International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

Supervisory Control of Systems Modeled by Petri Net from Adequate Admissible Constraints

Mohaman Gonza[1], Laurent Bitjoka[2]

[1,2] N'Gaoundéré University, National School of Agro-Industrial Sciences (ENSAI) ,
Energy, Signal, Imagery and Automatic Laboratory of) (LESIA), Camreroon

Abstract: ***The purpose of Supervisory Control of the system, considered as Discrete Event Systems (SED), is to synthesize an optimal and non-blocking controller. Indeed, the specifications of the SED are generally declined in forbidden states that must be avoided. Our work consisted in finding a structural synthesis approach based on the determination of the adequate admissible constraints stresses for the Petri net (PN) invariant method. The closed-loop system model (SED) is obtained by synchronous product of elementary PNs. First, we used the computational power of the Kumar algorithm based on finite state automata to determine the constraints related to forbidden states. But, exploration of the PN marking graph is laborious. This is what led us second, to represent the reachable marking graph by its transition matrix. The coding of this matrix, according to the state specification, allowed us to separate the states and identify the states relevant to the synthesis of a controller.***

Keywords: Discrete Event System, Supervisory Control, Finite State Automata, Petri Net, Reachable Marking Graph

1. Introduction

The initial theory of supervisory control based on the finite state automata permits to synthesize a controller, whose role is to impose the respect of the constraints defined by the specification [1]. But, the lack of structure and the combinatorial explosion of states [2], limit the development of simple and efficient synthesis methods for industrial [3]. One of the solutions is the use of Petri nets (PNs) which have a great similarity with automata [4] with a large class of languages [5]. Several formal approaches have been developed. For example: the theory of regions [6], the Holloway theory [7] and the approach based on the colored and labeled PN [8]. However, they are complex and partially structural. But the approach, based on markings invariant [9], is simple and efficient if the adequate set of admissible constraints is provided [10]. However, the existence of uncontrollable events, which the controller cannot forbid, poses a controllability problem. More, additional forbidden states can be generated by the structural synchronization of system PNs [11].

Our work is to propose a strategy for the determination of adequate admissible constraints to ensure the optimality of the controller compute by PN markings invariant [12]. Thus, the transition from synthesis to implementation will be systematic and simple [13]. To achieve our goal, we will consider the labeled PN class (Section II) to express the specification as states or sequences of forbided events [14]. First, we used the power of Kumar's Algorithm [2] to determine constraints related to forbidden states. But, exploration of the PN marking graph is laborious. This led us to secondly to represent the closed-loop system PN marking graph by its transition matrix (Section IV). The coding of the transition matrix [15], according to the specification, allowed us to separate the states of the marking graph and to identify the states relevant to the synthesis of a controller.

2. Modelization by Petri Net

The class of labeled PNs are connected to automata in the sense that they explicitly represent the transition function and offer the possibility of conditioning the firing of a transition to the occurrence of an event [16]; [17].

Definition 1. A labeled Petri net (ℓ-*PN*) defined on the alphabet E is a structure $R = (N, E)$ with $N = (P, T, W^-, W^+, \ell, M_0)$ where: P is a finished set of place; T is a finite set of transitions; E is the set of events associated with transitions; W^- and W^+ the incidence matrices of N indexed by P x $T \rightarrow \mathbb{N}$; $\ell : T \rightarrow E$ is the labeling function of transitions. A sequence of transitions is $\sigma = t_1 t_2 \ldots t_m \in T^*$ where $\ell(\sigma) = \ell(t_1)\, \ell(t_2) \ldots \ell(t_m)$; $M_0 : P \rightarrow \mathbb{N}$ is the initial marking. ❑

Example: Consider a system consisting of two machines and a robot (Figure 1). The machine M1 manufactures type I parts and the machine M2 manufactures parts of type II. When the machine Mi has finished it's machining (uncontrollable event *f_USi*), it can be free. The robot transports the raw part of the stock to the free machine according to the production order (controllable event of*_Pi*). And when the piece is on the machine Mi (uncontrollable event *Ps_Mi*), after the end of the transport (uncontrollable event *f_Tri*), the robot returns to its initial state. Specifications are imposed by the robot.

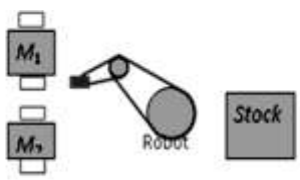
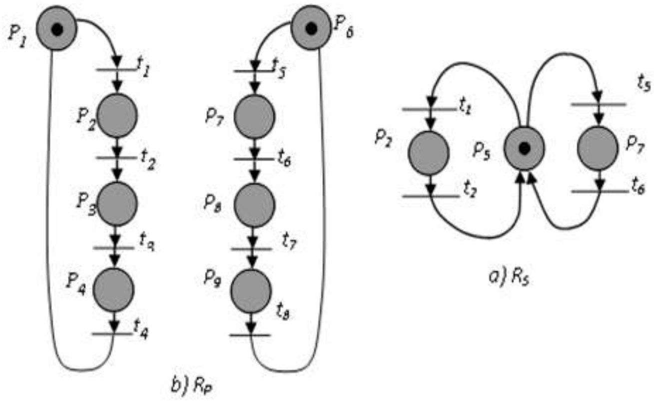


Figure 1: Manufacturing System

The following figure gives the PNs models



P_1	*M1_libre*	t_1	*of_P1*
P_2	*TR1 (Transport Robot-M1)*	t_2	*f_TR1*
P_3	*Pièce sur M1*	t_3	*Ps_M1*
P_4	*US1 (usinage M1)*	t_4	*f_US1*
P_5	*Robot libre*	t_5	*of_P2*
P_6	*M2_libre*	t_6	*f_TR2*
P_7	*TR2 (Transport Robot-M1)*	t_7	*Ps_M1*
P_8	*Pièce sur M2*	t_8	*f_US2*
P_9	*US1 (usinage M2)*		

Figure 2: Petri Nets of system: a) Specification b) Plant

2.1 Synchronous Product of Petri Nets

Let $R_1 = (N_1, E_1)$, $N_1 = (P_1, T_1, W^-_{1,} W^+_1, \ell_1, M_{10})$, defined on E_1 and $R_2 = (N_2, E_2)$, $N_2 = (P_2, T_2, W^-_2, W^+_2, \ell_2, M_{20})$, defined on E_2. The synchronous product of R_1 and R_2, designated $R = R_1||R_2$, is defined on the alphabet E, such that: $P = P_1 \cup P_2$; $T = T_1 \cup T_2 - T_{12}$ with $T_{12} := \{t_i \in T_1 | \exists\ t_j \in T_2$ such that $\ell(t_i) = \ell(t_j)\}$; $W^-(p, t)$: = $\{W^-_1(p, t)$ si $p \in P_1$ or $W^-_2(p, t)$ si $p \in P_2\}$; $W^+(p, t)$: = $\{W^+1(p, t)$ if $p \in P_1$ or $W^+_2(p, t)$ if $p \in P_2\}$; $M_0(p) = \{M_{10}(p)$ if $p \in P_1$ or $M_{20}(p)$ si $p \in P_2\}$; $\ell := T \rightarrow E$ is defined for any $t \in T$ by

$$\ell(t) = \begin{cases} \ell_1(t) & si\ t \in T_1, \ell_1(t) \notin E_1 \cap E_2 \\ \ell_2(t) & si\ t \in T_2, \ell_2(t) \notin E_1 \cap E_2 \\ \ell_1(t) = \ell_2(t) & si\ t \in T_{12}, \ell(t) \in E_1 \cap E_2 \end{cases}$$

The synchronous product of the example makes it possible to obtain the closed-loop control of the system, $R = R_P||R_S$..

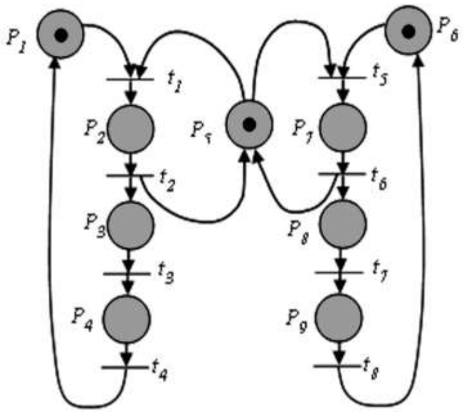


Figure 3: PN model of the closed-loop system

2.2 Reachable Marking graph of the Petri Net

The dynamics of the PNs is it that from the initial marking M_0, one can reach, via the occurrence of a sequence of events, a marking M_k, calculated by the fundamental equation

$$M_k = M_0 + W\sigma^T;\ W \text{ is the PN incidence matrix} \quad (1)$$

Simply noted $M_0\,[\sigma\rangle M_k$
The set of accessible markings M is the the state space $A(R, M_0)$ and can be represented as a reachable marking graph, if the PNs finite (Olive, 2011).

$$A(R, M_0) = \{M_k | \exists \sigma \in E^*, M_0\,[\sigma\rangle M_k\} \quad (2)$$

Definition 2. The reachable marking graph is a 4-tuple $G = \{M, E, \delta, M_0\}$ where :$M = A(R, M_0)$, is the (finite) set of states; E is the set of events associated with the transitions ;δ: M x $E \rightarrow M$ is the state transition function; $M_0 \in M$ is the initial state.

The reachable marking graph of the PN of Figure 2 is the follow

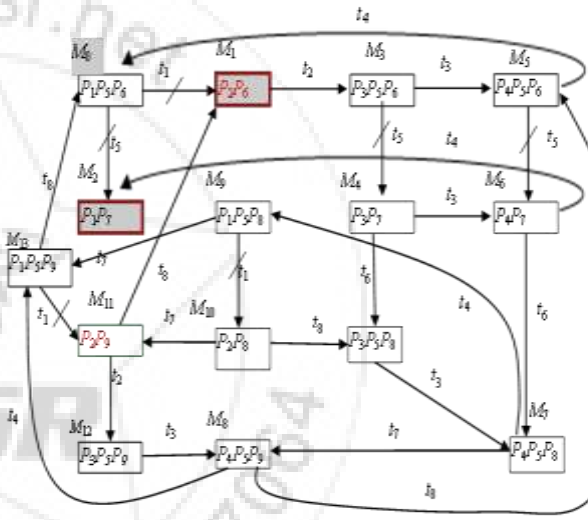


Figure 4: Marking graph of PN of Fig. 2

2.3 Petri Net Languages

A labeled PN (ℓ-*PN*) can define languages generated and accepted with a regular expression (Hopcroft et al., 2007)

Definition 3. Let $L(R)$ be the language generated by $R = (N, E)$ on the alphabet E, then there exists a marking graph G with a language $L(G)$, , such as:

$$L(R) = L(G) = \{\sigma \in E^* \mid M_0\,[\sigma\rangle M_k\}, \text{ with } \sigma \in T^* \rightarrow E^* \quad (3)$$

Definition 4. Let us consider the labeled PN (ℓ-*PN*) R which recognizes the language $L(R)$. Let F be the set of final states, which correspond to the achievement of a certain objective. The accepted language $L_m(R)$ is

$$L_m(R) = \{\sigma \in E^* | \exists (M_k \in F); M_0[\sigma\rangle M_k\} \subseteq L(R) \quad (4)$$

The language generated by R from any state M_k is

$$L(R, M_k) = \{\sigma \in E^* \mid \exists M_k \text{ et } \sigma \in L(R)\} \quad (5)$$

International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2016): 79.57 | Impact Factor (2015): 6.391

3. Supervisory Control Methods

3.1 Wonham and Ramadge Theory: Kumar's Algorithm

The theory initiated by Wonham and Ramadge (Ramadge & Wonham, 1983) enables to synthesize a maximal permissive controller (Cassandras & Lafortune, 2008) from the Kumar algorithm (Kumar, 1991).
Let P and S_{spec} be the automata models of the plant and specification of system, the Kumar algorithm permits to check the controllability of the specification language $L(S_{spec})$. The algorithm is based on the following 4 steps:

- Step 1. Build synchronous product D of P and S_{spec}, that is, $D = P \| S_{spec}$.
- Step 2. Determine the forbidden states: any state of D such that there exists an uncontrollable event defined in P but not defined in S_{spec}.
- Step 3. Determine weakly forbidden states: any state of D that is not a forbidden state and such that there is an uncontrollable sequence of events that leads to a forbidden state.
- Step 4. Remove from D the set of forbidden states also, the weakly forbidden states and states not accessible from the initial state.

3.2 Markings Invariant of PN Method

The marking invariant is a structural property of the PN that is determined by the X-vectors, from:

$$X^T W = 0, \text{ where } X \text{ is called P-semi-flot} \tag{9}$$

Such that $X^T = [l_1, \dots, l_i, \dots, l_n, 1]$ and the deduced invariant is given by the relation:

$$X^T M = X^T M_0 \tag{10}$$

The method makes it possible to calculate a controller based on the PNs (Iordache & Antsaklis, 2006) whose role is to force the SED to respect constraints of the type:

$$\sum_{i=1}^{n} l_i M(P_i) \leq \beta \tag{11}$$

Inequality of this type can be transformed into equality by adding a positive integer variable $M(P_C)$,

$$\sum_{i=1}^{n} l_i M(P_i) + M(P_C) = \beta \tag{12}$$

The incidence matrix of the SED under control, W, is composed of two matrices: W_R and W_C. The arcs connecting the places of controller with the transitions of PN of the closed loop system R are computed by the markings invariant equation (eq.10). Constraints are written in matrix form:

$$LM_R + M_C = b \tag{13}$$

And the marking invariant must satisfy:

$$X^T W = \begin{bmatrix} L & I \end{bmatrix} \begin{bmatrix} W_R \\ W_C \end{bmatrix} = 0 \Rightarrow W_C = -LW_R \tag{14}$$

The initial marking of the PN controller, M_{C0}, can be calculated from the equation:

$$LM_{R0} + M_{C0} = b \Rightarrow M_{C0} = b - LM_{R0} \tag{15}$$

4. Controller Synthesis from Adequate Admissible Constraints

4.1 Determination of Admissible Constraints by Kumar's Algorithm

Consider the specification of our example: the robot is transporting the part to the machine M1, and the machine M2 has already started and arrived at the end of the machining. In this state ($M_{11} = P_2P_9$) it is necessary for the robot to come to transport a new part to M2. Since this is not the case, production is slowed down. It is therefore a forbidden state that must never be reached.

Definition 5: Let $T_u \in T$ be the set of transitions associated with the uncontrollable events of $R = R_P \| R_S$, a state $M_k \in A(R, M_0)$ is forbidden if there is an uncontrollable transition that is allowed by M in R_P, , but not allowed by M in R_S .

On the other hand, if the structural synchronization (Figure 2) is performed via uncontrollable events it can generate additional forbidden states:

- The forbidden border States, M_B, corresponding to weakly forbidden states accessible by occurrence of controllable events.
- The set of critical admissible states, M_{AC}, corresponding to the states from which the occurrence of controllable events leads to a border state.

The application of Step 3 of Kumar's algorithm gives the forbidden border States:

$$M_{10} = P_2P_8 \text{ and } M_{13} = P_1P_5P_9.$$

These states give the following constraints:

$$M(P_2) + M(P_8) \leq 1 \text{ et } M(P_1) + M(P_5) + M(P_9) \leq 2$$

The associated constraint vector is:

$$L = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

The incidence matrix of PN, $R = R_P \| R_S$ is :

$$W_R = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & -1 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix}$$

The incidence matrix of controller is:

$$W_C = -LW_R = \begin{bmatrix} -1 & 1 & 0 & 0 & 0 & -1 & 1 & 0 \\ 2 & 0 & 0 & -1 & 1 & -1 & -1 & 1 \end{bmatrix}$$

The initial state of the PN, $R = R_P \| R_S$ is:

$$M_{R0} = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \end{bmatrix}^T \text{ and } b = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

The initial state of the controller is:

$$M_{C0} = b - LM_{R0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

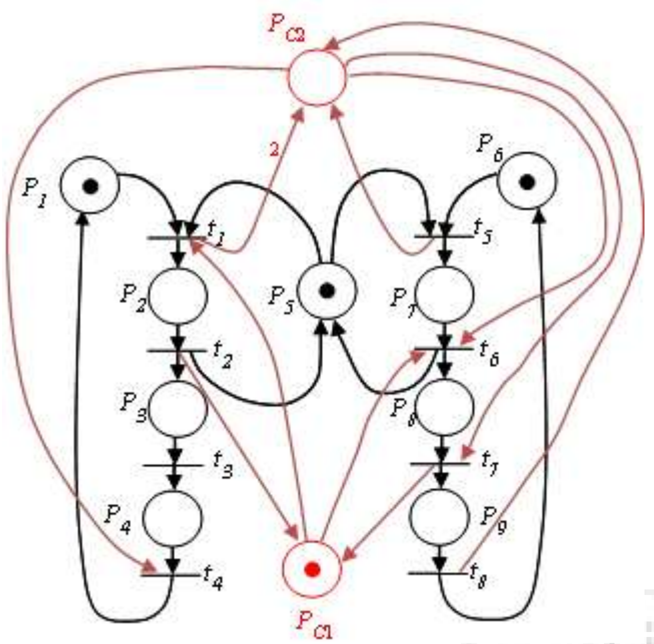


Figure 4: PN Model of the SED under supervisory

4.2 Determination of admissible constraint by separation of states of marking graph

The relevant state space for the supervisory control is compose by admissible states M_A and the forbidden states M_B . The transition function of the marking graph G is $\delta: M \times E \to M$, to which we associate a matrix $\Delta = [\delta_{ij}]$ called a transition matrix defined by:

$$\delta_{ij} = M_i[t_j\rangle \tag{16}$$

The marking graph of figure 3 can be represented by a table

Table 1: Transition Matrix of Marking Graph

δ	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8
M_0	$\{M_1\}$				$\{M_2\}$			
M_1		$\{M_3\}$						
M_2								
M_3			$\{M_5\}$		$\{M_4\}$			
M_4			$\{M_6\}$			$\{M_{14}\}$		
M_5				$\{M_0\}$	$\{M_6\}$			
M_6				$\{M_2\}$		$\{M_7\}$		
M_7				$\{M_9\}$			$\{M_8\}$	
M_8				$\{M_{13}\}$				$\{M_5\}$
M_9	$\{M_{10}\}$						$\{M_{13}\}$	
M_{10}							$\{M_{11}\}$	$\{M_{14}\}$
M_{11}		$\{M_{12}\}$						$\{M_1\}$
M_{12}			$\{M_{13}\}$					
M_{13}	$\{M_{11}\}$							$\{M_0\}$
M_{14}			$\{M_7\}$					

The elements $\delta_{ij} = M_i\,[t_j\rangle$ of the transitions matrix can be injectively coded [18] according to the state specification, which satisfy a predicate defined

$$g : \delta_{ij} \rightarrow \{0, -1\} \tag{17}$$

These states are then identified as the elements of the set

$$Mg = \{M_k | M_k \in M_I,\ g(M_k) = -1\} \tag{18}$$

The set of states allowed by the specification is the complement of Mg. Thus, we obtain a coded transition matrix $A = [a_{ij}] \in IR^{nxm}$ such that:

$$a_{ij} = \begin{cases} 1 & ;if\ \ M_k \in M_A \\ -1 & ;if\ \ M_k \in M_I \\ 0 & ;if\ \ M_k \in M_B \end{cases} \tag{19}$$

❖ *Separation of the states of the marking graph*

The separation consists of associating with each state M_k of the marking graph its forbidden set M_I or admissible M_A with respect to a set of states constituting the separation hyperplane, denoted H. We associate with H the decision function $g(M_k)$ defined by:

$$g(M_k) = \beta + \langle \alpha_i, M_k \rangle\ ;\ \alpha_i \in IR \text{ et } \beta = 0 \text{ for } \ell(t) = \varepsilon \tag{20}$$

Such that H strictly separate M_A and M_I if and only if:

$$\begin{cases} g(M_k) > 0 \ \ for\ any\ M_k \in M_A \\ g(M_k) < 0 \ \ for\ any\ M_k \in M_I \\ \qquad k = 0, \cdots, n \end{cases} \tag{21}$$

If H don't strictly separate M, then there existe a set M_H such that:

$$M_H = \{M_k : g(M_k) = 0\} \tag{22}$$

The set M_H can belong either to the set of forbidden border states M_B or to the set of critical admissible states M_{AC}.

Let the transition set $T \in \{0,1\}^n$, the firing transitions vector relative to the state M_k . T is a canonical basis of $g(M_k)$. The transition matrix codified in this base is the matrix $\mathbf{A}(t_1 \ldots t_m)$, denoted by $\mathbf{A} = [\alpha_i]$:

$$g(M_k) = [a_{ij}]\,[t_1 \ldots t_m]^T M_k$$
$$= \Sigma a_{ij}.\ M_k = \alpha_i\,.\ M_i \tag{23}$$

As the states M_k are positif ($M_i \geq 0$), the separation will depend on $\alpha_i = \sum_{j=1}^{m} a_{ij}$

- $\alpha_i < 0$: M_k is a state forbidden by the synchronization via uncontrollable transitions
- $\alpha_i > 0$: M_k is a state admissible by the specification
- $\alpha_i = 0$: M_k is a state belonging to the set M_H constituting the separation hyperplane.

We apply this approach to the marking graph (Figure 3) of our example 1. The specification requires the forbidden of markings of the places places P_2 and P_9, simultaneously. The state $M_{11} = P_2P_9$ of the marking graph is a state forbidden by the specification. With the rule of coding of the transition matrix, we obtain the separation of states (Table 2).

Table 2: Coding and Separation of states

δ	t_1	t_2	t_3	t_4	t_5	t_6	t_7	t_8	$\alpha_i = \Sigma a_{ij}$
M_0	1				1				2
M_1		1							1
M_2									x
M_3			1		1				2
M_4			1			1			2
M_5				1	1				2
M_6				1		1			2
M_7				1			1		2
M_8				1				1	2

M_9	1						1		2
M_{10}							-1	1	0
M_{11}		-1						-1	-2
M_{12}			1						1
M_{13}	-1							1	0
M_{14}			1						1

We have as a result:

- Admissible states: {M_0, M_1, M_2, M_4, M_5, M , M_8, M_9, M_{12}, M_{14}}
- Forbidden borders states: {M_{10}, M_{13}}$\in M_H$
- Forbidden states: M_{11}, M_2 (blocking state)

The forbidden border states are identical to those determined by the Kumar algorithm. So the controller obtained by the method of invariants is idem in fig. 4.

5. Conclusion

Supervisory control of the SED is a real need in the industrial sector that requires a formal and adapted response to the implementation problem of the initial theory based on the finite state automata. The first approach merges the invariant method with the Kumar algorithm. But the ideal was to absolve Kumar's algorithm as much as possible in order to stay in the full context of the PN. It is for this reason that we have proposed a simple method of separating the states of the marking graph, the PN analysis tool. It is based on the construction of the marking graph as a transition matrix. Our perspective is to establish an approach that defines the separation hyperplane as a directly exploitable constraint by the invariant method.

References

[1] P.J. Ramadge, and W.M. Wonham, "Supervisory control of a class of discrete event processes", Lecture Notes in Computer Science (LNCIS), Springer-Verlag, Germany, Vol. 63, pp. 477-498, 1983

[2] R. Kumar, "Supervisory Synthesis Techniques for Discrete Event Dynamical Systems". Thesis for the degree of Doctor of Philosophy, University of Texas, 1991

[3] M. Uzam, and W. Wonham, "A hybrid approach to supervisory control of discrete event systems coupling RW supervisors to Petri nets". Int. J. of Adv. Manuf. Technol. Vol 28, no 7 - 8, pp. 747 – 760, 2006

[4] R. David and H. Alla, "Discrete, Continuous, and Hybrid Petri Nets", Springer – Verlag , 2010

[5] W. M, Wonham (2011),"Supervisory control of discrete event systems", Technical report, University of Toronto, Dept. of Electrical and Computer Engineering, [Online]. Available: http://www.control.utoronto.ca/~wonham/.

[6] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions", IEEE Trans. Robot. Autom, Vol 19, no 1, pp. 137–141, 2003

[7] L. Holloway, and H. Krogh, "Synthesis of feedback control logic for a class of controlled Petri nets" IEEE Trans. Automatic control, Vol. 35, no 5, pp. 514-523, 1990

[8] A. Godon, "Contribution à la commande des systèmes à événements discrets par réseaux de Petri" Thèse de doctorat en Automatique Industrielle. Université d'Angers, 1996

[9] K. Yamalidou, J. O. Moody, M. Lemmon, and P. Antsaklis, "Feedback Control of Petri Nets Based on Place Invariants". Automatica,,Vol. 32, no. 1, pp. 15-28, 1996

[10] A.Vasiliu, A. Dideban, and H. Alla, "Control synthesis for manufacturing systems using non-safe Petri nets", J. of Control Eng. and Applied Informatics, Vol 11, no 2, pp. 43 –50, 2009

[11] A. Giua, and F. DiCesare, "Petri net structural analysis for supervisory control", IEEE Trans. Robot. Autom. Vol 10 n°2, pp. 185–195, 1994

[12] C. Cassandras, and S. Lafortune, "Introduction to Discrete Event Systems", 2nd Edition, Springer – Verlag, 2008

[13] H-E Gougam, A. Subias, et Y. Pencolé, "Diagnosticabilité de motifs de supervision par dépliage de réseaux de Petri", Manifestation avec acte : 5e Journées Doctorales /Journées Nationales MACS, Strasbourg, pp.139-144, 2013

[14] R.K Boel, L. Ben-Naoum and V. Van Breusegem, "On forbidden state problems for a class of controlled Petri nets", IEEE Transactions on Automatic Control, Vol. 40, no 10, pp. 1717-1731, 1995

[15] J. E. Falk, Y. Bandurova, and L. Yeganova, "Sets separation problems and global optimization", Third world congress of nonlinear analysts, Elsevier science Ltd, Vol 47, pp.1857-1867, 2011

[16] M. Pocci, I. Demongodin, N. Giambiasi, and A. Giua "Synchronizing sequences on not strongly connected Petri nets", In Symposium On Theory of Modelling and Simulation (DEVS/TMS'11), Boston, USA, 2011

[17] E-B. Dorsaf., S. Haddad, and R. Hennicker, "Refinement and asynchronous composition of modal Petri Nets", Lecture Notes in Computer Science LNCS 6900, K. Jensen, S. Donatelli, and J. Kleijn (Eds.), Springer, pp. 96–120. (2012

[18] E. Badouel, Darondeau, and L. Bernardinello, "Polynomial algorithms for the synthesis of bounded nets", Proccedings Caap 95, Lecture notes in computer science, Springer. Vol 915, pp. 364–378, 1995.

Author Profile

Mohaman Gonza received the Industrial Maintenance Engineer diploma and Master degree in computer sciences, Electronic and Automatic from National School Of Agro-Industrial Sciences (ENSAI) in 2008 and 2011, respectively. Since 2013, he stayed in Energy, Signal, Imagery and Automatic Laboratory of (LESIA), to study in PhD in area of Applied Physics and Engineering. His research interests include the control of discrete-event systems. The main used tools are Petri nets and their application to the manufacturing systems.