Supervisory Control of Partially Observed Discrete Event Systems based on a Reinforcement Learning*

Tatsushi YAMASAKI
School of Science and Technology
Kwansei Gakuin University
2-1 Gakuen, Sanda-shi,
Hyogo, 669-1337 Japan
tatsushi@ksc.kwansei.ac.jp

Toshimitsu USHIO
Graduate School of Engineering Science
Osaka University
1-3 Machikaneyama, Toyonaka-shi,
Osaka, 560-8531, Japan
ushio@sys.es.osaka-u.ac.jp

Abstract – *In discrete event systems, the supervisor controls events to satisfy the control specifications given by formal languages. However a precise description of the specifications and the discrete event systems is required for constructing the supervisor. So, this paper proposes a method to construct a supervisor based on a reinforcement learning for partially observed discrete event systems. In the proposed method, specifications are given by rewards, and an optimal supervisor is derived by considering rewards for the occurrence of events and disabling events. Moreover learning speed is accelerated by updating plural Q values. It is done by utilizing characteristics of a supervisory control. An efficiency of the proposed method is examined by computer simulation.*

The proposed method shows a new approach for applying a supervisory control in the case of implicit specifications and uncertain environment.

Keywords: Supervisory control, reinforcement learning, discrete event systems, partial observation, optimal control.

1 Introduction

Discrete event systems (DESs) are widely found in artificial systems, for example database management systems, communication systems, production systems, and operating systems. In DESs, discrete events make transitions asynchronously and concurrently. Various researches about the control of DESs have been made actively [2].

Ramadge and Wonham proposed the supervisory control as a logical control method for DESs [8], [12]. This synthesizes a controller called a supervisor. The supervisor disables controllable events in the DES so that all possible strings satisfy control specifications. A set of events permitted to occur is called a control pattern.

On the other hand, a reinforcement learning has been attracted as a learning method in recent years. Reinforcement learning is a learning method to obtain a policy based on rewards given from an environment through trial and error [1], [10].

The supervisor assigns an optimal control pattern in the sense that language generated by the system is maximized within given specifications. The reason is less restriction for the DES is preferred. The supervisory control requires a precise description of the specification and the DES by a formal language or an automaton. It is a tedious and difficult task practically. Moreover the supervisory control is a logical control method, and disables events of the DES. However costs for enabling or disabling events are not considered. On the other hand, in reinforcement learning, a learner obtains an optimal policy in the sense the expected total reward is maximized through trial and error.

In this paper, we propose a synthesis method of a supervisor based on a reinforcement learning. Costs for enabling and disabling events are considered. We introduce rewards for representation of control specifications, and a detail of the specifications is obtained through learning. By using reinforcement learning, automatization and simplification of synthesis of the supervisor is achieved. The supervisor learns desirable control patterns. Both satisfaction of the control specifications and costs of them are considered at the same time. Moreover the proposed method accelerates the learning speed by using characteristics of the supervisory control. This study is extension from our previous study, which is only considered a full observation case [13]. In general, it is difficult to meet the requirements of full observation in DESs. So we deal with a partial observation case about the occurrence of events in DESs.

Several studies is done about an optimal control of DESs. By using the supervisor constructed in advance, a synthesis method of the optimal supervisor which minimizes total costs about enabling and disabling event is proposed [5], [9]. An optimal supervisory control in a partial observation case is also proposed [7]. To compared with other studies, our method has the following advantages. Reinforcement learning is introduced so that implicit specifications are taken into consideration and the supervisor can adapt to changing en-

*0-7803-7952-7/03/$17.00

vironments. Moreover synthesis of the optimal supervisor which is considered the control specifications and costs simultaneously is realized.

This paper is organized as follows. Section 2 reviews supervisory control and reinforcement learning briefly. Section 3 describes a system model discussed in this paper, and section 4 proposes a synthesis method of the supervisor based on a reinforcement learning. Section 5 demonstrates the efficiency of the proposed method by computer simulation. Section 6 provides the conclusion.

2 Preliminaries

2.1 Supervisory control

A supervisor for a DES controls the occurrence of events for satisfying logical control specifications [8], [12]. Examples of specifications are FIFO transaction, avoidance of deadlock, etc. Events in the DES are divided into controllable and uncontrollable, and the control is done by enabling or disabling controllable events.

A DES controlled by a supervisor is illustrated as Fig. 1. The supervisor gives a control pattern to the DES. The control pattern is a set of events permitted to occur. Feasible uncontrollable events could not be disabled to occur. Therefore they are assumed to be included in the control pattern. Then the DES selects an event in the control pattern. As a result, the DES makes a transition to a new state. There is a mask between the DES and the supervisor, and the mask removes unobservable events from event strings [3], [6]. Therefore, the supervisor can observe only the occurrence of observable events. In other words, unobservable events may have been executed in the DES before observation of the event by the supervisor. If the supervisor observes an event, the supervisor changes the control pattern based on observed event strings. The control is achieved by repetition of the above process.

To construct a supervisor, the DES and the specifications are represented by a language and an automaton respectively. Then the supervisor is derived by operations between the automata. In full observation, a polynomial time algorithm which maximizes the language generated by the controlled system is shown. However a precise description of the specifications needs the computational cost. For example, the computational cost of the supervisor which satisfies plural specifications is *NP*-hard [4].

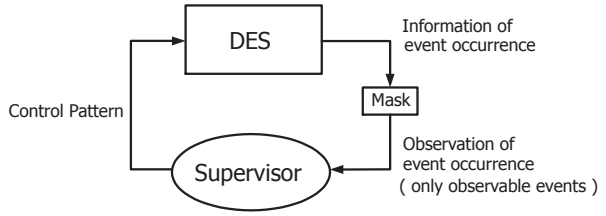

Figure 1: The DES controlled by the supervisor with mask

2.2 Reinforcement learning

Reinforcement learning is a learning method that a learner obtains numerical rewards from an environment and learns a desirable behavior policy. Learning through trial and error is effective in the case of an uncertain environment, and a learner adapt to changing environment autonomously [10].

Q-learning is one of the reinforcement learning algorithms. It updates Q values which are evaluations for state-action pairs. When a learner makes a transition from a current state x to a new state x' by action a and obtains reward r, Q values are updated as follows:

$$Q(x,a) \leftarrow Q(x,a) + \alpha\big[r + \gamma \max_{a'} Q(x',a') - Q(x,a)\big], \quad (1)$$

where $Q(x,a)$ is the estimation of the expected discounted total rewards when a learner takes an action a at a state x, α is a learning rate $(0 < \alpha < 1)$, and γ is a discounted rate of rewards $(0 < \gamma < 1)$. The Q values converges with probability 1 to a true value if α decays appropriately and update of the Q values is done an infinite number of times at each state-action pair [11].

3 System model

In this section, we show the system model considered in the paper. We consider the DES controlled by the supervisor shown in Fig. 1. We suppose that the system is a Markov Decision Process (MDP) with respect to observation from the supervisor. Then, Bellman optimal equation for the supervisor is described as follows:

$$Q^*(s,\pi) = \sum_{s' \in S} \mathcal{P}(s,\pi,s') \left[\mathcal{R}(s,\pi,s') + \gamma \max_{\pi' \in \Pi(s')} Q^*(s',\pi')\right], \quad (2)$$

where

- X : a set of states of the DES. The supervisor can't observe directly.
- Σ : a set of events. Σ is divided into a set of controllable events Σ^c and a set of uncontrollable events Σ^{uc}. Moreover it is also divided into a set of observable events Σ^o and a set of unobservable events Σ^{uo}. Namely, $\Sigma = \Sigma^c \cup \Sigma^{uc}$, $\Sigma^c \cap \Sigma^{uc} = \emptyset$, $\Sigma = \Sigma^o \cup \Sigma^{uo}$, and $\Sigma^o \cap \Sigma^{uo} = \emptyset$.
- S : a set of states of the supervisor. S is represented by a subset of the set of states of the DES, namely $S \subseteq 2^X$. The supervisor can only know candidates of a state of the DES as $s \in S$.
- $F(s)$: a feasible event set at state $s \in S$.
- $\Pi(s)$: a set of control patterns at state $s \in S$. Each control pattern $\pi \in \Pi(s)$ is a set of events permitted to

occur at state s. For all $\pi \in \Pi(s)$, $F(s) \cap \Sigma^{uc} \subseteq \pi \subseteq F(s) \subseteq \Sigma$. Namely, feasible uncontrollable events are always included in the control pattern.

- $\mathcal{P}(s, \pi, s')$: a probability to make a transition from state s to s' when the supervisor selects $\pi \in \Pi(s)$.
- $Q^*(s, \pi)$: a discounted expected total reward in the case that the supervisor selects $\pi \in \Pi(s)$ at state s, then continues to select control patterns optimally.
- $\mathcal{R}(s, \pi, s')$: an expected reward when the supervisor selects $\pi \in \Pi(s)$ at state s, and make a transition to state $s' \in S$.
- γ : a discount rate of rewards ($0 < \gamma < 1$).

We consider a partial observation case about the DES from the supervisor. Therefore we introduce the mechanism to estimate the current state of the DES from strings of events observed by the supervisor. First, we define several functions and symbols.

- f: $X \times \Sigma \rightarrow X$, a state transition function of the DES. In this paper, we suppose that f is known for the supervisor. Moreover, we extend f to a function $f : X \times \Sigma^* \rightarrow X$ as follows:

$$f(x, \epsilon) = x, \tag{3}$$

For $t \in \Sigma^*$ and $\sigma \in \Sigma$

$$f(x, t\sigma) = f(f(x, t), \sigma), \tag{4}$$

where ϵ is the empty string.

- $x_0 \in X$: an initial state of the DES.
- Σ^* : Kleene closure of Σ. That is a set of all finite strings over Σ, including the empty string ϵ.
- M_e: $\Sigma \rightarrow \Sigma^o \cup \{\epsilon\}$, a projection function from an event $\sigma \in \Sigma$ which occurred in the DES to an event $\sigma^o \in \Sigma^o$ or ϵ that the supervisor observes :

$$M_e(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Sigma^o \\ \epsilon & \text{if } \sigma \in \Sigma^{uo}. \end{cases} \tag{5}$$

Moreover, we extend M_e to a function $M_e : \Sigma \rightarrow \Sigma^{o*}$ as follows:

$$M_e(\epsilon) = \epsilon, \tag{6}$$

$\forall t \in \Sigma^*, \forall \sigma \in \Sigma$

$$M_e(t\sigma) = \begin{cases} M_e(t)\sigma & \text{if } \sigma \in \Sigma^o \\ M_e(t) & \text{if } \sigma \in \Sigma^{uo}. \end{cases} \tag{7}$$

$M_e(t)$ gives the observed string by removing unobservable events from a string t.

- M_s: $\Sigma^{o*} \rightarrow S$, a projection function from an observed strings $t \in \Sigma^{o*}$ to a state of the supervisor:

$$M_s(t) = \{x \in X \,; \exists u \in \Sigma^*, \\ M_e(u) = t, \quad f(x_0, u) = x\}. \tag{8}$$

$M_s(t)$ gives a set of states of the DES whose observed strings is t. It represents candidates of the current state of the DES, and the state of the supervisor s as already defined.

The DES selects an event to occur in the control pattern given by the supervisor. The supervisor decides the control pattern based on only the observed information. Therefore, the following equation holds:

$$\mathcal{P}(s, \pi, s') = \sum_{\sigma^o \in \pi \cap \Sigma^o} \mathcal{P}_1(s, \pi, \sigma^o)\mathcal{P}_2(s, \sigma^o, s'), \tag{9}$$

where

- $\mathcal{P}_1(s, \pi, \sigma^o)$: a probability that the supervisor observes the occurrence of an event $\sigma^o \in \pi \cap \Sigma^o$ when the supervisor selects $\pi \in \Pi(s)$ at $s \in S$.
- $\mathcal{P}_2(s, \sigma^o, s')$: a probability that the supervisor makes a transition from state s to s' by an observed event σ^o.

We make additional hypotheses for the DES.

1. The DES has a parameter $\eta^*(s, \sigma^o)$ for each $s \in S$ and $\sigma^o \in F(s) \cap \Sigma^o$. Then the following equations hold:

$$\mathcal{P}_1(s, \pi, \sigma^o) = \frac{\eta^*(s, \sigma^o)}{\sum_{\sigma^{o'} \in \pi \cap \Sigma^o} \eta^*(s, \sigma^{o'})}, \tag{10}$$

$$\eta^*(s, \sigma^o) > 0\,, \quad \sum_{\sigma^{o'} \in F(s) \cap \Sigma^o} \eta^*(s, \sigma^{o'}) = 1. \tag{11}$$

The DES selects an observable event in the control pattern, and the supervisor observes the event. Then, a rate that the supervisor observes the event is given by Eq. (10). The supervisor does not know the true value of η^*.

2. Reward $\mathcal{R}(s, \pi, x')$ has structure as follows:

$$\mathcal{R}(s, \pi, s') = \mathcal{R}_1(s, \pi) + \mathcal{R}_2(s, \sigma^o, s'), \tag{12}$$

where

• $\mathcal{R}_1(s, \pi)$: an expectation of reward when the supervisor selects π at s. It depends on the control pattern, and represents for the cost to disable events which is not contained in the control pattern intuitively.

• $\mathcal{R}_2(s, \sigma^o, s')$: an expectation of reward when the supervisor observes an event $\sigma^o \in \Sigma^o$ and makes a transition from s to s'. Intuitively, it represents for costs by execution of the event and evaluation about the achievement of the task.

We get the following equation from Eq. (2) by using the above hypotheses:

$$
\begin{aligned}
& Q^*(s,\pi) \\
&= \sum_{s'\in S}\left(\sum_{\sigma^o\in\pi\cap\Sigma^o}\frac{\eta^*(s,\sigma^o)}{\sum_{\sigma^{o'}\in\pi\cap\Sigma^o}\eta^*(s,\sigma^{o'})}\mathcal{P}_2(s,\sigma^o,s')\right) \\
&\qquad\left[\mathcal{R}_1(s,\pi)+\mathcal{R}_2(s,\sigma^o,s')+\gamma\max_{\pi'\in\Pi(s')}Q^*(s',\pi')\right] \\
&= \mathcal{R}_1(s,\pi)+\sum_{\sigma\in\pi\cap\Sigma^o}\frac{\eta^*(s,\sigma^o)}{\sum_{\sigma^{o'}\in\pi\cap\Sigma^o}\eta^*(s,\sigma^{o'})} \\
&\quad\sum_{s'\in S}\mathcal{P}_2(s,\sigma^o,s')\left[\mathcal{R}_2(s,\sigma^o,s')+\gamma\max_{\pi'\in\Pi(s')}Q^*(s',\pi')\right] \\
&= \mathcal{R}_1(s,\pi)+\sum_{\sigma^o\in\pi\cap\Sigma^o}\frac{\eta^*(s,\sigma^o)}{\sum_{\sigma^{o'}\in\pi\cap\Sigma^o}\eta^*(s,\sigma^{o'})}T^*(s,\sigma^o), \quad (13)
\end{aligned}
$$

where $T^*(s,\sigma^o)$ is a discounted expected total reward when the supervisor observes σ^o at state s, and continues to select the control pattern which has the maximum Q value in the following states, and defined as follows:

$$
\begin{aligned}
T^*(s,\sigma^o) &= \sum_{s'\in S}\mathcal{P}_2(s,\sigma^o,s') \\
&\qquad\left[\mathcal{R}_2(s,\sigma^o,s')+\gamma\max_{\pi'\in\Pi(s')}Q^*(s',\pi')\right]. \quad (14)
\end{aligned}
$$

In $T^*(s,\sigma^o)$, rewards by selecting π at s are not included.

4 The proposed algorithm

We propose a learning method of the supervisor for the system model mentioned in section 3. We illustrate its conceptual diagram in Fig. 2, and its algorithm is given by Fig. 3, where an episode is a series of events and states, and starts from an initial state and ends at a terminal state. By repeating episodes, the supervisor proceeds learning.

The objective of learning is not to learn what an event should be selected, but to learn what a control pattern should be selected.

By applying Q-learning to framework of the supervisory control, the supervisor learns an optimal control pattern which maximizes the expected discounted total reward through trial and error. Moreover, acceleration by learning of plural Q values is achieved in the proposed method.

When the state of the supervisor is $s \in S$, the supervisor selects a control pattern $\pi \in \Pi(s)$. There are several methods for this selection. We use the ϵ-greedy selection in this time. So, the supervisor selects the control pattern which has the maximum Q value with probability $1-\epsilon$, and selects it randomly with probability ϵ.

The DES selects an event $\sigma \in \pi$. This selection is not affected by the supervisor, but restricted by Eq. (10). If an observable event $\sigma^o \in \Sigma^o$occurred in the DES, the supervisor observes it. Since there is the mask between the DES and the supervisor, the supervisor can't know the execution

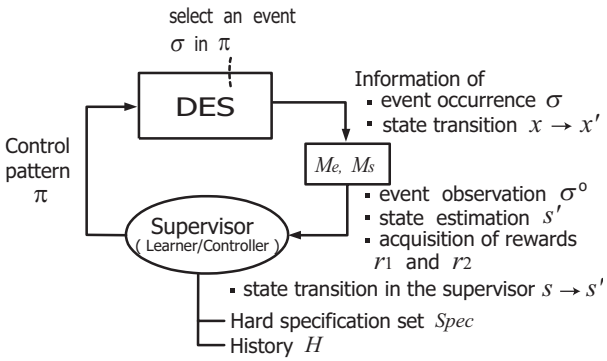

Figure 2: The DES controlled by the learning supervisor

of unobservable event in the DES. There is a possibility that unobservable events occur in the DES before the supervisor observes an event. The supervisor obtains two types of reward. One is evaluation for the control pattern π denoted by r_1, and the other is the evaluation for observation of σ^o denoted by r_2.

In Eq. (13), Q^* is calculated by $\mathcal{R}_1$, η^*, and T^*. In other words, it is possible to estimate Q^* indirectly by using $\mathcal{R}_1$, η^*, and T^*. Therefore, we prepare three leaning parameters R_1, η, and T. The supervisor updates them as follows:

$$
\begin{aligned}
T(s,\sigma^o) &\leftarrow T(s,\sigma^o)+\alpha[r_2 \\
&\quad +\gamma\max_{\pi'\in\Pi(s')}Q(s',\pi')-T(s,\sigma^o)], \quad (15) \\
R_1(s,\pi) &\leftarrow R_1(s,\pi)+\beta[r_1-R_1(s,\pi)], \quad (16)
\end{aligned}
$$

For all $\sigma^{o'} \in \pi\cap\Sigma^o$

$$
\eta(s,\sigma^{o'}) \leftarrow \begin{cases} (1-\delta)\,\eta(s,\sigma^{o'}) \\ \quad (\text{if } \sigma^{o'}\neq\sigma^o) \\ \eta(s,\sigma^{o'})+\delta\left[\sum_{\sigma^{o''}\in\pi\cap\Sigma^o}\eta(s,\sigma^{o''})\right. \\ \quad \left.-\eta(s,\sigma^{o'})\right] \\ \quad (\text{if } \sigma^{o'}=\sigma^o), \end{cases} \quad (17)
$$

where α, β, and δ are learning rates. Then, the supervisor updates Q values by using T, R_1, and η as follows:

For all $\pi' \in \Pi(s)\ s.t.\ \pi'\cap\pi\neq\emptyset$

$$
\begin{aligned}
Q(s,\pi') &\leftarrow R_1(s,\pi') \\
&\quad +\sum_{\sigma^{o''}\in\pi'\cap\Sigma^o}\frac{\eta(s,\sigma^{o''})}{\sum_{\sigma^{o'''}\in\pi'\cap\Sigma^o}\eta(s,\sigma^{o'''})}T(s,\sigma^{o''}). \quad (18)
\end{aligned}
$$

The updates is done for not only the control pattern π selected actually, but also control patterns which an event in π is included.

In ends of each step of the episode, the supervisor checks whether the current state is contained in $Spec$, which is a set

of states defined as minimum specifications. If $s \notin Spec$, the supervisor specifies the latest controllable observable event and the state at that time, and removes the event from the feasible event set at the state. At the same time, update η under the constraint $\sum \eta = 1$, and calculate Q value again. These process removes strings which does not satisfy $Spec$. It is regarded as a kind of pruning for efficiency of learning and assurance of the minimum specifications.

The proposed algorithm synthesizes a supervisor through learning of control patterns which maximize the expected total reward. Both specifications and costs of the DES is considered at the same time, and it is required to satisfy a hard specification $Spec$. The algorithm in the case of full observation is regarded as $\Sigma^o = \Sigma$ in the proposed algorithm.

5 Simulation

We demonstrate efficiency of the proposed method by computer simulation of the cat and mouse problem. It is a simple problem used in [12]. There are five rooms partitioned by one-way doors as shown in Fig. 4. Each door is used by a cat or a mouse exclusively. In Fig. 4, $c1 \sim c7$ are doors for a cat, and $m1 \sim m6$ are doors for a mouse. In the original problem, $c7$ is an uncontrollable door, but $c7$ is an uncontrollable and unobservable door in our setting. Other doors are controllable and observable. A goal is to control doors so as not to encounter a cat and a mouse in the same room. A control pattern means what doors should be closed.

In initial state, a cat is in room 2, and a mouse is in room 4. For closing each door except $c7$, it takes a cost. The cost depends on a normal distribution that the average is -1 and the variance is 0.1. Hence a reward r_1 is given by sum of costs to close doors. One episode ends when 20 step passed or the cat and the mouse encountered in a room. In the latter case, a reward $r_2 = -100$ is given for fail of control. Other parameters are set as follows: $\alpha = \beta = \delta = 0.1$, and $\gamma = 0.9$. We use $\epsilon-$greedy selection to select a control pattern, and set $\epsilon = 0.1$.

Fig. 5 shows the relationship between the number of episodes and the fraction that the supervisor found the optimal control pattern in the maze for cat and mouse problem. In order to ¡make a comparison with the proposed method, we used a simple Q-learning. In the simple Q-learning, Q value is updated only for a control pattern selected by the supervisor actually. The result is the average of learning of 100 times. In both methods, the optimal supervisor is obtained after several hundreds of episodes, but the proposed method is superior to the simple Q-learning with respect to speed of learning. Fig. 6 shows the transition diagram of the learned control pattern. In each circle, the first digit shows a room in which a cat exists, and the second digit shows a room in which a mouse exists, respectively. Each arrow shows a door allowed to open in the source state. The supervisor controlling doors so as not to encounter a cat and a mouse is synthesized through learning.

1. Initialize $T(s,\sigma^o)$, $R_1(s,\pi)$, and $\eta(s,\sigma^o)$ at each state in the supervisor.
2. Calculate the initial Q value at each state by

$$Q(s,\pi) \leftarrow R_1(s,\pi) + \sum_{\sigma^o \in \pi \cap \Sigma^o} \frac{\eta(s,\sigma^o)}{\sum_{\sigma^{o'} \in \pi \cap \Sigma^o} \eta(s,\sigma^{o'})} T(s,\sigma^o).$$

3. Repeat until s is a *terminal state* (for each episode):
 (a) Clear history H and initialize strings $t \leftarrow \epsilon$.
 (b) $s \leftarrow initial\ state$ in DES.
 (c) Repeat (for each step of episode)
  i. Select a control pattern $\pi \in \Pi(s)$ based on the Q values by the supervisor.
  ii. Observe the occurrence of event $\sigma^o \in \Sigma^o$, update $t \leftarrow t\sigma^o$, and estimate a new state $s'(= M_s(t))$.
  iii. Acquire rewards r_1 and r_2.
  iv. Make a transition $s \stackrel{\sigma^o}{\rightarrow} s'$ in the supervisor.
  v. Add (s,σ^o) to history H.
  vi. Update $T(s,\sigma^o)$, $\mathcal{R}_1(s,\pi)$, and $\eta(s,\sigma^{o'})$:

$$T(s,\sigma^o) \leftarrow T(s,\sigma^o) + \alpha[r_2 + \gamma \max_{\pi' \in \Pi(s')} Q(s',\pi') - T(s,\sigma^o)],$$

$$R_1(s,\pi) \leftarrow R_1(s,\pi) + \beta[r_1 - R_1(s,\pi)],$$

For all $\sigma^{o'} \in \pi \cap \Sigma^o$

$$\eta(s,\sigma^{o'}) \leftarrow \begin{cases} (1-\delta)\eta(s,\sigma^{o'}) \\ \quad (\text{if } \sigma^{o'} \neq \sigma^o) \\ \eta(s,\sigma^{o'}) \\ +\delta \left[\sum_{\sigma^{o''} \in \pi \cap \Sigma^o} \eta(s,\sigma^{o''}) - \eta(s,\sigma^{o'}) \right] \\ \quad (\text{if } \sigma^{o'} = \sigma^o). \end{cases}$$

  vii. Update the Q values:

For all $\pi' \in \Pi(s)\ s.t.\ \pi' \cap \pi \neq \emptyset$

$$Q(s,\pi') \leftarrow R_1(s,\pi') + \sum_{\sigma^{o''} \in \pi' \cap \Sigma^o} \frac{\eta(s,\sigma^{o''})}{\sum_{\sigma^{o'''} \in \pi' \cap \Sigma^o} \eta(s,\sigma^{o'''})} T(s,\sigma^{o''}).$$

  viii. If $s' \notin Spec$:
   A. Search the latest observable and controllable event $\sigma^{c,o} \in \Sigma^c \cap \Sigma^o$ and the corresponding state $w \in S$ from the history H.
   B. Remove $\sigma^{c,o}$ from the feasible event set $F(w)$.
   C. Normalize $\eta(w,\sigma^{o'})$ so as to satisfy $\sum_{\sigma^{o'} \in F(w) \cap \Sigma^o} \eta(w,\sigma^{o'}) = 1$, and update the Q values at the state w.
  ix. $s \leftarrow s'$.

Figure 3: The proposed algorithm for construction of a supervisor

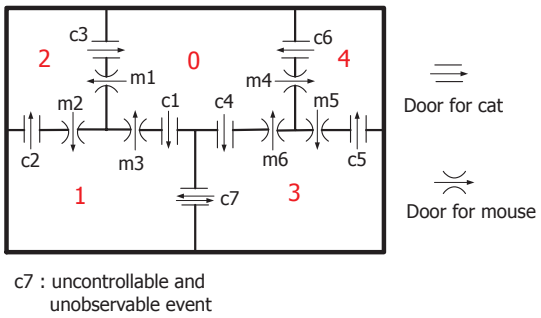

Figure 4: Maze for cat and mouse

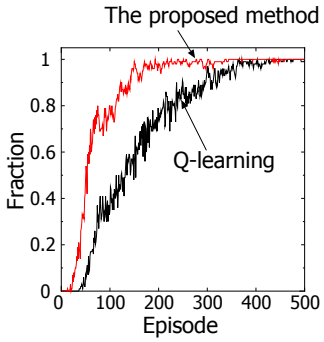

Figure 5: Relation between the number of episodes and the fraction that the supervisor found the optimal control pattern in the maze for cat and mouse problem

6 Conclusions

This paper proposed a synthesis method to construct a supervisor based on Q-learning by using characteristics of supervisory control. The partial observation case is considered. In ordinary supervisory control, it is a difficult and tedious task to describe control specifications. In the proposed method, detail of specifications are obtained through learning based on rewards. We show a method to apply supervisory control for more wide problems under implicit specifications and changing environment. In the proposed method, there is a problem that a number of control patterns increases exponentially as a number of events increases.

It is needed that extension to the decentralized supervisory control by using multi-agent oriented reinforcement learning. Theoretical analysis with respect to a maximal controllable languages in the proposed method is also future work.

Acknowledgements

This work has been supported by CREST of JST(Japan Science and Technology).

References

[1] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, 1996.

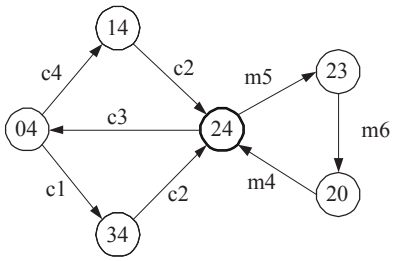

Figure 6: The representation of the learned control pattern by the transition diagram

[2] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Kluwer Academic Pub., 1999.

[3] R. Cieslak, C. Desclaux, A. S. Fawaz and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," *IEEE Trans. Automatic Contr.*, Vol. 33, No. 3, pp. 249–260, 1988.

[4] P. Gohari and W. M. Wonham, "On the complexity of supervisory control design in the RW framework," *IEEE Trans. Syst., Man, Cybern. B*, Vol. 30, No. 5, pp. 643–652, 2000.

[5] R. Kumar and V. K. Garg, "Optimal supervisory control of discrete event dynamical systems," *SIAM J. Control Optim.*, Vol. 33, No. 2, pp. 419–439, 1995.

[6] F. Lin and W. M. Wonham, "On observability of discrete-event systems," *Inf. Sci.*, Vol. 44, pp. 173–198, 1988.

[7] H. Marchand, O. Boivineau and S. Lafortune, "Optimal control of discrete event systems under partial observation", Proc. of the 40the IEEE Conference on Decision and Control, pp. 2335–2340, 2001.

[8] P. J. Ramadge and W. M. Wonham, "Supervisory control of a class of discrete-event processes," *SIAM J. Control Optim.*, Vol. 25, No. 1, pp. 206–230, 1987.

[9] R. Sengupta and S. Lafortune, "An optimal control theory for discrete event systems," *SIAM J. Control Optim.*, Vol. 36, No. 2, pp. 488–541, 1998.

[10] R. S. Sutton and A. G. Barto, *Reinforcement Learning*, MIT Press, 1998.

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

[12] W. M. Wonham and P. J. Ramadge, "On the supremal controllable sublanguage of a given language," *SIAM J. Control Optim.*, vol. 25, No. 3, pp. 637–659, 1987.

[13] T. Yamasaki and T. Ushio, "Supervisory control of discrete event systems based on a reinforcement learning," Trans. of ISCIE, Vol. 16, No. 3, pp. 118–124, 2003 (in Japanese).