On Nonblockingness Verification and Enforcement of Controlled Nondeterministic Discrete-Event Systems

Xiang Ren*, Zipei Wang

College of Electronic Information and Automation, Tianjin University of Science and Technology, 300222， China

*Email: *renxtust@163.com*

Abstract

Discrete event systems, as an important kind of cyber-physical systems, have been widely used in engineering field. In this paper, we first express the dynamics of a controlled nondeterministic discrete-event system (acronym is DES) as an algebraic state-space representation using the semi-tensor product (STP) theory. And then, we discuss the problems of state-based nonblockingness verification and enforcement of nondeterministic DESs. Specifically, we obtain a criterion of verifying whether a given controlled nondeterministic DES is nonblocking. Further, we develop an efficient matrix-based approach to enforce state-based nonblockingness. We illustrate the applications of the proposed theoretical results using an example.

Keywords: Discrete event systems, nondeterministic, nonblockingness, semi-tensor product theory

1. Introduction

With the development of information and network technologies, discrete event systems (DESs) have received considerable attention in information physics systems for many years, synthesizing multidisciplinary methods such as automatic control and computer science, see, e.g., [1], [2]. Verification and synthesis are two main research issues in the field of DESs. We refer the readers to the recent survey papers [3] for more details.

In the above-mentioned literature, the DESs of interest, in general, are deterministic DESs. The case of nondeterministic DESs has not been considered widely so far since they have more complex dynamic evolution behaviors than deterministic DESs. It should be pointed out that, however, there are many realistic plants that need to be modeled as nondeterministic DESs. Therefore, how to model and analyze effectively the dynamics of controlled nondeterministic DESs are still an interesting topic for both computer and control researchers.

In classical DESs, the nonblockingness is a basic and important problem in the study of DESs. Also, some efficient approaches have been proposed. For instance, in [4], the authors studied the problem of blocking detection of DESs by means of language-based approach. In this paper, we develop a new methodology to investigate how to model controlled nondeterministic DESs by using the STP of matrices. Based the proposed new model, we further study the property verification and enforcement of nonblockingness.

The rest of this article is organized as follows. The second section presents some basic notations, concepts and a matrix-based expression needed in this paper. In the third part, we formulate the state-based nonblockingness property verification problem, and we show how the proposed approach can be to applied to enforce nonblockingness. In the fourth section, an example is presented to illustrate the application of the proposed approach. The fifth part summarizes the main content of this paper.

2. Preliminaries

2.1. *Notations*

In this subsection, we introduce some notations, which will be used in the sequel. $\mathbb{R}^n$ is the set of all vectors of dimension n; $|X|$ is the cardinality of set X; $\mathcal{M}_{m\times n}$ is the set of $m\times n$ matrices; $M_{(i,j)}$ is the (i,j) element of matrix M; $Col_j(M)$ is the j-th column of matrix M; $Col(M)$ is the set of all columns of matrix M; $0_n := \underbrace{[0,0,\cdots,0]}_{n}$; $1_n := \underbrace{[1,1,\cdots,1]}_{n}$; $\delta_n^0 := \underbrace{[0,0,\cdots,0]}_{n}^T$; $\delta_n^k := Col_k(I_n)$, $1\le k\le n$; $\Delta_n := \{\delta_n^1,\delta_n^2,\cdots,\delta_n^n\}$; $\tilde{\Delta}_n := \{\delta_n^0,\delta_n^1,\cdots,\delta_n^n\}$; $L\in\mathcal{M}_{m\times n}$ is a logical matrix (resp., generalised logical matrix) if $Col(L)\subseteq\Delta_m$ (resp., $Col(L)\subseteq\tilde{\Delta}_m$). We denote the set of $m\times n$ logical matrices (resp., generalised logical matrix) by $\mathcal{L}_{m\times n}$ (resp., $\tilde{\mathcal{L}}_{m\times n}$); If matrix $L\in\tilde{\mathcal{L}}_{m\times n}$, then it can be expressed as $L \in \left[\delta_m^{i_1}, \delta_m^{i_2}, \cdots, \delta_m^{i_n}\right]$ and it is briefly denoted as $L \in \delta_m\left[i_1, i_2, \cdots, i_n\right]$, where $i_k \in \{0, 1, \cdots, m\}$, $1 \le k \le n$.

2.2. *Semi-tensor product (STP) of matrices*

In this subsection, we recall the notions of STP and swap matrix. We refer the readers to [5] and/or [6] for more details on them.

Definition 2.1 ([5]): Let $A \in \mathcal{M}_{m \times n}$, $B \in \mathcal{M}_{p \times q}$. The STP of A and B is defined as

$$A \ltimes B = (A \otimes I_{t/n})(B \otimes I_{t/p}), \tag{1}$$

where t denotes the least common multiple of n and p, i.e., $t = lcm(n, p)$; $\otimes$ is the Kronecker product.

Definition 2.2 ([5]): A swap matrix $W_{[m,n]}$ is a $mn \times mn$ logical matrix, which is defined as

$$\begin{aligned} W_{[m,n]} = \delta_{mn}[&1, m+1, 2m+1, \cdots, (n-1)m+1, \\ &2, m+2, 2m+2, \cdots, (n-1)m+2, \\ &\cdots, m, 2m, 3m, \cdots, nm]. \end{aligned} \tag{2}$$

Lemma 2.1([5]): Let $X \in \mathbb{R}^m$ and $Y \in \mathbb{R}^n$ be two column vectors. Then

$$W_{[m,n]} XY = YX, \; W_{[n,m]} YX = XY. \tag{3}$$

2.3. *System model*

In this subsection we recall the formalism used in the paper. More details on DESs can be found in [2].

A DES is modeled as a nondeterministic finite automaton (NFA) $G = (X, \Sigma, \delta, X_0, X_m)$, where X is the finite set of states, Σ is the finite set of events called alphabet or input symbols, $X_0 \subseteq X$ is the set of initial states, $X_m \subseteq X$ is the set of marked states (or accepted states), $\delta : X \times \Sigma \to 2^X$ is the partial transition function(2^X denotes the power set of X), which describes the system dynamics: given states $x, y \in X$ and an event $\sigma \in \Sigma$, $y \in \delta(x, \sigma)$ means the execution of σ from state x takes the system to state y . Note that $\delta(x, \sigma)$ is undefined when the event σ cannot be executed from the state x . $\delta(x, \sigma)!$ denotes $\delta(x, \sigma)$ is well-defined. Obviously, the transition function can be extended to $\delta : X \times \Sigma^* \to 2^X$ in terms of $\delta(x, e) := \delta(\delta(\cdots \delta(\delta(x, e_{j_1}), e_{j_2}) \cdots), e_{jt})$, where $e = e_{j_1} e_{j_2} \cdots e_{j_t} \in \Sigma^*$, Σ^* denotes the set of finite strings on the alphabet Σ , including the empty string $\bullet$. The objective of this paper is to investigate the controlled nondeterministic DESs. In this regard, the event set Σ can be partitioned into two disjoint subsets, i.e., $\Sigma = \Sigma_c \bigcup \Sigma_{uc}$, where Σ_c denotes the set of controllable events, Σ_{uc} denotes the set of uncontrollable events. We here assume that all events in Σ are observable.

In general, we wish to adjoin a supervisor or a controller S to interact with G in a feedback manner. More precisely, the transition function of G can be controlled by S in the sense that the controllable events of G can be dynamically enabled or disabled by S after each transition. Formally, a *state-feedback supervisor*, denoted by S , is a function $S : X \to 2^{\Sigma_c}$ that determines the set of events $S(x) \subseteq \Sigma_c$ to be disabled at each state $x \in X$, while events not belonging to the set $S(x)$ remain enabled at state x . The *controlled system* (or called *supervised system*) consisting of G and S , denoted by S/G , is another nondeterministic finite automation given as

$$S/G = (X, \Sigma, \delta_S, X_0, X_m), \tag{4}$$

where X , Σ , X_0 and X_m are as defined above, δ_S is the partial transition function of S/G, i.e.,

$$\delta_S(x, \sigma) = \begin{cases} \delta(x, \sigma), if \;\; \delta(x, \sigma)! \; and \;\; \sigma \notin S(x) \\ \text{undefined, otherwise.} \end{cases} \tag{5}$$

We use the notation $\mathrm{H}(x)$ to denote the set of feasible events of G at state x . Thus supervisor S is called permissible if for all $x \in X$, $S(x) \subseteq \mathrm{H}(x) \bigcap \Sigma_c$. Note that it is not difficult to see that nondeterministic DESs can be viewed as special case of controlled nondeterministic DESs with $S(x) = \varnothing$ for all $x \in X$. The controlled nondeterministic DES S/G is depicted in Fig.1.

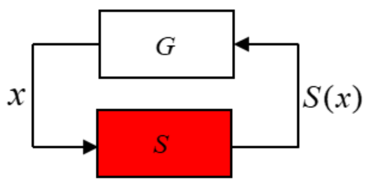

Fig. 1. The controlled nondeterministic DES S/G

3. Nonblockingness of Controlled Nondeterministic DESs

3.1. *Matrix-based expression of controlled nondeterministic DESs*

In order to obtain the matrix expression of the dynamics of controlled nondeterministic DESs, let us first give an equivalent description of the controlled system (4). For the nondeterministic DES G , we define a control pattern as a Boolean function: $\gamma : \Sigma_c \to \{0,1\}$ and we use the notation $\Gamma=\{0,1\}^{\Sigma_c}$ to denote the set of all Boolean functions on Σ_c. $\gamma\in\Gamma$ is interpreted as follows: for any $\sigma\in\Sigma_c$, $\gamma(\sigma)=1$ means that the control pattern γ allows σ to happen, while $\gamma(\sigma)=0$ means that the control pattern γ refuse σ to happen. Note that it is convenient to extend each $\gamma\in\Gamma$ to a function $\gamma:\Sigma\rightarrow\{0,1\}$ by defining $\gamma(\sigma)=1$ for each uncontrollable event $\sigma\in\Sigma_{uc}$. Further, we define a partial function of the form $f:X\rightarrow\Gamma$, called the *state-feedback control function* or *state-feedback* for short, that maps each state x in X into control pattern γ, i.e. $\gamma(\sigma)=f(x)(\sigma)$. Thus the controlled nondeterministic DES consisting of the state-feedback f, control pattern γ and nondeterministic DES G is described as

$$G_\gamma^f=(X,\Sigma,\delta_\gamma^f,X_0,X_m), \tag{6}$$

where X, Σ, X_0 and X_m are as defined above; δ_γ^f denotes the partial transition function of G_γ^f, which is defined as

$$\delta_\gamma^f(x,\sigma)=\begin{cases}\delta(x,\sigma), \text{ if } \delta(x,\sigma)! \text{ and } f(x)(\sigma)=1\\ \text{undefined, otherwise.}\end{cases} \tag{7}$$

For event $e_j\,(1\le j\le m)$, we define a $n\times n$ matrix F_j as

$$F_{j(s,t)}=\begin{cases}1, \text{ if } \delta_n^s\in\delta_\gamma^f(\delta_n^t,\delta_m^j)\wedge\delta_m^j \,\square\, e_j\in\Sigma_{uc}\\ r_{tj}, \text{ if } \delta_n^s\in\delta(\delta_n^t,\delta_m^j)\wedge\delta_m^j \,\square\, e_j\in\Sigma_c\\ 0, \text{ otherwise,}\end{cases} \tag{8}$$

where F_j is called transition structure matrix *w.r.t.* event e_j. Thus, the *transition structure matrix* (TSM) of controlled nondeterministic DES (6) is defined as

$$F=\left[F_1,F_2,\cdots,F_m\right], \tag{9}$$

where F is a $n\times mn$ symbol matrix.

Theorem 3.1: Given a controlled nondeterministic DES (6), the dynamics of (6) can be equivalently described as

$$x(t+1)=Fu(t)x(t), \tag{10}$$

where F is the TSM of system (6), which is defined in (9); $x(t)=(x_1(t),x_2(t),\cdots,x_n(t))^T$ is the vector form of state at step t, $x_i(t)$ denotes the number of different paths from the set of initial states to the state $x_i=\delta_n^i$ with a feasible event string of length $t-1$; $u(t)\in\Delta_m$ is vector form of event at step t.

Proof: We here omit this proof.

Remark 3.1: When the state-feedback f is known, the dynamics of (6) can be equivalently described by the following equation

$$x(t+1)=F_cu(t)x(t), \tag{11}$$

where $x(t)$ and $u(t)$ have the same interpretation as in Theorem 3.1; F_c is the TSM of (6), which is represented as

$$F_c=\left[F_1^c,F_2^c,\cdots,F_m^c\right]\in\tilde{\mathcal{L}}_{n\times mn}, \tag{12}$$

where the $n\times n$ matrix F_j^c is defined as

$$F_{j(s,t)}^c=\begin{cases}1, \text{ if } \delta_n^s\in\delta_\gamma^f(\delta_n^t,\delta_m^j)\\ 0, \text{ otherwise.}\end{cases} \tag{13}$$

3.2. *Nonblockingness verification of controlled nondeterministic DESs*

In classical DESs, the problem of nonblockingness verification is a basic and important topic. A controlled deterministic DES, also denoted by S/G is said to be nonblocking if and only if $\mathcal{L}(S/G)=\overline{\mathcal{L}_m(S/G)}$, where $\mathcal{L}(S/G)$ and $\overline{\mathcal{L}_m(S/G)}$ stand for the language generated and language marked by S/G, respectively. See, e.g., [2] for more details. As a generalization of deterministic model, we now investigate the nonblockingness of controlled nondeterministic DESs. We for simplicity use a *state-based description* to present the notion of nonblockingness. We call it *state-based nonblockingness,* while the nonblockingness defined in [2] is called *the language-based nonblockingness*. Actually, these two concepts are equivalent.

Definition 3.1: Given a controlled nondeterministic DES (6), a state $x\in X$ is called a *nonblocking state* if it is reachable from initial state set X_0 and it can reach a marked state from X_m. (6) is called *nonblocking* if its each state is nonblocking.

Remark 3.2: 1) By Definition 3.1, we know that a state of (6) is a *blocking state* if and only if it is not reachable from the set of initial states, or it can not reach any marked state.2) For the controlled nondeterministic DES (6), there exist two forms of blocking state, i.e., *deadlock state* and *livelock state*. More specifically, $x\in X$ is a deadlock state of (6) if it is an unmarked reachable state and $\mathrm{H}(x)=\varnothing$; $x\in X$ is a livelock state of (6) if it can not reach any marked state and $\mathrm{H}(x)\neq\varnothing$.

In what follows, we give a *matrix-based criterion* to verify the nonblockingness of controlled nondeterministic DES (6). Consider the controlled nondeterministic DES (6) with the matrix-based expression (11), from the definition of the TSM F_c presented in (13) and the viewpoint of graph theory, we readily know that the matrix $\sum_{j=1}^{m}F_j^c$ stands for the *adjacency matrix* of the directed graph representation of the controlled nondeterministic DES (6). Thus, state

$x_q = \delta_n^q$ is reachable from $x_p = \delta_n^p$ in t steps if and only if $(\sum_{j=1}^{m} F_j^c)^t_{(q,p)} > 0$. In this regard, we define a matrix $M =: \sum_{\alpha=1}^{n}(\sum_{j=1}^{m} F_j^c)^\alpha$, called *reachability matrix* of controlled nondeterministic DES (6), where $n = |X|$, $m = |\Sigma|$. Using it, we have the following result.

Proposition 3.1: Given a controlled nondeterministic DES (6) with the matrix-based expression (11), then

1) The set of states reachable from the set of initial states X_0 is

$$\mathrm{R}(X_0) = \{\delta_n^k \mid M_{(k,p)} > 0, \delta_n^p \in X_0, 1 \le k \le n\}, \quad (14)$$

where $M = \sum_{\alpha=1}^{n}(\sum_{j=1}^{m} F_j^c)^\alpha$ is the reachability matrix of (6).

2) The set of states that can reach X_m is

$$\tilde{\mathrm{R}}(X_m) = \{\delta_n^k \mid M_{(q,k)} > 0, \delta_n^q \in X_m, 1 \le k \le n\}, \quad (15)$$

where M is as defined above.

Proof: We here omit this proof.

Remark 3.3: By convention, we assume that $X_0 \subseteq \mathrm{R}(X_0)$ and $X_m \subseteq \tilde{\mathrm{R}}(X_m)$ hereinafter. Using Definition 3.1 and Proposition 3.1, the following main result on the verification of nonblockingness for the controlled nondeterministic DES (6) is very straightforward.

Theorem 3.2: Given a controlled nondeterministic DES (6) with the matrix-based expression (11), then (6) is nonblocking if and only if the following condition holds:

$$\mathrm{R}(X_0) = \tilde{\mathrm{R}}(X_m) = X, \quad (16)$$

where $\mathrm{R}(X_0)$ and $\tilde{\mathrm{R}}(X_m)$ have the same interpretation as in (14) and (15), respectively.

Notice that, from Theorem 3.2, it is not difficult to see that using our approach to verify nonblockingness of controlled nondeterministic DESs is very efficient and straightforward in comparison to the existing ones in the sense that it involves only some basic matrix manipulations that are completed via polynomial time.

3.3. *Nonblockingness enforcement of controlled nondeterministic DESs*

We know that, for uncontrolled DESs, the trimness property is closely related to the concept of nonblockingness. For instance, when a given original system does not satisfy the nonblockingness property, we can to enforce it via trim operation to restrict the system's behavior, see, e.g.,[2]. It is also meaningful for the case of controlled nondeterministic DESs, since a controlled nondeterministic DES, denoted by S/G, has a generated language and a marked language associated it. In this subsection, we present a matrix-based approach to enforcing nonblockingness property called nonblockingness enforcement hereinafter.

Consider the controlled nondeterministic DES (6) with the matrix-based expression (11), and we assume that, without loss of generality, system (6) is blocking. Using Lemma 2.1, equation (11) can be rewritten as

$$x(t+1) = \bar{F}_c x(t) u(t), \quad (17)$$

where $\bar{F}_c = F_c W_{[n,m]} \in \mathcal{M}_{n\times mn}$, called the *dual* TSM of (6). Partition matrix $\bar{F}_c$ into $\bar{F}_c = [\bar{F}_1^c, \bar{F}_2^c, \cdots, \bar{F}_n^c]$, where $\bar{F}_i^c \in \mathcal{M}_{n\times m}, 1 \le i \le n$. Also, construct matrix:

$$\bar{F}^{trim} = [\bar{F}_1^{trim}, \bar{F}_2^{trim}, \cdots, \bar{F}_n^{trim}], \quad (18)$$

where

$$\bar{F}_i^{trim} = \begin{cases} \bar{F}_i^c, & \text{if } \delta_n^i \in \mathrm{R}(X_0) \bigcap \tilde{\mathrm{R}}(X_m) \\ 0_{n\times m}, & \text{otherwise.} \end{cases} \quad (19)$$

Based on the above-constructed matrix (18), we have the following result.

Theorem 3.3: Given a controlled nondeterministic DES (6) with the matrix-based expression (11), then the controlled nondeterministic DES described by the following equation is nonblocking.

$$x(t+1) = F^{trim} u(t) x(t), \quad (20)$$

where $F^{trim} = \bar{F}^{trim} W_{[m,n]}$, $\bar{F}^{trim}$ is defined in (18).

Proof: we here omit the proof of Theorem 3.3.

Remark 3.4: From Theorem 3.3, we can obtain that, when the DES (6) is blocking, the computational complexity of enforcing the nonblockingness property by using the proposed approach is also polynomial time, i.e., $o(n^2 m)$, where n and m are the numbers of state nodes and event nodes, respectively. An example concerning the supervisory control of a submarine guard system is presented (see Example 4.1 below) to illustrate the effectiveness and application of the proposed results.

4. Illustrative Example

In this section, we give an example to illustrate the proposed results.

Example 4.1: Let us consider the supervisory control of a submarine guard system. A sea region is divided into several sections in terms of the shape of the seabed and it is depicted in Fig. 2. Two submarines, namely, and, cruise and guard the sea region. has its own separate repair port and home harbor, i.e., section No. 5 and No. 2, respectively. has repair and home harbor in section No. 4. Due to the seabed form and the sea underwater streams, as well as different size of the submarines, possible and controllable transitions of the submarines between these sections are also shown in Fig. 2.

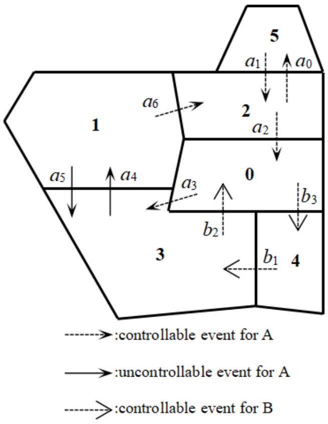

Fig. 2. Space guarded by the submarines.

At the beginning of inspection, A is in section No. 5 and B in section No. 4. It is required that each submarine can return its home harbor. Additionally, meeting of submarines A and B in any section of the sea region must be prohibited for safety reasons. The task of the supervisory control is how to synthesize the cruise trajectory of A and B to prevent the violation of the requirements described above.

Now we use the proposed approach to solve it.

According to Fig. 2, the dynamics of submarines A and B can be modeled by two finite-state automata shown in Fig. 3(a) and Fig 3(b), respectively. That is, $G_1=(X_1,\Sigma_1,\delta_1,x_0^1,X_m^1)$ with the initial state $x_0^1=5$, and the marked state $X_m^1=\{2\}$,while $G_2=(X_2,\Sigma_2,\delta_2,x_0^2,X_m^2)$ with the initial state $x_0^2=4$ and the marked state $X_m^2=\{4\}$.

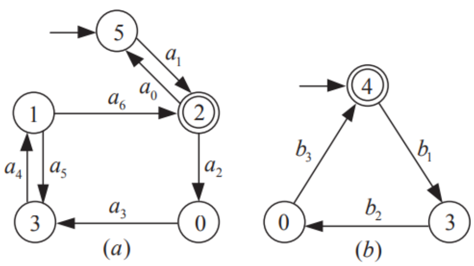

Fig. 3. (a) represents a DFA model of the submarine A cruise; (b) represents a DFA model of the submarine B cruise.

For the automation G_1, identifying $5 \sim \delta_5^1$, $2 \sim \delta_5^2$, $0 \sim \delta_5^3$, $3 \sim \delta_5^4$, $1 \sim \delta_5^5$; $a_{j-1} \sim \delta_{10}^j (1\le j\le 6)$. While for G_2, identifying $4 \sim \delta_3^1$, $3 \sim \delta_3^2$, $0 \sim \delta_3^3$; $b_k \sim \delta_{10}^{7+k} (1\le k\le 3)$.

According to lemma in reference [7], the dynamics of $G_1 \| G_2$ (called the parallel composition of G_1 and G_2) can be described by the following algebraic form

$$x(t+1)=Fu(t)x(t), \tag{21}$$

where $F=[F_1,F_2,\cdots,F_9,F_{10}]\in \tilde{\mathcal{L}}_{15\times 150}$, i.e.,

$$\begin{aligned} F=\delta_{15}[&0,0,0,1,2,3,0,0,0,0,0,0,0,0,0,4,5,6,0,0,0,0,\\ &0,0,0,0,0,0,0,0,0,0,0,7,8,9,0,0,0,0,0,0,0,0,\\ &0,0,0,0,0,0,0,10,11,12,0,0,0,0,0,0,0,0,0,0,0,\\ &0,0,0,0,13,14,15,0,0,0,0,0,0,0,0,0,0,0,0,0,\\ &0,10,11,12,0,0,0,0,0,0,0,0,0,0,0,0,4,5,6,2,0,\\ &0,5,0,0,8,0,0,11,0,0,14,0,0,0,3,0,0,6,0,0,9,0,\\ &0,12,0,0,15,0,0,0,1,0,0,4,0,0,7,0,0,10,0,0,13].\end{aligned}$$

Using Lemma 2.1, system (21) can be rewritten as

$$x(t+1)=\tilde{F}x(t)u(t), \tag{22}$$

where $\tilde{F}=FW_{[15,10]}\in \tilde{\mathcal{L}}_{15\times 150}$.

Partitioning $\tilde{F}=[\tilde{F}_1,\tilde{F}_2,\cdots,\tilde{F}_{15}]$, where $\tilde{F}_k\in \tilde{\mathcal{L}}_{15\times 10}$, $1\le k\le 15$. Since states **00** $\sim \delta_5^3\delta_3^3=\delta_{15}^9$ and **33** $\sim \delta_5^4\delta_3^2=\delta_{15}^{11}$ are forbidden states[2], all events attached to δ_{15}^9 and δ_{15}^{11} should be deleted and each state that can reach δ_{15}^9 or δ_{15}^{11} in 1 step by an uncontrollable event is also forbidden state. To this end, by applying Proposition 3.1, the sets of states reachable from δ_{15}^9 and δ_{15}^{11} in 1 step are $\{\delta_{15}^7,\delta_{15}^{12}\}$ and $\{\delta_{15}^{12},\delta_{15}^{14}\}$, respectively. The sets of states that can reach δ_{15}^9 and δ_{15}^{11} in 1 step are $\{\delta_{15}^6,\delta_{15}^8\}$ and $\{\delta_{15}^8,\delta_{15}^{10},\delta_{15}^{14}\}$, respectively. According to the conclusion of reachability analysis in reference[8], we have $\delta_{15}^9 \xrightarrow{\delta_{10}^{10}} \delta_{15}^7$, $\delta_{15}^9 \xrightarrow{\delta_{10}^{4}} \delta_{15}^{12}$, $\delta_{15}^6 \xrightarrow{\delta_{10}^{3}} \delta_{15}^{9}$, $\delta_{15}^8 \xrightarrow{\delta_{10}^{9}} \delta_{15}^{9}$; $\delta_{15}^{11} \xrightarrow{\delta_{10}^{9}} \delta_{15}^{12}$, $\delta_{15}^{11} \xrightarrow{\delta_{10}^{5}} \delta_{15}^{14}$, $\delta_{15}^{8} \xrightarrow{\delta_{10}^{4}} \delta_{15}^{11}$, $\delta_{15}^{10} \xrightarrow{\delta_{10}^{8}} \delta_{15}^{11}$, $\delta_{15}^{14} \xrightarrow{\delta_{10}^{6}} \delta_{15}^{11}$. Thus, state δ_{15}^{14} is also forbidden state. Analogous to δ_{15}^9 and δ_{15}^{11}, we have $\delta_{15}^{14} \xrightarrow{\delta_{10}^{7}} \delta_{15}^{5}$, $\delta_{15}^{14} \xrightarrow{\delta_{10}^{9}} \delta_{15}^{15}$, $\delta_{15}^{13} \xrightarrow{\delta_{10}^{8}} \delta_{15}^{14}$.

Based on the above analysis, we construct matrix

$$\tilde{F}_{c_1}=[\tilde{F}_1^{c_1},\tilde{F}_2^{c_1},\cdots,\tilde{F}_{15}^{c_1}], \tag{23}$$

where

$$\tilde{F}_i^{c_1}=\begin{cases}0_{15\times 10}, & i=9,11,14\\ \tilde{F}_i, & \text{otherwise.}\end{cases} \tag{24}$$

Consequently, system (22) becomes

$$x(t+1)=\tilde{F}_{c_1}x(t)u(t), \tag{25}$$

The next step to preserve nonblockingness of the controlled DES (25). Using Proposition 3.1, we can obtain easily that $\mathrm{R}(\delta_{15}^1)=\{\delta_{15}^1,\delta_{15}^2,\delta_{15}^3,\delta_{15}^4,\delta_{15}^5,\delta_{15}^6,\delta_{15}^7,\delta_{15}^8,\delta_{15}^{10},\delta_{15}^{13}\}$, $\tilde{R}(\delta_{15}^4)=\{\delta_{15}^1,\delta_{15}^2,\delta_{15}^3,\delta_{15}^4,\delta_{15}^5,\delta_{15}^6,\delta_{15}^7,\delta_{15}^{10},\delta_{15}^{12},\delta_{15}^{13},\delta_{15}^{15}\}$. We know that states δ_{15}^8, δ_{15}^{12} and δ_{15}^{15} are forbidden states. Thus, let

$$\tilde{F}_{c_2}=[\tilde{F}_1^{c_2},\tilde{F}_2^{c_2},\cdots,\tilde{F}_{15}^{c_2}], \tag{26}$$

where

$$\tilde{F}_i^{c_2} = \begin{cases} 0_{15\times 10}, & i = 8,12,15 \\ \tilde{F}_i^{c_1}, & \text{otherwise.} \end{cases} \tag{27}$$

Hence, the controlled system (25) becomes

$$x(t+1) = \tilde{F}_{c_2} x(t) u(t), \tag{28}$$

Again, using Lemma 2.1, we rewrite system (28) is

$$x(t+1) = \tilde{F}_{c_3} u(t) x(t), \tag{29}$$

where $\tilde{F}_{c_3} = \tilde{F}_{c_2} W_{[10,15]}$, specifically,

$$\begin{aligned} \tilde{F}_{c_2} = \delta_{15}[&0,0,0,1,2,3,0,0,0,0,0,0,0,0,0,4,5,6,0,0,0,0, \\ &0,0,0,0,0,0,0,0,0,0,0,7,0,0,0,0,0,0,0,0,0,0, \\ &0,0,0,0,0,0,0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0, \\ &0,0,0,0,13,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0, \\ &0,10,0,0,0,0,0,0,0,0,0,0,0,0,0,0,4,0,0,2,0, \\ &0,5,0,0,0,0,0,0,0,0,0,0,0,0,3,0,0,6,0,0,0,0, \\ &0,0,0,0,0,0,0,0,1,0,0,4,0,0,0,0,0,0,0,0,0]. \end{aligned}$$

It is readily verified that the controlled DES (29) satisfies the requirements described above. Consequently, the cruise trajectory of submarines A and B, by resorting to the algebraic representation (29), is depicted in Fig. 4.

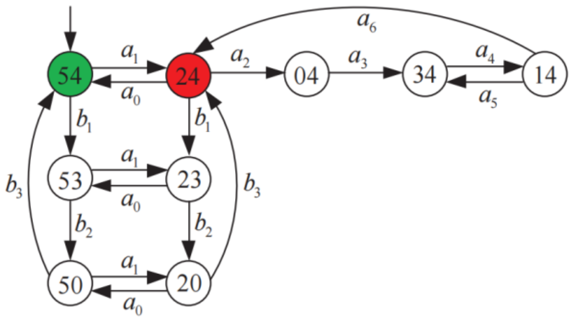

Fig. 4. The cruise trajectory of submarines *A* and *B* in the submarine system.

5. Conclusion

In this paper, we used matrix-based form to model the dynamics of controlled nondeterministic DESs. Using it, we investigated the verification of nonblockingness property of controlled nondeterministic DESs. Also, the criteria of verifying nonblockingness was presented in terms of the methodology described in this paper. Furthermore, the proposed approach can be applied to enforce nonblockingness of controlled nondeterministic DESs.

We hope that our work can establish the foundation for a new theory of supervisory control of controlled nondeterministic DESs. Future work will investigate the problem of synthesizing property-enforcing supervisor for controlled nondeterministic DESs by means of the proposed theoretical framework.

Acknowledgements

This work was supported in part by the Natural Science Foundation of China under Grant 61903274.

References

1. Y. F. Chen, Z. W. Li, K. Barkaoui, et al., "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," IEEE Trans. Syst., Man, Cybern., Syst., vol. 47, no. 2, pp. 364–379, Feb. 2017.
2. C. G. Cassandras and S. Lafortune, Introduction to Discrete Event Systems, 2nd Ed. New York: Springer, 2008.
3. A. Giua and M. Silva "Modeling, analysis and control of discrete event systems: A Petri net perspective," in Proc. the 20th IFAC World Congress, Toulouse, France, Jul. 2017, pp. 1808–1819.
4. S. Abdelwahed and W. M. Wonham, "Blocking detection in discrete event systems," American Control Conference, Denver, USA, Jun. 2003, pp. 1673–1678.
5. D. Z. Cheng, H. S. Qi, and Y. Zhao, An Introduction to Semi-tensor Product of Matrices and Its Applications. Singapore: World Scientific, 2012.
6. X. G. Han, Z. Q. Chen, Z. X. Liu, et al., "Calculation of siphons and minimal siphons in Petri nets based on semi-tensor product of matrices," IEEE Trans. Syst., Man, Cybern., Syst., vol. 47, no. 3, pp. 531–536, Mar. 2017.
7. X. R. Xu, Y. G. Hong, and H. Lin, "Matrix approach to simulation and bisimulation analysis of finite automata," in proc. 10th Word Congress on Intelligent Control and Automation, Beijing, China, Jul. 2012, pp. 2716–2721.
8. X. R. Xu and Y. G. Hong, "Matrix expression and reachability analysis of finite automata," J. Control Theory Appl., vol. 10, no. 2, pp. 210–215, 2012.

Authors Introduction

Mr. Xiang Ren

He received his B.S. degree from College of Electronic Information and Automation, Tianjin University of Science and Technology, China in 2021. He is currently a Master course student in Tianjin University of Science and Technology.

Ms. Zipei Wang

She received the B.S. degree from College of Electronic Information and Automation, Tianjin University of Science and Technology, China in 2023. She is currently a Master course student in Tianjin University of Science and Technology.