Recovery from Injury: Learning Bipedal Jumping Skills with a Motor Output Torque Limit Curriculum

Jiayi Li[1], Linqi Ye[2], Yujie Sun[3], Houde Liu[1*], and Bin Liang[4]

[1] Tsinghua Shenzhen International Graduate School, Shenzhen 518055, China
liu.hd@sz.tsinghua.edu.cn
[2] Shanghai University, 200444 Shanghai, China
[3] China North Vehicle Research Institute Postdoctoral workstation, 100072 Beijing, China
[4] Tsinghua University, 100084 Beijing, China

Abstract. Bipedal robots can walk and run on different terrains and show great capacity in fast-moving, however, it's still a daunting challenge for them to achieve highly dynamic whole-body motions such as jumping. In this paper, we propose a method to learn high jump skills for humanoid robots, and the validity of the method is proved in simulation on the Ranger Max humanoid robot model. Both 2D and 3D jumping locomotion for the one-legged and two-legged Ranger Max robots are generated naturally and stably with different scales of maximum motor output torque limit. A curriculum learning strategy inspired by the idea of "recovery from injury" to make the learning of the high jump more efficient for "weaker" robots is also proposed and confirmed by the simulation experiment.

Keywords: Humanoid robot, high jump, output torque, curriculum learning.

1 Introduction

The ability to demonstrate high athletic performance is one of the most attractive goals for bionic robots, and the high jump is a typical locomotion that demonstrates strength and control. The wheeled robot Handle developed by Boston Dynamics stands 6.5 ft tall and jumps 4 feet vertically [1]. A tiny robot designed by Dr. Elliot Hawkes et al weighs less than a tennis ball and can reach 31 meters, which is higher than any creature in the world [2]. Bipedal robots are often designed as general-purpose mobile robots which are required to perform a wide variety of tasks so their physical structures are not designed for one or a few specific actions. Therefore, how to control a robot flexibly and exert the maximum capacity of the existing mechanical structure and motors in challengeable locomotion like the high jump becomes an important research content.

There are mainly two ways to control a bipedal robot in general, the first one is the model-based trajectory optimization method which has led to many mature bipedal control algorithms. In the classical zero-moment point (ZMP) method [3], the robot is always seen as a simplified linear inverted pendulum model (LIPM) or spring-loaded inverted pendulum (SLIP), whose stability is guaranteed by making the ZMP lie within the interior of the supporting polygon during walking. Hybrid zero dynamics (HZD) is another popular framework [4-6]. It works by designing a set of virtual constraints that are enforced via feedback control of the actuated degrees of freedom (DoF). Model Predictive Control (MPC) is by far the best concerned model-based optimization approach to generating bipedal locomotion [7-11]. Highly dynamic movements such as jumping and running can be performed and robustness properties against unpredictable external disturbances increases due to its previewing ability in the prediction horizon. Though notable achievements have been made on advanced humanoid robots like Atlas [12], these model-based optimization methods have their disadvantages. An accurate model is always required and the computational cost is relatively high. Planning failures may occur due to unpredictable external disturbances in complex environments and model mismatch, which is more likely to happen on bipedal robots that are less stable compared to quadruped robots.

The data-driven method based on reinforcement learning (RL) provides a second solution. Artificial intelligence does well in solving high-dimensional, multiple-input-multiple-output problems. The agents continuously interact with the environment and learn from the reward feedback automatically, network designing, reward setting, curriculum designing, and engineering problems of sim-to-real become the new challenges. Some effective reinforcement learning frameworks have been proposed to handle continuous control tasks, such as Deep Deterministic Policy Gradients (DDPG) [13], Trust Region Policy Optimization (TRPO) [14], Actor-Critic with Experience Replay (ACER) [15], and Proximal Policy Optimization (PPO) [16]. Duan et al integrated learning a task space policy with a model-based inverse dynamic controller and demonstrated a successful sim-to-real transfer on Cassie [17]. Li et al present a model-free RL framework that can be transferred to a real robot with a gait library of diverse parameterized motions based on HZD [18]. Xie et al describe an iterative design approach with transfer learning and get robust locomotion policies [19]. Their robot even completed the 100 meters in 24.73 seconds sprinting to 100-meter World Record.

The advantages of RL make it a useful tool to solve the high dynamic bipedal motion problem, and we choose it as the basic framework to design a humanoid high jump method.

2 Related Work

2.1 Proximal Policy Optimization

PPO [16] is a kind of modified policy gradient method that strikes a balance between ease of implementation, sample complexity, and ease of tuning. It tries to compute an update at each step which minimizes the cost function and ensures the deviation from the previous policy is relatively small at the same time. The objective function of PPO is given as:

$$L^{CLIP}(\theta) = \hat{E}_t\ [min\,(r_t(\theta))\,\hat{A}_t, clip\,(r_t\ (\theta), 1\text{-}\varepsilon, 1+\varepsilon)\,\hat{A}_t\ )] \tag{1}$$

where θ is the policy parameter, $\hat{E}_t$ denotes the empirical expectation over timesteps, r_t is the ratio of the probability under the new and old policies respectively, $\hat{A}_t$ is the estimated advantage at the time t and ε is a hyperparameter. The novel clipped objective function was proven to outperform most other RL methods on almost all the continuous control problems, showing its excellent ability in the field of continuous robot motion control. We choose to do our research with a PPO trainer.

2.2 High Jump

A robot jumps by launching its body into the air with a single stroke of its joints. The amount of energy delivered in this single stroke determines the jump height and distance. The high jump which focuses mainly on the maximum jump height, is a good way to test the explosive strength and balance ability of a robot control method.

Xiong et al identify a spring-mass model from the kinematics and compliance of the 3D bipedal robot Cassie. Jumping and landing motions are planned based on leg length trajectories optimized via direct collocation to synthesize a control Lyapunov function based quadratic program. Whole body rotation in the underactuated flight phase is prevented through an additional centroidal angular momentum output in the control function. A ~7 inches (17.78cm) ground clearance and ~0.423s air-time are finally achieved. [20]. Kojima et al design a high-specific stiffness mechanical structure for dynamic jumping motions which is also lightweight. They achieve a 0.3m height in the jumping test [21]. Qi et al propose a vertical jump optimization strategy for a one-legged robot. Full-body dynamics are considered in their method to track the trajectory with virtual force control and human jumping motion capture data is collected and used as the reference center of mass (CoM) trajectory to realize a certain jumping height. A 50 cm jump is realized on a real robot platform [22,23]. Chen et al clarify the mathematical modeling and motor-joint model with practical factors considered. They optimize the hopping performance of the robot by maximizing the output power of the joint [24].

3 Preliminary

3.1 Problem Formulation

Although bipedal robots show great capacity in fast-moving, it’s still a daunting challenge for them to achieve highly dynamic whole-body motions such as high jump which is of great importance to improve the agility and adaptation of humanoid robots. Jumping is a hybrid dynamical phenomenon with ground and flight faces in essence, and existing control methods for jumping are almost model-based. Among previous works, an offline whole-body trajectory is always generated before jumping, and online control algorithms are used to ensure stability. Different controllers are elaborately designed for different phases, such as trajectory optimization designed for the launching phase, momentum control designed for the flight phase, and viscoelastic control designed for the landing phase. Except for the difficulty of the specific controller design, inherent intractability lies in the model-based control method:

1)Model mismatch: model mismatch introduced by modeling simplification, measurement errors, and load variation causes instability of the system. Also, specialized controller designs are difficult to replicate directly with other humanoid robots with different mechanical structures;

2)Lack of flexibility: more than needed degrees of freedom are always wasted on CoM trajectory following as well as other artificial constraints instead of pursuing higher jump height;

3)Limitation of locomotion: the jumping ability is limited by the reference locomotion but not the physical properties of the robot itself. It's hard to bring out the full potential of the robot based on a manual planning trajectory;

4)Unnatural jumping posture: the relaxation of the knee and ankle joints is a common phenomenon in the flight phase of animal jumping. However, they are always bent to keep the robot controllable in a model-based method. It's very difficult to summarize certain rules artificially that could generate natural and fluid body movement.

Considering the questions above, it's necessary to propose a relatively simple control framework to achieve natural jumping movements and make the jump as high as possible. Therefore, we design a high jump learning method based on PPO, and the main research contents are as follows:

1) Presenting a reinforcement learning method to generate natural and stable high jump locomotion for humanoid robots, and verifying it in the simulation on Ranger Max robot.

2) Analyzing the influence of different motor output limits on jump height and exploring a curriculum learning method to speed up locomotion generation and increase jump height.

3.2 Robot Model

The robot model used in this article is based on the open-source humanoid robot Ranger Max, which is known as Tik-Tok before [25]. We follow the original design (Fig. 1 (a)) and make some minor structural changes to our robot. One leg of the real robot has been built in our laboratory (Fig. 1 (b)). One-legged and two-legged simulation models have been built in Unity (Fig. 1 (c), (d)), and the basic specifications of Ranger Max robot hardware are shown in Table 1.

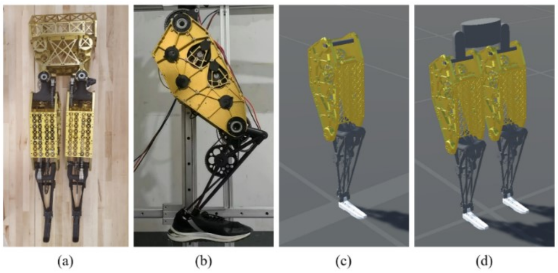

Fig. 1. Ranger Max humanoid robot. (a) The complete Ranger Max robot developed by Ruina et al. (b) One-legged Ranger Max robot we are building. (c) One-legged simulation model in Unity. (d) Two-legged simulation model in Unity.

Table 1. Table captions should be placed above the tables.

Link	Length/Width	Weight	Joint range (°)
pelvis	$w_1 = 0.24m$	2kg	-
hip	$l_1 = 0.1m$	1kg	(-20, 0) left / (0, 20) right
thigh	$l_2 = 0.4m$	6.4kg	(-50, 50)
shank	$l_3 = 0.35m$	1.3kg	(-110, 10)
foot	$l_4\, l_3 = 0.35m$	0.3kg	(-50, 50)

Each leg of Ranger Max has four controllable joints. A hip abduction-adduction (HAA) joint, a hip flexion-extension (HFE) joint, a knee flexion-extension (KFE) joint and an ankle plantar flexion and dorsiflexion (APD) joint. A one-legged robot is confined to move in a two-dimensional plane so that the HAA joint is omitted. The angle ranges of joints in the jumping task are indicated in Fig. 2 (a), and (b), and the detailed values are listed in Table 1. Height reference points are introduced to measure the jump height of the robot. The vertical distance of the point and its initial position will be recorded as the jump height as shown in Fig. 2 (c). Unlike some bipedal robots, the mass of Ranger Max is not concentrated in the pelvis but distributed on each link more evenly. Since most motors are mounted on the thigh links, they are the heaviest parts of the lower body which is quite similar to the human being.

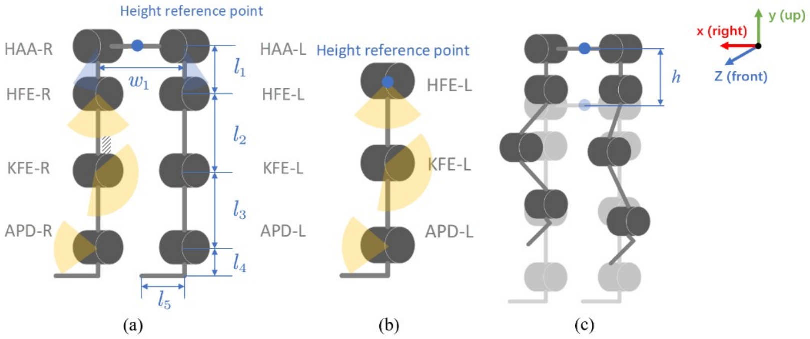

Fig. 2. Structural sketch of Ranger Max. (a) Structural sketch of two-legged robot model. (b) Structural sketch of one-Legged robot model. (c) Jumping height measurement method.

4 Control Method

4.1 Overview

An overview of our method is given in Fig. 3. The main objective of the presented method is to make the robot jump naturally and stably while achieving as high as possible in limited training steps. We use a curriculum learning (CL) [26] strategy inspired by "recovery from injury" (Fig. 3 A), and a simple joint position PD controller with a torque limiter is proposed as the control architecture.

The idea of “starting small” and gradually presenting more complex is called curricula summarized from human education. Unlike the human body which grows stronger during aging, the mechanical structure of a robot is fixed once it’s designed and built, however, the maximum power of joints is easy to limit. Thus, we simplify the task by making the robot stronger rather than reducing task difficulty directly. A specific curriculum for the high jump task is designed: we assume that the robot is an athlete who used to be strong and mastered in the high jump, he or she recovers the jumping ability quickly after injury with previous experience with a weaker body.

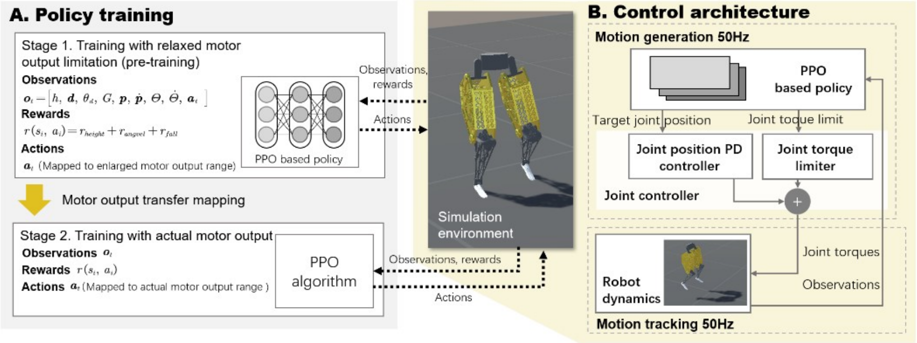
A. Policy training
Stage 1. Training with relaxed motor output limitation (pre-training)
Observations
$\boldsymbol{o}_t=[h,\ \boldsymbol{d},\ \theta_d,\ G,\ \boldsymbol{p},\ \dot{\boldsymbol{p}},\ \Theta,\ \dot{\Theta},\ \boldsymbol{a}_t\ ]$
Rewards
$r(s_i,\ a_i)=r_{height}+r_{angvel}+r_{fall}$
Actions
$\boldsymbol{a}_t$ (Mapped to enlarged motor output range)
PPO based policy
Observations, rewards
Actions
Motor output transfer mapping
Stage 2. Training with actual motor output
Observations $\boldsymbol{o}_t$
Rewards $r(s_i,\ a_i)$
Actions $\boldsymbol{a}_t$ (Mapped to actual motor output range)
PPO algorithm
Observations, rewards
Actions
Simulation environment
B. Control architecture
Motion generation 50Hz
PPO based policy
Target joint position
Joint toque limit
Joint position PD controller
Joint torque limiter
Joint controller
+
Joint torques
Observations
Robot dynamics
Motion tracking 50Hz

Fig. 3. Overall control framework. Left: Policy training. Right: Control architecture.

4.2 Training Parameters

An observation vector of 136 dimensions is set as the input of the network. It is defined at the time t as

$$\boldsymbol{o}_t=\left[h,\ \boldsymbol{d},\ \theta_d,\ G,\ \boldsymbol{p},\ \dot{\boldsymbol{p}},\ \Theta,\ \dot{\Theta},\ \boldsymbol{a}_t\ \right] \tag{2}$$

where h is the vertical jumping height measured with the help of the reference point (see Fig. 2 (c)). $\boldsymbol{d}$ is the target direction pointing forward and θ_d is the deviation angle from the target direction. These two values are not fully used since we are only doing a high jump in place, but we still put it here as it can be useful when facing direction is added to the reward. G is a 1×9 vector of Boolean values that demonstrate if the links touch the ground. $\boldsymbol{p}$ and $\dot{\boldsymbol{p}}$ denotes the positions and velocities of the leg links origin in the pelvis coordinate system. The position of the pelvis itself is not included. Θ and $\dot{\Theta}$ are the rotation presented in quaternions and angular velocities in Euler angle notation of all links in the global coordinate system. $\boldsymbol{a}_t$ is a vector of all the joint torques. The terrain of the environment is not considered yet since we start with a simple flat ground.

In a high jump task, height and stability are the most concerned factors. An intensive reward in real-time allows the agent to learn faster. The reward function consists of three components: (1) a height reward r_{height} with a restriction of the horizontal speed of the pelvis. (2) a penalty r_{angvel} for pelvis shaking and (3) a penalty r_{fall} for the robot falling over.

$$r(s_i,a_i)=r_{\text{height}}+r_{\text{angvel}}+r_{\text{fall}} \tag{3}$$

$$r_{height} = \begin{cases} K_{\mathrm{h}} f_1(h) & \text{if } h > H_{\mathrm{stimulus}} \text{ and } v_x, v_z \in (-v_{\max}, v_{\max}) \\ K_{\mathrm{h}} f_2(h) & \text{if } h < H_{\mathrm{penalty}} \text{ and } v_x, v_z \in (-v_{\max}, v_{\max}) \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

$$r_{angvel} = -K_{\mathrm{a}} \left(\left|\dot{\theta}_{px}\right| + \left|\dot{\theta}_{py}\right| + \left|\dot{\theta}_{pz}\right| \right) \tag{5}$$

$$r_{\mathrm{fall}} = \begin{cases} -P_{\mathrm{fall}} & \text{if } h < H_{\min} \text{ or } p_{\mathrm{yL}} < H_{\mathrm{kmin}} \text{ or } p_{\mathrm{yR}} < H_{\mathrm{kmin}} \\ 0 & \text{otherwise} \end{cases} \tag{6}$$

where (s_i, a_i) is the state-action pair, K_{h} and K_{a} are the jump height reward gain and angular velocity penalty gain respectively. $f_1(h)$ refers to the reward function with the variable of pelvis height h when the robot reaches a preset jump-ready height H_{stimulus} and $f_2(h)$ refers to the reward function when the robot is squatting too much or about to fall recognized by a height constant H_{penalty}. Their concrete form will be introduced in section 5. v_x and v_z are the pelvis velocity in the direction of x axis and y axis of the global coordinate. The action is only rewarded when linear velocities along these axes are within $\pm v_{\max}$ to prevent a large horizontal movement. $\dot{\theta}_{pi}$ is the angular velocity of the pelvis about the i axis ($i = x, y, z$). P_{fall} is the penalty of the robot falling down which is triggered when the pelvis height h or the height of left and right knees (p_{yL} and p_{yR}) is smaller than their minimum limitation H_{kmin}. An episode is also ended when the robot falls.

$$\boldsymbol{a}_t = [\boldsymbol{p}_{\mathrm{target}}{}^{1\times 8},\ \boldsymbol{T}_{\mathrm{lim}}{}^{1\times 8}\ ] \tag{7}$$

The action $\boldsymbol{a}_t$ is a 16-dimensional vector that consists of two parts as shown in equation (7): (1) the target position $\boldsymbol{p}_{\mathrm{target}}$ for all 8 joints and (2) a limitation of motor output torque $\boldsymbol{T}_{\mathrm{lim}}$. Each element in $\boldsymbol{T}_{\mathrm{lim}}$ is normalized to [-1, 1]

4.3 Curriculum Design

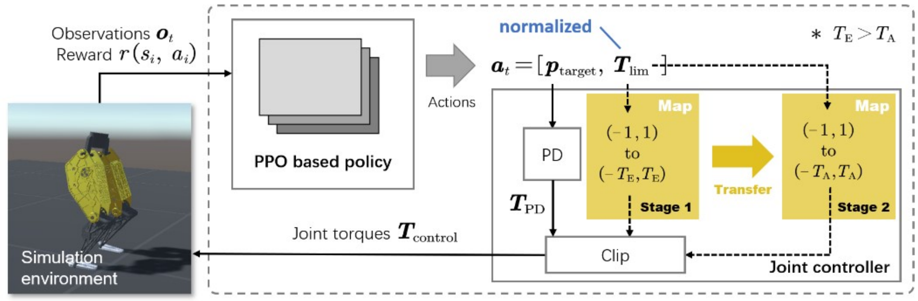
Observations $\boldsymbol{o}_t$
Reward $r(s_i,\ a_i)$
Simulation environment
PPO based policy
Actions
normalized
$\boldsymbol{a}_t = [\boldsymbol{p}_{\mathrm{target}},\ \boldsymbol{T}_{\mathrm{lim}}]$
$* \ T_{\mathrm{E}} > T_{\mathrm{A}}$
PD
$\boldsymbol{T}_{\mathrm{PD}}$
Map
$(-1,1)$ to $(-T_{\mathrm{E}}, T_{\mathrm{E}})$
Stage 1
Transfer
Map
$(-1,1)$ to $(-T_{\mathrm{A}}, T_{\mathrm{A}})$
Stage 2
Joint torques $\boldsymbol{T}_{\mathrm{control}}$
Clip
Joint controller

Fig. 4. Curriculum Design.

The idea of "recovery from injury" is used in the curriculum design. We implement this by taking a simple mapping from the pre-trained policy for a robot with enlarged maximum torque output to the actual "weaker" robot with smaller maximum torque output as shown in Fig. 4. The robot is first trained in stage 1 with a relaxed motor output torque limit T_{E}, and after basic jumping skills are learned, mapping each element in the torque limitation part $\boldsymbol{T}_{\text{lim}}$ from $(-T_{\text{E}}, T_{\text{E}})$ to $(-T_{\text{A}}, T_{\text{A}})$, where T_{A} is the actual motor output limit.

4.4 Control architecture

The joint torques are controlled by a simple position PD controller basically, while a torque limiter is added to restrict the output torques (see Fig. 3 B). We believe that a PD controller only will lead to some sudden movements when the target joint angle is far from the current joint angle causing unnatural and dangerous behaviors. Therefore, a second control variable $\boldsymbol{T}_{\text{lim}}$ is introduced to help with a softer control method as shown in Fig. 4. Now the final torque signal used to control the joint becomes

$$T_{\text{control}i} = \begin{cases} \text{clip}(T_{\text{PD}i}, -T_{\text{lim}i}T_E, T_{\text{lim}i}T_E) & \text{if in stage 1} \\ \text{clip}(T_{\text{PD}i}, -T_{\text{lim}i}T_A, T_{\text{lim}i}T_A) & \text{if in stage 2 or not in training mode} \end{cases} \tag{8}$$

where $T_{\text{PD}i}$ is the *i*th element in the torque vector $\boldsymbol{T}_{\text{PD}}$ calculated by the target joint position and the PD parameters of the PD controller. $T_{\text{control}i}$ is the *i*th element in the torque vector $\boldsymbol{T}_{\text{control}}$ used to control the joints directly.

5 Experiments and results

The following problems are addressed in this section:

1) Finding a suitable reward function for high jump training and demonstrating the feasibility of our control method;

2) Comparing the learning rate of jump height under different motor output torque limits and verifying the effectiveness of the curriculum learning method proposed.

To solve the first problem, we tried quite a few possibilities of function$f_1(h)$ and $f_2(h)$ in equation (4) and some empirical rules are found. A conservative reward like the jump height itself may lead to a timid policy in which the robot refuses to jump for fear of falling and keep trembling in situ. A radical form of reward like an exponential transformation of the jump height causes desperate attempts. Robots would rather fall to the ground to achieve greater heights. We finally made a tradeoff between jump height and stability: a cubed form of reward is adopted as follows:

$$\begin{cases} f_1(h) = K_{\text{h}}(h - H_{\text{stimulus}})^3 \\ f_2(h) = K_{\text{h}}(h - H_{\text{penalty}})^3 \end{cases} \tag{9}$$

We first train on a one-legged robot confined to its sagittal plane to eliminate the effects of lateral balance control. Three scales of motor torque output limit were chosen. We do training with the motor output torque limit of 50Nm (relatively low), 100Nm (normal), and 150Nm (relatively high) separately. Both three training generate natural and stable continuous jumping locomotion for a one-legged robot (see Fig. 5) validating the effectiveness of our method. The maximum jump heights reach 0.226m, 0.663m, and 0.920m respectively in a 30-million-step learning.

The snapshots in Fig. 5 show that not only does the jumping height differs, but diverse jumping posture are also learned with different motor output torque limit. The leaning back posture, dorsiflexion of the ankle joint in the flight phase, and the quick knee bend before touching the ground in the last line of the snapshots suggest that agents with different motor output torque limits may pick up different jumping skills.

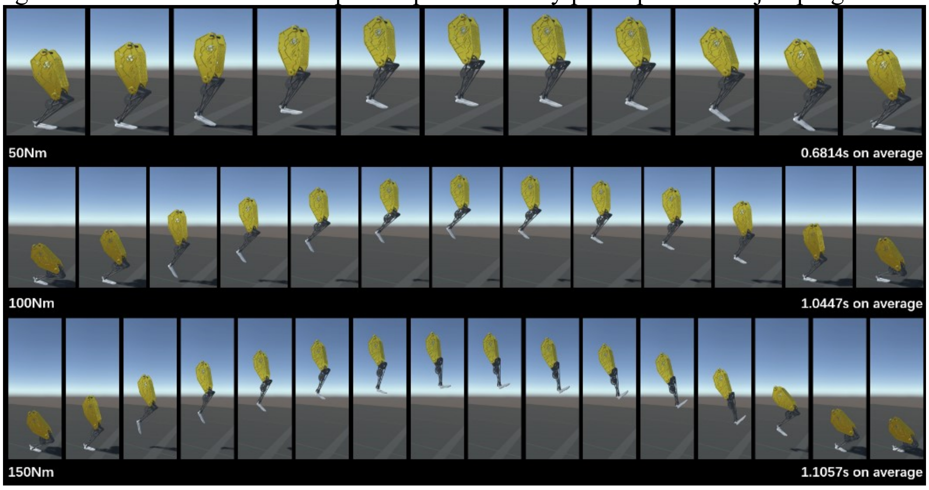

Fig. 5. Snapshots of one-legged jumping with different motor output torque limits.

We save the model once every 5 million steps and do a 30-second jump test for each model to record the maximum jump height as shown in Fig. 6 (a). The jump height learning rate shows a trend of rapid increase first and then slow growth. The agent with a greater torque limit also learns stable jumping locomotion faster. Agents with torque limits of ±100Nm and ±150Nm generate stable jumping in the first 2.5 million steps while the agent with a torque limit of ±50Nm achieves this until 5 million steps. To find out if the experience learned in the more efficient learning process with a wider range of torque limit can be utilized to help with an agent with a tighter motor output torque limit, we implement our curriculum learning method described in section 4.3 on the robot. A 10-million-step pre-training with a motor output torque limit of ±100Nm and ±150Nm is done first, and another 20-million-step training with a limit of ±50N is done with a torque limit mapping described in section 4.3. As a comparison, a 30-million-step experiment with a limit of ±50Nm from beginning to end is also completed. Results of the jumping height condition with and without the curriculum learning are shown in Fig. 6 (b).

It can be found that the agent with curriculum learning gets a higher jump height. A 39% jump height growth is made from 0.236m to 0.391m when a curriculum with a torque limit from ±100Nm to ±50Nm is taken. However, a smaller growth of 32% is made when a torque limit from ±150Nm to ±50Nm is made. The validity of the curriculum design is obvious while a higher pre-training limit may not lead to a better performance.

We repeat the above experiment on the two-legged Ranger Max robot and come to very similar results. A curriculum learning with a 10-million-step first stage and a 20-million-step second stage is done. Stable 3D jumping locomotion is generated successfully with the curriculum as shown in Fig. 7. A 0.254m jump height is achieved with a ±100Nm to ±50Nm curriculum, while the agent without curriculum and with a ±150Nm to ±50Nm curriculum even fails to produce a stable jumping strategy in 30 million steps. This phenomenon shows that a robot with more strength learns stable jumping much faster than a weak one, and a suitable scale of motor output torque limit enlarging does help with locomotion generation, but an excessive gap may backfire. The results demonstrate that our method works on both 2D and 3D conditions, and taking a curriculum learning with a wider range of torque limit first is beneficial for high jump locomotion generation and performance optimization. Our curriculum design provides a way to achieve high dynamic performance for robots with limited power.

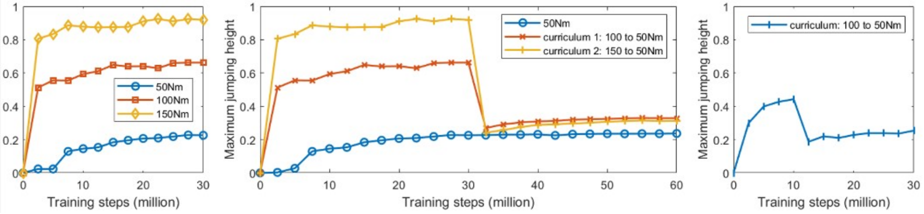

Fig. 6. The trend of jump height. (a) Maximum jump height of a one-legged Ranger Max robot with different motor output torque limit. (b) Comparison between normal training and curriculum training of a one-legged Ranger Max robot. (c) Maximum jump height of a two-legged Ranger Max robot taking a curriculum learning.

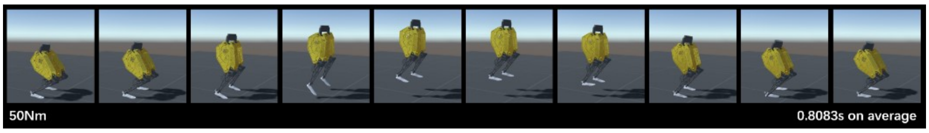

Fig. 7. Snapshots of two-legged jumping.

6 Conclusion and Future Work

In this paper, we propose a method to learn high jump skills for humanoid robots, and the validity of the method is proved on the Ranger Max humanoid robot model in simulation. Both 2D and 3D jumping locomotion for the one-legged and two-legged robots are generated naturally and stably with different scales of maximum motor output torque limit. A curriculum learning strategy inspired by the idea of "recovery from injury" to make the learning of the high jump more efficient for "weaker" robots is confirmed by the simulation experiment.

The real robot of Ranger Max is being built in our laboratory. The next step for our work is to verify the feasibility of our method on the real robot. This paper demonstrates a theoretical possibility of how high a robot can jump under certain drive capability constraints, while it's far more complex to achieve the jumping locomotion in the real world. Many sim-to-real problems need to be considered. The mathematical motor model should be built to imitate its real response performance in a simulation environment. Second, observations need to be filtered before being fed to the neural network and sensing errors should be introduced during training to improve the robustness of the strategy. Engineering problems like collision protection and fall protection should also be addressed.

The proposed curriculum learning of "recovery from injury" may not be beneficial only for the jumping task, humans easily relearn movements they used to excel at even muscles deteriorate as they age, therefore more attempts at this kind of curriculum on humanoid locomotion like balance keeping, obstacle crossing, and passive walking are worth exploring for us.

Acknowledgment

This work was supported by the National Natural Science Foundation of China No.92248304.

References

1. Introducing Handle, https://www.youtube.com/watch?v=-7xvqQeoA8c. Accessed 20 May 2023.
2. World's Highest Jumping Robot, https://www.youtube.com/watch?v=daaDuC1kbds. Accessed 20 May 2023.
3. Park, J., Youm, Y.: General ZMP preview control for bipedal walking. Proceedings 2007 IEEE international conference on robotics and automation, pp. 2682-2687. IEEE (2007).
4. Hereid, A., Hubicki, C. M., Cousineau, E. A., Ames, A. D.: Dynamic humanoid locomotion: A scalable formulation for HZD gait optimization. IEEE Transactions on Robotics 34(2), 370-387 (2018).
5. Hereid, A., Cousineau, E. A., Hubicki, C. M., Ames, A. D.: 3D dynamic walking with underactuated humanoid robots: A direct collocation framework for optimizing hybrid zero dynamics. 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 1447-1454. IEEE (2016).
6. Agrawal, A., Harib, O., Hereid, A., et al: First steps towards translating HZD control of bipedal robots to decentralized control of exoskeletons. IEEE Access 5, 9919-9934 (2017).
7. Brasseur, C., Sherikov, A., Collette, C., Dimitrov, D., Wieber, P. B.: A robust linear MPC approach to online generation of 3D biped walking motion. 2015 ieee-ras 15th international conference on humanoid robots (humanoids), pp. 595-601. IEEE (2015):.
8. Zamparelli, A., Scianca, N., Lanari, L., Oriolo, G.: Humanoid gait generation on uneven ground using intrinsically stable MPC. IFAC-PapersOnLine 51(22), 393-398 (2018).
9. Scianca, N., De Simone, D., Lanari, L., Oriolo, G.: MPC for humanoid gait generation: Stability and feasibility. IEEE Transactions on Robotics 36(4), 1171-1188 (2020).
10. Romualdi, G., Dafarra, S., L'Erario, G., Sorrentino, I., Traversaro, S., & Pucci, D.: Online non-linear centroidal mpc for humanoid robot locomotion with step adjustment. 2022 International Conference on Robotics and Automation (ICRA), pp.10412-10419. IEEE (2022).
11. Kashyap, A. K., Parhi, D. R.: Optimization of stability of humanoid robot NAO using ant colony optimization tuned MPC controller for uneven path. Soft Computing 25, 5131-5150 (2021).

12. Kuindersma, S., Deits, R., Fallon, M., et al. Optimization-based locomotion planning, estimation, and control design for the atlas humanoid robot. Autonomous robots 40, 429-455 (2016).
13. Lillicrap, T. P., Hunt, J. J., Pritzel, A., et al: Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971 (2015).
14. Schulman, J., Levine, S., Abbeel, P., Jordan, M., Moritz, P.: Trust region policy optimization. International conference on machine learning. PMLR (2015).
15. Wang, Z., Bapst, V., Heess, N., Mnih, V., Munos, R., Kavukcuoglu, K., & de Freitas, N.: Sample efficient actor-critic with experience replay. arXiv preprint arXiv:1611.01224 (2016).
16. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., Klimov, O.: Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347 (2017).
17. Duan, H., Dao, J., Green, K., Apgar, T., Fern, A., Hurst, J.: Learning task space actions for bipedal locomotion. 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 1276-1282. IEEE (2021).
18. Li, Z., Cheng, X., Peng, X. B., Abbeel, P., Levine, S., Berseth, G., & Sreenath, K.: Reinforcement learning for robust parameterized locomotion control of bipedal robots. 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 2811-2817. IEEE (2021).
19. Xie, Z., Clary, P., Dao, J., Morais, P., Hurst, J., Panne, M.: Learning locomotion skills for cassie: Iterative design and sim-to-real. Conference on Robot Learning, pp. 317-329. PMLR (2020).
20. Xiong, X., Ames, A. D.: Bipedal hopping: Reduced-order model embedding via optimization-based control. 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 3821-3828. IEEE (2018).
21. Kojima, K., Kojio, Y., Ishikawa, T., Sugai, F., Kakiuchi, Y., Okada, K., & Inaba, M.: A robot design method for weight saving aimed at dynamic motions: Design of humanoid JAXON3-P and realization of jump motions. 2019 IEEE-RAS 19th International Conference on Humanoid Robots (Humanoids), pp. 586-593. IEEE (2019).
22. Qi, H., Chen, X., Yu, Z., Huang, G., Meng, L., Hashimoto, K., Liao, W., Huang, Q.: A vertical jump optimization strategy for one-legged robot with variable reduction ratio joint. 2020 IEEE-RAS 20th International Conference on Humanoid Robots (Humanoids), pp. 262-267. IEEE (2021).
23. Qi, H., Chen, X., Yu, Z., Huang, G., Liu, Y., Meng, L., Huang, Q.: Vertical Jump of a Humanoid Robot With CoP-Guided Angular Momentum Control and Impact Absorption. IEEE Transactions on Robotics (2023).
24. Chen, X., Liao, W., Yu, Z., Qi, H., Jiang, X., Huang, Q.: Motion coordination for humanoid jumping using maximized joint power. Advances in Mechanical Engineering 13(6), 16878140211028448 (2021).
25. Cornell Tik-Tok: Efficient, robust, and nimble open-source legged robot, http://ruina.tam.cornell.edu/research/topics/locomotion_and_robotics/Tik-Tok/. Accessed 20 May 2023.
26. Wang X, Chen Y, Zhu W. A survey on curriculum learning. IEEE Transactions on Pattern Analysis and Machine Intelligence 44(9), 4555-4576 (2021).