SICE Annual Conference 2007
Sept. 17-20, 2007, Kagawa University, Japan

Implicit estimation of other's intention without direct observation of actions in a collaborative task: Situation-Sensitive Reinforcement Learning

Tadahiro Taniguchi[1,3] Kenji Ogawa[2] and Tetsuo Sawaragi[2]

[1]Graduate School of Informatics, Kyoto University, Kyoto, Japan.
(Tel : +81-75-753-3592; E-mail: tadahiro@tanichu.com)
[2]Graduate School of Engineering, Kyoto University, Kyoto, Japan.
(Tel : +81-75-753-5266; E-mail: sawaragi@me.kyoto-u.ac.jp)
[3]Japan Society for the Promotion of Science, Japan.

Abstract: An agent in a multi-agent environment should adapt to the diversities of dynamics that are caused by changes in the physical properties of the task environment and in social situations concerning how the partner is shifting his/her behaviors to achieve the task. When the partner's intention changes in the latter, a collaborator agent has to notice this from what is observed in the shared-task environment and to explore how to adaptively collaborate with the partner. A Situation-Sensitive Reinforcement Learning (SSRL) architecture is presented in this paper. SSRL enables a collaborator agent to implicitly estimate the partner's goal. The mathematical basis of the implicit estimates is also addressed. A simple truck-pushing task by a pair of agents is presented as a testbed example, and the simulation results show that organized collaboration could be achieved by an agent embedded with our model in adapting to the partner's intentional strategic changes.

Keywords: Reinforcement learning, multi-agent systems, cooperative systems, modular learning.

1. INTRODUCTION

Cooperation by all participating agents is necessary in many multi-agent tasks: e.g., playing football and carrying large tables. An agent in cooperative tasks, has to estimate the other agent's intention explicitly or implicitly. Implicit extimates are achieved by watching how the state variables he/she observes change. The state variables are usually considered to be the objectives to be controlled. The learning process of physical skills to control the target system and how to communicate with the partner agent seem to be closely connected in such cooperative tasks.

From the viewpoint of computational neuroscience, Wolpert et al. [10] suggested that MOSAIC, which is a modular learning architecture representing part of the human central nervous system (CNS), which acquires multiple internal models that play an essential part not only in adapting to the physical dynamic environment, but also in communicating with other autonomous agents. At the same time, Taniguchi et al. described an integrative learning architecture for spike timing-dependent plasticity (STDP) and the reinforcement learning schemata model (RLSM) [8], [7]. The learning architecture enables an autonomous robot to acquire behavioral concepts and signs representing the situation where the robot should initiate the behavior. They called this process "symbol emergence". The symbolic system plays a important role in human social communication. However, both Wolpert and Taniguchi seem to insist that the learning architecture constructing an adaptive symbolic system is more important than a static constructed symbolic system. However, is a symbolic system is necessary to communicate one's intentions to other agent?

One solution is to communicating one's intentions to another person is to express one's intention directly, e.g. by pointing to the goal and by commanding the otherbto act. The method of communication requires a shared symbolic system as a basic premise if the symbolic system, which is used in this communication, is completely shared by the participants of the cooperative-task environment. The receiver of the message can estimate the emitter's intentions based on externalized signs. This process is called the "explicit estimation" of the other's intention. In contrast, we occasionally undertake collaborative tasks with somebody without saying anything. Even if a leader says nothing to his followers, they can often perform the task by estimating the leader's intentions or goals based on their subjective motor outputs and/or sensory inputs that are causally related to the leader's intentions. Followers cannot observe any information except for their own sensorymotor information. However, they can estimate the

leader's intentions. To iterate, this processis called "implicit estimation" of the leader's intention.

It is important to treat physical-learning and social-learning processes simultaneously to discuss implicit estimates of the other's intention in a collaborative task from the viewpoint of computation, . We assumed that such a distributed learning architecture would be essential for an autonomous agent to cope not only with a physically dynamic environment but also with a socially dynamic environment that included changes in the other agent's intentions.

The computational model for implicit estimates of the other !s intention is described in this paper based on a framework of modular reinforcement learning. The computational model is called situation-sensitive reinforcement learning (SSRL). The mathematical basis for the implicit estimation of other's intention based on the framework of reinforcement learning is also provided. Furthermore, a simple truck-pushing task by a pair of agents is presented to evaluate the learning architecture.

2. SITUATION-SENSITIVE REINFORCEMENT LEARNING ARCHITECTURE

It is important for autonomous agents to accumulate the results of adaptation to various environments to cope with dynamically changing environments. Acquired concepts, models, and policies should be stored for similar situations that are expected to occur in the near future. Not only learning a certain behavior and/or a certain model, but also the obtained behaviors, policies, and models is essential to describe such a learning process,. Many modular learning architectures [3], [1] and hierarchical learning architectures [5], [4] have been proposed to describe this learning process. This section introduces such a modular-learning architecture called the situation-sensitive reinforcement learning architecture (SSRL). This enables an autonomous agent to distinguish changes the agent is facing in situations, and to infer the partner agent's intentions.

2.1 Discrimination of intentions based on changes in dynamics

"Intention" in everyday language denotes a number of meanings. Therefore, a perfect computational definition of "intention" is impossible. We simply consider "intention" as a goal the agent is trying to achieve in this paper. In the framework of reinforcement learning, an agent's goal is represented by a

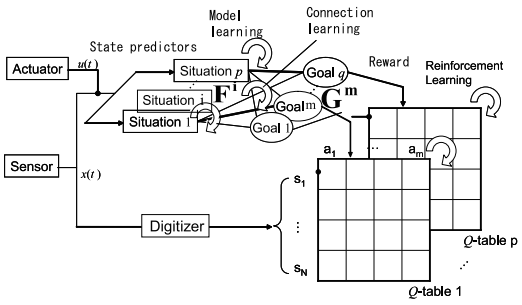

Fig. 1 Situation-Sensitive Reinforcement Learning architecture

reward function. Therefore, an agent who has several intentions has several internal goals, i.e. several internal reward functions, G^m. If a internal reward function, G^m, is selected, a policy, u^m, is selected and modified to maximize the cumulative future internal reward through interactions with the task environment.

The collaborative task consists of two agents in what follows. The system is described as

$$y = f(x, u_1, u_2^m) + n, \quad (1)$$
$$= f(x, u_1, u_2^m(x)) + n, and \quad (2)$$
$$= F^m(x, u_1) + n. \quad (3)$$

Here !$x is a state variable, u_i is i-th agent's motor output, and n is a noise term. We assumed that an agent would not be able to directly observe the other agent's motor output. In such cases, the environmental dynamics seem to be eq. **??** to the first agent. If the second agent changes its policy, the environmental dynamics for the first agent change. Therefore, in a physically stationary environment, the first agent can establish that the second agent has changed its intention by noticing changes in environmental dynamics.

The discussion can be summarized as follows. If the physical environmental dynamics, f, is fixed, agents can detect changes in the other agent(s intentions by detecting changes in subjective environmental dynamics, F.

We define "situation" as "how state variable x and motor output u change observed output y". In this case, a change in an agent's intentions leads to a change in the subjective situation of the other agent.

Fig. 1 is an overview of SSRL. SSRL has several state predictors, F^m, representing situations and internal goals, G^m, representing intentions. Each state

predictor F^m corresponds to each situation.

$$e_t^j = ||y_t - F^j(x_t, u_t)||^2, \quad (4)$$

$$P(j|\bar{e}_t^j) = \exp(-\frac{\bar{e}_t^j}{2\sigma^2}) / \sum_{k=1}^{p} \exp(-\frac{\bar{e}_t^k}{2\sigma^2}), \quad (5)$$

$$j^* = \arg\min_j P(j|\bar{e}_t^j), \quad (6)$$

where $\bar{e}_t^j$ is the temporal average of the prediction error, e_t^j, of the j-th state predictor, F^j. If averaged error $\bar{e}_t^j$ has a normal distribution when the system dynamics is F^j, the posterior probability, $P(j|\bar{e}_t^j)$, can be defined based on the Bayesian framework above under the condition that there is no other information. If there are no adequate state predictors in SSRL, the SSRL allocates one more state predictor based on hypothesis-testing theory [6].

This is an intermediate method for the MOSAIC model [11], [10], which is based on the Basian framework, and the schema model [6], which is based on hypothesis-testing theory. SSRL detects the current situation based on Eq. 6. During this an adequate state predictor is selected and assimilates the incoming experiences; SSRL acquires the state predictors by ridge regression based on the assimilated experiences.

2.2 Reinforcement Learning

Each policy corresponding to a goal is acquired by using reinforcement learning [2]. SSRL uses Q-Learning [9] in this paper. This method can be used to estimate the state-action value function, $Q(s, a)$, through interactions with the agent's environment. The optimal state-action value function directly gives the optimal policy. When we define $\mathcal{S}$ as a set of state variables and $\mathcal{A}$ as a set of motor outputs, and we assume the environment consists of a Markov decision process, the algorithm for Q-learning is described as

$$Q \leftarrow Q(s,a) + \alpha(r + \gamma V(s') - Q(s,a)),$$
$$V(s') = \max_{a' \in \mathcal{A}} Q(s', a'), and \quad (7)$$
$$u(s) = \operatorname*{argmax}_{a' \in \mathcal{A}} Q(s,a), \quad (8)$$

where $s \in \mathcal{S}$ is a state variable, $a \in \mathcal{A}$ is a motor output, $r(s, a)$ is a reward, and s' is a state variable at the next time step. In these equations, α is the learning rate and γ is a discount factor. After an adequate Q is acquired,the agent can utilize an optimal policy, u, as in Eq. 8. Boltzmann selection is employed during the learning phase.

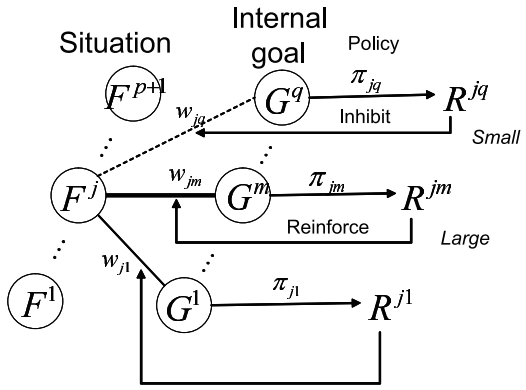

Fig. 2 Internal goal switching module

2.3 Switching Architecture of internal goals

An agent can detect changes in the other agent's intentions by distinguishing between situations he/she faces. However, the goals themselves cannot be estimated even if switching between several goals can be detected. Here, we describe a learning method, which enables an agent to estimate the other's intentions passively. The method requires three assumptions to be made.

A1 Physical environmental dynamics f do not change.

A2 Every internal goal is equally difficult to achieve.

A3 The leader agent always selects each optimal policy for each intention.

The mathematical explanation for these assumptions will be described in the next section. The rule to select the internal goals are described as

$$p(m|j) = \exp(Bw_{jm}) / \sum_{i=1}^{q} \exp(Bw_{ji}), \quad (9)$$

where $p(m|j)$ is the probability that G^m will be selected under situation, F^j, and B is the inverse temperature. The network connection, w_{jm}, between the current situation, F^j, and the current internal goal, G^m, is modified by the sum of the obtained reward, R_t^{jm}, during a certain period during the t-th trial, i.e.,

$$w_{jm} = \nu R_t^{jm} + (1 - \nu) w_{jm} \quad (10)$$

Here, ν is the learning rate. Eq. 10 shows that connection w_{jm} becomes strong if internal goal G^m is more easy to accomplish when the situation is F^j. Eq. 9 shows that an internal goal is more likely to be selected if its network connection is stronger than the other's. The abstract figure for the switching module is shown in Fig. 2. If the learning process for

the switching architecture of internal goals is preceded and converged, a certain internal goal corresponding to a situation is selected.

3. MATHEMATICAL BASIS FOR IMPLICIT ESTIMATES OF OTHER'S INTENTIONS

This section provides the mathematical basis for the implicit estimates of the other's intentions in this paper. The basis for this is simple. First, the Bellman equation for the i-th ($i = 1, 2$) agent of a system involving two agents are described as [1].

$$V_i^{\lambda}(x)|_{u_j} = \max_{u_i \in U_i} \sum_{x'} P(x'|x, u_i, u_j) \times [G^{\lambda}(x, x') + \gamma V_i^{\lambda}(x')], \quad (11)$$

where G^{λ} is a reward function for the λ-th goal, u_i is the i-th agent's motor output, and x' is the x in the next step. G^{λ} in this framework is not assumed to have motor outputs as variables of the function. The optimal value function for the i-th agent depends on the other agent's policy, u_j. Here, we define u_i^{λ} as the i-th agent's policy that maximizes the j-th agent's maximized value function.

$$u_i^{\lambda} = \operatorname*{argmax}_{u_i \in U_i} \max_{u_j \in U_j} \sum_{x'} P(x'|x, u_1, u_2) \times [G^{\lambda}(x, x') + \gamma V_i^{\lambda}(x')] and \quad (12)$$

$$V_i^{\lambda|\nu} \equiv V_i^{\lambda}|_{u_j^{\nu}}. \quad (13)$$

The assumptions, A2 and A3, we made in the previous section can be translated into the following,

A'2 : We assumed the j-th agent would use the controller, u_j^{λ}, and

A'3 : $V_i^{\lambda|\lambda}(x_0) = V_i^{\nu|\nu}(x_0)$,

where x_0 is the initial point of the task. The following relationship can easily be derived from the definition.

$$V_i^{\lambda|\lambda} = V_i^{\nu|\nu} \leq V_i^{\nu|\lambda}. \quad (14)$$

Therefore, the i-th agent's internal goal becomes the same as j-th agent's goal, if the i-th agent select a reward function that maximizes the value function under the condition that the j-th agent uses controller u_j^{λ}. When the initial point is not fixed, $V_i(x_0)$ is substituted by the averaged cumulative sum of rewards the $i-th$ agent obtains, who starts the task around the initial point, x_0. This leads us to the algorithm eq.10.

[1]In this section, we have assumed $i \neq j$ without making any remarks.

4. EXPERIMENT

We evaluate SSRL in this section. To fulfill all the assumptions made in Section 3 completely is difficult in a realistic task environment. The task described in this section roughly satisfies the assumptions, A'2 and A'3.

4.1 Conditions

We applied the proposed method to the truck-pushing task shown in Fig. 3. Two agents in the task environment, "Leader" and "Follower",cooperatively push a truck to various locations. Both agents can adjust the truck's velocity and the angle of the handle. However, a single agent cannot achieve the task alone because its control force is limited. In addition, the Leader has all fixed policies for all sub-goals beforehand, and holds a stake in deciding the next goal. However, the agents cannot communicate with each other. Therefore, an agent cannot "explicitly" estimate the other's intentions. The Follower perceives situation F^j by using SSRL, changes its internal goal G^m based on the situation, and learns how to achieve the collaborative task. The two agents output the angle of the handle, θ_L ! \$$\theta_F$, and the wheel's rotating speed, ω_L ! \$$\omega_F$. Here ! \$the final motor output to the truck, θ ! \$$\omega$, is defined as

$$\theta = K_{\theta}(\theta_L + \theta_F) and \quad (15)$$

$$\omega = K_{\omega}(\omega_L + \omega_F), \quad (16)$$

where K_{θ} and K_{ω} are the gain parameters of the truck. K_{θ} and K_{ω} were set to 0.5 in this experiment. The Leader's controller was designed to approximately satisfy the assumptions in Section 3. The controller in this experiment was a simple PD controller. The Follower's state, s, was defined as $s = [\rho, \alpha]$. The state space was digitized into 10×8 parts. The action space was defined as $\theta_F = \{-\pi/4, -\pi/8, 0, \pi/8, \pi/4\}$ and $\omega_F = \{0.0, 3.0\}$. As a result of the two agents' actions, the truck's angular velocity, Ω, was observed by the Follower agent. Ω, θ, and ω have a relationship of

$$\Omega \propto \omega \tan\theta. \quad (17)$$

The agents can carry the truck to a certain goal by cooperatively controlling Ω. The main state variables are shown in Fig. 4. Internal reward function G^m is defined as

$$G^m(x) = \begin{cases} 5 & \text{if } ||C - Goal_m|| < 1 \\ \kappa(1 - ||C - Goal_m||) & \text{otherwise,} \end{cases} \quad (18)$$

where C is the position of the truck, and $Goal_m$ is the position of the m-th goal.

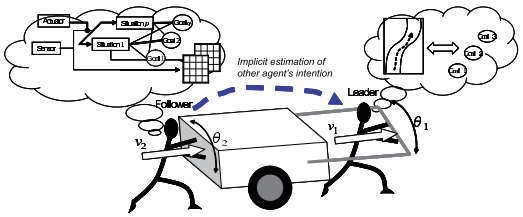

Fig. 3 Simple truck-pushing task by pair of agents

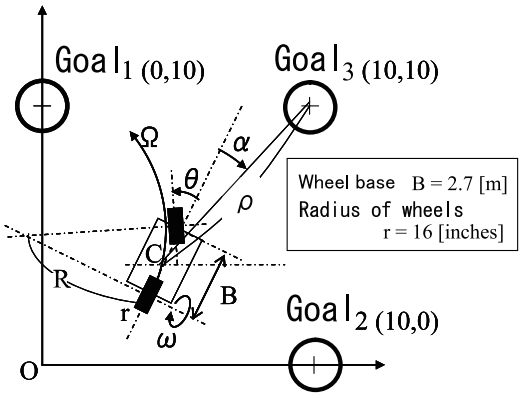

Fig. 4 State variables and parameters in task environment

4.2 Experiment 1: implicit estimates of other's intentions

Wwe fisrt conducted an experiment in which the Follower estimated the Leader's goal, where the Leader selected one of three sub-goals, and learned how to achieve the collaborative task (Fig. 5, top). There were three goals, and the Leader changed its goals from $G^1 - > G^2 - > G^3$ alternately every 1000 trials.

In contrast to simple reinforcement learning, the Follower agent not only has to learn the policies for the goals but also the state for predictors the relationship between the current situation and the internal goal by updating these parameters.

The 1000 trajectories of the truck corresponding to all 1000 trials in this experiment are shown in Figs. 6 and 7. Simple Q-learning with explicitly given internal goals and SSRL are compared. Fig. 6 shows the results obtained from the experiment using Q-learning, and Fig. 7 shows those from the experiment using SSRL. The task success rate is indicated in each figure. The red curves represent the trajectories for the team that reached the goal, and the gray curves represent the trajectories for the team that did not reach the goal. This shows that simple Q-learning achieves a single task. However, the Follower could not coordinate with the Leader agent after it had changed its goal because it could not discover the Leader agent's intentions. SSRL performs

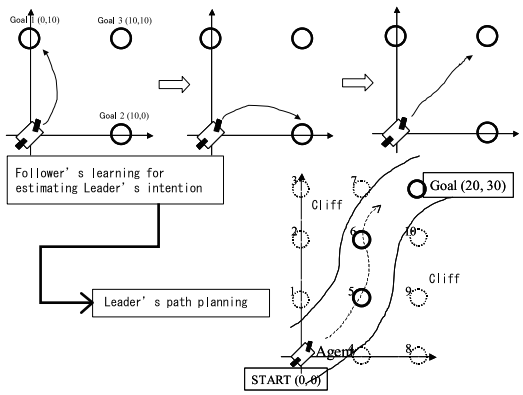

Fig. 5 Top: cooperative action is acquired by Follower, bottom: plan toward the goal is acquired by Leader

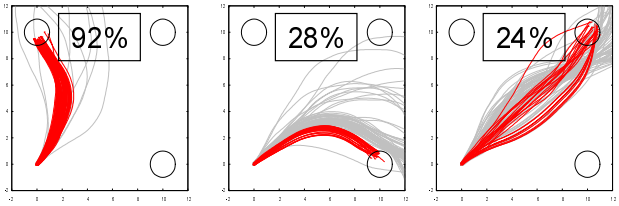

Fig. 6 Behaviors of truck at Follower's learning stage with single Q-table and internal goal-switching module without state predictors

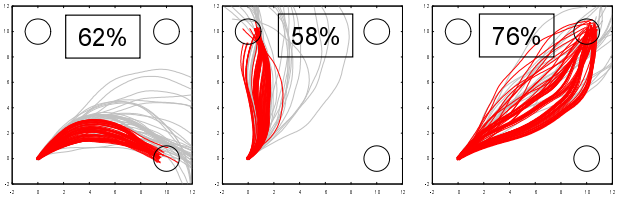

Fig. 7 Behaviors of truck at Follower's learning stage with SSRL

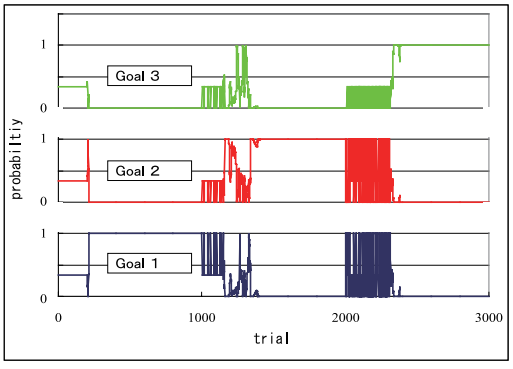

Fig. 8 Time course of probabilities where m-th internal goal is selected

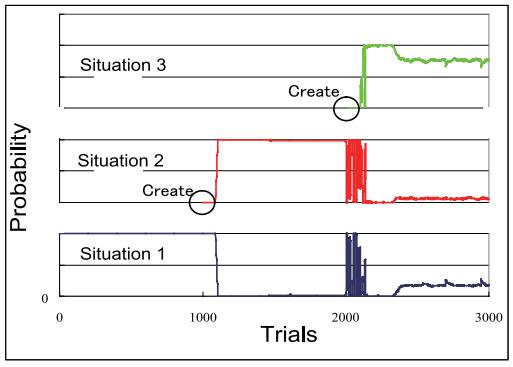

Fig. 9 Time course of probabilities that environment being faced is the i-th situation

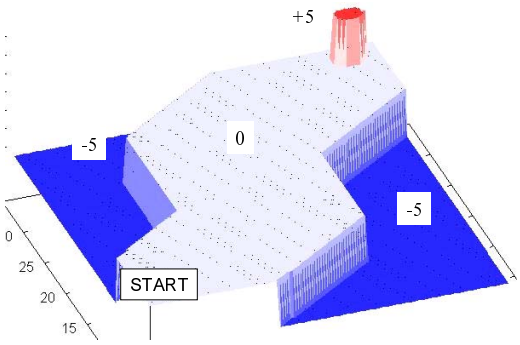

Fig. 10 Reward function for Leader agent for planning path

better when the Leader changes its intentions. Fig.9 shows that three predictors were generated that discover the Leader's intentions. Furthermore, Fig. 8 shows that appropriate internal goals were selected inside the Follower agent.

These results show that SSRL enabled the Follower to implicitly estimate the Leader's intention.

4.3 Experiment 2: Collaborative task

After the follower had acquired the ability to implicitly estimate the leader's intentions, the next experiment was carried out. The experimental environment is shown at the bottom of Fig. 5. The task required the agents to go through several checkpoints (sub-goals), and reach the final goal. The Follower in the next experiment exploited the SSRL acquired through Experiment 1, and the Leader explored and planed the path to the final goal. The Leader agent can chose the next sub-goal out of three check points that correspond to three goals in Experiment 1, i.e., "up", "upper right", and "right", from the current checkpoint as shown in Fig. 5. There are also two "cliffs" in this task environment. If the truck enters the cliffs, it can no longer move. The Leader learned the path to the final goal by using simple Q-Learning. The reward function for the Leader is shown in Fig. 10. Two kinds of Follower agents are compared in this experiment. The first has a single Q-learning architecture and a perfect internal goal switch. The second has SSRL.

Fig. 11 shows the results for the experiment using simple Q-learning. Fig. 12 shows the results for the experiment using SSRL. Fig. 13 shows the success rate representing the probability that the team will finally reach the final goal. The results reveal that the team whose Follower agent could not discriminate the Leader's intentions performed worse than the team whose Follower agent could distinguish

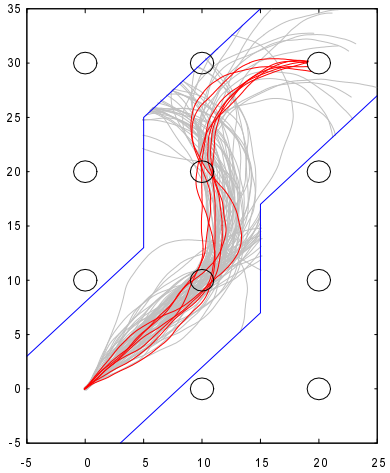

Fig. 11 Behaviors of truck at Leader's learning stage with single Q-table and internal goal switching module without Situation Recognizer

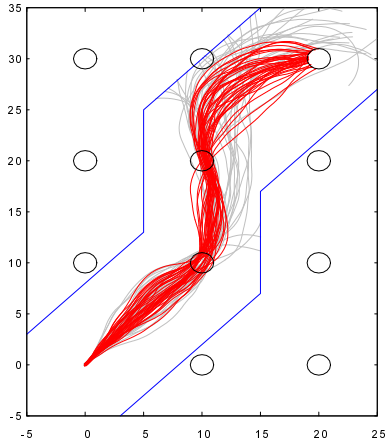

Fig. 12 Behaviors of truck at Leader's learning stage with SSRL

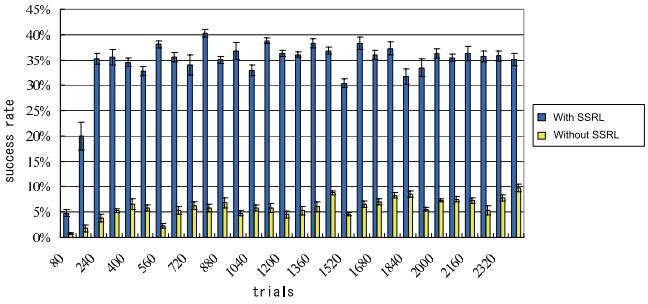

Fig. 13 Success rate for cooperative task

the Leader's intentions. Without such a distributed memory system like SSRL, the Follower would not be able to keep up with in the Leader's intentions. In addition to disadvantage, the poor performance of the Follower agent adversely affects the Leader's learning process. However, the Follower with SSRL could estimate the Leader's intentions and keep up with the Leader's plans although there was no explicit communication between the two agents. However, the success rate for the collaborative task saturated at about 40%. The reason for this is that the Follower notices changes in the Leader's intentions after these changes have sufficiently affected the state variables. The delay until the Follower becomes aware of the changes is sometimes critical, and the truck occasionally fell into the cliffs. To estimate the other's intentions without any explicit signs outside the state variables, the information has to be embedded in the state variables, which are the objectives of the team's control task. Our results suggest that it is not impossible to implicitly estimate the other's intentions, but it is important to have a communication channel whose variables are not related to the state variables, which are the objectives of the task.

5. CONCLUSION

We described a framework for implicitly estimating the other's intentions based on modular reinforcement learning. We applied the framework to a truck-pushing task by two agents as a concrete example. In the experiment, the Follower agent could perceive changes in the Leader's intentions and estimate his intentions without observing any explicit signs on any action outputs from the Leader. This demonstrated that autonomous agents can cooperatively achieve a task without any explicit communication. Self-enclosed autonomous agents can indirectly perceive the other's changes in intentions from changes in their surrounding environment. However, this framework for implicit estimates does not always work well. If the system does not satisfy the assumptions made in Section 3, the framework is not guaranteed to work. Moreover, the Leader's policies are fixed when the Follower agent is learning its policies, predictors, and network connections in our framework. Therefore, simultaneous multi-agent reinforcement learning is not taken into consideration. We intend to take these into account in future work.

REFERENCES

[1] K. Murphy. Learning switching kalman-filter models. Compaq Cambridge Research Lab Tech Report, pp. 98–10, 1998.

[2] R. Sutton and A.G. Barto. Reinforcement Learning : An Introduction. The MIT Press, 1998.

[3] Y. Takahashi, et al. Modular learning syatem and scheduling for behavior acquisition in multi-agent environment. In RoboCup 2004 Symposium papers and team description papers, CD-ROM, 2004.

[4] Shinya Takamuku, Yasutake Takahashi, and Minoru Asada. Lexicon acquisition based on object-oriented behavior learning. Advanced Robotics, Vol. 20, No. 10, pp. 1127–1145, 2006.

[5] J. Tani and S. Nolfi. Learning to perceive the world as articulated: an approach for hierarchical learning in sensory-motor systems. Neural Networks, Vol. 12, pp. 1131–1141, 1999.

[6] T. Taniguchi and T. Sawaragi. Incrimental acquisition of multiple nonlinear forward models based on differentiation process of schema model. (submitting).

[7] T. Taniguchi and T. Sawaragi. Symbol emergence by combining a reinforcement learning schema model with asymmetric synaptic plasticity. In 5th International Conference on Development and Learning, 2006.

[8] T. Taniguchi and T. Sawaragi. Incremental acquisition of behaviors and signs based on reinforcement learning schema model and stdp. Advanced Robotics (in press), 2007.

[9] C. Watkins and P. Dayan. Technical note: Q-learning. Machine Learning, Vol. 8, pp. 279–292, 1992.

[10] D.M. Wolpert, K. Doya, and M. Kawato. A unifying comuputational framework for motor control and social interaction. Phil Trans R Soc Lond B, Vol. 358, pp. 593–602, 2003.

[11] D.M. Wolpert and M. Kawato. Multiple paired forward and inverse models for motor control. Neural Networks, Vol. 11, pp. 1317–1329, 1998.