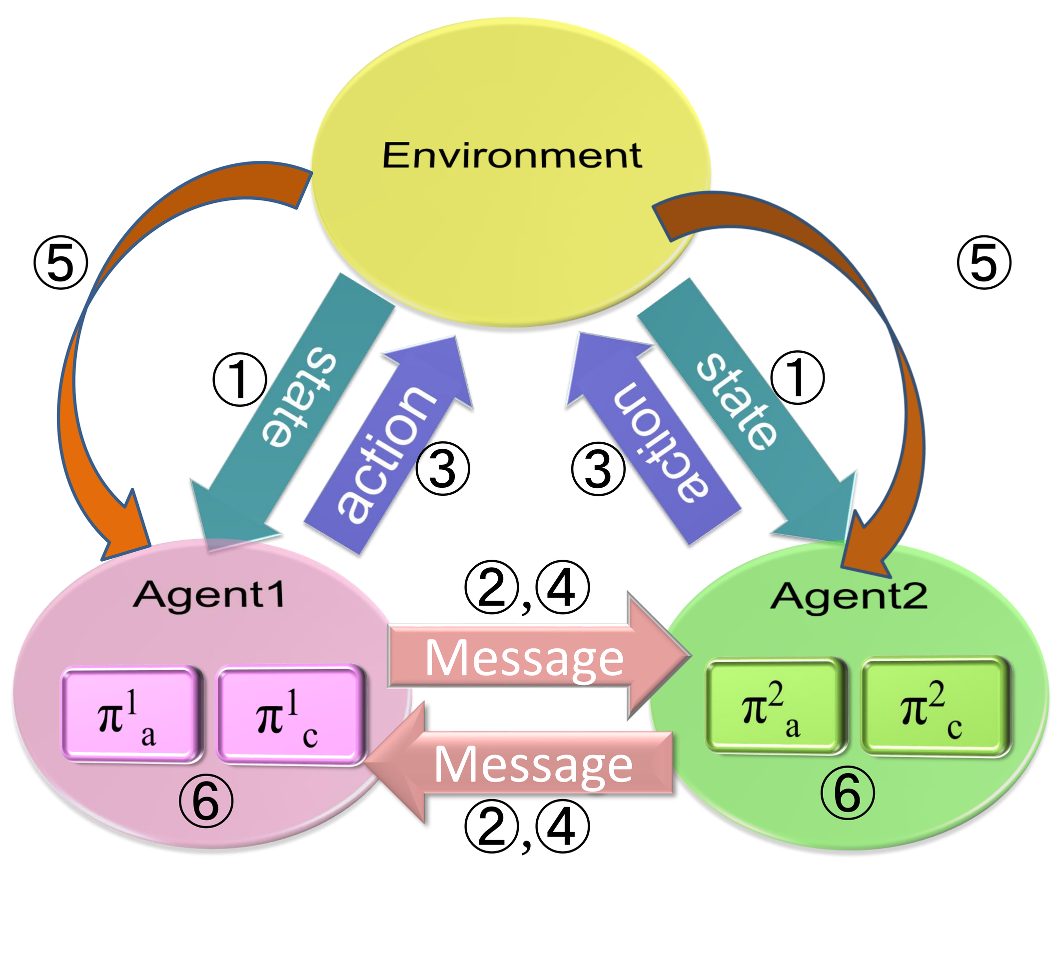
Improvement of the Performance Using Received Messages on Learning of Communication Codes

Tatsuya Kasai Hayato Kobayashi Ayumi Shinohara

Graduate School of Information Sciences, Tohoku University, Japan

Introduction

In Multi-Agent Reinforcement Learning (MARL), each agent learns a cooperative policy $\pi : S \to A$, where S and A are a set of states and a set of actions, respectively. If we utilize communication to facilitate multi-agent coordination, we must construct communication codes so that agents can communicate with each other. However, it is a hard task since we usually do not know workable communication codes and/or information on unknown problems. **We focus on a method that allows agents to learn communication codes autonomously.**



① observe a state $s \in S$
② receive a message $m \in M$
③ perform the action $a = \pi_a(s, m)$
④ send the message $m' = \pi_c(s, m)$
⑤ observe a reward $r \in \mathfrak{R}$
⑥ update π_c and π_a based on the reward r

Figure 1 : One-step dynamics of SL/SLM with two agents

■Extension of learning method

Previous work

Signal Learning (SL) [kasai08] allows agents to learn communication codes autonomously in MARL framework, where M is a set of messages whose meanings are not predetermined explicitly.

In SL, agents can learn **two policies** as follows, concurrently.

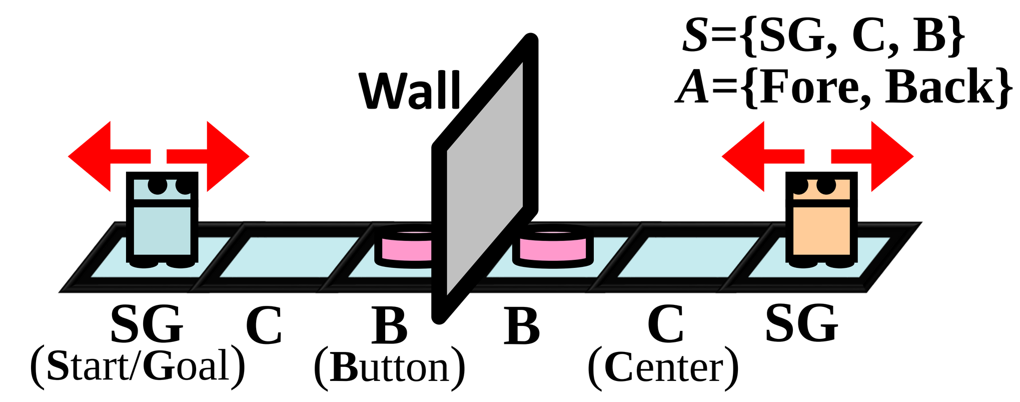
Communication policy
$\pi_c : S \to M$ → Extension → $\pi_c : S \times M \to M$

Action policy
$\pi_a : S \times M \to A$ $\pi_a : S \times M \to A$

This work

Our extension is just the change of π_c from $\pi_c : S \to M$ to $\pi_c : S \times M \to M$.

We call this method ***SL with Messages*** (SLM).



In order to activate the goal, both agents must occupy their B states at **the same time**.

Each agent can neither know the state of the other agent by the wall nor remember whether the goal has been activated since the agent is oblivious.

Figure 2 : Example problem

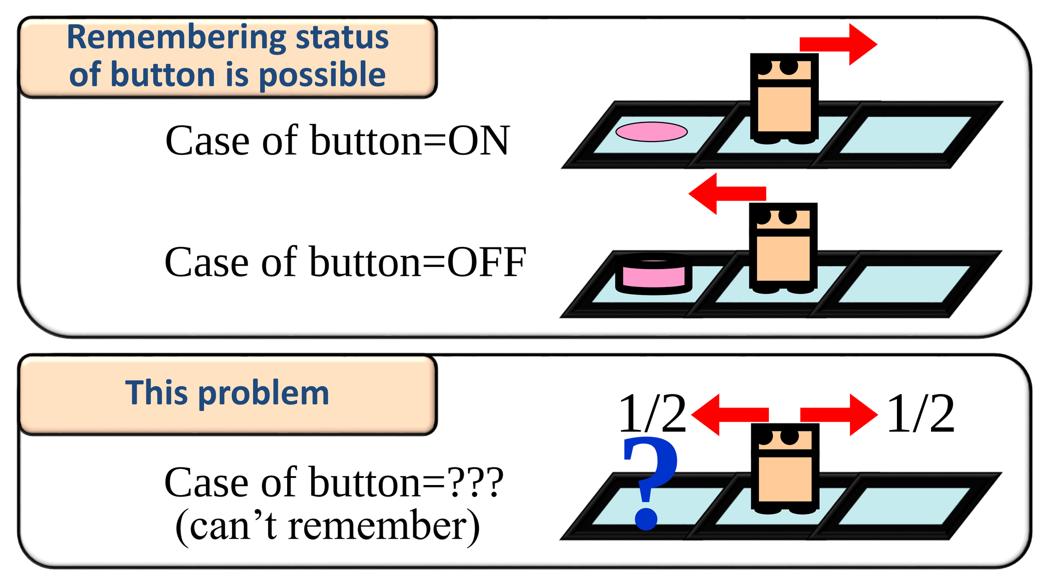
■Comparisons of SL and SLM

Example problem

The goal of the problem is that both agents, starting from their own SG states, go back to the SG states after ***activation*** (Figure 2, 3).

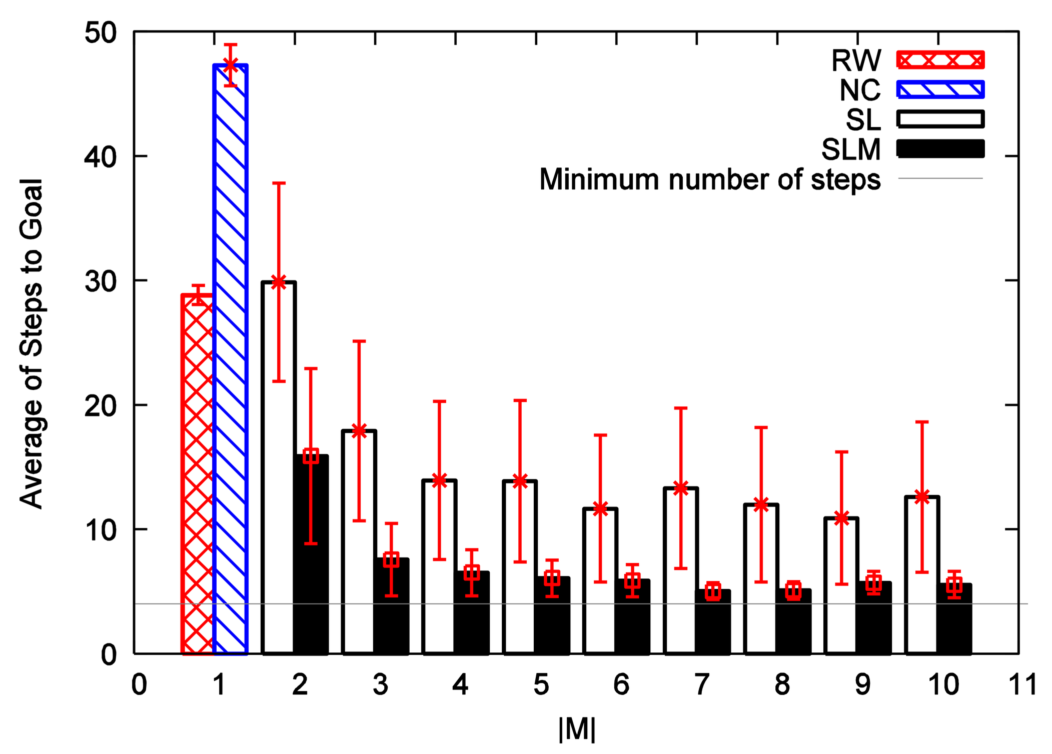
Experiments

We carried out experiments for comparing SL with SLM, where $|M|$ is varied **from 2 to 10**.



In this problem, the naive MARL framework and SL have no deterministic optimal policies so that each agent always can achieve the goal with minimum steps (4 steps), since **each agent can not remember (observe) status of button (ON or OFF).**

Figure 3 : Optimal policy with status of button



・RW : Random Walk
・NC : No Communication
・SL : Signal Learning
・SLM : SL with Messages

When $|M|$=1, since SL = SLM, we identify them as **No Communication** (NC). To verify the difficulty of our problem, we added the result of **Random Walk** (RW), which selects one action randomly in each time step.

We estimated the average number of steps to reach the goal in the last 100 episodes in 10,000 episodes in one trial.

Figure 4 : Comparisons of RW, NC, SL and SLM

■Results and Discussion

Discussion

By comparing NC with SL, SL is clearly better than NC (Figure 4). This shows that some beneficial meaning emerges in messages in M through the learning processes in SL. In SL, $\pi_c : S \to M$ probably allows each agent to include its own state in a message.

By comparing SL with SLM, SLM is **clearly better and more robust** than SL (Figure 4, 5). This means that SLM can allow each agent to include much more information in a message than SL.

By using an (deterministic) optimal policy, both agents reach the goal with the minimum number of steps. However, agents must remember the status of button for acquiring the deterministic optimal policies. In SLM, the messages should contain the information of status of button.

Table 1 shows the percentage of the successful trials in all 100 trials. As shown in the Table 1, SLM has the ability to acquire an deterministic optimal policy. Actually, SLM can allow the agent to get a deterministic optimal policy. Table 2 shows a simplest example of the acquired optimal policies ($|M|$=2) in SLM.

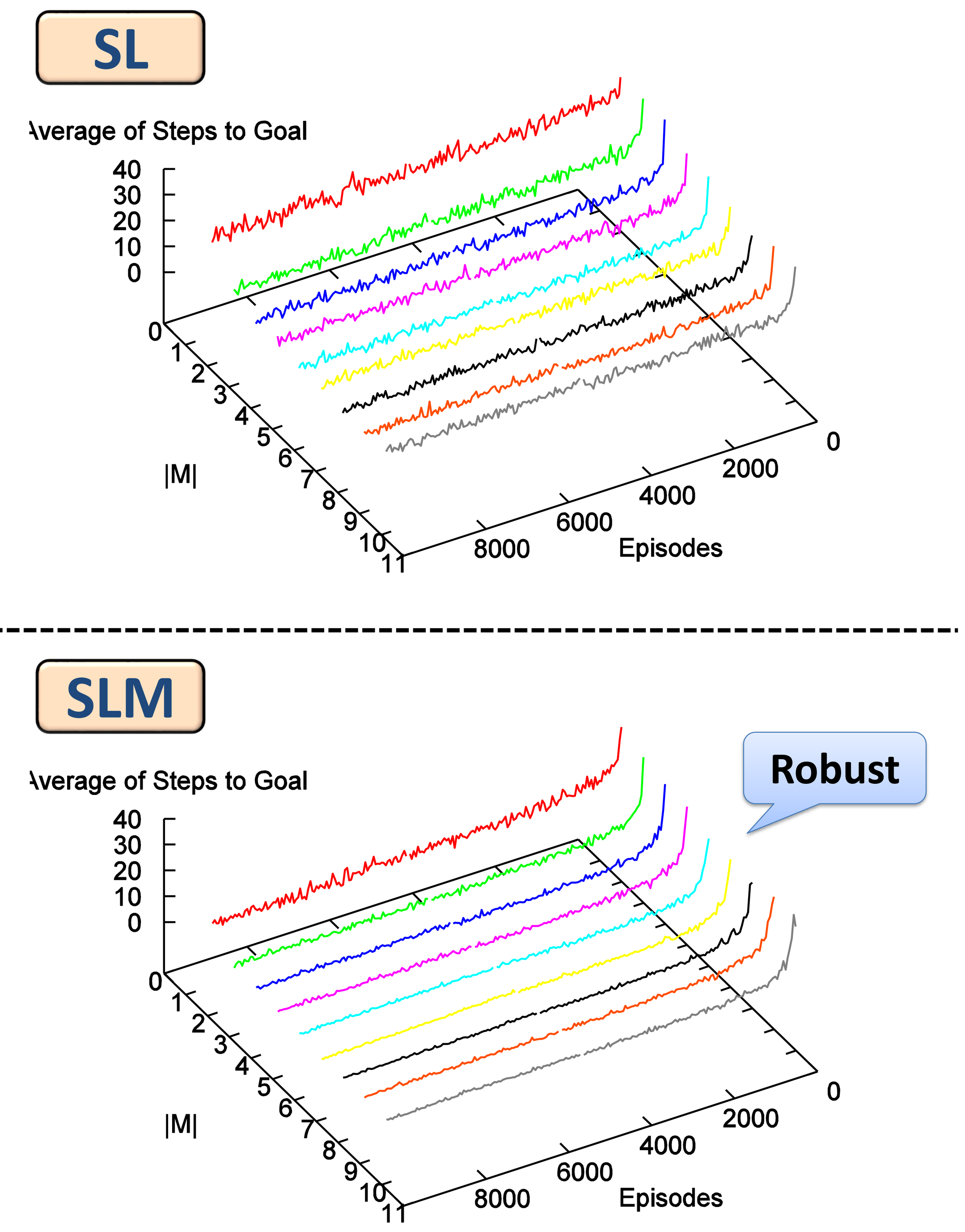


Figure 5 : Learning curves of SL/SLM

Table 1

|M|	2	3	4	5	6	7	8	9	10
SL	0	1	1	0	0	0	0	0	2
SLM	31	34	42	47	41	45	60	54	48

We say a trial is successful if both agents reach the goal in minimum number of steps, which is 4 steps in our problem.

Table 1 shows the percentage of the successful trials in all 100 trials. As shown in the Table 1, SLM has the ability to acquire an deterministic optimal policy.

Table 1 : Percentage of successful trials

■Conclusion

Conclusion

We proposed SLM, and empirically showed that the performance was improved dominantly by using SLM, which is an extension of SL. In addition, we confirmed that SLM has the ability to acquire a deterministic optimal policy, which cannot be achieved by SL.

Table 2

S × M	(SG,1)	(SG,2)	(C,1)	(C,2)	(B,1)	(B,2)
π_a	Fore	Fore	Fore	Back	Back	Back
π_c	1	1	1	2	2	2

The table 2 shows that $\pi_c : S \times M \to M$ obviously allows each agent to include the activation status in a message, i.e., $1 \in M$ as inactivated (status of button = OFF) and $2 \in M$ as activated (status of button = ON).

Table 2 : A simplest example of the acquired deterministic optimal policy ($|M|$=2)

The 8th International Conference on Autonomous Agents and Multiagent Systems, May 13, 2009, Budapest, Hungary.