Walking Control of 2D Biped Robot Based on Reinforcement Learning

Xin Yang [a], Yang Yang [b]

Automation College, Chongqing University of Posts and Telecommunications, Chongqing 400065, China;

[a]yangxinjob@yeah.net, [b]17784236941@163.com

Abstract

This paper studier the walking control of biped robot based on reinforcement learning. To solve the problem of foot rubbing and upper body tilting during the simplest robot walking, this paper proposed impulsive push and hip drive biped robot model. The 2D biped robot simulation model is built in the simulation environment, and the gait learning system of the robot is built by Asynchronous Advantage Actor-Critic (A3C) algorithm. Combining the actual debugging experience to further optimize the state observation vector and the motion execution vector of the designed controller, compare the training results of different learning network models. By constructing the related function between the update cycle with the reward value, the stability of the training period is further improved. Finally, the trained neural network can control the 2D biped robot to walk stably more than 5,500 steps in the simulated environment.

Keywords

Biped robot, reinforcement learning, neural network, walking control.

1. Introduction

The traditional robot control algorithm needs accurate mathematical model. However, the physical prototype structure is too complex to accurately model, influence the final control result. In recent years, with the rapid development of artificial intelligence which have the advantages of without model support, online training and fast convergence, more and more scholars have applied reinforcement learning to the motion control of robots.

Lin proposed a learning framework based on Q-learning algorithm. By mapping the action space from discrete domain to continuous domain, the complex control problem of robot automatic control is solved and the stability of the biped robot walking is improved [1]. Hwang uses two Q-learnings, one of which improves the unstable gait to stable gait, and the another adjust the gait to more precise gait, made robot walk more quicker and stably [2]. Gu uses off policy learning based on deep Q-learning for complex 3D biped robot motion training, the training time is reduced through multiple robot parallel training, multiple robots asynchronously aggregate their update strategies [3]. Zhou proposes a fuzzy reinforcement learning algorithm for biped walking control, which accepts fuzzy evaluation feedback instead of digital feedback. The proposed walking controller forms the initial gait through intuitive balanced knowledge and then trains through the fuzzy reinforcement learning algorithm [4]. Feirstein realizes two kinds of impedance controllers by reinforcement learning, and the biped robot model obtained the limit cycle gait on the plane [5]. Inspired by the periodic causality of human perception and action, Jin proposes a circular learning framework based on perceptual action. The robot improves the depth estimation accuracy and motion performance of the environment through iterative loop learning [6]. Chen used the sparse online Gaussian process regression method in supervised learning to fit the value function in reinforcement learning, and proposed two reinforcement learning methods based on sparse online Gaussian process, one is model free reinforcement learning, the another is model based reinforcement learning [7].

Based on the research results of current biped robots, this paper builds the robot simulation model in simulation software Vrep, trains the walking gait of 2D biped robots by reinforcement learning, and studies the influence of different network models and network weight update periods on the training results.

2. Biped robot model

In 1990, McGeer proposed the theory of passive walking. Without any drive device, robot can walk stably on a small dip slope by gravity [8]. Passive biped robots have catch researchers' wide attention cause of low energy consumption and simple structure. Based on the study of passive dynamic walking, Garcia proposed the simplest walking model, which consists of two rigid straight legs articulated by the hip joint. The overall mass is distributed on the hip and two foots points, regardless of the rubbing and friction problems. The feet have plastic collisions with the slope surface [9]. Fig. 1 shows the passive walking of the simplest walking model.

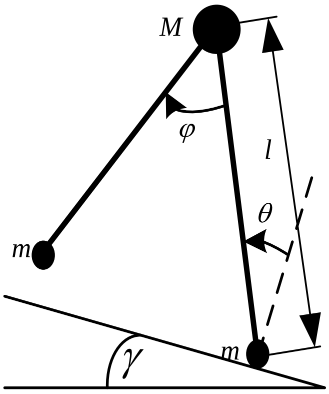


Fig. 1 Passive dynamic walking model

The leg length is l, the hip mass is M, and the foot mass is m. The angle between the support leg and the swinging leg is φ, the angle between the supporting leg and the normal line of the inclined surface is θ, and the inclination of the slope is γ.

By adding a drive, the simplest model can walk stably on plane. Collins adds drive controllers to some joints, thus the passive biped robot can walk on horizontal plane and he propose semi-passive walking [10]. Ni adds a driving torque to the hip. This torque can replace by a spring with a spring constant k and studies the effect of spring stiffness on passive walking stability [11]. Zhang adds impulsive push device to the leg and used impulsive push as the energy input of the robot to study the influence of impulsive push on the walking of semi-passive biped robot [12,13].

In this paper, the biped robot model that under the control of hip drive and impulsive push, is combined with the research results of Ni and other predecessors. In the actual project, the hip joint can't be completely smooth during the swinging of the leg, so the hip drive is used for the energy supplement; the collision of the swinging leg on the ground will also lose energy, so the impulsive push is used for the energy supplement.

2.1 Dynamic Equation

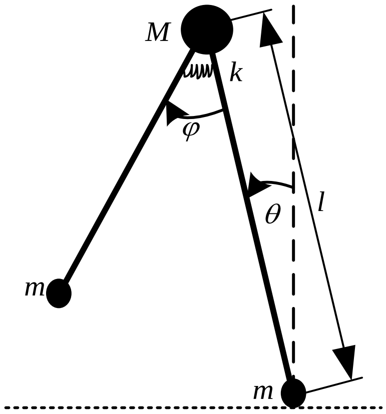


Fig. 2 Hip drive walking model

Fig. 2 shows the model of the swing phase of the robot under the action of the hip. The hip drive is replaced by a spring with a spring constant k. The other parameters are same with the simplest walking model. The Lagrange method is used to obtain the dynamic equation during swing the leg:

$$\left(1+2\frac{m}{M}\left(1-\cos\varphi\right)\right)\ddot{\theta}-\frac{m}{M}\left(1-\cos\varphi\right)\ddot{\varphi}+2\frac{m}{M}\dot{\varphi}\dot{\theta}\sin\varphi- \\ \frac{m}{M}\dot{\varphi}^2\sin\varphi+\frac{mg}{Ml}\left(\sin\left(\theta-\varphi\right)-\sin\theta\right)-\frac{g}{l}\sin\theta=0 \tag{1}$$

$$\left(\cos\varphi-1\right)\ddot{\theta}+\ddot{\varphi}-\dot{\theta}^2\sin\varphi+\frac{g}{l}\sin\left(\varphi-\theta\right)=-\frac{1}{ml^2}k\varphi \tag{2}$$

2.2 Collision Equation

When the swinging leg swings to the front of the support leg and equation (3) is satisfied, the swinging leg collides with the ground.

$$\varphi\left(\tau\right)=2\theta\left(\tau\right) \tag{3}$$

In the formula, τ is called the walking cycle, which refers to the time from the start of the swing to the time of the collision.

After the action of impulsive push, the swing leg immediately collides with the ground. Assuming the collision is inelastic collision, the collision equation can be obtained according to conservation theorem of angular momentum:

$$\begin{cases} \theta^+=-\theta^- \\ \dot{\theta}^+=-\dot{\theta}^-\cos\alpha+\dfrac{P}{Ml}\sin\alpha \\ \varphi^+=-2\theta^- \\ \dot{\varphi}^+=\dot{\theta}^+\left(1-\cos\alpha\right) \end{cases} \tag{4}$$

"+" means after the collision, "-" means before the collision, in Fig. 3 the velocity $V^+=l\dot{\theta}^+, V^-=l\dot{\theta}^-$, and the angle between the two legs φ is constant α.

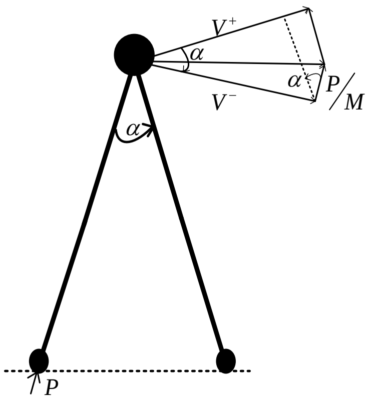


Fig. 3 Collision model under impulsive push

3. Building Training System

In theory, the controller of robot can be designed and numerically simulated by the dynamic equations (1) (2) and collision equation (3). In fact, the theoretical model is too simplified or even the model is unknown to express all information about physical biped robots. According to the theoretical model, the robot model was designed in the simulation environment Vrep. What's more, the theoretical model will guide the selection of the control variables and state variables of the simulation system.

3.1 Building Robot Construction

In this paper, the training robot is built under Vrep. Since the research object is a 2D biped robot, the lateral support is added, thus the robot is only allowed to fall back and forbidden side fall, as shown in Fig. 4. The robot in the simulation environment has three degrees of freedom of mechanism, which are hip joint, and two knee joints that move up and down.

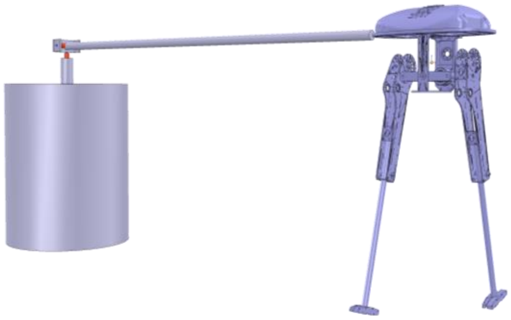


Fig. 4 Robot is walking in simulated environment

From equations (1) (2), it is seen that the robot model has two control variables, the impulsive push P and the hip torque K. The hip joint torque is output through the hip joint motor. And support leg knee joint generates a rapid line motion, the impulsive push is output through the ground reaction to the support legs. In fact, P and K are too difficult to control and measure directly. In this paper, P and K are replaced by new control quantity V_P and V_K which are easy to control and closely related to P, K, V_P indicates the maximum speed of the knee joint allowed by the positional PID when extend the leg to the maximum leg length, and the work done by ground reaction force in a certain time to the support leg to replace the impulsive push work to support leg. V_K indicates the maximum swing speed allowed by the hip joint positional PID during swinging the leg to the maximum angle, and the work done by the hip joint to the swing leg to replace the torque work to swing leg. Table 1 shown the key parameters of the simulation biped robot.

Table 1 Simulation Biped Robot Key Parameters

Parameter	Value (unit)	Parameter	Value (unit)
α	$\frac{\pi}{6}(rad)$	l	$0.52(m)$
M	$5.0(kg)$	m	$0.4(kg)$
V_P	$(0-0.6)(m/s)$	V_K	$(0-7)(rad/s)$

3.2 Robot Control in Simulation Environment

The robot control flow is designed by the scripting language in the simulation environment, and the state-to-action control strategy is fitted by neural network whose parameter obtained through reinforcement learning. The specific control flow of the biped robot gait learning system is:

1. The biped robot provides the state information of the biped robot in the simulation environment to the gait learning system through the API interface.

2. After the learning system obtains the state information, it trains and updates the neural network parameter according to the strategy, calculates the execution action information, and transmits the action result to the biped robot through the API interface.

3. After the robot obtains result, the specific data is calculated and transmitted to the executing agency for execution.

Repeat steps 1-3. The biped robot continuously learns iteratively, and the parameter information of the neural network is continuously iteratively optimized. Finally, the simulated robot can walk stably in the simulation environment.

3.3 Learning System Based on Reinforecement Learning

Robot motion control is continuous motion control, Q-learning just can solve discrete space motion, But DDPG and A3C algorithms can solve continuous and discrete space motion. A3C algorithms uses multi-agent training, and the training speed is better than DDPG algorithms. This paper picks A3C algorithm to build the learning system.

The A3C network consists of an actor network and a critic network, and the input and output vector dimensions of the network need to be determined. Through the foregoing, the state of the robot walking is described by θ, $\dot{\theta}$, φ and $\dot{\varphi}$, the control variables are V_P and V_K. The robot determines the control variables V_P and V_K for this step at the beginning of each step. At the beginning of each step, $\varphi=\alpha$ and $\dot{\varphi}=0$, so φ and $\dot{\varphi}$ can't provide valid information for learning system, the state vector is reduced to 2 dimensions $\boldsymbol{s}=\left(\theta,\dot{\theta}\right)^T$ and the action vector is $\boldsymbol{a}=\left(V_P,V_K\right)^T$. The A3C algorithm belong to random strategy, which does not directly output the value of the action vector, but output the probability of outputting the action vector. Therefore, the output of the action network in this study is 4 dimensional vector, expectations of normal distribution μ_1 and μ_2, variances of normal distribution σ_1 and σ_2. The control variable V_P takes a value according to the normal distribution $N\left(\mu_1,\sigma_1\right)$ in the range of its values, and V_K takes the value according to the normal distribution $N\left(\mu_2,\sigma_2\right)$ in the range of its values, so that the action vector is obtained. The input state vector of the critic network is also $\boldsymbol{s}=\left(\theta,\dot{\theta}\right)^T$, and the output is a scalar V .

The actor network and critic network have two hidden layers except the input layer and the output layer. The layers are fully connected. The hidden layers 1 and layer 2 are composed of 300 and 200 neurons, and the activation function is Rectified Linear Unit(ReLU), and the activation function of

μ_1 and μ_2 is tanh, and the activation function of σ_1 and σ_2 is Softplus. The actor and crtic network share the input layer, and the entire network structure is:

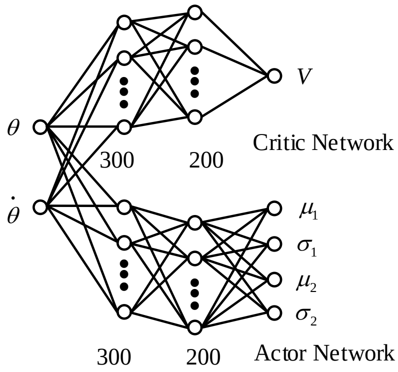
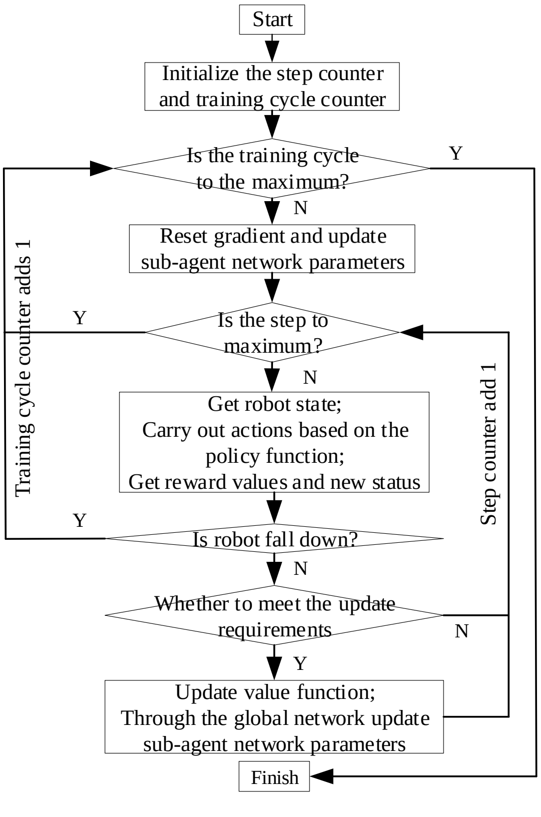


Fig. 5 Actor and critic network structure

The reward function affects the convergence direction of the algorithm. In this study, the reward value of robot will get 1 if the robot finished a successful step, otherwise get 0. The learning system will converge in the direction of the maximum number of steps to continuous walking in simulate environment.



Start
Initialize the step counter and training cycle counter
Is the training cycle to the maximum?
Y
N
Reset gradient and update sub-agent network parameters
Is the step to maximum?
Y
N
Get robot state; Carry out actions based on the policy function; Get reward values and new status
Is robot fall down?
Y
N
Whether to meet the update requirements
N
Y
Update value function; Through the global network update sub-agent network parameters
Finish
Training cycle counter adds 1
Step counter add 1

Fig. 6 Algorithm flow chart

The A3C algorithm is on policy algorithm based on the Advantage Actor Critic(A2C) algorithm. After successively executing several steps, the value function of each step is further reversed, and the estimation of each step is more accurate than Q-learning which is off policy algorithm. The purpose of this study is to enable the biped robot to walk stably and pay more attention to the influence of the neural network on continuous walking, not just only the single step walking. In this study, the robot updates the weight of the network every 10 steps. Fig. 6 shows the flow of the learning algorithm.

4. Training System Experiment Simulation

4.1 Different Control Models

When debugging a robot in an actual project, it is possible to roughly evaluate whether the gait of the robot is stable by the spent time of each step, and the state input vector can increase the spent time of each step Ts. Ts depends mainly on the hip joint angle, so the action output vector can increase the target angle of hip Sa. This paper makes some simulation to analysis the training results under different network control models.

Table 2 shows the state input vector and action output vector information of experiments S1-S4. The target reward value is 150 and the training cycle is 200.

Table 2 Model information of experiment S1-S4

	State Input	Actor Output
S1	$\boldsymbol{s}=\left(\theta,\dot{\theta}\right)^T$	$\boldsymbol{a}=\left(V_P,V_K\right)^T$
S2	$\boldsymbol{s}=\left(\theta,\dot{\theta}\right)^T$	$\boldsymbol{a}=\left(V_P,V_K,Sa\right)^T$
S3	$\boldsymbol{s}=\left(\theta,\dot{\theta},Ts\right)^T$	$\boldsymbol{a}=\left(V_P,V_K,Sa\right)^T$
S4	$\boldsymbol{s}=\left(\theta,\dot{\theta},Ts\right)^T$	$\boldsymbol{a}=\left(V_P,V_K\right)^T$

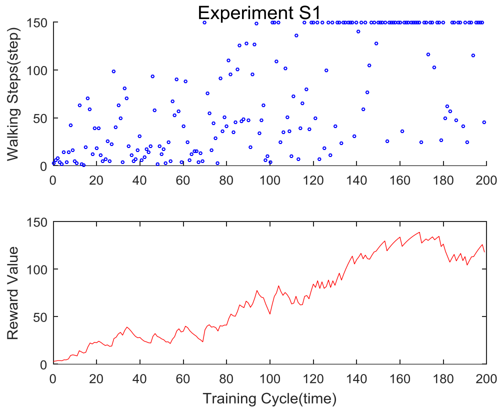


Fig. 7 Steps and reward values of S1

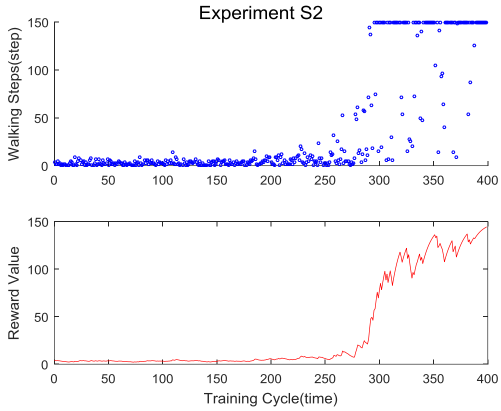


Fig. 8 Steps and reward values of S2

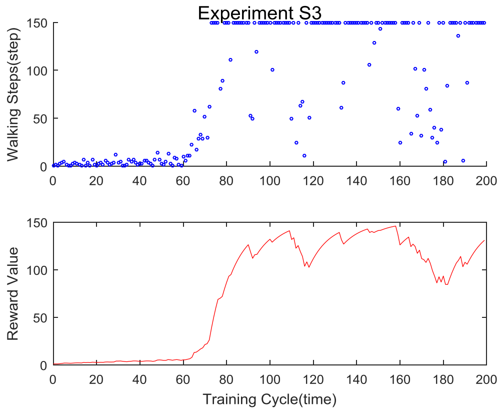


Fig. 8 Steps and reward values of S3

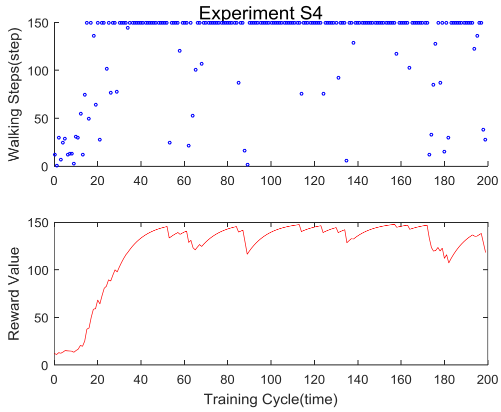


Fig. 10 Steps and reward values of S4

Table 3 Late training data of experiment S1-S4

	Average Reward Value	Variance of Reward Value	Average Steps	Varience of Steps
S1	120.86	094.18	123.37	1904.26
S2	122.16	158.85	126.61	1851.47
S3	122.95	236.29	125.01	1858.27
S4	135.92	098.10	136.15	1174.95

(Note 1. During the training, The training after the reward value reaches 70% of the target reward value for the first time is regarded as late training, which is used to compare the convergence of the training results. Note 2. When the training cycle of experimental S2 is 200, the result is too badly to control the biped robot walking, so the training cycle of experimental S2 is raised to 400.)

As shown in Fig. 7-10 and Table 3, the reward value curve of experiment S4 rises faster than S1, and the reward value curve of experiment S3 rises faster than S2, we can conclude that elevating the state dimension can speed up the training. The reward value curve of experiment S1 rises faster than S2, and the reward value curve of experiment S4 rises faster than S3, we can conclude that appropriately reducing the action dimension can also speed up the training. The experimental S4 reward value curve rises rapidly, and the average reward value, the variance of reward value, the average steps and the variance of steps in the later training were better than S1、S2 and S3, the average reward value, the variance of reward value, the average steps and the variance of steps can explain the stability of training. The training speed and stability are improved obviously, the best training results have been obtained. It is valid that increasing the spent time of each step to evaluate the robot's status information.

4.2 Change Weight Update Cycle

From preceding text, the model of experiment S4 can get best results, this section will explore how to further improve the stability of the later training?

During the training, when the reward value function has achieved greater reward value and good train results. The network parameters can make the biped robot walk stably for a long time. The update cycle can be increased to observe the influence of network parameters on the final stable walking. This study will build an update cycle function related to the reward value.

The update cycle of experiment B is 10, the network updates it's parameters every 10 steps. the update cycle of the experiment A is:

$$\begin{cases} 10 & Cv \le Tv*0.8 \\ (10+Cv-Tv*0.8) & Cv > Tv*0.8 \end{cases} \quad (5)$$

Cv is the current reward value, Tv is the target reward value. In order to better compare the data of experiments A and B, the target reward value of this experiment is 250, and the training period is 400.

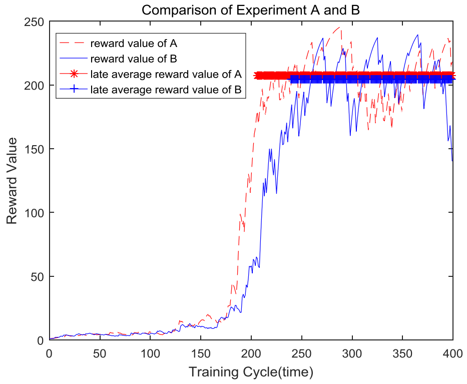


Fig. 11 Change weight update cycle

Table 4 Late training data of experiment A and B

	Average Reward Value	Variance of Reward Value	Average Steps	Varience of Steps
A	207.42	6354.71	208.73	6354.71
B	204.53	7285.03	202.19	7285.03

Fig. 11 shown that the reward value and the late average reward value of experiment A、B. Combining Fig. 11 and Table 4, we can conclude that the reward value of experiment A rises faster than B, and the training data of experiment A is more stable than B by comparison of the late training data. The strategy that the network weight update cycle changes with the reward value can get better training results than the fixed update cycle.

4.3 Verification of Training Results

After many experiments, the neural network trained by experiment A can realize that the 2D biped robot stably walk more than 5500 steps in the simulation environment. Figure 12 shows the spend time of every step under the control of trained network. The spend time of each step convergences between $600ms$ and $700ms$, showing a certain periodicity. At finally, when the 2D biped robot can walk in the simulate environment, and the walking speed is about $1.5km/h$.

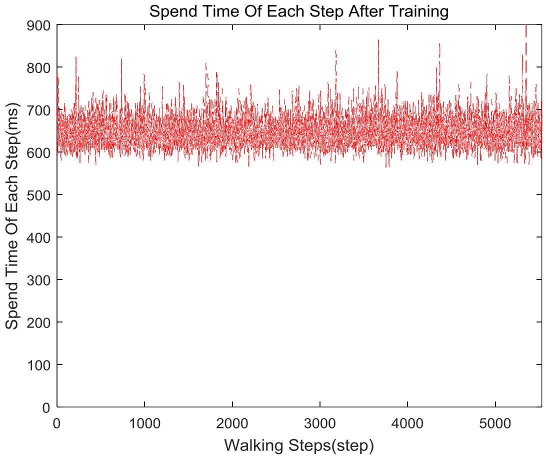


Fig. 12 Robots spend time in each step after training

5. Conclusion

In the simulation environment, the biped robot model with hip drive and impulsive push is built. The gait training is carried out on the model by the reinforcement learning A3C algorithm. The training result under different network models is analyzed. the stability of the training result is improved by constructing an update cycle function related to the reward value. And finally, the trained network can control the robot to walk stably in the simulation environment. Based on the research of this paper, the construction of physical objects and experiments will be further research.

Acknowledgements

Thanks to the laboratory platform and the guidance provided by the instructor.

References

[1] Lin J L, Hwang K S, Jiang W C, et al. Gait Balance and Acceler-ation of a Biped Robot Based on Q-Learning[J]. IEEE Access, 2016, 4:2439-2449.

[2] Hwang K S, Lin J L, Yeh K H. Learning to adjust and refine gait patterns for a biped robot[J]. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 2015, 45(12): 1481-1490.

[3] Gu S, Holly E, Lillicrap T, et al. Deep reinforcement learning for robotic manipulation with asynchronous off-policy updates[C]//2017 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2017: 3389-3396.

[4] Zhou C, Meng Q. Reinforcement learning with fuzzy evaluative feedback for a biped robot[C]//Proceedings 2000 ICRA. Millennium Conference. IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No. 00CH37065). IEEE, 2000, 4: 3829-3834.

[5] Feirstein D. S., Koryakovskiy I., Kober J, et al. Reinforcement Learning of Potential Fields to achieve Limit-Cycle Walk-ing[J].Ifac Papersonline, 2016, 49(14):113-118.

[6] Jin Y, Lee M. Enhancing Binocular Depth Estimation Based on Proactive Perception and Action Cyclic Learning for an Autonomous Developmental Robot[J]. IEEE Transac-tions on Systems, Man, and Cybernetics: Systems, 2018, 49(1):169-180.

[7] Chen Qishi. Study and Implement of Reinforcement Learning in Biped Robot Balance Control [D]. South China University of Technology, 2016.

[8] Mcgeer T. Passive Dynamic Walking[J]. The International Journal of Robotics Research, 1990, 9(2):62-82.

[9] Garcia M, Chatterjee A, Ruina A, et al. The simplest walking model: stability, complexity, and scaling[J]. Journal of biomechanical engineering, 1998, 120(2): 281-288.

[10] Collins S, Ruina A, Tedrake R, et al. Efficient bipedal robots based on passive-dynamic walkers[J]. Science, 2005, 307(5712): 1082-1085.

[11] Nie Xiuhua, Chen Weishan, Liu Junkao, etc. The influence of spring stiffness on the stability of passive walking [J]. Chinese Journal of Theoretical and Applied Mechanics, 2010, 42(3):541-547.

[12] Zhang Qizhi, Zhou Yali. Semi-passive walking of biped robot with an impulsive push action [J]. Journal of Southeast University (Natural Science Edition),2013, 43(s1):102-106.

[13] Zhou Yali, Zhang Qizhi. Model-free neural network control for quasi-passive biped robots based on impulsive push [J]. Application Research of Computers,2018, 35(01):56-61.