Implicit estimation of other’s intention based on modular reinforcement learning

Tadahiro Taniguchi$^{1,3}$ Kenji Ogawa$^{2}$ and Tetsuo Sawaragi$^{2}$

July 26, 2008

1 Introduction

When we try to achieve collaborative task: e.g., playing football and carrying large tables, we have to share a goal and a way of achieving the goal. Although people achieve such tasks, to achieve such cooperation is not so easy in the context of computational multi-agent learning system because the participating agents cannot observe other person’s intention directly. We cannot know directly what other participants intend to do and how they intend to achieve that. Therefore, we have to notice the partners’ intention by utilizing other hints or information. In other words, we have to estimate other’s intention to achieve collaborative tasks.

Especially, in multi-agent reinforcement learning tasks, such a problem in which other’s intention is unobservable harms the learning process fatally. When a participating agent of a collaborative task changes its intention and switches or modifies its controller, the system dynamics for each agent will inevitably changes. If other agents learn based on simple reinforcement learning architecture, they cannot catch up the changes in the task environment because most reinforcement learning architectures assume that the environmental dynamics is fixed. To overcome the problem, each agent must not have only a simple reinforcement learning architecture but also some additional capability, which solves the problem. We take a capability of “estimation of other’s intention” as an example of such kinds of capabilities.

$^{*}$Currently at Matsushita Electoronic Industrial Co..
We, human beings, can perform several kinds of collaborative tasks. This means that we have some computational skills, which enables us to estimate others’ intention to some extent even if we cannot observe others’ intention directly.

The computational model for implicit communication is described in this chapter based on a framework of modular reinforcement learning. The computational model is called situation-sensitive reinforcement learning (SSRL), which is a type of modular reinforcement learning architecture. 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 skill, estimation of others’ intention, seems to be a social skill. However, a human adaptability, which we believe to be equipped with in order to deal with physically dynamic environment, enables an agent to deal with such social dynamic environment, including intentional changes of collaborators. To clarifyify the computational relationship between the two skills is also a purpose of this chapter.

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 Communication and Estimation of other’s intention

To communicate one’s intention to the other person is to make the person estimate one’s intention. Therefore, communication and estimation of other’s intention are different aspects of the same phenomenon. The implicit estimation is key idea to supplement the classical communication model, i.e., Shannon-Weaver communication model, and to approach toward emergence of communication. We describe the background in this section. Additionally, an abstract mechanism of the implicit estimation is described based on the notion of multiple internal models.
2.1 Communication models

Classically, Shannon formulated “communication” mathematically [?]. In the Shannon’s communication model, a sender’s messages encapsulated in signals or signs are carried through an information channel to a receiver. An encoder owned by the sender encodes the message to the signal by referencing to its code table. When a receiver catches the signal, the receiver’s decoder decodes the signal back to a message by referencing its code table. After that, the receiver understands the leader’s intention and determines what to do. The general communication system described by Shannon is is shown in Fig. 1 schematically.

In contrast to Shannon, Peirce who started “semiotics” insisted that the basis of communication is symbol, and he defined as the symbol a triadic relationship of “sign”, “object”, and “interpretant”[?]. “Sign” is a signal that represents something to an interpreter, “object” is something that is represented by the sign, and “interpretant” is something that relates the sign and the object. In other words, “interpretant” is a mediator of “sign” and “object”. The words, “sign” and “object”, are easy for most people to understand. However, “interpretant” must be difficult to understand. “Interpretant” is sometimes a concept an interpreter comes up, an action the interpreter takes, or culture in which they consider the sign and object are related. The important point of Peirce’s semiotics is that the relationship between “sign” and “object” is not fixed. The relationship can be dynamically changing. The relationship is simply depends on the “interpretant”. The dynamic process through which a sign represents an object mediated by interpretant is called “semiosis”. Peirce’s semiotics is thoroughly constructed from the viewpoint of an interpreter. In the framework of Peirce’s semiotics
the third element “interpretant” plays an essential part in communication. In Shannon’s communication model, a shared code table is required in premise. However, autonomous agent cannot observe other agents’ internal goals or code table. In contrast, Peirce’s semiosis does not require such a premise. Semiosis is a phenomenon that emerges inside of an autonomous agent. The participants of a communication must create meaning from incoming signs based on their physical and social experience. Such an individual learning process is considered to supplement symbolic communication. However, the semiosis requires autonomous agents sufficient adaptability and capability to create meanings from superficial meaningless signs.

In a human collaborative task, a human participant will become able to distinguish several situations, which are modified by the other’s changing intentions. In such a case, what a kind of policy the participant should take in each situation is not clear beforehand. However, if the team continues to collaborate through trials and errors, some kind of shared rules will be formed as a kind of habits in the team, and the follower becomes able to perform adequately by referring the situation and the habit. This process is corresponding to “semiosis” in Peirce’s semiotics. Here, “sign”, “object” and “interpretant” are corresponding to “situation”, “the leader’s intention” and “acquired rule” or “the follower’s action” respectively.

What is an important point in this scenario is that the “situation” has no meaning before the follower becomes distinguishing the situation, perform adequately and the tacit rule are established between the two agents.

In this chapter, we describe a candidate of computational communication models which are based on Peirce’s semiosis.
2.2 Estimation of other’s intention

Roughly speaking, we assume there are two ways how we estimate other’s intention. Here, we explain the difference between the two ways of estimation of other’s intention.

For illustrative purpose, now, we assume that there is a leader in an organization who makes decision. A leader makes decisions to direct the team, and the followers play their roles based on the decision.

In such a case, the leader communicates his/her intention to the following members and members in the organization have to estimate a leading agent’s intention to cope with the cooperative tasks. The communication and the estimation of other’s intention are different aspects of the same phenomenon, as we described beforehand. How can the members estimate the leader’s intention? This is the problem.

Here, we take two kinds of estimation of other’s intention into consideration. One is “explicit estimation”, and the other is “implicit estimation”.

2.2.1 Explicit estimation of other’s intention

One solution is to communicating one’s intention to another person is to express one’s intention directly with predefined signals, e.g. by pointing to the goal and by commanding the other to act. The method of communication requires a shared symbolic system as a basic premise. The symbolic system is often called code table. If the symbolic system used in this communication must be completely shared by the participants of the cooperative-task environment, a participant who receives a message notices what the person transmitting the message want to do completely. The receiver of the message can estimate the sender’s intentions based on the externalized signs. We call this process the “explicit estimation” because what the leader intend is explicitly expressed as externalized signals. In this communication model, the both agents have to share a predefined code table before the tasks. In the explicit estimation model, the accuracy of the communication is measured by the coincidence between the transmitted message and the receiver’s interpreting sender’s message, which is obtained by decoding the incoming signal utilizing the shared code table. Figure 3 shows the process of estimating other’s intention in a collaborative task, schematically. A leader and a follower carry a truck collaboratively. How can the follower estimate the leader’s goal using explicit estimation when the leader changes his goal?
First of all, the leader agent changes his goal. In explicit estimation scheme

This seems natural framework of communication. After Shannon formulating “communication” mathematically, many sociologists and computer scientists describes “communication” as above. However, the communication model based on explicit estimation of other’s intention has two shortcomings. One is that how can the two agents share the code table is unknown. If we consider the two agents to be autonomous, each agent cannot observe other agent’s internal goals and code table. Therefore, each agent cannot utilize “teacher signal” as a feedback of its interpretation to upgrade its code table. The other is that the leader agent has to display his intention whenever he changes his goal. They are the problems of explicit estimation of other’s intention.

In contrast, when we think over what we do in collaborative tasks, we find that we do not always send verbal messages representing our intention to our collaborators. We sometimes execute a collaborative task without saying anything. In this case, the leader’s intention is transmitted to the follower not by the explicit linguistic sign but through the shared environmental dynamics implicitly. Explicit estimation of other’s intention is not an only way of communication. To complement or to back up the explicit estimation, implicit estimation is necessary.
2.2.2 Implicit estimation of other’s intention

People occasionally undertake collaborative tasks without saying anything. Even if a leader says nothing to his organization’s members, they can often perform the task by estimating the leader’s intentions based on their observation of their sensations. We call such estimation process “implicit estimation” of other’s intention. However, if there were no pathways through which information of the leader’s intention go to the followers, the followers could never estimate the leader’s intention. One of the reasons why the followers can estimate the leader’s intention is that the action and sensation of the followers are causally related to the leader’s intentions.

In other words, the sensations a participating agent can observe after he/she outputs actions are effected by the leader’s way of acting and the other agents’ way of acting. Therefore, the subjective environmental dynamics for a participating agent are causally affected by the leader’s intention because the other agents are assumed to behave based on the leader’s intention.

We assume all the members cannot observe any information except for their own sensory-motor information. However, they can estimate the leader’s intentions. We call this process “implicit estimation”. Implicit estimation is achieved by watching how the agent’s sensation changes. In control tasks, an agent usually observes state variables from its sensation.

In what follows, we assume that an agent obtains state variables, e.g., position, velocity, and angle. The state variables are usually considered to be the objectives to be controlled in many control tasks. However, in implicit
communication, the state variables also become an information media of other agent’s intention.

Figure 4 shows the process of implicit estimation of other’s intention, schematically. First, the leader changes his goal. When the leader’s goal changed, his controller, which produces his behavior, is switched. That, of course, affects the physical dynamics of the dynamical system shared between the leader and the follower. If a participating agent has a state predictor, he will become aware of the qualitative change in the shared dynamics because his prediction of the state value becomes collapsed. If the physical dynamics is stable, he can predict his state variables consistently. If the follower agent notices the change in the physical dynamics subjectively, the follower can notice the change in the leader’s intention based on the causal relationship between the leader’s intention and his facing dynamical system.

Therefore, the capability to predict the state variables seems to be required by not only physical skills but also social skills. This scenario suggests the learning process of physical skills to control the target system and how to communicate with the partner agent might be closely connected in such cooperative tasks.

3 Multiple internal models

Our computational model of implicit estimation of other’s intention is based on modular reinforcement learning architecture including multiple internal models. To achieve implicit estimation of other’s intention described in the previous section, an agent must have a certain learning architecture including state predictors. We focus on multiple internal model as a neural architecture which realizes such an adaptive capability.

3.1 Multiple internal models and social adaptability

The relationships between human brain’s social capability and physical capability are commanding interest. From the viewpoint of computational neuroscience, Wolpert et al. [?] suggested that MOSAIC, which is a modular learning architecture representing a 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. We regard this is a candidate of a brain function which connect human physical capa-
bility and social capability. An internal model is a learning architecture that predicts the state transition of the environment or other target system. It is believed that a person can operate his/her body and his/her grasping tool by utilizing obtained internal model[]. The internal model is acquired in the cerebellum through interactions. The learning system of internal models is considered to be a kind of schema that assimilate the exterior dynamics and accommodate the internal memory system, i.e., internal model. If a person encounters various kinds of environment and/or tools, which have different dynamical properties, human brain should differentiate them and acquire several internal models. However, segmentation of the dynamics is not given a priori. Therefore, a learning architecture representing multiple internal models should generate and learn internal models, and recognize changes of physical dynamics in its facing environment at the same time. To describe such a learning system, several computational models have been proposed, e.g., MPFIM [], the mixture of RNNs[], RNNPB[] and the schema model[]. Most of them are comprised of several learning predictors. The learning architecture switches the predictors and accommodates them through interactions with environment. Such a learning architecture is often called modular learning system. Among them, the RNNPB is not a modular learning system. Tani insisted internal models should be obtained in a single neural network in a distributed way. In most modular learning architecture, Bayesian rule is used to calculate the posterior probability in which a current predictor is selected. In contrast, the schema model[] is a modular learning architecture that does not use Bayesian rule but hypothesis testing theory. At the moment, a multiple internal models are usually considered to be a learning system for an autonomous system to cope with physically dynamic environment. Meanwhile, Wolpert et al. addressed a hypothesis that a person utilizes multiple internal models to estimate the other’s intention from the observation of other’s movement. Although this internal models described in the hypothesis seems to add a little different feature to the original definition of internal model, it is interesting that the hypothesis tries to connect the neural architectures for physical adaptability and social adaptability. Doya et al.[?] proposed a modular learning architecture which enables robots to estimate other’s intention and to communicate each other in a reinforcement learning task.

In addition, when a person performs a collaborative task with others, one can notice changes in other agent’s intention by recognizing the transition of his/her facing dynamical system without any direct observation of other
agents’ movement. This means multiple internal models enable an agent to notice changes in the other agents’ intention. This usage of multiple internal models do not require any additional features to the original definition of the multiple internal models.

3.2 Implicit estimation of other’s intention based on multiple internal models

“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 chapter. In the framework of reinforcement learning, an agent’s goal is represented by a reward function. Therefore, an agent who has several intentions has several internal goals, i.e. several internal reward functions, $G^m$. If an 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.

We assume that the collaborative task consists of two agents in what follows. The system is described as

$$y = f(x, u_1, u_2) + n,$$

$$= f(x, u_1, u_2(x)) + n, \text{and}$$

$$= F^m(x, u_1) + n. \text{(3)}$$

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. 3 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 who have multiple internal models can detect changes in the other agent’s intentions by detecting changes in subjective environmental dynamics, $F$. The computational process is completely equal to the process through which an agent detects changes of originally physical dynamics.
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. By utilizing multiple internal models, an agent is expected to differentiate situations and execute adequate actions. In the next section, we describe a concrete modular reinforcement learning architecture named Situation-Sensitive Reinforcement Learning (SSRL).

4 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 [? , ?] and hierarchical learning architectures [?, ?] have been proposed to describe this kind of 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 without any teacher signals from the partner.

4.1 Discrimination of intentions based on changes in dynamics

Fig. 5 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.

\[
e_i^j = ||y_t - F^j(x_t, u_t)||^2, \quad \text{Eq. (4)}
\]

\[
P(j|\bar{e}_t^j) = \exp\left(-\frac{\bar{e}_t^j}{2\sigma^2}\right) / \sum_{k=1}^{p} \exp\left(-\frac{\bar{e}_t^k}{2\sigma^2}\right), \quad \text{Eq. (5)}
\]

\[
j^* = \arg \min_j P(j|\bar{e}_t^j), \quad \text{Eq. (6)}
\]
where $\bar{e}_j^t$ is the temporal average of the prediction error, $\epsilon_j^t$, of the $j$-th state predictor, $F^j$. If averaged error $\bar{e}_j^t$ has a normal distribution and the system dynamics is $F^j$, the posterior probability, $P(j|\bar{e}_j^t)$, 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 [?].

We model the state predictors by using locally linear predictors, and we don’t estimate the standard deviation $\sigma$. The updating rule are switched based on hypothesis testing.

Case 1: $\bar{e}_j^t < \delta_l$

In this case, the learning system considers that incoming sample data are normal samples for the existing predictors, decides the current situation $j^*$, and update the corresponding function $F^{j^*}$ by using assimilated samples.

Case 2: $\bar{e}_j^t > \delta_u$

In this case, the learning system considers that incoming sample data are outliers for the existing predictors, and prepare a new function $F^{p+1}$. It decides the current situation $j^{p+1}$. However, the new predictor is considered as an exceptional state predictor until $\bar{e}_j^{p+1} < \delta_l$. If the predictor’s averaged error reaches under $\delta_l$, the function $F^{p+1}$ is taken into a list of existing predictors, and $p \leftarrow p + 1$. 

Figure 5: Situation-Sensitive Reinforcement Learning architecture
Case 3: $\delta_l < \bar{e}_j^t < \delta_u$

In this case, the system takes no account of the incoming sample.

- $\bar{e}_j^t < \delta_l$
  
  In this case, the learning system considers that incoming sample data are normal samples for the existing predictors. It decides the current situation $j^*$, and update the corresponding function $F_j^*$ using assimilated samples.

- $\bar{e}_j^t > \delta_u$
  
  In this case, the learning system considers that incoming sample data are outliers for the existing predictors, and prepare a new function $F_{p+1}$. It decides the current situation $j^{p+1}$. However, the new predictor is considered as an exceptional state predictor until $\bar{e}_{p+1} < \delta_l$. If the predictor’s averaged error reaches under $\delta_l$, the function $F_{p+1}$ is taken into a list of existing predictors, and $p \leftarrow p + 1$.

- $\delta_l < \bar{e}_j^t < \delta_u$
  
  In this case, the system takes no account of the incoming sample.

This is an intermediate method for the MOSAIC model [?, ?], which is based on the Bayesian framework, and the schema model [?], 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.

4.2 Reinforcement Learning

Each policy corresponding to a goal is acquired by using reinforcement learning [?]. SSRL uses Q-Learning [?] 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 $S$ as a set of state variables and $A$ 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

\[
Q \leftarrow Q(s, a) + \alpha (r + \gamma V(s') - Q(s, a)),
\]

\[
V(s') = \max_{a' \in A} Q(s', a'), \text{and}
\]

\[
u(s) = \arg\max_{a' \in A} Q(s, a'),
\]

where \( s \in S \) is a state variable, \( a \in A \) is a motor output, \( r(s, a) \) is a reward, and \( s' \) is a state variable at the next time step. In these equations, \( \alpha \) is the learning rate and \( \gamma \) 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.

\[
p(a|s) = \exp(\beta Q(s, a))/\sum_{a'}(\beta Q(s, a'))
\]

4.3 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 implicitly. 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. We employ Boltzmann selection for internal goal switch. The rule to select the internal goals are described as

\[
p(m|j) = \frac{\exp(Bw_{jm})}{\sum_{i=1}^{q} \exp(Bw_{ji})},
\]

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, \( R^m_t \), during a certain period during the \( t \)-th trial, i.e.,

\[
w_{jm} = \nu R_{jt}^{jm} + (1 - \nu)w_{jm}
\]

Here, \( \nu \) is the learning rate of the internal goal switching module. Eq. 11 shows that connection \( w_{jm} \) becomes strong if internal goal \( G^m \) is more easy to accomplish when the situation is \( F^j \). Eq. 10 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. 6. 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.

4.4 Mathematical basis for internal goal switching module

This section provides the mathematical basis for the learning rule of the implicit communication. First, the Bellman equation for the \( i \)-th \((i = 1, 2)\) agent of a system involving two agents are described as \( ^1 \).

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

---

\(^1\)In this section, we have assumed \( i \neq j \) without making any remarks.
where $G^\lambda$ is a reward function for the $\lambda$-th goal, $u_i$ is the $i$-th agent’s motor output, and $x'$ is the $x$ in the next step. $G^\lambda$ 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 $u_i^\lambda$ as the $i$-th agent’s policy that maximizes the $j$-th agent’s maximized value function whose goal is $G^\lambda$.

$$u_i^\lambda = \arg\max_{u_i \in U_i} \max_{u_j \in U_j} \sum_{x'} P(x'|x, u_i, u_j) \times [G^\lambda(x, x') + \gamma V_i^\lambda(x')] \text{and (13)}$$

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, $u_j^\lambda$, and

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

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

$$V_i^{\lambda|\nu} = V_i^{\nu|\nu} \geq V_i^{\nu|\lambda}. \quad (15)$$

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 $u_j^\lambda$. 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.11.

5 Experiment

We evaluate SSRL in this section. To fulfill all the assumptions made in Section 4 completely is difficult in a realistic task environment. The task described in this section roughly satisfies the assumptions, A’2 and A’3.

5.1 Conditions

We applied the proposed method to the truck-pushing task shown in Fig. 7. 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, the agents cannot “explicitly” communicate their 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, $\theta_L, \theta_F$, and the wheel’s rotating speed, $\omega_L, \omega_F$. Here, the final motor output to the truck, $\theta, \omega$, is defined as

$$\theta = K_{\theta}(\theta_L + \theta_F)$$
\hspace{1cm}(16)

$$\omega = K_{\omega}(\omega_L + \omega_F),$$
\hspace{1cm}(17)

where $K_{\theta}$ and $K_{\omega}$ are the gain parameters of the truck. $K_{\theta}$ and $K_{\omega}$ 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 $10 \times 8$ 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, $\Omega$, was observed by the Follower agent. $\Omega, \theta$, and $\omega$ have a relationship of

$$\Omega \propto \omega \tan \theta.$$  
\hspace{1cm}(18)

The agents can carry the truck to a certain goal by cooperatively controlling $\Omega$. The main state variables are shown in Fig. 8. Internal reward function $G^m$ is defined as

$$G^m(x) = \begin{cases} 5 & \text{if } ||C - Goal_m|| < 1 \\ \kappa(1 - ||C - Goal_m||) & \text{otherwise} \end{cases},$$
\hspace{1cm}(19)

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

5.2 Experiment 1: implicit estimation of other’s intention

We first 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. 9, 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. 10 and 11. Simple Q-learning with explicitly given internal goals and SSRL are compared. Fig. 10 shows the results obtained from the experiment using Q-learning, and Fig. 11 shows those from
Figure 9: Top: cooperative action is acquired by Follower, bottom: plan toward the goal is acquired by Leader

Figure 10: Behaviors of truck at Follower’s learning stage with single $Q$-table and internal goal-switching module without state predictors

Figure 11: Behaviors of truck at Follower’s learning stage with SSRL
Figure 12: Time course of probabilities where $m$-th internal goal is selected

Figure 13: Time course of probabilities that environment being faced is the $i$-th situation
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 better when the Leader changes its intentions. Fig. 13 shows that three predictors were generated that discover the Leader’s intentions. Furthermore, Fig. 12 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.

5.3 Experiment 2: sequential 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. 9. 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 planned the path to the final goal. The Leader agent can choose the next sub-goal out of three checkpoints that correspond to three goals in Experiment 1, i.e., “up”, “upper right”, and “right”, from the current checkpoint as shown in Fig. 9. 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 a simple Q-learning. The reward function for the Leader is shown in Fig. 15. 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. 16 shows the results for the experiment using simple Q-learning. Fig. 17 shows the results for the experiment using SSRL. Fig. 18 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 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
Figure 14: Reward function for Follower’s internal goals

Figure 15: Reward function for Leader agent for planning path
Figure 16: Behaviors of truck at Leader’s learning stage with single $Q$-table and internal goal switching module without Situation Recognizer

Figure 17: Behaviors of truck at Leader’s learning stage with SSRL
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, e.g., voice, color sign, or marker. This must be the reason why we use explicit sign in collaborative tasks. As we mentioned, the “implicit estimation” must back up and complement “explicit estimation”. “Explicit estimation” must be faster and better than “implicit estimation” as far as a code table was shared in a team. However, this does not mean “explicit estimation” is superior to “implicit estimation”. They are complementary architectures.
6 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. It is revealed that multiple internal models help an autonomous agent to achieve collaborative task.

In the context of artificial intelligence, “symbol grounding problem” is considered as an important problem. The problem deals with how robots and people can relate their symbolic system to their physical and embodied experiences. The symbolic system mentioned here is also used in communication, usually. Takamuku et al. presented a system for lexicon acquisition through behavior learning which is based on a modified multi-module reinforcement. The robot in their work is able to automatically associate words to objects with various visual features based on similarities in features of dynamics\. 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)\. 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 an important role in human social communication. They also utilize modular learning architecture to describe the process of symbol organization. However, they treat bottom-up organization of “explicit symbols”, which is assumed to be used explicit communication.

In many researches, “symbolic communication” means exchanging discrete signals. However, the essential point of symbolic communication is not such an externalized signs, but an adaptive formation of “interpretant” from the viewpoint of Peirce’s semiotics. Therefore, we focus on the implicit communication and its bottom-up process of organization.

However, the system we treated in this chapter is constrained to some extent. This framework for implicit estimates does not always work well. If the system does not satisfy the assumptions made in Section 4, the frame-
work is not guaranteed to work. The Leader’s policies are fixed when the
Follower agent is learning its policies, predictors, and network connections
in our framework. The model described in this chapter may not work in the
simultaneous multi-agent reinforcement learning environment. We intend to
take these into account in future work.

References

[1] Charles Hartshorne, Paul Weiss, and Arthur W. Burks, editors. Col-
lected Papers of Charles Sanders Peirce. Thoemmes Pr, 4 1997.


or acquisition in multi-agent environment. In RoboCup 2004 Symposium

[6] Shinya Takamuku, Yasutake Takahashi, and Minoru Asada. Lexicon ac-
quision based on object-oriented behavior learning. Advanced Robotics,

sented multiple behavior schemata in a mirror system: reviews of robots

[8] J. Tani and S. Nolfi. Learning to perceive the world as articulated:
an approach for hierarchical learning in sensory-motor systems. Neural

[9] T. Taniguchi and T. Sawaragi. Incremental acquisition of multiple non-
linear forward models based on differentiation process of schema model.
(submitting).
[10] T. Taniguchi and T. Sawaragi. Symbol emergence by combining a rein-
forcement learning schema model with asymmetric synaptic plasticity.
In 5th International Conference on Development and Learning, 2006.

signs based on reinforcement learning schema model and stdp. Advanced


framework for motor control and social interaction. Phil Trans R Soc

models for motor control. Neural Networks, Vol. 11, pp. 1317–1329,
1998.

[15] 杉本徳和ほか. ダイナミクスの線形性に基づいて状態空間を分割する階