machines

Article

Deep Reinforcement Learning for Model Predictive Controller Based on Disturbed Single Rigid Body Model of Biped Robots

Landong Hou [1], Bin Li [2,*], Weilong Liu [2], Yiming Xu [1], Shuhui Yang [2] and Xuewen Rong [3]

[1] School of Electrical Engineering and Automation, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
[2] School of Mathematics and Statistics, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
[3] School of Control Science and Engineering, Shandong University, Jinan 250100, China
* Correspondence: ribbenlee@126.com; Tel.: +86-1369-862-2129

Abstract: This paper modifies the single rigid body (SRB) model, and considers the swinging leg as the disturbances to the centroid acceleration and rotational acceleration of the SRB model. This paper proposes deep reinforcement learning (DRL)-based model predictive control (MPC) to resist the disturbances of the swinging leg. The DRL predicts the swing leg disturbances, and then MPC gives the optimal ground reaction forces according to the predicted disturbances. We use the proximal policy optimization (PPO) algorithm among the DRL methods since it is a very stable and widely applicable algorithm. It is an on-policy algorithm based on the actor–critic framework. The simulation results show that the improved SRB model and the PPO-based MPC method can accurately predict the disturbances of the swinging leg to the SRB model and resist the disturbance, making the locomotion more robust.

Keywords: biped robots; single rigid body; model predictive control; deep reinforcement learning

Citation: Hou, L.; Li, B.; Liu, W.; Xu, Y.; Yang, S.; Rong, X. Deep Reinforcement Learning for Model Predictive Controller Based on Disturbed Single Rigid Body Model of Biped Robots. *Machines* **2022**, *10*, 975. https://doi.org/10.3390/machines10110975

Academic Editor: Manuel F. Silva

Received: 19 September 2022
Accepted: 22 October 2022
Published: 26 October 2022

1. Introduction

In this paper, deep reinforcement learning (DRL) is used to predict the disturbances of the swinging leg to the single rigid body (SRB) model, and the SRB-based model predictive control (MPC) method is transplanted to the biped robots with a non-negligible leg mass.

Compared with other types of robots, legged robots have huge application value and development prospects. At present, quadruped robots and biped robots are the research hotspots in the field of legged robots. Due to the complex nonlinear dynamics and higher degrees of freedom of biped robots, it is a challenging task to realize the stable walking of biped robots [1]. Compared with quadruped robots, it is difficult to achieve static stability with biped robots due to their mechanical structure design. Since the rectangular foot area of biped robots is very small, some biped robots even have linear feet. This results in a small or even a non-existent support field for biped robots during static standing and locomotion. From the point of view of stability analysis, the biped robots do not have the condition of static stability, but only have the condition of dynamic stability. This means that bipedal robots can only stabilize themselves during locomotion. Therefore, the design of the locomotion controller of biped robots is much more difficult than that of quadruped robots.

At present, there are two main control methods for legged robots, namely model-based control methods and model-free control methods. DRL is the most dominant of the model-free methods. Currently, in the field of legged robots, proximal policy optimization [2] (PPO) and deep deterministic policy gradient [3] (DDPG) are the two most commonly used DRL methods. The DRL methods successfully realize the navigation of mobile robots [4] and the motion control of manipulators [5]. The DRL methods avoid the complex modeling and parameter adjustment process. Through the guidance of different reward functions, the agent can learn different target strategies, which is a more flexible control method.

Recent studies have achieved many control goals on bipedal robots with the help of DRL methods, such as blindly climbing stairs [6], adapting to rough terrains [7], and carrying various dynamic loads [8].

Most DRL methods learn joint positions in joint space [9,10] and then implement joint torque control through a low-level proportional derivative (PD) controller. However, the dimensions of observation space and action space of such methods are large, which require a long training time. The model-based control method is what we usually call the traditional control method.

To design a model-based control method, the controlled object must be modeled first. In the field of legged robots, there are many ways to model robots. These models can be roughly divided into two categories: one is full-order models and the other is reduced-order models. The main difference between them is whether the dynamic model of the robot is simplified. The full order model is to model the legged robot as accurately as possible, preserving all quantifiable dynamic properties. In physics, the full-order model of a legged robot is one of a multi-rigid-body floating-base dynamic model. The Newton–Euler method and the Lagrange method are two commonly used modeling methods for multi-rigid-body floating-base dynamic models.

Compared with the full-order model, the modeling process of the reduced-order model is much simpler. In the field of legged robots, the most classic reduced order model is the linear inverted pendulum (LIP) model [11]. The LIP model simplifies the robot into a system consisting of a mass point and two massless links with freely varying lengths attached to it. The model assumes that the height of the center of mass remains unchanged during the movement process, which degenerates the complex nonlinear dynamic equation into a linear equation and greatly reduces the design difficulty of the controller. The simplicity and effectiveness of LIP make it widely used in the field of legged robots. Subsequently, many improved models based on LIP appeared, such as the same classical spring-loaded inverted pendulum (SLIP) model [12]. Another commonly used model in the field of biped robots is the five-link model. The LIP model is mainly used to plan the landing point and gait cycle of the robot, and the five-link model can be used to directly solve the joint torque. The torso and legs of the biped robot are simplified into five mass-concentrated connecting links. As an approximate simplified model of the full-order models, the five-link model can also be modeled by the Newton–Euler method and Lagrange method mentioned above, and the modeling process is relatively simple. A recent study on quadruped robots [13] proposed a reduced-order model called the SRB model. Unlike the LIP model and the five-link model, the SRB model contains only one rigid body that can move freely. Relative to the point and the link, the SRB contains the posture information of the robot, so the SRB model is a more accurate simplified model than the LIP and the five-link model. The hybrid zero dynamics (HZD) is a feedback control method based on virtual constraints. It is commonly used in the control of full-order models and five-link models [14]. Recent research has improved the Cassie robot's adaptability to rough terrains with the help of HZD [15]. There are many control methods based on the LIP model, such as capture point (CP) control [16], zero moment point (ZMP) control [17], linear quadratic regulator (LQR), and other optimal control methods [18]. A recent study on bipedal robots [19] applied the SRB-based MPC method to bipedal robots. However, the popularization of the SRB-based MPC in the field of biped robots faces a major challenge. The SRB model does not consider the influence of the leg mass on the overall motion of the robot, which is a very reasonable assumption for a quadruped robot whose leg mass accounts for about 10%. At present, the mass of the legs of most biped robots accounts for a large proportion, and the influence of the mass of the legs on the overall motion cannot be ignored. Recent studies [20–22] have combined the whole body control (WBC) based on floating-base dynamics with the SRB-based MPC and achieved good control results. However, such methods are complex and computationally intensive. Recent studies have also attempted to combine model-based and model-free approaches, such as learning foothold locations in the task space [23] and combining DRL with HZD to achieve higher-level control objectives [24].

Based on the above reasons, we propose an MPC method based on DRL. In this paper, we use the PPO algorithm among DRL methods to learn the prediction policy of the disturbances of the swinging leg. Since it is a very popular algorithm, it is often used as the baseline method. Its effect may not be the best, but it is a very stable and widely applicable algorithm. It is an on-policy algorithm based on the actor–critic framework. Videos of the experimental results can be found at https://github.com/houshang1991/dsrb-mpc.git (accessed on 7 October 2020). The main contributions of this paper are as follows:

1. The SRB-based MPC method generally assumes that the robot's leg mass has a negligible effect on its locomotion. However, the mass of the legs of most biped robots accounts for a large proportion of the total mass, which will disturb the locomotion of the SRB model. We describe the disturbances of the swinging leg as the centroid acceleration disturbance and the rotational acceleration disturbance. Furthermore, we use the PPO algorithm to train a policy to accurately predict the disturbances of the swinging leg.
2. In this paper, we improve the SRB-based MPC method by adding the disturbances of the swing leg to the SRB model. This method can give the true optimal ground reaction forces (GRFs), enabling the biped robot to resist the disturbances of the swinging leg while tracking the desired forward velocity.
3. In this paper, we verify the improved SRB-based MPC method through three simulation experiments. Experiments show that the disturbances of the swinging leg can be accurately predicted and resisted. We expand the scope of application of the SRB-based MPC method, and provide a robust control method for biped robots with non-negligible leg mass.

2. Control Architecture

The control method proposed in this paper consists of four parts, as shown in Figure 1. A finite state machine (FSM) is responsible for generating a walking gait pattern. The swing leg controller is responsible for solving the joint torque of the swing leg. The stance leg controller uses the MPC based on a modified SRB model to solve for the optimal GRFs. The trained policy is responsible for predicting the disturbances caused by the swinging leg to the SRB model. The structure of the biped robot used in this paper is shown in Figure 2. This section mainly introduces the first three parts.

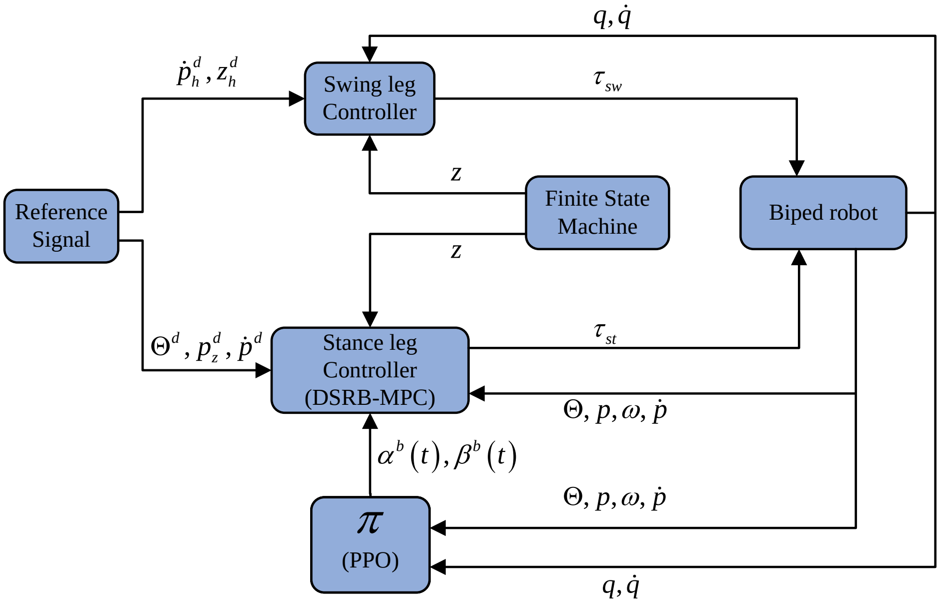


Figure 1. The block diagram of the control architecture. The stance leg controller and the policy run at 80 Hz, and the swing leg controller runs at 230 Hz. Reference signal includes desired torso Euler

angles, centroid velocity and centroid height, which are Θ^d, $\dot{p}^d$ and p_z^d, respectively. z_h^d represents the desired height of the hip joint of the swinging leg, which can be calculated from the above reference signals. $\dot{p}_h^d$ represents the desired velocity of the hip joint of the swinging leg in Cartesian space, which can also be calculated from the above reference signals. Θ and ω represent the actual torso Euler angles and angular velocity. p and $\dot{p}$ represent the actual centroid position and velocity. q and $\dot{q}$ represent the position and velocity of the leg joints in joint space, respectively. z indicates the completion percentage of swing or supporting. τ_{sw} and τ_{st} represent the swinging leg joints torques and the stance leg joints torques. $\alpha^b(t)$ and $\beta^b(t)$ represent the disturbances of the swinging leg to the centroid acceleration and rotational acceleration expressed in body frame. π is the disturbances prediction policy obtained by the PPO algorithm.

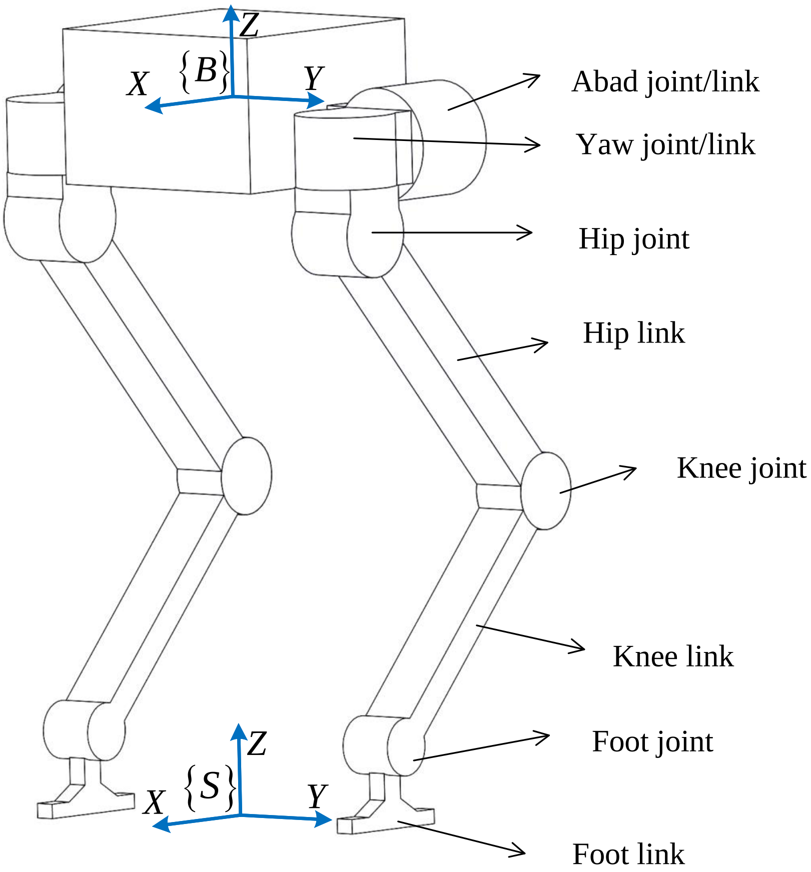


Figure 2. The structure of the biped robot. $\{B\}$ is the body frame, $\{S\}$ is the inertia frame.

2.1. Finite State Machine

The FSM generates a walking gait pattern based on the fixed swing duration and stance duration. It gives the time phase (swing phase or stance phase) that each leg is in at the current moment and the percentage $z \in [0, 1]$ of completion of the current time phase. The swing phase and stance phase of each leg accounted for 40% and 60% of the entire gait cycle, respectively, of which the double stance phase accounted for 10%. The walking gait pattern is shown in Figure 3. In this paper, the swing phase duration T_{sw} is equal to 0.12 s, and the stance phase duration T_{st} is equal to 0.18 s.

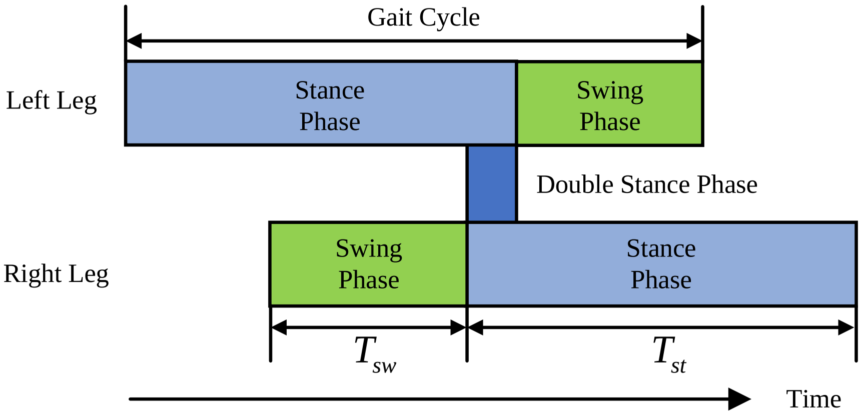


Figure 3. The walking gait pattern generated by the FSM.

2.2. *Swing Leg Controller*

The block diagram of the swing leg controller is shown in Figure 4. The swinging leg controller firstly solves the desired foothold point $\underline{p}_f^d$. Then it calculates the desired foot position p_f^d according to $\underline{p}_f^d$. Then it calculates the desired joint position q^d and velocity $\dot{q}^d$ according to the inverse kinematics of the leg. It finally calculates the joint torques τ_{sw} through a PD controller in the joint space. To alleviate the impact on the body when the swinging leg touches the ground, this paper applies three touchdown policies to ensure the stability of the robot's motion.

The touchdown policy 1 is to imitate the habit of human walking and increase the lateral stability of the biped robot. It adjusts the angle of the yaw joint in real time, making the toe slightly outward. It works from the beginning to the end of the swing phase, tracking a set initial yaw joint angle in real time. In the simulation environment, the collision between the foot of the swinging leg and the ground is the elastic collision. The touchdown policy 2 sets the desired speed of each joint to 0 so that the command speed output by the PD controller is as small as possible. It can reduce the impact of elastic collision on the motion state of the robot. It works just before the foot touches the ground until the end of the swing phase. Before the foot of the swinging leg is fully in contact with the ground, the yaw joint torque output by the PD controller may cause the front or rear end of the foot to contact the ground early. As a result, unexpected elastic collisions occur, and even relative sliding between the foot and the ground occurs. The above two situations will cause significant disturbances to the motion state of the robot. The touchdown policy 3 also starts working near the end of the swing phase, and by forcing the yaw joint torque to zero, the above-mentioned accidents can be avoided to the greatest extent possible. The above three touchdown policies are run synchronously, but the start time is different, and the specific situations to be dealt with are different. However, their ultimate goal is to improve the stability of the robot's locomotion.

In foothold planning, we use the following formula to calculate the real-time desired foothold position:

$$\underline{p}_f^d = \underline{p}_h + \tfrac{T_{st}}{2}\dot{\underline{p}}_h^d + \sqrt{\tfrac{z_h^d}{g}}\left(\dot{\underline{p}}_h - \dot{\underline{p}}_h^d\right), \tag{1}$$

where $\underline{p}_f^d$ is the desired foot landing point on the horizontal ground represented in $\{S\}$; $\underline{p}_h$ is the projection of the actual position of the hip joint on the horizontal ground represented in $\{S\}$; $\dot{\underline{p}}_h^d$ is the projection of the desired speed of the hip on the horizontal ground represented in $\{S\}$; $\dot{\underline{p}}_h$ is the projection of the actual speed of the hip on the horizontal ground represented in $\{S\}$; z_h^d is the desired height of the hip from the ground; T_{st} is the duration of the swing phase; and g is the acceleration of gravity.

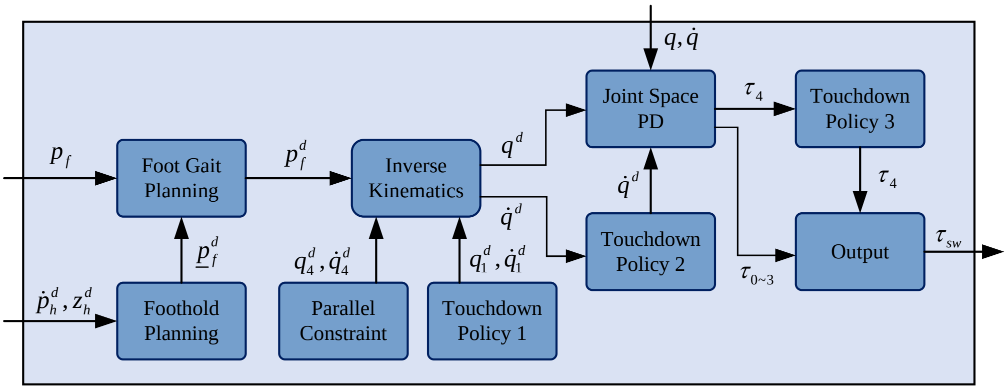


Figure 4. Block diagram of the swing leg controller. p_f represents the actual foot position. p_f^d represents the desired foot position. $\underline{p}_f^d$ represents the projection of p_f^d on the horizontal ground. q and $\dot{q}$ represent the actual position and velocity of the swinging leg joints in joint space. q^d and $\dot{q}^d$ represent the desired position and velocity of the swinging leg joints in joint space. τ represents the joint torque of the swinging leg. Subscripts 0, 1, 2, 3, and 4 refer to abad, yaw, hip, knee, and foot joints, respectively. $\dot{p}_h^d$ represents the desired velocity of the hip joint of the swinging leg in Cartesian space. z_h^d represents the desired height of the hip joint of the swinging leg.

In foot-gait planning, the desired foot position in the joint space is generated by fitting a 6th-order Bezier curve. Note that the joint positions and velocities mentioned below are representations in the joint space. To imitate the toe outstretching behavior of human walking and enhance the stability of dynamic walking, touchdown policy 1 adjusts the desired position and desired speed of the yaw joint of the biped robot in real-time. For the foot to be in full contact with the ground at the end of the swing phase, we add a horizontal constraint with the foot parallel to the ground. According to this constraint, the desired position and desired velocity of the foot joint can be solved.

Except for the yaw joint and the foot joint, each leg of the biped robot has 3 degrees of freedom, and the inverse kinematics happens to have a unique solution. According to the desired foot joint position, the desired position and velocity of the abad, hip, and knee joints can be solved. To reduce the impact on the ground, when $z \geq 0.85$, touchdown policy 2 sets the desired velocity of all joints of the swinging leg to 0 rad/s. Touchdown policy 2 further improves the stability of the biped robot when the leg transitions from the swing phase to the stance phase.

After solving for the desired positions and velocities of all joints, we filter the actual joint velocities using a first-order digital low-pass filter with a cutoff frequency lower than the operating frequency of the swing leg controller. Then, this paper uses a PD controller in joint space to calculate the torque of each joint according to the actual joint position errors and joint velocity errors:

$$\tau_{sw} = K_p\left(q^d - q\right) + K_d\left(\dot{q}^d - \dot{q}\right), \tag{2}$$

where q and $\dot{q}$ are the actual joint position and velocity vector, respectively; q^d and $\dot{q}^d$ are the desired joint position and velocity vector, respectively; and K_p and K_d are the error gain matrices of the PD controller, respectively.

When $z \geq 0.97$, touchdown policy 3 sets the foot joint torque to $0\ \mathrm{N}\cdot\mathrm{m}$. When the foot touches the ground, touchdown policy 3 can reduce the impact caused by the fluctuation of the foot joint torque.

2.3. Stance Leg Controller

The block diagram of the support leg controller is shown in Figure 5. We treat the body of the biped robot as the SRB model that can move and rotate freely in 3D space. We add external disturbances to the centroid acceleration and rotational acceleration of the model, and call the new model the disturbed single rigid body (DSRB) model. Unlike the original SRB-based MPC (SRB-MPC) method, we treat the effect of swinging the leg on the locomotion of the body as two predictable bounded disturbances. Then, the optimal GRFs at the foot of the stance leg can be solved by the DSRB-based MPC (DSRB-MPC) method.

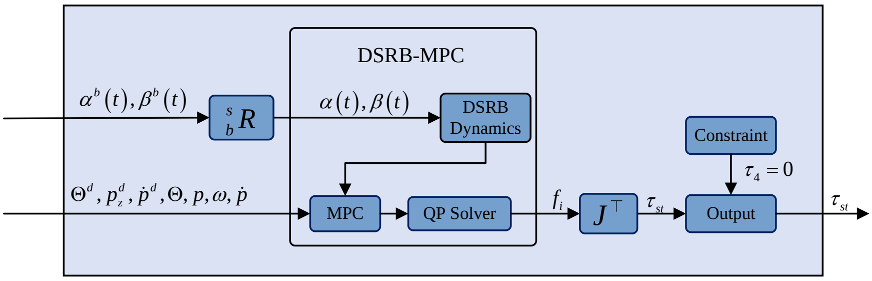


Figure 5. Block diagram of the stance leg controller. $\alpha^b(t)$ and $\beta^b(t)$ represent the disturbances to the centroid acceleration and rotational acceleration expressed in $\{B\}$, whereas $\alpha(t)$ and $\beta(t)$ express in $\{S\}$. Θ^d represents the desired Euler angles, whereas Θ represents the actual value. $\dot{p}^d$ represents the desired centroid velocity, whereas $\dot{p}$ represents the actual value. ω represent the actual angular velocity. p_z^d represents desired centroid height. p represents the actual centroid position. f_i is the GRF. τ_4 represents the torque of foot joint. τ_{st} represents the torques of the stance leg. ${}^s_bR$ is the rotation matrix which transforms from $\{B\}$ to $\{S\}$. J is the foot joint Jacobian.

The approximate linear dynamics of the DSRB model are as follows:

$$\ddot{p} = \frac{\sum_{i=1}^n f_i}{m} - g + \alpha(t), \tag{3}$$

$$\dot{\omega} \approx I^{-1}\left(\sum_{i=1}^n r_i \times f_i\right) + \beta(t), \tag{4}$$

where p is the position of the center of mass of the body; $\ddot{p}$ is the acceleration of the center of mass of the body; ω is the rotational angular velocity of the body; $\dot{\omega}$ is the rotational acceleration of the body; m is the mass of the body; f_i is the reaction force exerted by the ground on the center of mass of the body through the ith foot; r_i is the moment arm of f_i; I is the inertia tensor of the body; $\alpha(t)$ and $\beta(t)$ are the uncertain centroid acceleration and rotational acceleration disturbances imposed by the outside on the body expressed in $\{S\}$, respectively; and n is the number of legs of the robot.

Equations (3) and (4) can be rewritten as the following equation of state:

$$\begin{aligned}\frac{d}{dt}\begin{bmatrix}\Theta \\ p \\ \omega \\ \dot{p} \\ \beta(t) \\ \alpha(t)-g\end{bmatrix} = &\begin{bmatrix} 0_{3\times3} & 0_{3\times3} & {}^s_bR & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & E_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & E_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & E_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3} & 0_{3\times3}\end{bmatrix}\begin{bmatrix}\Theta \\ p \\ \omega \\ \dot{p} \\ \beta(t) \\ \alpha(t)-g\end{bmatrix} \\ &+ \begin{bmatrix} 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \\ I^{-1}[r_1]_\times & I^{-1}[r_2]_\times \\ \frac{1}{m}E_{3\times3} & \frac{1}{m}E_{3\times3} \\ 0_{3\times3} & 0_{3\times3} \\ 0_{3\times3} & 0_{3\times3}\end{bmatrix}\begin{bmatrix} f_1 \\ f_2 \end{bmatrix},\end{aligned} \tag{5}$$

where Θ is the Euler angles in ZYX order, representing the orientation of the body; ${}^{s}_{b}R$ is the rotation matrix, which transforms from $\{B\}$ to $\{S\}$; $0_{3\times3}$ is the zero matrix; $E_{3\times3}$ is the identity matrix; and $[d]_\times$ is the skew-symmetric matrix generated by the vector d.

We take the discrete form of the state Equation (5) as the equality constraint, the friction cone constraints of the GRFs as the inequality constraint, and the quadratic form of the state error and input as the objective function. The problem of solving the optimal GRFs can be written in the following standard MPC form:

$$\min_{x,u} \quad \textstyle\sum_{i=0}^{k-1} \left\| x_{i+1} - x_{i+1,ref} \right\|_Q + \|u_i\|_R, \tag{6}$$

$$s.t. \quad x_{i+1} = A_{di}x_i + B_{di}u_i, i = 0 \ldots k-1, \tag{7}$$

$$\underline{c} \le C_{di}u_i \le \overline{c}, i = 0 \ldots k-1, \tag{8}$$

where k is the horizon length; $x_i = \left[\Theta^\top, p^\top, \omega^\top, \dot{p}^\top, \beta^\top, (\alpha - g)^\top\right]^\top$ is the predicted state of the system at the ith moment; $x_{i+1,ref}$ is the reference state of the system at the next time after the ith moment; $u_i = \left[f_1^\top, f_2^\top\right]^\top$ is the input of the system at the ith moment; $x_{i+1} = A_{di}x_i + B_{di}u_i$ is the discrete form of Equation (5) at the ith moment; $\underline{c} \le C_{di}u_i \le \overline{c}, i = 0 \ldots k-1$ is the friction cone constraints at the ith moment; and Q and R are diagonal positive semi-definite matrices of weights.

Equation (7) can be rewritten as the following compact form:

$$X = A_{qp}x_0 + B_{qp}U, \tag{9}$$

where $X = \left[x_1^\top \ldots x_k^\top\right]^\top$ is the state trajectory of the system in the horizon; $U = \left[u_0^\top \ldots u_{k-1}^\top\right]^\top$ is the input sequence of the system in the horizon.

Finally, we put Equation (9) into Formula (6), and the standard MPC problem can be simplified as the following standard QP problem:

$$\min_{U} \quad \frac{1}{2}U^\top\left(B_{qp}^\top Q_{qp} B_{qp} + R_{qp}\right)U + U^\top\left[B_{qp}^\top Q_{qp}\left(A_{qp}x_0 - X_{ref}\right)\right], \tag{10}$$

$$s.t. \quad \underline{C} \le C_{qp}U \le \overline{C}, \tag{11}$$

where Q_{qp} is a block diagonal matrix composed of Q; R_{qp} is a block diagonal matrix composed of R; C_{qp} is a block diagonal matrix composed of C_{di}; $\underline{C}$ is made up of stacks of $\underline{c}$; $\overline{C}$ is made up of stacks of $\overline{c}$; and $X_{ref} = \left[x_{1,ref}^\top \ldots x_{k,ref}^\top\right]^\top$ is the state reference trajectory of the system in the horizon.

Considering that the output torque of the actuator at the foot joint of most biped robots is small, we take the foot joint as a passive joint and constrain the output torque of this joint to $0\,\mathrm{N}\cdot\mathrm{m}$. Under the above assumptions, the biped robot cannot achieve static stability, but can only achieve dynamic stability, and its control difficulty increases. The output torque of each joint except the foot joint is given by

$$\tau_{st} = J^\top {}^{s}_{b}R^\top u_0, \tag{12}$$

where J is the foot joint Jacobian.

To improve stability, we also perform first-order low-pass filtering on the centroid velocity and rotation velocity of the body in the stance leg controller. The cutoff frequency of the filter is lower than the operating frequency of the stance leg controller.

3. Learning Prediction Policy for Disturbances

The disturbances of the swinging leg are related to the joint position q and joint velocities $\dot{q}$, which is a complex time-varying nonlinear function. It is difficult to accurately model it using mathematical analysis methods. Therefore, in this paper, we formulate the prediction of the disturbances of the swinging leg as a Markov decision process. Then we use the PPO method to train a policy to predict the disturbances at different times.

3.1. State Space and Action Space

The state variable s_t only contains the current state of the biped robot and does not contain any desired state variables:

$$s_t = \{\Theta, \omega, \dot{p}, q, \dot{q}, z_1^s, z_2^s\} \in \mathbb{R}^{31}, \tag{13}$$

where Θ is the Euler angles of the body; ω is the angular velocity of the body; $\dot{p}$ is the centroid velocity of the body; $q \in \mathbb{R}^{10}$ is the joint position vector of the biped robot; $\dot{q} \in \mathbb{R}^{10}$ is the joint velocity vector of the biped robot; and $z_i^s = -1 + z_i$ if the ith leg is in the swing phase, otherwise $z_i^s = z_i$, and $z_i^s \in [-1, 1]$.

Because the disturbances of the swinging leg are independent of p and the disturbances are relative to the body, we choose to learn a prediction policy π for the disturbances expressed in $\{B\}$. We take the disturbances $\alpha^b(t)$ and $\beta^b(t)$ of the swinging leg as actions. The dimension of the action space we choose is 6, which is smaller than the dimension of the joint space. We assume that the disturbances of the swinging leg are bounded, limiting the range of the components of $\alpha^b(t)$ and $\beta^b(t)$ to be between -5 and 5. Through ${}^s_bR$, we can find the disturbances expressed in $\{S\}$, $\alpha(t)$ and $\beta(t)$.

3.2. Reward Function

This paper combines reinforcement learning with the DSRB-MPC method, so the design of the reward function is very simple. When designing the reward function, we preferentially encourage the biped robot to keep the Euler angles of the body unchanged to avoid falling due to drastic changes in posture. Therefore, the Euler angle errors of the body have the largest weights in the reward function, followed by the height error. The reward function secondly encourages the biped robot to track the forward velocity on a horizontal plane, so the forward and lateral velocity errors are weighted less, and the vertical velocity error is the least weighted. The reward function is as follows:

$$\begin{aligned} r = 0.22\exp(-6|\Theta_x^e|) + 0.22\exp\left(-6\left|\Theta_y^e\right|\right) + 0.22\exp(-6|\Theta_z^e|) + 0.15\exp(-4|p_z^e|) \\ +0.07\exp(-2|\dot{p}_x^e|) + 0.07\exp\left(-2\left|\dot{p}_y^e\right|\right) + 0.04\exp(-|\dot{p}_z^e|), \end{aligned} \tag{14}$$

where Θ_x^e, Θ_y^e and Θ_z^e are the roll, pitch, and yaw angle errors of the body, respectively; p_z^e is the height error of the center of mass of the body; $\dot{p}_x^e$, $\dot{p}_y^e$ and $\dot{p}_z^e$ are the three velocity errors of the center of mass of the body in the x, y and z directions, respectively.

3.3. Prior Knowledge

To reduce the difficulty of training and shorten the training time, this paper only hopes that the biped robot can track a horizontal positive reference velocity while keeping the body posture unchanged. Furthermore, this paper introduces two prior pieces of knowledge.

1. First, we determine the parameters K_p, K_d, Q and R of the SRB-MPC method on the model 1 in Table 1 with negligible leg mass (the mass of the two legs of the model accounts for 6.7% of the total mass). To make the stance leg controller stable, even when $\dot{p}_x^e$ is large, we use three very small velocity target weights: $Q_x^{\dot{p}}$, $Q_y^{\dot{p}}$ and $Q_z^{\dot{p}}$ in Q. The stance leg controller will give priority to ensuring that the Euler angle errors of the body are the smallest and maintain the body posture. The stance leg controller then tracks the positive horizontal reference speed as closely as possible without falling. When training the prediction policy for the disturbances of the swinging leg, we use Model 2 in Table 1 and use the same parameters in the DSRB-MPC method as the SRB-MPC method. The target weights of the MPC are shown in Table 2.

2. To avoid frequent falls of the biped robot at the beginning of the simulation, we use two small forward desired accelerations, and the forward velocity reference is shown in Figure 6. At the same time, in order to learn the disturbance prediction policy at different speeds, the reference trajectory contains 5 steps of one-second step-like uniform motion, and the step speeds are 0.6 m/s, 0.7 m/s, 0.8 m/s, 0.9 m/s and 0.95 m/s respectively. The end of the reference trajectory is a uniform motion of 1 m/s. The reference trajectory includes uniform and acceleration motion at different speeds. The learning difficulty of the policy is from easy to difficult, which satisfies the learning rules.

Table 1. The mass parameters of two models.

Link Name	Model 1 (Massless Legs)	Model 2 (Mass Legs)
Body	15 kg	11.8 kg
AbAd	0.1 kg	1.2 kg
Yaw	0.1 kg	1.0 kg
Hip	0.1 kg	0.2 kg
Knee	0.1 kg	0.1 kg
Foot	0.1 kg	0.1 kg

Table 2. The target weights of the MPC.

Weight	Value	Weight	Value	Weight	Value
Q_x^{Θ}	55	Q_y^{Θ}	260	Q_z^{Θ}	380
Q_x^{p}	1	Q_y^{p}	0.01	Q_z^{p}	50
Q_x^{ω}	1×10^{-4}	Q_y^{ω}	1×10^{-5}	Q_z^{ω}	1×10^{-5}
$Q_x^{\dot{p}}$	10	$Q_y^{\dot{p}}$	1	$Q_z^{\dot{p}}$	1

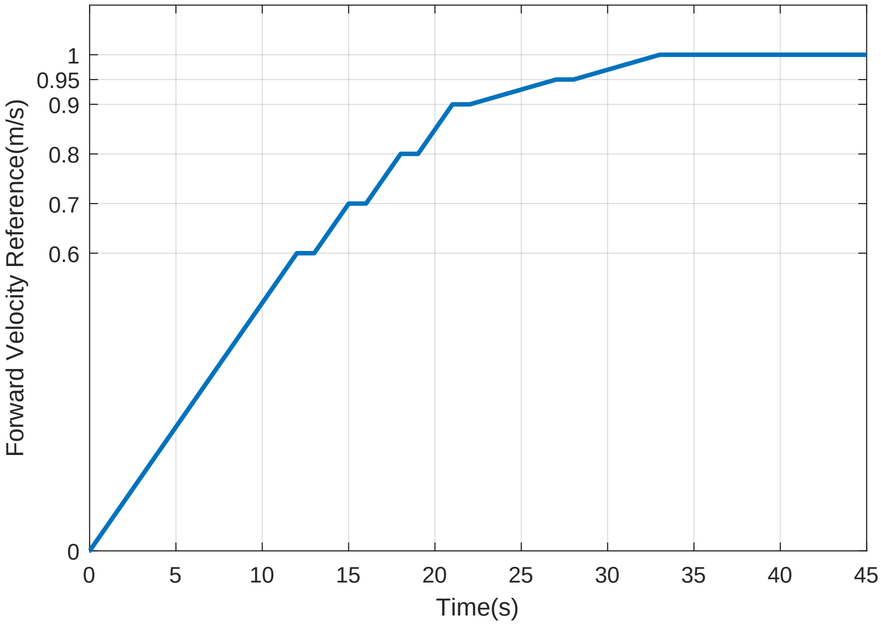


Figure 6. Forward velocity reference. Before the velocity reaches 0.9 m/s, the expected acceleration is 0.05 m/s^2; after the velocity reaches 0.9 m/s, the expected acceleration is 0.01 m/s^2.

3.4. Parameters of PPO Algorithm

We choose the PPO algorithm to train the disturbances prediction policy for the swinging leg. In PPO, the disturbances are output by a Gaussian distribution, and then the outputs are clipped to a valid value range. Moreover, this paper uses 10 key tricks to improve the performance of the PPO algorithm [25]. We use the generalized advantage estimator [26] (GAE) method to estimate the advantage in the PPO algorithm. We build two three-layer fully connected networks as actor and critic networks, with only 256 neurons in each layer. The other parameters of the PPO algorithm are the empirical values of the method. In order to speed up the training, we impose a limit on the maximum simulation step per episode. The maximum simulation step size is 11,000, which is about 45 s. In addition, we also add some restrictions on the errors; the maximum errors are shown in Table 3.

Table 3. The maximum errors of the training.

Maximum Errors	$\boldsymbol{\Theta_x^E}$	$\boldsymbol{\Theta_y^E}$	$\boldsymbol{\Theta_z^E}$	$\boldsymbol{\dot{p}_x^E}$	$\boldsymbol{\dot{p}_y^E}$	$\boldsymbol{p_z^E}$
Unit	rad	rad	rad	m/s	m/s	m
Value	1.4	1.4	1.4	1	3	0.3

4. Simulation and Experimental Results

In this work, we choose PyBullet as the physics simulation engine and use Gym to build the reinforcement learning environment. We choose the Blackbird robot as the platform for our simulation experiments. The mass parameters of the model 2 used in the simulation are shown in in Table 1, where it can be seen that the mass of the two legs accounts for 30.5% of the total mass. The forward reference velocity used is shown in Figure 6. After 5 million steps of learning, we obtained the target policy π for predicting the disturbance of the swinging leg, and the reward curve of the learning process is shown in Figure 7. The initial reward value fluctuates greatly, and after 1 million steps of learning, the reward value fluctuates less. After 4 million steps of learning, the reward value converges.

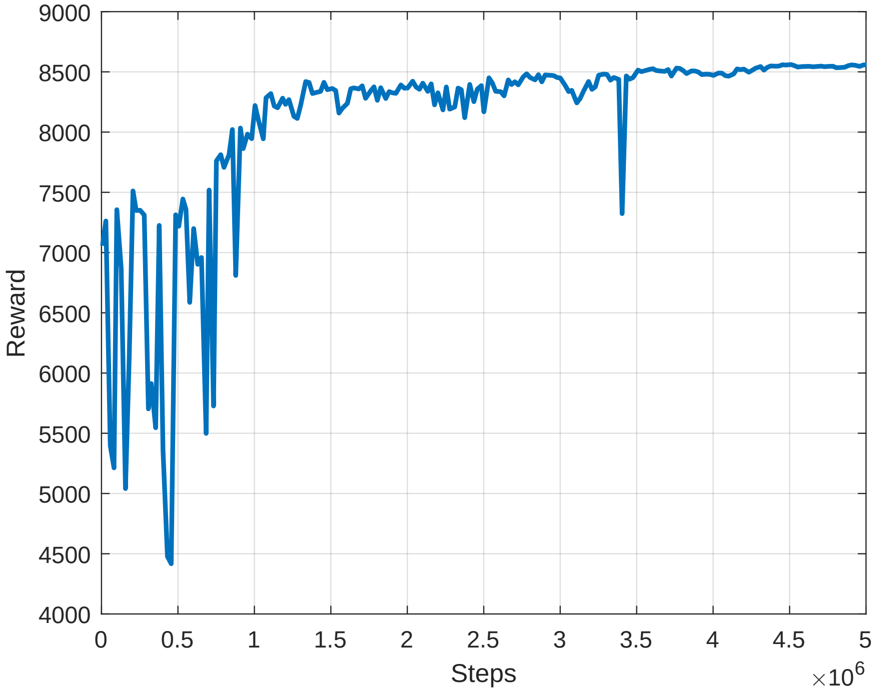
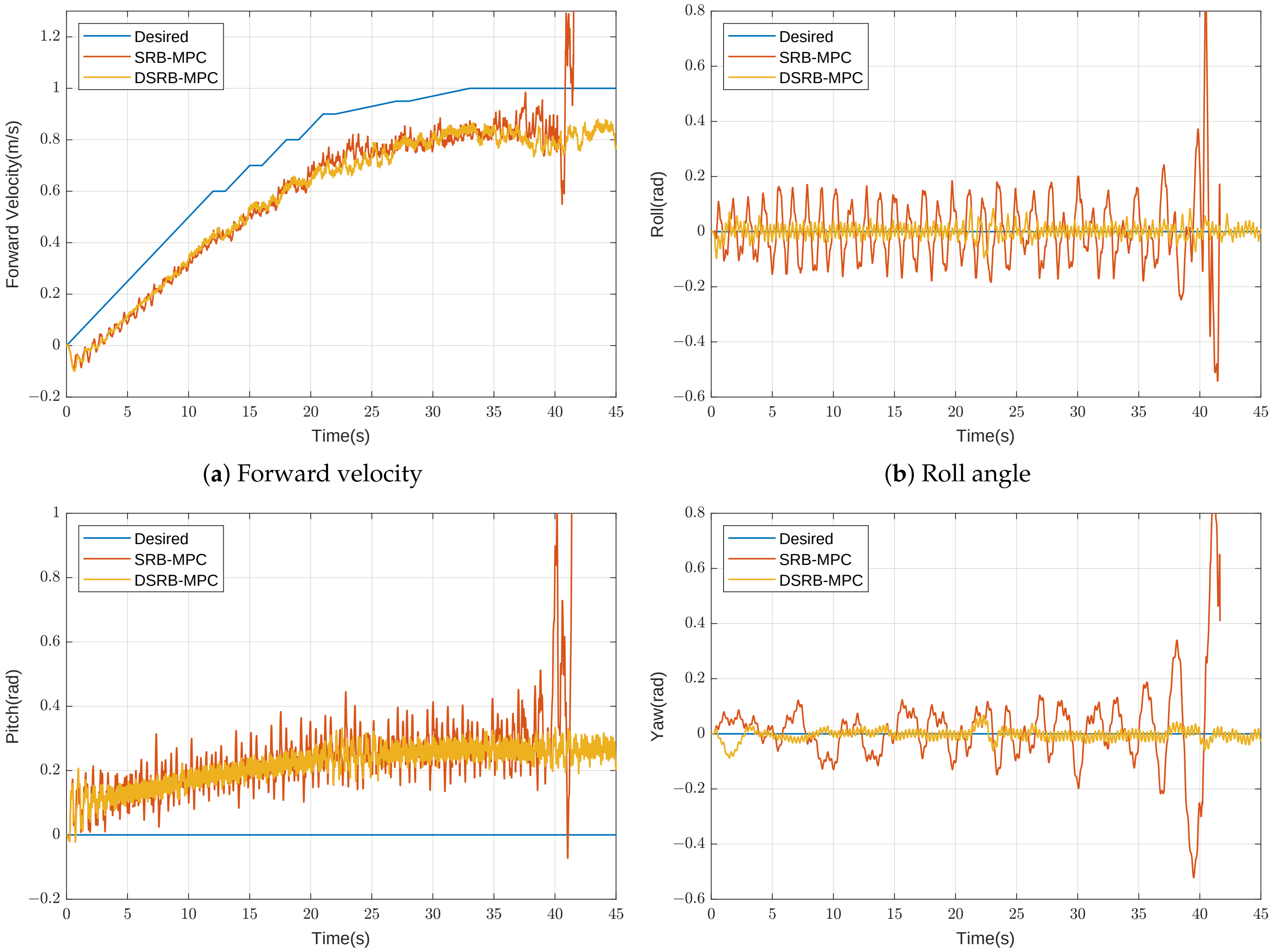


Figure 7. The reward of the learning process.

We use the trained target policy π as the external disturbances input of the DSRB-MPC method, and compare it with the SRB-MPC method. The results are shown in Figure 8. The results showed that the swinging leg has a larger effect on the Euler angles of the robot relative to the velocity of the center of mass. As the velocity of the center of mass increases, the SRB-MPC method cannot suppress the fluctuation of the Euler angles, especially the yaw angle. Due to the violent oscillation of Euler angles, the stance leg controller does not work properly. As a result, the center of mass velocity increases rapidly, and finally the robot falls at 40 s. It can be seen from the training results that the fluctuation of the Euler angles of the DSRB-MPC method is much smaller than that of the SRB-MPC method. The centroid velocity of the DSRB-MPC method is also slightly smoother than that of SRB-MPC method. It is further proved that the policy π learned by the PPO method can accurately predict the disturbances of the swinging leg to the biped robot during the forward locomotion. The disturbances consist of the centroid acceleration disturbance and the rotational acceleration disturbance.



(**a**) Forward velocity

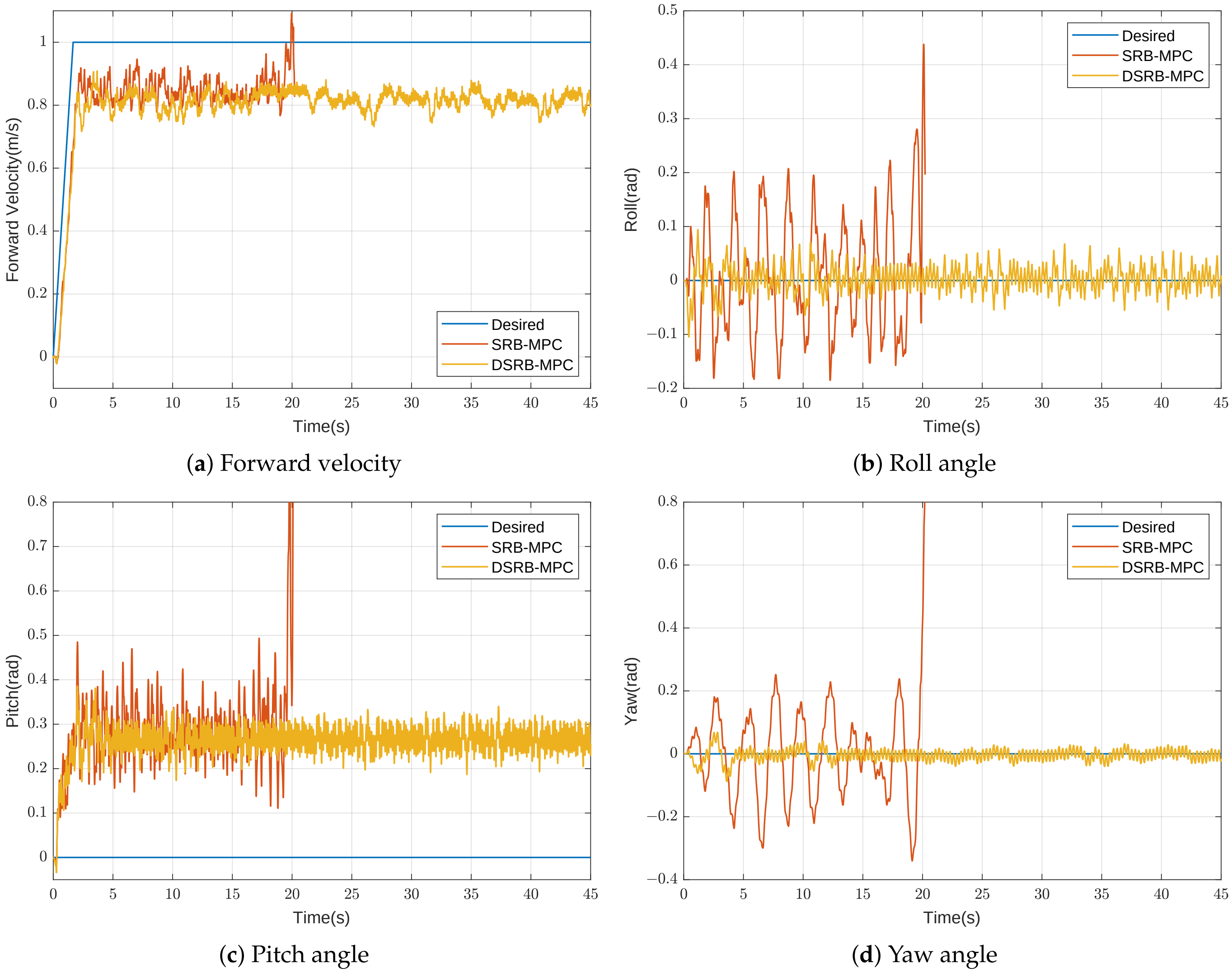
(**b**) Roll angle

(**c**) Pitch angle

(**d**) Yaw angle

Figure 8. The performance of the target policy π. (**a**) The command velocity and measured actual velocities of are presented. (**b**) The command roll angle and measured actual roll angles are presented. (**c**) The command pitch angle and measured actual pitch angles are presented. (**d**) The command yaw angle and measured actual yaw angles are presented. Note that all blue lines represent given reference signals. Red lines show the performance of SRB-MPC. Yellow lines show the performance of DSRB-MPC. The red line stops because the error of SRB-MPC exceeds the maximum error, the simulation stops at about 40 s. The reason for the instability of SRB-MPC is that the disturbances of the swinging leg are not considered in the modeling, and the modeling error is too large. In this situation, modeling errors for Θ and p are too large, and MPC cannot resist. However, DSRB-MPC can accurately model disturbances and resist them.

In order to verify the robustness of the DSRB-MPC under the higher acceleration, we tested the performance of the method under the extreme acceleration 0.6 $\mathrm{m/s}^2$, as shown in Figure 9. The SRB-MPC method falls at 20 s earlier than the previous experiment. From the performance of the SRB-MPC method, it can be seen that the fluctuation of the Euler angles is larger than that of the previous experiment, which indicates that the disturbances of the swinging leg are related to the commanded centroid acceleration. As the command acceleration increases, the centroid acceleration disturbance and the rotational acceleration disturbance are also larger. However, the centroid velocity and Euler angles fluctuations of the DSRM-MPC method are still small. The policy π still predicts the disturbances of the swinging leg accurately.



(**a**) Forward velocity

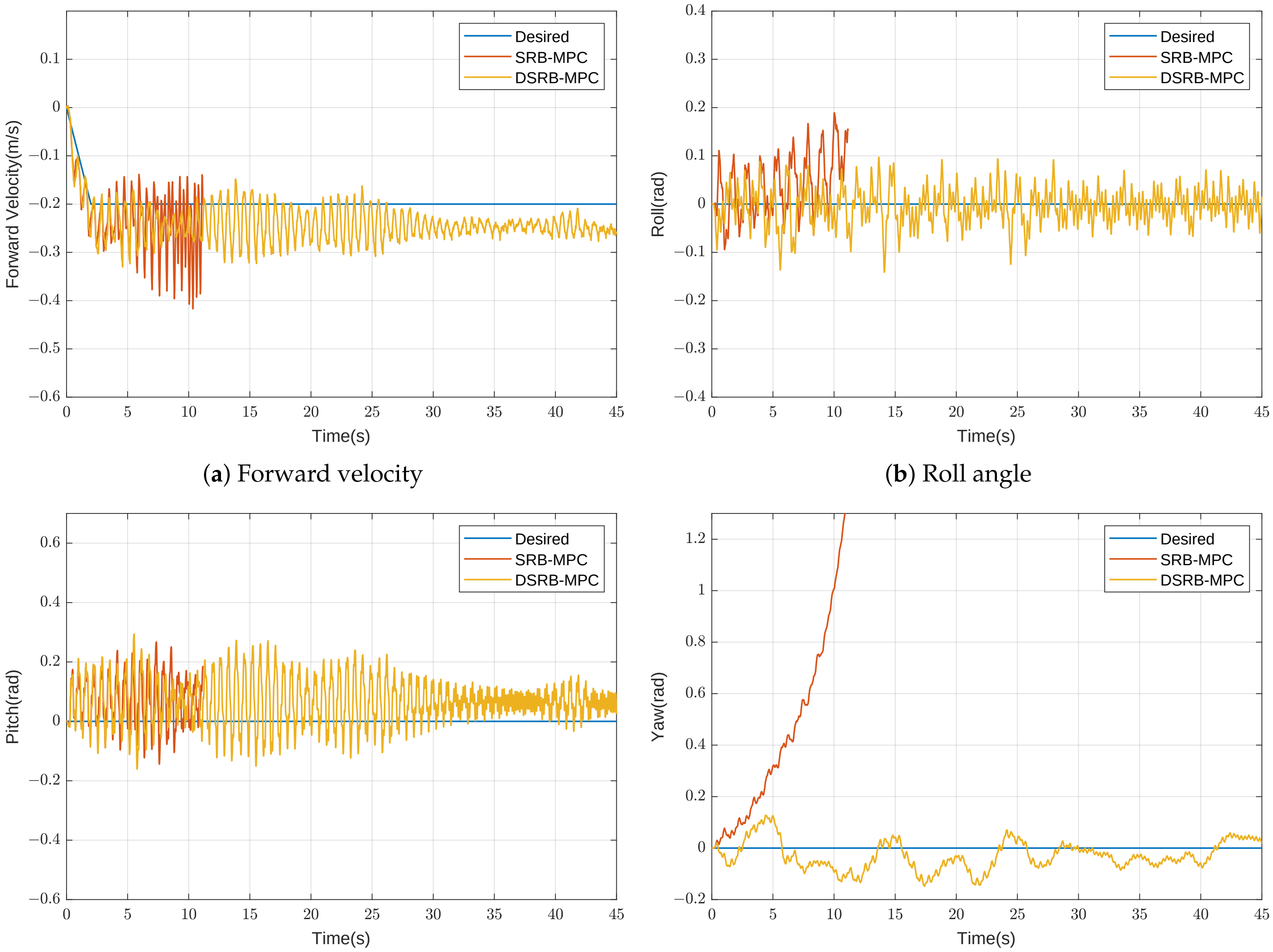
(**b**) Roll angle

(**c**) Pitch angle

(**d**) Yaw angle

Figure 9. The performance of the DSRB-MPC under the extreme acceleration. (**a**) The command velocity and measured actual velocities of are presented. (**b**) The command roll angle and measured actual roll angles are presented. (**c**) The command pitch angle and measured actual pitch angles are presented. (**d**) The command yaw angle and measured actual yaw angles are presented. Note that all blue lines represent given reference signals. Red lines show the performance of SRB-MPC. Yellow lines show the performance of DSRB-MPC. The red line stops because the error of SRB-MPC exceeds the maximum error, the simulation stops at about 20 s. The reason for being unstable of SRB-MPC is that the disturbances of the swinging leg are not considered in the modeling, and the modeling error is too large. In this situation, modeling error for Θ is too large, and MPC cannot resist. However, DSRB-MPC can accurately model disturbances and resist them.

Through experiments, we found that the DSRB-MPC method can also control the robot to move in the opposite direction at a smaller speed about -0.2 m/s as shown in Figure 10. The SRB-MPC method falls at 10 s, earlier than the fall time in the forward locomotion experiment. From the results of the SRB-MPC method, it can be seen that the opposite locomotion is more difficult than the forward. The specific phenomenon is that during the locomotion of the robot, the yaw angle continues to increase, resulting in self-rotation. Although the fluctuations of the Euler angles and the centroid velocity of the DSRB-MPC method are larger than those of the previous two experiments, this method can still control the stable walking of the robot, and the fluctuations tend to decrease gradually.



(**a**) Forward velocity

(**b**) Roll angle

(**c**) Pitch angle

(**d**) Yaw angle

Figure 10. The performance of moving in the opposite direction. (**a**) The command velocity and measured actual velocities of are presented. (**b**) The command roll angle and measured actual roll angles are presented. (**c**) The command pitch angle and measured actual pitch angles are presented. (**d**) The command yaw angle and measured actual yaw angles are presented. Note that all blue lines represent given reference signals. Red lines show the performance of SRB-MPC. Yellow lines show the performance of DSRB-MPC. The red line stops because the error of SRB-MPC exceeds the maximum error, the simulation stops at about 10 s. The reason for the instability of SRB-MPC is that the disturbances of the swinging leg are not considered in the modeling, and the modeling error is too large. In this situation, modeling errors for roll and yaw are too large, and MPC cannot resist. However, DSRB-MPC can accurately model disturbances and resist them.

5. Discussion

In this paper, we propose the DSRB-MPC method based on an improved SRB-MPC method and the DRL method. The new method takes into account the disturbances of the swinging leg, which enables the MPC method to be applied to bipedal robots with non-negligible leg mass. First, we add the external disturbances to the SRB model to further increase the anti-disturbance capability of the SRB-MPC method. Subsequently, we consider the effect of the swinging leg on the SRB model as the centroid acceleration disturbance and the rotational acceleration disturbance. We use the PPO method to train a policy to predict the disturbances of the swinging leg. Finally, we verify the effectiveness and robustness of the DSRB-MPC method through three experiments. The above experiments show that the SRB-MPC method has non-negligible robustness, and it does not require very high accuracy of the SRB model. The SRB-MPC method can resist a part of the disturbances causing by the swing leg and does not immediately make the robot fall. Therefore, it is meaningful for us to improve and research the SRB-MPC method and expand its application range. The above experiments also demonstrate that the policy obtained by the DRL method can accurately predict the disturbances of the swinging leg. The disturbances include two components, which are the disturbance of the centroid acceleration and the disturbance of the rotational acceleration. Based on the predicted swing leg disturbances, the DSRB-MPC method can give the truly optimal GRFs, enabling the biped robot to accurately track the forward velocity reference while resisting the disturbances. Therefore, the DSRB-MPC method is suitable for biped robots with non-negligible leg mass.

Our method is equally applicable to quadrupeds, as the SRB-MPC method is applicable to quadrupeds, and our method is an improvement on it. Some quadruped robots also have the problem that the portion of leg mass is relatively large. Our method can also alleviate the influence of the swing leg on the overall motion of the robot, thereby improving the stability. However, due to the structural characteristics of the quadruped robot itself, it has the better stability than the biped robot. Therefore, the improvement of this method for quadruped robots will not be as significant as that for biped robots.

However, this method also has some shortcomings. First, the larger the proportion of the leg mass to total mass, the more difficult it is to predict the disturbances. When the proportion exceeds 30%, the method cannot guarantee the stability of the biped robot. Second, it does not eliminate the static velocity error.

In the future, we consider to continue to enhance the robustness of the DSRB-MPC method to break through the limitation that the leg mass accounts for 30% of the total mass. We plan to take the following approaches to predict the disturbances caused by the larger proportion of leg mass. Improve the reward function so that it better reflects the relationship between disturbances and the state of the swinging leg. Replacing the existing actor and critic network with a long-short-term memory network; combining the current state of the swinging leg with the state of the previous moment can give a more accurate prediction of disturbances. Increasing the execution frequency of the disturbances prediction policy can predict the change of disturbances more quickly. We also consider eliminating the static velocity error of this method and realizing the stable walking of the biped robot in complex environments. We plan to take the following approaches to eliminate the static velocity error. By designing an adaptive algorithm, the weight coefficients and prediction horizon of MPC can be dynamically adjusted according to the desired velocity and the actual velocity. Since increasing the weight coefficient of the velocity error as much as possible can reduce the static velocity error, by increasing the running frequency of the MPC as much as possible, the influence of the model errors on the controlled object can be reduced, thereby reducing the static velocity error. We would add a feedback controller before MPC to output the compensated desired velocity according to the given desired velocity and actual velocity, and use the compensated desired velocity as the input of MPC.

Author Contributions: Conceptualization, L.H. and B.L.; methodology, L.H.; software, L.H.; validation, L.H., B.L., W.L., Y.X. and S.Y.; formal analysis, L.H.; investigation, L.H.; resources, B.L.; data curation, L.H.; writing—original draft preparation, L.H.; writing—review and editing, B.L. and X.R.; visualization, L.H.; supervision, B.L.; project administration, B.L.; funding acquisition, B.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the National Natural Science Foundation of China under Grant 61973185, in part the Natural Science Foundation of Shandong Province under Grant ZR2020MF097 and the Colleges and Universities Twenty Terms Foundation of Jinan City (2021GXRC100).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Mikolajczyk, T.; Mikołajewska, E.; Al-Shuka, H.F.N.; Malinowski, T.; Kłodowski, A.; Pimenov, D.Y.; Paczkowski, T.; Hu, F.; Giasin, K.; Mikołajewski, D.; et al. Recent Advances in Bipedal Walking Robots: Review of Gait, Drive, Sensors and Control Systems. *Sensors* **2022**, *22*, 4440. [CrossRef]
2. Schulman, J.; Wolski, F.; Dhariwal, P.; Radford, A.; Klimov, O. Proximal Policy Optimization Algorithms. *arXiv* **2017**, arXiv:1707.06347.
3. Lillicrap, T.P.; Hunt, J.J.; Pritzel, A.; Heess, N.; Erez, T.; Tassa, Y.; Silver, D.; Wierstra, D. Continuous Control with Deep Reinforcement Learning. *arXiv* **2015**, arXiv:1509.02971.
4. Ramezani Dooraki, A.; Lee, D.J. A Multi-Objective Reinforcement Learning Based Controller for Autonomous Navigation in Challenging Environments. *Machines* **2022**, *10*, 500. [CrossRef]
5. Lee, C.; An, D. AI-Based Posture Control Algorithm for a 7-DOF Robot Manipulator. *Machines* **2022**, *10*, 651. [CrossRef]
6. Siekmann, J.; Green, K.; Warila, J.; Fern, A.; Hurst, J. Blind Bipedal Stair Traversal via Sim-to-Real Reinforcement Learning. *arXiv* **2021**, arXiv.2105.08328.
7. Castillo, G.A.; Weng, B.; Zhang, W.; Hereid, A. Reinforcement Learning-Based Cascade Motion Policy Design for Robust 3D Bipedal Locomotion. *IEEE Access* **2022**, *10*, 20135–20148. [CrossRef]
8. Dao, J.; Green, K.; Duan, H.; Fern, A.; Hurst, J. Sim-to-Real Learning for Bipedal Locomotion Under Unsensed Dynamic Loads. *arXiv* **2022**, arXiv:2204.04340.
9. Xie, Z.; Berseth, G.; Clary, P.; Hurst, J.; van de Panne, M. Feedback Control For Cassie With Deep Reinforcement Learning. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1241–1246. [CrossRef]
10. Xie, Z.; Clary, P.; Dao, J.; Morais, P.; Hurst, J.; van de Panne, M. Iterative Reinforcement Learning Based Design of Dynamic Locomotion Skills for Cassie. *arXiv* **2019**, arXiv:1903.09537.
11. Kajita, S.; Kanehiro, F.; Kaneko, K.; Yokoi, K.; Hirukawa, H. The 3D Linear Inverted Pendulum Mode: A Simple Modeling for a Biped Walking Pattern Generation. In Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems. Expanding the Societal Role of Robotics in the the Next Millennium (Cat. No. 01CH37180), Maui, HI, USA, 29 October–3 November 2001; Volume 1, pp. 239–246. [CrossRef]
12. Rezazadeh, S.; Hubicki, C.; Jones, M.; Peekema, A.; Van Why, J.; Abate, A.; Hurst, J. Spring-Mass Walking with Atrias in 3D: Robust Gait Control Spanning Zero to 4.3 KPH on a Heavily Underactuated Bipedal Robot. Dynamic Systems and Control Conference. *Am. Soc. Mech. Eng.* **2015**, *1*, V001T04A003. [CrossRef]
13. Di Carlo, J.; Wensing, P.M.; Katz, B.; Bledt, G.; Kim, S. Dynamic Locomotion in the MIT Cheetah 3 Through Convex Model-Predictive Control. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1–9. [CrossRef]
14. Grizzle, J.; Chevallereau, C.; Shih, C.L. HZD-Based Control of a Five-Link Underactuated 3D Bipedal Robot. In Proceedings of the 2008 47th IEEE Conference on Decision and Control, Cancun, Mexico, 9–11 December 2008; pp. 5206–5213. [CrossRef]
15. Gong, Y.; Hartley, R.; Da, X.; Hereid, A.; Harib, O.; Huang, J.K.; Grizzle, J. Feedback Control of a Cassie Bipedal Robot: Walking, Standing, and Riding a Segway. In Proceedings of the 2019 American Control Conference (ACC), Philadelphia, PA, USA, 10–12 July 2019; pp. 4559–4566. [CrossRef]
16. Englsberger, J.; Ott, C.; Roa, M.A.; Albu-Schäffer, A.; Hirzinger, G. Bipedal Walking Control Based on Capture Point Dynamics. In Proceedings of the 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 4420–4427. [CrossRef]
17. Vukobratović, M.; Borovac, B. Zero-Moment Point—Thirty Five Years of Its Life. *Int. J. Humanoid Robot.* **2004**, *1*, 157–173. [CrossRef]

18. Shi, X.; Gao, J.; Lu, Y.; Tian, D.; Liu, Y. Biped Walking Based on Stiffness Optimization and Hierarchical Quadratic Programming. *Sensors* **2021**, *21*, 1696. [CrossRef]
19. Li, J.; Nguyen, Q. Force-and-Moment-Based Model Predictive Control for Achieving Highly Dynamic Locomotion on Bipedal Robots. In Proceedings of the 2021 60th IEEE Conference on Decision and Control (CDC), Austin, TX, USA, 14–17 December 2021; pp. 1024–1030. [CrossRef]
20. Kim, D.; Di Carlo, J.; Katz, B.; Bledt, G.; Kim, S. Highly Dynamic Quadruped Locomotion via Whole-Body Impulse Control and Model Predictive Control. *arXiv* **2019**, arXiv:1909.06586.
21. García, G.; Griffin, R.; Pratt, J. MPC-Based Locomotion Control of Bipedal Robots with Line-Feet Contact Using Centroidal Dynamics. In Proceedings of the 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), Munich, Germany, 19–21 July 2021; pp. 276–282. [CrossRef]
22. Sleiman, J.P.; Farshidian, F.; Minniti, M.V.; Hutter, M. A Unified MPC Framework for Whole-Body Dynamic Locomotion and Manipulation. *IEEE Robot. Autom. Lett.* **2021**, *6*, 4688–4695. [CrossRef]
23. Duan, H.; Dao, J.; Green, K.; Apgar, T.; Fern, A.; Hurst, J. Learning Task Space Actions for Bipedal Locomotion. In Proceedings of the 2021 IEEE International Conference on Robotics and Automation (ICRA), Xi'an, China , 30 May–5 June 2021; pp. 1276–1282. [CrossRef]
24. Castillo, G.A.; Weng, B.; Zhang, W.; Hereid, A. Hybrid Zero Dynamics Inspired Feedback Control Policy Design for 3D Bipedal Locomotion using Reinforcement Learning. In Proceedings of the 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 8746–8752. [CrossRef]
25. Engstrom, L.; Ilyas, A.; Santurkar, S.; Tsipras, D.; Janoos, F.; Rudolph, L.; Madry, A. Implementation Matters in Deep Policy Gradients: A Case Study on PPO and TRPO. *arXiv* **2020**, arXiv:2005.12729.
26. Schulman, J.; Moritz, P.; Levine, S.; Jordan, M.; Abbeel, P. High-Dimensional Continuous Control Using Generalized Advantage Estimation. *arXiv* **2015**, arXiv:1506.02438.