ISSN: 2639-0108

Research Article

Advances in Theoretical & Computational Physics

Structural Design of Supreme Controller with Uncontrollable Transitions

Mohaman Gonza[1*], Hassane Alla[2] and Laurent Bitjoka[3]

[1,3]Laboratoire LESIA, ENSAI, Ngaoundere University, Cameroon

[2]Grenoble Alpes University, France

[*]Corresponding author
Mohaman Gonza, Laboratoire LESIA, ENSAI, Ngaoundere University, Cameroon.

Submitted: 22 Feb 2022; **Accepted:** 26 Feb 2022; **Published:** 21 Mar 2022

Citation: *Mohaman Gonza, Hassane Alla and Laurent Bitjoka. (2022). Structural Design of Supreme Controller with Uncontrollable Transitions. Adv Theo Comp Phy, 5(1), 391-406.*

Abstract

In the framework of Petri net (PN), the structural supervisory control of discrete event system (DES) is an exciting method to design controller in the presence of uncontrollable transitions. Especially, this method addresses the controllability problem existing in the desired functioning PN. The controllability condition defined by uncontrollable synchronization can be expressed as constraints (GMEC). This leads to design place invariant-based controller, implemented through control places connected to plant transitions. The controller is maximally permissive if the transitions are controllable. Implying that the controller insures that the constraints are never violated directly or may be violated through the firing of uncontrollable transitions. But, if one control place is connected to uncontrollable transition, then the controller is non-admissible, since it cannot prevent such transition. The common idea is to transform the constraints or to displace the controller arcs. Unfortunately, the constraints transformation is computationally complex and not structural, while the arcs displacement approach is unsystematic. Our idea consists to iterate the structural supervisory control method to ensure a systematic displacement of controller arcs, so that no control place is input place of uncontrollable transition. This approach focuses on structural design of a less restrictive and admissible controller, namely the supreme controller.

Keywords: Discrete Event System, Controllability, Petri Net, Supervisory Control, Supreme Controller

Introduction

The paradigm of supervisory control theory (SCT) for discrete event systems (DES) is based on languages, but implemented on automata [1]. Since, a DES is a dynamic, event-driven system with discrete state, Petri nets (PNs) have become a successful tool for analysis and control [2]. PNs provide features — such as the synchronous product (Section 3.2) — to alleviate the state explosion. For supervisory control the synchronous product between plant and specification PNs, gives the desired functioning PN, typically grows linearly with the number of models [3]. Unfortunately, almost approaches of PN supervisory control are based on the reachability graph, which require the enumeration of the PN states [4]. Consequently, these approaches are partially structural and subjected again to the state explosion phenomenon. Moreover, very few works have addressed the controllability notion [5], even those focused to particular PNs [6, 7].

To address this issue, a structural method avoiding the construction of reachability graph and that take into account controllability condition, is proposed to design controller via labelled Petri nets [8]. This method is based on a one-to-one link between the supervisory control theory and the place invariant method [9]. Indeed, from the controllability condition, defined to deal with uncontrollable transitions in the desired functionning PN, the generalized mutual exclusion constraints (GMEC) are expressed [10]. To enforce these linear constraints, the place invariant-based controller is designed. As a result, controller is still be a relatively simple structure and implemented through control places and arcs connected to plant PN transitions. If the transitions are controllable, then the controller is maximally permissive (optimal). When the control place is connected to uncontrollable transition, the controller is not admissible since it cannot prevent the firing of such transition. The issue is that; we have no guarantee that at least one control place will not be connected to an uncontrollable transition.

To address that issue, an intuitive method was to ride up the branches of PN plant until finding a controllable transition that is upstream of the control place [9]. Unhappily, this method is not systematic and effectively applicable. Beside, some solutions proposed are based on constraint transformation to ensure that none control place is connected to uncontrollable transition [10-13]. But, the modified constraint itself may not represent the admissible controller corresponding to the original constraint [14, 13].

Although someone can use the approach based on constraint transformation proposed by Luo and Zhou, the high computational complexity of the control policy and the algorithm to transform a constraint into a disjunction of admissible ones, may not be efficient and optimal [15]. In order to guarantee the optimal control solution, a dynamic linear constraint must be introduced. Recently a method presented by Luo, et al. to design a maximally permissive controller require to partione the desired functioning PN into a set of dangerous regions to deal with uncontrollable events [16]. However, the control action is to maintain the most number of tokens not more than 1 for any sequence, contratry to the structural supervisory control where the control action is to maintain the number of tokens of specification PN more or egual to the number of tokens of the plant PN, with respect of arcs weight of transitions (see definition 8, section 3.2). In addition, there is a need for a algorithm to compute the set of controllable transitions that should be disabled by the controller.

Consider the complexity these approaches and the disadvantage of not offering structural control solution, it seems necessary for us to propose a new idea to obtain structurally the less restrictive and admissible controller, namely, the supreme controller [17]. Practically, the controller formed of the transitions of the plant PN and a separate set of places, is not admissible if there is an uncontrollable synchronization between the controller and plant PN. Intuitionally, the idea is to iterate the structural supervisory control method that was presented by Gonza et al. (2020), adapted to ride up the branches of PN plant until finding a controllable transition [8]. Naturally, this iteration deal with the uncontrollable synchronizations and the corresponding constraint in the controlled PN. We assume the supreme controller will be obtain systematically (section 4). This idea will be presented through the modified classic manufacturing system (section 3.1).

Following the recall on supervisory control related to controllability notion in section 2, a brief summary of PN tool and the structural supervisory control appears in section 3.

In section 4, the contribution of this paper is lights up to structurally design supreme controller, for plants with uncontrollable transition. To provide possibility to appreciate the computational simplicity or complexity, the proposed idea is applying to a case study (section 5).

Recall on supervisory control

Supervisory control theory (SCT) of DES, based on automata, is considered as one of the most successful approaches [17]. DES are systems that evolve in accordance with the occurrence of events e and their behavior may be described as a set of sequences σ over the alphabet Σ (the event set). Consider the unary operator Kleene star, the notation Σ^* gives infinite set of all possible sequences of events over Σ, including empty string ε.

Definition 1 (language). For a given alphabet Σ, the formal language L is a subset of Σ^*; it can be finite or infinite [18].

A supervisory control is a feedback control (figure 1) where the a controller C runs parallel with the plant G in order to enable/disable event occurrence based on the sequences generated by plant, so as to make the closed-loop behavior correspond to desired or legal language K. The legal behavior is defined by a given specification.

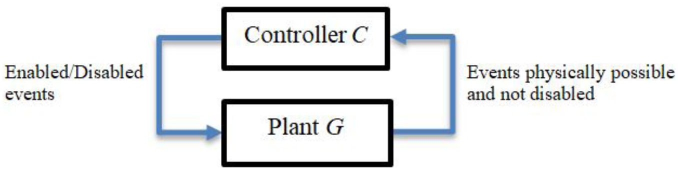

Figure 1: Basic principle of supervisory control

In that principle of supervisory control, the plant coupled with its controller C/G (read C controlling G) constitutes the closed-loop DES. Each time, the controller C provides a list of enabled/disabled events to occur in plant G.

The set of events Σ may be partitioned as $\Sigma=\Sigma_c \cup \Sigma_u$, where Σ_c and Σ_u are, respectively, the sets of controllable events, and uncontrollable events, whose occurrence cannot be prevented by the controller. Generally, the behavior of the plant G is unsatisfactory for a given specification S and needs to be "restrict". Since, the desired functioning (or legal behavior) is specify by the language K, the basic control problem is to design a controller that restricts the closed loop behavior DES to K∩L(G). But, the presence of uncontrollable events Σ_u, whose occurrence cannot be prevent by controller, leads to define the controllability condition.

Definition 2 (Controllability). Consider the event set Σ of the plant G, partioned into the sets of uncontrollable events Σ_u and controllable events Σ_c. A language K⊆L(G) is said to be controllable with respect to the plant language L(G) and Σ_u , if

$$K \Sigma u \cap L(G) \subseteq K \qquad (1)$$

K⊆L(G) is prefix closed by construction; any sequence σ ∈ K, implies that every prefix of σ is in K (Hopcroft et al., 2007), i.e, $\overline{K}$:={σ' ∈ Σ* | ∃ (σ ∈ K) such σ' is a prefix of σ}

The existence of controller C such that the language achieved by the closed-loop DES be L(C/G) = K is linked to the controllability condition (eq. 1). In the case where K ⊆ L(G) is not controllable with respect to the plant language L(G), it is necessary to get supreme controllable language, SupC(K), less permissive than K (Wonham and Kai, 2017) [18]. Thereby, the behavior of closed-loop DES is said to be maximally permissive.

Regarding the automata based supervisory control, from models of given a plant G and specification S, the Kumar's algorithm [19] allows to compute a maximally permissive controller corresponding to the supreme controllable language, such as SupC(K) ⊆ L(S)∩L(G).

In this paper, we focus on structural design of such controller, avoiding the complexity linked to languages or reachability graph, [6, 20]. Thus, we will use the structural supervisory method which address PN controllability condition to design the controller in the presence of uncontrollable transitions [8]. In fact, from the structure of desired functioning PN, obtained by the synchronous product between plant PN (N_G) and specification PN (N_S), namely $N_G \parallel N_S$. The language that characterizes the trajectory of the controller satisfies [18].

$$L(N_G \parallel N_S) = L(N_S) \cap L(N_G) \quad (2)$$

We can get via the PN controllability condition, the linear constraints (GMEC-type) to compute the place invariant-based controller (Section 3.2), without constructing the reachability graph. The controller is admissible when control places are connected to controllable transitions of the plant (N_G). It guarantees for any PN state that: if transition is enabled in the plant (N_G), it must also be enabled by the specification (N_S).

Nevertheless, there may exist situations where the control place is connected to uncontrollable transition, i.e, there exist uncontrollable synchronization between the controller and plant PN. Consequently, the designed controller is non-admissible, since it can never prevent plant-enabled uncontrollable transitions from firing.

In such a situation we need to obtain a less restrictive and admissible controller, such that the behavior of controlled PN (formed of the plant PN and that of the controller) being supreme controllable. For this, we propose a new idea based on the iteration of the structural supervisory control method adapted to ensure there exist no arc from a control place to an uncontrollable transition by using labelled PN [8]. This idea is explored through the classical manufacturing system where we have brought some modifications (example 1)

Petri net tools and structural supervisory control

Petri nets tools

The power of modeling DES is strictly related to the sequences of events that it can generate. For this reason, it is suitable to use Labelled PN, which permits to specify event corresponding to transition [21]. The graphical representation of PN is given in figure 3.

Definition 3 (Labelled PN). Let N denote a Labelled PN, it is defined to be the 7-tuplet, $N \coloneqq (P,T,\Sigma,D^-,D^+,M_0,L)$, where

- $P=\{p_1,\cdots p_i,\cdots p_n\}$ is the finite set of n places;
- $T=\{t_1,\cdots t_j, \cdots t_m\}$ is the finite set of m transitions;
- Σ is a finite set of events (labels) including the event always occurring ε;
- $D^-(\bullet,t_j) \coloneqq P \times T \rightarrow Z$ is the backwards incidence matrix that define the weights of the directed arcs $(\bullet,t_j)$ from places p_i to transitions t_j;
- $D^+(\bullet,t_j) \coloneqq P \times T \rightarrow Z$ is the forwards incidence matrix. that define the weights of the directed arcs $(\bullet,t_j)$ from transitions t_j, to places p_i;
- $M_0 \in N^n$ is the initial marking or state. It is given by the number of tokens (black dot) in each place p_i, denoted as $M(p_i)$;
- L: $T \rightarrow \Sigma \cup \{\varepsilon\}$ is a label function, which labels an event $e_j \in \Sigma$ for each transition $t_j \in T$, i.e, $e_j=L(t_j)$ and $L(\varepsilon) = \varepsilon$. If $L(\varepsilon) \neq \varepsilon$ for all $t_j \in T$ then L is ε – free. L is extended from transition sequence set T* into Σ^*, such that for $\tau=t_1 t_2 \cdots \in T^*$, $\sigma = L(t_1)L(t_2)\cdots \in \Sigma^*$

In a Labelled PN, firing a transition is linked to events occurrence, witch can be partioned into uncontrollable events set Σ_u and controllable events set Σ_c. By analogy, the set of uncontrollable transitions is denoted by $T_u \coloneqq \{t_j \in T \mid L(t_j) \in \Sigma_u\}$, and the controllable transitions set, $T_c \coloneqq \{t_j \in T \mid L(t_j) \in \Sigma_c\}$.

Example 1. Modified classic manufacturing system

The classic manufacturing system is composed of two machines (Mch_1 and Mch_2) working independently, draw raw parts upstream and reject processed parts downstream. The existing Buffer (Buf) between the machines receives the machined parts from the conveyor transfer station, after overturning. Machine (Mch_2) can only start working if it can take processed parts from the Buffer (Buf), assuming to be empty in its initial state. This modification supposes the existence of the turn over event r and transfer event v. To illustrate our contribution, we will consider that these events and the ending of the works as uncontrollable ($\Sigma_u=\{r,v,e_1,e_2\}$), while the starting of each machine is controllable event, ($\Sigma_c=\{s_1,s_2\}$)

We consider a given specification, which consists to ensure that a buffer (Buf) has a capacity limited to x parts, defined by the operator.

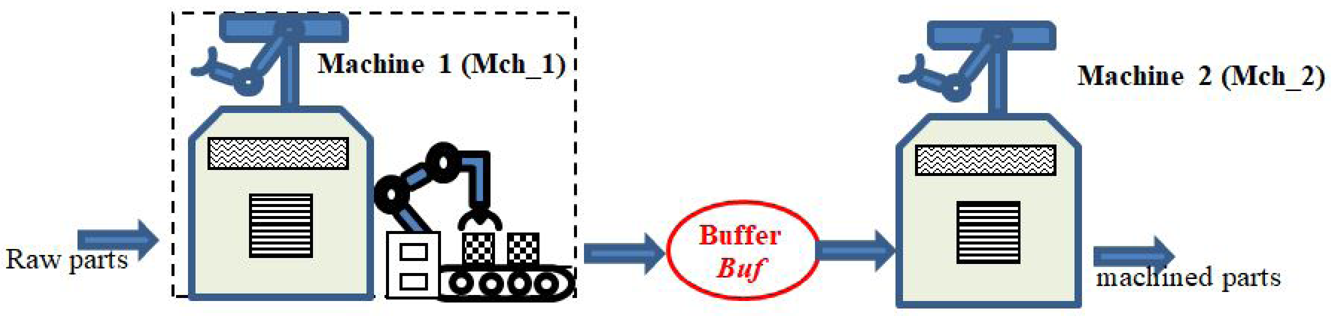

Figure 2: Topology of the modified classic manufacturing system

The graphical representation of the PN of this example 1 is shown in figure 3 bellow.

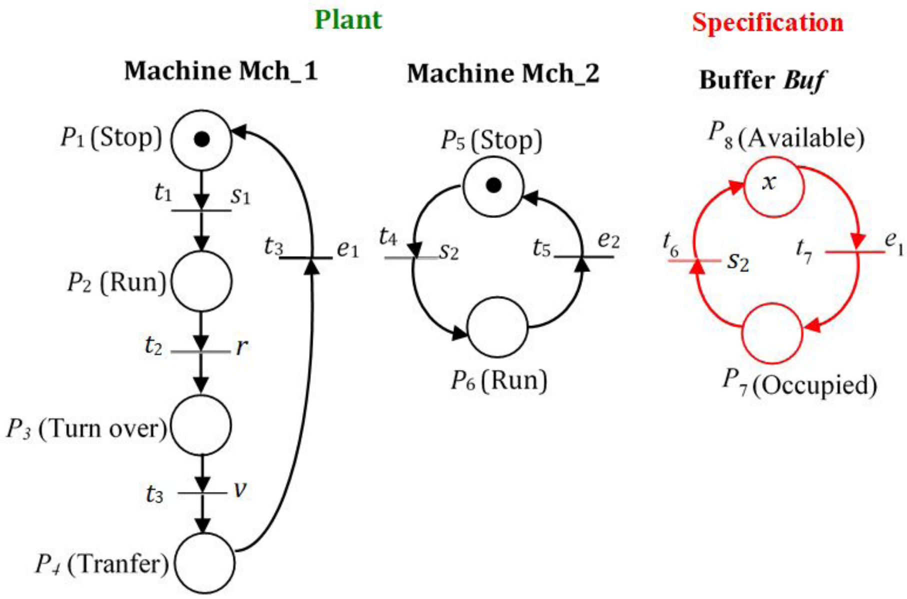

Figure 3: Labelled PN of this modified classic manufacturing system

For this given example, the controllable events set is T_c:={ t_1 ,t_4 ,t_6 } and the uncontrollable events set is T_u={t_2 , t_3 , t_5 , t_7 }

The PN dynamic can be represent by the presence / absence of tokens in the places. The marking or state M is a column vector, M:=P→Z is a mapping function that assigns a non-negative integer (tokencount) to each place. For a transition t_j∈T we define the set of input places as (•)t_j :={p_i∈P ┃ D^- (•,t_j)>0} . If and only if M(p_i) ≥ D^- (•,t_j), transition t_j is enabled under M.

From a state M_k, only enabled transitions can be fired, and the new state M_{k+1}'is resulted after t_j fires is denoted as M_k [t_j
M_{k+1} = M_k+ D(•,t_j), (3)
where D(•,t_j) = D^+ (•,t_j) -D^- (•,t_j) indicates for t_j the incidence matrix
If the transition sequence τ∈T* is enabled from initial state M_0, denoted as M_0 [τ┤⟩ the new state is reached, denoted as M_0 [τ┤⟩ M_(k+1)
We denote by R(N,M_0) the reachability graph, which is the set of reachable states from M_0, i.e,
R(N,M_0):={M_k ∈ N_n ┃∃τ∈T^(*);M_0 [τ⟩}

Given a Labelled PN N, if we consider instead the transitions sequence, the events sequence (finite set) generated, then we can define PN language (Hopcroft et al., 2007) as follows
L(N):={σ∈Σ^* ┃∃τ∈T*,L(τ)=σ "and" M_0 [τ⟩┤"is" "defined" }
Generally, PNs can represent more expressive and prefix closed languages in Σ^* than automata (Guia, 2013).

Structural supervisory control

The system in need of supervision, the plant and its specifications are modeled by PNs. From the desired functioning PN (figure 4), obtained by the synchronous product between plant PN (N_G) and specification PN (N_S), namely N_G ∥N_S ┤, the controllability condition is established.

Definition 4 (Synchronous product) Let
N_G:=(P_G,T_G,$Σ_G$,D_G^- ,D_G^+,M_{0G},L_G)
be the plant PN and N_S:=(P_S,T_S, $Σ_S$, D_S^- ,D_S^+, $M_0 S$, L_S) be the specification PN, both build on the same events set ($Σ_S$=$Σ_G$). Their synchronous product N_G ∥ N_S is another synchronized Petri net, N:=(P, T, Σ, D^-,D^+,M_0,L) , such that

- $P = P_G \cup P_S$
- $\Sigma = \Sigma_S \cup \Sigma_G$
- $T \coloneqq T_G \cup T_S - T_{GS}$,$T_{GS} \coloneqq \{(t_G, t_S) \in T_G \times T_S | \mathcal{L}_G(t_G) = \mathcal{L}_S(t_S)\}$;
- $\mathcal{L}(t_j) \coloneqq \mathcal{L}_G(t_j)$ si $t_j \in T_P$ or $\mathcal{L}_S(t_j)$ si $t_j \in T_S$
- $D^- \coloneqq: \{(\bullet, (t_G, t_S)) \in P \times T | (\bullet, t_G) \in D_G^-$ or $(\bullet, t_S) \in D_S^-\}$
- $D^+ \coloneqq \{((t_G, t_S), \bullet) \in T \times P | (t_P, \bullet) \in, D_G^+$ or $(t_S, \bullet) \in, D_S^+\}$
- $M_0(p_i) \coloneqq M_{0G}(p_i)$, if $p_i \in P_G$ *or* $M_{0S}(p_i)$, if $p_i \in P_S$

Intuitively, the synchronuous product is a matter of structural synchronization, where a pair of transitions (t_G,t_S) with the same event is replace with a single transition t_j = (t_G,t_S), Particularly, called synchronous transition. If there exist several transitions in each PN with the same event, then there exists one transition in the desired functioning PN for each transition pair combination (Kumar and Holloway, 1996) [5]. Without loss this generality, we applying this suitable operation to the PN of figure 3, where each event is associated with at most one transition in each PN.

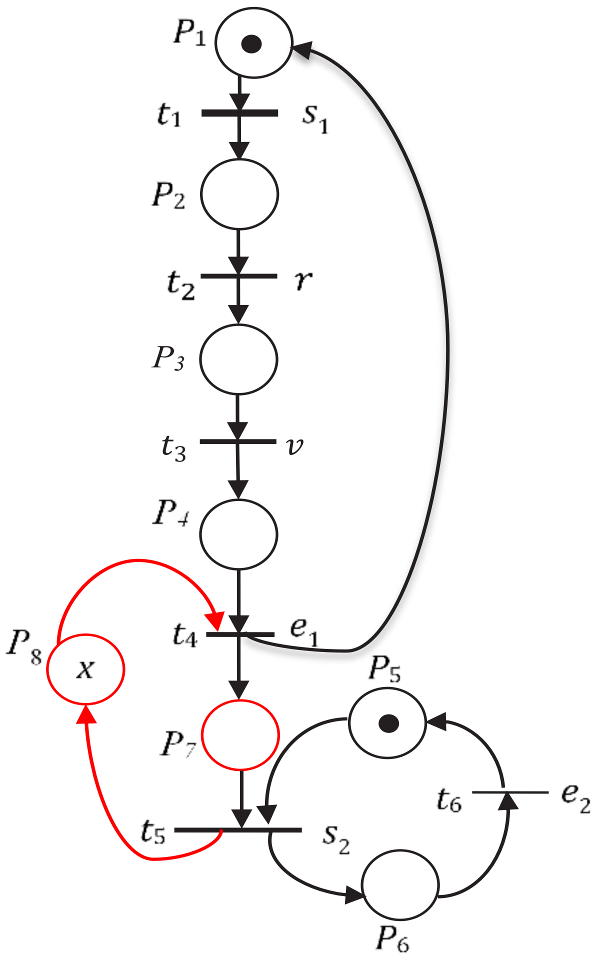

Figure 4: Desired functioning PN of the modified classic manufacturing system

From a desired functioning PN we can check the controllability condition, because the structural synchronization via uncontrollable transitions can be a potential source of uncontrollability and forbidden states (Dideban and Alla, 2008) [22].

Definition 5 (Forbidden states)

Let $t_j = (t_G, t_S) \in T_u$ be a synchronous transition in desired functioning PN, $M_G((\bullet)t_j)$ and $M_S(\bullet)t_j)$ the marking of input places belong to plant (N_P) and specification N_S respectively; we define the set of forbidden states

$$\mathcal{M}_b := \left\{ M_k \in \Re(N, M_0) \middle| \exists t_j = (t_G, t_S) \in T_u \text{, such that} \begin{Bmatrix} M_G({}^\bullet t_j) \geq D_G^-(\bullet\ t_j) \\ M_S({}^\bullet t_j) < D_S^-(\bullet\ t_j) \end{Bmatrix} \right.$$

The desired functioning PN, $N := N_G \parallel N_S$, is uncontrollable when a reachable state $M_k \in M_b$. In the figure 4 we face such situation when we consider the uncontrollable synchronous transition t_4, namely,

$$t_4 \in T_u ; \begin{cases} M_G(P_4) \geq 1 \\ M_S(P_8) < M_G(P_4) \end{cases}$$

This allows defining structural controllability condition of the desired functioning PN

Definition 6 (structural controllability)

For any uncontrollable synchronous transition $t_j = (t_G, t_S) \in T\_(u)$, the structural controllability condition for any reachable state M_k, when $M_P(\bullet t_j) \geq D_G^-(\bullet t_j)$ is (Gonza et al., 2020) [8].

$$M_S({}^\bullet t_j) \geq M_G({}^\bullet t_j) \text{, with } M_S({}^\bullet t_j) \geq D_S^-(\bullet\ t_j)$$

It has been proven that the structural controlability condition is equivalent to that defined on the PN languages (Gonza, 2019) [23].

$$M_S({}^\bullet t_j) \geq M_G({}^\bullet t_j) \Leftrightarrow K\Sigma_u \cap L(N_G) \subseteq K$$

From this it is seem not necessary scanning the reachability graph or PN languages to check the controllability and to define a set of admissible states.

Definition 7 (Admissible states)

Given a desired functioning PN, the set of admissible states, M_a, is the one in which the structural controllability condition is verified.

$$\mathcal{M}_a := \{ M_k \in \Re(N, M_0) | \exists t_j = (t_G, t_S) \in T_u ; M_S({}^\bullet t_j) \geq M_G({}^\bullet t_j) \}$$

The controllability condition can be is expressed into linear contraints of GMEC type, denoted as $L^T M_k \leq b$, $\forall M_k \in R(N,M_0)$, where

$L=[l_1 \cdots l_n] \in Z^{n_c \times m}$ nd $b \in Z^{n_c}$

For any $t_j=(t_G, t_S) \in T_u$ the controllability condition is expresses into inequality $M_G(\bullet t_j) - M_S(\bullet t_j) \leq 0$, where
$L=[0\cdots 0\ l_G\ 0 \cdots 0\ l_S\ 0 \cdots 0]$, with l_G)=1, l_S = -1 and b = 0
Hence, in the figure 4, the controllability condition is
$M_S(P_8) \geq M_G(P_4)$ and the corresponding constraint is
L=[0 0 0 1 0 0 0 - 1]

At this point, the control goal is to insure that the constraints are met during the plant's operation. At this point, the place invariant method provides the controller incidence matrix D_C and initial state M_{0C} of the PN that implements a controller C (Guia, 2013) [24]. The place invariant based controller identical to the monitors C (Uzam, 2010; Wang et al., 2013) [25, 26]. The controller is a PN with incidence matrix $D_C \in Z^{n_c \times m}$) with initial state $M_{0C} \in Z^{n_c}$, made up of the plant's transitions and a separate set of places.

$$D = -LD$$
$$M_{0C} = -LM_0 \quad (5)$$

The controller is maximmally permissive assuming that the plant's transitions are controllable

Definition 8 (maximally permissive controller)
A controller is maximally permissive if all the admissible state, M_a of the desired functioning *PN*, $N := N_G \| N_S$, are reachable under control, and the firing of transitions that leads the plant *PN* evolution to a forbidden state is prevented.

In incidence matrix D_C positive elements in refer to arcs connecting transitions to control places and negative elements refer to arcs connecting control places to transitions. From this, the controller C is coupled by synchronization to desired functioning PN, to give the controlled PN (figure 5).

Definition 9 (Controlled PN)
A controlled PN is a triple N = (N,C,B); where $N := N_G \| N_S$ is the desired functioning PN, C a PN model of the controller is a finite set of control places, C∩P=∅, and $B \subseteq C \times T$ is a set of arcs (with weight) connecting control places (p_c) to transitions set T.

Applying this to our current example 1 (figure 4), we have :

$$D_C = -[0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad -1]\begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \end{bmatrix} = [0 \quad 0 \quad -1 \quad 0 \quad 1 \quad 0]$$

$$M_{0C} = -[0 \quad 0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad -1]\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ x \end{bmatrix} = x$$

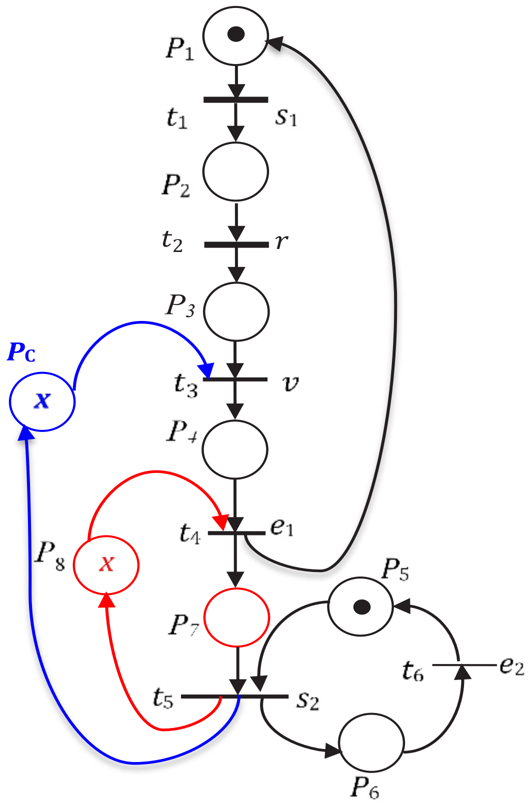

Figure 5: The controlled PN of the modified classic manufacturing system

In controlled PN, the controller must allow all control (connected) transitions to be fired only when it is both control place and plant enabled, otherwise it is prevented.

Consider the control transition $t_3 \mid L(t_3) = v$ as controllable, then the controller designed is maximally permissive. Unfortunately, it was specified (section 2) that the event v is uncontrollable, i.e, the transition $t_3 \mid L(t_3) = v$ is uncontrollable. Hence, the controller designed is non-admissible, since it cannot prevent such transition when it is enabled in plant PN.

The controllability of the controlled PN must be checked, in order to obtain the less restrictive and admissible controller (supreme controller). In fact, the controller designed may prevent plant-enabled uncontrollable transitions from firing.

Structural design of supreme controller (less restrictive and admissible)

Uncontrollable transitions can cause problem for controlled PN, due to arcs from the control places used to change the controller state based on the frings of plant transitions. For this reason, we propose an idea for structurally design the supreme controller, which is less restrictive and admissible.

For each control (connected) transition $t_j \in T$ of the controlled PN, $N = (N, C, B)$, where $M_N(\bullet t_j)$ is the marking of input places belonging to plant N_G, and $M_C(\bullet t_j)$ is the marking of input control places belonging to controller C. When the controller behaves correctly the connected transition t_j must be disabled if the marking of input control places $M_C(\bullet t_j)$ is less than weigh of their arcs $B(\bullet t_j)$, i.e, $M_C(\bullet t_j) \geq B(\bullet t_j)$. When the connected transition is uncontrollable ($t_j \in T_u$), there is no guarantee that will happen, since the firing of such transition is limited solely by the structure and state of the plant N_G. Consequently, the controller designed is non-admissible (Moody and Antsaklis, 2000) [10]. Given D the incidence matrix of $N := N_G \parallel N_S$ and $L = [0\cdots 0\ l_G\ \ 0\cdots 0\ l_S\ \ 0\cdots 0]$ the constraint from controllability condition $M_s(\bullet t_j) \geq M_G(\bullet t_j)$. Let D_u be sub-matrix representing the uncontrollable part of D, such that LD_u is the portion of controller corresponds to uncontrollable transitions. Let's see LD_u like the admissibility condition of designed controller. If LD_u contains at least one strictly positive element, i.e $LD_u \geq 0$, then there are control place connected to uncontrollable transition ($t_j \in T_u$).

Consider in our example (figure 4) the uncontrollable transitions set $T_u = \{t_2, t_3, t_4, t_6\}$ associated by label function to the uncontrollable events set transitions $\Sigma_u = \{r, v, e_1, e_2\}$ we will have,

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

For the controller to be less restrictive and admissible, the sufficient condition should be $LD_u \leq 0$, where the constraint $L = [0\cdots 0\ l_G\ 0\cdots 0\ l_S\ \ 0\cdots 0]$ is given by controllability condition $M_s(\bullet t_j) \geq M_G(\bullet t_j)$. When the condition $LD_u \leq 0$ is unsatisfied, the idea is to iterate the structural supervisory control method from the controlled PN until founding controllable transitions, which is upstream to the control places. Concretely, it is a question of extending the controlallitity condition to the controlled PN, N = (N,C,B). Hence, for any uncontrollable control transition $t_j \in T_u$), the structural controllability condition for any reachable state M_k, when $M_N(\bullet)\, t_j \geq D^-(\bullet t_j)$ is

$$M_c(\bullet t_j) \geq M_N(\bullet t_j)\text{, with } M_c(\bullet t_j) \geq B(\bullet t_j) \tag{6}$$

Corollary. Let $L_N = [0\cdots 0\ l_N\ \ 0\cdots 0\ l_C\ \ 0\cdots 0]$ be a constraint provided by an extending controllability condition $M_C(\bullet t_j) \geq M_N(\bullet)\, t_j$) to the controlled PN, N = (N,C,B), and DNu be the incidence sub-matrix representing the uncontrollable part of D_N The new controller C_1 is admissible, while the condition $L_N D_{Nu} \leq 0$

Proof. If the control place is connected to uncontrollable transition, the (extended) controllability condition $M_C(\bullet t_j) \geq M_N(\bullet t_j)$ is satisfied. The constraint L is systematically transformed to a new one L_N, in order to enforce the condition $L_N D_{Nu} \leq 0$. By iteration of the structural controller design, this will result to connecting control place to a controllable transition, since the number of plant transitions is finite.

Consider the desired functioning PN, $N := N_G \parallel N_S$ with controllability condition $M_s(\bullet t_j) \geq M_G(\bullet t_j)$ and the admissibility condition of controller $LD_u \geq 0$, one can then find a less restrictive and admissible controller using Algorithm 1.

Algorithm 1 Structural design of a supreme controller (less restrictive and admissible

Input: controlled PN, $\mathcal{N} = (N, C, B)$ and $D_{\mathcal{N}}$

Output: supreme controller C_i ,

Initialization step:

From controlled PN, $\mathcal{N} = (N, C, B)$ check **if** the controller C draws no arc to uncontrollable transit ($t_j \in T_u$), i.e, $LD_u \leq 0$

1: If not, set $L_{\mathcal{N}} = [0 \cdots 0 \; l_N \; 0 \; \cdots 0 \; l_C \; 0 \cdots 0]$ is constraint provided by $M_C(\,^{\bullet}t_j) \geq M_N(\,^{\bullet}t_j)$

Supreme control step:

2: Do *i*:=1, $D_{C_1} = -L_{\mathcal{N}} D_{\mathcal{N}}$ and check $L_{\mathcal{N}} D_{\mathcal{N}u}$, where $D_{\mathcal{N}u}$ is the sub-matrix representing uncontrollable part of $D_{\mathcal{N}}$

3: If $L_{\mathcal{N}} D_{\mathcal{N}u} \leq 0$, C_1 is less restrictive and admissible controller

4: If not i.e $L_{\mathcal{N}} D_{\mathcal{N}u} \nleq 0$,

5: **Repeat (1)** for the next controller PN $\mathcal{N}_1 = (N,\, C_1 ,\, B_1)$

6: **Do**

$D_{C_i} = -L_{\mathcal{N}_{i-1}} D_{\mathcal{N}_{i-1}}$ and check $L_{\mathcal{N}_{i-1}} D_{\mathcal{N}_{i-1}u}$

7: **Until** $L_{\mathcal{N}_{i-1}} D_{\mathcal{N}_{i-1}u} \leq 0$

Stop. C_i is less restrictive and admissible controller

8: **For all** $(t_j \in T)$ **if we have always** $L_{\mathcal{N}_{i-1}} D_{\mathcal{N}_{i-1}u} \nleq 0$, then the control solution is empty, ($C_i = \emptyset$). This is good solution, since the plant PN transitions is finite.

Let us apply the above to controlled PN (figure 5) of our current example 1 where the controller is non-admissible, because it is connected to the uncontrollable transition, $t_3|\mathcal{L}(t_3) = v$.

Iteration or step 1

- The characteristics of the controlled PN are

$$D_{\mathcal{N}} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 & 0 \end{bmatrix} \text{ and } M_{0_{\mathcal{N}}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ x \\ x \end{bmatrix}$$

We put in red the columns corresponds to $D_{\mathcal{N}u}$

- The controllability condition is $M_C(P_c) \geq M_P(P_3)$
- The constraint is $L_{\mathcal{N}} = [0 \quad 0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \; -1]$
- The new controller is $\begin{cases} D_{C_1} = [0 \quad -1 \quad 0 \quad 0 \quad 1 \quad 0] \\ M_{0C_1} = x \end{cases}$
- The controller portion corresponding to uncontrollable transitions $L_{\mathcal{N}} D_{\mathcal{N}u} = [1 \quad 0 \quad 0 \quad -1]$

The resulting controller is non-admissible, since $L_{\mathcal{N}} D_{\mathcal{N}u} \nleq 0$ one strictly positive element) and D_{C_1} draws an arc to the supposed uncontrollable transition $t_2|\mathcal{L}(t_2) = r$ (figure 6). Consequently, we must iterate the procedure again.

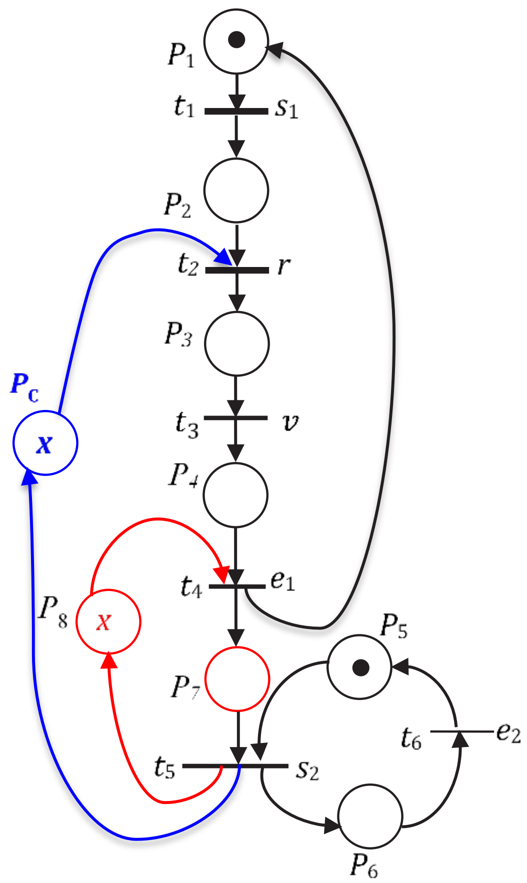

Figure 6: Controlled PN N_1 , after 1st step, with non-admissible controller C_1

Iteration or step 2

- Characteristic of controlled PN

$$D_{\mathcal{N}_1} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 0 & -1 & 0 & 0 & 1 & 0 \end{bmatrix} \text{and } M_{0\mathcal{N}_1} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ x \\ x \end{bmatrix}$$

- The controllability condition is $M_C(P_c) \geq M_P(P_2)$
- The constraint is $L_{\mathcal{N}_1} = [0 \quad 1 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 \quad 0 - 1]$
- The new controller is $\begin{cases} D_{C_2} = [-1 \quad 0 \quad 0 \quad 0 \quad 1 \quad 0] \\ \qquad M_{0C_2} = x \end{cases}$
- The controller portion corresponding to uncontrollable transitions $L_{\mathcal{N}_1} D_{\mathcal{N}_1 u} = [0 \quad 0 \quad 0 \quad -1]$

This solution is less restrictive and admissible because $L_{\mathcal{N}_1} D_{\mathcal{N}_1 u} \leq 0$ (none strictly positive element) and D_C will draw an arc to the controllable transition $t_1 | \mathcal{L}(t_1) = s_1$ (Figure 6). We must stop the iteration here, because we get the supreme controller.

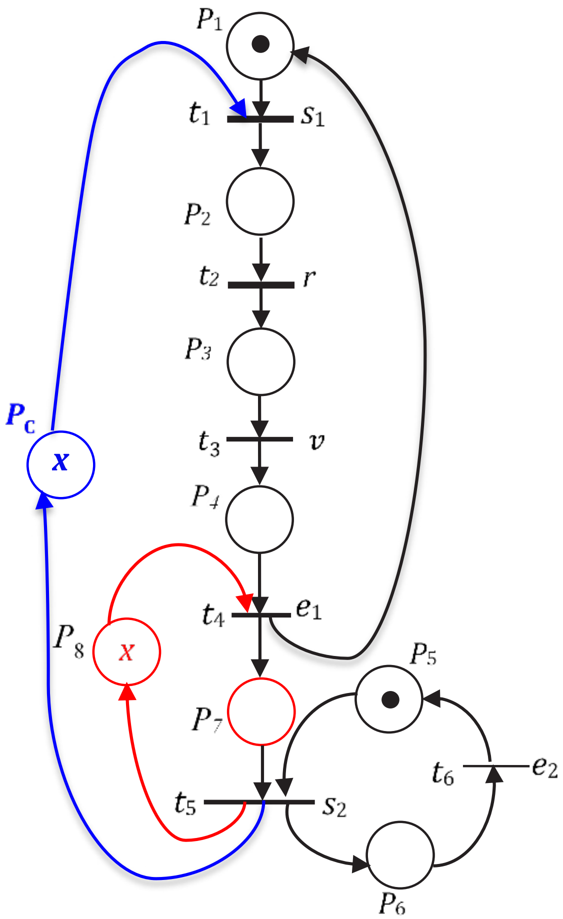

Figure 7: Controlled PN N_2 , after 2nd step, with non-admissible controller C_2

The approach is systematic and structural, since we find the solution similar to the intuitive approach of Yamalidou et al. (1996) [9].

Now, suppose that transition $t_l \mid L(t_l) = s_1$ is uncontrollable transition, then the control solution is empty. It can be noticed that, in Figure 6 place P_8 is implicit and can be suppressed.

Remark 1. Modeling considerations

Example 1 shows typical modeling plant PN's structures. It can be seen that the uncontrollable transition has only one input place. This is in fact a general modeling aspect, which leads us to precise the modeling characteristics of controllable and uncontrollable transitions.

Controllable transition: A controllable transition may have several input places. Indeed, the firing of this transition is conditioned by the synchronization of several tasks behaving concurrently. The controllable transition is fired when all the input places are marked and the controllable event occurs.

Uncontrollable transition: An uncontrollable transition has only on input place. The occurrence of an uncontrollable event, a breakdown or the end of a task for example cannot be blocked by several input places. It occurs when the plant is in a given state, represented in a global way by the input place.

Compare to existing methods (constraint transformation or algorithm to compute controllable transitions), we have present a very simple idea, systematic and easy to implement by using the iteration of structural supervisory control with respect of controllability condition. Also, the simplicity of linear constraints allows obtaining a controller structurally optimal (no addition of control places or arcs to the controlled PN). This solution problem has already been tackled in Yalamidou (1996) in an intuitive and unsystematic way (Dideban and Alla 2008) [9, 22]. We assume that, a good variety of DES control problems can be efficiently solved through advantages of this approach:

- The approach is elegant for implementation as it is based on constraints linking the supervisory control theory and the place invariant method.
- The synthesis technique makes use of an incidence matrix corresponding to the uncontrollable portion of the plant to controlled PN model
- The systematic handling of uncontrollable events is maintained with the controlled PN model.

Case study

As a case study, consider the real-life system taken from (Vasiliu, 2012) [27]. It is in an industrial assembly line, whose topology is illustrated in figure 7.

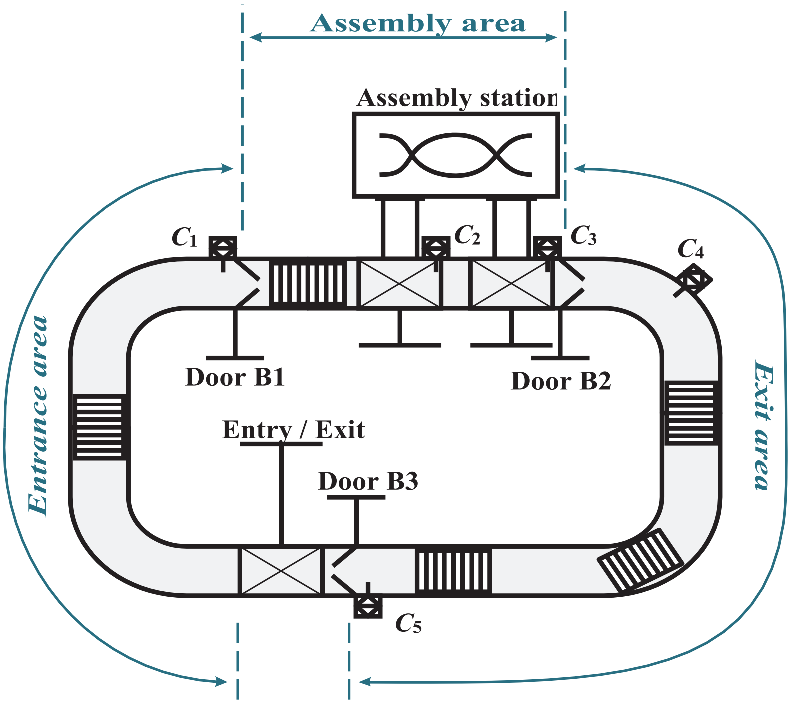

Figure 8: Topology of industrial assembly line (Vasiliu, 2012)

The assembly line consists of a conveyor, an assembly station, three barrier doors (B1−3) and five sensors (C1−5). An entry / exit station connects the assembly system with other systems in the line and provides entry / exit of parts into the assembly loop. The assembly loop is divided into three areas:

- the entrance area, between the entry / exit station and gate B1,
- the assembly area, between doors B1 and B2 and
- the exit area, between gates B2 and B3.

A part enters the system through the entry / exit station, travels the entry area, and then is admitted into the assembly area, where it is introduced inside the assembly station to be processed. Once it is complete, the assembled parts are returned to the conveyor, travel through the exit area and exit the assembly loop via the entry / exit station. The system (assembly line) must satisfy the following specifications:

- the maximum number of parts allowed at any time in the assembly area (the length of the assembly queue) is ten;
- the maximum number of parts allowed at any time in the exit area (the length of the exit queue) is twelve.

The PNs of figure 8, models the plant and the specification of the assembly line, while events associated with transitions and the place descriptions are shown in Table 1.

Table 1: Description of places and events associated with transitions

Places		Transitions	
P1	There is no room at the entrance to the assembly area	c1a	c1 active (part detected at the entrance to door B1)
P2	Part waiting to enter the assembly area	b1o	Door B1 opening
P3	Part entering the assembly area	c1i	c1 inactive (part has left door B1)
P4	Part entered in the assembly area	b1f	Door B1 closing
P5	No part is awaiting assembly	c2a	c2 active (the part detected at the entrance of the assembly station)
P6	Part awaiting assembly	da	Start of assembly
P7	Part being assembled	c3a	c3 active (the part has left the assembly station / part detected at the entrance of door B2)
P8	Part waiting to leave the assembly area	b2o	Door B2 opening
P9	Part leaving the assembly area	c4a	c4 active (the part has left door B2)
P10	Part taken out of the assembly area	b2f	Door B2 closing
P11	There are no parts waiting to leave the assembly loop	c5a	c5 active (part detected at the entrance of door B3)

P12	Piece waiting to leave the assembly loop	b3o	Door B3 opening
P13	Piece leaving the assembly loop	c5i	c5 inactive (the part is at the exit of B3)
P14	Piece taken out of the assembly loop	b3f	Door B3 closing
P15	Current number of parts in the assembly area		
P16	Number of parts waiting to leave the assembly loop		
P17	Number of places available in the assembly queue		
P18	Number of places occupied in the assembly queue		
P19	Number of places available in the exit queue		
P20	Number of places occupied in the exit queue		

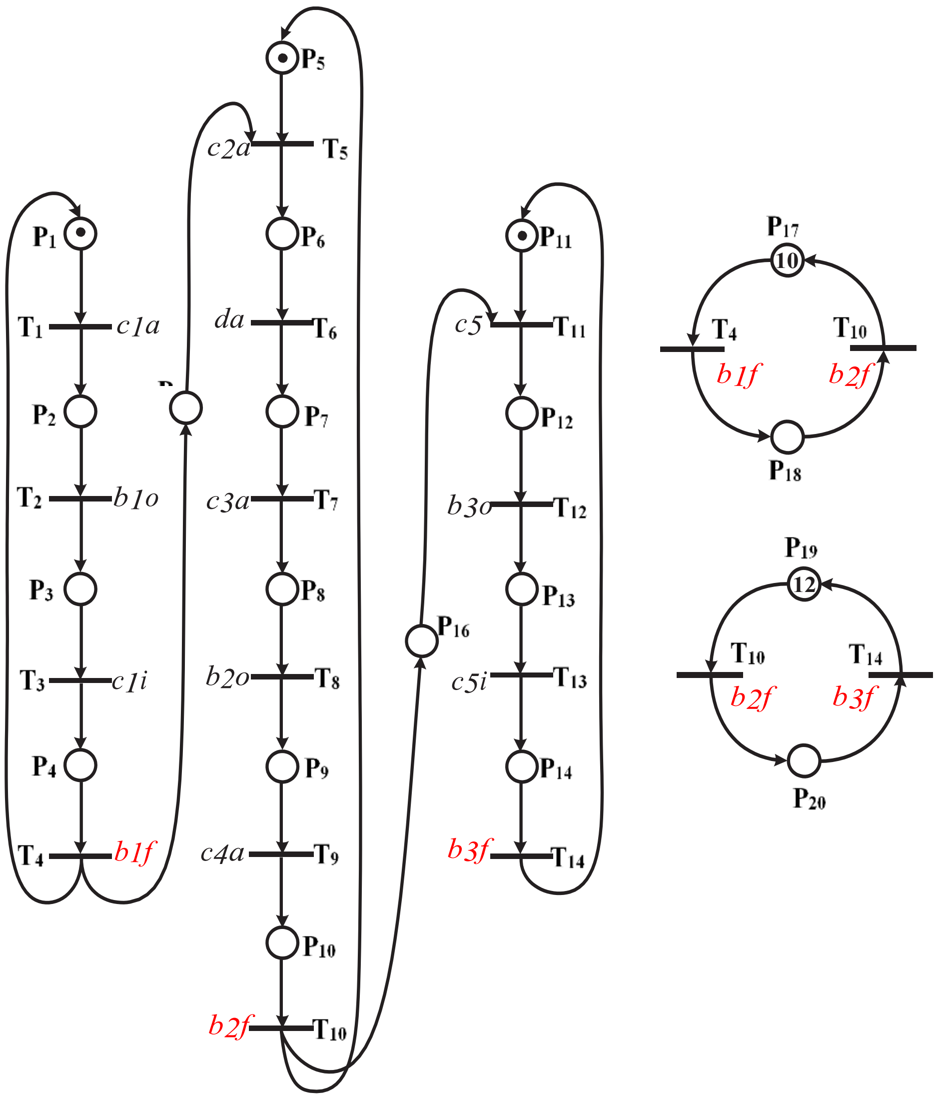

Figure 9: PNs of the plant and specification of assembly line [27]

All forbidden states $\mathcal{M}_b$ are consequences of the synchronization of plant PN with specification PN via uncontrollable transitions: $t_4|\mathcal{L}(t_4) = b1f$, $t_{10}|\mathcal{L}(t_{10}) = b2f$ and $t_{14}|\mathcal{L}(t_{14}) = b3f$. To ensure the respect of the specification, it is therefore necessary to define the controllability condition, namely

$$\begin{cases} M_S(P_{17}) \geq M_P(P_4) \\ M_S(P_{18}) \geq M_P(P_{10}) \\ M_S(P_{19}) \geq M_P(P_{10}) \\ M_S(P_{20}) \geq M_P(P_{14}) \end{cases}$$

The constraint is $L = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$

The characteristic of desired functioning PN (figure 9)

$$D = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \end{bmatrix}; \; M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 0 \\ 12 \\ 0 \end{bmatrix}$$

The controller of the assembly line can therefore be computed, that is

$$D_C = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix}; \qquad M_{0C} = \begin{bmatrix} 10 \\ 12 \\ 0 \\ 0 \end{bmatrix}$$

The controller portion corresponding to uncontrollable transitions $LD_u =$

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

The controller D_C is not admissible, since it draws an arc to uncontrollable transitions $t_3|\mathcal{L}(t_3) = c1i$, $t_9|\mathcal{L}(t_9) = c4a$, $t_{13}|\mathcal{L}(t_{13}) = c5i$ (see figure 9).

Iteration or step 1

- The Characteristic of controlled PN

$$D_{\mathcal{N}} = \begin{bmatrix} -1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & -1 & 0 \end{bmatrix} ; \; M_{0\mathcal{N}} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 10 \\ 0 \\ 12 \\ 0 \\ 10 \\ 12 \\ 0 \\ 0 \end{bmatrix}$$

- The controllability condition of controlled PN is

$$\begin{cases} M_C(P_{C1}) \geq M_P(P_3) \\ M_C(P_{C2}) \geq M_P(P_9) \\ M_C(P_{C3}) \geq M_P(P_9) \\ M_C(P_{C4}) \geq M_P(P_{13}) \end{cases}$$

- The new constraint is

$$L_{\mathcal{N}} = \begin{bmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix}$$

- The new controller is

$$D_{C_1} = \begin{bmatrix} 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -1 & 0 & 0 \end{bmatrix} ; M_{0C_1} = \begin{bmatrix} 10 \\ 12 \\ 0 \\ 0 \end{bmatrix}$$

- The controller portion corresponding to uncontrollable transitions:

$$L_{\mathcal{N}} D_{\mathcal{N}u} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}$$

The controller C_1 is admissible, since $L_{\mathcal{N}} D_{\mathcal{N}u} \leq 0$ and it draws no arc to uncontrollable transitions (modified arcs in Figure 9). It is the supreme controller.

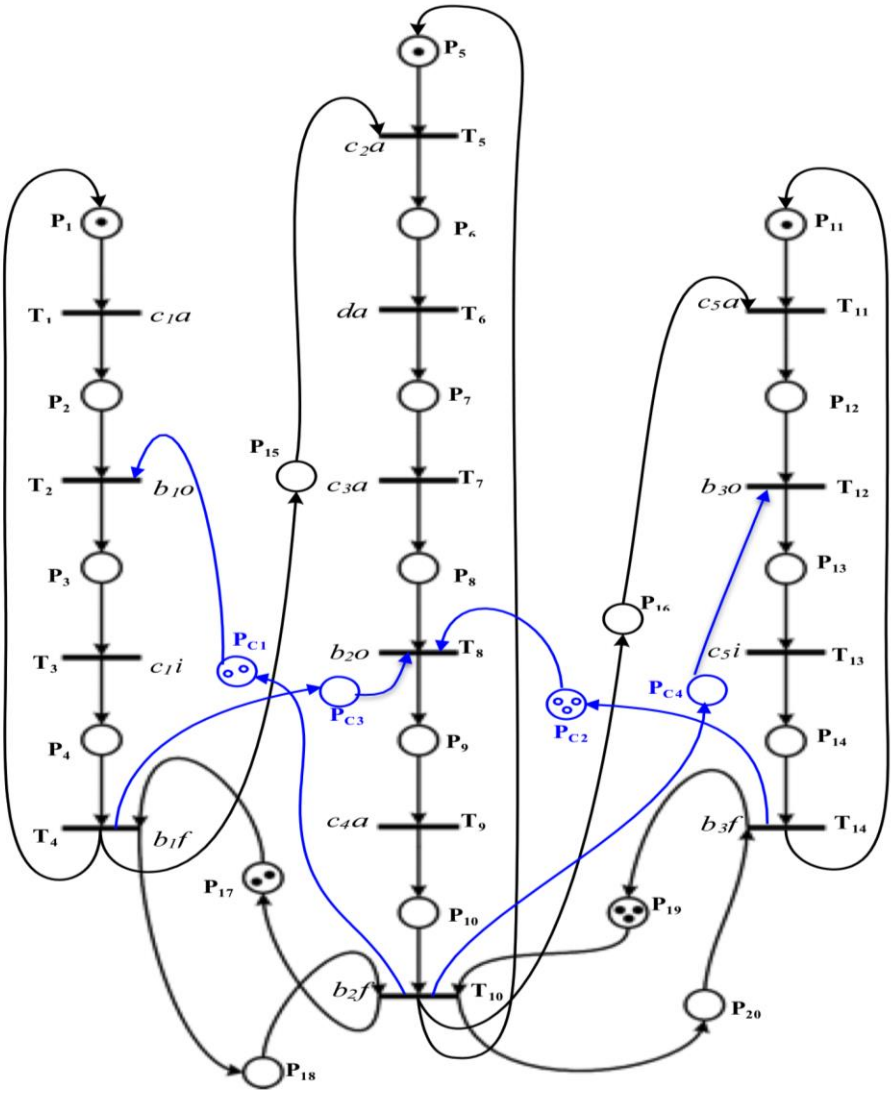

Figure 10: Supreme controller for the industrial assembly line

Conclusion

In this work, we propose a new idea based on iterations of the structural supervisory control method to ensure a systematic displacement of arcs in the non-admissible solution, so that no control place is the input place of uncontrollable transition. Note that the arcs displacement avoids creating implicit places in the final solution. Finally, the proposed approach has the advantage to offer, structurally, less restrictive and admissible controller, i.e, supreme controller; and allows implementing efficiently the supervisory control schemes using PN for the practical DES [28-30].

References

1. Cassandras, C. G., & Lafortune, S. (Eds.). (2008). *Introduction to discrete event systems*. Boston, MA: Springer US.
2. Cabasino, M. P., Hadjicostis, C. N., & Seatzu, C. (2016). Marking observer in labeled Petri nets with application to supervisory control. *IEEE Transactions on Automatic Control*, 62(4), 1813-1824.
3. Seatzu, C., Silva, M., & Van Schuppen, J. H. (2013). *Control of discrete-event systems* (Vol. 433). Springer.
4. David, R., & Alla, H. (2010). *Discrete, continuous, and hybrid Petri nets* (Vol. 1, pp. 17-130). Berlin: Springer.
5. Kumar, R., & Holloway, L. E. (1996). Supervisory control of deterministic Petri nets with regular specification languages. *IEEE Transactions on Automatic Control*, 41(2), 245-249.
6. Basile, F., Chiacchio, P., & Giua, A. (2006). Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places. *Automatica*, 42(6), 995-1004.
7. Wang, S., Wang, C., & Zhou, M. (2013). Design of optimal monitor-based supervisors for a class of Petri nets with uncontrollable transitions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(5), 1248-1255.
8. Gonza, M., Alla, H., & Bitjoka, L. (2020). Structural method for supervisory control of discrete event within the framework of Petri nets: a one-to-one link between the supervisory control theory and the place invariant method.

International Journal of Systems Science: Operations & Logistics, 7(3), 217-232.
9. Yamalidou, K., Moody, J. O., Lemmon, M. and Antsaklis, P. (1996). "Feedback Control of Petri Nets Based on Place Invariants. " *Automatica*, 32(1), 15-28.
10. Moody, J. O., & Antsaklis, P. J. (2000). Petri net supervisors for DES with uncontrollable and unobservable transitions. *IEEE Transactions on Automatic Control*, 45(3), 462-476.
11. Luo, J., Shao, H., Nonami, K., & Jin, F. (2012). Maximally permissive supervisor synthesis based on a new constraint transformation method. *Automatica*, 48(6), 1097-1101.
12. Ma, Z., Li, Z., & Giua, A. (2014). A constraint transformation technique for Petri nets with certain uncontrollable structures. *IFAC Proceedings Volumes*, 47(2), 66-72.
13. Wang, S., You, D., & Seatzu, C. (2017). A novel approach for constraint transformation in Petri nets with uncontrollable transitions. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 48(8), 1403-1410.
14. Ghaffari, A., Rezg, N., & Xie, X. (2003). Design of a live and maximally permissive Petri net controller using the theory of regions. *IEEE transactions on robotics and Automation*, 19(1), 137-141.
15. Luo, J., & Zhou, M. (2016). Petri-net controller synthesis for partially controllable and observable discrete event systems. *IEEE Transactions on Automatic Control*, 62(3), 1301-1313.
16. Luo, J., Wan, Y., Wu, W., & Li, Z. (2019). Optimal Petri-net controller for avoiding collisions in a class of automated guided vehicle systems. *IEEE Transactions on Intelligent Transportation Systems*, 21(11), 4526-4537.
17. Ramadge, P. J., & Wonham, W. M. (1989). The control of discrete event systems. *Proceedings of the IEEE*, 77(1), 81-98.
18. Wonham, W. M. and Kai, C. (2017). "Supervisory control of discrete-event systems. " Technical report, University of Toronto, 50(1), 1791-1797.
19. Kumar, R. (1991). "Supervisory Synthesis Techniques for Discrete Event Dynamical Systems. " Thesis for the degree of Doctor of Philosophy, University of Texas.
20. Iordache, M. V., Wu, P., Zhu, F., & Antsaklis, P. J. (2013, August). Efficient design of Petri-net supervisors with disjunctive specifications. In 2013 IEEE International Conference on Automation Science and Engineering (CASE) (pp. 936-941). IEEE.
21. Komenda, J., van Schuppen, J. H., Gaudin, B., & Marchand, H. (2008). Supervisory control of modular systems with global specification languages. *Automatica*, 44(4), 1127-1134.
22. Dideban, A., & Alla, H. (2008). Reduction of constraints for controller synthesis based on safe Petri nets. *Automatica*, 44(7), 1697-1706.
23. Gonza, M. (2019). "Development of a structural supervisory control method for discrete event systems modeled by Petri nets. " Thesis for the degree of Doctor of Philosophy, University of Ngagoundere. HAL open archives.
24. Giua, A. (2013). Supervisory control of Petri nets with language specifications. *In Control of discrete-event systems* (pp. 235-255). Springer, London.
25. Uzam, M. (2010). On suboptimal supervisory control of Petri nets in the presence of uncontrollable transitions via monitor places. *The International Journal of Advanced Manufacturing Technology*, 47(5), 567-579.
26. Wang, S., Wang, C., & Zhou, M. (2011, May). A transformation algorithm for optimal admissible generalized mutual exclusion constraints on Petri nets with uncontrollable transitions. *In 2011 IEEE International Conference on Robotics and Automation* (pp. 3745-3750). IEEE.
27. Vasiliu, A-I (2012). "Synthesis of controllers of discrete event systems based on Petri nets. " Thesis for the degree of Doctor of Philosophy, UJF Grenoble-France. HAL open archives.
28. Elhog-Benzina, D., Haddad, S., & Hennicker, R. (2012). Refinement and asynchronous composition of modal petri nets. *In Transactions on Petri Nets and Other Models of Concurrency V* (pp. 96-120). Springer, Berlin, Heidelberg.
29. Giua, A., & Seatzu, C. (2007). A systems theory view of Petri nets. *In Advances in control theory and applications* (pp. 99-127). Springer, Berlin, Heidelberg.
30. Hopcroft, J. E., Motwani, R., & Ullman, J. D. (2001). Introduction to automata theory, languages, and computation. Acm Sigact News, 32(1), 60-65.