Learning to Control Two-Wheeled Self-Balancing Robot Using Reinforcement Learning Rules and Fuzzy Neural Networks

Xiaogang Ruan
Beijing University of Technology Institute of Artificial Intelligence and Robotics
adrxg@bjut.edu.cn

Jianxian Cai
Beijing University of Technology Institute of Artificial Intelligence and Robotics
cjxlaq@163.com

Jing Chen
Beijing University of Technology Institute of Artificial Intelligence and Robotics
chenjing@bjut.edu.cn

Abstract

This paper present a novel method to control the balance of a two-wheeled robot by using reinforcement learning and fuzzy neural networks(FNN) which can guarantees the convergence and rapidity when the model of the robot is not available and the agent has no a prior knowledge. Furthermore it can effectively control the task of continuous states and actions. The simulation and experiment results demonstrate that it not only can learn to control the two-wheeled robot system in a short time, but also maintain the balance of two-wheeled robot when the parameters of two-wheeled change a lot.
Key words: *reinforcement learning; FNN; two-wheeled robot; balance control.*

1. Introduction

The two-wheeled self-balancing robot has become a hot subject in verifying various control theories since it occurred, which mainly due to its unstable dynamic performance and strong nonlinearity. The ability of self-learning and self-adaptive the robot possess has become a hot subject recently. Reinforcement learning (RL) has attracted researchers' attention during last few decades. Compared with Supervised Learning (SL), RL does not need a supervisor which teaches the learning agents how to do. But the practical application of RL has some questions such as the low convergence speed which will perhaps difficult to realization when there are more number of state space and action space. The effective way to solve the problem is using function approximation base on fuzzy inference system (FIS) or neural networks (NN) [1], which can approximate the mapping from state space to action space.

There have been many researches on realization of RL algorithm based on FIS [4] and NN [2, 3]. NN have merits of strong fault tolerant and adaptive learning, however it can't use priori knowledge which makes the networks have long learning time and hard converge to global extreme value. However FIS can fully apply priori knowledge and its inference pattern accordance with the thinking mode of human, but it has poor ability of self learning and self-adaptive.

The reference about two-wheeled robot self-balance control using RL based on FIS and NN is still few now, but there have many achievements in inverted pendulum balance control. Barto[5] designed two single layer neural networks in 1983 which adopt AHC(Adaptive Heuristic Critic) learning algorithm to realize inverted pendulum balance control with state discrimination. Anderson [6] use two double layer neural networks and AHC in 1989 to realize inverted pendulum balance control with continuous state. Guofei Jiang [7] combined Q learning with NN in 1998 to realize single inverted pendulum system control.

In this paper we combined FIS with NN to compose Fuzzy Neural Network (FNN) with which to approximate Q function. The scheme was applied in two-wheeled robot balance control to solve the reinforcement learning problem of robot with continuous state space and action space. The experiment and simulation demonstrate that the scheme effectively realizes two-wheeled robot balance control and have strong robustness.

2. Q-learning

Q learning proposed by Watkings is a reinforcement learning algorithm which independent with model. Reinforcement learning is an interaction-based paradigm wherein an autonomous agent learns to adjust its behavior according to feedback received from the environment. The learning paradigm is consistent with the notion of embodied cognition that intelligence is a process deeply rooted in the body's interaction with the world. Often formalized as a Markov decision process (MDP), an autonomous agent performs reinforcement learning through a sense, act, and learn cycle. First, the agent obtains sensory input from the environment representing the current state (S). Depending on the current state and its knowledge and goals, the system selects and performs the most appropriate action (A). Upon receiving feedback in terms of rewards (R) from the environment, the agent learns to adjust its behavior in the motivation of receiving positive rewards in the future.

978-0-7695-3304-9/08 $25.00 © 2008 IEEE
DOI 10.1109/ICNC.2008.361

IEEE computer society

Q-learning is an important algorithm of RL. In -learning, the idea is to directly optimize -function, which can be calculated recursively. The function of is to represent the evaluation of state-action. -learning is given by:

$$Q(s_t, a_t) = (1-\alpha)Q(s_t, a_t) + \alpha\left(r_{t+1} + \gamma \max_a Q(s_{t+1}, a_t)\right) \quad (1)$$

where (s_t, a_t) is a state-action pair at time step t . a_t denotes one of the possible actions chosen in current state. s_{t+1}expresses the new state when the action has been applied to the state . r_t represents the receipt of the reinforcement signal. The value of α is a learning rate; γ denotes the discount factor.

This means only perform iterative learning of current Q-function the global optimal sequence actions can be selected. The equation (1) updates the current Q-function based on the evaluation value of next state, which is noted one-step Q-learning. When achieve the convergence of Q-function, the optimal policy can be confirmed.

3. Structure of FNN based on RL

3.1. Structure of control system

In this paper the architecture of FNN based on reinforcement learning is depicted in Fig.1.

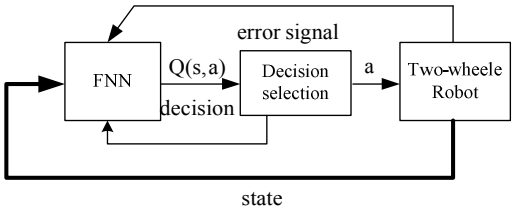

Figure1. Structure of control system

The input of FNN is decision a and state vector $X = [\dot{\theta}_L \quad \dot{\theta}_R \quad \theta \quad \dot{\theta}]$, the state vector denote respectively: Left angular velocity; right angular velocity; dip angle and dip angular velocity. The output of FNN is value function Q(s,a). It is obviously that the FNN based on RL represent the mapping from a pair of state-action (s_t,a_t) to Q-value.

The work process is as follow: The two-wheeled robot firstly observes the current state s_t at time step t, and then select an action a_t according to certain explore strategy based on the actual output of FNN. After that the robot check whether there is error signal r_t by observing the subsequent state s_{t+1} of the robot. In the end the Q-function is updated according to the equation (1). The fuzzy rules are updated based on the error signal r_t, which will make the actual output of FNN approximate the perfect value $Q^*(x_t,a_t)$.In the end going to the state s_{t+1} and cycling the process as above.

3.2. Structure of FNN

FNN employs six-layer architecture (see Fig.2).

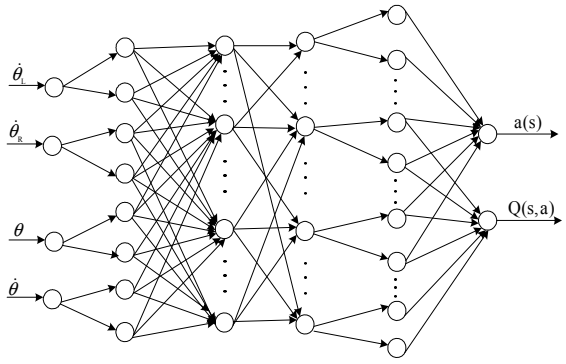

Figure2. Structure of FNN

The first layer which has four nodes is regarded as input layer, and the input signal is state vectors X. The first layer's weight is 1 as the same with the second layer, so the first layer doesn't need process.

The second lay which has eight nodes is the fuzzy processing of input state layer. The output of the layer is the membership degree of fuzzy sets. The fuzzy value of dip angular is "overstep when the angle is positive" and "overstep when the angle is negative". The fuzzy value of dip angular velocity is "overstep when the angle velocity is positive" and "overstep when the angle velocity is negative". The fuzzy value of left and right angle velocity is "overstep when the level angle is positive" and "overstep when the level angle is negative". All of the fuzzy variables are fuzzed by adopting Guss function,

$$\mu_i^j = \exp\left[\frac{-(x_i - c_{ij})^2}{\delta_{ij}^2}\right] \quad (2)$$

where c_{ij} is a mean, δ_{ij} is a variance, and μ is a fuzzy membership degree.

The third layer is the rules layer. The fuzzy control adopts 16 rules and the layer is applied to compute the activation degree of each rule.

$$a_j = \min(\mu_1^{i1}, \mu_2^{i2} \cdots) \quad (3)$$

The fourth layer is the normalization layer.

$$\bar{a}_j = a_j \Big/ \sum_{i=1}^{m} a_i \quad (4)$$

The fifth layer is action selection layer, and the form of the fuzzy rule is given as following:

R_j: If s is F_j Then y is $a(j,1)$ with $q(j,1)$
or y is $a(j,2)$ with $q(j,2)$
……

or y is $a(j, i)$ with $q(j, i)$

where, R_j is the fuzzy rule of FNN. S denotes the current state. Fj describes a fuzzy set. $a(j,i)$ and $q(j, i)$ are the consequence part of fuzzy rules. $a(j,i)$ is the possible actions that can be selected in state s, $q(j, i)$ express the q-value of action $a(j,i)$.

The final layer is the output layer. In this paper adopt ε-greedy policy to select the action of FNN in the RL process. Then the output of FNN is given by following:

$$Q(s,a)=\sum_{j=1}^{N}\bar{a}_j(s)\times q(j,i^*) \tag{5}$$

$$a(s)=\sum_{j=1}^{N}\bar{a}_j(s)\times a(j,i^*) \tag{6}$$

We adopt Q-learning to update the value $q(j, i)$. The agent perform the output action at (s_t) of FNN and transfer to the state s_{t+1} and receive a reward r_t *from* the environment.

The temporal error term is computed by

$$\varsigma = r_t + \gamma \max Q(s_{t+1},a_t) - \max Q(s_t,a_t) \tag{7}$$

$$\xi = r_t + \gamma \max Q(s_{t+1},a_t) - Q(s_t,a_t) \tag{8}$$

The above value then is used to update the q value action, compute the gradient (j, i) firstly by

$$\Delta q(j,i)=\begin{cases}\delta\cdot[\xi+\varsigma\cdot e_t(j,i)] & \text{if } i=i^*\\ \delta\cdot\varsigma\cdot e_t(j,i) & \text{else}\end{cases} \tag{9}$$

where δ is learning rate, and we will use accumulated Eligibility Trance (ET) to accelerate the RL, ET is update by

$$e(j,i)=\begin{cases}\gamma\cdot\lambda\cdot e_t(j,i)+1 & \text{if } i=i^*\\ \gamma\cdot\lambda\cdot e_t(j,i) & \text{else}\end{cases} \tag{10}$$

where λ is the learning rate of ET.

At last the q value is update by following:

$$q_t(j,i)=q_{t-1}(j,i)+\Delta q(j,i) \tag{11}$$

4. Experiments and Simulations

Experiment based the two-wheeled robot which is developed by institute of Artificial Intelligence and Robotics of Beijing University of Technology (see Fig.3).

The internal reinforcement r *is* given 1which express a success and given 0 on the contrary which express a failure. Given that

$$r=\begin{cases}-1 & \theta \succ 10^\circ\\ 0 & \theta \prec 10^\circ\end{cases} \tag{12}$$

Figure3. Two-wheeled robot

The Q-learning rate parameter α is set to 0.2 ; the discount factor γ is set to 0.95; and the mean μ and variance σ of Gauss noise $\xi(\mu,\sigma^2)$ is given 0 and 0.1 respectively. We will stop agent learning and renew the experiment when the failure number over 10000 trials at every experiment. If the failure number doesn't exceed 10000 trials at every experiment, we think that the two-wheeled robot can maintain balance successively.

Firstly, we simulate the dip angular of robot after we give robot a initial dip angular 1 *rad*. Compare to Fig.4, Fig.5 shows the robot can successfully maintain balance based on both PID control and FNN control, however the latter's robust performance is much better.

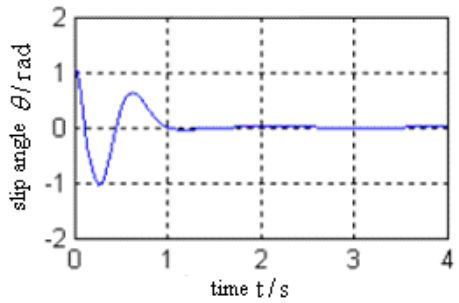

Figure4. PID control result of slip angle

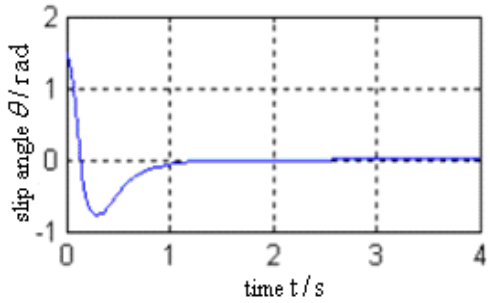

Figure5. FNN and RL control result of slip angle

Secondly we analyze the adaptive ability when the parameter of robot such as gravity center or friction coefficient is changed. Compare to Fig.6, Fig.7 shows that the dip angular has a long process of transition and fluctuation but in the end the robot can maintain balance

based on both PID control and FNN control, however the latter's has fast speed and better robust performance.

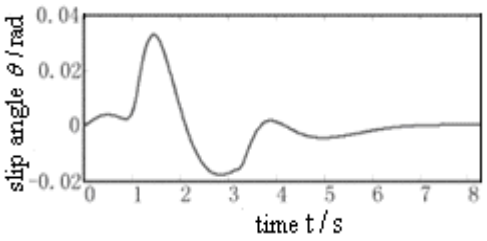

Figure6. PID control result of slip angle

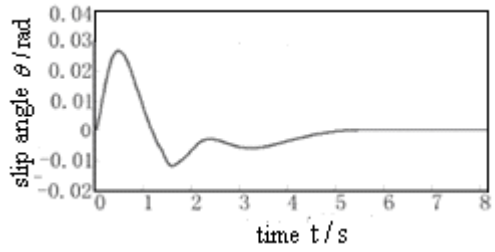

Figure7. FNN and RL control result of slip angle

5. Discussion

This paper presents a novel method to control the balance of a two-wheeled robot by using reinforcement learning and fuzzy neural networks (FNN). The FNN architecture based on RL is used to approximate Q-value. It can guarantees the convergence and rapidity when the model of the robot is not available and the agent has no a prior knowledge and also can effectively control the task of continuous states and actions. The simulation and experiment results demonstrate that it not only can learn to control the two-wheeled robot system in a short time, but also maintain the balance of two-wheeled robot when the parameters of two-wheeled change a lot. The simulation and experiment also express the performance of the two-wheeled robot about speed and robust based on FNN is better than the classical PID control.

References

[1] V.Y. Glizer. Homicidal chauffeur game with target set in the shape of a circular angular sector: Conditions for existence of a closed barrier. *Journal of Optimization Theory and Applications,* 101(3):581-598, 1999.

[2] M. Guelman, J. Shinar, and A. Green. Qualitative study of a planar pursuit evasion game in the atmosphere. *Journal of Guidance, Control and Dynamics*, 13 (6): 1136-1142, 1990.

[3] Turet sky V, J. Shinar. Missile guidance laws based on pursuit evasion game formulations. Automatica, 39 (4): 607-618, 2003.

[4] H.R. Berenji, and P. Khedkar. Learning and tuning fuzzy logic controllers through reinforcements. *IEEE Transactions on Neural Networks,* 3 (5): 724 – 740, 1992.

[5] A.G. Barto, S. Sutton, and C.W. Anderson. Neuron like adaptive elements that can solve difficult learning control problems. *IEEE Trans. on Systems, Man, and Cybernetics,* 13 (5): 834 -846, 1983.

[6] C.W. Anderson. Learning to control an inverted pendulum using neural networks. *IEEE Control System Magazine*, 9(3):31-37, 1989.

[7] Guofei Jiang. Learning to control an inverted pendulum using Q-learning and neural networks. *Journal of the automation*, 24 (5) : 662-667, 1998.

[8] H.J. Kelley, E. M. Cliff, and F. H. Lutze. Pursuit evasion in orbit. *Journal of the Astronautical Sciences,* 29 (3):277-288, 1981.

[9] P. Watking, and Dayan. Q-learning. Machine learning 8(3): 279-292, 1992.

[10] Ah-Hwee Tan, Ning Lu and Dan Xiao. Integration temporal difference methods and self-organizing neural networks for reinforcement learning with delayed evaluative feedback. IEEE Transactions on Neural Networks, 19(2): 230-244, 2008.