I. System Model

Definition 1: We represent a probabilistic discrete event system (DES) as a labeled Markov decision process (MDP). A DES is a tuple $M = (S, E, \mathcal{E}, P_T, P_E, s_{init}, AP, L)$, where S is a finite set of states; E is a finite set of events; $\mathcal{E} : S \to E$ is a mapping that maps each state to the set of feasible events at the state; $P_T : S \times S \times E \to [0, 1]$ is a transition probability such that $\sum_{s' \in S} P_T(s'|s, e) = 1$ for any state $s \in S$ and any event $e \in \mathcal{E}(s)$ and $P_T(s'|s, e) = 0$ for any $e \notin \mathcal{E}(s)$; $P_E : E \times S \times 2^E \to [0, 1]$ is the probability that an event occurs under a subset $\pi \in \mathcal{E}(s)$ of events allowed to occur at the state $s \in S$ such that $\Sigma_{e \in \pi} P_E(e|s, \pi) = 1$ and we call the subset the control pattern; for any $(s', s, \pi) \in S \times S \times 2^E$, we define the probability $P : S \times S \times 2^E \to [0, 1]$ such that $P(s'|s, \pi) = \sum_{e \in \pi} P_E(e|s, \pi) P_T(s'|s, e)$ and $\Sigma_{s' \in S} P(s'|s, \pi) = 1$; $s_{init} \in S$ is the initial state; AP is a finite set of atomic propositions; and $L : S \times E \times S \to 2^{AP}$ is a labeling function that assigns a set of atomic propositions to each transition $(s, e, s') \in S \times E \times S$. We assume that E can be partitioned into the set of controllable events E_c and the set of uncontrollable events E_{uc} such that $E_c \cup E_{uc} = E$ and $E_c \cap E_{uc} = \emptyset$. Note that each event e occurs probabilistically depending on only the current state and the subset of feasible events at the state given by a controller.

In the DES M, an infinite path starting from a state $s_0 \in S$ is defined as a sequence $\rho = s_0 e_0 s_1 \ldots \in S(ES)^\omega$ such that $P_T(s_{i+1}|s_i, e_i) > 0$ for any $i \in \mathbb{N}_0$, where $\mathbb{N}_0$ is the set of natural numbers including zero A finite path is a finite sequence in $S(ES)^*$. In addition, we sometimes represent ρ as ρ_{init} to emphasize that ρ starts from $s_0 = s_{init}$. For a path $\rho = s_0 e_0 s_1 \ldots$, we define the corresponding labeled path $L(\rho) = L(s_0, e_0, s_1) L(s_1, e_1, s_2) \ldots \in (2^{AP})^\omega$. $InfPath^M$ (resp., $FinPath^M$) is defined as the set of infinite (resp., finite) paths starting from $s_0 = s_{init}$ in the DES M. For each finite path ρ, $last(\rho)$ denotes its last state.

We define the supervisor as a controller for the DES that restricts the behaviors of the DES to satisfy a given specification.

Definition 2: For the DES M, a supervisor $SV : FinPath^M \to 2^E$ is defined as a mapping that maps each finite path to a set of allowed events at the finite path and we call the set the control pattern. In the following, the supervisor we consider is *state-based*, namely for any $\rho \in FinPath^M$, $SV(\rho) = SV(last(\rho))$. Note that the relation $E_{uc} \subset SV(\rho) \subset E$ holds for any $\rho \in FinPath^M$.

Definition 3: A sink state in state set X of an augmented tLDBA $\bar{B}_\varphi = (\bar{X}, \bar{x}_{init}, \bar{\Sigma}, \bar{\delta}, \bar{\mathcal{F}})$ is defined as a state such that there exist no accepting transition of $\bar{B}_\varphi$ that is accessible from the state. We denote the set of sink states as $SinkSet$.

Definition 4: Given an augmented tLDBA $\bar{B}_\varphi = (\bar{X}, \bar{x}_{init}, \bar{\Sigma}, \bar{\delta}, \bar{\mathcal{F}})$ and a DES M, a tuple $M \otimes \bar{B}_\varphi = M^\otimes = (S^\otimes, E^\otimes, \mathcal{E}^\otimes, s^\otimes_{init}, P^\otimes_T, P^\otimes_E, \delta^\otimes, \mathcal{F}^\otimes)$ is a product DES, where $S^\otimes = S \times \bar{X}$ is the finite set of states and we represent s and $\bar{x}$ corresponding with $s^\otimes = (s, \bar{x}) \in S^\otimes$ as $[\![s^\otimes]\!]_s$ and $[\![s^\otimes]\!]_q$, respectively, $E^\otimes = E$ is the finite set of events, $\mathcal{E}^\otimes : S^\otimes \to 2^{E^\otimes}$ is the mapping defined as $\mathcal{E}^\otimes((s, \bar{x})) = \mathcal{E}(s)$, $s^\otimes_{init} = (s_{init}, \bar{x}_{init})$ is the initial states, $P^\otimes_T : S^\otimes \times S^\otimes \times E^\otimes \to [0, 1]$ is the transition probability defined as

$$P^\otimes_T(s^{\otimes\prime}|s^\otimes, e) = \begin{cases} P_T(s'|s, e) & \text{if } (\bar{x}, L((s, e, s')), \bar{x}') \in \bar{\delta}, \\ 0 & \text{otherwise,} \end{cases}$$

$P^\otimes_E : E^\otimes \times S^\otimes \times 2^{E^\otimes} \to [0, 1]$ is the probability of the occurrence of the event defined as $P^\otimes_E(e|s^\otimes, \pi) = P_E(e|s, \pi)$, $\delta^\otimes = \{(s^\otimes, e, s^{\otimes\prime}) \in S^\otimes \times E^\otimes \times S^\otimes; P^\otimes_T(s^{\otimes\prime}|s^\otimes, e) > 0\}$ is the set of transitions, and $\mathcal{F}^\otimes = \{\bar{F}^\otimes_1, \ldots, \bar{F}^\otimes_n\}$ is the acceptance condition, where $\bar{F}^\otimes_i = \{((s, \bar{x}), e, (s', \bar{x}')) \in \delta^\otimes \ ; \ (\bar{x}, L(s, e, s'), \bar{x}') \in \bar{F}_i\}$ for each $i \in \{1, \ldots, n\}$.

II. Objective function for Control patterns

From the view point of reinforcement learning, the DES can be interpreted as the environment controlled by the supervisor and the supervisor can be interpreted as the agent. We introduce the two following assumptions.

1) The relative frequency of occurrence of each event does not depend on the control pattern.
2) We define a reward function $\mathcal{R} : S \times 2^E \times E \times S \to \mathbb{R}$ and the reward $\mathcal{R}$ can be decomposed into $\mathcal{R}_1$ and $\mathcal{R}_2$. The first reward $\mathcal{R}_1 : S \times 2^E \to \mathbb{R}$ is a reward that is determined by the control pattern selected by the supervisor, which depends on only the control pattern and the current state. The second reward $\mathcal{R}_2 : S \times E \times S \to \mathbb{R}$ is a reward that is determined by the occurrence of an event and the corresponding state transition. For any $(s, \pi, e, s') \in S \times 2^E \times E \times S$, we then have

$$\mathcal{R}(s, \pi, e, s') = \mathcal{R}_1(s, \pi) + \mathcal{R}_2(s, e, s'). \quad (1)$$

Under the above assumptions, we have the following *Bellman optimality equation*.

$$\begin{aligned} Q^*(s, \pi) = & \sum_{s' \in S} P(s'|s, \pi) \\ & \left\{ \mathcal{R}(s, \pi, e, s') + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s', \pi') \right\} \\ = & \sum_{s' \in S} \sum_{e \in \pi} P_E e|s, \pi) P_T(s'|s, e) \\ & \left\{ \mathcal{R}_1(s, \pi) + \mathcal{R}_2(s, e, s') + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s', \pi') \right\} \\ = & \mathcal{R}_1(s, \pi) + \sum_{e \in \pi} P_E(e|s, \pi) \sum_{s' \in S} P_T(s'|s, e) \\ & \left\{ \mathcal{R}_2(s'|s, e) + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s', \pi') \right\}, \end{aligned} \quad (2)$$

where $\gamma \in [0, 1)$.

We introduce the following function. $T^* : S \times E \to \mathbb{R}$ such that

$$T^*(s, e) = \sum_{s' \in S} P_T(s'|s, e) \left\{ \mathcal{R}_2(s'|s, e) + \gamma \max_{\pi' \in 2^{\mathcal{E}(s')}} Q^*(s', \pi') \right\}. \quad (3)$$

We then have

$$Q^*(s,\pi) = \mathcal{R}_1(s,\pi) + \sum_{e\in\pi} P_E(e|s,\pi)T^*(s,e). \quad (4)$$

Definition 5: We define an optimal supervisor SV^* as follows. For any state $s \in S$,

$$SV^*(s) = \pi^* \in \arg\max_{\pi\in\mathcal{E}(s)} Q^*(s,\pi). \quad (5)$$

Definition 6: The two reward functions $\mathcal{R}_1 : S^\otimes \times 2^{E^\otimes} \to \mathbb{R}$ and $\mathcal{R}_2 : S^\otimes \times E^\otimes \times S^\otimes \to \mathbb{R}$ are defined as follows.

$$\mathcal{R}_1(s^\otimes,\pi) = r_{n1}(|E| - |\pi|), \quad (6)$$

where $|E|$ means number of elements in the set E and r_{n1} is a negative value.

$$\mathcal{R}_2(s^\otimes,e,s^{\otimes\prime}) = \begin{cases} r_p \text{ if } \exists i \in \{1,\dots,n\},\ (s^\otimes,e,s^{\otimes\prime}) \in \bar{F}_i^\otimes, \\ r_{n2} \text{ if } [\![s^{\otimes\prime}]\!]_q \in SinkSet, \\ 0 \quad \text{otherwise}, \end{cases} \quad (7)$$

III. Learning Algorithm

We make the supervisor learn how to give the control patterns to satisfy an LTL specification while keeping costs associated with prohibited events low. We use Q-learning to estimate the function T^*. We then use Bayesian inference to robustly estimate the probability P_E. For the inference, we model P_E as Categorical distribution as $p^k_{s,\pi,e}$, where $p^k_{s,\pi,e}$ represents the estimated probability of $P_E(e|s,\pi)$ at the time step k and the prior distribution $\phi^k_{s,\pi}$ for the distribution of the parameter of $p^k_{s,\pi,e}$ is defined as Dirichlet.

To reflect the events prohibition by the supervisor on the estimated probability of the occurrence of allowed events, we introduce the function $RestProb : (0,1)^{|E|} \times 2^E \to [0,1]^{|E|}$ defined as

$$RestProb(\phi_{s,\pi},\pi)_i = \begin{cases} \dfrac{\phi^i_{s,\pi}}{\sum_{e^j\in\pi}\phi^j_{s,\pi}} & \text{if } e^i \in \pi, \\ 0 & \text{otherwise}, \end{cases} \quad (8)$$

where $e^1,\dots,e^{|E|}$ are the elements of the event set, $\phi^i_{s,\pi}$ is the i-th element of $\phi_{s,\pi}$ and $RestProb(\phi_{s,\pi},\pi)_i$ is the i-th element of $RestProb(\phi_{s,\pi},\pi)$.

We denote the probability vector of events occurrence at the time step k as $p^k_{s,\pi} = (p^k_{s,\pi,e^1},\dots,p^k_{s,\pi,e^{|E|}})$, where $s \in S$ and $\pi \in \mathcal{E}(s)$ is the state and the control pattern at the time step k. Let $n^k_{s,\pi,e}$ be the number of the occurrence of the event $e \in E$ up to the time step k under the state $s \in S$ and the control pattern $\pi \in \mathcal{E}(s)$, let $n^k_{s,\pi}$ denote $(n^k_{s,\pi,e_1},\dots,n^k_{s,\pi,e_{|E|}})$, and let $\bar{p}^k_{s,\pi}$ denote the expected value of $p^k_{s,\pi}$. The overall procedure of the inference is shown in Algorithm 1.

Under the estimation of P_E, we use TD-learning to estimate Q^* with the TD-error defined as $\mathcal{R}_1(s^\otimes,\pi) + \sum_{e\in\pi} p_{s^\otimes,\pi,e}T(s^\otimes,e) - Q(s^\otimes,\pi)$.

We show the all procedure of learning algorithm in Algorithm 2.

Algorithm 1 P_E inference.

Input: the event occurrence count $n^k_{s,\pi}$, a threshold $\xi^k_{s,\pi}$ for $p^k_{s,\pi}$
Output: the posterior distribution $p^k_{s,\pi}$

repeat
  $\phi^k_{s,\pi} \sim Dir(\cdot|n^k_{s,\pi})$
  $p^k_{s,\pi} = RestProb(\phi^k_{s,\pi},\pi)$
until $||p^k_{s,\pi} - \bar{p}^k_{s,\pi}||_1 < \xi^k_{s,\pi}$

Algorithm 2 RL-based synthesis of a supervisor satisfying the given LTL specification.

Input: LTL formula φ, DES M
Output: optimal supervisor SV^* on the product DES $M^\otimes$

Convert φ into tLDBA B_φ.
Augment B_φ to $\bar{B}_\varphi$.
Construct the product DES $M^\otimes$ of M and $\bar{B}_\varphi$.
Initialize $T : S^\otimes \times E^\otimes \to \mathbb{R}$.
Initialize $Q : S^\otimes \times 2^{E^\otimes} \to \mathbb{R}$.
Initialize $n : S^\otimes \times 2^{E^\otimes} \times E^\otimes \to \mathbb{R}$.
initialize $\xi : S^\otimes \times 2^{E^\otimes} \to \mathbb{R}$.
Initialize episode length L.
while Q is not converged **do**
  $s^\otimes \leftarrow (s_{init},(x_{init},\mathbf{0}))$.
  $t \leftarrow 0$
  while $t < L$ and $[\![s^\otimes]\!]_q \notin SinkSet$ **do**
    Choose the control pattern $\pi \in \mathcal{E}(s^\otimes)$ by the supervisor SV.
    Observe the occurrence of the event $e \in E$.
    Observe the next state $s^{\otimes\prime}$.
    $T(s^\otimes,e) \leftarrow (1-\alpha)T(s^\otimes,e) + \alpha\{\mathcal{R}_2(s^\otimes,e,s^{\otimes\prime}) + \gamma \max_{\pi'\in 2^{\mathcal{E}(s^{\otimes\prime})}} Q(s^{\otimes\prime},\pi')\}$
    $n(s^\otimes,\pi,e) \leftarrow n(s^\otimes,\pi,e) + 1$
    Obtain $p_{s^\otimes,\pi}$ from n and ξ by the P_E inference.
    $Q(s^\otimes,\pi) = (1-\beta)Q(s^\otimes,\pi) + \beta\{\mathcal{R}_1(s^\otimes,\pi) + \sum_{e\in\pi} p_{s^\otimes,\pi,e}T(s^\otimes,e)\}$
    $s^\otimes \leftarrow s^{\otimes\prime}$
    $t \leftarrow t+1$
    Update $\xi(s^\otimes,\pi)$
  end while
end while

IV. Example

We evaluate the algorithm by the maze of the cat and the mouse shown in Fig. 1. At the beginning, we define the settings for the example. The corresponding DES is as follows. The state set is $S = \{(s^{cat},s^{mouse}); s^{cat},s^{mouse} \in \{s_0,s_1,s_2,s_3\}\}$. The set of events (to open the corresponding door) is $E = \{m_0,m_1,m_2,m_3,c_0,c_1,c_2,c_3\}$, where $E_c = \{m_0,m_1,m_2,m_3,c_0,c_1,c_2\}$ and $E_{uc} = \{c_3\}$ and $\mathcal{E}(s) = E$ for any $s \in S$. The initial state is $s_{init} = (s_0,s_2)$. If the door of the room with the cat (resp., mouse) opens, the cat (resp., mouse) moves, with probability 0.95, to the room next to the room with it where the door is open or stays in the same room with probability 0.05. Otherwise, the

cat (resp., mouse) stays in the same room with probability 1. The labeling function is

$$L((s,a,s')) = \begin{cases} \{a\} & \text{if } s'_c = s_1, \\ \{b\} & \text{if } s'_m = s_1, \\ \{c\} & \text{if } s'_c = s'_m, \\ \emptyset & \text{otherwise,} \end{cases}$$

where s'_c and s'_m is the next room where the cat and the mouse is, respectively, i.e., $s' = (s'_c, s'_m)$.

In the example, we want the supervisor to learn to give control patterns satisfying that the cat and the mouse take the food in the room 1 (s_1) avoiding they come across. This is formally specified by the following LTL formula.

$$\varphi = \mathbf{GF}a \wedge \mathbf{GF}b \wedge \mathbf{G}\neg c.$$

The tLDBA $B_\varphi = (X, x_{init}, \Sigma, \delta, \mathcal{F})$ corresponding to φ is shown in Fig. 2. B_φ has the acceptance condition of two accepting sets.

We use ε-greedy policy and gradually reduce ε to 0 to learn an optimal supervisor asymptotically. We set the rewards $r_p = 10$, $r_{n1} = -0.1.-0.5, and -1$, and $r_{n2} = -100$; the epsilon greedy parameter $\varepsilon = \frac{1}{\sqrt{episode}}$, where $episode$ is the number of the current episode; and the discount factor $\gamma = 0.99$. $\xi^k_{s^\otimes,\pi}$ is initially set to 1 and changes to 0.6 during 1/3 to 2/3 of all episodes and to 0.3 after 2/3 of all episodes for any $(s^\otimes, \pi) \in S^\otimes \times 2^{E^\otimes}$. The learning rate α and β vary in accordance with *the Robbins-Monro condition*.

Figs. 3, 4, and 5 show the estimated optimal state value function at the initial state $V(s^\otimes_{init})$ with $r_{n1} = -0.1.-0.5$, and -1, respectively, for each episode when learning 5000 iterations and 15000 episodes by the algorithm 2. Fig. 6, 7, and 8 shows the average reward from $\mathcal{R}_2$ and the average cost from $\mathcal{R}_1$ with $r_{n1} = -0.1.-0.5$, and -1, respectively, of 5000 iteration and 1000 episodes by the supervisor obtained from the learning.

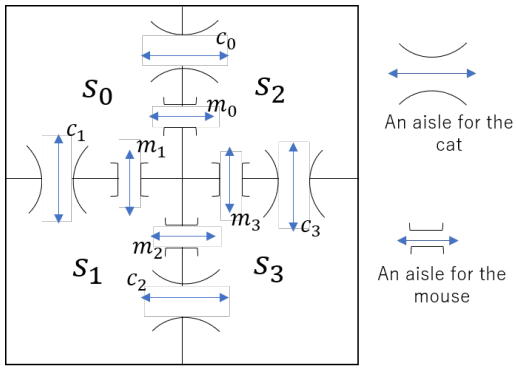

Fig. 1. The maze of the cat and the mouse. the initial state of the cat and the mouse is s_0 and s_2, respectively. the food for them is in the room 1 (s_1).

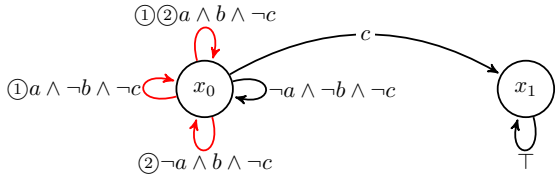

Fig. 2. The tLDBA recognizing the LTL formula $\mathbf{GF}a \wedge \mathbf{GF}b \wedge \mathbf{G}\neg c$, where the initial state is x_0. Red arcs are accepting transitions that are numbered in accordance with the accepting sets they belong to, e.g., ①$a \wedge \neg b \wedge \neg c$ means the transition labeled by it belongs to the accepting set F_1.

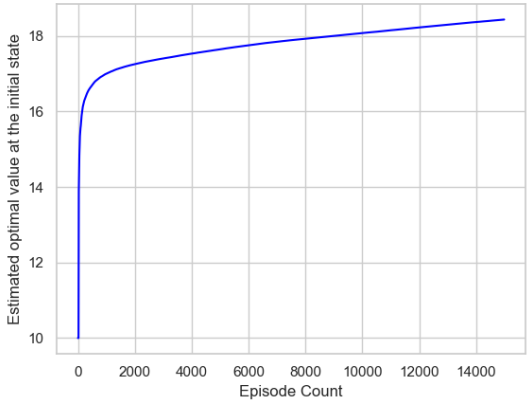

Fig. 3. the estimated optimal state value function at the initial state $V(s^\otimes_{init})$ with $r_{n1} = -0.1$ when using the algorithm 2.

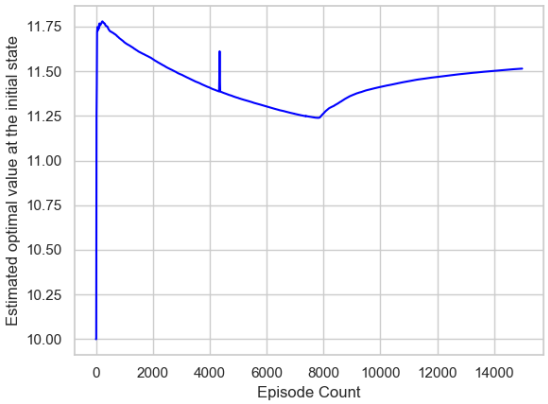

Fig. 4. the estimated optimal state value function at the initial state $V(s^\otimes_{init})$ with $r_{n1} = -0.5$ when using the algorithm 2.

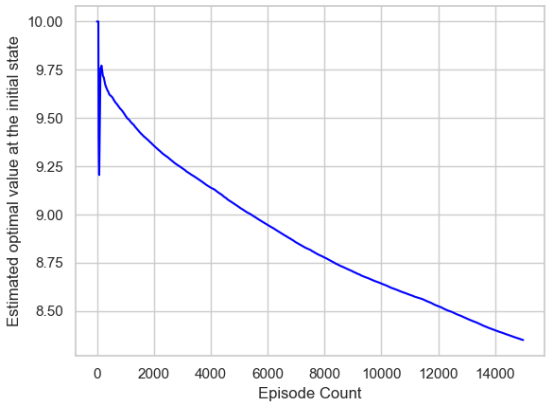

Fig. 5. the estimated optimal state value function at the initial state $V(s^{\otimes}_{init})$ with $r_{n1} = -1$ when using the algorithm 2.

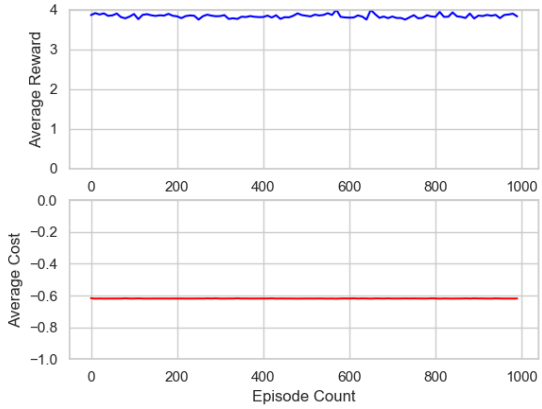

Fig. 6. The average reward and average cost by the supervisor obtained from the learning with $r_{n1} = -0.1$.

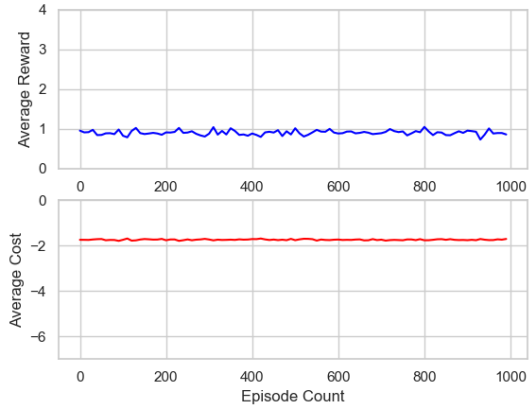

Fig. 8. The average reward and average cost by the supervisor obtained from the learning with $r_{n1} = -1$.

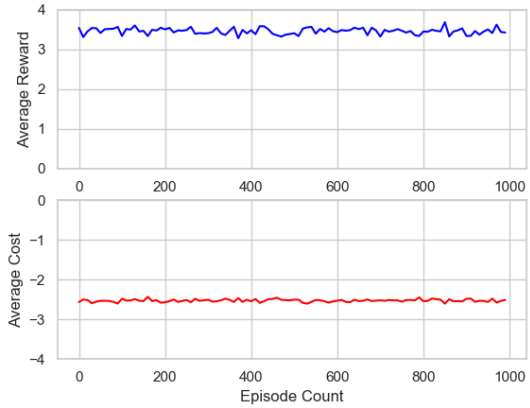

Fig. 7. The average reward and average cost by the supervisor obtained from the learning with $r_{n1} = -0.5$.