Balance Control of an Inverted Pendulum on a Quadruped Robot by Reinforcement Learning

Balance Control of an Inverted Pendulum on a Quadruped Robot by Reinforcement Learning

Shangyu Wu [1a*], Xianqing Lei[1b], Linqi Ye [2c]

[1]School of Mechatronics Engineering, Henan University of Science and Technology, Luoyang 471003, China

[2]Center for Artificial Intelligence and Robotics, Graduate School of Shenzhen of Tsinghua University, Shenzhen 518055, China

[a*]wushangyu2021@163.com, [b]gyy_guoyangyang@163.com, [c]a190615@163.com

Abstract—When a quadruped robot crosses some uneven terrain, its stability is a very important consideration. In practical applications, it is often difficult for researchers to obtain accurate models of quadruped robot systems and need quadruped robots to complete high-stability tasks. Balancing an inverted pendulum on a quadruped robot is a good way to test their balancing ability, which is of great research and practical value. According to the nonlinear, uncertain, and strong coupling characteristics of the quadruped robot inverted pendulum system, this paper proposes a quadruped robot balancing inverted pendulum algorithm based on reinforcement learning, the Q-learning algorithm. The advantage of this algorithm is that it can learn effective balancing policy directly from experience, and it does not depend on the accurate model of the quadruped robot. It has the characteristics of high efficiency and flexibility. Through a lot of training, it can break through the performance limit brought by model error to traditional methods. In this paper, a comparative experiment of a set of balanced inverted pendulums of quadruped robots with different sizes is completed in the V-REP simulation software. The experimental results show that the algorithm can effectively improve the balance ability of quadruped robots, and it also shows that the algorithm has good adaptability.

1. Introduction

Quadruped robots which are manufactured according to the bionics principle have great potential in complex terrain reconnaissance, field material transportation[1], and so on, and have become a hot spot in the robotics research. This may be because legged robots can adapt to complex terrain better than wheeled and tracked robots[2]. In all the research fields of quadruped robots, stability[3] is an important aspect, because if there is no stable state, the robot cannot walk in various terrains or complete specific tasks.

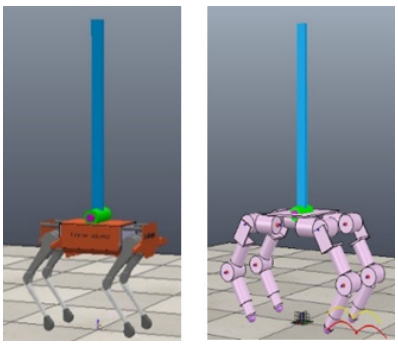
The inverted pendulum system has rapidly become a classic system in the field of control theory since it was designed and experimented with by MIT in the 1950s. The mathematical modeling and stability control of inverted pendulum, a typical nonlinear system, has always been an enduring hot issue in the field of control. Various inverted pendulum systems have correspondingly become experimental beds for testing the feasibility and robustness of control theory. The inverted pendulum is a nonlinear, strongly coupled, and unstable system[4], and its stability control is a typical problem in control theory. In recent years, reinforcement learning has developed rapidly in the field of inverted pendulum control. Li[5] introduced a manifold regularization reinforcement learning scheme for continuous Markov decision processes. Liu[6] combines the online least-squares policy iteration method with the empirical

playback method, stores the online generated samples, and updates the control policy with the least-squares method, which is applied to the inverted pendulum system. And Zhang Rong and Chen Weidong[7] realized the swing-up and balance control of inverted pendulum by the method of subsection task. Studying the balance control problem of inverted pendulum systems is of great significance to analyze and solve the control problem of nonlinear systems[8-9], so the system has been widely used in checking the effectiveness of control algorithms.

In this paper, a balanced inverted pendulum system of quadruped robot is designed to test the robustness and stability of quadruped robot, which makes quadruped robot balance an inverted pendulum automatically by reinforcement learning. Given the shortcomings of traditional control methods, such as low efficiency and difficult parameter setting, this paper proposes a balanced inverted pendulum algorithm for quadruped robots based on reinforcement learning Q-learning algorithm. By changing the parameters, the balancing algorithm can be easily applied to robots of different sizes. The main contributions of this paper are as follows: (1) Taking stability as the primary factor, a control algorithm for the balanced inverted pendulum of the quadruped robot is designed. This method is based on a reinforcement learning algorithm, and through interaction with the environment, it constantly tries and learns from scratch and does not need to accurately model the working environment of the robot, so it is flexible and efficient. (2) The parameters of the algorithm are easy to adjust, which can be widely used in different quadruped robots, and has strong robustness, which allows robots to perform tasks requiring high stability.

2. Problem Formulation

We constructed the balanced inverted pendulum models of two quadruped robots used in this paper, Model 1 and Model 2, as shown in Fig.1. The quadruped robot models are mainly composed of a body and four identical legs connected to the body. Each leg has three joints, including hip joint, elbow joint and wrist joint. Each leg includes two parts: an upper and a lower segment. The upper leg is connected to the body through a hip joint for lateral movement and an elbow joint for forward and backward movement, and the lower leg is connected to the upper leg through the wrist joint for forward and backward movement. The back of the quadruped robot model is connected with the vertical pole through a passive pivot joint. Our goal is to prevent the vertical pole from falling by controlling the back and forth movement of the quadruped robot. Model 1 is a simulation model of Laikago from Unitree Robotics Company. Model 2 is a simulation model based on the real robot in our laboratory, and its key mechanical parameters are listed in Table 1. We will design and verify the basic balancing algorithm on model 1, and verify the adaptability of the balancing algorithm on different quadruped robots on model 2.



(a) Model 1 (b) Model 2

Fig. 1 The simulation quadruped robot-inverted pendulum system in V-rep

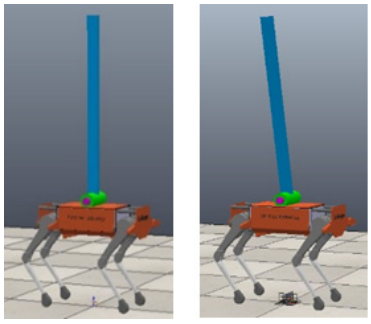
Table 1. Mechanical parameters of the simulation model 1 and model 2

	model 1	model 2
Body length	0.37m	0.40m
Body width	0.25m	0.30m
Body height	0.52m	0.58m
Upper leg length	0.25m	0.30m
Lower leg length	0.32m	0.27m
Total mass	20kg	25kg
Maximum joint angular velocity	15deg/s	17deg/s
Pole length	0.60m	0.80m

3. Experimental environment and algorithm design

3.1. Define the input and output of the simulation environment

We use the reinforcement learning to solve the problem of balancing an inverted pendulum on a quadruped robot. Fig. 2(a) shows the balanced state of the inverted pendulum, and (b) shows the unbalanced state of the inverted pendulum.



(a) Balanced state (b) Unbalanced state

Fig. 2 The simulation quadruped robot-inverted pendulum system in V-rep

In this paper, the inverse kinematics module in the V-REP simulation software is used. By giving the body position of the quadruped robot, the joint angles required by the 12 joints in the legs of the quadruped robot can be automatically calculated, thus controlling the quadruped robot to move to the specified position. The input and output of the quadruped robot-inverted pendulum system are defined by state and action, so the state and action of the quadruped robot-inverted pendulum system are designed. In this balancing control problem, the forward and backward movement of the quadruped robot is defined as the action A of the quadruped robot-inverted pendulum system, as shown in Eq. (1), which contains two-dimensional parameters.

$$A = \left[\Delta a_1, \Delta a_2\right] \tag{1}$$

Where: Δa_i is the position increment of the quadruped robot.

The state S of the quadruped robot-inverted pendulum system is shown in Eq. (2), which contains four-dimensional parameters.

$$S = \left\{x, v, c, w\right\} \tag{2}$$

Where: x is the displacement of quadruped robot, v is the speed of quadruped robot, c is the included angle between pole and vertical direction, and w is the angular velocity of the pole.

In this paper, it is stipulated that during reinforcement learning training, the termination conditions of each episode are (1) the included angle between the pole and the vertical direction is greater than or equal to 4.1^0, or (2) the forward or backward displacement of the quadruped robot is greater than or equal to 4.2cm, or (3) the episode length is greater than 500. For each step in the episode, the agent (quadruped robot-inverted pendulum system in this paper) will receive a reward, otherwise, the reward will be 0. When the average reward for 50 consecutive episodes reaches 400, it is considered that the agent has obtained a better control policy. The reward function is set to:

$$\mathrm{Re}\,ward = \begin{cases} 1, \ if \ \ |x| < 4.2 \ or \ \ |c| < 4.1 \\ 0 \qquad\qquad other \end{cases} \tag{3}$$

3.2. Balanced algorithm design based on Q-learning

Our task is to make the pole on the quadruped robot keep balance by moving back and forth. This process can be understood as the continuous interaction between the agent and the environment, taking actions according to the current state, and recording the rewards from the environmental feedback, so that the best action can be taken when the agent reaches the same state next time.

In this paper, a balanced inverted pendulum algorithm of quadruped robot based on reinforcement learning Q-learning algorithm is adopted. In the Q-learning algorithm, Q is $Q(s,a)$, which is the expectation that taking action a can get benefits in the state s at a certain moment, and the environment will feed back corresponding reward rewards according to the actions of the agent. therefore, the main idea of the algorithm is to construct the state and action into a Q-table to store the Q-value, and when the same state is reached next time, select the action that can get the maximum benefits according to the Q-value. Our goal is to explore all possible combinations of states and actions, update these Q function values through iterative process, and finally get a perfect Q-table to achieve the control of agents. Since Q-learning is suitable for discrete observation space, the continuous state values in this task are discretized into discrete state values.

The main steps of the balancing algorithm are as follows:

（1）Firstly, the policy of ϵ-greedy is introduced, ϵ is the exploration rate, and the initial value is 1 with the training, ϵ gradually decreases to the minimum value of 0.1, and will not change. Select actions randomly from the action set with the probability of $1-\epsilon$, that is, the quadruped robot moves forward or backward, and select the action with the highest reward corresponding to the current state from the Q-table with the probability of ϵ. The mathematical expression is:

$$a(s) = \arg\max_{a'} Q(s,a') \tag{4}$$

（2）After taking the above actions, the state of the quadruped robot-inverted pendulum system will change, and the reward value will be obtained according to the reward function.

（3）After obtaining the new state and reward, use the following formulation to update the Q-table:

$$Q(s,a) = Q(s,a) + \alpha\left[R(s,a) + \gamma \max_{a} Q(s',a) - Q(s,a)\right] \tag{5}$$

Where α is the learning rate, with an initial value of 1, which decreases monotonously from 1 to 0.1 with the training, γ is the discount factor, and the functions of these key parameters will be described in detail later.

（4）Update the current state and repeat the above steps in the next iteration.

Theoretically, if the learning rate α meets certain conditions, any state-action pair will converge to the optimal solution after infinite iterations using Eq. (5), and then the optimal policy can be obtained:

$$\pi^*(s,a) = \arg\max_{a} Q^*(s,a) \tag{6}$$

3.3. Key parameter analysis

To make more exploration and avoid falling into local optimum, we introduce ϵ-greedy policy, with ϵ as the exploration rate. Every time we move the quadruped robot, we select the action with the largest Q value in the current state from the Q-table with the probability of ϵ and randomly select the action with the probability of $1-\epsilon$, so that we can ensure enough iterations. At the same time, an attenuation function is used to reduce ϵ, because, with the progress of exploration, the exploration space of agents will be wider and wider, so the scope of exploration should be continuously narrowed, so that agents tend to choose the best action verified by previous experience.

The learning rate α is used to weigh the results of the last study and the results of this study. From Eq. (5), it can be seen that the smaller the value, the more the agents attach importance to the previously learned policy, and the larger the value, the more they attach importance to the newly learned policy. In this paper, the initial value is set to 1, which decreases monotonously to 0.1 with the training, which can not only speed up the convergence of the algorithm but also help the algorithm converge to the global optimal solution.

The discount factor γ is the attenuation value of future rewards. It can be seen from Eq. (5) that the larger the discount factor is, the more the agent pays attention to the future rewards, and the smaller the discount factor is, the more attention is paid to the current rewards. This paper will set it to 0.99, which can fully consider the influence of future rewards when updating the Q-table.

One advantage of this balancing algorithm is that it is versatile, and it is not only designed for specific robots. When applied to different robots, better learning results can be achieved only by setting appropriate exploration rate ϵ, learning rate α and discount factor γ. The values of key parameters of simulation experiments in this paper are shown in Table 2.

Table 2. Values of key parameters for the simulation.

Parameter	Value
Discount factor γ	0.99
Learning rate α	1
α_{min}	0.1
Exploration rate ε	1
ε_{min}	0.1

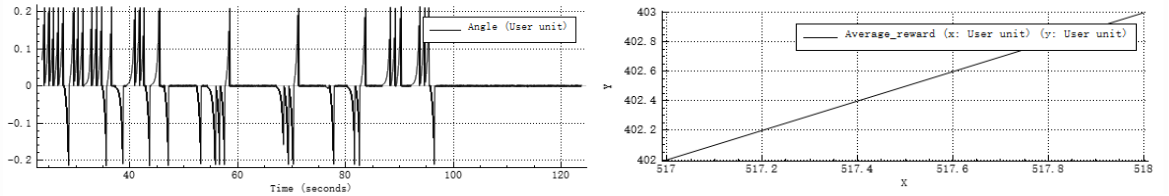
4. Simulation results

4.1. Simulation experiment and analysis in model 1

To verify the feasibility of this method, we use the Newton physics engine to simulate in V-REP software.

We will simulate these two simulation models separately. Firstly, the feasibility of the algorithm will be verified on Model 1, and the convergence speed of the algorithm will be observed by adjusting the parameters ϵ and α. Secondly, the balancing algorithm will be applied to model 2, and its structure and size are quite different from those of model 1. The application of the balancing algorithm in model 2 can verify the adaptability of the algorithm in different robots.

The algorithm is verified in V-REP simulation software. The simulation results are shown in Fig. 3 (a), the horizontal axis represents time, and the vertical axis represents the angle between the pole and the vertical direction. The parameters ϵ and α are preset values, namely $\epsilon_{min} = 0.1$ and $\alpha_{min} = 0.1$. It can be observed that the algorithm of the quadruped robot-inverted pendulum system gradually converges after 100 episodes of training, and the angle between the pole and the vertical direction can be stabilized within the specified range. Fig. 3(b) shows the number of episodes on the horizontal axis and the average award on the vertical axis. It can be seen that after 518 episodes of training, the average reward reaches 403 points, and it can be considered that the quadruped robot-inverted pendulum system has achieved a better balance control policy. This fully reflects the effectiveness and rapidity of the balancing algorithm.



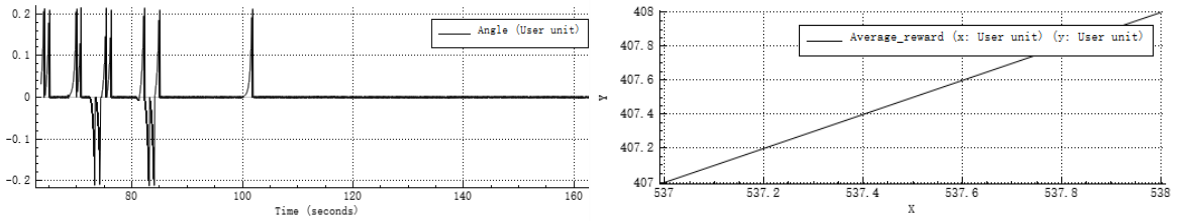
(a) Variation curve of angle (b) Change curve of average episode reward

Fig. 3 Simulation results for Robot 1

4.2. Balance algorithm adaptability verification and analysis in model 2

After adjusting the parameters ϵ and α several times and comparing the balance effect on model 2, the algorithm achieved faster convergence speed when the parameters ϵ and α were finally set to $\epsilon_{min} = 0.01$ and $\alpha_{min} = 0.3$.

The simulation results in V-REP are shown in Fig. 4. It can be observed from Fig. 4 (a) that after 100 episodes of training, the algorithm gradually converges, and the included angle between the pole and the vertical direction can be kept within the specified range. It can be seen from Fig. 4 (b) that after 538 episodes of training, the average reward reaches 408 points, and it can be considered that the quadruped robot-inverted pendulum system has also obtained a better balance control policy. It can be seen from this that our algorithm can accomplish different tasks of balancing the inverted pendulum of a quadruped robot. Therefore, the applicability of this method on different robots is fully proved.



(a) Variation curve of angle (b) Change curve of average episode reward

Fig. 4 Simulation results for Robot 2

5. Conclusion

In this paper, a balancing control algorithm for a quadruped robot-inverted pendulum system based on reinforcement learning is proposed to improve the stability of a quadruped robot. For different robot configurations, the key parameters such as learning rate α, discount rate γ, and exploration rate ϵ can be adjusted to achieve better balancing control. The algorithm can directly learn effective control policy from experience. Compared with traditional control methods, this method does not depend on accurate models and has the characteristics of high efficiency and flexibility.

In V-REP simulation software, simulation on Robot 1 shows that the algorithm can improve the stability of the robot. In another simulation, the algorithm is applied to Robot 2, and a good balance effect is obtained. This shows the adaptability of the algorithm to different robots and proves that the balancing algorithm is insensitive to the size of robots. In the future, this balancing algorithm is expected to be applied to more complex environments to improve the adaptability of robots.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Project No.51775172).

References

[1] Chen, X., Yu, Z., Zhang, W., Zheng, Y. (2017) Bioinspired Control of Walking with Toe-off, Heel-strike and Disturbance Rejection for a biped robot. IEEE Transactions on Industrial Electronics, 64: 7962-7971.

[2] Kolter, J.Z., Polegers, M.P. (2008) A control architecture for quadruped locomotion over rough terrain. IEEE International Conference on Robotics and Automation (ICRA), Boston. pp: 811-818.
[3] McGhee, R.B., Frank, A.A. (1968) On the stability properties of quadruped creeping gaits. Mathematical Biosciences, 3:331-351.
[4] Barkat, A., Hamayun, M.T. (2018) Model identification and real-time implementation of a linear parameter–varying control scheme on lab-based inverted pendulum system. Proceedings of the Institution of Mechanical Engineers, Munich. pp: 168-181.
[5] Li, H., Liu, D. (2018) Manifold regularized reinforcement learning. IEEE Transactions on neural networks and Learning Systems, 29(4): 932-943.
[6] Liu, Q., Zhou, X. (2014) Experience replay for least-squares policy iteration. IEEE/CAA Journal of Automatica Sinica, 1(3): 274-281.
[7] Zhang, R., Chen, W. (2004) Whole process control of swing up and balance of inverted pendulum based on Reinforcement Learning. Systems engineering and electronic technology, 52(2): 72-76.
[8] Yi, L., Zhang, R. (2021) Research on self-control experimental platform based on straight inverted pendulum. Experimental Technology and Management, 38(1): 99–104.
[9] Yu, S., Chu, J., Wang, Y. (2020) Output feedback control of first-order rotating inverted pendulum. Experimental Technology and Management, 37(3): 165–170.