Reinforcement Learning for a New Piano Mover's Problem

Yuko ISHIWAKA
Hakodate National College of Technology,
Hakodate, Hokkaido, Japan

Tomohiro YOSHIDA
Muroran Institute of Technology,
Muroran, Hokkaido, Japan

and

Yukinori KAKAZU
Research Group of Complex Systems Engineering, Hokkaido University,
Sapporo, Hokkaido, Japan

ABSTRACT

We attempt to achieve corporative behavior of autonomous decentralized agents constructed via Q-Learning, which is a type of reinforcement learning. As such, in the present paper, we examine the piano mover's problem. We propose a multi-agent architecture that has a training agent, learning agents and intermediate agent. Learning agents are heterogeneous and can communicate with each other. The movement of an object with three kinds of agent depends on the composition of the actions of the learning agents. By learning its own shape through the learning agents, avoidance of obstacles by the object is expected. We simulate the proposed method in a two-dimensional continuous world. Results obtained in the present investigation reveal the effectiveness of the proposed method.

Keywords piano mover's problem, reinforcement learning, hierarchy agent system

1. Introduction

Original Piano mover's problem was introduced by Schwartz and Sharris [1] for a mathematical model. The Piano Mover's Problem is treated as a problem which controls the attitude of an object by iterating rotations and parallel translations in order to move the object outside of a room. The problem is easy to imagine how difficult to solve it because usually people have been faced with similar problem when you try to move a sofa in a small room or move a grand piano into a room and so on. Schwartz and Sharris proved to calculate the optimal path mathematically for the case in which the geometries of the space that the object passes is set. However, in the real world, it is mostly impossible to calculate exactly their environment and the objects, for example the sofa is now 132.5cm far from the table and if it is moved to 30cm to write, it is 15cm far from the chair and so on. The environment is dynamics.

There are many approaches for motion planning problem to solve this problem in real world. Configuration space method and potential field method are major approaches. The Configuration space method (C-space) treats a robot as a point without physical size in the space, and any obstacles and free spaces known. Lozano-Perez [2] generates the c-space obstacles using Minkowski sum of the robot and environment. Dorst,L. and etc.[3] used a rasterized c-space approach to plan for a two-link arm. Lengyel.J and etc. [4] rasiterized configuration space obstacles into a series of bitmap slices, and dynamic programming to create a create a navigation function. Khatib.O and Maitre [5] proposed potential field methods first. The obstacles were represented as zero level surfaces of scalar valued analytic functions. However this method has one major problem which is spurious local minima, especially for concave robots. Yokoi et al. [6] solved this problem using the vibrating potential method. In their study, the shape of the object was a rectangle, called the AGV (autonomous guided vehicle).

We propose a new piano mover's problem. The new piano mover's problem includes finding path problem in the complex space and in addition the attitude must be controlled autonomously. Generally these two problems, i.e. finding path problem and controlling attitude problem, contradict each other to solve them simultaneously. In order to find a path, the global learning is needed but to control the attitude, the local learning is necessary for collision avoidance.

We view this contradict problem as a geometric ambiguous. What the geometric ambiguous means includes not matched evaluated signals spatially. To solve this ambiguity problem we propose herein a hierarchy agent system which has three kinds of agents, a training agent, learning agents and an intermediate agent. Training agent has no perception about its own shape then it cannot be applied to mathematical method. Learning agents is also unknown about the shape, but grasp it vacantly by learning obstacle avoidance. Intermediate agent try to match the evaluated signals between a training agent and learning agent. Each agent makes its own decision for taking action, so usually the

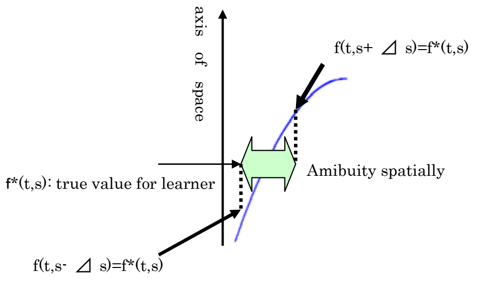


Fig.1 Definition of ambiguity spatially

decisions of the training agent and learning agents are different from each other. In other words there is high possibilities for training agent to show the wrong path which learning agents cannot pass because of the size. Intermediate agent is necessary for solving this kind of problem.

We propose herein a heterogeneous multi-agent system in which homogeneous multi-agents treated as learning agents, one training agent and one intermediate agent are constructed. Each learning agent takes simple actions, and the training agent is it treated as an object. Q-Learning [7] is performed for each learning agent, which consequently learns obstacle avoidance. The TD-method [8] is used for the training agent, which learns the path from start to finish. The simple rule is applied to intermediate agent. The proposed system is a form of hierarchic reinforcement learning.

2. The definition of ambiguity

In this paper a spatially ambiguity is shown (geometry). **Fig.1** shows the geometric ambiguity. In **Fig.1** f*(t,s) means the true value for learners. Δt and Δs mean the finite differences of time and state respectively. The ambiguity is defined as the differences, i.e. $t \pm \Delta t$ or $s \pm \Delta s$. **Fig. 2** shows an example the geometric ambiguity. Training agent orders to pass the narrow path to learner and training agent dose not grasp the learner's shape exactly. Learning agents try to avoid the obstacles but the direction of taken action is not same as shown by training agent. It causes dead lock. In this kind of unmatched evaluated signals case, intermediate agent decided the new action by observing both training agent and learning agents evaluated signals. The behaviors of intermediate signals are important.

3. Proposed System

Three kinds of agent are proposed as follows. The training agent indicates the entire object of outline, learning agents are constructed inside of the training agent and the intermediate agent matches between evaluated signals from a training agent and taken action by learning agents. **Fig.3** shows the outline of our system.

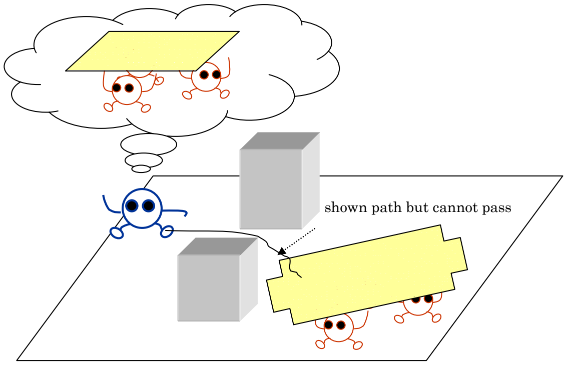


Fig.2 The ambiguity restricted spatially (geometry)

The advantages of this method are 1) correspondence to the variant environment, 2) although each agent requires simple architecture, complicated movement is possible, 3) faults of agents are addressed, and 4) application to other problems (not only piano mover's or motion control problems) according to separate heterogeneous agents is possible. For example, learning agents learn obstacle avoidance, but they learn also suspension, rolling and coming and going. These agents simply attempt to avoid walls and do not learn how to find the path. If the rewards are set finely, then the path can be learned. However, in this case the reward setting is focused and should be reset as the problem (even the environment) is changed. In order to avoid this reward-setting problem, we propose the training agent, which learns the path, and shows the direction to the learning agents. In this method the reward can be set simply. In the following section, we describe each agent in detail.

3.1 The architecture of an object

Let the shape of the object be rectangular, and let the length of long boundary and short boundary be 2a and 2b respectively (a>b). The magnitude of a vector indicating an action taken is f, and the mass is M.

Let the initial coordinates of the center-of-gravity be (X_0,Y_0), where t=0. The movement of an object can be

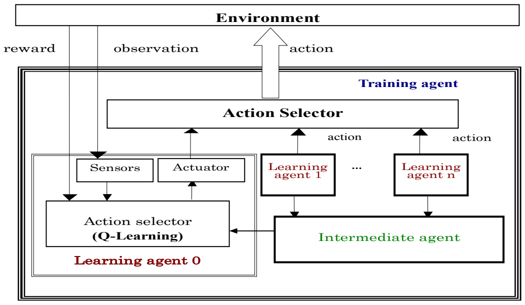


Fig.3 Relation ships of three kinds of agent for piano mover's problem.

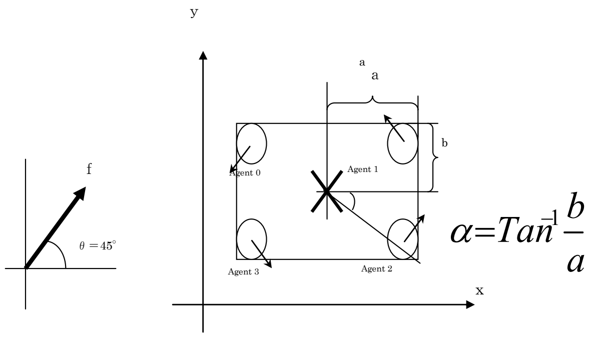


Fig. 4 An example of physical calculation.

resolved into a parallel translation component and a rotation component. The component vectors of force are obtained respectively as follows:

For *x* component,

$$F_x = \left(f\cos\left(-\frac{3}{4}\pi\right) + f\cos\left(\frac{3}{4}\pi\right) + f\cos\left(\frac{\pi}{4}\right) + f\cos\left(-\frac{\pi}{4}\right) \right)$$
$$= 0,$$

For *y* component,

$$F_y = \left(f\sin\left(-\frac{3}{4}\pi\right) + f\sin\left(\frac{3}{4}\pi\right) + f\sin\left(\frac{\pi}{4}\right) + f\sin\left(-\frac{\pi}{4}\right) \right)$$
$$= 0,$$

Then, we obtain the coordinates after movement in time t as

$$X = X_0 + 0t = X_0$$
$$Y = Y_0 + 0t = Y_0.$$

In this case, the object has no parallel translation component because the center-of-gravity of the object has not moved.

The moment-of-force M_P for the emphasis of agent 2 is calculated as follows:

$$M\_P = \begin{vmatrix} i & j & k \\ a & -b & 0 \\ t\cos\frac{\pi}{4} & t\sin\frac{\pi}{4} & 0 \end{vmatrix} = \left(at\sin\frac{\pi}{4} + bt\cos\frac{\pi}{4} \right)k.$$

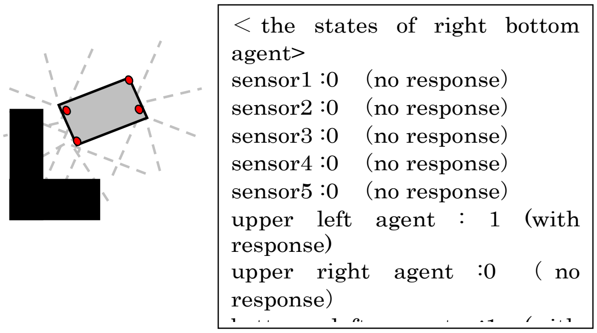
The moment-of-inertia I around the center of the object is

$$I = \frac{1}{3}M\left(a^2 + b^2\right).$$

The equation of motion for rotation around the center-of-gravity is

$$M\_P = I\frac{d^2\theta}{dt^2}.$$

The equation of motion about θ can then be solved as follows:



< the states of right bottom agent>
sensor1 :0 (no response)
sensor2 :0 (no response)
sensor3 :0 (no response)
sensor4 :0 (no response)
sensor5 :0 (no response)
upper left agent : 1 (with response)
upper right agent :0 (no response)
bottom left agent :1 (with

Fig.5 An example of state sets for an agent

$$\theta = \frac{\left(at\sin\frac{\pi}{4} + bt\cos\frac{\pi}{4} \right)}{2\left(\frac{1}{3}M\left(a^2 + b^2\right) \right)}t.$$

The coordinates of Agent 2 at time t (x_t, y_t) are then

$$x_t = X + \sqrt{a^2 + b^2}\cos(\theta - \alpha)$$
$$y_t = Y + \sqrt{a^2 + b^2}\sin(\theta - \alpha)$$
$$, where\, \alpha = \tan^{-1}\frac{b}{a}.$$

Since the object is a rigid body, its movement can be calculated using the relative movement between the center-of-gravity and a point not on the body. We therefore omit the other agents' calculation.

3.2 Learning agents

Q-learning, which is reinforcement learning, is constructed in each learning agent. The state of each learning agent is given by sensors information of five directions {s0,s1,⋯,s4}, and one compressed information is given by communications. The range of each sensor is divided into six stages to obstacles, and each learning agent can take actions in four directions, as shown in **Fig.4**. **Fig. 5** shows an example of state sets for an agent. Each agent can obtain the sensor information (five directions) using both their own sensors and the sensors of other agents (0 or 1) through communication. Therefore, learning agents learn obstacle avoidance by cooperating with each other.

Updating Q-values in learning agents is defined in the following equation:

$$Q_i(s_i(t), o_i(t))$$
$$\leftarrow (1-\alpha)Q_i(s_i(t), o_i(t)) + \alpha\left[r + \gamma \max_{o_i' \in O_i} Q_i(s_i(t+1), o_i(t+1)) \right]$$

,where s(t) is the state at time t, o(t) is the taken action ate time t, α is the step-size parameter, γ is the discount-rate

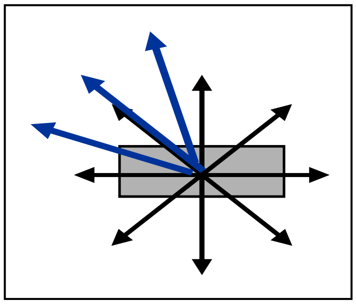


Fig.6 Internal agent select the direction among the blue arrows.

parameter and r is the reward from the environment. If r is obtained as a penalty, then both learning and training agents can learn obstacle avoidance. The function max selects the maximum Q-value action from the finite action set O_i. When learning agents take actions, if the traning agent collides with an obstacle or does not move at all, then the function max becomes,

$$\max_{o_i' \in O_i} Q_i(s_i(t+1), o_i(t+1)) = Q_i(s_i(t), o_i(t)) \cdot$$

3.3 Training Agent

The shape of a training agent is assumed as a rectangle, and in the rectangle of each corner four learning agents are put in position respectively. A training agent is only one. The training agent learns the path from a given start point to goal, and shows the direction to the goal location to the learning agents. The training agent does not have its own sensors, so the movement depends on the learning agents.

We used TD-learning for the training agent. The state of a training agent is the x-y coordinate. The training agent can select an action from a combination of the action directions of the four learning agents, which gives a total of 256 possible actions. Updating V-values is performed using the following equation:

$$V(s_t) \leftarrow V(s_t) + \alpha\left[r_t + \gamma V(s_{t+1}) - V(s_t)\right],$$

We employed TD-learning because Q-leaning depends on the dimensionality of the course, and is impossible to implement due to the state sets, i.e. the actual coordinates of the environment, and the action sets from physical calculation. Here, we use reinforcement learning for both learning agents and the training agent. However, for the training agent, several types of learning systems can be applied. If it is easy to give the path roughly, the path can be given directly to the training agent. Therefore, the autonomous robot can follow the path while avoiding obstacles intelligently. In this case, even the learning system for an training agent may not be necessary.

3.4 Internal agent

Internal agent matches between evaluated signals from training agent and action taken by learning agents. **Fig. 6** shows the strategy of internal agent. A strategy of an internal agent is employed here. It is to select the action for which the Q-value is maximum among the combinations that show the direction decided by the training agent. The indicated direction is also described in **Fig.6**.

4. Experiments

Fig. 7 shows the trajectories for different environments of size 800 x 800. Learning agents and training agent are learning simultaneously. **Fig. 8** uses similar environment but the goal position is different. In **Fig. 8** learning agents must switch back near goal. The trajectories are shown respectively. Lower small figure shows the expansion of appearance of switch back. **Fig.9** shows the results of different environment.

The common parameters for all simulations are as follows: for learning agents; sensor range = 30, sensibility = 6 levels, and the movement length is 4 per step, for Q-learning; step-size parameter α_i= 0.1, discount rate γ_i = 0.9, initial Q-values = 2.0, and penalty r_i = − 1/(max step number), for the training agent; size = (20 x 40), and mass M = 1, for TD learning; step-size parameter α_e= 0.1, discount rate γ_e = 0.9, initial V-value = 2.0, and reward r_e =1 when the agents reach the goal. The max step number is 5000 and the episode is 1000. The policies of both reinforcement learning are greedy algorithms.

6. Conclusion

We propose a new piano mover's problem which includes both path finding problem and controlling attitude problem. Generally these two problems are incongruous with each other. One of them is needed global learning and the other is needed local learning. We treated the problem as a problem not matched evaluated signals spatially (geometric ambiguous). In order to solve this new piano mover's problem, we suggest hierarchy agent system which has three kinds of agent, i.e., training agent, learning agents and intermediate agent. Training agent has no perception about its own shape then it cannot be applied to mathematical method. Learning agents is also unknown about the shape, but grasp it vacantly by learning obstacle avoidance. In this problem there is high possibilities for training agent to show the wrong path which learning agents cannot pass because of the size. Intermediate agent is necessary for solving this kind of problem.

From the various experiments it is shown the robustness of agent system architecture, it is compared between the

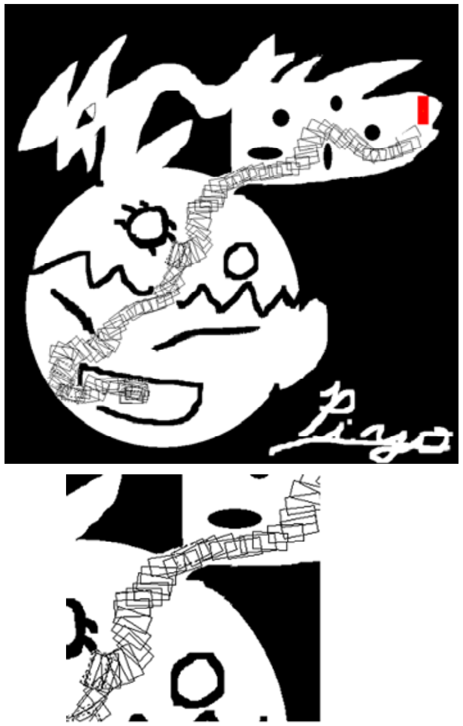


Fig.7 The trajectories of a training agent. Start point is middle of the mouse and goal position is shown in red squared right upper side. Environment size is 800 x 800. Lower figure shows zoomed of the complex behavior, i.e. switch back in the above environment.

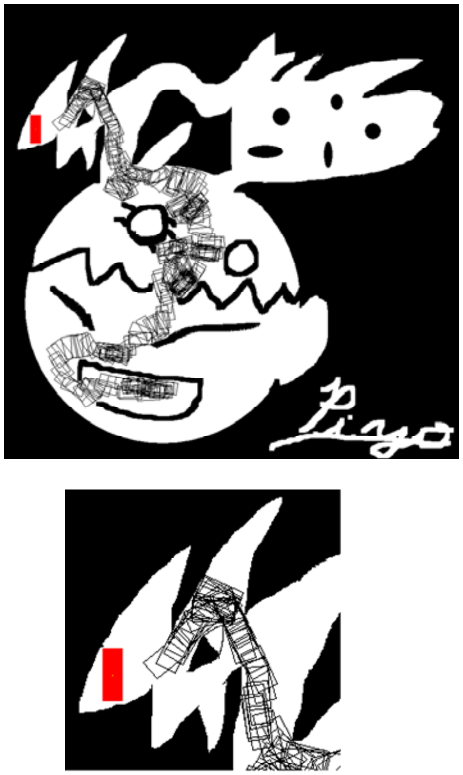


Fig.8 An example of the trajectories of a training. Start point is middle of the mouse and goal position is shown in red squared left upper side in the puteranodon. Environment size is 800 x 800. Lower figure shows zoomed of the complex behavior, i.e. switch back in the above environment.

case of changing only training agent, and the case of changing only internal agent respectively. As a result the problem can be solved even though the system has changed. According these experiments the easiness of changing system and robustness are shown.

Simultaneous learning by two types of agent, i.e. learning agents and a training agent, is particularly effective. However, the problem of our algorithms is also appeared. The learning of the training agent is dominant over that of the learning agents, so intermediate agent applies the efficiency of the learning of the learning agents is diminished. In order to avoid this problem, we must investigate adaptive internal agent. In the future, the experimental environment will be extended to three dimensions.

7. References

[1] Jacob T. Schwartz and Micha Sharir : “On the Piano Movers’ Problem: Ⅱ.General Techniques for Computing Topological Properties of Real Algebraic Manifolds, Planning, Geometry, and Complexity of Robot Motion”, pp. 51－96, 1983

[2] Lozano-Prez,T. and M.A.Wesly, "An Alogorithm for planning Collision-Free Paths Among Polyhedral Obstacles", Communications of the ACM, 22, pp560-570,1979

[3] Dorst,L. and K.Troato.,"Optimal path Planning with Six Degrees of Freedom", Artificial Intelligence ,31 , pp295-353,1987.

[4] Jed Lengyel and Mark Reichert and Bruce R. Donald and Donald P. Greenberg:"Real-Time Robot Motion Planning Using Rasterizing Computer Graphics Hardware",journal of Computer Graphics, vol.24, No.4, pp327-335,1990

[5] Khatib,O. and J.Le Maitre,"Dynamic Control of Manipulators Operating in a Complex Environment", Proceedings Third International CISM-IFToMM Symposium,September 1978, pp.267-282.

[6] Yokoi H., Mizuno T., Takita M and Kakazu Y.: ”Obstacle Avoidance Using Vibrating Potential Method (Self-Organization in a Narrow Path”, Journal of Robotics and Mechatronics, Vol.8 No.4, 1996

[7] Watkins, C.J.H. and Dayan, P.: Technical note: Q-learning, Machine Learning, Vol.8,pp.55-68,1992

[8] Sutton R. and A.G. Barto: “Reinforcement Learning”, MIT Press, Cambridge, MA,1998

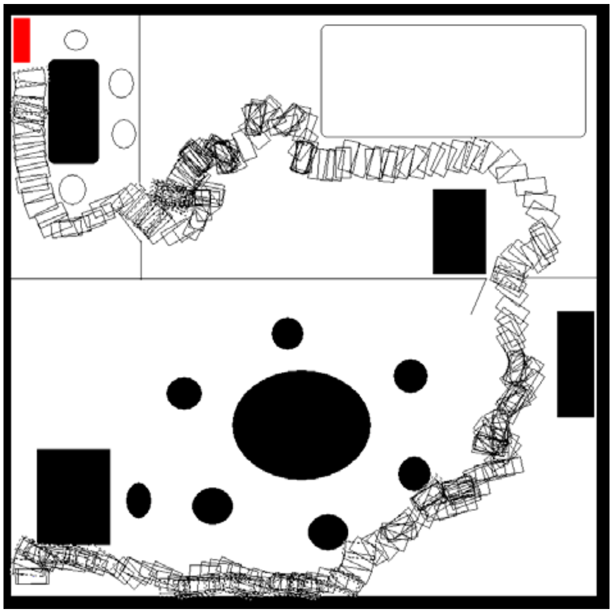


Fig.9 An example of the trajectories of an training agent using Exp.3a after learning in a room. The size of environment is 800 x 800. Start point is the left bottom and goal is shown in red squired left upper side. There is only one exit door per room from start to goal.