Realizing Human-like Walking and Running with Feedforward Enhanced Reinforcement Learning

Linqi Ye[1], Xueqian Wang[2], Bin Liang[3]

[1] Institute of Artificial Intelligence, Collaborative Innovation Center for the Marine Artificial Intelligence, Shanghai University, 200444 Shanghai, China.
[2] Center of Intelligent Control and Telescience, Tsinghua Shenzhen International Graduate School, Tsinghua University, 518055 Shenzhen, China.
[3]Navigation and Control Research Center, Department of Automation, Tsinghua University, 100084 Beijing, China.
yelinqi@shu.edu.cn

Abstract. Locomotion control of legged robots is a challenging problem. Recently, reinforcement learning has been applied to legged locomotion and made a great success. However, the reward signal design remains a challenging problem to produce a humanlike motion such as walking and running. Although imitation learning provides a way to mimic the behavior of humans or animals, the obtained motion may be restricted due to the over-constrained property of this method. Here we propose a novel and simple way to generate humanlike behavior by using feedforward enhanced reinforcement learning (FERL). In FERL, the control action is composed of a feedforward part and a feedback part, where the feedforward part is a periodic time-dependent signal generated by a state machine and the feedback part is a state-dependent signal obtained by a neural network. By using FERL with a simple feedforward of two feet stepping up and down alternately, we achieve humanlike walking and running for a simulated biped robot, Ranger Max. Comparison results show that the feedforward is key to generating humanlike behavior, while the policy trained with no feedforward only results in some strange gaits. FERL may also be extended to other legged robots to generate various locomotion styles, which provides a competitive alternative for imitation learning.

Keywords: Reinforcement Learning, Biped Robot, Locomotion Control.

1 Introduction

Biped robots have long been a hot research field, especially when Tesla showcased their humanoid robot, the prototype of Optimus on September 30, at AI Day 2022. It is reported that the aim of Optimus is to perform “repetitive or boring” tasks in homes or factories for people. However, although the hardware of Optimus is super powerful, which can even lift a piano with the linear actuator used in Optimus, the locomotion performance of the robot is still far behind human beings as can be seen from their video demos. The fundamental reason is that a biped robot is unstable and has many degrees of freedom, which makes it difficult to control.

Reinforcement learning has shown its feasibility to control legged robots recently. Reinforcement learning control is usually model-free, which learns the optimal behavior to maximize a given reward. In 2017, Deepmind applies reinforcement learning to train several simulated legged robots on a diverse set of challenging terrains and obstacles, using a simple reward function based on forward velocity [1]. They obtained some simulated agents that are good at traversing obstacles. However, the behavior of the agents looks strange. In 2019, ETH achieved a breakthrough by using reinforcement learning to control a real quadruped robot ANYmal [2]. The robot has gained locomotion skills like high-speed running and recovering from any falling states, which goes beyond what had been achieved with traditional control methods. Since then, reinforcement learning has received more and more attention in the field of legged locomotion.

To generate natural-looking locomotion behaviors, many researchers have focused on reward design. An intuitive way is to set an imitation-related reward to encourage the robot to mimic a given reference motion. In [3, 4], some reference joint angles are designed and incorporated into the reward signal to train the robot Cassie. In [5], motion capture data from a real dog is used as reference trajectories in the reward to train the robot Laikago. In [6], by imitating some reference trajectories with different foot sequences, a quadruped robot learns to walk, trot, pacing, bounding, and transit among them. In [7], simple sine waves are applied to the robot's foot to implement imitation learning. Besides, it is also possible to use a reference-free reward. In [8], a time-varying reward is designed, which can generate all common bipedal gaits and can achieve blind bipedal stair traversal by using stair-like terrain randomization [9]. Using a similar method with a foot-swing reward, robust high-speed running for quadruped robots is achieved [10]. Recently, a method called adversarial motion priors has been proposed [11, 12], which trains a discriminator to predict whether a motion produced by the agent is good or not. In this way, natural gait can be learned efficiently.

Another way to realize natural motion by reinforcement learning is to use a hierarchical control structure. In [13], a structured neural network controller is proposed for the robot ATRIAS, where reinforcement learning takes care of the high-level policy and the low-level policy is a feedback-based reactive stepping controller. In [14], a hierarchical learning framework was proposed for quadruped robots, which uses reinforcement learning as the high-level policy to adjust the low-level trajectory generator. In [15], a cascade-structure controller was proposed for the robot Digit, which combines reinforcement learning with intuitive feedback in a cascade structure. In [16], a hybrid locomotion policy was proposed for a biped robot, which uses a model-free learning-based control for the stance phase and a heuristics control for the swing phase.

In this paper, we propose FERL, which has a feedforward-feedback control structure and is different from the aforementioned work. FERL removes the need for tedious reward design. It is similar to the hierarchical control structure, which uses reinforcement learning to learn only a part of the controller. But FERL uses a parallel control structure rather than a hierarchical one. We apply FERL to a simulated biped robot, Ranger Max, and compare it with the pure reinforcement learning method, which shows that FERL can learn much more natural locomotion behavior under the same reward setting.

2 Methods

2.1 The FERL Control Framework

The proposed FERL control framework has a feedforward-feedback structure, as shown in Fig. 1. Compared to traditional reinforcement learning, which uses a controller purely based on a neural network, FERL adds an additional feedforward control action, which works parallelly with the neural network to generate the control action. In this framework, the control action is a summation of the output of the neural network and the time-based feedforward signal. Considering that walking and running are periodic motions that can be well characterized by a state machine, we decide to adopt a state machine to generate the feedforward signal. Using other methods such as central pattern generator (CPG) are also suitable for generating the feedforward signal.

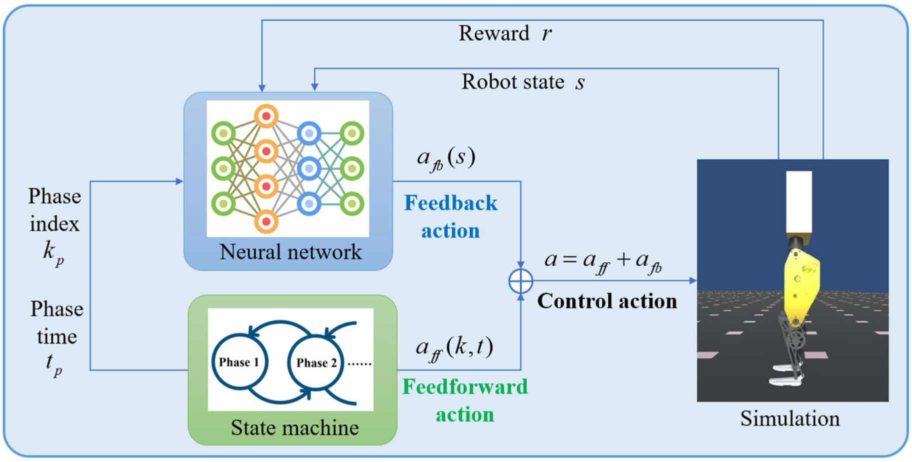

Fig. 1. The control framework of FERL. The control action is composed of a feedforward part and a feedback part, where the feedforward part is a periodic time-dependent signal generated by a state machine and the feedback part is a state-dependent signal obtained by a neural network.

In this controller, if we remove the neural network, then it becomes open-loop control. If we hang the robot in the air, its legs can move in a given motion sequence with the feedforward signal. However, since a biped robot has a floating base, it will likely fall if we put it on the ground. Although the robot may walk stably, say if the open-loop control satisfies static stability criteria, it is not robust to even a small disturbance. Therefore, we need necessary stabilization to make a feedback action, while this is exactly what the reinforcement learning part is doing. Therefore, the proposed controller combines the advantages of both reinforcement learning and open-loop control, where open-loop control provides a simple way to generate a periodic motion and reinforcement learning searches the optimal solution to stabilize that motion. As a result, the combination of feedforward and feedback can result in natural and robust bipedal locomotion behavior.

2.2 Feedforward Signal Design

The feedforward we applied is a simple stepping motion that actuates the two feet to go up and down alternately. To achieve this, a state machine is used as shown in Fig. 2. We insert two double stance phases to make the motion smoother. During the foot lifting phase, each joint angle follows a sinusoidal command as follows

$$\theta^i = \theta_1^i + \left(\theta_2^i - \theta_1^i\right)\frac{1-cos\left(2\pi * t_p/T_{step}\right)}{2} \qquad (1)$$

where θ_1^i, θ_2^i represent the angle for joint i when the foot is put down and lifted at the top, respectively. t_p is the phase time, and T_{step} is related to the step period. i refers to three joints (pitch of hip, knee, and ankle), while the hip roll angle keeps at zero.

With this feedforward, the robot will have its two legs lifting up and down alternately if we hang up the robot. However, the robot falls quickly if we put it on the ground.

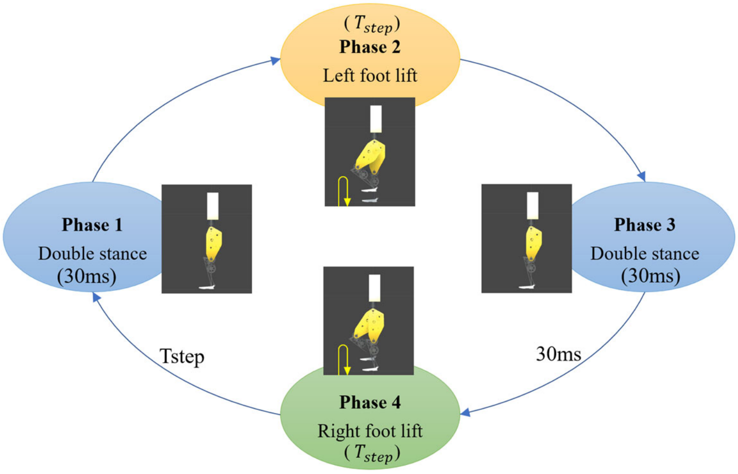

Fig. 2. The state machine for the feedforward action.

2.3 Reinforcement Learning

We explore three different control methods in order to make a comparison.

Case 1. RL.

In this case, the feedforward is turned off and the control action is obtained only by the neural network.

Case 2. FERL (fixed period).

In this case, we enable the feedforward and use a fixed step period T_{step} for the state machine. Using a fixed step period may constrain the style of gaits.

Case 3. FERL (varied period).

In this case, we enable the feedforward and use a varied step period T_{step} for the feedforward. T_{step} is also assigned as a control action and is adjusted by the neural network. This gives more flexibility for the resulting gait style.

In all cases, the observations are the same, $(q, \omega, v, \theta_i, \dot{\theta}_i, k_p, t_p)$, which consists of 28 variables in total, explained as follows

q is the orientation of the body expressed by a quaternion, containing 4 variables.

ω is the angular velocity of the body, containing 3 variables.

v is the velocity of the body, containing 3 variables.

θ_i is the angle of each joint, containing 8 variables.

$\dot{\theta}_i$ is the angular velocity of each joint, containing 8 variables.

k_p, t_p are phase variables, containing 2 variables.

For RL and FERL (fixed period), we use the desired angle for each joint θ_i^{cmd} as the action, which contains 8 variables in total. While for FERL (varied period), T_{step} is added as an additional action, which has 9 actions in total. In all cases, the joints are set to position control mode with a spring factor of 200 and a damper factor of 10.

2.4 Reward Design

Due to the incorporation of the feedforward signal, there is no need for a tedious reward design. Indeed, we found a simple reward is enough to generate a natural motion with FERL. While the feedforward signal gives a good initial reference motion for the robot, the main function of reinforcement learning is to serve as a stabilizer and regulator. On one hand, to stabilize the robot, we should keep the robot not falling (body height not too low) and keep the body upright. On the other hand, to avoid ending up stepping in place, we should use a reward to encourage forward walking. To this end, we design the reward as a combination of three components

$$r = r_a + r_u + r_v \tag{2}$$

Each component is explained as follows

r_a to stay alive, $r_a = 1$, the robot receives this reward per time step for not falling.

r_u to keep the body upright, $r_u = -0.05\theta_{\text{pitch}} - 0.05\theta_{\text{roll}} - 0.05\theta_{\text{yaw}}$.

r_v to encourage forward velocity, $r_v = v_x$, where v_x is the forward walking velocity.

Besides, we end the episode if the robot falls (body height is 0.4 meters lower than its initial height) or if the time step reaches 1000.

3 Simulation Results

3.1 Simulation Settings

The robot model used in the simulation is the Ranger Max biped robot, which is an upgraded version of the Cornell Ranger [17] robot. The model specifications including dimension and mass distribution are shown in Fig. 3. The robot uses a three-stage chain drive for all joints, which can provide a peak torque of about 200 Nm and a peak angular velocity of about 10 rad/s for each joint. Although each leg only has four degrees of freedom, it is capable of exhibiting excellent dynamic walking as shown later.

We use the ML-Agents package in Unity software for the simulation and reinforcement training. The PhysX physics engine is used and a fixed timestep of 0.01 s is adopted for simulation. For reinforcement learning, we use the PPO algorithm for training and the parameters are shown in Table 1.

With the three methods introduced in the previous section, we trained 8 million steps for each of them. The reward setting is the same and no curriculum is applied. As a result, the cumulative reward curves of the three methods are shown in Fig. 4. It can be seen that the trends of the three curves are similar, which all have a significant rise after 1 million training steps and finally achieved a large cumulative reward. Specifically, the maximum cumulative rewards achieved by the three methods in the order from large to small are: FERL (varied period) 11429, RL 11114, and FERL (fixed period) 9079, respectively. However, although RL without any feedforward can achieve a large cumulative reward close to FERL (varied period), the obtained policy will possibly lead to strange behavior, which will be shown later.

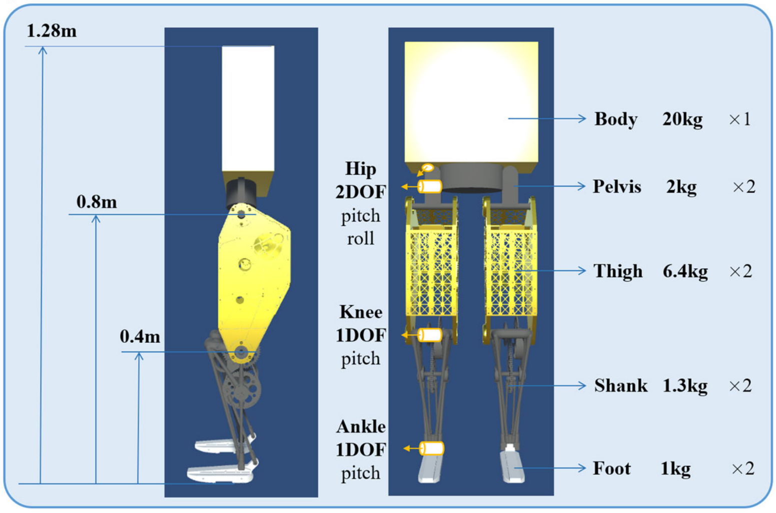

Fig. 3. Specifications of the Ranger Max biped simulation model.

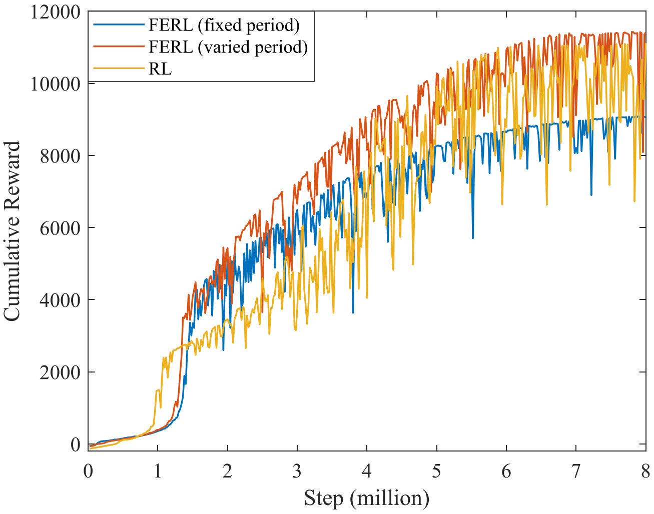

Fig. 4. Cumulative reward with respect to the training step.

Table 1. Training parameters of PPO.

Hyperparameters	**Network_settings**
batch_size: 2048	normalize: true
buffer_size: 20480	hidden_units: 512
learning_rate: 0.0003	num_layers: 3
beta: 0.005	vis_encode_type: simple
epsilon: 0.2	memory: null
lambd: 0.95	goal_conditioning_type: hyper
num_epoch: 3	deterministic: false

3.2 Simulation Results of RL

We select two policies trained by RL to apply to the robot, one at 3.6 million training steps, and one at 8 million training steps. The forward velocities of the robot's body for the two policies are shown in Fig. 5. It can be seen that for the 3.6M policy, the robot achieves a forward velocity of about 2.7 m/s, and for the 8M policy, the robot achieves a forward velocity of about 4.3 m/s.

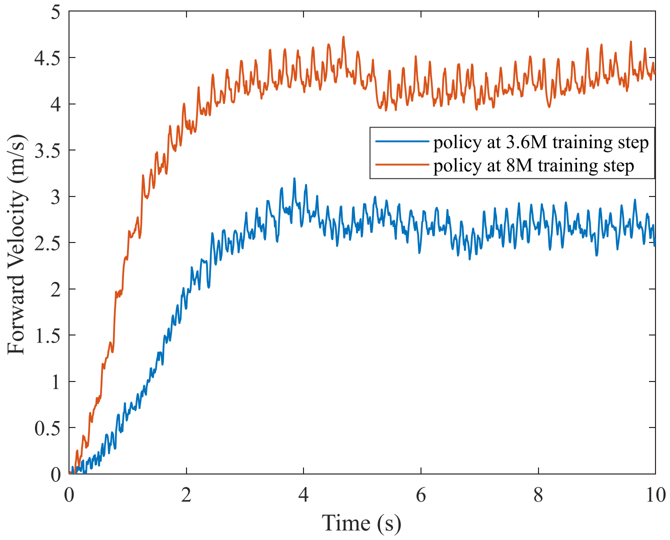

Fig. 5. Forward walking velocity for policies trained with RL.

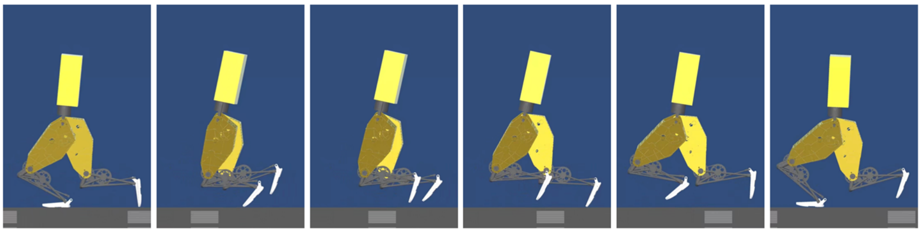

Fig. 6. Gait sequence resulting from policy trained with RL at 3.6 million training steps. This results in a non-human-like hopping gait without leg alternation.

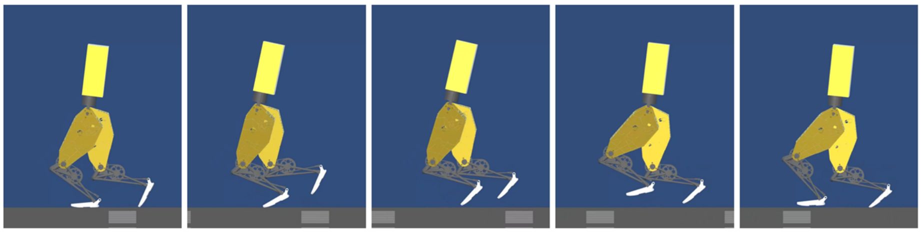

Fig. 7. Gait sequence resulting from policy trained with RL at 8 million training steps. This still results in a non-humanlike hopping gait without leg alternation.

The gait sequences resulting from the two policies are shown in Fig. 6 and Fig. 7, respectively. It can be observed that no leg alternation occurs in both gait sequences, where the left leg always keeps in the front. The joint angles seem to vary slightly and the robot hops frequently to move forward. Although the robot achieves a high speed in this way, it is obviously not humanlike. From the video, we can see that the leg is shaking with a high frequency, which looks strange and is not feasible for a real robot.

3.3 Simulation Results of FERL with Fixed Step Period

We select two policies trained by FERL with a fixed step period to apply to the robot, one at 2.4 million training steps, and one at 8 million training steps. The forward velocities of the robot's body for the two policies are shown in Fig. 8. It can be seen that for the 2.4M policy, the robot achieves a forward velocity of about 1.7 m/s, and for the 8M policy, the robot achieves a forward velocity of about 3 m/s.

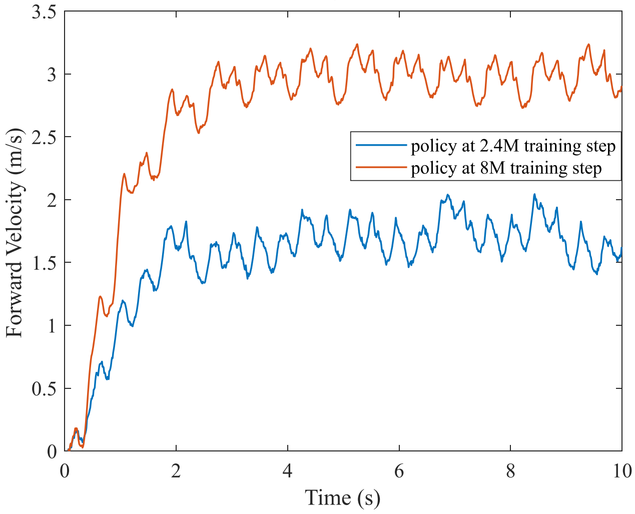

Fig. 8. Forward walking velocity of policy trained with FERL (fixed period).

The gait sequences resulting from the two policies are shown in Fig. 9 and Fig. 10, respectively. It can be observed that the gaits look much more natural now, where leg alternation occurs normally in both gait sequences. Specifically, the 2.4M policy leads to a humanlike walking gait, where the stance leg keeps relatively straight and a significant foot push-off is observed. Interestingly, the 8M policy leads to a humanlike skipping gait, where one leg hops for a small distance during the other leg swing. Why does running gait not occur? We speculate this might be because the step period ($T_{step} =$ 400 ms) we selected is too big for running, and skipping happens to be the optimal gait for the selected step period.

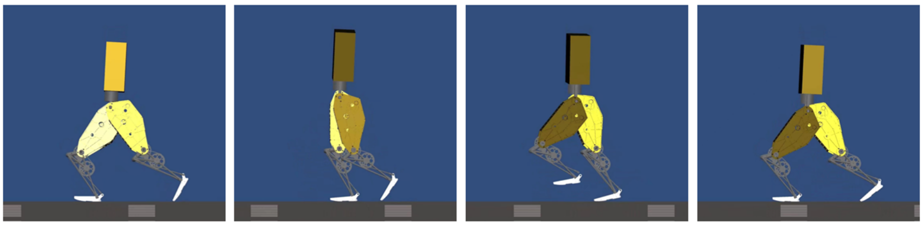

Fig. 9. Gait sequence resulting from policy trained with FERL (fixed period) at 2.4 million training steps. This results in a humanlike walking gait.

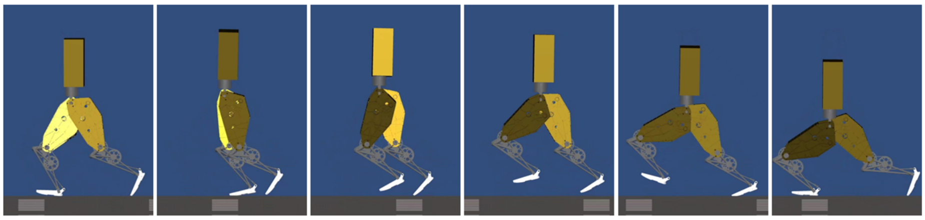

Fig. 10. Gait sequence resulting from policy trained with FERL (fixed period) at 8 million training steps. This results in a humanlike skipping gait.

3.4 Simulation Results of FERL with a Varied Step Period

We select two policies trained by FERL with a varied step period to apply to the robot, one at 2.4 million training steps, and one at 8 million training steps. The forward velocities of the robot's body for the two policies are shown in Fig. 11. It can be seen that for the 2.4M policy, the robot achieves a forward velocity of about 2.2 m/s, and for the 8M policy, the robot achieves a forward velocity of about 4.3 m/s.

The gait sequences resulting from the two policies are shown in Fig. 12 and Fig. 13, respectively. It can be observed that the gait of the 2.4M policy is similar to that obtained by the previous method, which is a humanlike walking gait. But the 8M gait is a little different from the previous one, which is a humanlike running gait now.

Running can occur mainly because of the application of a varied step period T_{step}. To verify this, we show the histograms of T_{step} for the 2.4M and 8M policy in Fig. 14. It can be seen that distributions of T_{step} are quite different for the two policies. For the 2.4M policy, T_{step} mostly distributes in the middle around 300~450, while for the 8M policy, T_{step} has the highest frequency at the shortest period 200. Therefore, during the training process, in order to go fast to get more rewards, the robot decreases T_{step} and transits to a running gait, which is similar to what we humans do when we need to go faster.

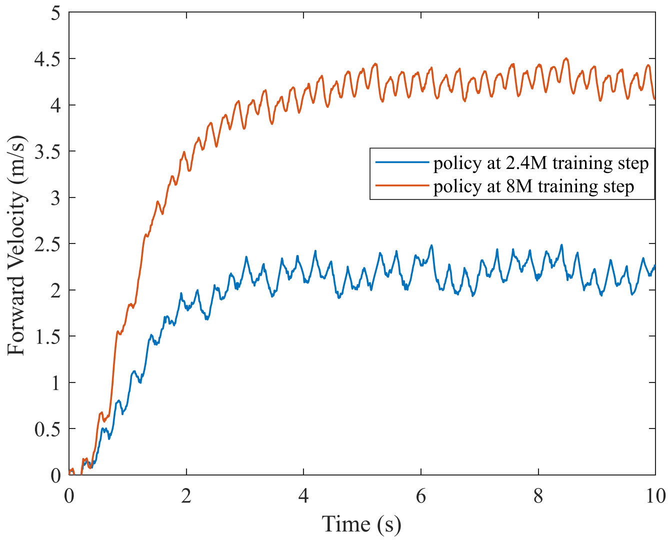

Fig. 11. Forward walking velocity of policy trained with FERL (varied period).

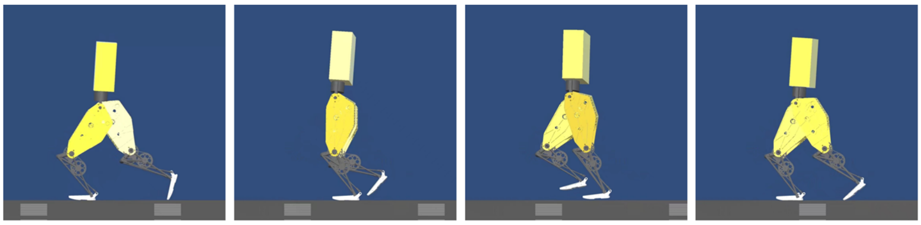

Fig. 12. Gait sequence resulting from policy trained with FERL (varied period) at 2.4 million training steps. This results in a humanlike walking gait.

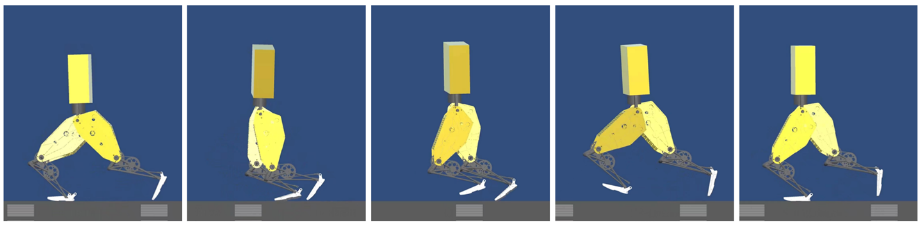

Fig. 13. Gait sequence resulting from policy trained with FERL (varied period) at 8 million training steps. This results in a human-like running gait.

Simulation video link: https://www.bilibili.com/video/BV1Rc411G7xd/

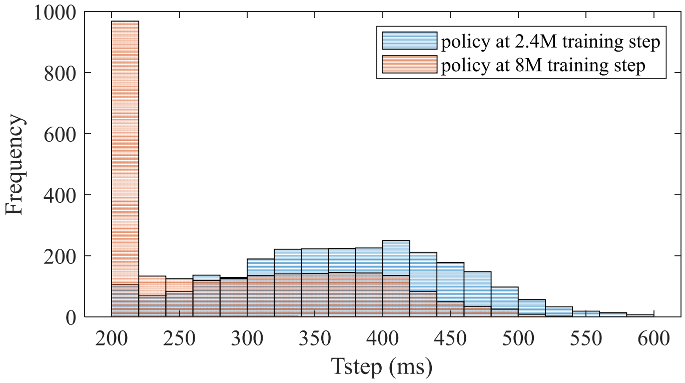

Fig. 14. Histogram of T_{step} for FERL (varied period).

4 Conclusion

How to generate humanlike behavior by reinforcement learning is a challenging problem. While traditional methods usually focus on the reward design, we instead consider modifying the control structure by using a feedforward-feedback structure, which allows for generating humanlike motion with a simple reward and no need for a curriculum. The proposed FERL method combines the advantages of both reinforcement learning and open-loop control, leading to robust and natural walking and running gaits in a simple and fast manner. We compare three different control methods in this paper, which are pure RL, FERL with a fixed step period, and FERL with a varied step period. With the same reward design and training setup, the three methods finally result in different gait styles. While pure RL leads to a strange hopping gait with no leg alternation, FERL can lead to natural human-like gaits. Specifically, FERL with a fixed step period leads to walking and skipping, and FERL with a varied step period leads to walking and running. It verifies the importance of the control structure and demonstrates the effectiveness of the proposed feedforward-feedback control structure. We have verified the proposed method on a simulated biped robot Ranger Max in this paper. The real robot of Ranger Max is still under assembly now and we will test our methods on the real robot once it is completed

Acknowledgment.

This work was supported by the National Natural Science Foundation of China No. 62003188 and No.92248304.

References

1. Heess, N., TB, D., Sriram, S., Lemmon, J., Merel, J., Wayne, G., ... & Silver, D. (2017). Emergence of locomotion behaviours in rich environments. arXiv preprint arXiv:1707.02286.

2. Hwangbo, J., Lee, J., Dosovitskiy, A., Bellicoso, D., Tsounis, V., Koltun, V., & Hutter, M. (2019). Learning agile and dynamic motor skills for legged robots. Science Robotics, 4(26), eaau5872.
3. Xie, Z., Berseth, G., Clary, P., Hurst, J., & van de Panne, M. (2018, October). Feedback control for cassie with deep reinforcement learning. In 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 1241-1246). IEEE.
4. Li, Z., Cheng, X., Peng, X. B., Abbeel, P., Levine, S., Berseth, G., & Sreenath, K. (2021, May). Reinforcement learning for robust parameterized locomotion control of bipedal robots. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 2811-2817). IEEE.
5. Peng, X. B., Coumans, E., Zhang, T., Lee, T. W., Tan, J., & Levine, S. (2020). Learning agile robotic locomotion skills by imitating animals. arXiv preprint arXiv:2004.00784.
6. Shao, Y., Jin, Y., Liu, X., He, W., Wang, H., & Yang, W. (2021). Learning free gait transition for quadruped robots via phase-guided controller. IEEE Robotics and Automation Letters, 7(2), 1230-1237.
7. Wu, Q., Zhang, C., & Liu, Y. (2022, August). Custom Sine Waves Are Enough for Imitation Learning of Bipedal Gaits with Different Styles. In 2022 IEEE International Conference on Mechatronics and Automation (ICMA) (pp. 499-505). IEEE.
8. Siekmann, J., Godse, Y., Fern, A., & Hurst, J. (2021, May). Sim-to-real learning of all common bipedal gaits via periodic reward composition. In 2021 IEEE International Conference on Robotics and Automation (ICRA) (pp. 7309-7315). IEEE.
9. Siekmann, J., Green, K., Warila, J., Fern, A., & Hurst, J. (2021). Blind bipedal stair traversal via sim-to-real reinforcement learning. arXiv preprint arXiv:2105.08328.
10. Bellegarda, G., Chen, Y., Liu, Z., & Nguyen, Q. (2022, October). Robust high-speed running for quadruped robots via deep reinforcement learning. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 10364-10370). IEEE.
11. Vollenweider, E., Bjelonic, M., Klemm, V., Rudin, N., Lee, J., & Hutter, M. (2022). Advanced skills through multiple adversarial motion priors in reinforcement learning. arXiv preprint arXiv:2203.14912.
12. Escontrela, A., Peng, X. B., Yu, W., Zhang, T., Iscen, A., Goldberg, K., & Abbeel, P. (2022, October). Adversarial motion priors make good substitutes for complex reward functions. In 2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 25-32). IEEE.
13. Li, T., Geyer, H., Atkeson, C. G., & Rai, A. (2019, May). Using deep reinforcement learning to learn high-level policies on the atrias biped. In 2019 International Conference on Robotics and Automation (ICRA) (pp. 263-269). IEEE.
14. Tan, W., Fang, X., Zhang, W., Song, R., Chen, T., Zheng, Y., & Li, Y. (2021, September). A hierarchical framework for quadruped locomotion based on reinforcement learning. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 8462-8468). IEEE.
15. Castillo, G. A., Weng, B., Zhang, W., & Hereid, A. (2021, September). Robust feedback motion policy design using reinforcement learning on a 3d digit bipedal robot. In 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (pp. 5136-5143). IEEE.
16. Wang, Z., Wei, W., Xie, A., Zhang, Y., Wu, J., & Zhu, Q. (2022). Hybrid Bipedal Locomotion Based on Reinforcement Learning and Heuristics. Micromachines, 13(10), 1688.
17. Bhounsule, P. A., Cortell, J., Grewal, A., Hendriksen, B., Karssen, J. D., Paul, C., & Ruina, A. (2014). Low-bandwidth reflex-based control for lower power walking: 65 km on a single battery charge. The International Journal of Robotics Research, 33(10), 1305-1321.