wheel-legged robot: Experimental results," *IEEE/ASME Transactions on Mechatronics*, vol. 27, no. 4, pp. 2246–2257, 2021.
- [74] X. Xing, K. Maqsood, C. Zeng, C. Yang, S. Yuan, and Y. Li, "Dynamic motion primitives-based trajectory learning for physical human-robot interaction force control," *IEEE Transactions on Industrial Informatics*, 2023.
- [75] S. Kolathaya, "Local stability of pd controlled bipedal walking robots," *Automatica*, vol. 114, pp. 108 841–108 857, 2020.
- [76] M. S. Shahabi, A. Shalbaf, B. Nobakhsh, R. Rostami, and R. Kazemi, "Attention-based convolutional recurrent deep neural networks for the prediction of response to repetitive transcranial magnetic stimulation for major depressive disorder," *International Journal of Neural Systems*, vol. 33, no. 2, pp. 2 350 007:1–2 350 007:13, 2023.
- [77] T. Hu, L. Xie, L. Zhang, G. Li, and Z. Yi, "Deep multimodal neural network based on data-feature fusion for patientspecific quality assurance," *International Journal of Neural Systems*, vol. 32, no. 1, pp. 2 150 055:1–2 150 055:14, 2022.
- [78] Z. Yu, L. Albera, R. L. B. Jeannes, A. Kachenoura, A. Karfoul, C. Yang, and H. Shu, "Epileptic seizure prediction using deep neural networks via transfer learning and multi-feature fusion," *International Journal of Neural Systems*, vol. 32, no. 7, pp. 2 250 032:1–2 250 032:19, 2022.
- [79] H. A. Ardakani, M. Taghizadeh, and F. Shayegh, "Diagnosis of autism disorder based on deep network trained by augmented EEG signals," *International Journal of Neural Systems*, vol. 32, no. 11, pp. 2 250 046:1–2 250 046:16, 2022.
- [80] M. H. Rafiei and H. Adeli, "A new neural dynamic classification algorithm," *IEEE transactions on neural networks and learning systems*, vol. 28, no. 12, pp. 3074–3083, 2017.
- [81] D. R. Pereira, M. A. Piteri, A. N. Souza, J. P. Papa, and H. Adeli, "Fema: A finite element machine for fast learning," *Neural Computing and Applications*, vol. 32, pp. 6393–6404, 2020.
- [82] K. M. R. Alam, N. Siddique, and H. Adeli, "A dynamic ensemble learning algorithm for neural networks," *Neural Computing and Applications*, vol. 32, pp. 8675–8690, 2020.
- [83] M. H. Rafiei, L. V. Gauthier, H. Adeli, and D. Takabi, "Self-supervised learning for electroencephalography," *IEEE Transactions on Neural Networks and Learning Systems*, 2022.
- [84] S. Kolathaya, "Pd tracking for a class of underactuated robotic systems with kinetic symmetry," *IEEE Control Systems Letters*, vol. 5, no. 3, pp. 809–814, 2020.
- [85] J. Chen, D. Sun, M. Zhao, Y. Li, and Z. Liu, "A new lane keeping method based on human-simulated intelligent control," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 7, pp. 7058–7069, 2021.
- [86] J. Chen, D. Sun, Y. Li, M. Zhao, W. Liu, and S. Jin, "Humanmachine cooperative scheme for car-following control of the connected and automated vehicles," *Physica A: Statistical Mechanics and its Applications*, vol. 573, p. 125949, 2021.