Decentralized Supervisory Control of Discrete Event Systems Based on Reinforcement Learning

Article *in* Transactions of the Institute of Systems Control and Information Engineers · November 2005

DOI: 10.1093/ietfec/e88-a.11.3045 · Source: OAI

2 authors, including:

Toshimitsu Ushio
Nanzan University
373 PUBLICATIONS **3,063** CITATIONS

SEE PROFILE

PAPER Special Section on Concurrent/Hybrid Systems: Theory and Applications

Decentralized Supervisory Control of Discrete Event Systems Based on Reinforcement Learning

Tatsushi YAMASAKI[†a)] ***and*** **Toshimitsu USHIO**[††b)]**,** ***Members***

SUMMARY A supervisor proposed by Ramadge and Wonham controls a discrete event system (DES) so as to satisfy logical control specifications. However a precise description of both the specifications and the DES is needed for the control. This paper proposes a synthesis method of the supervisor for decentralized DESs based on reinforcement learning. In decentralized DESs, several local supervisors exist and control the DES jointly. Costs for disabling and occurrence of events as well as control specifications are considered. By using reinforcement learning, the proposed method is applicable under imprecise specifications and uncertain environment.

key words: *discrete event systems, decentralized control, supervisory control, reinforcement learning, optimal control*

1. Introduction

Various man-made systems, such as communication systems, transportation systems, and manufacturing systems can be modeled by discrete event systems (DESs) [1]. The supervisory control initiated by Ramadge and Wonham is a logical control method for DESs [2]–[4]. It controls the occurrence of controllable events so as to satisfy logical control specifications, such as FIFO transaction and avoidance of deadlock. A controller called a supervisor assigns a control pattern to the DES, which is a set of events permitted to occur at the current state of the DES. In a standard approach, based on the DES and specifications represented by a formal language or an automaton, the supervisor which maximizes the language generated by the controlled system is derived. However, in order to synthesize the supervisor, a precise description of the specifications and the DES is required and computation takes a large cost if the DES is under partial observation or decentralized. Moreover, costs for disabling and occurrence of events are not considered.

Several researchers studied the supervisory control problems which takes into account the cost of events. Brave and Heymann introduced the concept of attraction and proposed the method to control the system within a prescribed state set with the minimum cost of the convergence path [5]. Kumar and Garg considered the optimal supervisory control in the sense the cost of disabling events and the reward for reaching desired or undesired states are minimized [6]. Sengupta and Lafortune proposed an algorithm to compute an optimal supervisor based on dynamic programming. They considered a cost of occurrence and disabling events, and adopted a worst-case cost as a condition of optimality [7]. Wang and Ray introduced a signed real measure, called a language measure, for formal languages [8] and an optimal supervisory control based on a language measure is also proposed [9]. A language measure is a performance index given for the languages generated by DESs. It is possible to evaluate the performance of the DESs quantitatively based on the language measure.

The reinforcement learning is a learning method to obtain a policy based on rewards given from an environment through trial and error [10], [11]. A learner learns an optimal policy which maximizes a total expected reward under uncertain environment.

The authors have proposed a supervisory control method based on reinforcement learning under partial observation [12]. In this paper, a decentralized supervisory control method based on reinforcement learning is proposed [13]. Because of inherent restrictions of the systems or controllers, many DESs are often controlled in a decentralized manner where local supervisors assign control patterns based on their local observations. In the proposed method, supervisors exist locally and control the DES jointly without direct negotiation between the supervisors. Each supervisor obtains a control pattern through learning without knowledge of detailed specifications and precise costs in the decentralized DES. Automatization and simplification of synthesis of the supervisors is achieved by learning. Costs for disabling and occurrence of events are also considered by getting rewards. By using reinforcement learning, the proposed method is applicable under imprecise specifications and uncertain environment.

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

2. Reinforcement Learning

Reinforcement learning is a learning method such that a learner obtains numerical rewards from an environment and learns a desirable behavior policy. Learning through trial

Manuscript received April 7, 2005.
Manuscript revised June 3, 2005.
Final manuscript received July 11, 2005.

[†]The author is with the School of Science and Technology, Kwansei Gakuin University, Sanda-shi, 669-1337 Japan.
[††]The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560-8531 Japan.
a) E-mail: tatsushi@ksc.kwansei.ac.jp
b) E-mail: ushio@sys.es.osaka-u.ac.jp

DOI: 10.1093/ietfec/e88–a.11.3045

and error is effective in the case of uncertain environment, and a learner can adapt to changing environment [11].

Q-learning is one of the reinforcement learning algorithms [14]. 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 an action a and obtains a 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 $Q(x,a)$ denotes an estimation of the expected discounted total rewards when a learner takes an action a at a state x, α denotes a learning rate ($0 \le \alpha < 1$), and γ denotes a discounted rate of rewards ($0 \le \gamma < 1$).

The Q values converges with probability 1 to a true value as the number of updates of the Q values goes to the infinity if the following conditions are satisfied:

$$\sum_{k=1}^{\infty} \alpha_k(x,a) = \infty, \qquad \sum_{k=1}^{\infty} \alpha_k(x,a)^2 < \infty, \quad (2)$$

where $\alpha_k(x,a)$ denotes a learning rate at a state x when an action a is selected k times. The Q-learning is applicable for which the environment is a Markov decision process (MDP). However, it often shows good performance even if the environment is not a MDP strictly.

3. System Model

The decentralized supervisory control architecture is composed of n local supervisors and a DES [15]. Each local supervisor is denoted by SV_i ($i = 1, \cdots, n$) and has the corresponding learning unit denoted by LU_i. Figure 1 illustrates the system model in the case that two supervisors exist.

A DES G is modeled by a 4-tuple (X, Σ, f, x_0), where

- X is a set of states of the DES.
- Σ is a set of events. $\Sigma^c \subseteq \Sigma$ is a set of controllable events and $\Sigma^o \subseteq \Sigma$ is a set of observable events. Each SV_i has a set of controllable events $\Sigma_i^c \subseteq \Sigma^c$, where $\cup_{i=1}^n \Sigma_i^c = \Sigma^c$. Furthermore, each SV_i has a set of observable events $\Sigma_i^o \subseteq \Sigma^o$, where $\cup_{i=1}^n \Sigma_i^o = \Sigma^o$. Σ_i^{uc} denotes a set of uncontrollable events of SV_i and is defined by $\Sigma_i^{uc} = \Sigma - \Sigma_i^c$. Σ_i^{uo} denotes a set of unobservable events of SV_i and is defined by $\Sigma_i^{uo} = \Sigma - \Sigma_i^o$.
- f is a state transition function of the DES. In this paper, f is assumed to be a deterministic transition function from $X \times \Sigma$ to X for simplicity. Moreover, f is extended to a function $f : X \times \Sigma^* \to X$ as follows:

$$f(x,\epsilon) = x, \quad (3)$$

$$\forall t \in \Sigma^*, \forall \sigma \in \Sigma \quad f(x,t\sigma) = f(f(x,t),\sigma), \quad (4)$$

  where ϵ is the empty string and Σ^* denotes the Kleene closure of Σ, that is, a set of all finite strings over Σ, including ϵ. In this paper, f is assumed to be known for each supervisor.
- $x_0 \in X$ is an initial state of the DES.

We also define an active event set $F_G(x)$ as follows:

$$F_G(x) = \{\sigma \in \Sigma \mid \text{if } f(x,\sigma) \text{ is defined}\}. \quad (5)$$

Each supervisor can not necessarily observe all events in the decentralized system. So, a projection between the DES and each SV_i are introduced [16]. Each supervisor observes the occurrence of events in the DES through the corresponding projection. M_i^e: $\Sigma \to \Sigma_i^o \cup \{\epsilon\}$ denotes a projection for SV_i and defined as follows:

$$M_i^e(\sigma) = \begin{cases} \sigma & \text{if } \sigma \in \Sigma_i^o, \\ \epsilon & \text{if } \sigma \in \Sigma_i^{uo}. \end{cases} \quad (6)$$

Moreover, M_i^e is extended to a function $M_i^e : \Sigma \to \Sigma_i^{o*}$ as follows:

$$M_i^e(\epsilon) = \epsilon, \quad (7)$$

$$\forall \in \Sigma^*, \ \forall \sigma \in \Sigma \quad M_i^e(t\sigma) = \begin{cases} M_i^e(t)\sigma & \text{if } \sigma \in \Sigma_i^o, \\ M_i^e(t) & \text{if } \sigma \in \Sigma_i^{uo}. \end{cases} \quad (8)$$

$M_i^e(t)$ gives the observed string to each SV_i by removing all unobservable events from a string t.

We model each SV_i by an automaton $(S_i, \Sigma_i^o, g_i, x_0)$, where

- $S_i \subseteq 2^X$ is a set of states of SV_i, which is represented by a subset of the set of states of the DES. Its element is candidates of the current states of the DES G.
- $g_i : S_i \times \Sigma_i^o$ is a state transition function and is extended to a function $g : S_i \times \Sigma_i^{o*} \to S_i$ in the similar way of f.

Since SV_i observes the occurrence of events through a projection M_i^e, it can only know candidates of a state of the DES as $s_i \in S_i$. To determine the state of SV_i, a state estimate function $M_i^s : X \times \Sigma_i^{o*} \to 2^X$ is introduced as follows:

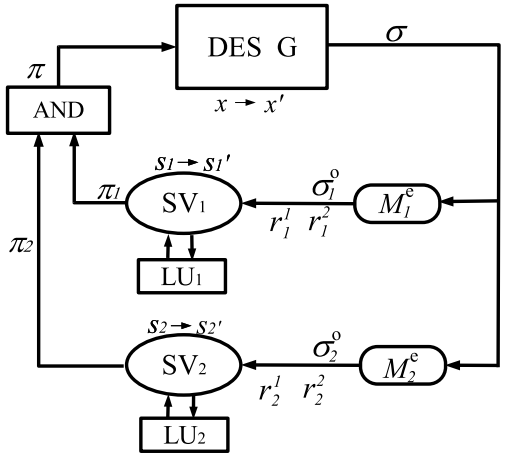

Fig. 1 The DES controlled by two decentralized supervisor with two learning units.

$$M_i^s(t,u) = \{x \in X \mid \exists v \in \Sigma^*, \\ M_i^e(v) = u, f(t,v) = x\}. \quad (9)$$

Then, $M_i^s(t,u)$ gives a set of states of the DES which is reachable from a state t via a string observed as u. By using M_i^s, g_i is described by

$$g_i(s_i,\sigma) = \bigcup_{x\in s_i} M_i^s(x,\sigma). \quad (10)$$

In this paper, a control pattern means a set of events permitted to occur by the supervisor. A set of control patterns at state $s_i \in S_i$ of SV_i is denoted by $\Pi_i(s_i)$. Each SV_i selects a control pattern $\pi_i \in \Pi_i(s_i)$ based on the current state s_i. Active uncontrollable events for SV_i are always included in the control pattern since they could not be disabled to occur by SV_i. Therefore, the following inclusion is satisfied:

$$\forall \pi_i \in \Pi_i(s_i)\ \ F_i(s_i) \cap \Sigma_i^{uc} \subseteq \pi_i \subseteq F_i(s_i) \subseteq \Sigma, \quad (11)$$

where $F_i(s_i)$ denotes an active event set over Σ at state s_i and is defined as follows:

$$F_i(s_i) = \bigcup_{x\in s_i} F_G(x) \quad (12)$$

The DES receives a net control pattern π, which is the intersection of each supervisor's control pattern, as follows:

$$\pi = \bigcap_{i=1}^{n} \pi_i. \quad (13)$$

An event included in π occurs, a state of the DES changes to a new state and the control pattern is updated.

Next, we show the system model based on the Bellman optimal equation. The system is assumed to be n MDPs and each MDP is defined by a 4-tuple $(S_i, \Pi_i, \mathcal{P}_i, \mathcal{R}_i)$, where

- $\mathcal{P}_i(s_i,\pi_i,s_i')$ is a probability of a transition from state s_i to s_i' when SV_i selects $\pi_i \in \Pi_i(s_i)$,
- $Q_i^*(s_i,\pi_i)$ is a discounted expected total reward in the case that SV_i selects $\pi_i \in \Pi_i(s_i)$ at state s_i and continues to select control patterns optimally, and
- $\mathcal{R}_i(s_i,\pi_i,s_i')$ is an expected reward when SV_i selects $\pi_i \in \Pi_i(s_i)$ at state s_i and makes a transition to state $s_i' \in S_i$.

Each MDP consists of a supervisor SV_i and the DES G. The Bellman optimal equation for each MDP is described as follows:

$$Q_i^*(s_i,\pi_i) = \sum_{s_i'\in S_i} \Bigg[\mathcal{P}_i(s_i,\pi_i,s_i') \\ \times \left(\mathcal{R}_i(s_i,\pi_i,s_i') + \gamma \max_{\pi_i'\in\Pi_i(s_i')} Q_i^*(s_i',\pi_i')\right)\Bigg], \quad (14)$$

where γ denotes a discount rate of rewards ($0 \le \gamma < 1$).

Each supervisor selects a control pattern according to the occurrence of observable events for the supervisor. Therefore, for each SV_i, the following equation holds:

$$\mathcal{P}_i(s_i,\pi_i,s_i') = \sum_{\sigma_i^o\in\pi_i\cap\Sigma_i^o} \mathcal{P}_i^1(s_i,\pi_i,\sigma_i^o)\mathcal{P}_i^2(s_i,\sigma_i^o,s_i'), \quad (15)$$

where

- $\mathcal{P}_i^1(s_i,\pi_i,\sigma_i^o)$ is a probability that SV_i observes the occurrence of an event $\sigma_i^o \in \pi_i \cap \Sigma_i^o$ when SV_i selects $\pi_i \in \Pi_i(s_i)$ at $s_i \in S_i$, and
- $\mathcal{P}_i^2(s_i,\sigma_i^o,s_i')$ is a probability that SV_i makes a transition from state s_i to s_i' by the observed event σ_i^o.

Two additional assumptions for the system is posed in this paper.

1. The DES has a parameter $\eta_i^*(s_i,\sigma_i^o)$ for each $s_i \in S_i$ and $\sigma_i^o \in F_i(s_i) \cap \Sigma_i^o$. Then the following equations hold:

$$\mathcal{P}_i^1(s_i,\pi_i,\sigma_i^o) = \frac{\eta_i^*(s_i,\sigma_i^o)}{\sum_{\sigma_i^{o'}\in\pi_i\cap\Sigma_i^o} \eta_i^*(s_i,\sigma_i^{o'})}, \quad (16)$$

$$\eta_i^*(s_i,\sigma_i^o) > 0, \sum_{\sigma_i^{o'}\in F_i(s_i)\cap\Sigma_i^o} \eta_i^*(s_i,\sigma_i^{o'}) = 1. \quad (17)$$

   An observable event σ_i^o included in the control pattern π_i occurs in the DES, and SV_i observes the event. Then, a probability of the occurrence of the event is given by Eq. (16). In other words, an event occurs in the DES based on a weight of events included in the control pattern and the weight is independent of selected control patterns. The supervisor does not know the true value of η_i^*. This assumption may not be satisfied strictly in many systems. But, if Eq. (16) holds approximately, each supervisor can learn an approximate optimal solution using the proposed method.
2. The reward $\mathcal{R}_i(s_i,\pi_i,s_i')$ consists of two terms as follows:

$$\mathcal{R}_i(s_i,\pi_i,s_i') = \mathcal{R}_i^1(s_i,\pi_i) + \mathcal{R}_i^2(s_i,\sigma_i^o,s_i'), \quad (18)$$

   where

   - $\mathcal{R}_i^1(s_i,\pi_i)$ is an expectation of a reward when SV_i selects π_i at s_i. It depends on the control pattern that SV_i assigns. Intuitively, it represents for the cost to disable controllable events which is not included in the control pattern, and
   - $\mathcal{R}_i^2(s_i,\sigma_i^o,s_i')$ is an expectation of a reward when SV_i observes an event $\sigma_i^o \in \Sigma_i^o$ and makes a transition from s_i to s_i'. Intuitively, it represents for costs by the occurrence of the event and evaluation about the achievement of the task.

By using the above assumptions and Eq. (14), for each SV_i, the following equation is obtained:

$$Q_i^*(s_i,\pi_i) \\ = \mathcal{R}_i^1(s_i,\pi_i) + \sum_{\sigma_i^o\in\pi_i\cap\Sigma_i^o} \frac{\eta_i^*(s_i,\sigma_i^o)}{\sum_{\sigma_i^{o'}\in\pi_i\cap\Sigma_i^o} \eta_i^*(s_i,\sigma_i^{o'})}$$

$$\times \left(\sum_{s_i' \in S_i} \mathcal{P}_i^2(s_i, \sigma_i^o, s_i') \times \left(\mathcal{R}_i^2(s_i, \sigma_i^o, s_i') + \gamma \max_{\pi_i' \in \Pi_i(s_i')} Q_i^*(s_i', \pi_i') \right) \right)$$

$$= \mathcal{R}_i^1(s_i, \pi_i) + \sum_{\sigma_i^o \in \pi_i \cap \Sigma_i^o} \frac{\eta_i^*(s_i, \sigma_i^o)}{\sum_{\sigma_i^{o'} \in \pi_i \cap \Sigma_i^o} \eta_i^*(s_i, \sigma_i^{o'})} T_i^*(s_i, \sigma_i^o), \quad (19)$$

where $T_i^*(s_i, \sigma_i^o)$, defined by the following equation, denotes a discounted expected total reward when SV_i observes σ_i^o at state s_i and selects the control pattern which has the maximum value Q_i^* at the new states:

$$T_i^*(s_i, \sigma_i^o) = \sum_{s_i' \in S_i} \left[\mathcal{P}_i^2(s_i, \sigma_i^o, s_i') \times \left(\mathcal{R}_i^2(s_i, \sigma_i^o, s_i') + \gamma \max_{\pi_i' \in \Pi_i(s_i')} Q_i^*(s_i', \pi_i') \right) \right]. \quad (20)$$

Note that rewards by selecting π_i at s_i are not included in $T^*(s_i, \sigma_i^o)$.

4. Learning of Supervisors

This section proposes an algorithm for learning supervisors in a decentralized manner. In a learning process, an episode means a series of events and states, and starts from an initial state and ends at a terminal state of the DES.

When the state of SV_i is $s_i \in S_i$, SV_i selects a control pattern $\pi_i \in \Pi_i(s_i)$. Several methods are proposed for this selection in literature of reinforcement learning, for example, the ϵ-greedy selection and the Boltzmann selection. By giving n control patterns by n supervisors, the DES receives a net control pattern $\pi = \cap_{i=1}^n \pi_i$. An event $\sigma \in \pi$ occurs in the DES. This occurrence is not affected by supervisors, but restricted by Eq. (16). If an observable event $\sigma_i^o \in \Sigma_i^o$ occurs in the DES, SV_i observes it. At the same time, SV_i obtains two types of rewards. One is evaluation for the control pattern π_i, denoted by r_i^1, and the other is for observation of σ_i^o, denoted by r_i^2. SV_i makes a transition to a new state based on $g_i(s_i, \sigma_i^o)$.

In Eq. (19), Q_i^* is calculated by $\mathcal{R}_i^1$, η_i^*, and T_i^*. In other words, it is possible to estimate Q_i^* indirectly by using $\mathcal{R}_i^1$, η_i^*, and T_i^*. Therefore, three leaning parameters R_i^1, η_i, and T_i are used in the proposed method. SV_i updates them as follows:

$$T_i(s_i, \sigma_i^o) \leftarrow T_i(s_i, \sigma_i^o) + \alpha[r_i^2 + \gamma \max_{\pi_i' \in \Pi_i(s_i')} Q_i(s_i', \pi_i') - T_i(s_i, \sigma_i^o)], \quad (21)$$

$$R_i^1(s_i, \pi_i) \leftarrow R_i^1(s_i, \pi_i) + \beta[r_i^1 - R_i^1(s_i, \pi_i)], \quad (22)$$

$$\eta_i(s_i, \sigma_i^{o'}) \leftarrow \begin{cases} (1-\delta)\, \eta_i(s_i, \sigma_i^{o'}) & \text{for all } \sigma_i^{o'} \in \pi_i \cap \Sigma_i^o \\ \quad (\text{if } \sigma_i^{o'} \neq \sigma_i^o), & \\ \eta_i(s_i, \sigma_i^{o'}) + \delta & \\ \left[\sum_{\sigma_i^{o''} \in \pi_i \cap \Sigma_i^o} \eta_i(s_i, \sigma_i^{o''}) - \eta_i(s_i, \sigma_i^{o'}) \right] & \\ \quad (\text{if } \sigma_i^{o'} = \sigma_i^o), & \end{cases} \quad (23)$$

where α, β, and δ are learning rates. Then, the supervisor updates Q values by using T_i, R_i^1, and η_i as follows:

$$\forall \pi_i' \in \Pi_i(s_i)\ s.t.\ \pi_i' \cap \pi_i \neq \emptyset$$
$$Q_i(s_i, \pi_i') \leftarrow R_i^1(s_i, \pi_i') + \sum_{\sigma_i^{o''} \in \pi_i' \cap \Sigma_i^o} \frac{\eta_i(s_i, \sigma_i^{o''})}{\sum_{\sigma_i^{o'''} \in \pi_i' \cap \Sigma_i^o} \eta_i(s_i, \sigma_i^{o'''})} T_i(s_i, \sigma_i^{o''}). \quad (24)$$

The updates is done for not only the control pattern π_i selected actually, but also control patterns which include an event belonging to π_i. Then SV_i assigns a new control pattern π_i' to the DES based on the current new state of the supervisor. Therefore, the DES receives a new net control pattern π'. Each SV_i updates its control pattern when an observable event for SV_i occurred. The above process is repeated until the DES reaches a terminal state, and an episode ends. By repetition of episodes, learning parameters are updated, and supervisors learn what a control pattern should be selected. Summary of the proposed algorithm is shown in Fig. 2.

In the proposed algorithm, each supervisor learns control patterns so as to maximize the own expected total reward. Both specifications and costs of the DES are considered by introduction of two types of rewards. Moreover, plural Q values are updated for acceleration of the leaning speed by using assumptions based on characteristics of the supervisory control. If the system is under full observa-

1. Initialize T_i, R_i^1 , and η_i of all SV_i.
2. Initialize Q values of all SV_i by Eq. (24).
3. Repeat until any s_i is a terminal state (for each episode):
   a. Initialize a state $s_i \leftarrow x_0$ for all SV_i.
   b. Repeat for each SV_i (for each step of an episode):
      i. Select a control pattern $\pi_i \in \Pi_i(s_i)$ based on the Q_i values by SV_i. (As a result, a net control pattern π is assigned to the DES G.)
      ii. Observe the occurrence of event $\sigma_i^o \in \Sigma_i^o$.
      iii. Acquire rewards r_i^1 and r_i^2.
      iv. Make a transition $s_i \xrightarrow{\sigma_i^o} s_i' (= g_i(s, \sigma_i^o))$ in SV_i.
      v. Update $T_i(s_i, \sigma_i^o)$, $R_i^1(s_i, \pi_i)$, and $\eta_i(s_i, \sigma_i^{o'})$ by Eq. (21), Eq. (22) and Eq. (23) respectively.
      vi. Update the Q_i values by Eq. (24).
      vii. $s_i \leftarrow s_i'$.

Fig. 2 The learning algorithm for decentralized supervisory control.

tion and the reward is given by the sum of the evaluation of marked states and the event disabling cost, the supervisor learns the control patterns which maximizes the language measure [17].

5. Example

We demonstrate the efficiency of the proposed method by the cat and mouse problem. There are 5-rooms partitioned by doors as shown in Fig. 3. Each door is one-way, and used by a cat or a mouse exclusively. A goal is to control doors so as not to encounter a cat and a mouse in the same room simultaneously. A control pattern in this problem means what doors should be closed. There are 2 supervisors. SV_1 can observe the occurrences of events in room 1, room 2, and room 3, that is, $m1$, $m2$, $m3$, $c2$, and $c3$. SV_2 can observe the occurrences of events in room 3, room 4, and room 5, that is, $c1$, $c2$, $c3$, $m2$, and $m3$. SV_1 can control $m1$, $m2$, and $m3$. SV_2 can control $c1$, $c2$, and $c3$. Since $c2$ is uncontrollable for SV_1, SV_1 could not prevent a cat from entering room 3. Similarly, SV_2 could not prevent a mouse from entering room 3. Therefore, cooperation between supervisors is required to satisfy the control specification unless one of supervisors shuts a cat or a mouse into a room.

In the initial state, a mouse is in room 2 and a cat is in room 4. For closing each controllable door, it takes a cost -2. Hence, a reward r_i^1 is given by sum of costs to close doors. Each supervisor acquires a reward $r_i^2 = 1$ when the supervisor observes a cat or a mouse entering a new room. Moreover, observation noise is added to rewards based on the normal distribution whose variance is 0.1. One episode ends when 20 step passed or the cat and the mouse encountered in a room. In the latter case, a reward $r_i^2 = -10$ is given for fail of control. All Q values are initialized by 0. Other parameters are set as follows: $\alpha = \beta = \delta = 0.1$, and $\gamma = 0.9$. The $\epsilon-$greedy selection with $\epsilon = 0.1$ is used to select a control pattern. In the $\epsilon-$greedy selection, the supervisors select the control pattern which has the maximum Q value with probability $1 - \epsilon$, and select another one randomly with probability ϵ.

Figures 4 and 5 show the transition diagrams of the learned supervisors SV_1 and SV_2, respectively. 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.

By the setting $r_i^2 = -10$, each supervisor learns an encounter of the cat and the mouse causes the large negative reward (penalty) and tries to disable the occurrence of events which lead to the encounter. However it takes a cost to close controllable doors. Therefore, each supervisor prefers to leave doors open if the encounter does not occur. In this problem, each supervisor could not observe all occurrences of events, and the current state of the DES is not always determined uniquely. Moreover, as shown in diagrams, the supervisor could not control doors so as not to encounter the cat and the mouse because of uncontrollable events $c2$ for SV_1 and $m2$ for SV_2. For example, the mouse at room 3 may encounter the cat since the event $c2$ is uncontrollable for SV_1. However, as shown in Fig. 5, SV_2 disables $c2$ if the mouse is in room 3 since SV_2 does not desire to get the reward $r_2^2 = -10$. In consequence, the cat can't enter room 3 and the mouse does not encounter the cat. In other words, the cooperation without direct negotiation is achieved between SV_1 and SV_2.

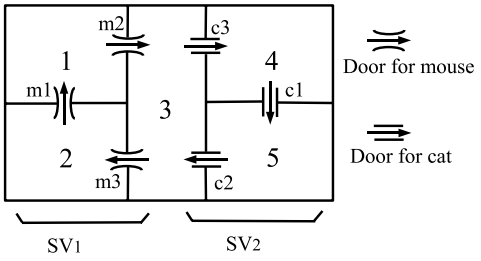

Fig. 3 Maze for cat and mouse with 2 supervisors.

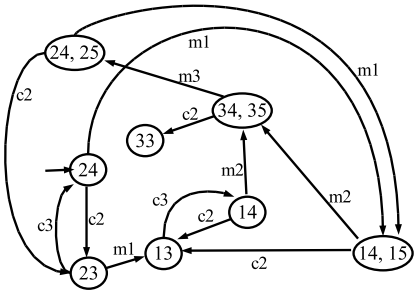

Fig. 4 The transition diagram of the learned supervisor SV_1.

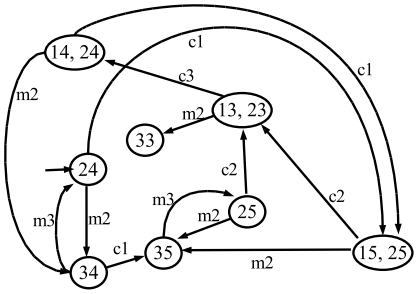

Fig. 5 The transition diagram of the learned supervisor SV_2.

The supervisors controlling doors so as not to encounter a cat and a mouse is synthesized through learning, and the decentralized supervisory control is achieved.

6. Conclusion

In this paper, a decentralized supervisory control method based on reinforcement learning has shown. The proposed method uses a framework of supervisory control, and each supervisor learns assignment of control patterns through acquisition of rewards which reflect the specifications and costs of events under imprecise specifications and uncertain environment.

In the proposed method, a net control pattern is derived by the intersection of control patterns given by each supervisor and negotiation between supervisors is not considered. A decentralized supervisory control which has a different setting is a future work. An introduction of a hierarchical mechanism to the proposed method is also future work.

References

[1] C.G. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, Kluwer Academic Publishers, Boston, 1999.
[2] 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.
[3] 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.
[4] P. Gohari and W.M. Wonham, "On the complexity of supervisory control design in the RW framework," IEEE Trans. Syst. Man Cybern. B, Cybern., vol.30, no.5, pp.643–652, 2000.
[5] Y. Brave and M. Heymann, "On optimal attraction of discrete-event processes," Inf. Sci., vol.67, pp.245–276, 1993.
[6] 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.
[7] 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.
[8] X. Wang and A. Ray, "A language measure for performance evaluation of discrete-event supervisory control systems," Applied Math. Modelling, vol.28, no.9, pp.817–833, 2004.
[9] A. Ray, J. Fu, and C. Lagoa, "Optimal supervisory control of finite state automata," Int. J. Control, vol.77, no.12, pp.1083–1100, 2004.
[10] D.P. Bertsekas and J.N. Tsitsiklis, Neuro-Dynamic Programming, Athena Scientific, Belmont, 1996.
[11] R.S. Sutton and A.G. Barto, Reinforcement Learning, MIT Press, Cambridge, 1998.
[12] T. Yamasaki and T. Ushio, "Supervisory control of partially observed discrete event systems based on a reinforcement learning," Proc. 2003 IEEE SMC, pp.2956–2961, 2003.
[13] T. Yamasaki and T. Ushio, "Decentralized Supervisory Control of Discrete Event Systems based on Reinforcement Learning," IFAC LSS 2004, pp.379–384, 2004.
[14] C.J.C.H. Watkins and P. Dayan, "Q-learning," Mach. Learn., vol.8, pp.279–292, 1992.
[15] K. Rudie and W.M. Wonham, "Think globally, act locally: Decentralized supervisory control," IEEE Trans. Autom. Control, vol.37, no.11, pp.1692–1708, 1992.
[16] R. Cieslak, C. Desclaux, A.S. Fawaz, and P. Varaiya, "Supervisory control of discrete-event processes with partial observations," IEEE Trans. Autom. Control, vol.30, no.3, pp.249–260, 1988.
[17] K. Taniguchi, T. Ushio, and T. Yamasaki, "A reinforcement learning of optimal supervisor based on language measure," IEICE Technical Report (Japanese), CST2004-11, 2004.

Tatsushi Yamasaki received B.E., M.E. and Ph.D. degrees in 1997, 1999, 2003, respectively, from Osaka University. He joined Kwansei Gakuin University as a Contract Assistant in 2002. His research interests include control of discrete event systems and reinforcement learning. He is a member of ISCIE and IEEE.

Toshimitsu Ushio received B.S., M.S. and Ph.D. degrees in 1988, 1982, 1985, respectively, from Kobe University. He was a Research Assistant at the University of California, Berkeley in 1985. From 1986 to 1990, he was a Research Associate at Kobe University, and became a Lecturer at Kobe College in 1990. He joined Osaka University as an Associate Professor in 1994, and is currently a Professor. His research interests include nonlinear oscillation and control of discrete event and hybrid dynamical systems. He is a member of SICE, ISCIE, and IEEE.