Cooperative behavior acquistion by asynchronous policy renewal that enables simultaneous learning in multiagent enviroment

Shoichi Ikenoue,Minoru Asada, and Koh Hosoda

Dept. of Adaptive Machine Systems,Osaka University,Osaka,Japan,
{ikenoue,asada,hosoda}@er.ams.eng.osaka-u.ac.jp

Abstract

This paper presents a method for simultaneous learning in multiagent environment to emerge the cooperative behaviors. Each agent has one policy and one action value function: the former is for action execution based on the the action value function updated in the previous stage, and the latter is for learning based on the episodes experienced by the ϵ-greedy method. This makes all agents behave based on the fixed policies, by which the non-Markovian problem can be avoided except for the update periods that depends on the learning progress of each agent. In order to avoid the local maxima due to such asynchronous renewal of action value functions, optimistic action values are given as initial ones, that helps the exploration process not to be trapped in the local maxima. The experimental results applied to one of the cooperative task in dynamic, multiagent environment, RoboCup, is shown and a discussion is given.

1 Introduction

Multiagent cooperation is one of the issues to extend the capability of single robot not simply to increase the efficiency owing to parallel operation but also to enable the task accomplishment that cannot be achieved by a single robot such as carrying a heavy load or passing and shooting in a soccer game situation. As the environment dynamics increase much more, centralized control of multirobot seems difficult because the variety of environment changes become too wide to predict everything in advance. Therefore, learnig methods are expected to cope with this problem, and a number of researchers have considered to apply reinforcement learning [1] to multiagent domain [2][3][4][5][6][7].

A typical scheme of the reinforcement learning is that an agent acquires its policy to achieve the goal by learning the action value function based on the reward given at the current state and the taken action. During this process, the Markovian assumption is necessary that the state transition depends on only the pair of the current state and the taken action. However, to realize simultaneous reinforcement learning in a multiagent environment seems very hard because of non-Markovian process due to the change of the environment caused by the mutual learning process of agents [8].

In order to avoid this problem, Asada et al. [9] proposed a method of global scheduling by limiting the number of learning agents only one and by letting the rests execute the fixed policies acquired in the previous learning stage. This system needs a kind of centralized control of switching the learners, which requires the explicit communication lines from the central system to the individual learning agents. From a viewpoint of autonomy, less centralized control is more preferable.

This paper presents a method for simultaneous learning in multiagent environment to emerge the cooperative behaviors. Each agent has one policy and one action value function: the former is for action execution based on the the action value function updated in the previous stage, and the latter is for learning based on the episodes experienced by the ϵ-greedy method. This makes all agents behave based on the fixed policies, by which the non-Markovian problem can be avoided except for the update periods that depends on the learning progress of each agent. In order to avoid the local maxima due to such asynchronous renewal of action value functions, optimistic action values are given as initial ones, that helps the exploration process not to be trapped in the local maxima. The experimental results applied to one of the cooperative task in dynamic, multiagent environment, RoboCup [10], is shown and a discussion is given.

2 Reinforcement Learning

Reinforcement learning has recently been receiving increased attention as a method for robot learning with little or no *a priori* knowledge and higher capability of reactive and adaptive behaviors. Fig.1 shows

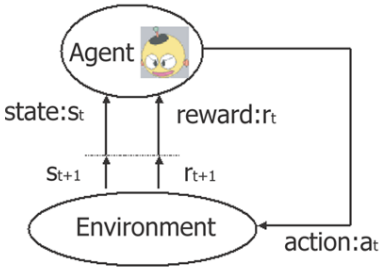


Figure 1: *The interaction with the environment in reinforcement learning*

the basic model of robot-environment interaction [1], where a robot and environment are modeled by two synchronized finite state automatons interacting in a discrete time cyclical processes. The robot senses the current state $s_t \in S$ of the environment and selects an action $a_t \in A$. Based on the state and action, the environment makes a transition to a new state and generates a reward r_t that is passed back to the robot. Through these interactions, the robot learns a purposive behavior to achieve a given goal. In order for the learning to converge correctly, the environment should satisfy the Markovian assumption that the state transition depends on only the current state and the taken action.

2.1 Q-learning

In Q-learning designed by Watkins [11], the action value function $Q(s, a)$ shows the value of taking the action $a \in A$ at the state $s \in S$, and based on the reward function $r(s, a)$ given by the designer, it is updated as follows to approximate the optimal one $Q^*(s, a)$.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - Q(s_t, a_t)], \quad (1)$$

$$V(s_t) = \max_{a \in A} Q(s_t, a). \quad (2)$$

The optimal policy π^* is given by:

$$\pi^*(s) = arg \max_{a \in A} [Q^*(s, a)]\{\forall s \in S\}, \quad (3)$$

where α and γ (between 0 and 1) denote the learning rate and the discounting factor, respectively. Both are parameters to control the learning process. If α is larger, the learning converges fast but more possibility to be trapped at the local maxima. Else, the learning becomes more conservative and takes longer time to converge. γ controls to what degree rewards in the distant future affect the total value of a policy and is just slightly less than 1. When γ is small, the learned behavior tends to be reflexive.

The finally obtained action value function Q through this process can be approximated to the optimal one Q^* independently from the exploration strategy adopted during the approximation process. This property is called "policy-off" type, by which the exploration strategy to determine the pairs of the state and the action to be visited can be arbitrary, but it is required that all pairs of them should be continuously updated.

2.2 ϵ-greedy method

To resolve the famous problem of the trade-off between exploitation and exploration to maximize the total rewards, ϵ-greedy method is often used in which random action selection is performed with the probability ϵ and the optimal action based on the current action value function Q is selected with probability (1-ϵ). Usually, the action values are initialized pessimistically, that is, all zeros (see Fig.2:left), and gradually approximated to the optimal one Q^* (see Fig.3).

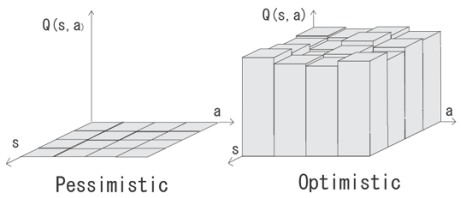


Figure 2: *Initialization of action-value function*

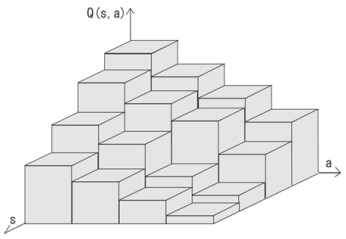


Figure 3: *Optimal action-value function*

3 The problem of simultaneuos learning

If multiple Q-learning agents adopt ϵ-greedy method, the policies often change, and therefore, the Markovian assumption does not hold any more. That is, to realize simultaneous resinforcement learning in a

multiagent environment seems very hard because of non-Markovian process due to the change of the environment caused by the mutual learning process of agents[8], except for a case where the exploration area is considerably limited.

In order to avoid this problem, it is necessary to devise an environment that is satisfies the Markovian assumption.Asada et al.[9] proposed a method of global scheduling by limiting the number of learning agents only one and by letting the rests execute the fixed policies acquired in the previous learning stage. The central system monitors the learnig process to switch the learning agents, and the cooperative behavior is acquired using ϵ-greedy method. They applied their method to one of the cooperative task in a soccer game situation, pass and shoot cooperation using two robots in the middle size league of RoboCup. Their system requires the explicit communication lines from the central system to the individual learning agents. From a viewpoint of autonomy, less centralized control is more preferable.

4 Simultaneous learning based in asynchronous policy renewal

To realize the more decentralized system while achieving the cooperative behaviors, we propose a method of asynchronous policy renewal with one policy and one action value function. In order to realize the Markovian environment, each agent utilizes the fixed policy π^n based on Q^{n-1} for action selection during the n-th learning process while collecting the episodes for learning (to update Q^n). If the learning of the individual agent converges, it update the policy π^{n+1} based on Q^n. Thus, the Markovian environment is realized by adopting the fixed policy against the other learning agents who have also the fixed policy based on the same scheme. Each agent has its own threshold to judge if its learning converges independently, therefore the update timing is asyncronous.

Actually, each agent computes the summation of optimal action values σ_Q as follows:

$$\sigma_Q = \sum_s \max_{a \in A} [Q^n(s,a)] \{\forall s \in S\}, \tag{4}$$

and then, its derivative is compared with the prespecified threshold θ_{σ_Q}. If the derivative is smaller than θ_{σ_Q}, then the learning function Q^n is judeged as converged.

As the initial action values, we set the higher values than the reward (1.0) (see Fig.2:right). This is a kind of exploration strategy for the optimal action value function (see Fig.3) by reducing the action values when the selected action is not appropriate at the current state, which is opposite from the exploration strategy starting from the zero initial values (see Fig.2:left). The former is called "optimistic" while the latter "pessimistic" [1]. Since all agents behaves under the fixed policy π^n based on Q^{n-1}, the optimistic strategy seems preferable because of its efficient exploration (this is discussed later).

Each agent execute the following algorithm:

1. prepare Q^n and π^n, and initialize them optimistically,
2. observe the state $s_t \in S$ in the environment,
3. select an action $a_t \in A$ at the current state s under the action policy π^n,

$$\pi^n(s) = arg \max_{a \in A} [Q^{n-1}(s,a)] \{\forall s \in S\} \tag{5}$$

4. then, the state transits to the next one $s_{t+1} \in S$ after exection of the action a_t, and
5. the immediate reward r_{t+1} is given from the environment,

$$Q^n(s_t,a_t) \leftarrow Q^n(s_t,a_t) + \alpha[r_{t+1} + \gamma V(s_{t+1}) - Q^n(s_t,a_t)] \tag{6}$$

$$V(s_t) = \max_a Q^n(s_{t+1},a) \tag{7}$$

Thus, by updating the action value function Q^n, the optimal one Q^* can be approxinated,

6. if Q^n converges, update the policy $\pi^{n+1}(s)$.

$$\pi^{n+1}(s) \leftarrow arg \max_{a \in A} Q^n(s,a) \tag{8}$$

7. return 2.

5 Experiments

We applied the proposed method to a soccer game situation, more correctly, a cooperative task of pass and shoot using two mobile robots in the middle size league of RoboCup [10]. The success of cooperative behavior is that both passer and shooter are able to get reward during one trial of learning. Actually, This cooperative behavior is most reasonable behavior of getting goal in the limited time Fig.4 indicates the competition filed [8m x 4m] where two learning robots (passer and shooter) moves around and a ball is located at randomly BALL AREA (one of ten positions inside the area). The sampling rate is 33 [msec] corresponding to video frame frequency, and the agent repeats the same action execution until the state changes [12].

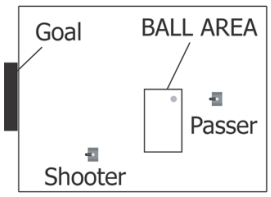


Figure 4: *Initial positions in the computer simulator*

Figure 5: *A mobile robot,a ball and goal(back)*

Fig.5 indicates the ball and non-holonomic real robot (PWS) that has the omnidirectional vision system and the kicking device. The internal structure is shown in Fig.6.

The state space is given by quantizing the perceived visual field in terms of orientation (eight diretions) and distance (four) as shown in Fig.7. The observed object (an opponent, a ball, and an opponent goals) is localized as one of tessellated trapezoids. if any object isn't observed(overlap,etc),it is counted a missing state. The front direction has finer resolution than others so that the kicking devise can work well. Totally, the dimension of the state space is six (three kinds of objects and their directions and distances) and the number of element states including a missing state is $(8 \times 4 + 1)^3 = 35937$, and the action space consists of four kinds of actions of forward, backward, right turn, and left turn. The kicking devise works everytime it can kick the ball.

The reward functions are defined as follows:

- For the passer: the reward 1 is given when it succeeded in passing the ball to the shooter, that is, when the state $s =$ { ball direction, ball distance, goal direction, goal distance, shooter direction, shooter distance } $= \{0, 0, a, b, c, d\}(a = 0 or 1, c = 0 or 7, b and d$ are arbitrary), then take a forward motion,
- For the shooter:the reward 1 is given when it succeeded in shooting the ball into the goal, that is, when the state $s =$ {ball direction, ball distance, goal direction, goal distance, shooter direction, shooter distance } $= \{0, 0, a, b, c, d\}(a = 0, b and c d$ are arbitrary), then take a forward motion,
- Any collision between the passer and the shooter gives the negative reward -1 to the passer,
- else, 0 rewards.

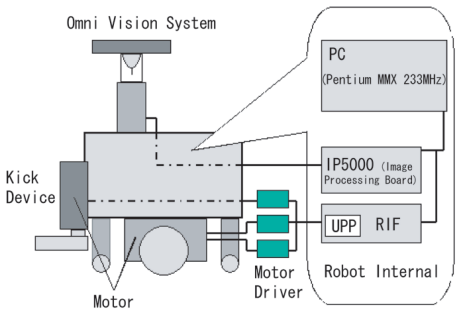


Figure 6: *An overview of the robot system*

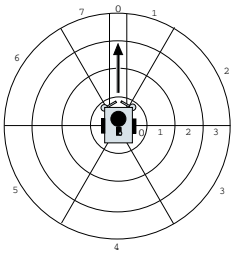


Figure 7: *A state space of the agent*

The learning pareameters $\alpha = 0.2$, $\gamma = 0.9$.

6 Experimental results

We tested the following three methods,

- The proposed method with optimistic initial values ($1.0 \sim 1.0001$)
- The global scheduling method with pessimistic initial values (0.0)
- No scheduling with pessimistic initial values (0.0)

and the changes of the success rate of the cooperative behaviors are shown in Fig.8 where the convergence

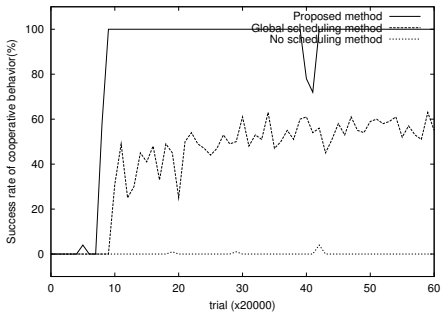


Figure 8: *Comparison of scheduling methods*

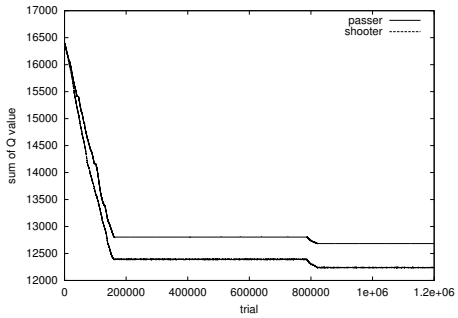


Figure 9: *Transition of σ_Q in optimistic initialization*

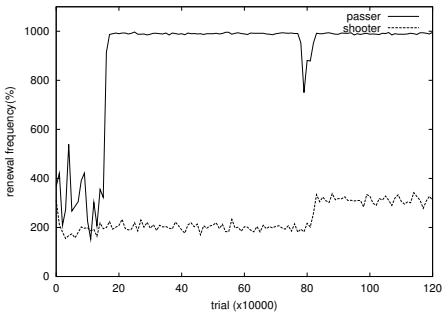


Figure 10: *Renewal frequency in optimistic initialization*

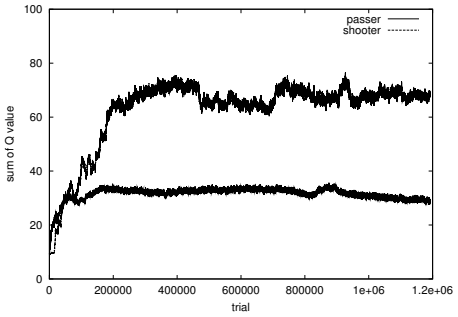


Figure 11: *Transition of σ_Q in pessimistic initialization*

threshold $\theta_{\sigma_Q} = 0.01$, and the change of the transition of σ_Q and the change of the renewal frequencies for both agents are shown in Fig.9, Fig.10. We checked that the proposed method with pessimistic initial values has not been able to learn any cooperative behaviors , and the change of the transition of σ_Q and the change of the renewal frequencies for both agents are shown in Fig.11, Fig.12.

while the global scheduling and no scheduling with optimistic initial values has the almost same performance with the pessimistic ones. Further, the chenge of the success rate of the cooperative behavior during 1000000-th trial and 1200000-th one in terms of the threshold $\theta_{\sigma_Q}$is shown Fig.13.

¿From these results, we may conclude:

1. no scheduling: regardless of the initial values, the cooperative behavior was not obtained because of non-Markovian environment caused by mutual learning,

2. global scheduling: regardless of the initial values, the cooperative behavior was obtained because the Markovian environment is held, and

3. proposed method: the success rate depends on the initial value and also the threshold, too.

The third point can be interpreted as follows. In the case of the pessimistic initial values, almost no change in σ_Q because of narrow exploration in the very early stage, therefore high frequency of renewal, however, the immediate rewards after this stage biased the exploration and therefore much more chances of being trapped into the local maxima, and consequently σ_Q tends to be unstable that corresponds to low frequency of the renewal (see Fig.11,Fig.12). On the other hand, in the case of the optimistic initial values, the early rewards may have big changes in σ_Q because of the broad exploration due to high values and as a result the optimal action is easily found and action values for inappropriate pairs of the states and actions immediately decreased, which prevents the frequent renewals of the action value functions (σ_Q tends to be unstable that corresponds to low frequency of the renewal in the early stage: see Fig.9,Fig.10). After this stage, the action value function converges and therefore high renewal frequency.

However, this causes the sensitivity to the threshold of learning convergence (see Fig.13). If the threshold is too large, it means too frequent renewal and

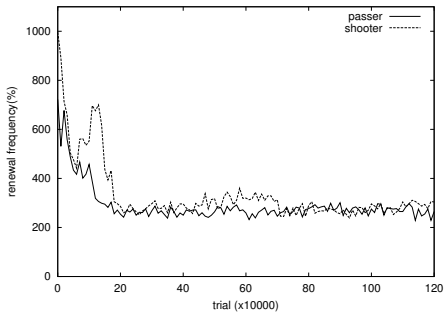


Figure 12: *Renewal frequency in pessimistic initialization*

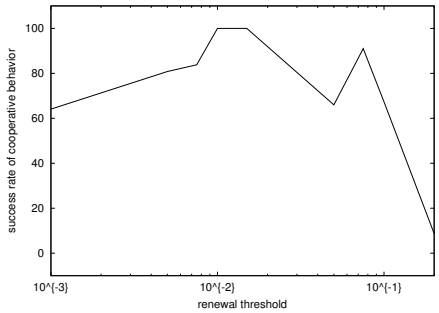


Figure 13: *Comparison of threshold value*

is almost equivalent to no scheduleing. On the other hand, if the threshold is too small, it means much less frequecy of renewal, that is, less exploration while more explotation, and it is easily trapped into the local maxima. Therefore, the selection of the threshold is important.

Fig.14 shows a sequence of cooperative behavior relaized by transferring the learned final policies into our robots. Due to the difference between the computer simulation and the real world, the success rate is not so high as the computer simulation. However, reasonable performance was obtained.

7 Conclusion

The method that enables simultaneous learning in the multiagent environment is proposed and it is applied to a typical cooperative task of pass and shoot in a soccer game situation, RoboCup. Asynchronous renewal of action value functions with optimistic initial values is a key idea of the method.

The method is sensitive to the threshold of the learning convergence decision. A systematic way to determine the threshold is one of our future work.

References

[1] Richard S.Sutton and Andrew G.Barto: "Reinforcement learning:An Introduction", MIT Press/Bradford Books, March 1998.

[2] Peter Stone and Richard S.Sutton : "Scaling Reinforcement Learning toward RoboCup Soccer", 18th International Conference on Machine Learning, 2001.

[3] P.Stone: "Layered Learning", Eleventh European Conference on Machine Learning, 2000.

[4] Yasuo Nagayuki,Shin Ishii, and Kenji Doya: "Multi-Agent Reinforcement Learning:An Approach Based on the Other Agent's Internal Model", Fourth International Conference on MultiAgent System(ICMAS) Los Alamitos:IEEE Computer Soceiety, pp.215-221,2000.

[5] T.Andou: "Refinement of Soccer Agent's Positions Using Reinforcement Learning.H.Kitano(Ed.).", RoboCup-97:Robot soccer World Cup I,Springer,1998.

[6] M.Ohta: "Learning Cooperative Behaviors in RoboCup Agents.H.Kitano(Ed.).", RoboCup-97:Robot Soccer World Cup I,Springer,1988.

[7] M.Tan: "Multi-agent reinforcement learning:Independent vs. cooperative agents", Proceedings of the Tenth International Conference on Machine Learning, pp.330-337.

[8] M.L.Littman: "Markov games as a framework for multi-agent reinforcement learning", In Proc.of the 11th International Conference on Machine Learning, pp.157-163,1994

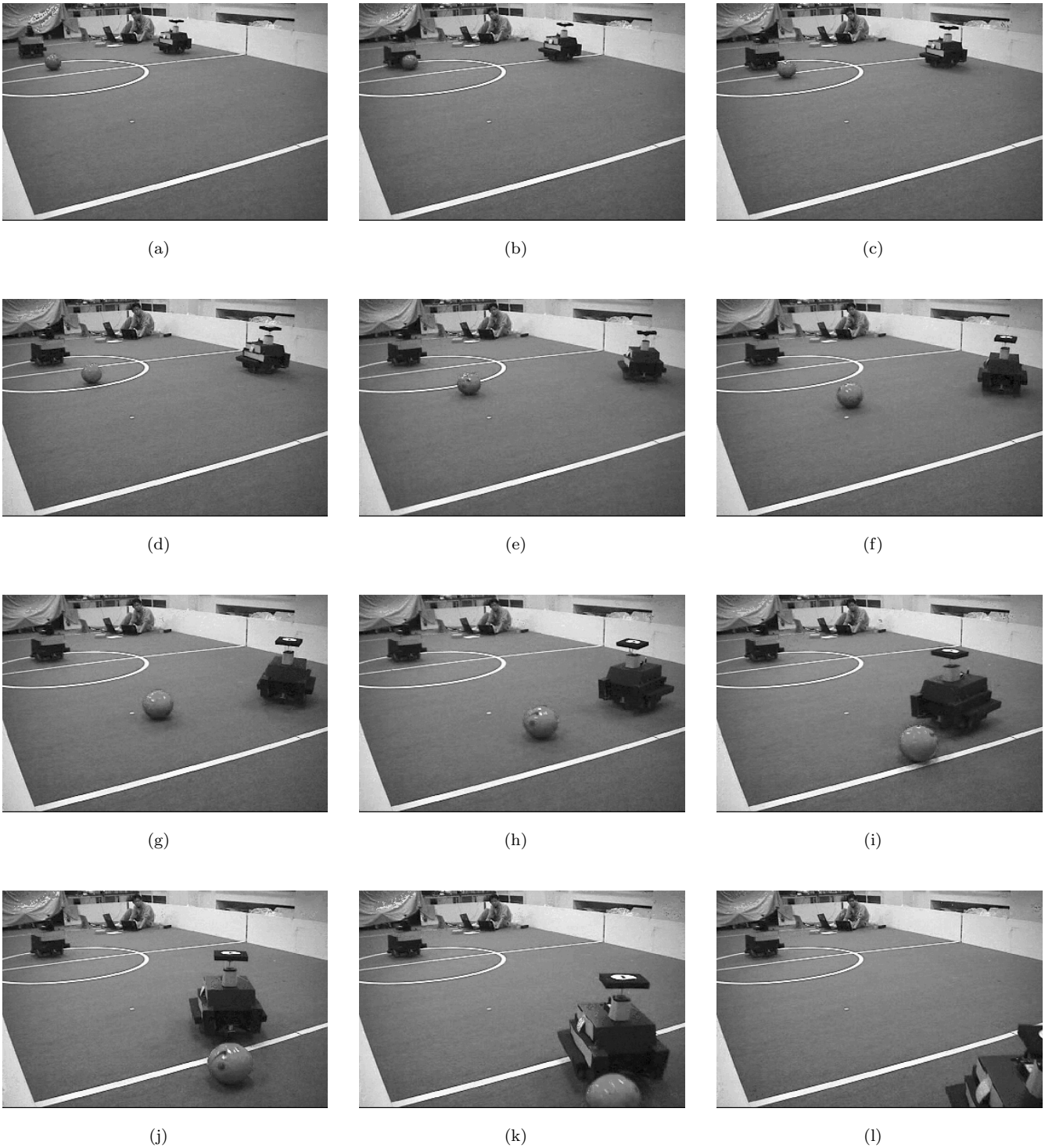
[9] M.Asada,E.Uchibe, and K.Hosoda: Cooperative Behavior Acquistion for Mobile Robots in Dynamically Changing Real World via Vision-Based Reinforcement Learning and Development. Artificial Intelligence, Vol.110,pp.275-292,1999.

[10] M.Asada,H.Kitano,I.Noda,and M.Veloso: "RoboCup:Today and tomorrow - what we have learned", Artificial Intelligence,pp.193-214,1999.

[11] C.J.C.H.,Watkins, and Dayan P: "Technical note:Q-learning", Machine Learning, Vol.8,pp.279-292,1992.

[12] M.Asada,S.Noda,S.Tawaratsumida,and K.Hosoda: "Vision-Vased Reinforcement Learning for Purposive Behavior Acquistion", Proc.of IEEE Int.Conf.on Robotics and Automation, pp.146-153,1995

[13] K.Yamazawa,Y.Yagi and M.Yachida: "Obstacle avoidance with omnidirectional image sensor hyperomni vision", In Proc.of IEEE Int.Conf.on Robotics and Automation, pp.1062-1067,1995.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 14: *A sequence of cooperative behavior by real robots*