Modeling an Optimal Supervisor of Multi-Events and Multi-States Discrete-Event Systems with Automata

Shunping JI, Lyu JIN

College of Mechanical and Electrical Engineering, Sanjiang University, Nanjing, China
School of Information Engineering, Chang'an University, Xi'an 710064, China

***Abstract* — A model called a generator is developed for a multi-events and multi-states discrete-event system using vectors and matrices. Languages generated from the model represent processes of the system. In order to adapt to the compound transitions, the controllability of transitions is sorted into three kinds: controllable, uncontrollable and partly controllable. We also present controllability of languages. And RW supervisor theory is adopted to prohibit the undesired controllable and partly controllable transitions to meet the needs of expected behaviors of the system. Our proposed model is verified to be feasible and efficient through two examples.**

Keywords - Discrete-event system; multi-events and multi-states; automata; supervisory control

I. INTRODUCTION

A DES is a system which is discrete in time and in state transitions. And in DES, different states transit to others with the happening of events. A multi-events and multi-states (MEMS) DES is a system which could be at multiple states and with multiple events happening at the same time.

Many researches have been carried out on discrete-event system (DES). Ramadge and Wonham (RW) introduced the theory of supervisory control of DES [1-2]. Every discrete event is described by a letter and different letters construct an alphabet. A DES is usually modeled as a generator, which will generator languages, which are strings containing some letters from the alphabet. Language is used to represent the process of the DES. The supervisory control uses a supervisor to supervise the generator to get the desired behaviors.

Some subsequent works on DES are focused on timed DES (e.g. Brandin and Wonham 1994[3]; Zhang, Cai and Gan 2013[4]). Timed DES has time dimension and make the system have time properties. Another subsequent work on DES is hierarchical control. The hierarchical control with modularity are effective ways to deal with a distributed systems (e.g. Leduc, Lawford and Dai 2006[5]; Feng and Wonham 2008[6]; Schmidt, Marchand and Gaudin 2006[7]; Liu and Lin 2013[8]).

For distributed systems such as flexible manufacturing systems (FMS) and communication networks, modular control architecture can be more suitable than a monolithic centralized one. Takai [9] and Su [10] studied the DES with concurrent events. In recent years, state tree structures (STS) are used to describe DES (e.g. Ma and Wonham 2003 [11]).

All those achievements assume that the system could be only at one state at the same time and also only one event could happen at the same time.

Petri net has been used to analyze DES for many years [12-14]. Li, Branislav and Zhou [12, 13] concentrate on Petri nets and their use in the modeling and control design for DES, which serve as a basis for extending to other tools and approaches such as supervisory control theory and other functions. On the other hand, vectors and matrix are used to describe and control the manufacturing systems (e.g. Bogdan, Lewis and Kovacic 2006[15]; Lennartson, Basile, and Miremadi[16]). Lennartson, Basile, and Miremadi[16] established a generic state-vector transition (SVT) model and developed a synthesis procedure to guarantee a controllable, nonblocking and maximally permissive supervisor. Using vectors and matrix is a way to analyze multi-states DES, but it's not based on languages.

II. MODELING OF MEMS DES

Now we model the multi-events and multi-states (MEMS) DES. The DES is modeled as a 5-tuple set.

Let

$$G = (V, \Sigma, \Delta, v_o, V_m) \tag{1}$$

Here V is the state vector set. Let $v \in V$, then v is a state vector. DES includes some states and the dimension of v will be equal to number of states. Let *n* be the number of DES states, then the dimension of v will be *n*, which is at most countable.

Lemma 1: The amount of the elements in V is not larger than 2^n.

Let a DES have 3 states, v = (1, 0, 0) means the system is at the first states. And v = (1, 0, 1) means the system is at the first and the third states at the same time. According to the permutation theory, a DES with 3 states has 8 kinds of permutations. That is to say, in V there are no more than 8 state vectors, which agree with Lemma 1. Actually, the amount of the elements in V is far less than 2^n, and only some concerned state vectors are selected to form V.

$v_o \in V$ is the initial state vector. And $V_m \subseteq V$ is the set of marked state vectors.

Δ is the set of transition matrixes. Let n be the dimension of v and $v \in V$,

$$\Delta := \{ \begin{bmatrix} e_{1,1} & e_{1,2} & \cdots & e_{1,n} \\ \cdots & \cdots & e_{i,j} & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ e_{n,1} & \cdots & \cdots & e_{n,n} \end{bmatrix} \mid e_{i,j} = 1 \text{ or } e_{i,j} = 0 \}. \quad (2)$$

Let $\delta \in \Delta$, then the values of all elements in δ are Boolean. So, δ is a Boolean matrix. Here use 1 for TRUE and 0 for FALSE. If the element at row *i* and column *j* is 1, that means the transition from state *i* to *j* is defined here. The event $e_{i,j}$ is also abbreviated as e_{ij}.

Let ε be an identity matrix with the dimension $n \times n$, ε is special element in Δ, which represents the transition from a state vector to itself.

Definition 1. Let $v = (a_1, a_2, \ldots, a_n)$,

$$\delta = \begin{bmatrix} b_{11} & b_{12} & \cdots & b_{1n} \\ b_{21} & b_{22} & \cdots & \cdots \\ \cdots & \cdots & \cdots & \cdots \\ b_{n1} & \cdots & \cdots & b_{nn} \end{bmatrix},$$ then the operation

$$v \times \delta := (a_1 \times b_{11} + a_2 \times b_{21} + \ldots + a_n \times b_{n1},\ a_1 \times b_{12} + a_2 \times b_{22} + \ldots + a_n \times b_{n2},\ \ldots,\ a_1 \times b_{1n} + a_2 \times b_{2n} + \ldots + a_n \times b_{nn}). \quad (3)$$

The operator "$\times$" means "AND", the "+" means "OR". The expression $v \times \delta$ could be written as $v\delta$ for abbreviation.

Lemma 2. Let $v \in V$, $\delta \in \Delta$ is related to v, then $v\delta \in V$.

From Definition 1 and Lemma 2, Δ constructs a partial function: $V \times \Delta \to V$. The function is partial, which means that part of the transit matrixes are defined in Δ.

Σ is a finite alphabet of symbols used to represent the transitions. The natural alphabet is from the names of elements in Δ. So, $\Sigma := \{$ 'δ_i' $\mid \delta_i \in \Delta$, *i*=1,2,3,…}.

G will generate languages on the alphabet Σ. The languages generated by the *G* represent the processes of the transitions, which happen in series. The languages generated by *G* is called *L*(*G*). And those *L*(*G*) from the initial state vector to the marked state vector construct $L_m(G)$.

Example 1. Modeling and supervisor of a machine. The machine has 3 states: idle (state 1), working (state 2) and defunctioned (state3), shown as Fig.(1). And the machine has 4 events. Event 12 (e12 in abbr.) means start, which is a controllable event; Event 21 (e21 in abbr.) means complete the work or abort the work, which is an uncontrollable event; Event 23(e23 in abbr.) means some faults happen, which is also an uncontrollable event; And event 31 (e31 in abbr.) means the job of repair completed, which is controllable event. The 3rd state is different to the RW example. Here, when the machine is defunctioned, it still could work, which like that we often drive a car with some unserious faults.

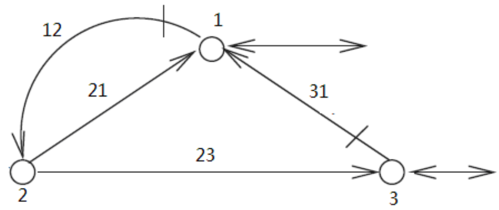


Figure 1. The process of a machine

TABLE I. STATE VECTORS AND TRANSITIONS

NO.	States and Events			
	v_b	v_a	δ	*Comment of* δ
1	v_o	v_1	δ_1	e12
2	v_1	v_o	δ_2	e21
3	v_1	v_2	δ_3	e21 and e23
4	v_2	v_3	δ_4	e12(with fault)
5	v_3	v_2	δ_5	e21(with fault)
6	v_2	v_o	δ_6	e31
7	v_3	v_1	δ_7	e31
8	v_3	v_o	δ_8	e21 & e31

We model the machine as a generator *Mach*. $Mach = (V, \Sigma, \Delta, v_o, V_m)$; $V = \{v_o, v_1, v_2, v_3\}$, $v_o = (1,0,0)$, $v_1 = (0,1,0)$, $v_2 = (1,0,1)$, $v_3 = (0,1,1)$; $V_m = \{v_o, v_2\} \subset V$; Δ is the set of transition matrixes. Let $\delta_i \in \Delta$, *i*=1,2,3….,8 , then $\Sigma := \{$'δ_1', 'δ_2','δ_3',…'δ_8' }. The symbols v_b and v_a represent the state vector before transition and the one after transition.

A transition is transit matrix, which is associated with its state vector. For instance, the value of δ_1 is the same as δ_4 in form, but they are associated with different state vector, so

DOI 10.5013/IJSSST.a.17.38.01

ISSN: 1473-804x online, 1473-8031 print

they are different transitions. From the comment of δ, we can present the value of every transition δ:

$$\delta_1=\delta_4=\begin{bmatrix}0&1&0\\0&0&0\\0&0&0\end{bmatrix};\delta_2=\delta_5=\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix};$$

$$\delta_3=\begin{bmatrix}0&0&0\\1&0&1\\0&0&0\end{bmatrix};\delta_6=\delta_7=\begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix};$$

$$\delta_8=\begin{bmatrix}0&0&0\\1&0&0\\1&0&0\end{bmatrix}. \quad (4)$$

If the system is at v_o, e12 happens, $v_b=v_o$, v_a = $v_b\delta_1$=(0,1,0)= v_1 $\in$V; If the system is at v_1, e12 and e23 happen, v_b =v_1, v_a = $v_b\delta_3$=(1,0,1)= v_2 $\in$V.

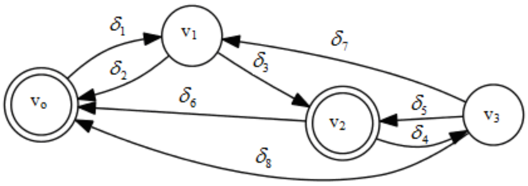


Figure 2. State vectors and transitions of the machine

State vectors and transitions of the machine are illustrated in Figure 2. The state vector with double circles is the marked state vector. State vector v0 is the initial vector, from which the process begins and goes toward the marked state vectors. The paths that the system follows construct strings with the names of transitions. The strings are so-called languages.

The languages generated by *Mach* is *L*(*Mach*),which form a set named *L*(*Mach*) , and *L*(*Mach*)={' ε ',' δ_1 ', ' δ_1 δ_2 ',' δ_1 δ_2 δ_1 δ_2 ', ' δ_1 δ_3 ', ' δ_1 δ_3 δ_4 ', ' δ_1 δ_3 δ_4 δ_5 δ_6 ' ,…}. The marked languages generated by *Mach* are in L_m(*Mach*), and L_m(*Mach*)={ ' ε ', ' δ_1 δ_2 ', ' δ_1 δ_2 δ_1 δ_2 ', ' δ_1 δ_3 δ_4 δ_5 δ_6 ', ' δ_1 δ_3 δ_4 δ_5 δ_8 ' ,…}.

Lemma 3. Let ' δ_i δ_j … δ_k ' $\in$ *L*(*G*), *i*,*j*,*k*$\in$ {1,2,3,…}, and $v_o \in V$ be the initial state vector, after the process illustrated by this language, the system will be at the state vector v and $v=v_o$ δ_i δ_j … δ_k .

III. CONTROLLABILITY

A. *Controllability of transitions*

Controllability is a concept associated with the system. Whether events happen or not are decided by the system, then those events are controllable events. Otherwise, they are uncontrollable events.

Definition 2. Let $\delta \in \Delta$, if all the events associated with δ are controllable, then δ is a controllable transition; if all the events associated with δ are uncontrollable, then δ is a uncontrollable transition; if the events associated with δ include both controllable events and uncontrollable events, then δ is a partly controllable transition.

Definition 3. δ is a single transition if there is only one element valued '1' in the matrix; δ is a compound transition if δ can be divided into other transitions; If *power* (δ) is defined, δ is self-closed. If each δ in Δ is self-closed, then *G* is transition self-closed.

For example, let δ be a compound transition and be divided into 3 single transitions: δ^1 , δ^2 and δ^3 , then power (δ)= {($\delta^1,\delta^2,\delta^3$), (δ^1,δ^2), (δ^1,δ^3), (δ^2,δ^3), δ^1 , δ^2 , δ^3 }. (5)

Of course, $\delta = \delta^1 + \delta^2 + \delta^3$. If all the elements (transitions) are defined, δ is called self-closed.

In fact, every self-closed compound transition could be divided into the single ones. A single transition means only one event happens. And a compound transition means more than one event happen at the same time, which represents a parallel process.

Lemma 4. Let δ be a partly controllable transition ($\delta \in \Delta$) and be a compound transition, then δ could be divided to two parts: δ_c (controllable part) and δ_u (uncontrollable part). And $\delta=\delta_c+\delta_u$. Let $v \in V$, which a vector associated with δ, then $v\delta=v\delta_c+v\delta_u$. If δ is self-closed, $\delta_c \in \Delta$ and $\delta_u \in \Delta$

With respect to *Mach* in Exampe1, δ_8 is a partly controllable transition and is could be divided into two parts: δ_5 and δ_7 . The δ_5 is the controllable part, which is $\begin{bmatrix}0&0&0\\0&0&0\\1&0&0\end{bmatrix}$. And δ_7 , which is $\begin{bmatrix}0&0&0\\1&0&0\\0&0&0\end{bmatrix}$, is the uncontrollable part.

From Lemma 4, two parts of a partly controllable transition could happen at the same time or respectively. If one part of δ happens, δ is degraded to δ_c or δ_u . If the δ_c of δ is forbidden, then δ is forbidden, but δ_u is

DOI 10.5013/IJSSST.a.17.38.01 ISSN: 1473-804x online, 1473-8031 print

permitted. However, δ_u will be thought as another transition in Δ . With respect to Mach in Exampe1, the divided δ_5 and δ_7 from δ_8 are in Δ , δ_8 is self-closed.

For the alphabet Σ , we have the partition $\Sigma = \Sigma_c \bigcup \Sigma_u \bigcup \Sigma_{cu}$. Σ_c is a subset of Σ , with which the transitions associated are controllable transitions; Σ_u is a uncontrollable transition subset; And Σ_{cu} is a subset of compound transitions containing controllable and uncontrollable sub-transitions.

With respect to *Mach* in Exampe1, Σ_c ={ 'δ_1' , 'δ_4' , 'δ_6' , 'δ_7' ,...}, Σ_u ={ 'δ_2' , 'δ_3' , 'δ_5' , ...} and Σ_{cu} ={ 'δ_8' ,...}.

B. *Controllability of a language*

We follow the RW's symbols in the definition about controllability [1-3]. But the implication of those symbols may be extended.

Definition 4. Let a language $K \subseteq \Sigma^*$, then K is controllable (with respect to G) if

($\forall$ 'l', 'δ_j') 'l' $\in \overline{K}$ & 'δ_j' $\in \Sigma_u$ & '$l\delta_j$' $\in L(G) \Rightarrow$ '$l\delta_j$' $\in \overline{K}$.

In other words, K is controllable if and only if no $L(G)$-string that is already a prefix of K, when followed by an uncontrollable transition in G, thereby exits from the prefixes of K: the prefix closure $\overline{K}$ is invariant under the occurrence in G of uncontrollable events.

With respect to *Mach* in Exampe1, $\overline{L_1}$ = {'$\delta_1 \delta_3 \delta_6$'} is not controllable, since'$\delta_1 \delta_2$' consists of a prefix 'δ_1' of $\overline{L_1}$ followed by an uncontrollable event 'δ_2' such that '$\delta_1 \delta_2$' belongs to L(*Mach*) but not to $\overline{L_1}$. On the other hand L_2 = {'$\delta_1 \delta_2$', '$\delta_1 \delta_3 \delta_6$'}is controllable, since none of its prefixes 'δ_i' $\in \overline{L_2}$ = { 'ε', 'δ_1', '$\delta_1 \delta_2$', '$\delta_1 \delta_3$', '$\delta_1 \delta_3 \delta_6$'} can be followed by an uncontrollable transition δ_j , such that '$\delta_i \delta_j$' belongs to $L(Mach) - \overline{L_2}$.

IV. SUPERVISOR

A. *Structure of the supervised MEMS DES*

MEMS DES has been modeled as a generator. The languages generated by G are called $L(G)$. And those $L(G)$ from the initial state to the marked states construct $L_m(G)$, which represent some completely process of the system. If we want that G generates the desired languages to fulfill the expected process, a supervisor is needed to permit the desired transitions and prohibit the undesired ones. Here we continue to adopt RW supervisory frame [1-3], shown in Figure 3.

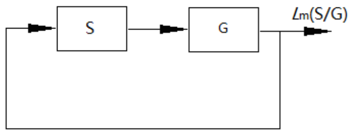


Figure 3. Structure of the supervisory MEMS DES

In Figure 3, S is the supervisor to supervise the generator G. It's convenient to presume that all the uncontrollable transitions are permitted naturally, for the system can't control the uncontrollable transitions and just let them go. For the controllable transitions, the supervisor S could prohibit all or part of the transitions. For the partly controllable transitions, the supervisor S could prohibit them happening. If they are permitted, it doesn't mean they will happen. A permitted partly controllable transition is like an uncontrollable transition. A prohibited partly controllable transition is like a controllable transition.

If a compound transition contains uncontrollable transitions, it had better to define all the sub-transitions in the system. Otherwise, it means the analyst pays his attention to the low probability of transition and ignores the higher ones. For example that two uncontrollable transitions happening at the same time is of a low probability. If a compound transition contains more undefined controllable transitions, the compound transition will be permitted or prohibited as a whole.

Definition 5. Let $L_1 \subseteq \Sigma^*$, $L_2 \subseteq \Sigma^*$, the meet of L_1 and L_2 is denoted as $L_1 \cap L_2$. Let $l_1 \in L_1$, $l_2 \in L_2$, result of $l_1 \cap l_2$ is the same sub-string from the left to right in l_1 and l_2. The operation of $L_1 \cap L_2$ is to match $L_1$and L_2 to find the common substrings.

For example, L_1={'ad', 'abc'}, L_2={'ab', 'adc'}, $L_1 \cap L_2$={ 'ad', 'ab'}.

The pair (S, G) will be written as S/G to suggest 'G under the supervision of S ' . The marked behavior of S/G is $L_m(S/G) = L(S/G) \cap L_m(G)$. Thus the marked behavior of S/G consists exactly of the strings of $L_m(G)$ that 'survive' under supervision by S [1].

B. *Optimal supervisor of MEMS DES*

The generator G is to represent the regular process of the system. And the supervisor S is used to supervise the generator G to generate the desired languages, which will fulfill additional control goals. That control goals are represented by the specifications. Specifications in DES are languages which are also generated by a generator. The set of languages for specifications is E.

Lemma 5 [1]. Let $E \subseteq \Sigma^*$, be $L_m(G)$-marked, and let $K = \sup\mathcal{C}(E \cap L_m(G))$. If $K \neq \Phi$, there exists a nonblocking supervisory control (NSC) S for G such that $L_m(S/G) = K$.

E is $L_m(G)$-marked :$E = \overline{E} \cap L_m(G)$, which means that E contains everyone of its prefixes that belong to $L_m(G)$. In Lemma 5, $\sup\mathcal{C}(E \cap L_m(G))$ is the upper bound of $E \cap L_m(G)$. $K = \sup\mathcal{C}(E \cap L_m(G))$ is the optimal result under the supervisor S with respect to E. If we compare $L_m(S/G)$ with

DOI 10.5013/IJSSST.a.17.38.01

ISSN: 1473-804x online, 1473-8031 print

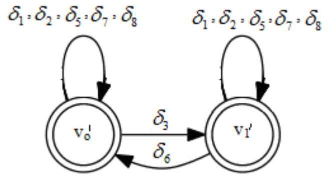
$L_m(G)$, the function of the supervisor will be discovered. If $L_m(S/G) \subset L_m(G)$, some characters in the language are pro hibited. That is to say some transitions are prohibited.

V. EXAMPLES

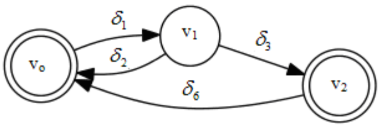
A. *Example1 (continued): Modeling and supervisor of a machine.*

The model of the machine has been established above. Here the example 1 will be continued. A specification will be requested, and then the optimal supervisor of the machine could be figured out according to Lemma 5.

The specification here is to stop the system when fault happens. The working is not permitted when system is with faults.



(a)Specification *E*



(b) Supervised machine Lm(S/G)

Figure 4. Specification and supervised machine

The process of machine has been modeled as a generator *G* shown as Figure 2. Then Specification is modeled as another generator named *E* shown as Figure 4(a). And the optimal supervised machine is $L_m(S/G)= \sup \mathcal{C}\,(E \cap L_m(G))$, shown as Figure 4(b).

B. *Example 2: A parallel process plant*

A Parallel process plant has two parallel branches, shown in Figure 5. At state 1, the system will transit to state 2 and state 5 simultaneously. Then two branches will go respectively from state 2 to state 4 and from state 5 to state 7. The two parallel branches will meet together at last and the earlier arriver needs wait for the later. When the system is at state 4 and state 7, it will transit to state 1 simultaneously.

We model the parallel process as a generator named *Plant* .We present it as $Plant = (V, \Sigma, \Delta, v_o, V_m)$ in which $V = \{v_o, v_1, v_2, v_3, ..., v_9\}$, $v_o = (1,0,0,0,0,0,0)$, $v_1 = (0,1,0,0,1,0,0)$, $v_2 = (0,1,0,0,0,1,0)$, $v_3 = (0,1,0,0,0,0,1)$, $v_4 = (0,0,1,0,1,0,0)$, $v_5 = (0,0,1,0,0,1,0)$, $v_6 = (0,0,1,0,0,0,1)$, $v_7 = (0,0,0,1,1,0,0)$, $v_9 = (0,0,0,1,0,0,1)$), $V_m = \{v_o\} \subseteq V$. Δ is the set of transition matrixes. Let $\delta_i \in \Delta$, i=1,12,14,….,90 , then Σ :={‘ δ_1 ’, ‘ δ_{12} ’, ‘ δ_{14} ’, ‘ δ_{15} ’, ‘ δ_{23} ’, ‘ δ_{25} ’, ‘ δ_{26} ’, ‘ δ_{36} ’, ‘ δ_{45} ’, ‘ δ_{47} ’, ‘ δ_{48} ’, ‘ δ_{56} ’, ‘ δ_{58} ’, ‘ δ_{59} ’, ‘ δ_{69} ’, ‘ δ_{78} ’, ‘ δ_{89} ’, ‘ δ_{90} ’ }.

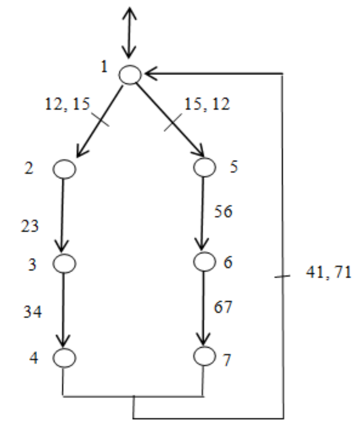
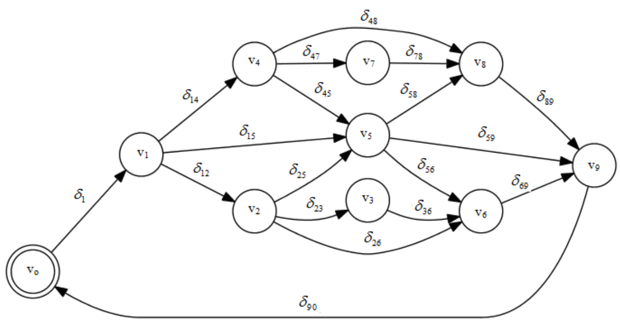
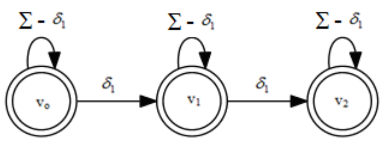


Figure 5. A parallel process plant

The generator Plant is illustrated in Figure 6(a). The specification here is to limit the times of system loops to be less than or equal to twice. So the transition δ_1 will happen no more than twice. The third one will be prohibited by the supervisor *S*. The model of the specification is shown in Figure 6(b). And the languages of the optimal supervised plant are $L_m(S/G)= \sup \mathcal{C}\,(E \cap L_m(G))$, shown as Figure 6(c).

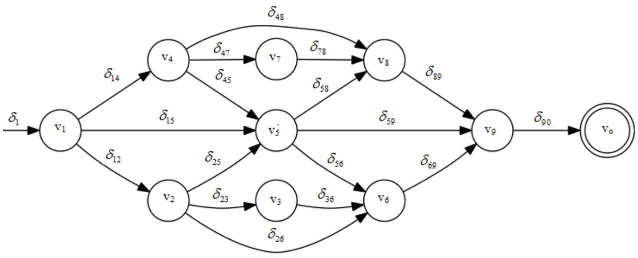


(a) The generator of plant



(b) The generator of specification E

DOI 10.5013/IJSSST.a.17.38.01 ISSN: 1473-804x online, 1473-8031 print

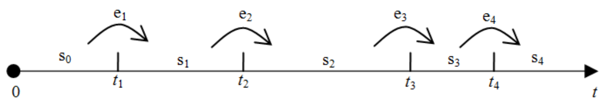


(c) The supervised generator of plant $L_m(S/G)$

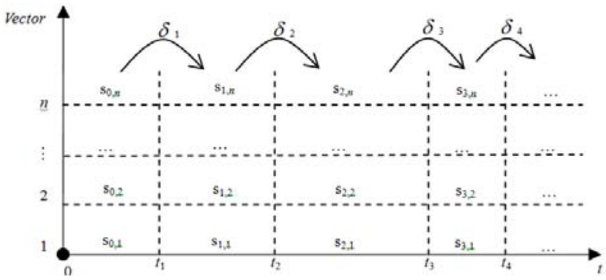
Figure 6. Optimal supervisor control of the parallel process plant

VI. CONCLUSIONS

The multi-events and multi-states (MEMS) DES is a more common kind of DES compared with the RW DES. However, the model and the control theory presented in this paper are extended from RW theory. In RW DES model, the system could be only at one state at the same time and also only one event could happen at the same time, shown in Fig7.a. When more events happen, it should be considered as one event. It is reasonable in theory, but it is not easy to analyze the MEMS DES with a universal model.



(a) RW model of DES



(b) Model of MEMS DES

Figure 7. Comparison of RW model and MEMS model

MEMS model in this paper has 2 extensions. The first one is to extend the states from the single one to a vector with all states. The second one is to extend the transition from single event to transition matrix. Those two extensions help the system deal with problems with MEMS DES.

However, the concepts of language, controllability and supervisor are reserved. The control of MEMS DES is still based on languages. So the model of the MEMS DES is inherited from the RW DES, but with a significant progress.

We also conclude that a MEMS DES has three essential factors: state vectors, transitions and languages. A state vector is used to represent states of a system, which has duration in time. A transition will happen instantly to transfer the system from one state vector to another one. And a language shows the process of the system.

Later we will study the hierarchical control of MEMS DES and the timed MEMS DES.

ACKNOWLEDGMENT

This work was financially supported by Natural Science Foundation of Higher Education of Jiangsu Province, China (15KJD510010).

REFERENCES

[1] W. M. Wonham. Supervisory control of discrete-event systems, http://www.control.toronto.edu/DES, Updated 2015.07.01.

[2] P. J. Ramadge, W. M. Wonham, "Supervisory control of a class of discrete-event processes", SIAM Journal of Control and Optimization, vol.25, no.1, pp.206-230, 1987.

[3] B. A. Brandin, W. M. Wonham, "Supervisory control of timed discrete-event systems", IEEE Trans. Autom. Control, vol.39, no.2, pp. 329-342, 1994.

[4] R. Y. Zhang, K. Cai and Y. Gan, "Supervision localization of timed discrete-event systems", Automatica, vol.49, no.9, pp.2786-2794, 2013.

[5] R. J. Leduc, M. Lawford and P. Dai. Hierarchical interface-based supervisory control of a flexible manufacturing system. IEEE Trans. Control Systems Technology ,vol.14, no.4, pp. 654-668, 2006.

[6] L. Feng, W. M. Wonham, "Supervisory control architecture for discrete-event systems", IEEE Trans. Autom. Control, vol53, no.6, pp.1449-1461, 2008.

[7] K. Schmidt, H. Marchand and B. Gaudin, "Modular and decentralized supervisory control of concurrent discrete event systems using reduced system models", Discrete Event Systems, 8th International Workshop on. IEEE, pp.149-154, 2006.

[8] F. Liu, H. Lin, "Reliable decentralized supervisors for discrete-event systems under communication delays: existence and verification", Asian Journal of Control, vol.15, no.5, pp.1346-1355, 2013.

[9] S. Takai, T. Ushio, "Supervisory control of a class of concurrent discrete event systems", IEICE Trans. Fundam., vol.87, no.3, pp. 850-855, 2004.

[10] R. Su, "Supervisory control of concurrent discrete-event systems", 50th IEEE Conference on Decision and Control and European Control Conference, pp.1811-1816, 2011.

[11] C. Ma, W. M. Wonham, "Nonblocking supervisory control of state tree structures", IEEE Trans. Autom. Control, vol.51, no.5, pp.782-793, 2006.

[12] Z. W. Li, M. C. Zhou, "Deadlock resolution in automated manufacturing systems: a novel Petri net approach", Springer, 2009.

[13] Hrúz, M. C. Zhou. Modeling and control of discrete-event dynamic systems: With petri nets and other tools, Springer, 2007.

[14] M. dos Santos Soares, "Modeling and analysis of discrete event systems using a Petri net component. Systems", Man, and Cybernetics (SMC), 2011 IEEE International Conf. on, pp.814-819, 2011.

[15] S. Bogdan, F.L. Lewis and Z. Kovacic. Manufacturing Systems Control Design: A Matrix-based Approach, Springer, 2006.

[16] B. Lennartson, F. Basile, and S. Miremadi, "Supervisory Control for State-Vector Transition Models—A Unified Approach", Automation Science and Engineering, IEEE Trans. on, vol.11, no.1, pp.33-47, 2014.