Learning of Soccer Player Agents Using a Policy Gradient Method : Coordination Between Kicker and Receiver During Free Kicks

Harukazu Igarashi arashi50@sic.shibaura-it.ac.jp
Information Science and Engineering
Shibaura Institute of Technology
Tokyo, 135-8548, Japan

Koji Nakamura k-naka@rio.odn.ne.jp
Information Science and Engineering
Shibaura Institute of Technology
Tokyo, 135-8548, Japan

Seiji Ishihara ishihara@hiro.kindai.ac.jp
Electronic Engineering and Computer Science
Kinki University
Hiroshima, 739-2116, Japan

Abstract

As an example of multi-agent learning in soccer games of the RoboCup 2D Soccer Simulation League, we dealt with a learning problem between a kicker and a receiver when a direct free kick is awarded just outside the opponent's penalty area. We propose how to use a heuristic function to evaluate an advantageous target point for safely sending/receiving a pass and scoring. The heuristics include an interaction term between a kicker and a receiver to intensify their coordination. To calculate the interaction term, we let a kicker/receiver agent have a receiver's/kicker's action decision model to predict a receiver's/kicker's action. Parameters in the heuristic function can be learned by a kind of reinforcement learning called the policy gradient method. Our experiments show that if the two agents do not have the same type of heuristics, the interaction term based on prediction of a teammate's decision model leads to learning a master-servant relation between a kicker and a receiver, where a receiver is a master and a kicker is a servant.

Keywords: RoboCup, Soccer Simulation, Multiagents, Policy-Gradient methods, Reinforcement Learning.

1. INTRODUCTION

Recently, much work is being done on the learning of coordination in multi-agent systems [1, 2]. The RoboCup 2D Soccer Simulation League [3] is recognized as a test bed for such research because there is no need to control real robots and one can focus on learning coordinative behaviors among players. However, multi-agent learning continues to suffer from several difficult problems such as state-space explosion, concurrent learning [4], incomplete perception [5], and credit assignment [2]. In the games of the Robocup 2D Soccer Simulation League, the state-space explosion problem is the most important and difficult. Solving it is the main objective of this paper. However, the other three problems should be considered in varying degrees.

As an example of multi-agent learning in a soccer game, we dealt with a learning problem between a kicker and a receiver when a direct free kick is awarded just outside the opponent's penalty area. The kicker must make a shoot or pass the ball to the receiver to score a goal. To which point in the soccer field should the kicker kick the ball and the receiver run in such a

situation? We propose how to use a heuristic function to evaluate an advantageous target point for safely sending/receiving a pass and scoring. The heuristics include an interaction term between a kicker and a receiver to intensify their coordination. To calculate the interaction term, we let a kicker/receiver agent have a receiver's/kicker's action decision model to predict a receiver's/kicker's action. The soccer field is divided into cells whose centers are candidate targets of a free kick. The target point of a free kick is selected by a kicker using Boltzmann selection with the heuristic function. The heuristic function makes it possible to handle a large space of states consisting of the positions of a kicker, a receiver, and their opponents. Parameters in the function can be learned by a kind of reinforcement learning called the policy gradient method. The point to which a receiver should run to receive the ball is concurrently learned in the same manner.

We found the following two points from our learning experiments. First, if a kicker and a receiver have the same type of heuristics in their action evaluation functions, they obtain policies similar to each other by learning and that makes both action decisions agree well. Second, if the two agents do not have the same type of heuristics, the interaction term based on prediction of a teammate's decision model leads to learning a master-servant relation between a kicker and a receiver, where a receiver is a master and a kicker is a servant. This paper will present some clues to multi-agent learning problems through solving this type of free-kick problems.

2. COORDINATION OF SOCCER AGENTS

2.1 Cooperative Play in RoboCup 2D Soccer Simulation

Reinforcement learning is widely used [6,7] in the research areas of multi-agent learning. In the RoboCup 2D Soccer Simulation League, Andou used Kimura's stochastic gradient ascent (SGA) method [8] to learn the dynamic home positions of 11 players [9]. Riedmiller et al. applied TD learning to learn such individual skills as intercepting the ball, going to a certain position, or kicking and selecting those individual skills [10]. They dealt with attacking problems with 2v1 (2 attackers and 1 defender), 2v2, 3v4, and 7v8. Stone et al. studied keepaway problems with 3v2 [11] and half-field offense problems with 4v5 [12] using Sarsa [6] to learn the selection of macro behaviors such as ball holding, passing, dribbling, and shooting.

2.2 Coordination at Free Kicks

In the previous section, we cited several researches on the cooperative behaviors of soccer agents. However, a crucial problem remains. In their research, each agent apparently learns its policy of action selection "autonomously" to complete the received task. However, Riedmiller et al. assumed that all agents share input information, which are the x-y positions of all players and the ball, with other agents [10]. Stone et al. used other agents' experiences, which are time-series data on state, action, and reward, to accelerate learning in a large problem [12]. For that purpose, agents must communicate their experiences to their partners to facilitate information sharing among themselves. If agents share input information or experiences with other agents, all agents will obtain the same value function by learning. That will simplify the realization of various cooperative plays among agents. However, if an agent's observation is imperfect or uncertain as in games of the RoboCup 2D Soccer Simulation League, all agents cannot share the same input information with other agents. If communication between agents is not perfect, they cannot share their experiences with other agents. Moreover, if only agents that have identical value functions are assumed, agent individuality and division of roles among them may not emerge from agents' learning. In the next section, we propose a method where all agents learn autonomously without assuming perfect communication or identical input information.

3. LEARNING SOCCER AGENTS BY A POLICY GRADIENT METHOD

3.1 Policy Gradient Method

A policy gradient method is a kind of reinforcement learning scheme that originated from Williams' REINFORCE algorithm [13]. The method locally increases and maximizes the expected reward per episode by calculating the derivatives of the expected reward function of the parameters included in a stochastic policy function. This method, which has a firm mathematical basis, is easily applied to many learning problems. It is extended by Kimura to learning problems in Partially Observable Markov Decision Processes (POMDPs) , which is known as Stochastic Gradient Ascent (SGA) method [8]. Moreover, Igarashi et al. proved that a policy gradient method can be applied to learning problems in non-Markov Decision Processes [14,15]. They applied it to pursuit problems where the policy function consists of state-action rules with weight coefficients that are parameters to be learned [15]. In this paper, we take this approach for agents to learn how to make their action decisions.

3.2 Stochastic Policy for Action Decision

In this section, we propose a stochastic policy for determining kicker's and receiver's actions during direct free kicks. We divide the opponent's penalty area into 32 cells (5m×5m) and assume additional three cells (5m×4.5m) inside the goal net area, as shown in Fig. 1. A_{cell} denotes the set of center points of these 35 cells. Selecting a kicker's/ receiver's action $a_K/a_R(\in A_{cell})$ is defined as selecting a cell to the center of which a kicker/receiver should kick the ball or run. If the two agents select the same cell, i.e. $a_K=a_R$, their intentions agree well with each other and a pass between them would succeed with a high possibility.

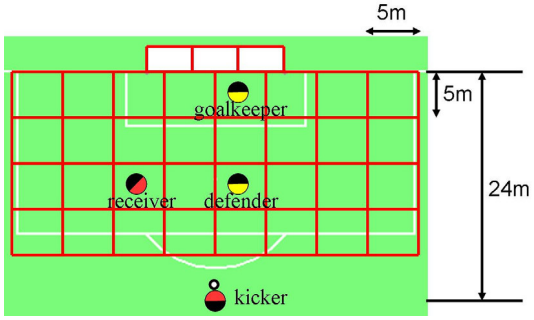


FIGURE 1: Example of Player Arrangements.

We consider objective function $E_\lambda(a_\lambda; s,\{\omega^\lambda_j\})\ (\le 0)$ of an agent such as

$$E_\lambda\left(a_\lambda;s,\omega^\lambda\right)=-\sum_i \omega^\lambda_j\cdot U^\lambda_j\left(a_\lambda;s\right)\ , \tag{1}$$

where functions $U^\lambda_j(a_\lambda;s)\ (\ge 0)$are the *j*-th heuristics that evaluate action $a_\lambda(\in A_{cell})$. Symbol λ $(\in\{K,R\})$ indicates a type of agent, where *K/R* means a kicker/receiver. State $s\ (\in S)$ means a state of the whole multi-agent system, which includes information of all players on the field and the ball. A set of parameters $\{\omega^\lambda_j\}$ is denoted simply by ω^λ in the left-hand side of (1). Next, we define agent's policy $\pi_\lambda(a_\lambda;s,\omega^\lambda)$ by a Boltzmann distribution function:

$$\pi_\lambda\left(a_\lambda;s,\omega^\lambda\right)\equiv\frac{e^{-E\left(a_\lambda;s,\omega^\lambda\right)/T}}{\sum_{x\in A_{cell}} e^{-E\left(x;s,\omega^\lambda\right)/T}}\ . \tag{2}$$

Note that action a_λ with a lower value of $E_\lambda(a_\lambda; s,\omega^\lambda)$, which means an action with a higher value of $U_j^\lambda(a_\lambda;s)$, is selected by (2) with a higher possibility. Weight parameters ω^λ in (1) and (2) are determined by a policy gradient method summarized in the next section.

3.3 Autonomous Action Decision and Learning

For the autonomous action decisions and the learning of each agent, we approximate policy function $\pi(a;s)$ for the whole multi-agent system by the product of each agent's policy function $\pi_\lambda(a_\lambda;s, \omega^\lambda)$ in (2) as [15,16]

$$\pi(a;s) \approx \prod_\lambda \pi_\lambda\left(a_\lambda;s,\omega^\lambda\right), \tag{3}$$

where $a=(a_K,a_R)$.

In (3), it seems that the correlation among agent action decisions is neglected. However, each agent can see all other agent states and use them in its policy function $\pi_\lambda(a_\lambda;s, \omega^\lambda)$. Thus, the approximation in (3) will contribute to learn coordination among agents. Note that each agent cannot get perfect information on state s and communication among agents is limited in games of the RoboCup 2D Soccer Simulation League. Therefore, agents cannot get accurate and complete information on the whole system nor share their observations with each other perfectly.

At the end of each episode, common reward r is given to all agents after evaluating results and behaviors of the whole agent system. That is a promising idea to avoid causing concurrent learning problems. The derivative of expectation of reward E[r] for parameter ω^λ is given as

$$\frac{\partial E[r]}{\partial \omega^\lambda} \approx E\left[r\sum_{t=0}^{L-1} e_{\omega^\lambda}(t) \right], \tag{4}$$

if we use (3) and assume that $E_{\lambda'}(a_{\lambda'};s)$ $(\lambda' \neq \lambda)$ does not depend on ω^λ. L is the size of an episode.

With (3), characteristic eligibility e_ω on the right-hand side of (4) can be written as [15,16]:

$$e_{\omega^\lambda}(t) \equiv \frac{\partial}{\partial \omega^\lambda} \ln \pi_\lambda\left(a_\lambda(t);s(t),\omega^\lambda\right), \tag{5}$$

where $a_\lambda(t)$ and $s(t)$ are the action and state of agent λ at discrete time t.

The derivative of E[r] in (4) leads to the learning rule of parameters ω^λ as

$$\Delta\omega^\lambda = \varepsilon \cdot r \sum_{t=0}^{L-1} e_{\omega^\lambda}(t), \tag{6}$$

where ε is a positive small number called learning ratio. Let assume that each agent makes an action decision by policy π_λ in (2) only once at the start of every episode, i.e. at t=0, and updates ω^λ by the learning rule in (6) at the end of every episode. The learning rule for agent λ is given by

$$\Delta\omega_j^\lambda = \varepsilon \cdot r \cdot \frac{1}{T}\left[U_j^\lambda(a_\lambda;s) - \sum_{x\in A_{cell}} U_j^\lambda(x;s)\pi_\lambda\left(x;s,\{\omega_j^\lambda\}\right) \right] \cdot \tag{7}$$

4. FREE-KICK PROBLEM AND LEARNING EXPERIMENTS

4.1 Arrangement of Players

We only consider the part of a soccer field used in the RoboCup 2D Soccer Simulation League. A kicker and a receiver are agents that learn team play during free kicks. There are a defender and a goalie that are opponents, who do not learn anything. An example of the arrangement of the four players is shown in Fig. 1. The origin of the coordinate axes is located at the center mark of the soccer field. The x/y-axis is set parallel to the touch/goal line.

A kicker is only given the role of kicking the ball during direct free kicks. The x-coordinate of the free-kick position is fixed to 24 m from the opponent goal line, while the y-coordinate is selected at random. A receiver is assumed to run to receive the pass and to immediately shoot toward the goal. The receiver is set at random to a cell that is not an offside position and not any of the three cells behind the goal's mouth. An opponent goalie/defender is randomly located in the goal/penalty area. The two defenders try to intercept the pass and thwart the two offense agents. In addition to the four players, a coach-client called a "trainer" changes the play mode from kickoff to free kick, sets the players and the ball, watches the game, and informs the two agents whether a kicker's pass was safely received and whether their shots successfully scored a goal.

4.2 Heuristics Used in Objective Function

In learning experiments, four kinds of heuristic functions $\{U_i^{\lambda}(a_{\lambda})\}$ (i=1,2,3,4) were used to evaluate the suitability of the selected cell for kickers and receivers. In this section, let cell k be the one selected by kicker's/receiver's action a_{λ}. $U_1^{\lambda}(a_{\lambda};s)$ considers the existence of opponents in the pass direction. $U_2^{\lambda}(a_{\lambda};s)$ expresses heuristics where shooting from nearer to the goal mouth has a greater chance of scoring a goal. $U_3^{\lambda}(a_{\lambda};s)$ evaluates a distance between the center of cell k and the nearest opponent to the center of cell k. $U_4^{\lambda}(a_{\lambda};s)$ considers the distance between the center of cell k and the current receiver's position.

U_1^{λ}, U_3^{λ} and U_4^{λ} are heuristics for passing the ball safely, while U_2^{λ} is heuristics for making an aggressive pass. These four functions are normalized to avoid exceeding 10.0 and their strict definitions are given in Appendix.

4.3 Interaction Between Two Action Decisions

In this section, we introduce an interaction between the action decisions of a kicker and a receiver. If any discrepancy exists in the information that the two agents get or perceive, learning the cooperative play described in Section 3 may be difficult. In actual human soccer games, all teammates cannot be expected to share identical information and heuristic knowledge to make their decisions. For this reason, we consider another heuristics, U_5, an interaction term. That makes one agent select its action that fits well an action selected by a teammate agent. The interaction term helps accelerate cooperation between the two agents. We define this interaction term by function $U_5(a_K,a_R;s)$:

$$U_5(a_K,a_R;s) \equiv \left(-|X_{KR}|-|Y_{KR}|+50.0\right)/5.0, \tag{8}$$

where a_K/a_R is kicker's/receiver's action and (X_{KR},Y_{KR}) is a difference vector between a cell to which a kicker intends to kick the ball and a cell to which a receiver runs. Adding interaction term $U_5(a_K,a_R;s)$ in (8) to the objective function in (1) , we use the following objective functions $E_K(a_K;s, \omega^K)$ and $E_R(a_R;\, s,\omega^R)$ for a kicker and a receiver:

$$E_K\left(a_K;s,\omega^K\right) = -\sum_i^4 \omega_i^K \cdot U_i^K\left(a_K;s\right) - \omega_5^K \cdot U_5\left(a_K,a_R^*;s\right), \tag{9}$$

$$E_R\left(a_R;s,\omega^R\right) = -\sum_i^4 \omega_i^R \cdot U_i^R\left(a_R;s\right) - \omega_5^R \cdot U_5\left(a_K^*,a_R;s\right). \tag{10}$$

To calculate interaction term $U_5(a_K,a_R;s)$ in (8), a kicker/receiver needs information on the action that the other teammate is going to select. One solution is sending the information by say command. But completely assuming the sending and receiving of all teammate actions is neither realistic in actual human soccer games nor desirable even in games of the RoboCup 2D Soccer Simulation League. In this paper, we adopt a method in which an agent has the other teammate's action-decision model inside itself and uses it for predicting the teammate's next action without asking it of the teammate. Thus receiver's action a^*_R in (9) is an action predicted by a kicker, and kicker's action a^*_K in (10) is an action predicted by a receiver. The next section describes how to predict a teammate's action.

4.4 Prediction of a Teammate's Action

Let us discuss a kicker's action decision. Objective function $E_K(a_K;s,\omega^K)$ in (9) is used to determine kicker action a_K. Function $E_K(a_K;s,\omega^K)$ includes receiver's action a^*_R in its interaction term $U_5(a_K,a^*_R;s)$ shown in (8). Receiver's action a^*_R should be determined by minimizing receiver objective function $E_R(a_R;s,\omega^R)$ shown in (10). Function $E_R(a_R;s,\omega^R)$ includes kicker action a^*_K in $U_5(a^*_K,a_R;s)$. That is, determining kicker's action a_K needs receiver's action a_R and vice versa. To break this endless loop, we use receiver's/kicker's action a^*_R/a^*_K predicted by minimizing $E_R(a_R;s,\omega^R)/E_K(a_K;s,\omega^K)$ that has no interaction term $U_5(a_K, a_R;s)$: i.e.,

$$a_R^* = \arg\min_{a_R}\left[-\sum_{i=1}^{4}\omega_i^R U_i^R(a_R;s)\right], \tag{11}$$

$$a_K^* = \arg\min_{a_K}\left[-\sum_{i=1}^{4}\omega_i^K U_i^K(a_K;s)\right] \tag{12}$$

for calculating the right-hand sides of (9) and (10). This method represents that an agent predicts another agent's action using the other agent's action-decision model. The receiver and kicker models are represented by weight coefficients $\{\omega_i^R\}$ $(i=1,2,3,4)$ and $\{\omega_i^K\}(i=1,2,3,4)$ in (11) and (12), respectively. However, the values of the coefficients are updated every episode during learning. To make the prediction as accurate as possible, a kicker and a receiver teach each other the values of their own weight coefficients every ten episodes by say command in our experiments. This helps update a teammate's action-decision model and keeps it current. This teaching in games is not necessary at all if agents are not learning their policy functions.

4.5 Reward

We deal with a cooperative play between a kicker and a receiver during direct free kicks. For learning this cooperative play, a large reward must be given to the two agents only if a kicker's pass is successfully received by a receiver and a receiver's shot successfully scores a goal. For this purpose, reward r depends on the results of a kicker's passing and a receiver's shooting. In preliminary experiments, we defined reward function $r(\sigma)$ given to episode σ, such as

$r(\sigma) = -30.0$ if P_{pass}=*false*,
$r(\sigma) = 5.0$ if P_{pass}=*true* and P_{shot}=*false*,
and
$r(\sigma)=100.0$ if P_{pass}=*true* and P_{shot}=*true*.

Proposition $P_{\text{pass}}/P_{\text{shot}}$ means that a kicker's pass/receiver's shot is successful. Identical reward $r(\sigma)$ is given to a kicker and a receiver by a trainer agent who judges whether a pass from a kicker to a receiver and the receiver's shot has succeeded. An episode consists of 50 simulation cycles, and takes five seconds in actual time. When a kicker or a receiver successfully scores a goal, the episode is concluded even before 50 simulation cycles have been completed.

4.6 Learning Experiments for a Kicker and a Receiver

We made four experiments to verify whether a kicker and a receiver can simultaneously learn their policies by applying the policy gradient method described in Section 3 to free-kick problems under the following conditions. A kicker and a receiver play against a defender and a goalie, as shown in Fig. 1. We exploited the programs of Trilearn Base [17] for the defender and goalie of the opponent. Trilearn Base is a program based on the UvA Trilearn2003 team's program that won the RoboCup2003 championship. Trilearn Base's defender and goalie are excellent players, while Trilearn2003's high level strategy is not implemented in Trilearn Base. Experiment 1 assumes that a kicker and a receiver have the same type of objective function, as shown in (1). However, they do not have interaction term $U_5(a_K,a_R;s)$, which Experiment 2 considers. In Experiment 3, a kicker has an objective function consisting of U_1^K, U_2^K, and U_3^K, and a receiver has an objective function consisting of U_2^R, U_3^R, and U_4^R. In Experiment 4, interaction term $U_5(a_K,a_R;s)$ was added to the objective functions used in Experiment 3. Experiments 3 and 4 deal with cases where a kicker and a receiver do not have heuristics U_4^K and U_1^R for deciding their next actions, respectively. Note that a receiver immediately makes a shoot if he can get the ball because we are only concentrating on making a receiver learn where to run to receive a pass, not how to dribble a ball.

5. EXPERIMENTAL RESULTS AND DISCUSSION

Four learning experiments described in Section 4.6 were conducted under conditions where ε=0.1 and T=10.0. Temperature parameter T is not lowered during the learning of ω^λ but fixed to search the parameter space for different good combinations of agents' policies. The initial values of weight ω^λ are selected at random from an interval between 10.0 and 30.0. Figures 2-4 show the changes of the passing success rate, the scoring rate, and the expectation of reward while learning 2000 episodes, respectively. Only their averages over every 50 episodes are plotted in the three figures. Changing the initial values of ω^λ, all four experiments were carried out ten times, and the values plotted in Figs. 2-4 are ensemble averages of the ten sets of experiments. Fig. 5 shows a typical example of the change of weight coefficients $\{\omega_i^\lambda\}$in the ten trials.

5.1 Experiment 1

In Experiment 1, a kicker and a receiver use the same type of objective functions consisting of four heuristics from U_1^λ to U_4^λ. Figs. 5a and b shows that the ratio of the magnitudes of a kicker's $\{\omega_i^K\}$ approaches that of a receiver's $\{\omega_i^R\}$ as learning continues, even if the initial values of a kicker's $\{\omega_i^K\}$ are different from a receiver's $\{\omega_i^R\}$. The passing success rate increases from 42% to 78% in Fig. 2, and the scoring rate increases by 4 times from 8% to 32% in Fig. 3. These two increases mean that both the kicker and the receiver have acquired the same policy for action decisions and that successful passes contribute to scoring goals.

5.2 Experiment 2

In Experiment 2, interaction term U_5 is added to the objective function used in Experiment 1. Figs. 5c and 5d show the same tendency of $\{\omega_i^\lambda\}$ as in Experiment 1 except ω_5^λ. The value of kicker's ω_5^K became much larger than receiver's ω_5^R. This means that a kicker follows a receiver's action to realize cooperative play. As a result, the scoring rate is improved and becomes slightly larger than in Experiment 1 where there is no interaction term U_5.

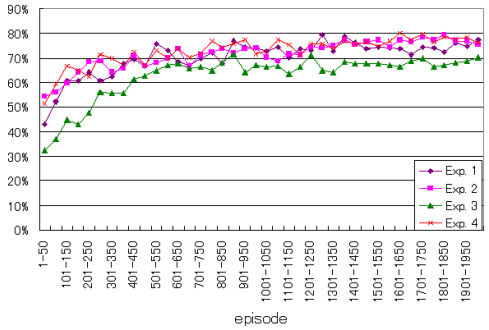


FIGURE 2: Passing Success Rate.

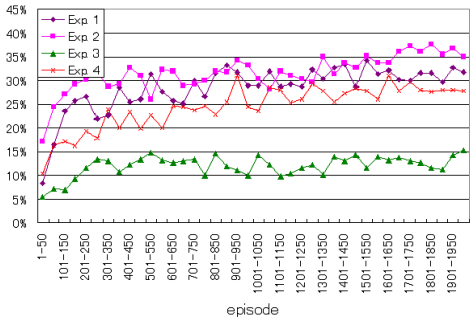


FIGURE 3: Scoring Rate.

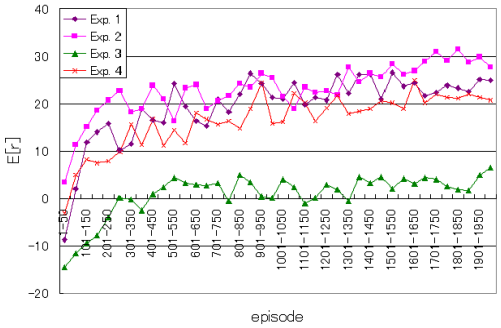


FIGURE 4: Expectation of Reward.

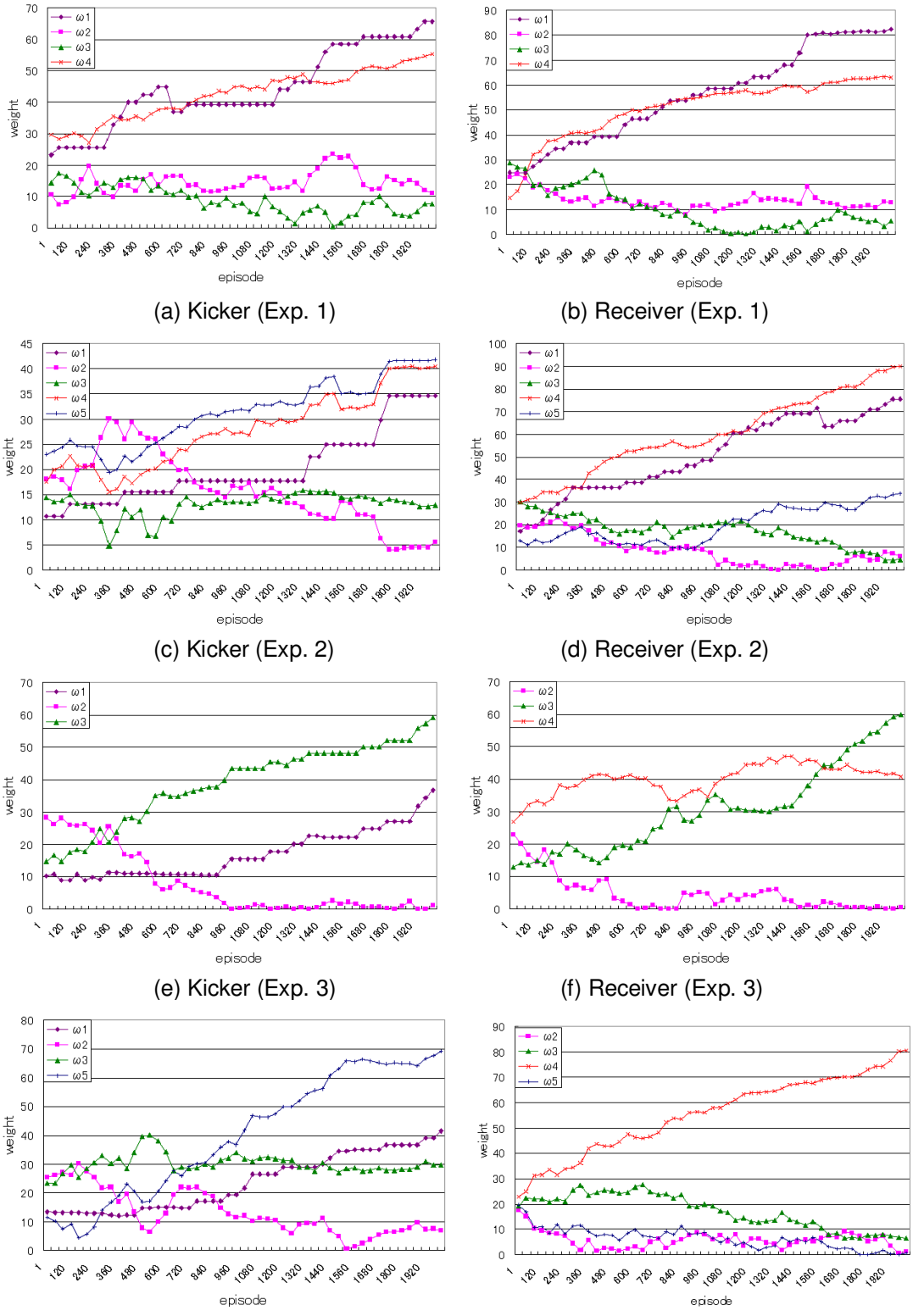
5.3 Experiment 3

In Experiment 3, a kicker has a set of heuristics, U_1^K, U_2^K, and U_3^K, which is different from a receiver's set of heuristics, U_2^R, U_3^R, and U_4^R. Passing success rate increases from 30% to 70%, as shown in Fig. 2, and the scoring rate increases by three times, from 5% to 15%, after learning 2000 episodes, as shown in Fig. 3. Figs. 5e and 5f show a common tendency on the changes of ω_2^λ and ω_3^λ for kickers and receivers, who realize cooperative play by making the common two heuristics, U_3^λ and U_4^λ, dominant in their policies. However, while watching their actual play in Experiment 3, we observed that their actions do not completely agree with each other. A small discrepancy exists between the cell to which a kicker passes the ball and the cell to which a receiver runs to receive it. This small discrepancy allows the defenders and the goalie time for defending. That is proved by the very low scoring rate shown in Fig. 3, whereas the passing success rate is not so bad when compared with other three experiments in Fig. 2.

5.4 Experiment 4

Interaction term U_5 is added to the policy used in Experiment 3. After learning 2000 episodes, the observed passing success rate is about 6 points higher than that in Experiment 3 (Fig. 2). Fig. 3 shows that the scoring rate increases about twofold the rate observed in Experiment 3. Moreover, expectation of reward E[r] increases by three times that obtained in Experiment 3 (Fig. 4).

In Figs. 5g and 5h, kicker's ω_5^K and receiver's ω_4^R become much larger than the other weight coefficients. Since ω_5^R nearly becomes zero after learning (Fig. 5h), a receiver does not try to predict a kicker's action. The receiver does not leave its current position, because ω_4^R is very large and the other weights are nearly zero. That is, the receiver stays and waits at his own current position. A kicker understands the receiver's intention and follows the receiver's choice. This kicker's policy was obtained by increasing weight ω_5^R of interaction term U_5 because a kicker does not have heuristics U_4^K that favors a pass close to a receiver. If the two agents do not have the same type of heuristics, the interaction term based on prediction of a teammate's decision model accelerates learning a master-servant relation between a kicker and a receiver, where a receiver is a master and a kicker is a servant as shown in Fig.6.



(a) Kicker (Exp. 1)

(b) Receiver (Exp. 1)

(c) Kicker (Exp. 2)

(d) Receiver (Exp. 2)

(e) Kicker (Exp. 3)

(f) Receiver (Exp. 3)

(g) Kicker (Exp. 4)

(h) Receiver (Exp. 4)

FIGURE 5: Example of Change of Weight Coefficients $\{\omega_i^{\lambda}\}$ during Learning Process in Experiments 1-4.

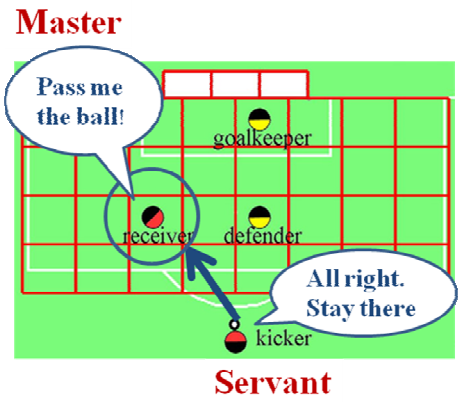


FIGURE 6: Master-servant Relation Obtained by the Interaction Term in Experiment 4, where a Kicker and a Receiver do not Have the Same Type of Heuristics.

6. CONSLUSION & FUTURE WORK

As an example of multi-agent learning problems, we considered a learning problem between a kicker and a receiver when a direct free kick is awarded just outside the opponent's penalty area in the soccer games of the RoboCup 2D Soccer Simulation League. We proposed a function that expresses heuristics for evaluating how target position is advantageous for safely sending/ receiving a pass and scoring a goal. This evaluation function does not depend on the dimension of the soccer field and the number of players. The weight coefficients in the function were learned by a policy gradient method. However, we did not try to make the weight coefficients converged to certain values, for example, by decreasing ε in learning rule (7). Our method needs criteria for deciding which weight coefficients should be best.

The heuristics includes an interaction term between a kicker and a receiver to intensify their coordination. The interaction term works to make an agent follow the teammate's action. To calculate the interaction term, we let a kicker/receiver agent have a receiver's/kicker's action-decision model to predict the teammate's action. Except for the interaction term, information on the action-decision model is exchanged with a teammate at a certain time interval, i.e, every 10 episodes during the learning process. The results of our learning experiments show that even if a kicker's and a receiver's heuristics are different, scoring rate is increased about two times of that obtained by learning without the interaction term. This means that adding an interaction term, which makes an agent follow a teammate's action, and predicting the teammate's action using the teammate's action-decision model are very effective for two agents to acquire a common policy, even if they do not completely have identical action-decision heuristics. However, our method needs human heuristics enough for solving the target problem, and the action-decision model of a teammate for predicting the teammate's action.

In the future, our agents will have and learn both passer and receiver policies so that they can learn wall-passes. An agent must switch its role from a receiver to a passer, and vice versa. We will also apply our method to more general cooperative play such as keeping the ball away from opponents and making through passes. Using action-decision models of teammates would be very useful for learning such team play.

7. REFERENCES

1. G. Weiss, S. Sen, editors. "*Adaption and Learning in Multi-agent Systems*", Springer-Verlag, Germany, 1996

2. S. Sen, G. Weiss, "*Learning in Multiagent Systems*". In G. Weiss, editor, "*Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence*", pp. 259-208, The MIT Press, 1999

3. RoboCup 2D Soccer Simulation League, http://sourceforge.net/apps/mediawiki/sserver/index.php?title=Main_Page (access time: 14.03.2011)

4. S. Arai, K. Miyazaki. "*Learning Robust Policies for Uncertain and Stochastic Multi-agent Domains*". In Proceedings of 7th International Symposium on Artificial Life and Robotics, pp.179-182, 2002

5. W.S. Lovejoy. "*A survey of algorithmic methods for partially observed Markov decision processes*". Annals of Operations Research, 28(1): 47-65, 1991

6. R. S. Sutton, A. G. Barto."*Reinforcement Learning*", The MIT Press, 1998

7. L. P. Kaelbling, M. L. Littman and A. W. Moore. "*Reinforcement Learning: A Survey*". Journal of Artificial Intelligence Research, 4:237-285, 1996

8. H. Kimura, K. Miyazaki and S. Kobayashi. "*Reinforcement Learning in POMDPs with Function Approximation*". In Proceedings of the 14th International Conference on Machine Learning, pp. 152-160, 1997

9. T. Andou. "*Refinement of Soccer Agents' Positions Using Reinforcement Learning*". In H. Kitano, editor, "*RoboCup-97: Robot Soccer World Cup I"*, pp. 373-388, Springer-Verlag, Berlin, 1998

10. M. Riedmiller, T. Gabel. "*On Experiences in a Complex and Competitive Gaming Domain –Reinforcement Learning Meets RoboCup–*". In Proceedings of the 2007 IEEE Symposium on Computational Intelligence and Games (CIG2007), pp.17-23, 2007

11. P. Stone, G. Kuhlmann, M. E. Taylor and Y. Liu. "*Keepaway Soccer: From Machine Learning Testbed to Benchmark*". In A. Bredenfeld, A. Jacoff, I. Noda and Y. Takahashi, editors, "*RoboCup 2005: Robot Soccer World Cup IX*", pp. 93-105, Springer-Verlag, New York, 2006

12. S. Kalyanakrishnan, Y. Liu and P. Stone. "*Half Field Offense in RoboCup Soccer –A Multiagent Reinforcement Learning Case Study*". In G. Lakemeyer, E. Sklar, D. G. Sorrenti, T. Takahashi, editors, "*RoboCup-2006: Robot Soccer World Cup X*", pp.72-85, Springer-Verlag, Berlin, 2007

13. R. J. Williams. "*Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning*". Machine Learning, 8(3-4): 229-256, 1992

14. H. Igarashi, S. Ishihara and M. Kimura. "*Reinforcement Learning in Non-Markov Decision Processes -Statistical Properties of Characteristic Eligibility-*", IEICE Transactions on Information and Systems, J90-D(9):2271-2280, 2007 (in Japanese). This paper is translated into English and included in The Research Reports of Shibaura Institute of Technology, Natural Sciences and Engineering (ISSN 0386-3115), 52(2): 1-7, 2008

15. S. Ishihara, H. Igarashi. "*Applying the Policy Gradient Method to Behavior Learning in Multiagent Systems: The Pursuit Problem*", Systems and Computers in Japan, 37(10):101-109, 2006

16. L. Peshkin, K. E. Kim, N. Meuleau and L. P. Kaelbling. "*Learning to cooperate via policy search*". In Proceedings of 16th Conference on Uncertainty in Artificial Intelligence (UAI2000), pp. 489-496, 2000

17. J. R. Kok. "*UvA Trilearn*", http://remote.science.uva.nl/~jellekok/robocup/ (access time: 05.01.2011)

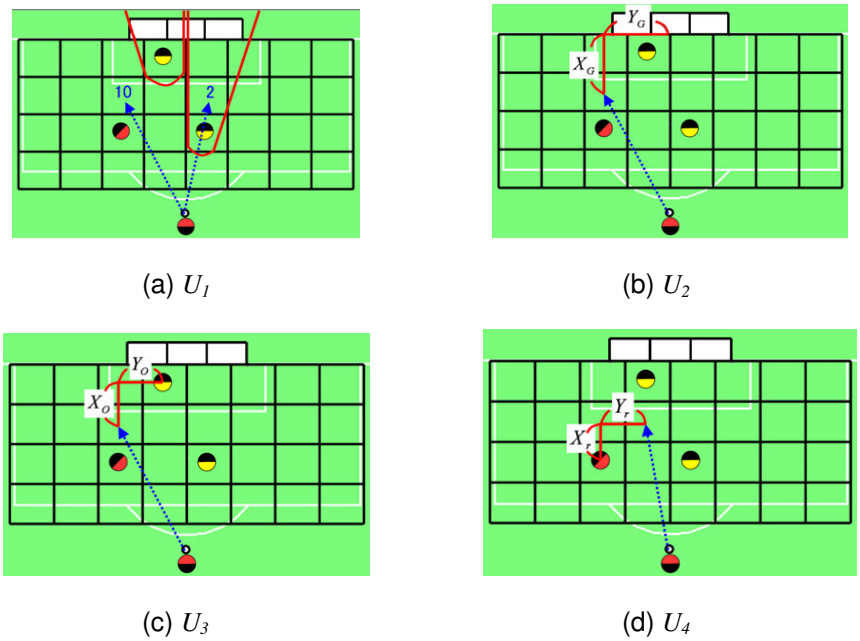
Appendix: Heuristic Functions from U_1^{λ} to U_4^{λ}

In our learning experiments, the following four heuristic functions $\{U_i^{\lambda}(a)\}$ (i=1,2,3,4) were used to evaluate the suitability of the selected cell for kickers and receivers. In this section, let cell k be the one selected by kicker or receiver action a. All heuristic functions are normalized to avoid exceeding 10.0. Fig. 7 illustrates definitions of the four heuristic functions schematically.

(i) $U_1^{\lambda}(a;s)$: considers the existence of opponents in the pass direction and is defined by

$$U_1^{\lambda}(a;s) = \begin{cases} 2.0 & \text{if } \theta_{\text{p-opp}} \le 15^{o} \text{ and } d_{\text{k}} \ge d_{opp} - 2.0 \\ 10.0 & \text{else} \end{cases}, \tag{13}$$

where $\theta_{\text{p-opp}}$ is an angle between a pass direction and a direction to an opponent. d_{k} is the distance between the center of cell k and a kicker. d_{opp} is the distance between a kicker and the nearest opponent to the center of cell k.



(a) U_1 (b) U_2

(c) U_3 (d) U_4

FIGURE 7: Heuristics $\{U_i^{\lambda}\}$ (i=1,2,3,4).

(ii) $U_2^{\lambda}(a;s)$: expresses a heuristics where shooting from nearer to the goal mouth has a greater chance of scoring a goal. It is defined by

$$U_2^{\lambda}(a;s) = \left(-|X_G| - |Y_G| + 37.5\right)/3.5 , \tag{14}$$

where vector $(X_{\text{G}},Y_{\text{G}})$ is a distance vector between the center of cell k and the center of the goal mouth.

(iii) $U_3^{\lambda}(a;s)$: evaluates a distance between the center of cell k and the nearest opponent to the center of cell k as follows:

$$U_3^{\lambda}(a;s) = \left(|X_0| + |Y_0|\right)/5.0 , \tag{15}$$

where vector (X_0, Y_0) is a distance vector between the center of cell k and the position of the nearest opponent. Heuristics U_3 means that it is desirable to receive a pass near a place without opponents.

(iv) $U_4^{\lambda}(a;s)$: considers the distance between the center of cell k and the current receiver's position. It is defined by

$$U_4^{\lambda}(a;s) = \left(-|X_r| - |Y_r| + 50.0\right)/5.0 \ , \tag{16}$$

where vector (X_r, Y_r) is a distance vector between the center of cell k and a receiver. If the distance is small, receivers can easily receive a pass.