Reinforcement Learning of Optimal Supervisor Based on Language Measure

Tatsushi Yamasaki, Kazutaka Taniguchi and Toshimitsu Ushio

Abstract—Recently, Wang and Ray introduced a signed real measure for formal languages, called a language measure, to evaluate performance of strings generated by discrete event systems. They proposed a synthesis method of an optimal supervisor based on the language measure. If exact description of a discrete event system and the specification is not available, a learning-based approach is useful. In this paper, first, we clarify the relationship between the Bellman equation and a performance index of the languages generated by the controlled discrete event systems. Next, using the relationship, we propose a learning method of the optimal supervisor based on reinforcement learning where costs of disabling of events and the evaluation of reaching states are taken into consideration. Finally, by computer simulation, we illustrate an efficiency of the proposed method.

I. INTRODUCTION

The supervisory control initiated by Ramadge and Wonham is a logical control method for discrete event systems (DESs) [1]–[3]. DESs are widely found in various man-made systems, such as transportation systems, communication systems, and operating systems [4]. In the supervisory control, a controller, called a supervisor, assigns the occurrence of controllable events so as to satisfy logical control specifications. They proposed a synthesis method of the optimal supervisor in the sense that the language generated by the controlled system is maximized within given specifications. In the supervisory control, precise descriptions of the specifications and the DESs are required. From the practical point of view, it is not always possible to know the information a priori.

Several researchers have studied optimal control problems of DESs which take into account the cost of events. Kumar and Garg [5] proposed a synthesis method of an optimal supervisor based on network flow techniques. They considered costs of disabling of events and rewards for reaching desired or undesired states. Sengupta and Lafortune [6] show an algorithm to compute an optimal supervisor based on dynamic programming. They considered costs of occurrence and disabling of events, and adopted a worst-case cost as a condition of optimality. Marchand et al. [7] extended the framework of [6] to a partial observation case.

Recently, Wang and Ray introduced a signed real measure, called a language measure, for formal languages [8] [9] [12]. 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. Ray et al. proposed a synthesis method of the optimal supervisor which maximizes the performance index of the language generated by the controlled DES [10] [11]. Phoha et al. applied the supervisory control to model the execution of a software application and quantify the performance based on the language measure [13]. Wang et al. applied the language measure for simulation of behavioral selections of robots in the case that the probability of the occurrence of the events is unknown [14].

Reinforcement learning has been attracted as a learning method [15] [16]. In reinforcement learning, a policy of control is updated based on rewards given from an environment through trial and error. Q-learning [17] is one of the reinforcement learning methods and applied to various problems because of the simplicity of the algorithm and ease of use. It is based on the Bellman equation and a learner obtains the optimal policy in the sense the expected total reward is maximized.

This paper considers a synthesis method of an optimal supervisor with respect to a language measure. In our previous work, we proposed a synthesis method of a supervisor based on reinforcement learning [18] [19]. However, it was not clear the relationship of the optimality between the reinforcement learning and the formal language theory. In this paper, we clarify the relationship between the Bellman equation and a performance index by a language measure. Then, we propose a synthesis method of the supervisor based on the Q-learning where costs of disabling of events and the evaluation of reaching desirable or undesirable states are taken into consideration. Reinforcement learning is used so that implicit specifications are considered and the supervisor can adapt to changing environments. Rewards from the DES represent control specifications and a detail of the specifications is obtained through learning. The proposed method synthesizes the optimal supervisor which maximizes the performance index of the controlled system by the language measure.

This paper is organized as follows. Section II reviews the reinforcement learning and the language measure briefly. Section III shows the relationship between a language measure and the Bellman equation. Section IV proposes a synthesis method of the supervisor based on reinforcement learning. Section V demonstrates the efficiency of the proposed method. Section VI provides the conclusion.
II. PRELIMINARIES

A. 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 and error is effective in the case of an uncertain environment. Moreover, a learner can adapt the policy to changing environment based on rewards autonomously [16]. Q-learning is one of the reinforcement learning algorithms [17]. It updates Q value which is evaluation for state-action pairs. When a learner makes a transition from a current state to a new state by an action and obtains a reward of a set. In the DES G, Let \( |X| = n \) and \( |\Sigma| = m \). Denoted by \( \sigma_i^k \) is the index set of events by which a transition from state \( x_i \) to \( x_k \) occurs, i.e., \( \sigma_i^k = \{ j | \delta(x_i, \sigma_j) = x_k \} \). Denoted by \( \tilde{\sigma_i} \) is the index set of active events at state \( x_i \), i.e., \( \tilde{\sigma_i} = \{ j | \delta(x_i, \sigma_j) \text{ is defined} \} \), \( \mathcal{I} = \{ 1, 2, \ldots, n \} \) is an index set. The language \( L(G, x_i) \) generated by the DES G starting from the state \( x_i \in X \) is defined by

\[
L(G, x_i) = \{ s \in \Sigma^* | \delta(x_i, s) \in X \}. 
\]

The set of all strings which start from the state \( x_i \) and terminate at the state \( x_j \) is defined by

\[
L(x_i, x_j) = \{ s \in \Sigma^* | \delta(x_i, s) = x_j \in X \}. 
\]

The marked state set \( X_m \) is partitioned into a set of desired states \( X^+_m \) and that of undesired states \( X^-_m \), that is, \( X_m = X^+_m \cup X^-_m \). \( X^+_m \cap X^-_m = \emptyset \).

For the purpose of evaluation of each state, a characteristic function \( y : X \rightarrow [1, 1] \) is introduced for all \( i \in \mathcal{I} \) as follows:

\[
y(x_i) = y_i \in \begin{cases} 
0 & \text{if } x_i \notin X_m, \\
0, 1 & \text{if } x_i \in X^+_m, \\
-1, 0 & \text{if } x_i \in X^-_m. 
\end{cases} 
\]

\( Y = [y_1, y_2, \ldots, y_n]^T \) is called a state weighting vector.

An event cost function of the DES G is defined by \( \pi : \Sigma^* \times X \rightarrow [0, 1] \). \( \pi \) satisfies the following conditions for all \( x_i \in X, \sigma_j \in \Sigma, \) and \( s \in \Sigma^* \):

1. \( \pi[\sigma_j | x_i] = \pi_{ij} \in [0, 1), \sum_j \pi_{ij} < 1 \),
2. \( \pi[\sigma_j | x_i] = 0 \) if \( \delta(x_i, \sigma_j) \) is undefined, \( \pi[\epsilon | x_i] = 1 \),
3. \( \pi[\sigma_j, s | x_i] = \pi[\sigma_j | x_i] \pi[s | \delta(x_i, \sigma_j)] \).

A signed real measure is given for the language generated by the DES G [8]–[12]. The signed real measure of \( L(x_i, x_j) \) is defined by:

\[
\mu(L(x_i, x_j)) = \sum_{s \in L(x_i, x_j)} \pi[s | x_i] y(x_j). 
\]

Moreover, the signed real measure of the language \( L(G, x_i) \) is defined by:

\[
\mu(L(G, x_i)) = \sum_{x_j \in X} \mu(L(x_i, x_j)). 
\]

A control action by which a supervisor \( S \) determines the disabling of controllable events \( \sigma_j \in \Sigma^c \) at state \( x_i \) is defined as follows:

\[
dx_{ij} = \begin{cases} 
1 & \text{if } \sigma_j \text{ is disabled at state } x_i, \\
0 & \text{otherwise}. 
\end{cases} 
\]

Denoted by \( \tilde{d}_{ij} \) is the index set of disabling of events at state \( x_i \), i.e., \( \tilde{d}_{ij} = \{ j | d_{ij} \neq 0 \} \), and is called a control pattern. \( D^S(x_i) \) denotes a set of control patterns at state \( x_i \).

A state transition cost \( \pi^S : X \times X \rightarrow [0, 1] \) of the controlled system \( S/G \) is defined as follows:

\[
\pi^S[x_k | x_i] = \pi_{ik}^S = \sum_{j \in \sigma_i^k} \pi[j | x_i] \pi[s | \delta(x_i, \sigma_j)] \text{ if } \sigma_i^k - d_{ij}^S \neq \emptyset, \\
0 \text{ otherwise.} 
\]
\(\Pi^S\) denotes a state transition cost matrix whose \((i, k)\)-th element is \(\pi^S_{ik}\) and is called a \(\Pi^S\)-matrix.

Let \(c^S_{ij}\) be a cost by disabling of the controllable event \(\sigma_j \in \Sigma^*\) which causes a transition from state \(x_i\) to \(x_k\). We abbreviate it to \(c_{ij}\) if \(x_k\) is obvious from the context. An \(n \times m\) matrix \(C = [c_{ij}]\) is called a disabling cost matrix.

A disabling cost characteristic when a supervisor \(S\) disables the occurrence of controllable events at state \(x_i\) is defined as follows:

\[
\xi^S_i = \xi(x_i, d^S_i) = \sum_{j \in d^S_i} c_{ij},
\]

\(\xi^S = [\xi^S_1, \xi^S_2, \ldots, \xi^S_n]^T\) is called a disabling cost characteristic vector of a supervisor \(S\).

The characteristic vector of the controlled DES depends on control patterns given by the supervisor \(S\). A modified characteristic vector by a supervisor \(S\) is defined by

\[
Y^S = [y^S_1, y^S_2, \ldots, y^S_n]^T = Y - \xi^S,
\]

where \(y^S_i = y_i - \xi^S_i\). Then, a performance vector \(\mu^S\) of the controlled system \((S/G)\) is given as follows [10]:

\[
\mu^S = [\mu^S_1, \mu^S_2, \ldots, \mu^S_n]^T = [I - \Pi^S]\bar{y}^S,
\]

where \(\mu^S_i\) is a language measure of \(L(G, x_i)\) under the supervisor \(S\) and represents a performance index at state \(x_i\). It is possible to evaluate the performance of the supervisor quantitatively by the language measure.

III. LANGUAGE MEASURE AND BELLMAN EQUATION

For the controlled system \((S/G)\), the following Bellman equation holds:

\[
V^d(x_i) = P(x_i, d^S_i, x_k) \times [r^*(x_i, d^S_i, x_k) + \gamma V^d(x_k)],
\]

where \(V^d(x_i)\) is a discounted expected total reward at state \(x_i \in X\) under a policy \(d^S_i\) and called a value function, \(P(x_i, d^S_i, x_k)\) is a probability of a transition from state \(x_i\) to \(x_k\) when a supervisor \(S\) assigns a control pattern \(d^S_i \in D^S(x_i)\), \(r^*(x_i, d^S_i, x_k)\) is an expected reward when a transition from state \(x_i\) to \(x_k\) occurs by assigning the control pattern \(d^S_i\), and \(\gamma\) is a discount rate of rewards. In (14), a policy \(d\) is deterministic, and represented by a mapping from each state \(x_i\) to a control pattern \(d^S_i\).

An event which is not included in the assigned control pattern occurs in the DES \(G\). Therefore, the following equation holds:

\[
P(x_i, d^S_i, x_k) = \sum_{j \in \sigma - d^S_i} P_1(x_i, d^S_i, \sigma_j)P_2(x_i, x_j, x_k),
\]

where \(P_1(x_i, d^S_i, \sigma_j)\) is a probability that an event \(\sigma_j (j \in \sigma)\) occurs in the DES \(G\) when the supervisor \(S\) assigns the control pattern \(d^S_i\) at state \(x_i\) and \(P_2(x_i, x_j, x_k)\) is a probability that the DES \(G\) makes a transition from the state \(x_i\) to \(x_k\) when an event \(\sigma_j (j \in \sigma^S)\) occurs.

We assume that the DES \(G\) has a (hidden) parameter \(\tilde{\pi}^*(x_i, \sigma)\) for each state \(x_i\) and event \(\sigma \in \Sigma \cup \sigma^*\) which represents an weight of occurrence of the event and the following equations hold:

\[
P_1(x_i, d^S_i, \sigma) = \sum_{l \in \sigma - d^S_i} \tilde{\pi}^*(x_i, \sigma_l) + \tilde{\pi}^*(x_i, \sigma),
\]

\[
\tilde{\pi}^*(x_i, \sigma_j) \in [0, 1], \quad \tilde{\pi}^*(x_i, \sigma) > 0,
\]

\[
\tilde{\pi}^*(x_i, \sigma_j) = 0 \text{ if } \delta(x_i, \sigma_j) \text{ is undefined},
\]

\[
\sum_{j \in \sigma^S} \tilde{\pi}^*(x_i, \sigma_j) = 1.
\]

We interpret the special event \(\sigma\) as 1-step passage of time without any occurrence of events in the DES \(G\) and acquisition of rewards. It is uncontrollable for the supervisor \(S\) and occurs with probability \(P_1(x_i, d^S_i, \sigma)\) at each state \(x_i\). Therefore, if \(\tilde{\pi}^*(x_i, \sigma)\) is large, the DES stays at the current state with the high possibility. Moreover, we consider a discount rate \(\gamma\) as a function of a state \(x_i\) and a control pattern \(d^S_i\). Thus, we have

\[
\gamma = \gamma(x_i, d^S_i) = \sum_{l \in \sigma - d^S_i} \tilde{\pi}^*(x_i, \sigma_l) + \tilde{\pi}^*(x_i, \sigma),
\]

which implies \(\gamma \in (0, 1)\) by (19).

We define a function \(\pi^S: X \times X \rightarrow [0, 1]\) as follows:

\[
\pi^S(x_k | x_i) = \pi^S_{ik} = \left\{ \begin{array}{ll}
P_1(x_i, d^S_i, \sigma_j)(x_i, d^S_i) & \text{if } \sigma^S - d^S_i \neq \emptyset, \\
0 & \text{otherwise.}
\end{array} \right.
\]

From (16) and (20), we have the following equation:

\[
\pi^S_{ik} = \left\{ \begin{array}{ll}
\sum_{j \in \sigma^S - d^S_i} \tilde{\pi}^*(x_i, \sigma_j) & \text{if } \sigma^S - d^S_i \neq \emptyset, \\
0 & \text{otherwise.}
\end{array} \right.
\]

Let \(\Pi^S\) be a matrix whose \((i, k)\)-th element is \(\pi^S_{ik}\) and call it a \(\Pi^S\)-matrix.

We define the reward \(r^*(x_i, d^S_i, x_k)\) as follows:

\[
r^*(x_i, d^S_i, x_k) = r^*(x_i, d^S_i) = y(x_i) - \xi(x_i, d^S_i),
\]

which means \(r^*\) is based on the evaluation of the current state \(x_i\) and the cost by assigning the control pattern \(d^S_i\).

Next, we show a performance vector is derived from a value function in the Bellman equation.

We define a vector \(R\) as follows:

\[
R = [r^*(x_1, d^S_1), r^*(x_2, d^S_2), \ldots, r^*(x_n, d^S_n)]^T.
\]
If a state transition is deterministic, (14) is transformed into
\[
V^d(x_i) = r^*(x_i, d^S_i, x_k) + \sum_{x_k \in X} \sum_{j \in \sigma_i - d^S_i} P_d(x_i, d^S_i, \sigma_j) \\
\times P_2(x_i, \sigma_j, x_k) (x_i, d^S_i) V^d(x_k) \\
= r^*(x_i, d^S_i) \sum_{x_k \in X} \sum_{j \in \sigma_i - d^S_i} P_d(x_i, d^S_i, \sigma_j) V^d(x_k) \\
= r^*(x_i, d^S_i) + \sum_{x_k \in X} \sum_{j \in \sigma_i - d^S_i} \hat{\pi}^*(x_i, \sigma_j) V^d(x_k).
\]
(25)
(26)
We define a vector \( V \) as
\[
V = [V^d(x_1), V^d(x_2), \ldots, V^d(x_n)]^T.
\]
(27)

Then, the following equation holds:
\[
V = R + \Pi^S V,
\]
(28)
and (28) is transformed as follows:
\[
(I - \Pi^S)^{-1} V = R \\
\leftrightarrow V = (I - \Pi^S)^{-1} R.
\]
(29)
(30)

By considering \( \Pi^S = \Pi^S \) and \( R = Y \), the value function \( V \) of the Bellman equation corresponds to the performance vector \( \mu^S \) defined by (13). It shows that a language measure, which is quantitative evaluation based on the language generated by the controlled system, is derived from the value function of the Bellman equation if the parameters in the value function can be selected in an appropriate way.

IV. LEARNING ALGORITHM OF THE SUPERVISOR

From the result of section III, we will propose the learning method based on the Bellman equation for calculation of the performance vector. If some parameters are unknown, a synthesis of the supervisor by learning is required. In this paper, the supervisor learns a control pattern so as to maximize a performance vector by a method based on Q-learning.

The Bellman optimal equation with regard to \( Q \) values is described as follows [18]:
\[
Q^*(x_i, d^S_i) = \sum_{x_k \in X} P(x_i, d^S_i, x_k) \\
\times \left[ r^*(x_i, d^S_i, x_k) + \gamma \max_{d^S_k} Q^*(x_k, d^S_k) \right],
\]
(31)
where \( Q^*(x_i, d^S_i) \) is a discounted expected total reward when the supervisor \( S \) assigns \( d^S_i \) at state \( x_i \) and continues to assign the optimal control patterns until the controlled behavior reaches a terminal state. If a state transition is deterministic, the Bellman optimal equation is rewritten as follows:
\[
Q^*(x_i, d^S_i) = r^*(x_i, d^S_i) \\
+ \sum_{j \in \sigma_i - d^S_i} \hat{\pi}^*(x_i, \sigma_j) V^*(\delta(x_i, \sigma_j)),
\]
(32)
where for the state \( x_k = \delta(x_i, \sigma_j) \in X \),
\[
V^*(x_k) = \max_{d^S_k \in D^S(x_k)} Q^*(x_k, d^S_k).
\]
(33)

In the DES \( G \), an event \( e \) occurs with the probability given by (16). If the event is included in the active event set at \( x_i \) and enabled by the assigned control pattern \( d^S_i \), then, \( \sigma = \sigma_j (j \in \sigma_i - d^S_i) \), then the DES \( G \) makes a transition from state \( x_i \) to \( x_k \) by the occurrence of the event and the supervisor \( S \) acquires a reward \( r \). If the special event \( e = \hat{\sigma} \) occurs, the DES stays at the current state without acquisition of rewards.

By (32) and (33), \( Q^* \) is updated by \( Q^*, \hat{r}^*, \) and \( \hat{\pi}^* \). We update \( \hat{r}^* \) and \( \hat{\pi}^* \) by the following equations:
\[
r'(x_i, d^S_i) \leftarrow r'(x_i, d^S_i) + \alpha [r - r'(x_i, d^S_i)],
\]
(34)
and, for all \( \sigma' = \sigma_l (l \in \hat{\sigma}_i - d^S_i) \) and \( \sigma' = \hat{\sigma} \),
\[
\hat{\pi}'(x_i, \sigma') \leftarrow \begin{cases} 
(1 - \beta) \hat{\pi}(x_i, \sigma') \\ \hat{\pi}(x_i, \sigma') + \beta \left( \sum_{m \in \sigma_l - d^S_i} \hat{\pi}(x_i, \sigma_m) \\
+ \hat{\pi}(x_i, \hat{\sigma}) - \hat{\pi}(x_i, \sigma') \right) 
\end{cases}
\]
(35)
where \( \hat{r}' \) and \( \hat{\pi}' \) are estimated values of \( r^* \) and \( \hat{\pi}^* \), respectively, and both \( \alpha \in [0,1] \) and \( \beta \in [0,1] \) are learning rates. If the special event \( e = \hat{\sigma} \) occurs, (34) is not applied since it does not affect the expected reward. By using \( r^* \) and \( \hat{\pi}^* \), we can update several \( Q \) values of all control patterns which do not disable all events permitted by the assigned control pattern \( d^S_i \). In other words, for all \( d^S_i \in D^S(x_i) \) which satisfies \( (\hat{\sigma}_i - d^S_i) \cap (\hat{\sigma}_i - d^S_i) \neq \emptyset \), \( Q \) values are updated as follows:
\[
Q(x_i, d^S_i) \leftarrow r'(x_i, d^S_i) \\
+ \sum_{j \in \hat{\sigma}_i - d^S_i} \hat{\pi}'(x_i, \sigma_j) V'(\delta(x_i, \sigma_j)),
\]
(36)
where \( Q(x_i, d^S_i) \) is the estimated value of \( Q^*(x_i, d^S_i) \) and for the state \( x_k = \delta(x_i, \sigma_j) \in X \),
\[
V'(x_k) = \max_{d^S_k \in D^S(x_k)} Q^*(x_k, d^S_k).
\]
(37)

Let \( \hat{d}_{ij} \in [0,1] \) be a probability that the supervisor \( S \) disables the event \( \sigma_j \) at state \( x_i \). The supervisor \( S \) assigns a control pattern according to \( \hat{d}_{ij} \). Let \( \hat{d}^S_i \) be a control pattern which maximizes the \( Q \) value at state \( x_i \) defined by
\[
\hat{d}^S_i = \arg \max_{d^S_i \in D^S(x_i)} Q(x_i, d^S_i) \in D^S(x_i).
\]
(38)

By using \( \hat{d}^S_i \), \( \hat{d}_{ij} \) is updated as follows:
\[
\hat{d}_{ij} \leftarrow \begin{cases} 
\hat{d}_{ij} + \lambda (1 - \hat{d}_{ij}) & \text{if } j \in \hat{d}^S_i, \\
\hat{d}_{ij} + \lambda (0 - \hat{d}_{ij}) & \text{if } j \notin \hat{d}^S_i,
\end{cases}
\]
(39)
where \( \lambda \in [0,1] \) is a learning rate. The above equation means that the supervisor \( S \) increases (resp. decreases) the probability of disabling of events if the event is (resp. is not) included in \( \hat{d}^S_i \). We summarize the learning algorithm in Fig. 2.
V. EXAMPLE

We consider a dining philosopher problem used in [8]. The DES \( G \) of the problem is represented by the automaton in Fig. 3. There are two philosophers denoted by \( P_1 \) and \( P_2 \), and two forks denoted by \( F_1 \) and \( F_2 \). Table I shows the definition of each event \( \sigma_1, \sigma_2, \ldots, \) and \( \sigma_6 \), where \( \sigma_1, \ldots, \) and \( \sigma_4 \) are controllable events and \( \sigma_5 \) and \( \sigma_6 \) are uncontrollable events.

The initial state \( 1 \) means both \( P_1 \) and \( P_2 \) are thinking, and marked state \( 10 \) (resp. \( 11 \)) means \( P_1 \) (resp. \( P_2 \)) is thinking after eating and the other is thinking. State 8 (resp. 9), which is a deadlock state, means \( P_1 \) (resp. \( P_2 \)) has 1 fork. In the proposed algorithm, the supervisor \( S \) acquires a reward after transition and can’t acquire it at the deadlock state. Therefore, we add a dummy state 12 in simulation so as to avoid such a situation. By the occurrence of uncontrollable event \( \sigma_7 \), the supervisor \( S \) makes a transition from state 8 or 9 to the dummy state 12.

We set the state weighting vector \( Y = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -0.5 \ -0.5 \ 1]^T \). In this example, the disabling cost is not considered. Therefore, a reward \( r \) is decided by \( y(x_i) \). For states which do not have a transition to the dummy state, we set \( \hat{\pi}(x_i, \bar{\sigma}) = 0.04 \) and \( \hat{\pi}(x_i, \sigma_j) = 0.96/|\bar{\sigma}| \) for all \( j \in \bar{\sigma} \). Event \( \sigma_7 \) occurs with probability 1. Such information is unknown for the supervisor \( S \) and the supervisor \( S \) obtains them through learning. Learning rates are set as follows: \( \alpha = 0.7 \), \( \beta = 0.01 \), and \( \lambda = 0.2 \). Each \( d_{ij} \) is initialized by \( 0.5 \) and all \( Q \) values are initialized by \( 0 \). Each \( \pi'(x_i, \sigma) \) is initialized by \( 1/(|\bar{\sigma}| + 1) \) for all \( \sigma = \sigma_j (j \in \bar{\sigma}) \) and \( \sigma = \bar{\sigma} \). We adopt the \( \epsilon \)-greedy selection with \( \epsilon = 0.1 \) when the supervisor \( S \) assigns a control pattern. The supervisor \( S \) assigns the control pattern based on \( d_{ij} \) with probability \( 1 - \epsilon \) and assigns another control pattern randomly with probability \( \epsilon \) so that the supervisor can explore various control patterns. The supervisor \( S \) proceeds learning by repetition of episodes. One episode ends by 20-steps or reaching a deadlock state.

The control objective is as follows:

1) Increase a possibility that philosophers reach state 10 or 11.

2) Decrease a possibility that philosophers reach state 8 or 9.

Fig. 4 shows a result of learning of the controlled system \( S/G \) by computer simulation. Dashed lines show the disabled
The number of control patterns increases exponentially as the number of events increases. Therefore, the improvement of computational cost is future work. An extension to a partial observation case is also future work.

VI. CONCLUSION

We showed a value function in the Bellman equation corresponds to a performance vector obtained by the language measure. We proposed a learning method of control patterns based on reinforcement learning, which synthesizes the optimal supervisor with regard to the language measure. The language measure provides the quantitative evaluation of the supervisor based on the language generated by the controlled system. The proposed method is applicable for design of the supervisor under implicit specifications and changing environment by using reinforcement learning. We applied the proposed method to a dining philosopher problem and showed that the optimal supervisor with regard to the language measure is obtained.

REFERENCES