Structural Design of Supreme Controller with Uncontrollable Transitions

Mohaman Gonza\textsuperscript{1,2}, Hassane Alla\textsuperscript{2}, Laurent Bitjoka\textsuperscript{1}

\textsuperscript{1} LESIA-Lab, GEEA Department, ENSAI, Ngaoundere University, BP 455 Ngaoundere, Cameroon
\textsuperscript{2} GIPSA-Lab, Grenoble Alpes University, BP 46 Grenoble, France

Corresponding Author Email: ecstudents06@yahoo.fr

\textbf{ABSTRACT}

This paper proposes an iterative control policy to design a less restrictive and admissible Petri net (PN) controller for a Discrete Event System (DES) with uncontrollable transitions, when a maximally permissive controller is not obtains by. This paper exploits a structural supervisory control method, avoiding reachability graph. This exciting method addresses the controllability condition of the desired functioning PN to define it as General Mutual Exclusion constraints (GMEC), which leads to design place invariant-based controller. The controller is non-admissible if one control place is connected to uncontrollable transition. To obtain an admissible PN controller, such a controller exists, authors propose constraints transformation, which is computationally complex, while the controller arcs displacement approach is unsystematic. Based on it, we develop the idea to iterate the structural supervisory control method, to ensure that control place is connected to controllable transition. Through this, it was found that the displacement of arcs is systematic and the controller is less restrictive and admissible, namely the supreme controller. The results indicate that the linear constraints are never violated through the firing of uncontrollable transitions. The finding of this work may serve to evaluate the structural optimality of the controller, in order to perform it practically.

1. INTRODUCTION

Discrete event systems (DES) are a dynamic event-driven system with discrete state. Compared to automata, the basic tool for modeling, Petri nets (PNs) have become interesting tool for analysis and control [1, 2]. For supervisory control, synchronous product between plant and specification models helps to mitigate the state explosion and gives structurally the desired functioning PN [3]. Unfortunately, almost the supervisory control methods are based on the reachability graph, subjected again to the state explosion phenomenon [4]. Moreover, very few works have addressed the controllability condition [5], even those focused to particular PNs [6, 7].

To solve these defects, this paper exploits the structural supervisory control method [8], established to design maximally permissive controller, avoiding reachability graph and that take into account controllability condition. Indeed, from the controllability condition, defined structurally on the desired functioning PN, the Generalized Mutual Exclusion Constraints (GMEC) is expressed to design the place invariant-based controller [9]. The structure of the resulting controller is relatively simple and implemented through control places connected to plant transitions. Naturally, if the control place is connected to uncontrollable transition, the controller is non-admissible since it cannot prevent the firing of such transition. At this point, it appears necessary to find the less restrictive and admissible controller. An intuitive method is to ride up the branches of PN plant until finding a controllable transition that is upstream of the control place [10]. Unhappily, this method is not systematic and effectively applicable. Besides, some solutions based on constraints transformation or modification [11, 12] may not represent the admissible controller corresponding to the original constraint [13], while others are computationally complex. For example, the algorithm to transform a constraint into a disjunction of admissible ones proposed by Luo and Zhou [14] requires the introduction a dynamic linear constraint in order to guarantee the optimal control solution. Recently a method presented by Luo et al. [15] to design a maximally permissive controller, requires partitioning the desired functioning PN into a set of dangerous regions to deal with uncontrollable transitions. However, the control action is to maintain the greatest number of tokens not more than 1 for any sequence, contrary to the structural supervisory control where the control action is to maintain the number of tokens of specification PN more or equal to the number of tokens of the plant PN, with respect of arcs weight of transitions. In addition, there is a need for algorithm to compute the set of controllable transitions that should be disabled by the controller.

This paper establishes an iterative process based on structural supervisory control, and applies it to controlled PN, with the aim to obtain structurally the less restrictive and admissible controller. The findings shed new light on constraint transformation if one considers the complexity the concerned approaches, and the systematic control policy adapted to ride up the branches of PN plant until finding a controllable transition. The less restrictive and admissible controller is considering as the supreme controller [16].

The remainder of this paper is organized as follows: Section 2 introduces the recall on supervisory control related to controllability notion, Section 3 describes briefly the PN tool and the structural supervisory control method and in Section 4...
the contribution of this paper is lights up the iterative process to structurally design supreme controller. Eventually, and in order to determine the relevance and the simplicity of the approach proposed, a case study is presented.

2. 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 [18], 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 [19]. A supervisor control is a feedback control (Figure 1) where a supervisor 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

This principle shows that the plant coupled with 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 Σ = Σc ∪ Σu, where Σc and Σu 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 specified 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 Σu, whose occurrence cannot be prevented by controller, leads to define the controllability condition.

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

\[ KΣ_u \cap L(G) \subseteq K \]  (1)

K ⊆ L(G) is prefix closed by construction; any sequence σ ∈ K, implies that every prefix of σ is in K, i.e., \(K\sigma = (\sigma' \in \Sigma^* \mid \exists \sigma \in K)\) 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 [19]. 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 [16] allows to compute a maximally permissive controller corresponding to the supreme controllable language, such as SupC(K) ≤ L(N_S) ∩ L(N_G).

In this paper, we focus on structural design of such controller, avoiding the complexity linked to languages or reachability graph [5]. Thus, we will use the structural supervisory method [8], which address PN controllability condition to design the controller in the presence of uncontrollable transitions [20]. 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_S||N_G. The language that characterizes the trajectory of the controller satisfies [19].

\[ L(N_S||N_G) = L(N_S) \cap L(N_G) \]  (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 adapted to ensure there exist no arc from a control place to an uncontrollable transition by using labeled PN. This idea is explored through the classical manufacturing system where we have brought some modifications (example 1).

3. PETRI NET TOOLS AND STRUCTURAL SUPERVISORY CONTROL

3.1 Petri nets tools

Figure 2. Topology of the modified classic manufacturing system

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 2.

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

- \( P = \{p_1, \ldots, p_n\} \) is the finite set of \( n \) places;
- \( T = \{t_1, \ldots, t_m\} \) is the finite set of \( m \) transitions;
- \( \Sigma \) is a finite set of events (labels) including the event always occurring \( \varepsilon \);
- \( D^+(\bullet, t_j) := P \times T \to \mathbb{Z} \) is the backwards incidence matrix that defines the weights of the directed arcs \((\bullet, t_j)\) from places \( p_i \) to transitions \( t_j \);
- \( D^-(\bullet, t_j) := P \times T \to \mathbb{Z} \) is the forwards incidence matrix, that define the weights of the directed arcs \((\bullet, t_j)\) from transitions \( t_j \), to places \( p_i \);
- \( M_0 \in \mathbb{N}^n \) is the initial marking or state. It is given by the number of tokens (black dot) in each place \( p_i \), denoted as \( M(p_i) \);
- \( L : T \to \Sigma \cup \{\varepsilon\} \) 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 \( t \) in \( T \) then \( L \) is \( \varepsilon \)-free. \( L \) is extended from transition sequence set \( T^* \) into \( \Sigma^* \), such that for \( \tau = t_1t_2 \cdots \in T^* \), \( \alpha = L(t_1)L(t_2) \cdots \in \Sigma^* \).

In a Labelled PN, firing a transition is linked to events occurrence, which can be portioned into uncontrollable events set \( \Sigma_u \) and controllable events set \( \Sigma_c \). By analogy, the set of uncontrollable transitions is denoted by \( T_u := \{t_j \in T | L(t_j) \in \Sigma_u\} \), and the controllable transitions set, \( T_c := \{t_j \in T | L(t_j) \in \Sigma_c\} \).

**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.

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

For this given example, the controllable transitions set is \( T_c := \{t_1, t_2, t_3, t_4\} \) and the uncontrollable transitions set is \( T_u := \{t_1, t_2, t_3, t_4\} \).

The PN dynamic can be represent by the presence/absence of tokens in the places. The marking or state \( M \) is a column vector; \( M = P \to \mathbb{Z} \) is a mapping function that assigns a non-negative integer (token count) to each place. For a transition \( t_j \in T \) we define the set of input places as \( ^\cdot t_j := \{p_i \in P | D^-(\bullet, t_j) > 0\} \).

If and only if \( M(p_j) \geq D^+(\bullet, t_j) \), 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 \( M_k[t_j] \);

\[
M_{k+1} = M_k + D(\bullet, t_j)
\]

where, \( D(\bullet, t_j) = D^+(\bullet, t_j) - D^-(\bullet, t_j) \) indicates for \( t_j \) the incidence matrix.

If the transition sequence \( \tau \in T^* \) is enabled from initial state \( M_0 \), denoted as \( M_0[\tau] \) the new state is reached, denoted as \( M_0[\tau]M_{k+1} \). We denote by \( R(N, M_0) \) the reachability graph, which is the set of reachable states from \( M_0 \), i.e,

\[
R(N, M_0) := \{M_k \in \mathbb{N}^n | \exists \tau \in T^* ; M_0[\tau]\}
\]

Given a Labelled PN \( N \), if we consider instead the transitions sequence, the events sequence (finite set) generated, then we can define PN language [21] as follows:

\[
L(N) = (\sigma \in \Sigma^* | \exists \tau \in T^* ; L(\tau) = \sigma \text{ and } M_0[\tau] \text{ is defined})
\]

Generally, PNs can represent more expressive and prefix closed languages in \( \Sigma^* \) than automata [22].

**3.2 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 \( N_p \) and specification PN \( N_s \), namely \( N_D \parallel N_S \), the controllability condition is established.

![Figure 3. Labelled PN of the modified classic manufacturing system](image)

![Figure 4. Desired functioning PN of the modified classic manufacturing system](image)
Definition 4 (Synchronous product). Let $N_G := (P_G, T_G, \Sigma_G, \Delta_G, D_G^+, M_{\text{BG}}, L_G)$ be the plant PN and $N_S := (P_S, T_S, \Sigma_S, D_S^+, M_{\text{BS}}, L_S)$ be the specification PN, both build on the same events set ($\Sigma = \Sigma_G$). Their synchronous product $N_G||N_S$ is another synchronized Petri net, $N := (P, T, \Sigma, \Delta, D^+, M_G, L)$, such that

- $P = P_G \cup P_S$
- $\Sigma = \Sigma_G \cup \Sigma_S$
- $T := T_G \cup T_S - T_{GS} \cap T_{OS} := \{(t_{G}, t_s) \in T_G \times T_S | L_C(t_G) \notin L_S(t_s)\}$
- $\Delta(t) := L_G(t) \cap T_G \cup L_S(t) \cap T_S$
- $D^* := \{\{(t_{G}, t_s), *\} \in P \times P | (t_{G}, *), t_s), \in D^+_G \text{ or } (t_{G}, *), t_s), \in D^+_S\}$
- $M_G(p_i) := M_{\text{BG}}(p_i), \text{if } p_i \in P_G \text{ or } M_{\text{BS}}(p_i), \text{if } p_i \in P_S$

Intuitively, the synchronous product is a matter of synchronous 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). 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.

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 [23, 24].

Definition 5 (Forbidden states). Let $t_j = (t_G, t_s) \in T_u$ be a synchronous transition in desired functioning PN, $M_G(\ast t_j)$ and $M_S(\ast t_j)$ the marking of input places belongs to plant $N_P$ and specification $N_S$ respectively; we define the set of forbidden states.

$$\mathcal{M}_k := \{ M_k \in \Gamma(\mathcal{N}, \mathcal{M}_0) | \exists t_j = (t_G, t_s) \in T_u, \text{ such that } M_G(\ast t_j) \geq D_C(\ast t_j), \text{ and } M_S(\ast t_j) < D_S(\ast t_j) \}$$

The desired functioning PN, $N := N_G||N_S$, is uncontrollable when a reachable state $M_G \in \mathcal{M}_k$. In the Figure 4 we face such situation when we consider the uncontrollable synchronous transition $t_4$, namely,

$$t_4 \in T_u; \{ M_G(P_1) \geq 1, M_S(P_3) < M_G(P_3) \}$$

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 $M_P(\ast t_j) \geq D_C(\ast t_j)$ is [8].

$$M_S(\ast t_j) \geq M_G(\ast t_j), \text{ with } M_S(\ast t_j) \geq D_S(\ast t_j) \hspace{1cm} (4)$$

It has been proven that the structural controllability condition is equivalent to that defined on the PN languages.

$$M_S(\ast t_j) \geq M_G(\ast t_j) \iff K \Sigma_u \cap L(N_G) \subseteq K$$

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, $\mathcal{M}_a$, is the one in which the structural controllability condition is verified.

$$\mathcal{M}_a := \{ M_k \in \Gamma(\mathcal{N}, \mathcal{M}_0) | \exists t_j = (t_G, t_s) \in T_u, M_G(\ast t_j) \geq M_G(\ast t_j) \}$$

The controllability condition can be expressed into linear constraints of GMEC type, denoted as $L^T M_k \leq b$, $\forall M_k \in \Gamma(\mathcal{N}, \mathcal{M}_0)$, with $L = [l_1 \cdots l_n] \in \mathbb{Z}^{n \times m}$ and $b \in \mathbb{Z}^m$.

For any $t_j = (t_G, t_s) \in T_u$, the controllability condition is expresses into inequality $M_G(\ast t_j) - M_S(\ast t_j) \leq 0$, where, $L = [0 \cdots 0, 0 \cdots 0, 0 \cdots 0]$, with $l_G = 1, l_S = -1$ and $b = 0$.

Hence, in the Figure 4, the controllability condition is $M_S(P_1) \geq M_G(P_1)$ and the corresponding constraint is $L = [0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 0 \ 0 \ -1]$, $b = 0$.

At this point, the control goal is to ensure 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 $M_0$ of the PN that implements a controller $C$ [22]. The place invariant based controller identical to the monitors [7, 20]. The controller is a PN with incidence matrix $D_C \in \mathbb{Z}^{n \times m}$ with initial state $M_0 \in \mathbb{Z}^n$, made up of the plant's transitions and a separate set of places.

$$D_C = -LD, M_0 = -LM_0 \hspace{1cm} (5)$$

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

Definition 8 (maximally permissive controller). A controller is maximally permissive if all the admissible state, $\mathcal{M}_a$, 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).

Figure 5. The controlled PN of the modified classic manufacturing system
Definition 9 (Controlled PN). A controlled PN is a triple $N = (N, C, B)$; where $N := N_G \cup N_C$ is the desired functioning PN, $C$ a PN model of the controller is a finite set of control places, $C \cap P = \emptyset$, 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:

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

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

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 $t_3|L(t_3) = v$ 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 $t_3|L(t_3) = v$ 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.

4. 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 firings 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 $M_N(\cdot t_j)$ is the marking of input places belonging to plant $N_G$, and $M_C(\cdot t_j)$ 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 $M_C(\cdot t_j)$ is less than weigh of their arcs $B(\cdot t_j)$, i.e., $M_C(\cdot t_j) \geq B(\cdot t_j)$. 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 [11]. Given $D$ the incidence matrix of $N := N_G \cup N_C$ and $L = [0 \cdots 0_C 0 \cdots 0_L 0 \cdots 0]$ the constraint from controllability condition $M_C(\cdot t_j) \geq M_N(\cdot t_j)$. Let $D_u$ be sub-matrix representing the uncontrollable part of $D$, such that $L_Du$ is the portion of controller corresponds to uncontrollable transitions. Let’s see $L_Du$ like the admissibility condition of designed controller. If $LD_u$ contains at least one strictly positive element, i.e $L_Du \geq 0$, then there is control place connected to uncontrollable transition ($t_j \in T_u$).

Consider in our example (Figure 4) the uncontrollable transitions set $T_u = \{t_2, t_5, t_6, t_6\}$ associated by label function to the uncontrollable events set transitions $\Sigma_u = \{r, v, e_1, e_2\}$ we will have,

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

For the controller to be less restrictive and admissible, the sufficient condition should be $L_Du \leq 0$, where the constraint $L = [0 \cdots 0_C 0 \cdots 0_C 0 \cdots 0]$ is given by controllability condition $M_N(\cdot t_j) \geq M_C(\cdot t_j)$.

When the condition $L_Du \leq 0$ is unsatisified, the idea is to iterate the structural supervisory control method from the controlled PN until finding uncontrollable transitions, which is upstream to the control places. Concretely, it is a question of extending the controllability 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 $M_N(\cdot t_j) \geq D(\cdot t_j)$ is

$$M_C(\cdot t_j) \geq M_N(\cdot t_j), \text{ with } M_C(\cdot t_j) \geq B(\cdot t_j)$$

(6)

Corollary. Let $L_N = [0 \cdots 0_C 0 \cdots 0_C 0 \cdots 0]$ be a constraint provided by an extending controllability condition $M_C(\cdot t_j) \geq M_N(\cdot t_j)$ to the controlled PN, $N = (N, C, B)$, and $D_{Nu}$ 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 $M_C(\cdot t_j) \geq M_N(\cdot t_j)$ 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 condition, this will result to connecting control place to a controllable transition, since the number of plant transitions is finite.

Consider the desired functioning PN, $N := N_G \cup N_C$ with controllability condition $M_N(\cdot t_j) \geq M_C(\cdot t_j)$ and the admissibility condition of controller $L_Du \geq 0$, 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, $N = (N, C, B)$ and $D_N$

Output: supreme controller $C_1$

Initialization step: From controlled PN, $N = (N, C, B)$ check if the controller $C$ draws no arc to uncontrollable transition ($t_j \in T_u$), i.e., $L_Du \leq 0$.

1: If not, set $L_N = [0 \cdots 0_C 0 \cdots 0_C 0 \cdots 0]$ is constraint provided by $M_C(\cdot t_j) \geq M_N(\cdot t_j)$. 

159
Supreme control step:
2: Do $i := 1$, $D_{c_1} = -L_N D_N$ and check $L_N^T D_{N_{iu}}$, where $D_{N_{iu}}$ is the sub-matrix representing the uncontrollable part of $D_N$
3: If $L_N^T D_{N_{iu}} \leq 0$, $C_i$ is less restrictive and admissible controller
4: If not i.e $L_N^T D_{N_{iu}} \not\leq 0$,
5: Repeat (1) for the next controller $N_i = (N, C_1, B_1)$. 
6: Do $D_{c_i} = -L_{N_{i-1}} D_{N_{i-1}}$ and check $L_{N_{i-1}}^T D_{N_{i-1u}}$
7: Until $L_{N_{i+1}}^T D_{N_{i+1u}} \leq 0$
8: For all $(t_j \in T)$ if we have always $L_{N_{i+1}}^T D_{N_{i+1u}} \leq 0$, 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, $t_1 | L(t_1) = v$.

**Iteration or step 1**

The characteristics of the controlled PN are

$$D_N = \begin{bmatrix}
-1 & 0 & 0 & 1 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0
\end{bmatrix} \quad \text{and } M_{0_N} = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
x \\
x
\end{bmatrix}$$

We put in red the columns correspond to $D_{N_{iu}}$.
The controllability condition is $M_C(P_C) \geq M_P(P_S)$;
The constraint is $L_N = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1]$;
The new controller is $D_{c_1} = \begin{bmatrix}
-1 \\
0 \\
0 \\
0 \\
0 \\
x \\
x
\end{bmatrix}$;

The controller portion corresponding to uncontrollable transitions $L_N^T D_{N_{iu}} = [1 \ 0 \ 0 \ -1]$.

The resulting controller is non-admissible, since $L_N^T D_{N_{iu}} \leq 0$ (one strictly positive element) and $D_{c_1}$ draws an arc to the supposed uncontrollable transition $t_2 | L(t_2) = r$ (Figure 6). Consequently, we must iterate the procedure again.

**Iteration or step 2**

Characteristic of controlled PN

$D_{N_{c_1}} = \begin{bmatrix}
-1 & 0 & 0 & 1 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & 0 & 1 & -1 \\
0 & 0 & 0 & -1 & 0 & 0 \\
0 & 0 & -1 & 0 & 1 & 0
\end{bmatrix}$ and $M_{0_{N_{c_1}}} = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
x \end{bmatrix}$

The controllability condition is $M_C(P_C) \geq M_P(P_S)$.
The constraint is $L_{N_{c_1}} = [0 \ 1 \ 0 \ 0 \ 0 \ 0 \ 0 \ -1]$.
The new controller is $D_{c_2} = \begin{bmatrix}
-1 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 1 & -1 & 0 \\
0 & 0 & 0 & -1 & 1 & 0 \\
0 & 0 & -1 & 0 & 1 & 0
\end{bmatrix}$ and $M_{0_{0_{c_2}}} = x$.

The controller portion corresponding to uncontrollable transitions $L_{N_{c_1}}^T D_{N_{c_1u}} = [0 \ 0 \ 0 \ -1]$.

This solution is less restrictive and admissible because $L_{N_{c_1}}^T D_{N_{c_1u}} \leq 0$ (none strictly positive element) and $D_{c_1}$ will draw an arc to the controllable transition $t_1 | L(t_1) = s_1$ (Figure 6). We must stop the iteration here, because we get the supreme controller.

The approach is systematic and structural, since we find the solution similar to the intuitive approach of Yamalidou et al. [10]. Now, suppose that transition $t_j | L(t_j) = s_1$ is uncontrollable transition, then the control solution is empty. It can be noticed that, in Figure 6 place $P_h$ 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 presented 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 by Yamalidou et al. [10] in an intuitive and unsystematic way. 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.

5. CASE STUDY

As a case study, consider the real-life system taken from Ref. [25]. It is in an industrial assembly line, whose topology is illustrated in Figure 7.

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 PN 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.

![Figure 7. Controlled PN $\mathcal{X}_2$, after 2nd step, with supreme controller $\mathcal{C}_2$](image)

![Figure 8. Topology of industrial assembly line [10]](image)

Table 1. Description of places and events associated with transitions

| Places          | Description                                      |
|-----------------|--------------------------------------------------|
| P1              | There is no room at the entrance to the assembly area |
| P2              | Part waiting to enter the assembly area          |
| P3              | Part entering the assembly area                  |
| P4              | Part entered in the assembly area                |
| P5              | No part is awaiting assembly                     |
| P6              | Part awaiting assembly                           |
| P7              | Part being assembled                             |
| P8              | Part waiting to leave the assembly area          |
| P9              | Part leaving the assembly area                   |
| P10             | Part taken out of the assembly area              |
| P11             | There are no parts waiting to leave the assembly loop |
| P12             | Piece waiting to leave the assembly loop         |
| P13             | Piece leaving the assembly loop                  |
| P14             | Piece taken out of the assembly loop             |
| 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      |

| Transitions     | Description                                      |
|-----------------|--------------------------------------------------|
| c1a             | There is no room at the entrance to the assembly area |
| b1o             | Part waiting to enter the assembly area          |
| c1i             | Part entering the assembly area                  |
| b1f             | Part entered in the assembly area                |
| c2a             | No part is awaiting assembly                     |
| D1              | Part awaiting assembly                           |
| c3a             | Part being assembled                             |
| b2o             | Part waiting to leave the assembly area          |
| c4a             | Part leaving the assembly area                   |
| b2f             | Part taken out of the assembly area              |
| c5a             | There are no parts waiting to leave the assembly loop |
| b3o             | Piece waiting to leave the assembly loop         |
| c5i             | Piece leaving the assembly loop                  |
| b3f             | Piece taken out of the assembly loop             |

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

\[
\begin{align*}
M_5(P_{12}) & \geq M_5(P_6) \\
M_5(P_{10}) & \geq M_5(P_{10}) \\
M_5(P_{14}) & \geq M_5(P_{14}) \\
M_5(P_{20}) & \geq M_5(P_{20})
\end{align*}
\]

The constraint $L$ is in following equation.

The characteristic of desired functioning PN (Figure 9).

The controller of the assembly line can therefore be computed, that is in following equation (DC).
\[
L = \begin{bmatrix}
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
\end{bmatrix}
\]

\[
D = \begin{bmatrix}
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \\
\end{bmatrix}, \quad M_0 = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
10 \\
0 \\
0 \\
\end{bmatrix}
\]

\[
D_C = \begin{bmatrix}
0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \quad M_{0C} = \begin{bmatrix}
10 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

The controller portion corresponding to uncontrollable transitions.

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

The controller \(D_C\) is not admissible, since it draws an arc to uncontrollable transitions \(t_4|L(t_3) = c1i, t_9|L(t_9) = c4a\), \(t_{13}|L(t_{13}) = c5i\) (see Figure 9).

**Iteration or step 1**

- The Characteristic of controlled PN

\[
D_N = \begin{bmatrix}
-1 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{bmatrix}, \quad M_{0N} = \begin{bmatrix}
1 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

- The controllability condition of controlled PN is

\[
\begin{align*}
M_C(P_{c1}) & \geq M_P(P_0) \\
M_C(P_{c2}) & \geq M_P(P_0) \\
M_C(P_{c3}) & \geq M_P(P_0) \\
M_C(P_{c4}) & \geq M_P(P_{13})
\end{align*}
\]
• The new constraint $L_N$ is

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

• The new controller is

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

• The controller portion corresponding to uncontrollable transitions:

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

The controller $C_1$ is admissible, since $L_ND_{N_u} \leq 0$ and it draws no arc to uncontrollable transitions (modified arcs in Figure 10). It is the supreme controller.

![Figure 9](image1.png)  
**Figure 9.** PNs of the plant and specification of assembly line

![Figure 10](image2.png)  
**Figure 10.** Supreme controller for industrial assembly line

### 6. CONCLUSIONS

This topic is intensive so that it has been the subject of several studies for many years. In this work, we propose a structural framework based on iteration to obtain a less restrictive and admissible controller (supreme controller), when the control place is connected to uncontrollable transition. We observed that, the iteration of the structural supervisory control that deal with uncontrollable synchronizations, ensures a systematic displacement of arcs in the controlled PN, with minimum of control structures. As extension of this work; we are going to cover the design of observers/controllers in labeled PN with unobservable transitions.

### ACKNOWLEDGMENT

This research has no financial support. The authors like to thank the editors and reviewers for their constructive comments and suggestions, which helped us to improve the quality of these papers.

### REFERENCES

[1] Cabasino, M.P., Hadjicostis, C.N., Seatzu, C. (2017). Marking observer in labeled Petri nets with application to supervisory control. IEEE Transactions on Automatic Control, 62: 1813-182.
Zhu, G., Wang, Y., Wang, Y.J. (2019). Model identification of unobservable behavior of discrete event systems using petri nets. Journal of Control Science and Engineering, 2019: 2895685. https://doi.org/10.1155/2019/2895685

Seatzu, C., Silva, M., Van Schuppen, J.H. (2013). Control of discrete-event systems (Vol. 433). Springer.

Rashidinejad, A., Van der Graaf, P., Reniers, M., Fabian, M. (2020). Non-blocking supervisory control of timed automata using forcible events. IFAC-PapersOnLine, 53(4): 356-362. https://doi.org/10.1016/j.ifacol.2021.04.035

Han, X., Chen, Z., Zhang K., Liu, Z., Zhang, Q. (2020). Modeling, reachability and controllability of bounded petri nets based on semi-tensor product of matrices. Asian Journal of Control, 22(1): 500-510. https://doi.org/10.1002/asjc.19115

Basile, F., Chiachio, P., Giua, A. (2006). Suboptimal supervisory control of Petri nets in presence of uncontrollable transitions via monitor places. Automatica, 42(6): 995-1004. https://doi.org/10.1016/j.automatica.2006.02.003

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.

Gonzá, 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. https://doi.org/10.1080/23302674.2019.1567860

Chen, C., Hu, H. (2022). Extended place-invariant control in automated manufacturing systems using petri nets. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 52(3): 1807-1822. https://doi.org/10.1109/TSMC.2020.3035668

Yamalidou, K., Moody, J.O., Lemmon, M., Antsaklis, P. (1996). Feedback control of petri nets based on place invariants. Automatica, 32(1): 15-28. https://doi.org/10.1016/0005-1098(95)00103-4

Moody, J., Antsaklis, P. (2000). Petri net supervisors for DES with uncontrollable and unobservable transitions. IEEE Transactions on Automatic Control, 45: 462-476. https://doi.org/10.1109/9.847725

Ma, Z., Li, Z., Giua, A. (2014). A constraint transformation technique for petri nets with certain uncontrollable structures. IFAC Proceedings, 47(2): 66-72. https://doi.org/10.3182/20140514-3

Wang, S., You, D., Seatzu, C. (2018). A novel approach for constraint transformation in petri nets with uncontrollable transitions. IEEE Transactions on Systems, Man, and Cybernetics: Systems, 48(8): 1403-1410. https://doi.org/10.1109/TSMC.2017.2665479

Luo, J., Zhou, M. (2017). Petri-net controller synthesis for partially controllable and observable discrete event systems. IEEE Transactions on Automatic Control, 62(3): 1301-1313. https://doi.org/10.1109/TAC.2016.2586604

Luo J., Wan, Y., Wu, W., Li, Z. (2020). 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. https://doi.org/10.1109/TITS.2019.2937058

Kumar, R., Holloway, L. (1996). Supervisory control of deterministic petri nets with regular specification languages. IEEE Transactions on Automatic Control, 41(2): 245-249. https://doi.org/10.1109/9.481527

Lafortune, S. (2019). Discrete event systems: Modeling, observation, and control. Annual Review of Control, Robotics, and Autonomous Systems, 2: 141-159. https://doi.org/10.1146/annurev-control-053018-023659

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. https://doi.org/10.1007/978-3-642-29072-5_4

Wonham, W.M., Cai, K., Rudie, K. (2017). Supervisory control of discrete-event systems: A brief history-1980-2015. IFAC-PapersOnLine, 50(1): 1791-1797. https://doi.org/10.1016/j.ifacol.2017.08.164

Uzam, M. (2010). On suboptimal supervisory control of Petri nets in the presence of uncontrollable transitions via monitor places. International Journal of Advanced Manufacturing Technology, 47(5-8): 567-579. https://doi.org/10.1007/s00170-009-2219-0

López, J., Santana-Alonso, A., Díaz-Cacho Medina, M. (2019). Formal verification for task description languages. A petri net approach. Sensors, 19(22): 4965. https://doi.org/10.3390/s19224965

Giua, A. (2013). Supervisory control of Petri nets with language specifications. Control of Discrete-Event Systems, pp. 235-255. https://doi.org/10.1007/978-1-4471-4276-8_12

Dideban, A., Alla, H. (2008). Reduction of constraints for controller synthesis based on safe petri nets. Automatica, 44(7): 1697-1706. https://doi.org/10.1016/j.automatica.2007.10.031

Cong, X., Wang, A., Chen, Y., et al. (2019). Most permissive liveness-enforcing Petri net supervisors for discrete event systems via linear monitors. ISA Transactions, 92: 145-154. https://doi.org/10.1016/j.isatra.2019.02.003

Vasiliu, A.I. (2012). Synthesis of controllers of discrete event systems based on petri nets. Ph.D. Thesis. UJF Grenoble-France. HAL open archives.