A Strategy Learning Model for Robot Brain

Masahiro Ono, Mamoru Sasaki and Atsushi Iwata

Graduate school of Advanced Sciences of Matter, Hiroshima University,
1-3-1, Kagamiyama, Higashi-Hiroshima-shi 739-8527, Japan
Phone and Fax: +81-824-22-7358, E-mail: {masa, sasaki, iwa}@dsl.hiroshima-u.ac.jp

1 Introduction

Recently, the robots interacting with human being such as welfare robots are expected, with the progress of the robot technology. So it is necessary for the robot to have a brain with capabilities similar to human's one (we call it "Robot Brain"). It can recognize our characteristics and choose the most suitable action according to memorized experiences obtained by learning. We propose a model for the Robot Brain which accomplishes the above process. The ability of the model is confirmed by a simulation program of "air hockey game" as an example of the tasks. The game needs interactions with a person and the person tends to well show his characteristics in the game, because he must recognize a puck and act so quickly. Finally, the effectiveness of the proposed model is quantitatively evaluated in the simulation.

2 A Strategy Learning Model

The proposed model is shown in Fig.1. It consists of the three sections: 1. selection of a strategy, 2. construction of a new strategy by learning, 3. addition of it or elimination of some strategies. At first, the opponent's behavior is observed in order to estimate his characteristics. Secondly, using the estimation result, the optimum strategy for the current opponent is selected from many candidates stored in the memory. This is carried out in the first section. The many strategies have been obtained by past learning. The selected strategy is copied into working memory. The robot acts based on the copied one. Thirdly, actions are decided according to the strategy in the working memory, and simultaneously the strategy is tuned in order to obtain better one. This is carried out in the second section. Finally, the tuned one is estimated with a criterion. It is added in the memory, if the estimation judges that it is needed. And, some needless strategies are eliminated to restrain memory overflow. This is carried out in the third section. In this report, we explain only the first and second sections. Here, we define a strategy. It consists of a set of "situation" and "action" pairs. By another expression, it consists of a set of if-then rules. "Situation" and "action" have generally many terms. Thus, let us define them as vector: s and a, respectively.

Next, we show the detail of the first and second sections in Fig.1. We start the explanation of the second section, to easily understand the model. In the section, Q-function, $\mathrm{Q}_x(\mathbf{s}_i,\mathbf{a}_j)$ has very important role. It expresses the quality of an action $\mathbf{a}_j$ in a situation $\mathbf{s}_i$ (larger Q_x means that an action is better). x is an index expressing each strategy. The value of Q_x is updated by Q-PSP Learning[1], which is one of Reinforcement Learning (RL) methods[2]. In the scheme of RL, the robot is given a positive reward (or a negative reward) in the case of achieving its goal (or not). In the progress of RL, the expectation of reward becomes larger. The update rule of Q-PSP Learning has two cases. One is that the current action leads to the final goal. This case is called "case-1". Another is one in the sequence which is going to the final goal. It is called "case-2". In the "case-1", the update rule is as follows. For the Q_x of all selected actions in the sequence until the final goal, $\mathrm{Q}_x(\mathbf{s}(n),\mathbf{a}(n))\leftarrow\mathrm{Q}_x(\mathbf{s}(n),\mathbf{a}(n)+(\alpha_1\cdot\mathrm{r}\cdot\gamma_1^n$-$\mathrm{Q}_x(\mathbf{s}(n),\mathbf{a}(n))$, α_1 and $\gamma_1(0\leq\alpha_1, \gamma_1\leq 1)$ are parameters called step-size parameter and discounted parameter, respectively. r is the reward, and n $(= 0, 1, 2\cdots)$ is the time index from the current to the past. On the other hand, in "case-2", $\mathrm{Q}_x(\mathbf{s},\mathbf{a}_j)\leftarrow\mathrm{Q}_x(\mathbf{s},\mathbf{a}_j)+\alpha_2(\gamma_2(\max_{\mathbf{a}_k}(\mathrm{Q}_x(\mathbf{s}^{'},\mathbf{a}_k)$ - $\mathrm{Q}_x(\mathbf{s},\mathbf{a}_j)))$, where $\mathbf{s}^{'}$ means the current situation, α_2 and $\gamma_2(0\leq\alpha_2,\gamma_2\leq 1)$ are discounted parameter and step-size parameter in "case-2". The max operation is carried out among the index k.

Next, we explain the first section in Fig.1. The Q-function can be also employed in order to evaluate the opponent's characteristics. So, we prepare $\mathrm{Q}_{obs}(\mathbf{s}_i,\mathbf{a}_j)$ which is Q-function, for evaluating the opponent's characteristics. As well as the Q-function in the second section, Q_{obs} is also updated during a sequence. The way to select a strategy is as follows. R_x, which is the difference between the updated Q_x and Q_{obs}, is calculated. The strategy that has the maximum value of R_x is selected. This means that the strategy with the closest Q_x to Q_{obs} is selected. However, if some strategies have the values being close to the maximum R_x, the proper strategy may not be selected. So, we define another measure V_x, which is the sum of reward. It can express the justice of the selected strategy. When the strategy is selected, we consider also the measure V_x as follows. If V_x is the smallest, the strategy must not be selected. In order to avoid too large value of V_x, it is reset when the value is larger than a constant.

3 Experiment

In order to confirm the ability of the model, we applied it to "air hockey game". The robot selects an action according to probability. The probability $\mathrm{p}(\mathbf{s},\mathbf{a})$ is calculated based on Q_x: $\mathrm{p}(\mathbf{s},\mathbf{a})=\exp(\mathrm{Q}_x(\mathbf{s},\mathbf{a})/\mathrm{T})/\sum_{\mathbf{a}'}\exp(\mathrm{Q}_x(\mathbf{s},\mathbf{a}^{'})/\mathrm{T})$, where T is called thermal parameter. The reward is +1 or -1 when robot gets or loses a point, respectively. In an experiment, an opponent is the same program with a simple strategy. The strategy is fixed and is not tuned by learning. The opponent's characteristic has been varied by stick position as shown in Fig.2. Fig.3 shows the result of the experiment. The stick position of the opponent changes from A to E (see Fig.2) every set. It is one set in 20 points. The number of the stored strategies is four. Before the experiment, the four strategies have been obtained by learning in cases that the stick positions of opponent are fixed at A, B, C, and D, respectively. The strategies are called S_A, S_B, S_C and S_D. Fig.3 (a) shows that the proper strategy except B was quickly selected. Here, we consider the case of the position B. It is the second set. Fig.3(b) shows that the robot overwhelmingly won even by using the S$_C$. So, the model has also selected the proper strategy in this case. Next, let us consider the case of the position E. Note that no strategy has been previously prepared for the position E. In order to confirm the selection of S$_C$ for the position E is appropriate, we had another experiment. In the experiment, we applied the strategies from S$_A$ to S$_D$ to the position E. Fig.4 shows the result: The S$_C$ is the best strategy because the robot overwhelmingly won as comparison with the others. From the above discussions, the model's effectiveness has been confirmed apparent.

4 Conclusions

To realize Robot brain, which can execute tasks interacting with human being, we have proposed a model with RL. The model can select the most suitable strategy so quickly and construct a new strategy by learning. It was confirmed by the simulation experiment. In present study, we use the virtual opponent. As the next step, we'll have the experiment using a person as the opponent.

References

[1] Horiuchi, T., Fujino, A., Katai, O. and Sawaragi, T. Q-PSP Learning: An Exploitation-Oriented Q-Learning Algorithm and Its Applications. SICE (in Japanese), 35(5), 645-653 (1999).

[2] Richard S.Sutton and Andrew G.Barto, Reinforcement Learning, MIT Press, 1998.

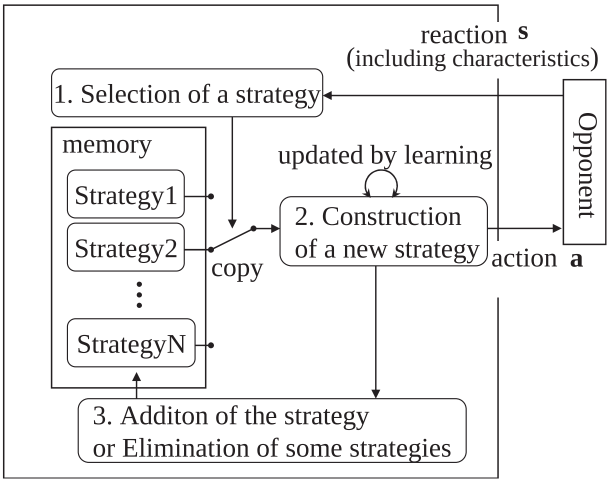


Figure 1: A proposed model

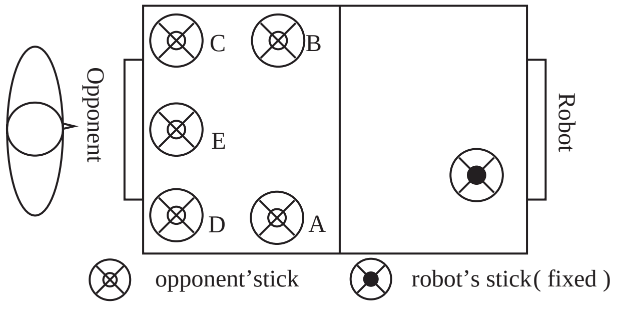
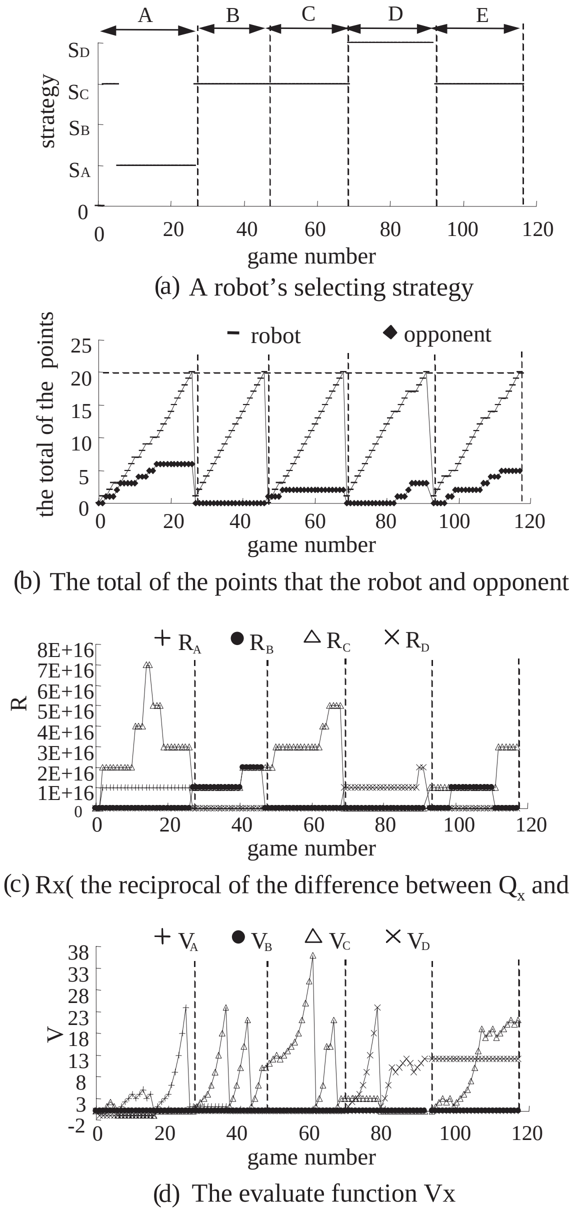


Figure 2: An example of the opponent's characteristics



(a) A robot's selecting strategy

(b) The total of the points that the robot and opponent got

(c) Rx(the reciprocal of the difference between Q$_x$ and Q$_{obs}$)

(d) The evaluate function Vx

Figure 3: The result of the experiment where the opponent's stick position changes A from E by one set (=twenty points)

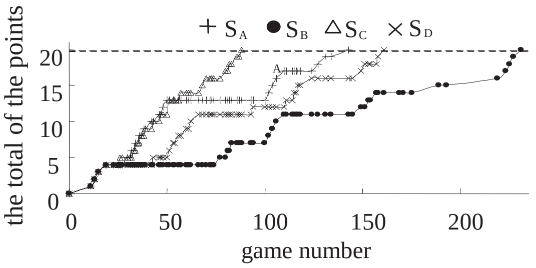


Figure 4: The total points given to the robot by using each strategy during position E