ISSN 2229-5518

Optimal control for discrete event systems applying multi-model approach

Asmae El Ghadouali, Oulaid Kamach, Benaissa Amami

Abstract— This work deals with operating mode management applied to discrete event systems (DES). Studied system presents several operating modes due to the state space of explosion problem for complex systems. We propose a multi-model approach, where each model describes a system in a given operating mode. We assume that only one attempted operating mode is activated at a time whilst other modes must be inactivated. The commutation problem can be defined as compatibility problem when the behavior of the physical system switches from an operating mode to another. The determination of compatible state, in the active mode, based on a succession of generating events in past trace. This past events trace generates more information. Indeed, there are the other traces containing loops and bring to the same accessible state. The treatment of this trace renders our compatibility approach [15] more complex and difficult to apply to real cases. For that, we propose an optimal solution of events trace generated in the mode to be inactivated. We will be, thus, present an algorithm to determine the optimal trace presented same information carried by the accessible current state after generated trace when we change the active mode.

Index Terms— discrete event system, operating mode management, multi-model, switching mode.

1 Introduction

ABSTRATED decomposition in modes is a current method in industry to reduce the complexity of a system and describes it. Several works on DES have attempted to design a complex system through a mode management [1, 2]. However, the operating mode management remains a problem not yet perfectly restrained in the modal decomposition [3- 5]. Some studies have focused on the automaton use for representing modes [6,7] in bearing a reasonable size, while either research deal with the problem of using several models [8, 9]. However, very few takes into consideration the commutation problem between different modes [10, 11].

In the proposed approach, we are interested in modeling these operating modes by applying a multi-model approach [7], which involves the process design model for each operating mode. The commutation between these operating modes takes place when a particular event occurs, called commutation event. This event allows the switching from the mode in which a system performs perfectly in its task, known as nominal mode, to a mode for continuing a task in spite of a failure, known as degraded mode. In order to ensure the alternation between these operating modes, we assume that only one attempted operating mode will be activated at a time, whilst other modes must be inactivated. Thus, the system can only be in one mode, called active mode. Switching from a mode to another mode returning to enable and disable the current mode. The commutation problem can be defined as compatibility problem between different operating modes. The compatibility problem treated as the consistency of current states [9-12] when a mode generates an event activating the other mode. For is sure switching between different operating modes when the observable commutation event occurs, a static algebraic approach has been proposed in ours works [13, 15, 16].

Using a multi-model approach allows us to materialize the commutation mechanism between different modes. The dynamic study of a system based on a succession of generating events in past trace. This past events trace generates more information. Indeed, there are the other sequences containing loops and bring to the same accessible state. The treatment of this trace renders our compatibility approach [15, 16] more complex and difficult to apply to real cases. For that, we propose an optimal solution of events sequence generated in the disable mode. We will be, thus, present an algorithm to determine the optimal trace presented same information carried by the accessible current state after generated trace when we change the active mode. Our algorithm ensures to reduce the complexity treatment to determine the current state in the active mode applying the equation of compatibility proposed in works [15, 16]. Our approach based on mathematical techniques and the theory of finite state automaton.

This paper organized as follows: Section 2 covers the selected design terminology of DES multi-model where the new notions and extended models are introduced. Commutation formalism between designed process models is also briefly recalled in this section. Section 3 introduces our optimal solution for treatment of generated traces activated the current state in the active mode. Study conclusions presented in section 5.

2 Multi-model approach

The system has various operating modes depending on the configuration of engaged resources. An operating mode corresponds to requirements and hence different commands, and different resources. We discuss variable typology of model or multi-model system. We use the automation model to describe a system in a given operating mode where involves the activity of certain resources [1]. The management models have carried out the decisions of resources. It leads resources of the system from a configuration to another. The operating mode and its engaged resources are modeled by the automaton introduced by the following definitions:

Definition 1. *Let* $R_{M_j} = \{R_1, R_2, \ldots R_i\}$ *a set of all resources en-*

gaged in the operating mode M_j, where $i \in \mathbb{N}^*$. A resource R_i is modeled by an automaton $G_{R_i} = (Q_{R_i}, \Sigma_{R_i}, \delta_{R_i}, q_{R_i,0}, Q_{R_i,m})$, where:

- *Q_{R_i} is the activities set of the resource R_i,*
- *Σ_{R_i} is the events set of the resource R_i,*
- *$\delta_{R_i}: Q_{R_i} \times \Sigma_{R_i} \rightarrow Q_{R_i}$ is the transition function;*
  $\forall q \in Q_{R_i}$ and $\forall \sigma \in \Sigma_{R_i}$. We note $\delta_{R_i}(q,\sigma)!$ (resp. $\delta_{R_i}(q,\sigma)\neg!$) si $\delta_{R_i}(q,\sigma)$exist (resp. inexist).
  This function can be extended as follows:
  $\delta_{R_i}: Q_{R_i} \times \Sigma_{R_i}^* \rightarrow Q_{R_i}$ which $\forall s \in \Sigma_{R_i}^*$ and $\forall \sigma \in \Sigma_{R_i}$, $\delta_{R_i}(q, s\sigma) = \delta_{R_i}(\delta_{R_i}(q,s),\sigma)$ and $\forall q \in Q_{R_i}, \delta_{R_i}(q,\varepsilon) = q$,
- *$q_{R_i,0}$ is the initial activities of the resource R_i,*
- *$Q_{R_i,m}$ is the final activities set of the resource R_i, $Q_{R_i,m} \subseteq Q_{R_i}$.*

Definition 2. Let M a set of operating modes, where $M = \{M_1, M_2, \ldots, M_j\}$, $j \in \mathbb{N}^*$. For each operating mode M_j, we associate to an automaton model $G_{M_j} = (Q_{M_j}, \Sigma_{M_j}, \delta_{M_j}, q_{M_j,0}, Q_{M_j,m})$ where:

- Q_{M_j} is the set of states of mode M_j,
- Σ_{M_j} is the alphabet of symbols,
- $\delta_{M_j}: Q_{M_j} \times \Sigma_{M_j} \rightarrow Q_{M_j}$ is the partial transition function.
- $q_{M_j,0}$ is the initial state in the mode M_j, $q_{M_j,0} \in Q_{M_j}$;
- $Q_{M_j,m}$ is the subset of marker states in the mode M_j, $Q_{M_j,m} \subseteq Q_{M_j}$.

The set $\Sigma_{M_j}^*$ contains all possible finite strings (i.e., sequence) over Σ_{M_j} and the null string ϵ. The language generated by G_{M_j}, denoted by $L(G_{M_j})$, is also called the closed behavior of G_{M_j}:

$$L(G_{M_j}) := \{s \in \Sigma_{M_j}^* | \delta_{M_j}(q_{M_j,0}, s)!\} \quad (1)$$

The global set of events, noted by Σ_{global}, of a system is given by the union of all alphabets Σ_{M_j} of elementary automaton models G_{M_j} increasing by the set of commutation events $\Sigma_{global}^{\leftrightarrows}$. Furthermore, the set of commutation events is disjoint of the different set of models : $\Sigma_{global}^{\leftrightarrows} \cap \Sigma_{M_j} = \emptyset$ (for $M_j \in M$). Although, $\Sigma_{M_j} \cap \Sigma_{M_k}$ (with $(j,k) \in (\mathbb{N})^{*2}$ and $j \neq k$) can't be empty because it can be existed the common components between these two modes $M_j\ et\ M_k$.

In the proposed approach, the set $\Sigma_{global}^{\rightleftarrows}$ of commutation events is defined as $\cup_{j,k,j\neq k}^{n}\{\sigma_{M_j,M_k}\}$ where σ_{M_j,M_k} presents the event ensuring the switching between mode M_j to mode M_k . Moreover, from a mode M_j a commutation event σ_{M_j,M_l} can lead to another mode M_l where l can be lower or higher than j (with also $j \neq l$). In other words, the classification of modes doesn't present an activation order. Thus, the commutation event σ_{M_j,M_l} occurs and the process model becomes G_{M_l}.

Definition 3. *We define for each resource R_i the automaton model of normal functioning: $G_{R_i}^{\circlearrowleft} = (Q_{R_i}^{\circlearrowleft}, \Sigma_{R_i}^{\circlearrowleft}, \delta_{R_i}^{\circlearrowleft}, q_{R_i,0}^{\circlearrowleft}, Q_{R_i,m}^{\circlearrowleft})$, with:*

- *$\Sigma_{R_i}^{\circlearrowleft} = \Sigma_{R_i} \setminus \Sigma_{R_i}^{\rightleftarrows}$ ($\Sigma_{R_i} \setminus \Sigma_{R_i}^{\rightleftarrows}$: difference set of Σ_{R_i} and $\Sigma_{R_i}^{\rightleftarrows}$, this is set of events of Σ_{R_i} which don't belong to $\Sigma_{R_i}^{\rightleftarrows}$.};*
  $\Sigma_{R_i}^{\rightleftarrows} = (\Sigma_{R_i} \cap \Sigma_{global}^{\leftrightarrows})$ is the set of commutation events of resource R_i;
- *$Q_{R_i}^{\circlearrowleft} = Q_{R_i} \setminus Q_{R_i}^{\rightleftarrows}$.*
  $Q_{R_i}^{\rightleftarrows}$ is the states set from which the commutation event occurs, such that:
  For each $q \in Q_{R_i}^{\rightleftarrows}, \exists \sigma \in \Sigma_{R_i}^{\rightleftarrows}$such that $\delta_{R_i}(q,\sigma)!$;
  For each $q \in Q_{R_i}^{\rightleftarrows}$ et $\sigma \in \Sigma_{R_i}^{\circlearrowleft}, \delta_{R_i}(q,\sigma)\neg!$.
- *$\delta_{R_i}^{\circlearrowleft}: Q_{R_i}^{\circlearrowleft} \times \Sigma_{R_i}^{\circlearrowleft *} \rightarrow Q_{R_i}^{\circlearrowleft}$ is the transition function. For each $q \in Q_{R_i}^{\circlearrowleft}$ and $s \in \Sigma_{R_i}^{\circlearrowleft *}$ if $\delta_{R_i}(q,s)!$ then $\delta_{R_i}^{\circlearrowleft}(q,s) := \delta_{R_i}(q,s)$;*
- *$q_{R_i,0}^{\circlearrowleft} = q_{R_i,0}$;*
- *$Q_{R_i,m}^{\circlearrowleft} = Q_{R_i,m}$.*

Example 1.

The proposed approach is illustrated by means of a production example in Fig.1. This system comprises five machines R_1, R_2, R_3, R_4, and by one buffer B. The machines are used to process a part and the buffer is used as storage between the machines with a maximal capacity of 1.

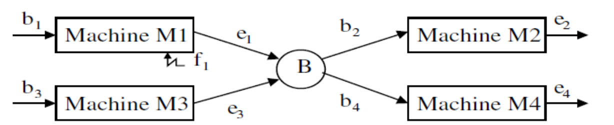

Fig. 1. Production system with four machines and intermediate stock.

In this system, each machine R_i is a resource used to respond to a task defined by the designer. The machines operate independently, where R_i picks up a work piece from an infinite being (modeled by event b_i) and places it in buffer after completing its function (symbolized by the occurrence of event e_i).

The shutdown state of machine R_i is labeled by the activity Ai and the running state by the activity Mi. It is assumed that only R_1 can break down: $P1$ represents a state where R_1 is broken due to malfunction and modeled by the event f_1. Repair of machine R_1 is modeling by the event r_1. The automates models machines R_1 (resp. R_2, $R_3$and R_4) are denoted G_{R_1} (resp. G_{R_2}, $G_{R_3}$and G_{R_4})(see Fig.2(a) (resp.Fig.2(b))). The dotted arrows represent the commutation events. In this article, we assume that the failure commutation events are observable events.

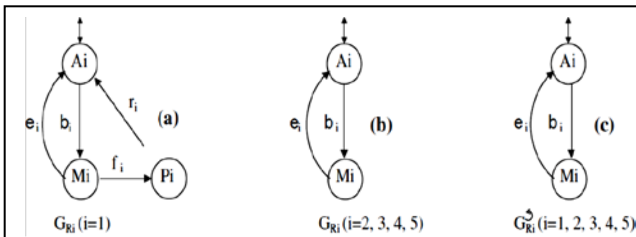

Fig. 2. Automaton models of machines R_i with $i \in \{1,2,3,4\}$.

In the operating mode model G_{M_j}, each state is described by the activities set of resources and each engaged resource have a unique activity in a given state. This will be introduced by the following definition:

Definition 4. *Each state $q_j \in Q_{M_j}$ is the cartesian product of all activities of resources engaged in the mode M_j.*

For example, we assume that two resources R_1 and R_2 are engaged in the mode M_j respectively with $\{a^1_{R_1}, a^2_{R_1}\}, \{a^1_{R_2}, a^2_{R_2}\}$.

The activities of engaged resources in the state q_j is defined by:

$$A_{q_j} = \{a^l_{R_1}, a^k_{R_2}\} \text{ with } l, k \in \{1,2\} \quad (2)$$

Proposition 1. *The automaton model $G_{M_j} = (Q_{M_j}, \Sigma_{M_j}, \delta_{M_j}, q_{M_j,0}, Q_{M_j,m})$ of operating mode M_j is defined by the synchronous product of normal functioning automaton models $G^{\circlearrowleft}_{R_i}$ of the engaged resources $R_i \in R_{M_j}$ in this mode:*

Formellement, $G_{M_j} = \|_{(R_i \in R_{M_j})} G^{\circlearrowleft}_{R_i} \quad (3)$

Example 2. Two operating modes are designed for the overall system treated in the example (see Fig.1): a nominal mode N, in which R_1, R_2 and R_4 produce and a degraded mode D, in which R_3 replaces R_1.

The automaton model of nominal mode G_N and degraded mode G_D, represented in the Fig.3 , are obtained by the synchronous product of nominal automatons $G_{R_i^{\circlearrowleft}}$ modeling machines engaged in the studied mode. The automaton model $G_{R_i^{\circlearrowleft}}$ (see Fig.2(c)) built by removing commutation events $\{f_1, r_1\}$ from the automaton model G_{R_i}.

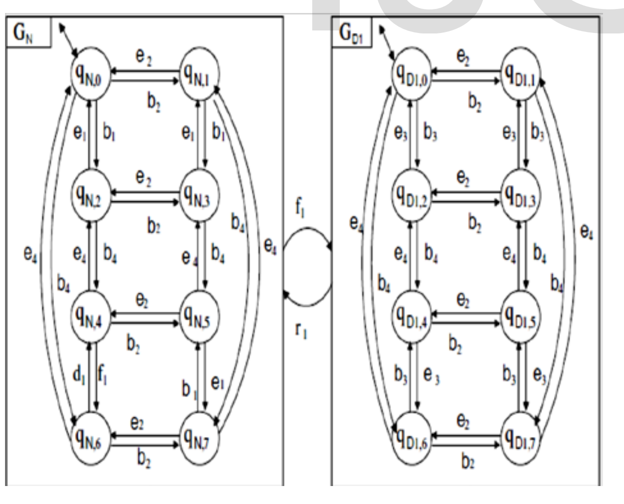

Fig. 3 Automaton models of modes $M = \{N, D\}$

In this example, the set of events Σ_{global} can be partitioned into four sets: $\Sigma_N = \{b_1, e_1, b_2, e_2, , b_4, e_4\}$ alphabet of nominal mode N, $\Sigma_D = \{b_2, e_2, b_3, e_3, b_4, e_4\}$ alphabet of degraded mode D and $\Sigma^{\rightleftarrows}_{global} = \{f_1, r_1\}$ commutation events.

However, in each switching mode M_k, we must correctly determine the start state after the commutation from model G_{M_j}. To do this, we first have to extend the model G_{M_k} (resp. G_{M_j}) by adding respectively inactive state q_{in,M_k} (resp. q_{in,M_j}) to states set of the model G_{M_k} (resp. G_{M_j}) based on the concept of signification state suggested by Dangoumau [1]. We use the inactive state to ensure that the only one mode is active in a time. The occurrence of commutation event σ_{M_j,M_k} will lead model G_{M_j} to its inactive state q_{in,M_j} and the process model G_{M_k} will be activated from its inactive state q_{in,M_k}. Thus, the active process model at a given time is the only model from which the current state is different from the inactive state. Reciprocally, the state of all inactive models is their inactive state.

Definition 5. *So the extended model for each operating mode $M_j \in M$ is given by automaton model $G_{M_j,et} = (Q_{M_j,et}, \Sigma_{M_j,et}, \delta_{M_j,et}, q_{M_j,0,et}, Q_{M_j,m,et})$ in which:*

- $Q_{M_j,et} = Q_{M_j} \cup \{q_{M_j,in}\}$, *extended set states with* $q_{M_j,in}$ *an inactive state ;*
- $\Sigma_{M_j,et} = \Sigma_{M_j} \cup \Sigma^{\rightleftarrows}_{M_j}$ *with* $\Sigma^{\rightleftarrows}_{M_j} = \Sigma^{\leftarrow}_{M_j} \cup \Sigma^{\rightarrow}_{M_j}$ *with* $\Sigma^{\leftarrow}_{M_j}$ *(resp.* $\Sigma^{\rightarrow}_{M_j}$*) is the events set which activates (resp. deactivates) the mode* M_j*;*
- $q_{M_j,0,et} = \begin{cases} q_{M_j,0} & \text{if } j = 1 \text{ is in its initial state} \\ q_{M_j,in} & \text{if } j \neq 1 \text{ are assumed deactivated} \end{cases}$
- $Q_{M_j,m,et} = Q_{M_j,m}$. (q_{in,M_j} *will never be marked state);*
- *The extended transition function* $\delta_{M_j,ext}$ *is defined as follows :*

$\forall q \in Q_{M_j}$ *and* $\forall \sigma \in \Sigma_{M_j}$ *if* $\delta_{M_j}(q,\sigma)!$ *then* $\delta_{M_j,et}(q,\sigma) := \delta_{M_j}(q,\sigma)$. *This extended function is the same as transition function if we consider only non extended alphabet* Σ_{M_j}.

$\forall q \in Q_{M_j}$ *from which commutation event* $\sigma \in \Sigma^{\rightarrow}_{M_j}$ *can occurs, then* $\delta_{M_j,et}(q,\sigma) = q_{M_j,in}$: *extended transition function allows model* G_{M_j} *to be deactivated if the commutation event occurs.*

With regard to the process, the main aim of operating mode management is to define the start state of model G_{M_k} , where the commutation event σ_{M_j,M_k} occurs ($q_{M_k,start} := \delta_{M_k,ext}(q_{M_k,in}, \sigma_{M_j,M_k})$) and, in turn, the return state of process model G_{M_j} ($q_{M_j,start} := \delta_{M_j,ext}(q_{M_j,in}, \sigma_{M_k,M_j})$). The notion of start state defined in our algebraic approach has been proposed in ours works [15, 16].

Example 3. By hypothesis, the occurrence of failure event f_1 of machine R_1 can take place from states in the nominal mode or the degraded mode D where the machines R_1 is in normal functioning. Before continuing, we assume, henceforth that the event return r_1 can occur only from states in the degraded mode D when the machine R_3 has finished its task. Since there is a common resource R_2 and R_4 engaged between the nominal mode and the degraded D mode. Then, the start states in the degraded model are related to traces generated in the nominal model. During a change of operating mode, the activity of common resource remains fixed. To ensure unicity of active current mode, we add in each model G_{M_j}, with $M_j \in \{N, D\}$ the inactive state $q_{M_j,in}$ illustrated in the extended model $G_{M_j,et}$ (see Fig.4). To raise the indeterministic problem, we consider the following terminology: In each extended model $G_{M_j,et}$, the commutation event σ will be replaced by σ_i if only if $\delta_{M_j,et}(q_{M_j,i}, \sigma) = q_{M_j,in}$.

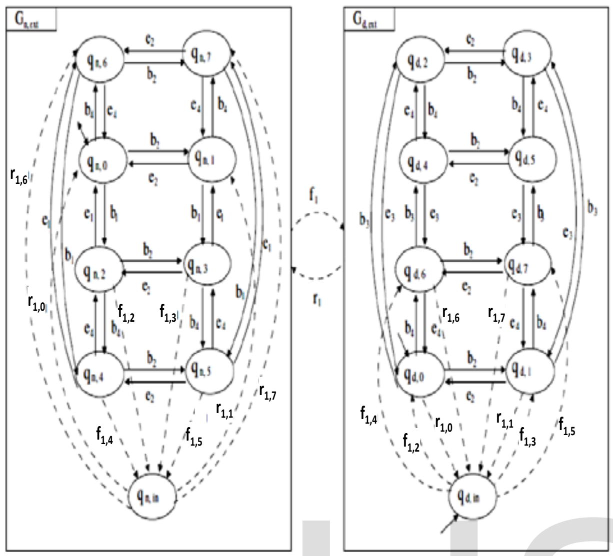

Fig. 4 Extended automaton models of different operating models

3 OPTIMAL SOLUTION

Generally, we can consider the problem of finding a better optimal trace of an achievable strategy as the accessibility problem of a state. In this paper, all automata are considered an accessible model; without loss of generality that is all states are reachable from the initial state. The state trace over time can be mathematically represented by a step function whose value can change whenever an event occurs. Indeed, there are events sequence containing loops bring to the same accessible state. We propose an optimal solution of this sequence arriving at an accessible state in the active mode. We will describe the proposed algorithm to find the optimal events sequence contains the same information as the original activation sequence.

Definition 6. *A sate $q_{acc} \in Q_{M_j,et}$ is accessible state of state $q_{Start} \in Q_{M_j,et}$ if exist the events sequence $S \in \Sigma^*_{M_j,et}$ such that $\delta_{M_j}(q_{Start}, S) = q_{acc}$.*

Definition 7. *The traces set that leave the state q_{Start} arriving at the accessible state q_{acc} is defined by the following language:*

$L(G_{M_j,et}, q_{Start}, q_{acc}) = \{S \in \Sigma^*_{M_j,et} | \delta_{M_j}(q_{Start}, S) = q_{acc}\}$ (4)

To facilitate the calculation, we propose an algorithm automatically generates the optimal sequence to search the accessible state q_{acc} from the state q_{Start}.

Algorithm 1. *Finding an optimal sequence S_{opt}*

Require: $S \in L(G_{M_j}, q_{Start})$, $q_{Start} \in Q_{j,et}$, $Q_{acc} \subset Q_{M_j,et}$

Ensure: S_{opt}, $q_{acc} \in Q_{acc} \subset Q_{M_j,et}$

Initially $S_{opt} = \varepsilon$
$Q_{acc} = \{q_{Start}\}$
$q_0 = q_{Start}$
i=j=k=1

For all event $\sigma_k \in S$ *and* $q_i \in Q_{acc}$
$q_i = \delta_{M_i}(q_{i-1}, \sigma_k)$
For all state $q_j \in Q_{acc}$
If $q_j = q_i$ *then*
$i = j$
$S_{opt} = \sigma_1 . \sigma_2 \dots \sigma_j$
Else $i = i + 1$
$S_{opt} = S_{opt} . \sigma_k$
end if
end for
$k = k + 1$
end for
$q_{acc} = q_i$
Return S_{opt}, q_{acc}

Example 4. We deal as example the manufacturing system presented in the example (see Fig.1). We consider two example of generated sequence in the nominal mode (see Fig.5) in which: $S_1 = b_1 b_2 e_1 e_2 b_1 b_2 b_4$ and $S_2 = b_1 b_2 e_2 e_1 b_1$; $S_1, S_2 \in L(G_N, q_{N,0})$ exists among the possible traces in the nominal model G_N, with the start state being the initial state $q_{Start} = q_{N,0}$. We apply our proposition to determine the optimal sequence S_{opt}. The proposed algorithm determines the following solutions: $S_{1,opt} = b_1 b_2 b_4$ and $S_{2,opt} = b_1$, with $S_{1,opt} \in L(G_N, q_{N,0}, q_{N,5})$ and $S_{2,opt} \in L(G_N, q_{N,0}, q_{N,2})$.

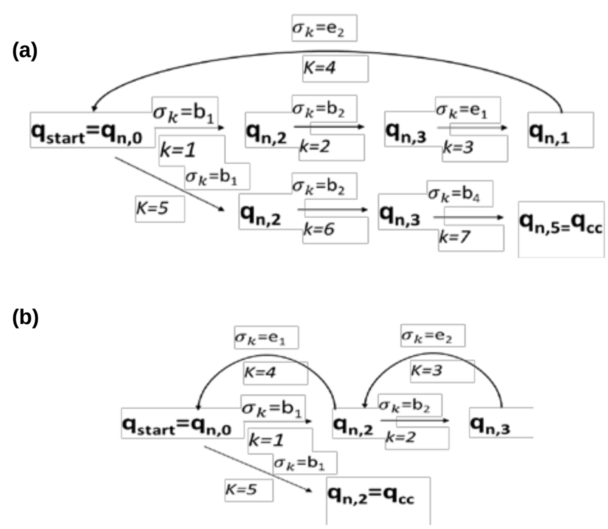

Fig. 5. Example of tow differents sequences d'activation

Lemme 1: Let two different traces $S_{j,1}, S_{j,2} \in L(G_{M_j}, \Sigma^{\rightleftarrows}_{M_j})$ be generated in the current model G_{M_j}. According to the compatibility approach [15], if the two sequences $S_{j,1}$ and $S_{j,2}$ have the same equa-

tion projection in the alphabet Σ_j, then they lead to the same accessible state in the mode to be activated where a switching event occurs.

Lemme 2. Let S_j be an activation sequence generated in the mode M_j belonging to the language $L\left(G_{M_j}, \Sigma_{M_j}^{\rightleftarrows}\right)$, with $: S_{M_j} = s_1 s_2 \ldots (s_l s_{l+1})^n \ldots s_m$. This sequence contains an infinite loop $(s_l s_{l+1})^n$. Applying the proposed approach of optimal trace, the sequence S_{M_j} has same optimal solution than $S'_{M_j} = s_1 s_2 \ldots s_{l-1} s_{l+2} \ldots s_m$. Indeed, since the event s_l (resp. s_{l+1}) activates (resp. deactivates) a given state q_j a given state in the mode M_j.

According to this lemma, even for an infinite activation sequence S_{M_j} generated in the mode to be deactivated M_j and belonging to the language $L\left(G_{M_j}, \Sigma_{M_j}^{\rightleftarrows}\right)$, we can apply our step to determine compatible detailed in our paper [15]. Our approach aims to determine the attainable and optimal trace to minimize computational complexity.

4 Conclusion

This paper deals with the optimal solution of generated events trace in the mode to be deactivate when a switching between operating modes takes place. The determination of compatible state, in active mode, is based on a succession of generated events in past trace. Indeed, this past events trace generates more information. Our approach propose the optimal trace to reduce the complex and difficult to be applied the proposed approach of compatibility to real cases. Our current research is attempting to resolve the diagnosis of unobservable events in real-time in a physical system applying multi-model approach.

References

[1] N. Dangoumau, A.Toguyéi, E. Craye, "Functional and behavioral modeling for dependability in automated production systems", *Journal of engineering manufacture,* vol. 216, pp. 389-405, 2002.

[2] M. Zefran and J. Burdick, " Design of switching controllers for systems with changing dynamics ", *37th Conference on Decision and Control*, pp. 2113-2118, 1998.

[3] M. Nourelfath, " *Extension of the Supervisory Control Theory to the Monitoring and to the Control of Discrete Event Systems : Application to Automated Manufacturing Systems Operational Safety*", PhD thesis, INSA of Lyon, France.

[4] M. Nourelfath and E. Niel, "Modular supervisory control of an experimental automated manufacturing system ", *Control Eng. Pract*., vol. 12, pp. 205-216, 2004.

[5] P. Charbonnaud and F. Rotellaand and S. Mauoar, " Process operating mode monitoring process: switching online the right controller",
IEEE Trans. Control Syst. Tech., vol. 31, 2002.

[6] F. Maraninchi and Y. Rémond, " Mode-automata: a new domain-specific construct for the development of safe critical systems ", Sci. Comput. Program, vol. 1, pp. 219-254, 2003.

[7] J-P. Talpin and C. Brunette and T. Gautier and A.Gamatie, "Polychronous mode automata ", *Proceedings of the 6th ACM \& IEEE International Conference on Embedded Software*, New York, USA, pp. 83-92, 2006.

[8] O. Kamach, L. Piétrac and E. Niel, " Multi-model approach to discrete events systems: application to operating mode management", *Mathematics and Computers in Simulation,* vol. 70, pp. 394-407, 2005.

[9] G. Faraut, L. Piétrac and E. Niel. "Formal Approach to Multimodal Control Design: Application to Mode Switching, *IEEE Transactions on Industrial Informatics,* vol. 5, pp. 443- 453, 2009.

[10] E. Asarin and O. Bournez and T. Drang and O. Maler and A.Pnueli, " Effective synthesis of switching controllers for linear systems ", IEEE. Proc., vol. 88, pp. 1011-1025, 2000.

[11] K. Andersson, B. Lennartson, M. Fabian, " Restarting flexible manufacturing systems: Synthesis of restart states", *Proc. of the 8th International Workshop on Discrete Event Systems*, pp. 201- 206, 2006.

[12] O. Kamach and L. Pi\'etrac and E. Niel, " Repulsive/ Attractive discrete state space sets for switching management ", J. Stud. Inform. Control, vol. 16, pp. 83-96, 2007.

[13] G. Faraut and L. Piétrac and E. Niel, " Identification of incompatible states in mode switching ",*ETFA IEEE Conference*, pp. 121-128, 2008.

[14] A. El ghadouali, O. Kamach and B. Amami, "Identification of Compatible States in Switching Mode ", *International Journal of Computer Applications,* vol. 72, pp. 42-45, 2013.

[15] A. El ghadouali, O. Kamach and B. Amami, " Approche algébrique pour la gestion des modes de fonctionnement ", *Journal Européen des Systèmes Automatisées,* vol. 48, pp. 547 -568, 2014.

[16] A. El ghadouali, O. Kamach and B. Amami, "Static approach for switching between different operating modes ", *IEEE xplore, in Proc. CCCA'12*, Marseilles, France, 2012.