

Supervisory Control of Partially Observed Quantitative Discrete Event Systems for Fixed-Initial-Credit Energy Problem

Sasinee PRUEKPRASERT^{†*a)}, Nonmember and Toshimitsu USHIO^{††b)}, Fellow

SUMMARY This paper studies the supervisory control of partially observed quantitative discrete event systems (DESs) under the fixed-initial-credit energy objective. A quantitative DES is modeled by a weighted automaton whose event set is partitioned into a controllable event set and an uncontrollable event set. Partial observation is modeled by a mapping from each event and state of the DES to the corresponding masked event and masked state that are observed by a supervisor. The supervisor controls the DES by disabling or enabling any controllable event for the current state of the DES, based on the observed sequences of masked states and masked events. We model the control process as a two-player game played between the supervisor and the DES. The DES aims to execute the events so that its energy level drops below zero, while the supervisor aims to maintain the energy level above zero. We show that the proposed problem is reducible to finding a winning strategy in a turn-based reachability game.

key words: supervisory control, discrete event system, partial observation, optimal control, energy game

1. Introduction

The supervisory control, which was introduced by Ramadge and Wonham in [1], is a formal approach to the design of a controller for a discrete event systems (DES). In [1], a DES is modeled by an automaton that spontaneously generates events that are partitioned into controllable and uncontrollable events. A supervisor observes the sequence of events generated by the DES, and then control it by disabling any of the controllable events at the current state of the DES. The supervisory control problem is to design a supervisor such that the language generated by the controlled DES satisfies a control specification, i.e., a given target language. Controller design problems can be formulated as two-player games played between the controller and the system. In [2], the control problem under budget constraints was considered under a two-player game setting. In [3], the minimum attention controller synthesis for omega-regular objectives was studied using a two-player game automaton.

The supervisor may not be able to clearly observe some events or some states due to the lack of sensors. As a result, the supervisor may not be able to explicitly determine

the current state of the system. The framework of supervisory control under partial observation was introduced in [4], where the set of events is partitioned into the set of observable and unobservable ones. In real world, sensors can be classified into 2 kinds: state-based sensors (i.e., location sensors) and event-based sensors (i.e., touch sensors). In [5], Takai et al. proposed another framework for partial observability by introducing mappings from each event and each state to the corresponding masked event and masked state, respectively. Then, the supervisor determines the set of possible current states of the DES and controls it based on the observed masked events and masked states.

Energy games are two-player games played on weighted graphs where player-1 aims to maintain the energy of the system in a given range, and player-2 aims to prevent player-1 from achieving her goal. This class of games is applicable in the design of resource-constrained reactive systems, for example, an automatic lawn mower system with rechargeable battery [9]. In [10], a polynomial-time algorithm for deciding the strategy of player-1 in energy games under full observation was proposed. The energy game under partial observation with fixed initial credit is shown to be in ACK-complete [11]. However, the game with unknown initial credit is undecidable [12].

In this paper, we study the supervisory control of partially observed quantitative DESs under the energy objective where the initial credit energy is fixed [12]. We model the control of the DES using a two-player game played between the supervisor and the DES on a weighted automaton. The DES aims to execute the events so that its energy level drops below zero after a finite number of events occur. On the other hand, the supervisor aims to maintain the energy level above zero. The fixed-initial-credit energy problem is to compute a supervisor under which the controlled DES contains no deadlock and the energy level of the DES never goes below zero. We show that the proposed problem is reducible to finding a winning strategy in a turn-based reachability game [12], [13]. The preliminary version of this paper was presented in [14].

The rest of the paper is organized as follows. Section 2 introduces quantitative DESs and provides the basic notations. Section 3 introduces supervisory control under partial observation based on two-player game setting. Section 4 formulates the control problem and proposes algorithms. Section 5 provides an application example of a path planning robot problem that is a fundamental problem in mobile robots. Finally, Sect. 6 presents the conclusions.

Manuscript received August 29, 2016.

Manuscript revised January 21, 2017.

Manuscript publicized March 7, 2017.

[†]The author is with the National Institute of Informatics, Tokyo, 101-8430 Japan.

^{††}The author is with the Graduate School of Engineering Science, Osaka University, Toyonaka-shi, 560-8531 Japan.

*She was with the Graduate School of Engineering Science, Osaka University, when the current work was done.

a) E-mail: sasinee@hopf.sys.es.osaka-u.ac.jp

b) E-mail: ushio@sys.es.osaka-u.ac.jp

DOI: 10.1587/transinf.2016FOP0008

2. Quantitative Discrete Event Systems

In this paper, \mathbb{N} denotes the set of natural numbers including zero, and \mathbb{Z} denotes the set of integers.

We consider a quantitative DES modeled by a weighted automaton $G = \langle X, \Sigma, \delta, x_{G0}, w \rangle$, where X is a finite set of states, Σ is a finite set of events, $\delta \subseteq X \times \Sigma \times X$ is a set of transition relations, $x_{G0} \in X$ is the initial state, and $w : \delta \rightarrow \mathbb{Z}$ is a function that assigns a weight to each transition. The event set Σ is partitioned into two disjoint sets: the uncontrollable events set Σ_u and the controllable events set Σ_c . Likewise, transitions in δ is partitioned into a set of controllable transitions $\delta_c = \delta \cap (X \times \Sigma_c \times X)$ and a set of uncontrollable transitions $\delta_u = \delta \setminus \delta_c$. At each state $x \in X$, an event $\sigma \in \Sigma$ is called an *active event* if there exists an out-going transition $(x, \sigma, x') \in \delta$. Moreover, the transition $(x, \sigma, x') \in \delta$ is said to be an *active transition* at the state x . Let $\Sigma(x) = \{\sigma \in \Sigma \mid \exists x' \in X, (x, \sigma, x') \in \delta\}$ be the set of active events at the state x . A state $x \in X$ such that $\Sigma(x) = \emptyset$ is called a *deadlock state*.

The DES G repeatedly executes one of the active events at its current state. Starting from the initial state x_{G0} , the DES changes its states according to any of the active transitions that are labeled by the generated events. The behavior of the DES is represented by the generated sequence, called the *run*, of its executed events and system states. Formally, a *run* is a finite sequence $r = x_0 \sigma_1 x_1 \dots x_n \in X(\Sigma X)^*$ such that $(x_i, \sigma_{i+1}, x_{i+1}) \in \delta$ for each $i \in \{0, 1, \dots, n-1\}$. Let $last(r)$ denote the last state x_n of the run r . Namely, x_n is the current state of the DES after the run r is generated. The run r is called a *cycle* if $x_0 = x_n$ (i.e., the DES state returns the state x_0), and is called a *deadlock run* if $\Sigma(x_n) = \emptyset$ (i.e., the DES enters a state with no active events and halts). For each state $x \in X$, $Run(G, x) = \{x_0 \sigma_1 x_1 \dots x_n \mid x_0 = x \text{ and } (x_i, \sigma_{i+1}, x_{i+1}) \in \delta, \forall i \in \{0, 1, \dots, n-1\}\}$ is the set of all runs generated by the DES G starting from x . Let $Run(G) = Run(G, x_{G0})$.

3. The Control under Partial Observation

We consider partially observed DESs whose states or events cannot be completely observed by a supervisor. The partially observed DESs studied in this paper is based on the framework proposed in [5]. Since the supervisor may not be able to determine the current state of the DES, the supervisor controls the DES based on the set of possible candidates of the current state of the DES.

Let Y be the set of masked states and Λ be the set of masked events [5]. A surjective function $M_Y : X \rightarrow Y$ (resp. $M_\Lambda : \Sigma \rightarrow \Lambda$) maps each state (resp. event) to its masked state (resp. masked event). A supervisor cannot detect the states of the DES and the executed events, but their masked states and masked events. We assume that the masked state of the initial state is $M_Y(x_{G0}) = y_{G0}$. An observation function $M_O : \bigcup_{x \in X} Run(G, x) \rightarrow Y(\Lambda Y)^*$ is defined as follows: for each run $r = x_0 \sigma_1 x_1 \dots \sigma_n x_n \in \bigcup_{x \in X} Run(G, x)$,

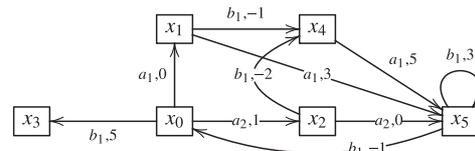


Fig. 1 A partially observed DES, where x_0 is the initial state, $\Sigma_u = \{a_1, a_2\}$, $\Sigma_c = \{b_1\}$, $M_Y(x_0) = M_Y(x_3) = y_0$, $M_Y(x_1) = M_Y(x_2) = y_1$, $M_Y(x_4) = M_Y(x_5) = y_2$, $M_\Lambda(a_1) = M_\Lambda(a_2) = a$, and $M_\Lambda(b_1) = b$. The label of each transition represents the corresponding event and weight.

$$M_O(r) = \begin{cases} M_Y(x_0), & \text{if } r = x_0, \\ M_O(x_0 \sigma_1 \dots x_{n-1}) M_\Lambda(\sigma_n) M_Y(x_n) & \text{otherwise.} \end{cases}$$

The function M_O maps each generated run to the sequence of masked states and masked events, called the *masked run*, that is observed by the supervisor. For the DES in Fig. 1, we have $M_O(x_0 a_1 x_1) = M_O(x_0 a_2 x_2) = y_0 a y_1$. Therefore, if the supervisor observes the sequence $y_0 a y_1$, the set of possible current states of the DES is $\{x_1, x_2\}$. For each set of runs $R \subseteq Run(G)$, let $M_O(R) = \{M_O(r) \mid r \in R\}$ be the set of all masked runs of R .

A supervisor observes the generated masked runs, and controls the DES by disabling some controllable events at the current state of the system. A set of controllable events $\gamma \subseteq \Sigma_c$ is called a *control pattern*. Let $\Gamma = 2^{\Sigma_c}$ be the set of all control patterns. If the DES generates a run r and the supervisor observes the masked run $M_O(r)$ and assigns a control pattern $\gamma \in \Gamma$, then the events included in γ are disabled at the state $last(r)$ in the controlled DES.

We consider the control process as a two-player game played between the supervisor and the DES. A strategy of the supervisor (resp. the DES) is a function $\pi_S : M_O(Run(G)) \rightarrow \Gamma$ (resp. $\pi_D : Run(G) \times \Gamma \rightarrow \delta$). We impose the following conditions on the strategy π_D : for each run $r \in Run(G)$ and each control pattern $\gamma \in \Gamma$ such that $\Sigma(last(r)) \setminus \gamma \neq \emptyset$, if $\pi_D(r, \gamma) = (x, \sigma, x')$, then $x = last(r)$ and $\sigma \in \Sigma \setminus \gamma$. In other words, the strategy π_D selects an active transition at the current state that is enabled by the supervisor if it exists.

For each generated run r , the supervisor first selects the control pattern $\gamma = \pi_S(M_O(r))$, then the DES executes the transitions $(last(r), \sigma, x') = \pi_D(r, \gamma)$. This process is repeated so that the set of runs generated under the pair of strategies (π_S, π_D) is $Play(G, \pi_S, \pi_D) = \{x_0 \sigma_1 x_1 \dots x_n \in Run(G) \mid \sigma_{i+1} \in \Sigma(last(x_0 \sigma_1 \dots x_i)) \setminus \pi_S(M_O(x_0 \sigma_1 \dots x_i)) \text{ and } (x_i, \sigma_{i+1}, x_{i+1}) \in \pi_D(x_0 \sigma_1 \dots x_i, \pi_S(M_O(x_0 \sigma_1 \dots x_i))) \text{ for each } i \in \{0, 1, \dots, n-1\}\}$. Notice that there may exist more than one run generated under (π_S, π_D) because the strategy π_S is a function defined on the set of masked runs $M_O(Run(G))$. Thereby, $Play(G, \pi_S, \pi_D)$ is a set of runs. Π_S and Π_D denote the sets of strategies of the supervisor and the DES, respectively. Let $Play(G, \pi_S) = \{r \in Play(G, \pi_S, \pi_D) \mid \pi_D \in \Pi_D\}$ (resp. $Dead(G, \pi_S) = \{r \in Play(G, \pi_S) \mid \Sigma(last(r)) \setminus \pi_S(M_O(r)) = \emptyset\}$) be the set of all runs (resp. deadlock runs) generated under the strategy $\pi_S \in \Pi_S$.

4. Fixed-Initial-Credit Energy Problem

Let $e_0 \in \mathbb{N}$ be the initial credit energy given at the initial

state x_{G0} of the DES. For each transition $(x, \sigma, x') \in \delta$, the weight $w(x, \sigma, x')$ indicates the energy that the system gains (if $w(x, \sigma, x') \geq 0$) or loses (if $w(x, \sigma, x') < 0$) by executing the transition. Then, we consider the following problem.

Problem 1. For a given initial credit energy $e_0 \in \mathbb{N}$, the *fixed-initial-credit energy problem* is to find a strategy $\pi_s \in \Pi_S$ such that

- 1) $Dead(G, \pi_S) = \emptyset$, and
- 2) $EL(r) \geq 0$ for all $r = x_0\sigma_1x_1\dots\sigma_nx_n \in Play(G, \pi_S)$, where $EL(r) = e_0 + \sum_{i=0}^{n-1} w(x_i, \sigma_{i+1}, x_{i+1})$.

$EL(r)$ is called an energy level of the system by the run r .

In other words, the strategy π_s controls the DES in such a way that 1) there is no deadlock run and 2) the energy level of the system never drops below zero. Next, we show that the problem is reducible to computing a winning strategy in a turn-based reachability game [13].

In order to solve the problem, we introduce an *observation function* which indicates the set of possible current states of the DES and their energy levels. Formally, an *observation function* is a function $o : X \rightarrow \mathbb{Z} \cup \{\perp\}$. For each state $x \in X$, if $o(x) \in \mathbb{Z}$, then x is a possible current state and the energy level of the system is $o(x)$. Otherwise, x is not the current state. Denoted by $supp(o) = \{x \in X | o(x) \neq \perp\}$ is the set of all possible current states of the DES according to the observation function o . The function o is said to be *non-negative* if $o(x) \geq 0$ for all $x \in supp(o)$. Let O be the set of all observation functions of the DES G . We consider a relation \leq defined on O as follows: for each $o_1, o_2 \in O$, we have $o_1 \leq o_2$ if (1) $supp(o_1) = supp(o_2)$ and (2) $o_1(x) \leq o_2(x)$ for each $x \in supp(o_1)$.

For a control pattern $\gamma \in \Gamma$, a masked event $\lambda \in \Lambda$, and a masked state $y \in Y$, o_2 is the (γ, λ, y) -successor of o_1 if the following conditions hold.

1. $supp(o_2) = \{x_2 \in X | \exists (x_1, \sigma, x_2) \in \delta, x_1 \in supp(o_1), \sigma \in M_\Lambda^{-1}(\lambda) \setminus \gamma, M_Y(x_2) = y\}$.
2. For each $x_2 \in supp(o_2)$, $o_2(x_2) = \min\{o_1(x_1) + w(x_1, \sigma, x_2) | (x_1, \sigma, x_2) \in \delta, x_1 \in supp(o_1), \sigma \in M_\Lambda^{-1}(\lambda) \setminus \gamma\}$.
3. For each $x_1 \in supp(o_1)$, there exists $(x_1, \sigma, x_2) \in \delta$ such that $\sigma \in M_\Lambda^{-1}(\lambda) \setminus \gamma$.

Namely, the (γ, λ, y) -successor of o_1 the set of possible current states changes from $supp(o_1)$ to $supp(o_2)$ if the supervisor selects the control pattern γ , and observes the masked state y and the masked event λ . The condition 3 guarantees that the selected control pattern γ enables at least one event at each state in $supp(o_1)$. Therefore, assigning the control pattern γ for the observation $supp(o_1)$ does not generate a deadlock run. Moreover, the observation o_2 indicates the worst-case energy of the DES at each state in $supp(o_2)$. For any observation $o \in O$, $succ(o, \gamma, \lambda, y)$ denotes the (γ, λ, y) -successor of o .

Then, we construct a game automaton $H = \langle Q_H = Q_S \cup Q_D, \Sigma_H = \Gamma \cup (\Lambda \times Y), \delta_H = \delta_S \cup \delta_D, o_{H0} \rangle$, where $Q_S \subseteq O((\Lambda \times Y)O)^*$ (resp. $Q_D \subseteq O((\Lambda \times Y)O)^*\Gamma$) is the set of states

Algorithm 1 $H(G)$

Require: $G = \langle X, \Sigma, \delta, x_{G0}, w \rangle, e_0$

- 1: $Q^+ \leftarrow \emptyset, Q_D \leftarrow \emptyset, \delta_S \leftarrow \emptyset, \delta_D \leftarrow \emptyset$
- 2: $o_{H0}(x_{G0}) \leftarrow e_0$ and $o_{H0}(x) \leftarrow \perp$ for all $x \in X \setminus \{x_{G0}\}$
- 3: $Q_S \leftarrow \{o_{H0}\}$
- 4: **for all** $q_s = o_0(\lambda_1, y_1)o_1 \dots (\lambda_n, y_n)o_n \in Q_S$ **do**
- 5: **if** o_n is non-negative **and** $PRE(q_s) = -1$ **then**
- 6: **for all** $(\gamma, \lambda, y) \in \Gamma \times \Lambda \times Y$ such that exists $o = succ(o_n, \gamma, M_\Lambda(\sigma), M_Y(x'))$ **do**
- 7: add $q_s\gamma$ in Q_D and $(q_s, \gamma, q_s\gamma)$ in δ_S
- 8: **if** $q'_s = q_s(\lambda, y) \notin Q_S$ **then**
- 9: add q'_s in Q_S
- 10: **end if**
- 11: add $(q_s\gamma, (\lambda, y), q'_s)$ in δ_D
- 12: **end for**
- 13: **else if** o_n is non-negative **then**
- 14: add $q_s \in Q^+$
- 15: **end if**
- 16: **end for**
- 17: **return** $H = \langle Q_H = Q_S \cup Q_D, \Sigma_H = \Gamma \cup (\Lambda \times Y), \delta_H = \delta_S \cup \delta_D, o_{H0} \rangle, Q^+$
- 18: **function** $PRE(o_0(\lambda_1, y_1)o_1 \dots (\lambda_n, y_n)o_n)$
- 19: **if** $\exists m \in \{0, 1, \dots, n-1\}, o_m \leq o_n$ **then**
- 20: **return** $o_0(\lambda_1, y_1) \dots o_m$
- 21: **else**
- 22: **return** -1
- 23: **end if**
- 24: **end function**

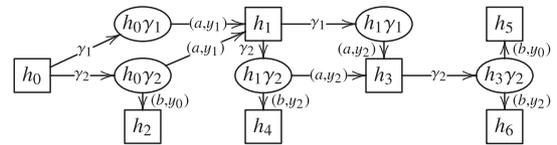


Fig. 2 The game automaton constructed from the DES in Fig. 1 using Algorithm 1. The control patterns are $\gamma_1 = \{b_1\}$ and $\gamma_2 = \emptyset$. Observations o_0, o_1, o_2, o_3, o_4 , and o_5 are defined as follows. $supp(o_0) = \{x_0\}$, $o_0(x_0) = e_0 = 0$. $supp(o_1) = \{x_1, x_2\}$, $o_1(x_1) = 0$, $o_1(x_2) = 1$. $supp(o_2) = \{x_3\}$, $o_2(x_3) = 5$. $supp(o_3) = \{x_5\}$, $o_3(x_5) = 1$. $supp(o_4) = \{x_4\}$, $o_4(x_4) = -1$. $supp(o_5) = \{x_5\}$, $o_5(x_5) = 4$. Then, we have $h_0 = o_0$, $h_1 = h_0(a, y_1)o_1$, $h_2 = h_0(b, y_0)o_2$, $h_3 = h_1(a, y_2)o_3$, $h_4 = h_1(b, y_2)o_4$, $h_5 = h_3(b, y_0)o_0$, $h_6 = h_3(b, y_2)o_5$.

of the supervisor (resp. the DES).[†] $\delta_S \subseteq Q_S \times \Gamma \times Q_D$ (resp. $\delta_D \subseteq Q_D \times (\Lambda \times Y) \times Q_S$) is the set of out-going transitions from the supervisor's (resp. the DES's) states, and $o_{H0} \in Q_S$ is the initial state. Algorithm 1 computes the game automaton H , and the set $Q^+ = \{q = o_0(\lambda_1, y_1)o_1 \dots (\lambda_n, y_n)o_n \in Q_S | o_n \text{ is non-negative, } \Sigma_H(q) = \emptyset, \text{ and there exists an integer } m \in \{0, 1, \dots, n-1\} \text{ such that } o_m \leq o_n\}$. For each $q = o_0(\lambda_1, y_1)o_1 \dots (\lambda_n, y_n)o_n \in Q^+$, $PRE(q)$ is the index $m < n$ such that $o_m \leq o_n$. This algorithm is modified from the algorithm for solving the fixed initial credit energy problem for a two-player game played on graph proposed in [12].

Figure 2 shows the game automaton constructed from the DES in Fig. 1 using Algorithm 1 where the initial credit energy $e_0 = 0$. The deadlock states in this automaton are h_2, h_4, h_5 , and h_6 . Notice that $h_2 = o_0(b, y_0)o_2$, and $supp(o_2)$ contains the deadlock state x_3 . Moreover, $h_4 = o_0(a, y_1)o_1(b, y_2)o_4$, and the function o_4 is not non-negative because $o_4(x_4) = -1$.

[†]In these concatenations, each element in O is regarded as a symbol representing the corresponding observation function.

Therefore, we have $Q^+ = \{h_5, h_6\}$.

Theorem 2. *The state set Q_H is finite.*

Proof. Let O be regard as the set of symbols representing the observation functions. Consider an infinite sequence $o_0o_1\dots \in O^\omega$ such that for each $k \in \mathbb{N}$, o_k represents a non-negative observation function. Since the state set X is finite, $\{supp(o)|o \in O\} \subseteq 2^X$ is also finite. Then, there exist $i, j \in \{0, 1, \dots\}$ such that $i < j$ and $o_i \leq o_j$ by Dickson's lemma [15].

We prove this theorem using a contradiction. Suppose that Q_H is infinite. Since X and Σ are finite, the number of outgoing transitions at each state in Q_H is also finite. By König's lemma [16], [17], there exists an infinite sequence $q_0^S \gamma_1 q_1^D(\lambda_1, y_1) q_1^S \dots$ such that $(q_i^S, \gamma_{i+1}, q_{i+1}^D) \in \delta_S$ and $(q_{i+1}^D, (\lambda_{i+1}, y_{i+1}), q_{i+1}^S) \in \delta_D$ for each $i \in \mathbb{N}$. From the construction of H , there exists $o_0(\lambda_1, y_1) o_1 \dots \in O((\Lambda \times Y)O)^\omega$ such that $o_0(\lambda_1, y_1) o_1 \dots (\lambda_k, y_k) o_k \in Q_S$ for each $k \in \mathbb{N}$, all observations o_0, o_1, \dots are non-negative, and there is no $i, j \in \mathbb{N}$ such that $i < j$ and $o_i \leq o_j$. This is a contradiction to Dickson's lemma in the above discussion. \square

From Theorem 2, the construction of H always terminates. From the construction of H , the set $Run(H)$ does not contain a cycle. Moreover, each deadlock state q of H (i.e., each $q \in Q$ such that $\Sigma_H(q) = \emptyset$) is included in Q_S .

Then, we consider a turn-based reachability game played on H between the supervisor and the DES. A strategy of the supervisor (resp. the DES) is a function $\phi_S : Q_S \rightarrow \delta_S$ (resp. $\phi_D : Q_D \rightarrow \delta_D$). We restrict that for each $q_d \in Q_D$ (resp. $q_s \in Q_S$), $\phi_D(q_d)$ (resp. $\phi_S(q_s)$) is an out-going transition from q_d (resp. q_s). Let Φ_S and Φ_D be the sets of strategies of the supervisor and the DES for the game H , respectively. Both players play the game by alternately selecting an out-going transition from the current state of the game in their turns. Recall that we have $Q^+ = \{q = o_0(\lambda_1, y_1) o_1 \dots (\lambda_n, y_n) o_n \in Q_S | o_n \text{ is non-negative, } \Sigma_H(q) = \emptyset, \text{ and } \exists m \in \{0, 1, \dots, n-1\} \text{ such that } o_m \leq o_n\}$. The objective of the supervisor is to reach a state in Q^+ , while the objective of the DES is to prevent the game from entering Q^+ . Since $Run(H)$ does not contain any cycle, this game always terminate at a deadlock state of the automaton H . Let $Play(H, \phi_S, \phi_D) \in Run(H)$ be the set of runs generated under strategies ϕ_S and ϕ_D by the automaton H , and $Dead(H, \phi_S) = \{h \in Play(H, \phi_S, \phi_D) | \phi_D \in \Phi_D, \Sigma_H(last(h)) = \emptyset\}$.

Problem 3. The Q^+ -reachability problem is to find $\phi_S \in \Phi_S$ such that $last(Dead(H, \phi_S)) \subseteq Q^+$.

Let $H' = \langle Q_H \setminus Q^+, \Sigma_H, \delta'_H, o_{H0} \rangle$ be the automaton modified from H as follows: for each $(q_d, (\lambda, y), q_s) \in \delta_H$, $(q_d, (\lambda, y), PRE(q_s)) \in \delta'_H$ if $q_s \in Q^+$, and $(q_d, (\lambda, y), q_s) \in \delta'_H$ otherwise. For each $r \in Play(H, \phi_S)$ visiting $PRE(q_s)$ and q_s such that $q_s \in Q^+$, there is a corresponding run $r' \in Play(H', \phi_S)$ that visits the state $PRE(q_s)$ twice. Recall that $PRE(q_s) \leq q_s$. Therefore, the runs generated by H' represent the sequences of observation functions in Q^+ .

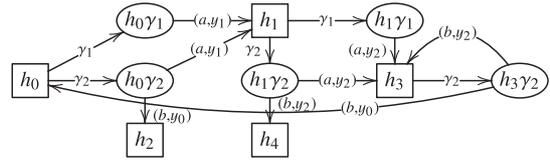


Fig. 3 The automaton modified from one in Fig. 2.

$q_0^S \gamma_1 q_1^D(\lambda_1, y_1) q_1^S \dots q_m^S \dots q_n^S \in Run(H)$ such that $q_n^S \in Q^+$ and $q_m^S = PRE(q_n^S)$, there exists runs $q_0^S \gamma_1 q_1^D(\lambda_1, y_1) q_1^S \dots q_m^S \gamma_{m+1} q_{m+1}^D(\lambda_{m+1}, y_{m+1}) (q_{m+1}^S \dots q_n^S)^* \in Run(H')$ that contains the corresponding non-negative cycle in $q_m^S \dots q_n^S$. The modified automaton of the game automaton in Fig. 2 is illustrated in Fig. 3. Then, from a given strategy $\phi_S \in \Phi_S$ for the game H , we define a strategy $\pi_{\phi_S} \in \Pi_S$ for the game G as follows. For each $r = y_0 \lambda_1 y_1 \dots \lambda_n y_n \in M_O(Run(G))$,

1. if there exists a run $r_H = q_0^S \gamma_1 q_1^D(\lambda_1, y_1) q_1^S \dots \gamma_n q_n^D(\lambda_n, y_n) q_n^S \in Run(H')$ such that $\gamma_i = \phi_S(q_i^S)$ for each $i \in \{0, 1, \dots, n-1\}$, then $\pi_{\phi_S}(r) = \phi_S(r_H)$;
2. otherwise, $\pi_{\phi_S}(r) = \emptyset$.

Theorem 4. *There exists $\pi_S \in \Pi_S$ that satisfies the fixed-initial-credit energy problem if and only if there exists $\phi_S \in \Phi_S$ that satisfies the Q^+ -reachability problem. Moreover, for a given strategy $\phi_S \in \Phi_S$ that satisfies the Q^+ -reachability problem, the strategy $\pi_{\phi_S} \in \Pi_S$ satisfies the fixed-initial-credit energy problem.*

Proof. (\rightarrow) Let $\pi_S \in \Pi_S$ be a strategy of the supervisor for the game G that satisfies the fixed-initial-credit energy problem. Let $\phi_S \in \Phi_S$ be a strategy of the supervisor for the game H defined as follows.

1. $\phi_S(q_{H0}) = \pi_S(M_Y(x_{G0}))$.
2. For each $q = o_0(\lambda_1, y_1) o_1 \dots (\lambda_n, y_n) o_n \in Q_S$,

$$\phi_S(q) = \pi_S(M_Y(x_{G0}) \lambda_1 y_1 \dots \lambda_n y_n).$$

Then, we show that $last(Dead(H, \phi_S)) \subseteq Q^+$ using a contradiction. From the construction of H , we have $Dead(H, \phi_S) \subseteq Q_S$. Suppose that there exists $r_H \in Dead(H, \phi_S)$ where $last(r_H) \notin Q^+$. Since Algorithm 1 does not add any out-going transition from the state $last(r_H)$ in δ_S , at least one of the following cases holds.

1. $last(r_H)$ is not non-negative.
2. There does not exist a transition $(\gamma, \lambda, y, o) \in \Gamma \times \Lambda \times Y \times O$ such that o is the (γ, λ, y) -successor of o_n .

For case 1, from the construction of H , there exists $r_G = x_0 \sigma_1 x_1 \dots x_n \in Play(G, \pi_S)$ such that $M_O(r_G) = y_{G0} \lambda_1 y_1 \dots \lambda_n y_n$, and $e_0 + EL(r_G) < 0$. For case 2, from the construction of H , there exists $r_G = x_0 \sigma_1 x_1 \dots x_n \in Play(G, \pi_S)$ such that $M_O(r_G) = y_{G0} \lambda_1 y_1 \dots \lambda_n y_n$ and $\Sigma(x_n) = \emptyset$. Therefore, $Dead(G, \pi_S) \neq \emptyset$. Both cases are contradictions to the definition of Problem 1.

(\leftarrow) Let $\phi_S \in \Phi_S$ be a strategy of the supervisor for the game H that satisfies the Q^+ -reachability problem. We show

that the strategy $\pi_{\phi_s} \in \Pi_s$ satisfies the fixed-initial-credit energy problem using a contradiction. Suppose that π_{ϕ_s} does not satisfy the problem. By the definition of Problem 1 1, we have $Dead(G, \pi_{\phi_s}) \neq \emptyset$ or there exists $r_G \in Play(G, \pi_{\phi_s})$ with $e_0 + EL(h) < 0$.

First, consider the case where $Dead(G, \pi_{\phi_s}) \neq \emptyset$. Let $r_G = x_0\sigma_1x_1 \dots \sigma_nx_n \in Dead(G, \pi_{\phi_s})$. From the constructions of H and H' , there exists $r_H = q_0^S\gamma_1q_1^D(\lambda_1, y_1)q_1^S \dots \gamma_nq_n^D(\lambda_n, y_n)q_n^S \in Play(H', \phi_s)$ such that $\Sigma_H(q_n^S) = \emptyset$ and $M_O(r_G) = y_{G0}\lambda_1y_1 \dots \lambda_ny_n$. However, since $q_n^S \in Q'_H$ and $\Sigma_H(q_n^S) = \emptyset$, we have $q_n^S \notin Q^+$. Thus, $r_H \in Play(H, \phi_s)$ is the run that visits a deadlock state which is not in Q^+ . This is a contradiction to the definition of Problem 3.

Next, consider the case where there exists $r_G = x_0\sigma_1x_1 \dots \sigma_nx_n \in Play(G, \pi_{\phi_s})$, $e_0 + EL(h) < 0$. From the constructions of H and H' , there exists $r_H = q_0^S\gamma_1q_1^D(\lambda_1, y_1)q_1^S \dots \gamma_nq_n^D(\lambda_n, y_n)q_n^S \in Play(H', \phi_s)$ such that $o = last(q_n^S) \in O$ is not non-negative and $M_O(r_G) = y_{G0}\lambda_1y_1 \dots \lambda_ny_n$. From the construction of H' , r_H must also be included in $Play(H, \phi_s)$. Obviously, $q_n^S \notin Q^+$. This is a contradiction to the definition of Problem 3. \square

From Theorem 4, the fixed-initial-credit energy problem can be solved by algorithms for computing a winning positional strategy of the first player turn-based reachability games [13]. By the similar discussion in [111], [18], the upper bound of the size of game H , as well as the upper bound of the time complexity of Algorithm 1, is bounded by the Ackermann function.

5. Application Example

We consider the synthesis of a supervisor for robot path planing for exploring the rooms shown in Fig. 4(a). The paths are represented by the weighted automaton shown in

Fig. 4 (b). The task of the robot is to explore an area while receiving the control commands sent via a wireless network. The initial state is x_0 , the initial credit energy is $e_0 = 0$, and the transition from x_0 to x_1 represents the energy recharge. The negative weight of the other transitions represent the energies needed for traveling through the paths. The uncontrollable event u_0 represents a doorway that always open, while the controllable events c_0 and c_1 represent ones that are controlled by the supervisor. Since the doorways c_0 and u_0 at each state are close to each other, they are observed by the masked event λ_0 . Moreover, the states in the same areas are mapped to the same masked state. The control objective is to allow the robot to explore the rooms provided that its energy never drops below zero.

By applying algorithm 1, we obtain the game automaton in Fig. 5. The square states are the states of the supervisor (in Q_S) and the circle states are ones of the DES (in Q_D). The square states with thick frames belong to the set Q^+ . Due to limitations of space, we only show the last observation function at each state of the supervisor. Moreover, instead of showing the labels of the transitions in δ_D , we

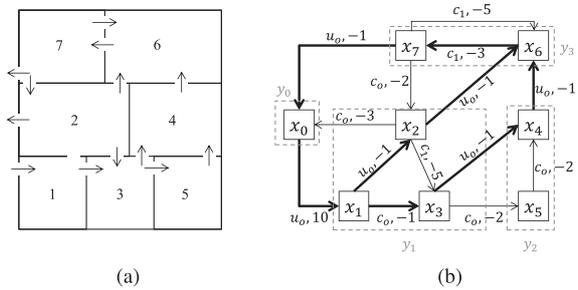


Fig. 4 (a) The area map for the robot system to explore. (b) The paths of the DES represented by an automaton where x_0 is the initial state and u_0 is the only uncontrollable event. The functions M_Y and M_Λ are as follows: $M_Y(x_0) = y_0$, $M_Y(x_1) = M_Y(x_2) = M_Y(x_3) = y_1$, $M_Y(x_4) = M_Y(x_5) = y_2$, $M_Y(x_6) = M_Y(x_7) = y_3$, $M_\Lambda(u_0) = M_\Lambda(c_0) = \lambda_0$, and $M_\Lambda(c_1) = \lambda_1$.

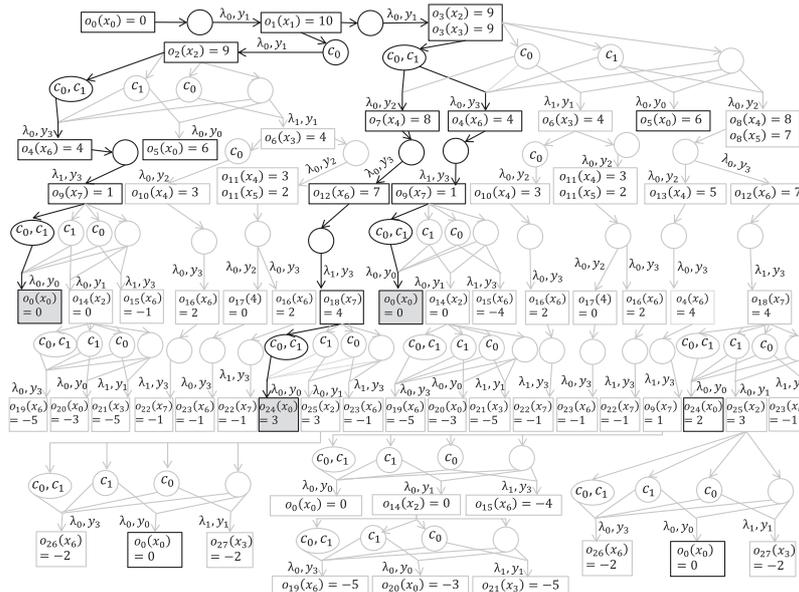


Fig. 5 The game automaton that is computed from the DES in Fig. 4 (b) using Algorithm 1.

show the disabled events at each state of the DES. That is, the circle states labeled by (c_0, c_1) , c_1 , c_0 , and the no-labeled circle states represent the control patterns $\{c_0, c_1\}$, $\{c_1\}$, $\{c_0\}$ and \emptyset , respectively. We only show the states of the DES where at least one controllable event is disabled. For example, at the initial state o_0 , we only consider the control pattern \emptyset because there is no outgoing controllable transition at the state x_0 of the DES in Fig. 4 (b).

By using the algorithms for reachability games [13], we compute the strategies of the supervisor that can control the generated runs to reach the states in Q^+ . The runs are shown as the black-line transitions in Fig. 5. Notice that there are two possible ways to control the DES for the masked run $o_0(\lambda_0, y_1)o_1$: by disabling or enabling the event c_0 . If we enable the event c_0 , the corresponding strategy π_s of the supervisor for the DES in Fig. 4 (b) is as follow:

- $\pi_s(o_0) = \pi_s(o_0(\lambda_0, y_1)o_1) = \emptyset$,
- $\pi_s(o_0(\lambda_0, y_1)o_1(\lambda_0, y_1)o_3) = \{c_0, c_1\}$,
- $\pi_s(o_0(\lambda_0, y_1)o_1(\lambda_0, y_1)o_3(\lambda_0, y_2))o_7 = \emptyset$,
- $\pi_s(o_0(\lambda_0, y_1)o_1(\lambda_0, y_1)o_3(\lambda_0, y_3))o_4 = \emptyset$,
- $\pi_s(o_0(\lambda_0, y_1)o_1(\lambda_0, y_1)o_3(\lambda_0, y_2))o_7(\lambda_0, y_3)o_{12} = \emptyset$,
- $\pi_s(o_0(\lambda_0, y_1)o_1(\lambda_0, y_1)o_3(\lambda_0, y_3))o_4(\lambda_1, y_3)o_9 = \{c_0, c_1\}$,
- $\pi_s(o_0(\lambda_0, y_1)o_1(\lambda_0, y_1)o_3(\lambda_0, y_2))o_7(\lambda_0, y_3)o_{12}(\lambda_1, y_3)o_{18} = \{c_0, c_1\}$, and
- $\pi_s(r) = \emptyset$ for any other masked run r .

The runs generated by the supervised DES under π_s are shown in thick lines in Fig. 4. In the supervised DES, the energy of each run never drops below zero.

6. Conclusions

We studied the supervisory control of partially observed quantitative DESs under the fixed-initial-credit energy objective. Partial observation is modeled by a mapping from each event and state of the DES to the corresponding masked event and masked state that are observed by a supervisor. The fixed-initial-credit energy problem is to synthesize a supervisor under which the controlled DES does not contain a deadlock and the energy level of the system never goes below zero. Then, the proposed problem was reduced to computing a winning strategy in a turn-based reachability game whose size is bounded by the Ackermann function. It is future work to consider the lower bound of the size of reachability game.

References

- [1] P.J. Ramadge and W.M. Wonham, "Supervisory control of a class of discrete-event process," *SIAM J. Control Optim.*, vol.25, no.1, pp.206–230, Jan. 1987.
- [2] K. Chatterjee, R. Majumdar, and T.A. Henzinger, "Controller synthesis with budget constraints," *Lecture Notes in Computer Science*, vol.4981, pp.72–86, 2008.
- [3] K. Chatterjee and R. Majumdar, "Minimum attention controller synthesis for omega-regular objectives," *Lecture Notes in Computer Science*, vol.6919, pp.145–159, 2011.
- [4] F. Lin and W.M. Wonham, "On observability of discrete event systems," *Information Sciences*, vol.44, no.3, pp.173–198, 1998.

- [5] S. Takai, T. Ushio, and S. Kodama, "A new class of supervisors for partially observed discrete event systems," *Proc. ACC*, pp.2631–2635, 1995.
- [6] J.G. Thistle and W.M. Wonham, "Supervision of infinite behavior of discrete-event systems," *SIAM J. Control Optim.*, vol.32, no.4, pp.1098–1113, 1994.
- [7] P.J.G. Ramadge, "Some tractable supervisory control problems for discrete-event systems modeled by Büchi automata," *IEEE Trans. Automat. Control*, vol.34, no.1, pp.10–19, 1989.
- [8] S. Pruekprasert, T. Ushio, and T. Kanazawa, "Quantitative supervisory control game for discrete event systems," *IEEE Trans. Automat. Control*, vol.61, no.10, pp.2987–3000, Oct. 2016.
- [9] U. Fahrenberg, L. Juhl, K.G. Larsen, and J. Srba, "Energy games in multiweighted automata," *Lecture Notes in Computer Science*, vol.6916, pp.95–115, 2011.
- [10] L. Brim, J. Chaloupka, L. Doyen, R. Gentilini, and J.F. Raskin, "Faster algorithms for mean-payoff games," *Formal Methods in System Design*, vol.38, no.2, pp.97–118, 2011.
- [11] G.A. Pérez, "The fixed initial credit problem for partial-observation energy games is Ack-complete," *Information Processing Letters*, vol.118, pp.91–99, Feb. 2017.
- [12] A. Degorre, L. Doyen, R. Gentilini, J.-F. Raskin, and S. Toruńczyk, "Energy and mean-payoff games with imperfect information," *Proc. CSL'10*, vol.6247, pp.260–274, 2010.
- [13] L. de Alfaro, T.A. Henzinger, and O. Kupferman, "Concurrent reachability games," *Theoretical Computer Science*, vol.386, no.3, pp.188–217, 2007.
- [14] S. Pruekprasert and T. Ushio, "Energy-base deadlock-free supervisory control of quantitative discrete event systems under Partial Observation," *Proc. ITC-CSCC*, pp.137–140, 2016.
- [15] R. Fröberg, *An introduction to Gröbner bases*, West Sussex, England, John Wiley & Sons, 1997.
- [16] D. König, "Über eine schlussweise aus dem endlichen ins unendliche," *Acta Sci. Math. (Szeged)*, vol.3, pp.121–130, 1927 (in German).
- [17] H.W. Kuhn, "Extensive games and the problem of information," in *Classics in games theory*, Princeton University Press, Princeton, NJ, US, 1997.
- [18] D. Figueira, S. Figueira, S. Schmitz, and P. Schnoebelen, "Ackermannian and primitive-recursive bounds with dickson's lemma," in *LICS*, IEEE Computer Society, pp.269–278, 2011.



Sasinee Pruekprasert received the B. E. degree from Kasetsart University, Bangkok, Thailand, in 2010, and the M. E. degree and Ph. D. degree from Osaka University, Osaka, Japan, in 2012 and 2016, respectively. She is currently a postdoctoral researcher in the National Institute of Informatics, Tokyo, Japan. Her research interests include control of discrete event systems and game theory applications.



Toshimitsu Ushio received his B. E., M. E., and Ph. D. degrees in 1980, 1982, and 1985, respectively, from Kobe University. He joined Osaka University as an Associate Professor in 1994, and is currently a Professor. His research interests include control and analysis of discrete event systems and scheduling of real-time systems. He is a member of IEEE, SICE, and ISClE.