Role of properties of real world in evolution and learning
- Verification using two-link manipulator that has viscosity and elasticity -

Y. Yoshioka and K. Ito

Department of Systems and Control Engineering, Hosei University, 3-7-2 Kajino-cho,Koganei,Tokyo, Japan
(Tel : 81-42-387-6093)
(ito@hosei.ac.jp, yasutaka.yoshioka.6v@gs-eng.hosei.ac.jp)

Abstract: In this paper, we focus on the curse of dimensionality, and to solve the problem we utilize properties of the real world. We consider an artificial agent that has suitable mechanism that can reduce state-action space by utilizing the properties of the real world, and we show that the agent can acquire the suitable mechanism autonomously in evolution. We employ a two-link manipulator as the body of the agent. Task of the agent is to move own links to the desired positions by Q- learning. We employ simple genetic algorithm as method of the evolution and viscosity and elasticity as the properties of the real world. We set the fitness of the genetic algorithm as ease of learning, and evolve the mechanical body and parameters of the Q-learning. Simulation has been conducted and the body that utilizes the viscosity and the elasticity has been acquired, and as a result, state-action space has been extremely reduced.

Keywords: reinforcement learning, Q-learning, genetic algorithm, property of real world

I. Introduction

Recently, autonomous robots which operate in unpredictable complex environment, for example in rescue operation, in space development and so on have attracted much attention, and new autonomous controller for the robots is required. Reinforcement learning is one of possible candidate of the autonomous controller, because it can adapt robot to the unknown environment without supervisor [1]-[3].

However, conventional reinforcement learning algorithms have a significant problem in practical use. The problem is the curse of dimensionality. In practical use, real world is very complex, so state-action space becomes huge. As a result, learning can not be completed in real time. Moreover, trivial changes of environment cause failure, so robot has to re-learn every time the environment changes. Thus, it is impossible to apply conventional typical reinforcement learning for practical use.

To solve the problems, various approaches to improve learning algorithm has been proposed. However, this problem has not been solved completely.

On the other hand, animals and human beings can learn by trial-and-error in real-time and we can apply acquired policy for other similar situations without re-learning. The reason why the animals can adapt themselves so quickly is not solved completely, but recently, it is considered that dynamics of the real world plays an important role [3]-[6], and in our previous works, we showed that state-action space of reinforcement learning can be reduced by using properties of the real world [7].

In this paper, we focus on the viscosity and the elasticity as properties of the real world and consider their role in evolution and learning by conducting simulations of two-link manipulator.

II. Task and manipulator

We consider a two-link manipulator that moves on horizontal plane. Fig. 1 shows the model of the manipulator. Aim of the task is to control both joints to desired angles.

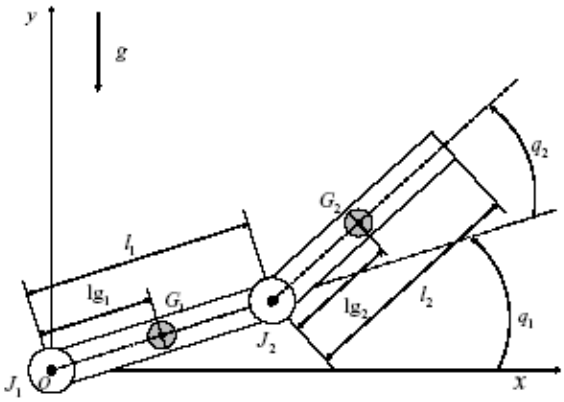

Fig. 1. Two-link manipulator

III. Properties of the real world

In this paper, we employ viscosity and elasticity as properties of the real world. Model of the manipulator in which these properties are considered is written in equation (1), (2) and Table 1.

$$\begin{aligned}&(I_1 + M_1 l_{g1}^2 + I_2 + M_2(l_1^2 + l_{g2}^2 + 2l_1 l_{g2}\cos q_2))\ddot{q}_1 \\ &+ (I_2 + M_2(l_{g2}^2 + l_1 l_{g2}\cos q_2))\ddot{q}_2 - M_2 l_1 l_{g2}\sin q_2(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2) \\ &+ k_1 q_1 + c_1\dot{q}_1 = \tau_1\end{aligned} \quad (1)$$

$$\begin{aligned}&(I_1 + M_2(l_{g2}^2 + l_1 l_{g2}\cos q_2))\ddot{q}_1 \\ &+ (I_2 + M_2 l_{g2}^2)\ddot{q}_2 + M_2 l_1 l_{g2}\sin q_2\dot{q}_1^2 \\ &+ k_2 q_2 + c_2\dot{q}_2 = \tau_2\end{aligned} \quad (2)$$

Table 1. Parameters of two-link manipulator

Mass	M_1, M_2
Length of link	l_1, l_2
Angle of joint	q_1, q_2
Moment of inertia link	I_1, I_2
Torque of joint	τ_1, τ_2
Spring constant	k_1, k_2
Damping coefficient	c_1, c_2

IV. Genetic Algorithm

We employ simple Genetic Algorithm (GA) for simulating evolution. Fig. 2 shows flowchart of the GA. In this paper we employ one point crossover and roulette selection.

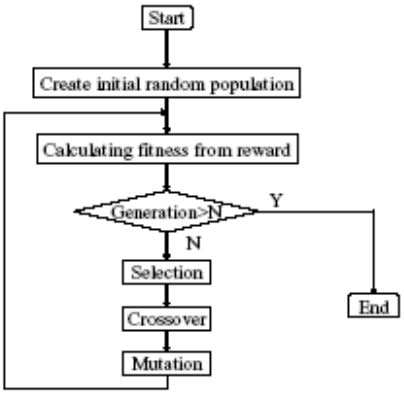

Fig. 2. Flowchart of GA

V. Q-learning

In this paper, we use Q-Learning as algorithm of the reinforcement learning. The flow is written below.

(a) The agent observes state s.
(b) The agent selects an action and conducts it.
(c) The agent receives reward r from environment.
(d) The agent observes transition state s'.
(e) The agent updates Q-value by the following equation.

$$Q(s,a) \leftarrow (1-\alpha)Q(s,a) + \alpha\{r(s,a) + \gamma \max_{a' \in a} Q(s',a')\} \quad (3)$$

(f) The agent repeats this cycle.

VI. Role of properties of the real world in evolution and learning

We consider a role of properties of the real world in evolution and learning. We set the fitness of the genetic algorithm as ease of learning, and evolve the mechanical body and parameters of the Q-learning.

VII. Simulation

1. The setting of the two-link manipulators and the Q-Learning

Table 2 shows setting of two-link manipulators and the Q-Learning. Table 3 shows parameters of the reward of Q-learning. The state and the action are decided by GA.

Table 2. Parameters of Q-learning and two-link manipulator

M_1, M_2	3.0 kg
l_1, l_2	0.1 m
Sampling time	0.001 s
Number of trials	20000
Update interval time of Q-value	0.001 s
Initial value of q_1	$\frac{\pi}{6}$
Initial value of q_2	$\frac{\pi}{6}$
α	0.5
γ	0.9999
Epsilon greedy	0.01

Table 3. Setting of reward

reward	state
100	$(-0.1 \le q_1 < 0.1) \wedge$
	$(-0.1 \le q_2 < 0.1) \wedge$
	$(-0.1 \le \dot{q}_1 < 0.1) \wedge$
	$(-0.1 \le \dot{q}_2 < 0.1)$
-100	$q_1 < -0.9$
-100	$q_2 < -0.9$
-100	$q_1 \ge 0.7$
-100	$q_2 \ge 0.7$
-100	$\dot{q}_1 < -0.9$
-100	$\dot{q}_2 < -0.9$
-100	$\dot{q}_1 \ge 0.7$
-100	$\dot{q}_2 \ge 0.7$

2. Setting of GA

Fig. 3 shows setting of gene. Length of gene is 7. First gene is a flag that indicates whether q_1 and q_2 are employed as state or not. In the same way, second gene is a flag that indicates whether $\dot{q}_1$ and $\dot{q}_2$ are employed as state or not. Third gene is a flag that indicates whether τ_1 and τ_2 are employed as action. From fourth to seventh genes indicate value of each parameter.

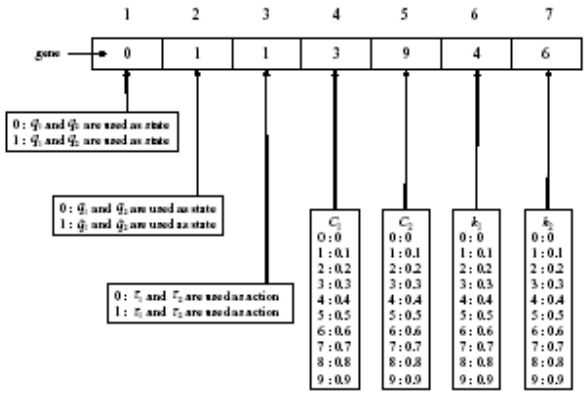

Fig. 3. Setting of gene

If q_1, q_2 or $\dot{q}_1$, $\dot{q}_2$ are used as state, state space is composed based on Table 4. In the same way, if τ_1 and τ_2 is used as action, action space is composed based on Table 5. Table 6 shows the other parameter setting of GA, and equation (4) shows the fitness function.

Table 4. State space

q_1, q_2[rad]	$\dot{q}_1$, $\dot{q}_2$[rad/s]
q<-0.9	$\dot{q}$<-0.9
-0.9≤q<-0.7	-0.9≤$\dot{q}$<-0.7
-0.7≤q<-0.5	-0.7≤$\dot{q}$<-0.5
-0.5≤q<-0.3	-0.5≤$\dot{q}$<-0.3
-0.3≤q<-0.1	-0.3≤$\dot{q}$<-0.1
-0.1≤q<0.1	-0.1≤$\dot{q}$<0.1
0.1≤q<0.3	0.1≤$\dot{q}$<0.3
0.3≤q<0.5	0.3≤$\dot{q}$<0.5
0.5≤q<0.7	0.5≤$\dot{q}$<0.7
0.7≤q	0.7≤$\dot{q}$

Table 5. Action space

τ_1, τ_2 [Nm]
-0.5
-0.4
-0.3
-0.2
-0.1
0.0
0.1
0.2
0.3
0.4

Table 6. Parameters setting of GA

Number of generations	100
Number of individuals	100
Probability of crossover	0.3
Probability of mutation	0.1

$$fitness = \sum_{n=1}^{N} r_n \quad N\text{: Number of trial of Q-learning} \quad (4)$$

3. Result of simulation

Fig. 4 shows average of fitness in each generation. Fig. 5 shows the best individual in 100th generation, and Table 7 shows its phenotype.

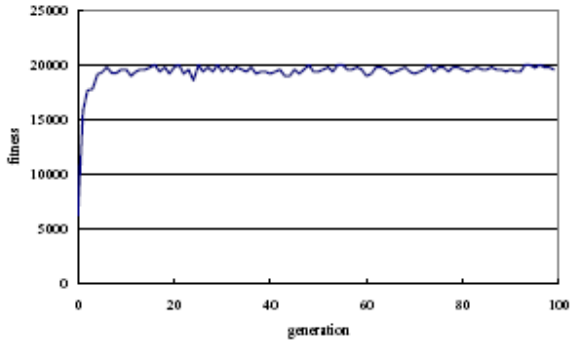

Fig. 4. Average of fitness

0	0	0	7	8	4	5

Fig. 5. The individual that is obtained in the 100th generation

Table 7. The phenotype type of Fig. 5

q_1, q_2	q_1 and q_2 are not used as state
$\dot{q}_1$, $\dot{q}_2$	$\dot{q}_1$ and $\dot{q}_2$ are not used as state
τ_1, τ_2	τ_1 and τ_2 are not used as action
c_1	0.7
c_2	0.8
k_1	0.4
k_2	0.5

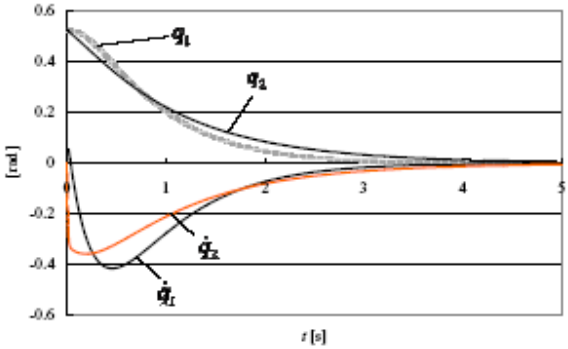

Fig. 6. Response

From Table 7, we can find that state-action space is not required, and it means that learning is not required. Instead of learning, the manipulator has the viscosity and the elasticity, and by using these properties of the real world, the manipulator is controlled as shown in Fig. 6.

We can consider that the role of the properties of the real world is to reduce load of learning, and the role of the evolution is to design the manipulator to utilize the properties of the real world and the learning can be a fitness of the evolution.

4. Comparison with simple Q-learning

To discuss roles of the evolution and properties of the real world, we conduct simulation without evolution. In this simulation, the manipulator has no viscosity and elasticity. Torque of each joint is adjusted using simple Q-learning. Table 8 and 9 shows setting of the Q-learning. Setting of the state space is written in Table 4.

Table 8. Setting of the action space

Action	τ_1	τ_2
0	0	0
1	0	0.3
2	0	-0.3
3	0.3	0
4	0.3	0.3
5	0.3	-0.3
6	-0.3	0
7	-0.3	0.3
8	-0.3	-0.3

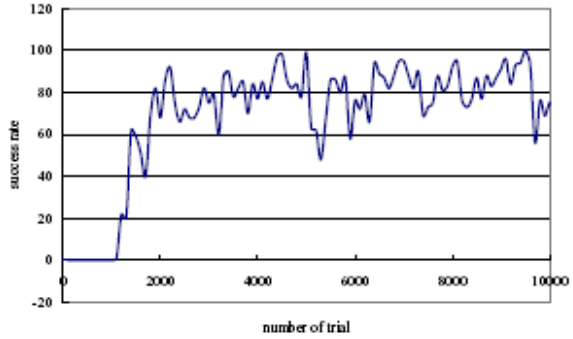

Fig. 7. Success rate

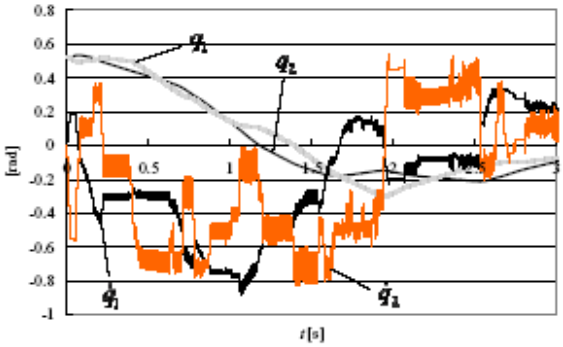

Fig. 8. Result after 20000 trials

Fig. 7 and Fig. 8 show simulation results. From these results, we can find that learning is completed successfully, and the manipulator can be controlled. However, huge number of trials is required to obtain the suitable policy. By comparing these results with the results of subsection VII-3, we can confirm that by the evolution load of the learning is reduced by utilizing the properties of the real world.

VIII. Conclusion

In this paper, we have considered a role of properties of the real world in reinforcement learning. We have employed a two-link manipulator, and applied Q-learning to control of the manipulator. Simulations have been conducted and as a result, state-action space has been reduced completely and the manipulator can be controlled by utilizing the viscosity and the elasticity without learning.

We can conclude that state-action space can be reduced by utilizing the properties of the real world and as a result, learning time is extremely reduced without improvement of the learning algorithm. We can consider that cause of the curse of dimensionality is not due to defects of the conventional learning algorithm but due to neglecting properties of the real world.

REFERENCES

[1] Sutton S R (1988), Reinforcement Learning: An Introduction. The MIT Press

[2] Kaelbling P L (1996) Reinforcement learning: A survey. In Journal of Artificial Intelligence Research 4, pp. 237-285

[3] Ito K (2005) Reinforcement Learning for Redundant Robot -Solution of state explosion problem in real world-, Proc. of ROBIO'05 Workshop on Biomimetic Robotics and Biomimetic Control, pp. 36-41

[4] Arimoto S (2004) What are the fundamentals of bio-mimetic control. In Proc. of IEEE Int. Conf. on Robotics and Biomimetics, CD-ROM

[5] Takegaki M (1981) A new feedback method for dynamical control of manipulators. ASME, J. DSMC, 103(2):119-125

[6] Pfeifer R (1999) Understanding intelligence. MIT Press, Cambridge, Massachusetts

[7] Ito K (2006) Autonomous control of a snake-like robot utilizing passive mechanism, Proceedings of the 2006 IEEE International Conference on Robotics and Automation