Learning Control of Manipulator with a Free Joint

T.Goto, H.Lee, K.Abe
Dept. of Elec. & Comm. Eng.
School of Eng.,Tohoku Univ.
Sendai, Aoba05. 980-8579

Hiroyuki Kamaya
Dept. of Elec. Eng.
Hachinohe National College of Technology
Hachinohe. 039-1192

Abstract

In this paper, reinforcement learning approach to motion control of 2-link planer underactuated manipulator is described. This manipulator has one passive joint and is difficult to control. The experiments of learning to control this manipulator by RL and human are executed. Using the experimental results, the associations between RL and human learning are considered.

1 introduction

Reinforcement learning (RL) is a general framework for describing learning problems in which an autonomous agent learns strategies for interacting with its environment[1]. RL has been applied to many research areas. Motion learning is one of such areas. For the robot, in order to adapt to a dynamic environment, motion learning is one of key issues. Therefore, many algorithms for motion learning have been intensively discussed for years[2]. Many control objects are tested in *acrobot*, inverted pendulum, walking robot etc., and most of these tasks are nonlinear. They have some equilibrium points which can be stabilized by continuous feedback control. In these cases, by using the information of these equilibrium points as prior knowledge, more distinguished motion learning algorithms can be designed, e.g. a hierarchical RL algorithm composed of linear controllers and an adequate reward function[2]. However, there are several typical nonlinear systems which are not able to apply such hierarchical algorithms. 2-link planer under-actuated manipulator (2PUAM) is one of those systems. This system has been widely studied by control engineers[3]. But only a few researches have been done from view point of learning. On design of RL algorithm for 2PUAM, setting of the reward is difficult, and in addition, the state space is multidimensional and continuous. Therefore, the approximation of value function is needed to solve the local optimal problem.

On the other hand, some researchers have applied the cognitive and learning capability of human to complex control systems. Even though, a human operator fails to control complex system in the beginning, but after enough training, he, she can find a way to control it satisfactory. It is apparent that he, she does not use a mathematical model. This fact shows that human can find a satisfactory control law by a trial-and-error without the knowledge of the mathematical model to control object. For example, it is reported that the joint angle control of 2PUAM can be achieved by human operator[4]. This indicates the capability of learning the behavior which can not be achieved by using continuous feedback.

In order to solve such difficult control object, realization of RL algorithm reproducing human abilities is desirable. Therefore, the investigation of human learning mechanism from the perspective of RL is basically necessary. In this paper, a 2PUAM is selected as the learning problem. And learning experiments by RL and human are tested. In the RL experiment, Q-learning is implemented. For the experiment of human manual control, a 2PUAM simulator is developed. It includes a policy evaluation module. This module automatically approximates the Q-value function according to the action series of human. By using these experimental results, the comparison between RL agent and human learning is described.

2 Model of manipulator

In this paper, a 2PUAM is used for the learning task. It has only one active joint and one passive joint, and neither gravity nor friction torque acts on it. It is one of the simplest forms of underactuated manipulators. The equation of motion of the manipulator is:

$$\mathbf{M}_{11}(\theta)\ddot{\theta_1} + \mathbf{M}_{12}(\theta)\ddot{\theta_2} + \mathbf{c}_1(\theta, \dot{\theta}) = \tilde{\tau}, \tag{1}$$

$$\mathbf{M}_{21}(\theta)\ddot{\theta_1} + \mathbf{M}_{22}(\theta)\ddot{\theta_2} + \mathbf{c}_2(\theta, \dot{\theta}) = 0, \tag{2}$$

where θ_1 and θ_2 are angle of each joint, $\mathbf{M}$ is inertia matrix, $\mathbf{c}$ denotes centrifugal term, and right sides of boss Eq. (1), (2) are the input torque. Therefore Eq. (2) means the dynamic constraint caused by the zero torque at the passive joint. This manipulator has two main characteristics. The first is that the inertia matrix includes the passive joint angle θ_2 as usual[3], then, Eq. (2) is nonintegrable. It is called second-order nonhoronomic constraint. The second principal characteristic is that this manipulator is not bound by gravity or friction, i.e., arbitrary angle become equilibrium points. However, it is not controllable by a continuous feedback. To stabilize an equilibrium point, this manipulator must be controlled by discontinuous or time variant feedback control.

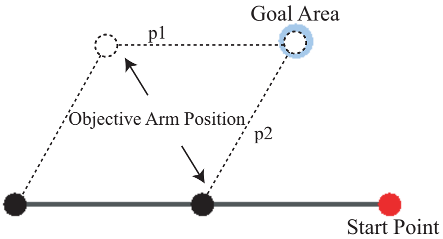

Figure 1: Environment of Learning.

2.1 Environment of Learning

In learning the task of 2PUAM called "manipulator task", the actuator installed in active joint is regarded as an agent. As shown in Fig. 1, by selecting an input torque, $\tau \in (-0.1, 0, 0.1)$ the RL agent or the subjects must drive the end effector to the goal area and bring to halt or keep between $\pm 1 (degree/s)$ angular velocity. Because of the 2PUAM's mechanism, it would be two objective positions, the upper position p_1 and the lower p_2.

3 Reinforcement Learning

At each time step $t (\in 0, 1, 2, \ldots)$, the agent observes its environmental state, $s_t \in S$ and selects an action, $a_t \in A(s_t)$. As a consequence of the action, the agent receives a scalar reinforcement signal, referred reward, $r_t \in R$. One time step later, the agent observes a new state, s_{t+1}.The aim of the agent is to maximize the expected discounted reward $E\{\sum_{t=0}^{\infty} \gamma^t r_t\}$, where γ is the discount factor. In this paper, Sarsa(λ)[5] is employed to learn estimates of optimal Q-value functions that map state-action pairs (s, a) to optimal return on the action taken in the current state.

3.1 Function Approximation

In motion learning such as manipulator, continuous state variables are dealt with. Thus, tile coding is employed[1] here. In tile coding, the receptive fields of the features are grouped into exhaustive partitions of input space. Each partition is called a tiling, and each element of the partition is called a tile. When the agent observes its environmental state s, and selects an action a, the Q-value function is calculated as

$$\mathrm{Q}(s, a) = \sum_{i,j} q_i(j, a) \phi_i^s(j) \tag{3}$$

where $i (i = 1, 2, \ldots, m)$ is the number of tiling, $j (j = 1, 2, \ldots, n)$ is the number of tile. q is the parameter vector of each tiling and ϕ is binary feature vector. If the state is inside the tile of each tiling, the corresponding feature has the value 1, otherwise the feature is 0.

3.2 Sarsa(λ)

In Sarsa(λ), on experiencing transition $< s, a, r, s', a' >$, the following updates are performed in order:

$$\eta_i(j, \bar{a}) = \begin{cases} \phi_i^s(j) & \text{for } \bar{a} = a \\ 0 & \text{for } \bar{a} \neq a \end{cases} \tag{4}$$

$$\delta = r + \gamma \mathrm{Q}(s', a') - \mathrm{Q}(s, a) \tag{5}$$

for all i and j

$$q_i(j, \bar{a}) \leftarrow q_i(j, \bar{a}) + \alpha \delta \eta_i(j, \bar{a}), \tag{6}$$

$$\eta_i(j, \bar{a}) \leftarrow \gamma \lambda \eta_i(j, \bar{a}), \tag{7}$$

where $\alpha (0 < \alpha \leq 1)$ is the learning rate, $\gamma (0 \leq \gamma \leq 1)$ is the discount factor, η is the replacing eligibility trace function, and $\lambda (0 \leq \lambda \leq 1)$ eligibility factor.

During learning, at time step t, the agent will select an action according to some strategies. In the experiments of this paper, Max-Boltzmann distribution[6] rule is employd. In Max-Boltzmann distribution, an action with maximal Q value is chosen with probability p_{max}, and an action according to the Boltzmann distribution is chosen with probability $(1 - p_{max})$. The probability of selecting action a_i in state s is

$$\mathrm{Prob}(a_i \mid s) = \frac{e^{\frac{\mathrm{Q}(s, a_i)}{T}}}{\sum_k e^{\frac{\mathrm{Q}(s, a_k)}{T}}} \tag{8}$$

where temperature T adjusts the degree of randomness of action selection.

3.3 Learning Parameters

In manipulator task, the observable parameters of the agent are the angles and angular velocities of the joints, θ_1, θ_2, $\dot{\theta_1}$, and $\dot{\theta_2}$. Thus, the state space in this task is a bounded rectangular region in four dimensions. In this environment, the state space is divided into $21 \times 21 \times 11 \times 11$ tiles, and 10 tilings are used. The remaining parameter of tile coding and Sarsa(λ) algorithm are $\alpha = 0.1/m$, $\lambda = 0.9$, $\tau = 0.1$, and $Q_0 = 0$. The parameter of Max-Boltzmann, p_{max} is linearly increased from 0.9 to 1.0. The constant physical parameters are; the mass of the arms, $m_1 = m_2 = 1.0$; the length, $l_1 = l_2 = 0.2$; the length from joint to the center of each arm, $r_1 = r_2 = 0.1$. The time step $t = 0.03$. The action is chosen after every one time step. A trial ends whether 10000 steps are elapsed, or the goal is reached.

4 Manual Control Experiment

Experiment of manual control is conducted with the cooperation of 5 subjects. They have no knowledge about the dynamic response. they observe the manipulator's states from visual data, and input torque $\tau \in (-0.1, 0, 0.1)$ given by a joystick. The time limit is set at 2000 steps, and the number of trial is set at 20 trials a day. The subjects do the same task for a week. In this experiment, the Q-value is recorded according to Sarsa(λ). The other details of this experiment are roughly same as RL's.

5 Experimental Results

5.1 RL Experiment

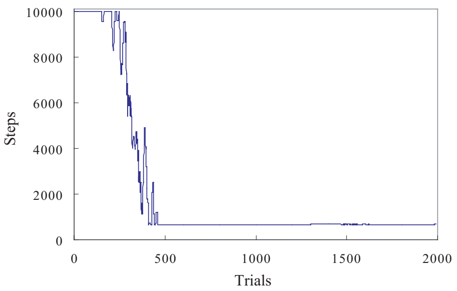

Figure 2: Result of learning by Tile Coding Sarsa(λ).

The result of RL experiment is shown in Figs. 2 and 3. In Fig. 2, the number of steps indicates how long the agent takes to achieve the goal. It is clear from Fig. 2 that the steps required to achieve the goal is reduced and converged. This result shows that it is possible to learn manipurator task by using RL algorithm. Fig. 3 shows the best trajectory acquired by RL agent. First, the agent inputs the torque crock wise (CW) in order to make angle of free joint θ_2 take the value at which the arm forms in lower objective position p_2. Then the agent manages to keep the value θ_2, and drives the end effector to the goal area with low speed.

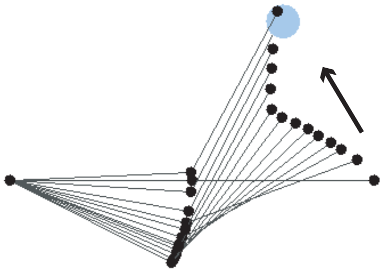

Figure 3: Motion of 2PUAM.

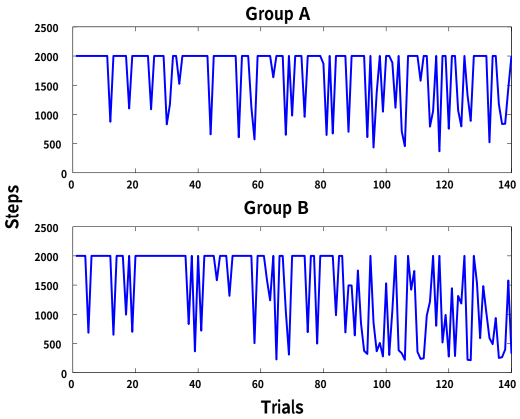

Figure 4: Result of learning by Subjects

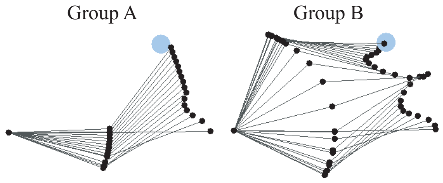

Figure 5: Motions of 2PUAM controled by Subjects.

5.2 Manual Control Experiment

From the result of Manual Control Experiment , the learning pattern of subjects is divided into Group A and Group B. Fig. 4 shows the typical learning curve

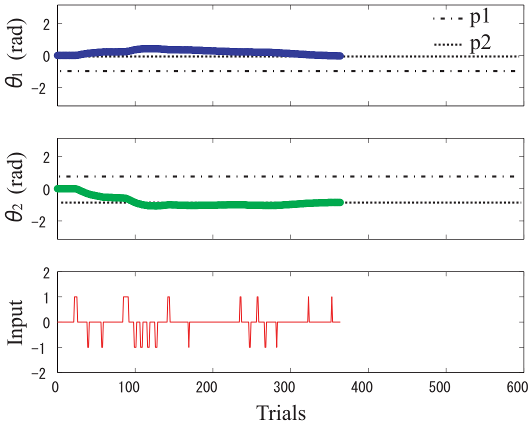

Figure 6: Input and Trajectory(Group A).

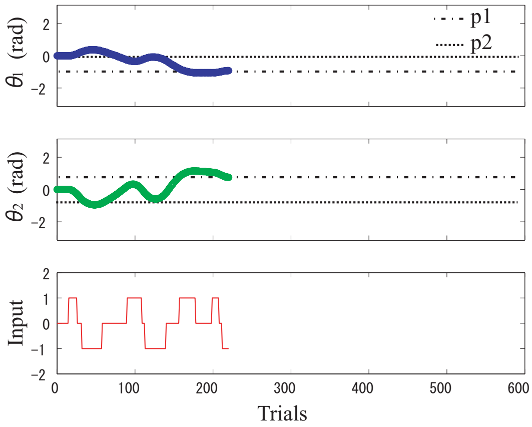

Figure 7: Input and Trajectory(Group B).

of each group. As shown in Fig. 4, each group first achieve the goal within a day. However, the failures and required steps of Group B are notably reduced from 60 to 80 trials.

Fig. 5 shows the best trajectories, and Fig. 6, Fig. 7 show the time series of each joint angle and input torque. It is clear from Fig. 5 that Group A ’s is very similar to RL ’s. That of Group B is also same as Group A at the beginning. However, at the end, Group B changed their policy and achieve the goal much faster than Group A.

5.3 Discussions

As the result of these experiments, some noticeable points of human learning in 2PUAM environment are found. The first is that, after once the subjects find a trajectory to the goal, such as the left side of Fig. 5, all subjects try to trace the same trajectory and achieve the goal faster by increasing input. It partly makes them possible to shorten the time passing through the trajectory. However, because of 2PUAM's mechanical property, it is impossible to slow down the angular velocity of second joint near the end of this trajectory. Therefore it causes the failure of the first approach. In this case, Group B searches another way to the goal in a way of changing the objective position, e.g., from p_2 to p_1. Therefore, the failure of Group B temporarily increases during 20 to 40 trials. In an alternating succession of such approach, they are enable to find the better trajectory like the right side of Fig. 5. In this trajectory, it is possible to decelerate each joint speed near the goal. Hence, it is thought that Group B come to achieve the goal much faster.

6 Summary

In this paper, RL approach for motion control of 2PUAM was proposed. And the associations between RL and human learning were investigated. As the result, some noticeable characteristics about human learning process were found. In particular, it was realized that the structure of human learning in 2PUAM has two processes. One is finding the trajectory to the goal, and another is shortening the time of passing through the found trajectory.

As future works, we aim to apply these characteristics into the machine learning process.

References

[1] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, (1998).

[2] J. Yoshimoto, S. Ishii, M. Sato, Application of reinforcement learning based on on-line EM algorithm to balancing of acrobot, Systems and Computers in Japan, 32Vol. 5, 12-20, (2001).

[3] G. Oriolo and Y. Nakamura, Free-Joint Manipulators: Motion Control under Second-Order Nonholonomic Constraints, Proc. of IROS'91, 1248-1253, (1991).

[4] M. Yagai, T. Ishihara, H. Inooka, Manual Control of the Positioning of Two-link Arm with a Free Joint, SICE Tohoku chapter workshops, 206-1, (2002)(in Japanese).

[5] G. A. Rummery and M. Niranjan, On-line Q-learning using Connectionist Systems, Technical Report CUED/F-INFENG/TR 166, Engineering Department, Cambridge University, (1994).

[6] M. Wiering and J. Schmidhuber, HQ-learning, Adaptive Behavior, 6-2, 219-246, (1997).