12

Modular Learning Systems for Behavior Acquisition in Multi-Agent Environment

Yasutake Takahashi and Minoru Asada
Osaka University
Japan

1. Introduction

There has been a great deal of research on reinforcement learning in multirobot/agent environments during last decades[1]. A wide range of applications, such as forage robots (Mataric, 1997), soccer playing robots (Asada et al., 1996), prey-pursuing robots (Fujii et al., 1998) and so on, have been investigated. However, a straightforward application of the simple reinforcement learning method to multi-robot dynamic systems has a lot of issues, such as uncertainty caused by others, distributed control, partial observability of internal states of others, asynchronous action taking, and so on. In this paper we mainly focus on two major difficulties in practical use :

- unstable dynamics caused by policy alternation of other agents
- curse of dimension problem

The policy alternation of others in multi-agent environments may cause sudden changes in state transition probabilities of which constancy is needed for behavior learning to converge. Asada et al. (Asada et al., 1999) proposed a method that sets a global learning schedule in which only one agent is specified as a learner with the rest of the agents having fixed policies to avoid the issue of the simultaneous learning. As a matter of course, they did not consider the alternation of the opponent's policies. Ikenoue et al. (Ikenoue et al., 2002) showed simultaneous cooperative behavior acquisition by fixing learners' policies for a certain period during the learning process. In the case of cooperative behavior acquisition, no agent has any reason to change policies while they continue to acquire positive rewards as a result of their cooperative behavior with each other. The agents update their policies gradually so that the state transition probabilities can be regarded as almost fixed from the viewpoint of the other learning agents. Kuhlmann and Stone (Kuhlmann and Stone, 2004) have applied a reinforcement learning system with a function approximator to the keep-away problem in the situation of the RoboCup simulation league. In their work, only the passer learns his policy is to keep the ball away from the opponents. The other agents (receivers and opponents) follow fixed policies given by the designer beforehand.

The amount of information to be handled in multi-agent system tends to be huge and easily causes the curse of dimension problem. Elfwing et al. (Elfwing et al., 2004) achieved the cooperative behavior learning task between two robots in real time by introducing the

[1] For example, a survey (Yang and Gu, 2004) is available.

Source: Reinforcement Learning: Theory and Applications, Book edited by Cornelius Weber, Mark Elshaw and Norbert Michael Mayer
ISBN 978-3-902613-14-1, pp.424, January 2008, I-Tech Education and Publishing, Vienna, Austria

macro action that is an abstracted action code predefined by the designer. However, only the macro actions do not seem sufficient to accelerate the learning time in a case that more agents are included in the environment. Therefore, the sensory information should be also abstracted to reduce the size of the state space. Kalyanakrishnan et al. (Kalyanakrishnan et al., 2006) showed that the learning rate can be accelerated by sharing the learned information in the 4 on 5 game task. However, they still need long learning time since they directly use the raw sensory information as state variables to determine the situation that the learning agent encounters.

Keys for cooping with the above difficulties are to divide a whole complex situation into several ones in which state transition can be regarded as stable enough, and to keep exploration space as small as possible based on abstracted task specific information instead of the row sensory information. A modular learning system might be a practical solution for those difficulties.

This chapter briefly introduces examples of application of modular learning systems for cooperative/competitive behavior acquisition in scenarios of RoboCup Middle Size League. A modular learning system is successfully applied for adaptation to the policy alternation of others by switching modules each of which corresponds to different situation caused by the policy alternation of the other. Introduction of macro actions enables reduction of exploration space and simultaneous multi-agent behavior learning. The experimental results of 2 on 3 passing task are shown. Furthermore, in order to attack the problem of curse of dimension, a state abstraction method based on state value function of a behavior learning module is proposed and applied to the 4 on 5 passing task. A player can acquire cooperative behaviors with its teammates and competitive ones against opponents within a reasonable learning time. Finally, conclusions and future work are shown.

2. Modular learning system for policy alternation of others

In this section, a modular learning system for behavior acquisition in the multiagent environment is introduced. A multi-module learning system for even single agent learning in a multi-agent environment is shown difficult when we straightforwardly apply it. A simple learning scheduling is introduced in order to make it relatively easy to assign modules automatically. Second, macro actions are introduced to realize simultaneous learning in multi-agent environments in which each agent does not need to fix its policy according to some learning schedule. More detailed description was given in (Takahashi et al., 2005).

2.1 3 on 1 game

Before describing the modular learning system in details, a task in the RoboCup middle size league context is introduced as a testbed to evaluate the learning system. The game is like a three-on-one involving one opponent and three other players. The player nearest to the ball becomes a passer who passes the ball to one of its teammates (receivers) while the opponent tries to intercept it. Fig.2 shows the viewer of our simulator for the robots and the environment and a situation the learning agents are supposed to encounter. Fig.1 shows a mobile robot we have designed and built. The robot has an omni-directional camera system. A simple color image processing is applied to detect the ball, the interceptor, and the receivers on the image in real-time (every 33ms.) The left of Fig.2 shows a situation a

learning agent can encounter while the right images show the simulated ones of the normal and omni vision systems. The mobile platform is an omni-directional vehicle (roration and translation in any direction on the plane are possible at any moment).

Fig. 1. A real Robot

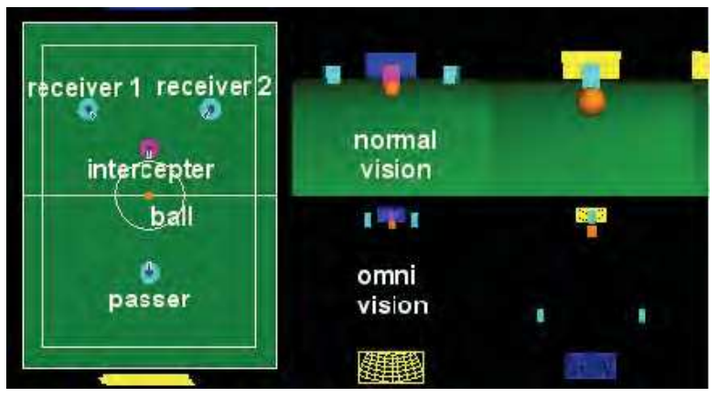

Fig. 2. A 3 on 1 game (left) and the viewer of the game simulator (right)

2.1 Modular learning system

The basic idea is that the learning agent could assign one behavior learning module to one situation which reflects another agent's behavior and the learning module would acquire a purposive behavior under the situation if the agent can distinguish a number of situations, each in which the state transition probabilities are almost constant. We introduce a modular learning approach to realize this idea (Fig.3). A module consists of both a learning component that models the world and an action planner. The whole system follows these procedures:

- select a module in which the world model is estimated best among the modules;
- update the model in the module; and
- calculate action values to accomplish a given task based on the estimated model using dynamic programming.

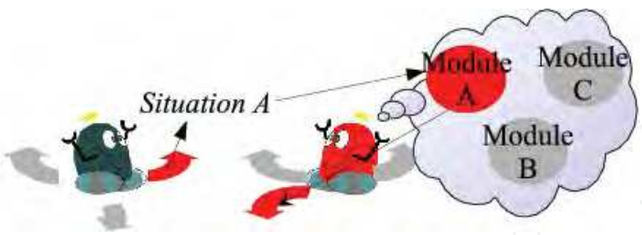

Fig. 3. Adaptive behavior selection based on Multi-module learning system

As an experimental task, we suppose ball passing with the possibility of being intercepted by the opponent (Fig.2). The problem for the passer (interceptor) here is to select one module of which model can most accurately describe the interceptor's (passer's) behavior from the viewpoint of the agent and then to take an action based on the policy which is planned with the estimated model.

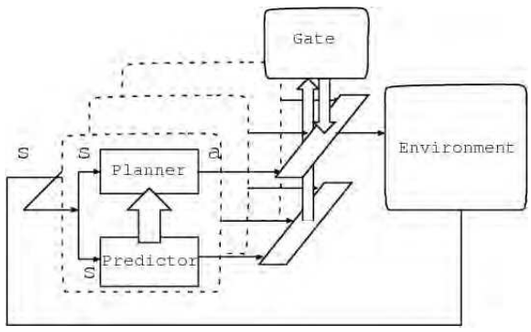

Fig. 4. A multi-module learning system

Fig. 4. shows a basic architecture of the proposed system, i.e., a modular reinforcement learning system. Each module has a forward model (predictor) which represents the state transition model and a behavior learner (action planner) which estimates the state-action value function based on the forward model in a reinforcement learning manner. This idea of a combination of a forward model and a reinforcement learning system is similar to the H-DYNA architecture (Singh, 1992) or MOSAIC (Doya et al., 2000). The system selects one module which has the best estimation of a state transition sequence by activating a gate signal corresponding to the module while deactivating the gate signals of the other modules; the selected module then sends action commands based on its policy.

2.3 Behaviors acquisition under scheduling

First, we show how it is difficult to directly introduce the proposed multi-module learning system in the multi-agent system. A simple learning scheduling is introduced in order to make it relatively easy to assign modules automatically.

The initial positions of the ball, passer, interceptor, and receivers are shown in Fig. 2. The opponent has two kinds of behaviors: it defends the left side or right side. The passer agent has to estimate which direction the interceptor will defend and go to the position so as to kick the ball in the direction the interceptor does not defend. From the viewpoint of the

multi-module learning system, the passer will estimate which situation of the module is going on and select the most appropriate module as its behavior. The passer acquires a positive reward when it approaches the ball and kicks it to one of the receivers.
A learning schedule is composed of three stages to show its validity. The opponent fixes its defending policy as a right-side block at the first stage. After 250 trials, the opponent changes the policy to block the left side at the second stage and continues this for another 250 trials. Finally, the opponent changes the defending policy randomly after one trial.

2.4 Configuration

The state space is constructed in terms of the centroid of the ball on the image, the angle between the ball and the interceptor, and the angles between the ball and the potential receivers (see Figs. 9 (a) and (b)). The action space is constructed in terms of the desired three velocity values (xd, yd, wd) to be sent to the motor controller (Fig. 6). The robot has a pinball-like kick device which allows it to automatically kick the ball whenever the ball comes within the region to be kicked. It tries to estimate the mapping from sensory information to appropriate motor commands by the proposed method.

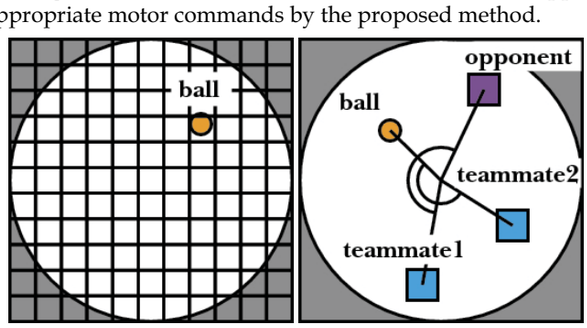

Fig. 5. State variables : Left : (a) state variables (position) Right: (b) state variables (angle)

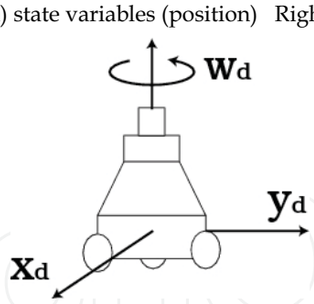

Fig. 6. Action variables

2.5 Simulation results

We have applied the method to a learning agent and compared it with only one learning module. The performances between the methods with and without the learning scheduling are compared as well. Fig.7 shows the success rates of those during the learning process. ″success″ indicates the learning agent successfully kicked the ball without interception by the opponent. The success rate shows the number of successes in the last 50 trials. The

"mono. module" in the figure means a "monolithic module" system which tries to acquire a behavior for both policies of the opponent.

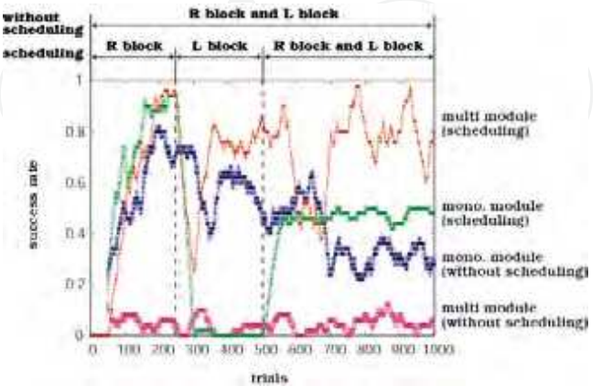

Fig. 7. Success rates during the learning

The multi-module system with scheduling shows a better performance than the one-module system. The monolithic module with scheduling means we applied the learning scheduling mentioned in 2.3 even though the system has only one learning module. The performance of this system is similar to the multi-module system until the end of the first stage between the fist and the 250th trials; however, it goes down at the second stage because the obtained policy is biased by the experiences at the first stage and cannot follow the policy change of the opponent. Because the opponent uses one of the policies at random in the third stage, the learning agent obtains about 50% of the success rate.

The term "without scheduling" means we do not apply learning scheduling and the opponent changes its policy at random from the beginning. Somehow the performance of the monolithic module system without learning scheduling gets worse after 200 trials. The multi-module system without a learning schedule shows the worst performance in our experiments. This result indicates it is very difficult to recognize the situation at the early stage of the learning process because the modules have too few experiences to evaluate their fitness; thus, the system tends to select the module without any consistency. As a result, the system cannot acquire any valid policies.

3. Simultaneous learning with macro actions

The exploration space with macro actions becomes much smaller than the one with primitive actions; therefore, the macro action increases the possibility of creating cooperative/competitive experiences and leads the two agents to find a reasonable solution in a realistic learning time frame. Here, macro actions are introduced in order to realize simultaneous learning in a multi-agent environment in which each agent does not need to fix its policy according to some learning schedule. In this experiment, the passer and the interceptor learn their behaviors simultaneously. The passer learns behaviors for different situations caused by the alternation of the interceptor's policies, i.e., blocking to the left side or the right. The interceptor also learns behaviors for different situations caused by the alternation of the passer's policies, i.e., passing a ball to a left receiver or a right one.

3.1 Macro actions and state spaces

Fig. 8 shows the macro actions of the passer and the interceptor. The macro actions by the interceptor are blocking the pass way to the left receiver and the right one. On the other hand, the macro action by the passer are turning left, turning right around the ball, and approaching the ball to kick it. A ball gazing control is embedded in both learners. The number of the actions is 2 and 3, respectively.

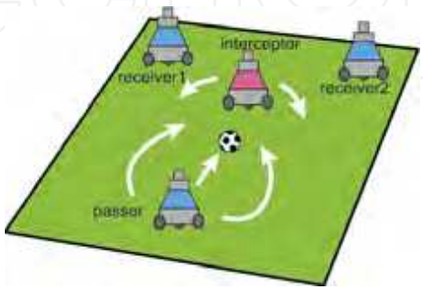

Fig. 8. Macro actions

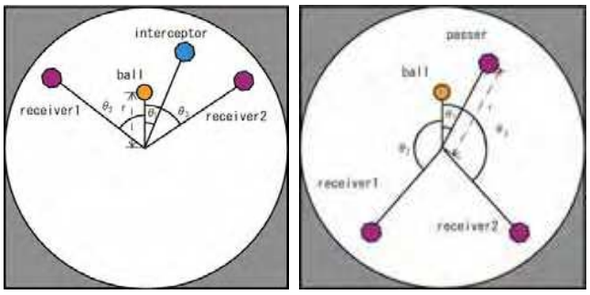

Fig. 9. State variables Left : (a) passer Right : (b) interceptor

The state space for the passer is constructed in terms of the y position of the ball on the normal image, the angle between the ball and the centers of interceptor, and the angles between the balls and the two receivers on the image of omni-directional vision. The number of the states is reduced because the set of macro actions enable us to select a smaller number of state variables and coarser quantization. The state space for the interceptor is constructed in terms of the y position of the passer on the image of normal vision system, the angle between the ball and the passer, and the angles between the ball and the two receivers on the image of omni-directional vision.

3.2 Experimental results

We have checked how the simultaneous learning of the passer and interceptor works on our computer simulation. Both agents start to learn their behaviors from scratch and have 1500 trials without any scheduling. To check whether both learners acquired appropriate behaviors against the opponent's behaviors, we fixed one agent's policy and checked to see if the other could select an appropriate behavior, then determined its success rate. Table 1 shows these results.

Passer	Interceptor	Passer's success rate [%]	Interceptor's success rate [%]	Draw rate [%]
LM0, LM1	LM0	59.0	23.0	18.0
LM0,LM1	LM1	52.7	34.3	13.0
LM0	LM0,LM1	25.6	55.0	19.4
LM1	LM0,LM1	26.0	59.3	14.7
LM0,LM1	LM0,LM1	27.6	37.3	25.1

Table 1. Success rates for a passer and an interceptor in different cases

Both players have two modules and were assigned to appropriate situations by themselves. LM and the digit number right after the LM indicate the Learning Module and the index number of the module, respectively. For example, if the passer uses both LM0 and LM1 and the interceptor uses only LM0, then the passer's success rate, interceptor's success rate, and draw rate are 59.0 %, 23.0%, and 18.0%, respectively. Apparently, the player with multi-modules switching achieves a higher success rate than the opponent using only one module. These results demonstrate the multi-module learning system works well for both.

The same architecture is applied to the real robots. Fig. 10 shows one example of behaviors by real robots. First, the interceptor tried to block the left side, then the passer approached the ball with the intention of passing it to the right receiver. The interceptor found it was trying to block the wrong side and changed to block the other (right) side, but it was too late to intercept the ball and the passer successfully passed the ball to the right receiver.

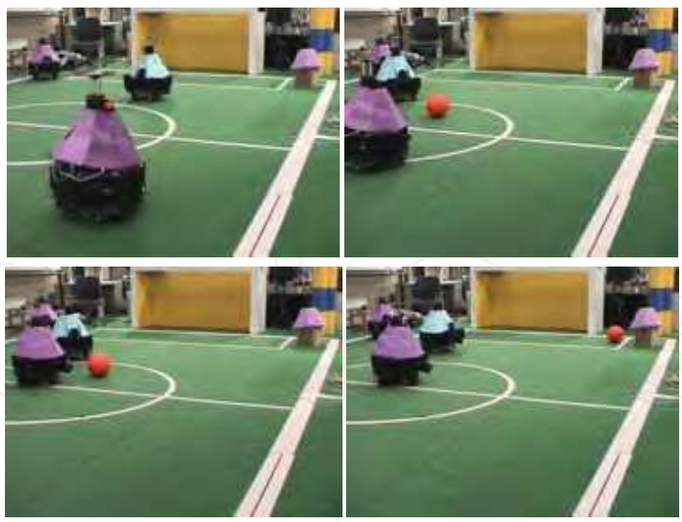

Fig. 10. A sequence of a behavior of passing a ball to the right receiver while

4. Cooperative/competitive behavior learning with other's state value estimation modules

Conventional approaches, including ones described in the previous sections, have been suffering from the curse of dimension problem when they are applied to multiagent dynamic environments. State/action spaces based on sensory information and motor commands easily become too huge for a learner to explore. In the previous section, macro actions are introduced to reduce the exploration space and enable agents to learn purposive competitive behaviors according to the situation caused by the opponent. As the next step, state space should be constructed as small as possible to enable cooperative/competitive behaviour learning in practical time. The key ideas to resolve the issue are as follows. First, a two-layer hierarchical system with multi learning modules is adopted to reduce the size of the sensor and action spaces. The state space of the top layer consists of the state values from the lower level, and the macro actions are used to reduce the size of the physical action space. Second, the state of the other to what extent it is close to its own goal is estimated by observation and used as a state value in the top layer state space to realize the cooperative/competitive behaviors. The method is applied to 4 (defense team) on 5 (offense team) game task, and the learning agent successfully acquired the teamwork plays (pass and shoot) within much shorter learning time. Here, the method is briefly introduced. More detailed description was given in (Noma et al., 2007).

Fig.11 shows a basic architecture of the proposed system, i.e., a two-layered multi-module reinforcement learning system. The bottom layer consists of two kinds of modules: behavior modules and other's state value estimation ones. The top layer consists of a single gate module that learns which behavior module should be selected according to the current state that consists of state values sent from the modules at the bottom layer. The gate module acquires a purposive policy to select an appropriate behavior module based on reinforcement learning.

The role of the other's state value estimation module is to estimate the state value that indicates the degree of achievement of the other's task through observation, and to send this value to the state space of the gate module at the top layer. In order to estimate the degree of achievement, the following procedure is taken.

1. The learner acquires the various kinds of behaviors that the other agent may take, and each behavior corresponds to each behavior module that estimates state value of the behavior.
2. The learner estimates the sensory information observed by the other through the 3-D reconstruction of its own sensory information.
3. Based on the estimated sensory information of the other, each other's state value estimation module estimates the other's state value by assigning the state value of the corresponding behavior module of its own.

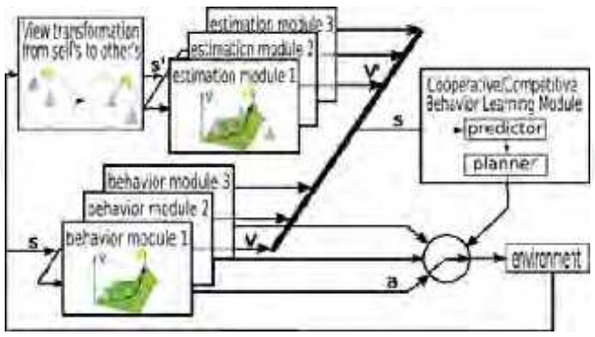

Fig. 11. A multi-module learning system

4.1 5 on 4 game

The game consists of the offense team (five players and one of them can be the passer) and the defense team (four players attempt to intercept the ball). The offense player nearest to the ball becomes a passer who passes the ball to one of its teammates (receivers) or shoot the ball to the goal if possible while the opposing team tries to intercept it (see Fig. 12).

Fig. 12. A passer and the defense formation

Only the passer learns its behavior while the receivers and the defense team members take the fixed control policies. The receiver becomes the passer after receiving the ball and the passer becomes the receiver after passing the ball. After one episode, the learned information is circulated among team members through communication channel but no communication during one episode. The behavior and the state value estimation modules are given a priori. The offense (defense) team color is magenta (cyan), and the goal color is blue (yellow) in the following figures.

The passer who is the nearest to the ball passes the ball to one of four receivers or dribble-shoots the ball to the goal. After its passing, the passer shows a pass-and-go behavior that is a motion to the goal during the fixed period of time. The receivers face to the ball and move to the positions so that they can form a rectangle by taking the distance to the nearest

teammates (the passer or other receivers) (see Fig. 12). The initial positions of the team members are randomly arranged inside their territory.
The defense team member who is nearest to the passer attempts to intercept the ball, and each of other members attempts to "block" the nearest receiver. "Block" means to move to the position near the offense team member and between the offense and its own goal (see Fig. 12). The offense team member attempts to catch the ball if it is approaching. In order to avoid the disadvantage of the offense team, the defense team members are not allowed inside the penalty area during the fixed period of time. The initial positions of the team members are randomly arranged inside their territory but outside the center circle.

4.2 Structure of the state and action spaces

The passer is only one learner, and the state and action spaces for the lower modules and the gate one are constructed as follows. The action modules are four passing ones for four individual receivers, and one dribble-shoot module. The other's state value estimation modules are the ones to estimate the degree of achievement of ball receiving for four individual receivers, that is how easily the receiver can receive the ball from the passer. These modules are given in advance before the learning of the gate module.
The action spaces of the lower modules adopt the macro actions that the designer specifies in advance to reduce the size of the exploration space without searching at the physical motor level. The state space **S** for the gate module consists of the following state values from the lower modules:

- four state values of passing behavior modules corresponding to four receivers,
- one state value of dribble-shoot behavior module, and
- four state values of receiver's state value estimation modules corresponding to four receivers.

In order to reduce the size of the whole state space, these values are binarized, therefore its size is $2^4 \times 2 \times 2^4 = 512$.
The rewards are given as follows:

- 10 when the ball is shot into the goal (one episode is over),
- -1 when the ball is intercepted (one episode is over),
- when the ball is successfully passed,
- when the ball is dribbled.

When the ball is out of the field or the pre-specified time period elapsed, the game is called "draw" and one episode is over.

4.3 Experimental results

The success rate is shown in Fig. 13(a) where the action selection is 80% greedy and 20% random to cope with new situations. Around the 900th trial, the learning seems to have converged at 30% success, 70% failure, and 10% draw. Compared to the results of (Kalyanakrishnan et al., 2006) that has around 30% success rate with 30,000 trials, the learning time is drastically improved (30 times quicker). Fig. 13(b) indicates the number of passes where it decreases after the 350 trials that means the number of useless passes decreased.
In cases of the success, failure, and draw rates when 100% greedy and 100% random are 55%, 35%, 10%, and 2%, 97%, 1%, respectively. The reason why the success rate in case of

100% greedy is better than in case of 80% greedy seems that the control policies of the receivers and the defense players are fixed, therefore not so new situations happened.
An example of acquired behavior is shown in Fig. 14 where a sequence of twelve top views indicates a successful pass and shoot scene.

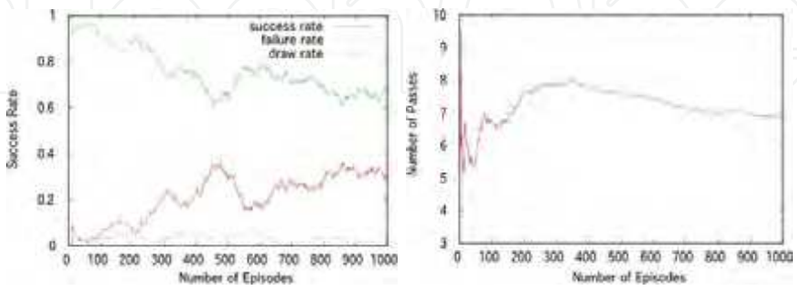

Fig. 13. (a) Success rates and (b) the number of passes

Fig. 14. An example of the acquired behavior in 5 on 4 game

Although we have not used the communication between agents during one episode, the receiver's state value estimation modules seem to take the similar role. Then, we performed the learning without these modules. Fig. 15 shows the success rate, and we can see that the converged success rate is around 21% that is close to 23% of the success rate of the result of the existing method (Kalyanakrishnan et al., 2006).

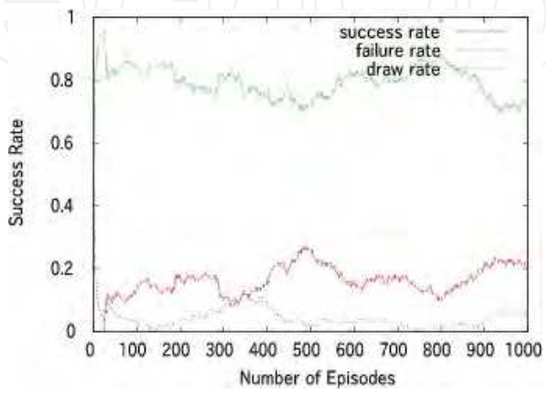

Fig. 15. Success rate without the receiver's state value estimation modules

5. Conclusion

In this chapter, we have showed a method by which multiple modules are assigned to different situations caused by the alternation of the other agent's policy so that an agent may learn purposive behaviors for the specified situations as consequences of the other agent's behaviors.

Macro actions are introduced to realize simultaneous learning of competitive behaviors in a multi-agent system. Results of a soccer situation and the importance of the learning scheduling in case of none-simultaneous learning without macro actions, as well as the validity of the macro actions in case of simultaneous learning in the multi-agent system, were shown.

We have also showed another learning system using the state values instead of the physical sensor values and macro actions instead of the physical motor commands, and adopted the receiver's state value estimation modules that estimate how easy for each receiver to receive the ball in order to accelerate the learning. The state and action space abstraction (the use of state values and macro actions) contributes to the reduction of the learning time while the use of the receiver's state value estimation modules contributed to the improvement of the teamwork performance.

6. References

Asada, M., Noda, S., Tawaratumida, S., and Hosoda, K. (1996). Purposive behavior acquisition for a real robot by vision-based reinforcement learning. *Machine Learning*, 23:279–303.

Asada, M., Uchibe, E., and Hosoda, K. (1999). Cooperative behavior acquisition for mobile robots in dynamically changing real worlds via vision-based reinforcement learning and development. *Artificial Intelligence*, 110:275–292.

Doya, K., Samejima, K., ichi Katagiri, K., and Kawato, M. (2000). Multiple model-based reinforcement learning. Technical report, Kawato Dynamic Brain Project Technical Report, KDB-TR-08, Japan Science and Technology Corporatio.

Elfwing, S., Uchibe, E., Doya, K., and Christensen1, H. I. (2004). Multi-agent reinforcement learning: Using macro actions to learn a mating task. In *Proceedings of 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages CD–ROM.

Fujii, T., Arai, Y., Asama, H., and Endo, I. (1998). Multilayered reinforcement learning for complicated collision avoidance problems. In *Proceedings of the 1998 IEEE International Conference on Robotics and Automation*, pages 2186–2198.

Ikenoue, S., Asada, M., and Hosoda, K. (2002). Cooperative behavior acquisition by asynchronous policy renewal that enables simultaneous learning in multiagent environment. In *Proceedings of the 2002 IEEE/RSJ Intl. Conference on Intelligent Robots and Systems*, pages 2728–2734.

Kalyanakrishnan, S., Liu, Y., and Stone, P. (2006). Half field offense in robocup soccer: A multiagent reinforcement learning case study. In Lakemeyer, G., Sklar, E., Sorrenti, D., and Takahashi, T., editors, *RoboCup 2006 Symposium papers and team description papers*, pages CD–ROM.

Kuhlmann, G. and Stone, P. (2004). Progress in learning 3 vs. 2 keepaway. In Polani, D., Browning, B., Bonarini, A., and Yoshida, K., editors, *RoboCup- 2003: Robot Soccer World Cup VII*. Springer Verlag, Berlin.

Mataric, M. J. (1997). Reinforcement learning in the multi robot domain. *Autonomous Robots*, 4(1):77–83.

Noma, K., Takahashi, Y., and Asada, M. (2007). Cooperative/competitive behavior acquisition based on state value estimation of others. In Visser, U., Ribeiro, F., Ohashi, T., and Dellaert, F., editors, *Proceedings of the RoboCup2007 Symposium*, pages CD–ROM.

Singh, S. P. (1992). Reinforcement learning with a hierarchy of abstract models. In *National Conference on Artificial Intelligence*, pages 202–207.

Takahashi, Y., Edazawa, K., Noma, K., and Asada, M. (2005). Simultaneous learning to acquire competitive behaviors in multi-agent system based on a modular learning system. In *Proceedings of the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 153–159.

Yang, E. and Gu, D. (2004). Multiagent reinforcement learning for multi-robot systems: A survey. Technical report, University of Essex Technical Report CSM-404.

Reinforcement Learning

Edited by Cornelius Weber, Mark Elshaw and Norbert Michael Mayer

ISBN 978-3-902613-14-1
Hard cover, 424 pages
Publisher I-Tech Education and Publishing
Published online 01, January, 2008
Published in print edition January, 2008

Brains rule the world, and brain-like computation is increasingly used in computers and electronic devices. Brain-like computation is about processing and interpreting data or directly putting forward and performing actions. Learning is a very important aspect. This book is on reinforcement learning which involves performing actions to achieve a goal. The first 11 chapters of this book describe and extend the scope of reinforcement learning. The remaining 11 chapters show that there is already wide usage in numerous fields. Reinforcement learning can tackle control tasks that are too complex for traditional, hand-designed, non-learning controllers. As learning computers can deal with technical complexities, the tasks of human operators remain to specify goals on increasingly higher levels. This book shows that reinforcement learning is a very dynamic area in terms of theory and applications and it shall stimulate and encourage new research in this field.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

IntechOpen