Proceedings of the
44th IEEE Conference on Decision and Control, and
the European Control Conference 2005
Seville, Spain, December 12-15, 2005

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.

T. Yamasaki is with the School of Science and Technology, Kwansei Gakuin Universiy, Sanda-shi, Hyogo, 669-1337 Japan tatsushi@ksc.kwansei.ac.jp

K. Taniguchi and T. Ushio are with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, Osaka, 560-8531 Japan taniguti@hopf.sys.es.osaka-u.ac.jp, ushio@sys.es.osaka-u.ac.jp

0-7803-9568-9/05/$20.00 ©2005 IEEE

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 x to a new state x' by an action a and obtains a reward $r(x,a)$, Q value is updated as follows:

$$\begin{aligned} Q(x,a) \leftarrow\ & Q(x,a) + \alpha\big[r(x,a) \\ & + \gamma \max_{a'} Q(x',a') - Q(x,a)\big], \end{aligned} \tag{1}$$

where $Q(x,a)$ is an estimation of the expected discounted total rewards when a learner takes an action a at a state x, $\alpha \in [0,1]$ is a learning rate, and $\gamma \in [0,1]$ is a discounted rate of rewards. The Q value converges with probability 1 to a true value if α decays appropriately and the number of updates of the Q value goes to the infinity. Q-learning is applicable if the environment is modeled by a Markov Decision Process (MDP).

B. Supervisory control and language measure

In the supervisory control, the supervisor controls the occurrence of controllable events so as to satisfy logical control specifications of the DES [1]–[3]. A DES G controlled by a supervisor S is illustrated as Fig. 1.

The DES G is modeled by a 5-tuple $(X,\ \Sigma,\ \delta,\ x_1,\ X_m)$, where X is a set of states, Σ is a finite set of events, $\delta : \Sigma \times X \to X$ is a state transition function, $x_1 \in X$ is an initial state, and $X_m \subseteq X$ is a set of marked states. Σ^* is a set of all finite strings over Σ including the empty string ϵ. δ is extended into a function $\delta : X \times \Sigma^* \to X$ as follows:

$$\delta(x,\epsilon) = x, \tag{2}$$

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

$$\delta(x,s\sigma) = \delta(\delta(x,s),\sigma). \tag{3}$$

Σ is partitioned into a set of controllable events Σ^c and a set of uncontrollable events Σ^{uc}, that is, $\Sigma = \Sigma^c \cup \Sigma^{uc}$, $\Sigma^c \cap \Sigma^{uc} = \emptyset$. We use the notation $|\cdot|$ to indicate the cardinality

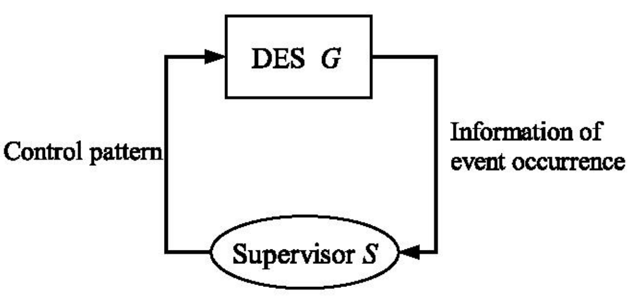

Fig. 1. Discrete event system controlled by the supervisor

of a set. In the DES G, Let $|X| = n$ and $|\Sigma| = m$. Denoted by σ_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 $\hat{\sigma}_i$ is the index set of active events at state x_i, *i.e.*, $\hat{\sigma}_i = \{\, j \,|\, \delta(x_i,\sigma_j) \text{ is defined}\}$. $\mathcal{I} = \{1,\ 2,\ \ldots,\ n\}$ 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

$$L(G,x_i) = \{s \in \Sigma^* | \delta(x_i,s) \in X\}. \tag{4}$$

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

$$L(x_i,x_j) = \{s \in \Sigma^* \mid \delta(x_i,s) = x_j \in X\}. \tag{5}$$

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 $y : X \to [1,1]$ is introduced for all $i \in \mathcal{I}$ as follows:

$$y(x_i) = y_i \in \begin{cases} \{\ 0\ \} & \text{if } x_i \notin X_m, \\ (\ 0,\ 1\ ] & \text{if } x_i \in X_m^+, \\ [\ -1,\ 0\ ) & \text{if } x_i \in X_m^-. \end{cases} \tag{6}$$

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

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

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

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:

$$\mu(L(x_i,x_j)) = \sum_{s \in L(x_i,x_j)} \tilde{\pi}[s \,|\, x_i] y(x_j). \tag{7}$$

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

$$\mu(L(G,x_i)) = \sum_{x_j \in X} \mu(L(x_i,x_j)). \tag{8}$$

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:

$$d_{ij}^S = \begin{cases} 1 & \text{if } \sigma_j \text{ is disabled at state } x_i, \\ 0 & \text{otherwise.} \end{cases} \tag{9}$$

Denoted by d_i^S is the index set of disabling of events at state x_i, *i.e.*, $d_i^S = \{\, j \mid d_{ij}^S = 1 \,\}$, and is called a control pattern. $D^S(x_i)$ denotes a set of control patterns at state x_i.

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

$$\begin{aligned} \pi^S[x_k|x_i] &= \pi_{ik}^S \\ &= \begin{cases} \displaystyle\sum_{j \in \sigma_i^k - d_i^S} \tilde{\pi}[\sigma_j|x_i] & \text{if } \sigma_i^k - d_i^S \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{10}$$

Π^S denotes a state transition cost matrix whose (i,k)-th element is π_{ik}^S and is called a Π^S-matrix.

Let c_{ij}^k be a cost by disabling of the controllable event $\sigma_j \in \Sigma^c$ 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:

$$\xi_i^S = \xi(x_i, d_i^S) = \sum_{j \in d_i^S} c_{ij}. \tag{11}$$

$\xi^S = [\, \xi_1^S,\ \xi_2^S,\ \ldots,\ \xi_n^S \,]^T$ 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

$$Y^S = [\, y_1^S,\ y_2^S,\ \ldots,\ y_n^S \,]^T = Y - \xi^S, \tag{12}$$

where $y_i^S = y_i - \xi_i^S$. Then, a performance vector μ^S of the controlled system S/G is given as follows [10]:

$$\begin{aligned} \mu^S &= [\, \mu_1^S,\ \mu_2^S,\ \ldots,\ \mu_n^S \,]^T \\ &= [\, I - \Pi^S \,]^{-1} Y^S, \end{aligned} \tag{13}$$

where μ_i^S 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:

$$\begin{aligned} V^d(x_i) &= \sum_{x_k \in X} P(x_i, d_i^S, x_k) \\ &\quad \times \left[r^*(x_i, d_i^S, x_k) + \gamma V^d(x_k) \right], \end{aligned} \tag{14}$$

where $V^d(x_i)$ is a discounted expected total reward at state $x_i \in X$ under a policy d and called a value function, $P(x_i, d_i^S, x_k)$ is a probability of a transition from state x_i to x_k when a supervisor S assigns a control pattern $d_i^S \in D^S(x_i)$, $r^*(x_i, d_i^S, x_k)$ is an expected reward when a state transition from x_i to x_k occurs by assigning the control pattern d_i^S, and γ 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_i^S.

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

$$P(x_i, d_i^S, x_k) = \sum_{j \in \hat{\sigma}_i - d_i^S} P_1(x_i, d_i^S, \sigma_j) P_2(x_i, \sigma_j, x_k), \tag{15}$$

where $P_1(x_i, d_i^S, \sigma_j)$ is a probability that an event σ_j ($j \in \hat{\sigma}_i$) occurs in the DES G when the supervisor S assigns the control pattern d_i^S at state x_i and $P_2(x_i, \sigma_j, x_k)$ is a probability that the DES G makes a transition from the state x_i to x_k when an event σ_j ($j \in \sigma_i^k$) occurs.

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

$$P_1(x_i, d_i^S, \sigma) = \frac{\tilde{\pi}^*(x_i, \sigma)}{\displaystyle\sum_{l \in \hat{\sigma}_i - d_i^S} \tilde{\pi}^*(x_i, \sigma_l) + \tilde{\pi}^*(x_i, \bar{\sigma})}, \tag{16}$$

$$\tilde{\pi}^*(x_i, \sigma_j) \in [0, 1),\ \tilde{\pi}^*(x_i, \bar{\sigma}) > 0, \tag{17}$$

$$\tilde{\pi}^*(x_i, \sigma_j) = 0 \text{ if } \delta(x_i, \sigma_j) \text{ is undefined}, \tag{18}$$

$$\sum_{j \in \hat{\sigma}_i} \tilde{\pi}^*(x_i, \sigma_j) + \tilde{\pi}^*(x_i, \bar{\sigma}) = 1. \tag{19}$$

We interpret the special event $\bar{\sigma}$ 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 $P_1(x_i, d_i^S, \bar{\sigma})$ at each state x_i. Therefore, if $\tilde{\pi}^*(x_i, \bar{\sigma})$ is large, the DES stays at the current state with the high possibility. Moreover, we consider a discount rate γ is a function of a state x_i and a control pattern d_i^S. Thus, we have

$$\begin{aligned} \gamma &= \gamma(x_i, d_i^S) \\ &= \sum_{l \in \hat{\sigma}_i - d_i^S} \tilde{\pi}^*(x_i, \sigma_l) + \tilde{\pi}^*(x_i, \bar{\sigma}), \end{aligned} \tag{20}$$

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

We define a function $\pi^{*S} : X \times X \to [0, 1)$ as follows:

$$\begin{aligned} \pi^{*S}(x_k \mid x_i) &= \pi_{ik}^{*S} \\ &= \begin{cases} \displaystyle\sum_{j \in \sigma_i^k - d_i^S} P_1(x_i, d_i^S, \sigma_j) \gamma(x_i, d_i^S) & \\ \qquad\qquad \text{if } \ \sigma_i^k - d_i^S \neq \emptyset, & \\ 0 \qquad \text{otherwise.} & \end{cases} \end{aligned} \tag{21}$$

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

$$\pi_{ik}^{*S} = \begin{cases} \displaystyle\sum_{j \in \sigma_i^k - d_i^S} \tilde{\pi}^*(x_i, \sigma_j) & \text{if } \ \sigma_i^k - d_i^S \neq \emptyset, \\ 0 & \text{otherwise.} \end{cases} \tag{22}$$

Let Π^{*S} be a matrix whose (i,k)-th element is π_{ik}^{*S} and call it a Π^{*S}-matrix.

We define the reward $r^*(x_i, d_i^S, x_k)$ as follows:

$$r^*(x_i, d_i^S, x_k) = r^*(x_i, d_i^S) = y(x_i) - \xi(x_i, d_i^S), \tag{23}$$

which means r^* is based on the evaluation of the current state x_i and the cost by assigning the control pattern d_i^S.

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

We define a vector R as follows:

$$R = \left[r^*(x_1, d_1^S), r^*(x_2, d_2^S), \ldots, r^*(x_n, d_n^S) \right]^T. \tag{24}$$

If a state transition is deterministic, (14) is transformed into

$$\begin{aligned} V^d(x_i) &= r^*(x_i, d_i^S, x_k) + \sum_{x_k \in X} \sum_{j \in \hat{\sigma}_i - d_i^S} P_1(x_i, d_i^S, \sigma_j) \\ &\quad \times P_2(x_i, \sigma_j, x_k) \gamma(x_i, d_i^S) V^d(x_k) \\ &= r^*(x_i, d_i^S) \\ &\quad + \sum_{x_k \in X} \sum_{j \in \sigma_i^k - d_i^S} \tilde{\pi}^*(x_i, \sigma_j) V^d(x_k) \qquad (25) \\ &= r^*(x_i, d_i^S) + \sum_{x_k \in X} \pi_{ik}^{*S} V^d(x_k). \qquad (26) \end{aligned}$$

We define a vector V as

$$V = \left[V^d(x_1), V^d(x_2), \ldots, V^d(x_n)\right]^T. \qquad (27)$$

Then, the following equation holds:

$$V = R + \Pi^{*S} V, \qquad (28)$$

and (28) is transformed as follows:

$$\begin{aligned} &(I - \Pi^{*S})V = R \qquad (29) \\ \Longleftrightarrow\ &V = (I - \Pi^{*S})^{-1} R. \qquad (30) \end{aligned}$$

By considering $\Pi^{*S} = \Pi^S$ and $R = Y$, the value function V of the Bellman equation corresponds to the performance vector μ^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]:

$$\begin{aligned} Q^*(x_i, d_i^S) &= \sum_{x_k \in X} P(x_i, d_i^S, x_k) \\ &\quad \times \left[r^*(x_i, d_i^S, x_k) + \gamma \max_{d_k^S} Q^*(x_k, d_k^S) \right], \qquad (31) \end{aligned}$$

where $Q^*(x_i, d_i^S)$ is a discounted expected total reward when the supervisor S assigns $d_i^S \in D^S(x_i)$ at state $x_i \in X$ 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:

$$\begin{aligned} Q^*(x_i, d_i^S) &= r^*(x_i, d_i^S) \\ &\quad + \sum_{j \in \hat{\sigma}_i - d_i^S} \tilde{\pi}^*(x_i, \sigma_j) V^*(\delta(x_i, \sigma_j)), \qquad (32) \end{aligned}$$

where for the state $x_k = \delta(x_i, \sigma_j) \in X$,

$$V^*(x_k) = \max_{d_k^S \in D^S(x_k)} Q^*(x_k, d_k^S). \qquad (33)$$

In the DES G, an event σ 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_i^S, that is, $\sigma = \sigma_j (j \in \hat{\sigma}_i - d_i^S)$, 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 $\sigma = \bar{\sigma}$ occurs, the DES stays at the current state without acquisition of rewards.

By (32) and (33), Q^* is updated by Q^*, r^*, and $\tilde{\pi}^*$. We update r' and $\tilde{\pi}'$ by the following equations:

$$r'(x_i, d_i^S) \leftarrow r'(x_i, d_i^S) + \alpha[r - r'(x_i, d_i^S)], \qquad (34)$$

and, for all $\sigma' = \sigma_l$ $(l \in \hat{\sigma}_i - d_i^S)$ and $\sigma' = \bar{\sigma}$,

$$\tilde{\pi}'(x_i, \sigma') \leftarrow \begin{cases} (1-\beta)\tilde{\pi}'(x_i, \sigma') \\ \qquad \text{if} \quad \sigma' \neq \sigma, \\ \tilde{\pi}'(x_i, \sigma') + \beta \left[\sum_{m \in \hat{\sigma}_i - d_i^S} \tilde{\pi}'(x_i, \sigma_m) \right. \\ \qquad \left. + \tilde{\pi}'(x_i, \bar{\sigma}) - \tilde{\pi}'(x_i, \sigma') \right] \\ \qquad \text{if} \quad \sigma' = \sigma, \end{cases} \qquad (35)$$

where r' and $\tilde{\pi}'$ are estimated values of r^* and $\tilde{\pi}^*$, respectively, and both $\alpha \in [0, 1]$ and $\beta \in [0, 1]$ are learning rates. If the special event $\bar{\sigma}$ occurs, (34) is not applied since it does not affect the expected reward. By using r^* and $\tilde{\pi}^*$, we can update several Q values of all control patterns which do not disable all events permitted by the assigned control pattern d_i^S. In other words, for all $d_i^{S'} \in D^S(x_i)$ which satisfies $(\hat{\sigma}_i - d_i^{S'}) \cap (\hat{\sigma}_i - d_i^S) \neq \emptyset$, Q values are updated as follows:

$$\begin{aligned} Q(x_i, d_i^{S'}) &\leftarrow r'(x_i, d_i^{S'}) \\ &\quad + \sum_{j \in \hat{\sigma}_i - d_i^{S'}} \tilde{\pi}'(x_i, \sigma_j) V'(\delta(x_i, \sigma_j)), \qquad (36) \end{aligned}$$

where $Q(x_i, d_i^S)$ is the estimated value of $Q^*(x_i, d_i^S)$ and for the state $x_k = \delta(x_i, \sigma_j) \in X$,

$$V'(x_k) = \max_{d_k^S \in D^S(x_k)} Q^*(x_k, d_k^S). \qquad (37)$$

Let $\tilde{d}_{ij} \in [0, 1]$ be a probability that the supervisor S disables the event σ_j at state x_i. The supervisor S assigns a control pattern according to $\tilde{d}_{ij}$. Let $\hat{d}_i^S$ be a control pattern which maximizes the Q value at state x_i defined by

$$\hat{d}_i^S = \arg \max_{d_i^S \in D^S(x_i)} Q(x_i, d_i^S) \in D^S(x_i). \qquad (38)$$

By using $\hat{d}_i^S$, $\tilde{d}_{ij}$ is updated as follows:

$$\tilde{d}_{ij} \leftarrow \begin{cases} \tilde{d}_{ij} + \lambda(1 - \tilde{d}_{ij}) & \text{if} \quad j \in \hat{d}_i^S, \\ \tilde{d}_{ij} + \lambda(0 - \tilde{d}_{ij}) & \text{if} \quad j \notin \hat{d}_i^S, \end{cases} \qquad (39)$$

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 $\hat{d}_i^S$. We summarize the learning algorithm in Fig. 2.

1) Initialize $r'(x_i, d_i^S)$ and $\tilde{\pi}'(x_i, \sigma)$ at each state.
2) Calculate the initial Q value at each state by (36).
3) Repeat (for each episode):
   a) $x_i \leftarrow initial\ state\ x_1$.
   b) Repeat until x_i is a terminal state (for each step of episode) :
      i) Assign a control pattern $d_i^S \in D^S(x_i)$ based on $\tilde{d}_{ij}$.
      ii) Observe the occurrence of event σ and state transition $x_i \xrightarrow{\sigma} x_k$ in the DES G.
      iii) Acquire a reward r and update $r'(x_i, d_i^S)$ by (34) if $\sigma \neq \bar{\sigma}$.
      iv) Update $\tilde{\pi}'(x_i, \sigma)$ by (35).
      v) Update the Q values for all $d_i^{S'} \in D^S(x_i)$ s.t. $(\hat{\sigma}_i - d_i^{S'}) \cap (\hat{\sigma}_i - d_i^S) \neq \emptyset$ by (36).
      vi) Calculate $\hat{d}_i^S$ by (38) and update the probability $\tilde{d}_{ij}$ by (39).
      vii) $x_i \leftarrow x_k$.

Fig. 2. Proposed algorithm

TABLE I
EVENT DEFINITION FOR THE DINING PHILOSOPHERS

Event	Description
σ_1	P_1 picks up F_1 from the table
σ_2	P_1 picks up F_2 from the table
σ_3	P_2 picks up F_1 from the table
σ_4	P_2 picks up F_2 from the table
σ_5	P_1 places F_1 and F_2 on the table
σ_6	P_2 places F_1 and F_2 on the table

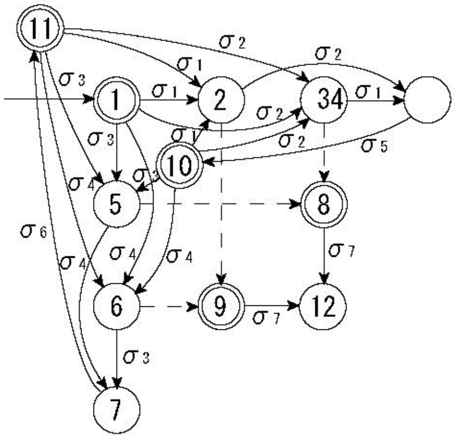

Fig. 4. Automaton of the controlled system

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 σ_6, where $\sigma_1, \ldots,$ and σ_4 are controllable events and σ_5 and σ_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

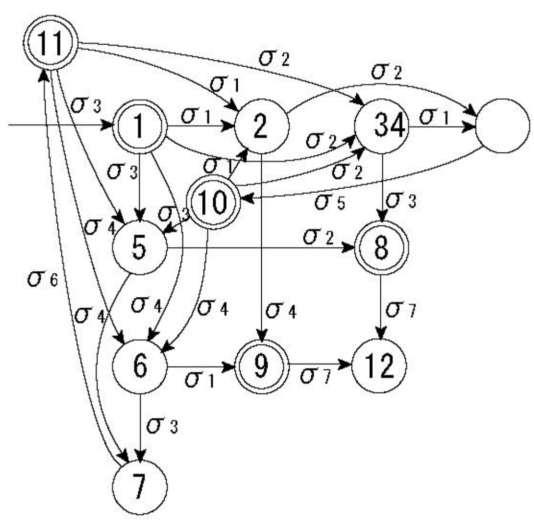

Fig. 3. Automaton of dining philosophers

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 σ_7, the supervisor S makes a transition from state 8 or 9 to the dummy state 12.

We set the state weighting vector $Y = [1\ 0\ 0\ 0\ 0\ 0\ 0\ -0.5\ -0.5\ 1\ ]^T$. 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 $\tilde{\pi}^*(x_i, \bar{\sigma}) = 0.04$ and $\tilde{\pi}^*(x_i, \sigma_j) = 0.96/|\hat{\sigma}_i|$ for all $j \in \hat{\sigma}_i$. Event σ_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 $\tilde{d}_{ij}$ is initialized by 0.5 and all Q values are initialized by 0. Each $\tilde{\pi}'(x_i, \sigma)$ is initialized by $1/(|\hat{\sigma}_i| + 1)$ for all $\sigma = \sigma_j (j \in \hat{\sigma}_i)$ and $\sigma = \bar{\sigma}$. We adopt the ϵ-greedy selection with $\epsilon = 0.1$ when the supervisor S assigns a control pattern. The supervisor S assigns the control pattern based on $\tilde{d}_{ij}$ with probability $1-\epsilon$ and assigns another control pattern randomly with probability ϵ 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

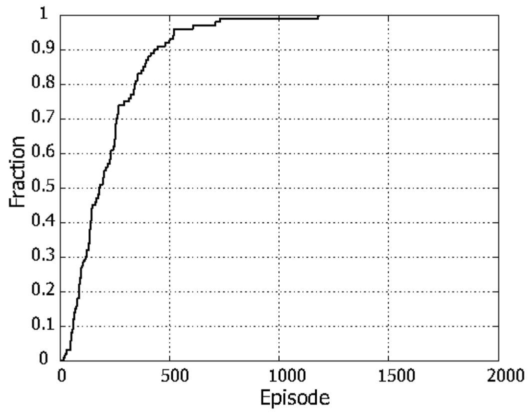

Fig. 5. Relationship between the number of episodes and the fraction what the supervisor selects the optimal control pattern

events by the supervisor S. The supervisor S prevents transitions to state 8 and 9 from occurring. The supervisor S is an optimal supervisor theoretically obtained by [8]. Thus, the supervisor S learned the optimal control pattern.

Next, we show the learning curve of the proposed method. Fig. 5 shows the relationship between the number of episodes and the fraction that the supervisor found the optimal control pattern given by Fig. 4. The supervisor learns the optimal control pattern by experience of many episodes.

In [8], a theoretical value of the performance vector at the initial state is $\mu_1^S = 1.7933$. In our simulation, the Q value at state 1, which is corresponding to μ_1^S, converged to the value after about 8000 episodes.

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.

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.

REFERENCES

[1] 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.

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

[3] P. J. Ramadge and W. M. Wonham, "The control of discrete event systems," *Proc. of IEEE*, vol. 77, no. 1, pp. 81–98, 1989.

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

[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] 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.

[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] X. Wang and A. Ray, "Signed Real Measure of Regular Languages," *American Control Conference*, Anchorage, pp. 3937–3942, 2002.

[9] X. Wang and A. Ray, "A language measure for performance evaluation of discrete-event supervisory control systems," *Applied Mathematical Modelling*, vol. 28, no. 9, pp. 817–833, 2004.

[10] 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.

[11] J. Fu, A. Ray, and C. M. Lagoa, "Unconstrained Optimal Control of Regular Languages," *Automatica*, vol. 40, pp. 639–646, 2004.

[12] A. Ray, V. V. Phoha, and S. Phoha, Eds., *Quantitative Measure for Discrete Event Supervisory Control*, Springer, 2005.

[13] V. V. Phoha, A. U. Nadgar, A. Ray, and S. Phoha, "Supervisory control of software systems," *IEEE Trans. on Computers*, vol. 53, no. 9, pp. 1187–1199, 2004.

[14] X. Wang, J. Fu, P. Lee, and A. Ray, "Robot Behavioral Selection Using Discrete Event Language Measure," *American Control Conference*, Boston, pp. 5126–5131, 2004.

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

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

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

[18] T. Yamasaki and T. Ushio, "Supervisory control of partially observed discrete event systems based on a reinforcement learning," *Proc. of the 2003 IEEE International Conference on Systems, Man & Cybern.*, pp. 2956–2961, 2003.

[19] T. Yamasaki and T. Ushio, "Decentralized Supervisory Control of Discrete Event Systems based on Reinforcement Learning," *Proc. of 10th IFAC/IFORS/IMACS/IFIP Symposium on Large Scale Systems: Theory and Application*, pp. 379–384, 2004.