

# Online Activation and Deactivation of a Petri Net Supervisor

Sadok Rezig <sup>\*</sup>, Nidhal Rezg and Zied Hajej 

Laboratoire de Génie Informatique, de Production et de Maintenance, UFR MIM, Université de Lorraine, 57000 Metz, France; nidhal.rezg@univ-lorraine.fr (N.R.); zied.hajej@univ-lorraine.fr (Z.H.)

\* Correspondence: sadok.rezig@univ-lorraine.fr

**Abstract:** This paper highlights algebraic and mathematical properties in symmetry with Petri nets in order to control automated systems such as flexible workshops, which represent one of the most important examples in industry and for discrete event systems in general. This project deals with the problem of forbidden state transition by using a new application of the theory of regions for supervisory control. In the literature, most control synthesis methods suffer greatly from a cumbersome calculation burden of the Petri net supervisor given the complex exploration of the state graph. Our new methodology lightens the computational load of the Petri net supervisor by choosing specific regions on the reachability graph, on which the control is calculated offline using CPLEX. The determined controller is activated online if the process enters the chosen region, and deactivated otherwise. All our experiments were applied in a flexible workshop implemented in our research laboratory, which was used to engrave selected models on glass blocks of different colors.

**Keywords:** supervisory control; flexible manufacturing system; discrete event system; theory of regions; activation; deactivation



**Citation:** Rezig, S.; Rezg, N.; Hajej, Z. Online Activation and Deactivation of a Petri Net Supervisor. *Symmetry* **2021**, *13*, 2218. <https://doi.org/10.3390/sym13112218>

Academic Editor: Manuel Lafond

Received: 21 October 2021

Accepted: 11 November 2021

Published: 20 November 2021

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Unlike continuous systems, discrete event systems (DESs) represent dynamic systems in which the reachability graph is discrete. The time use of these kinds of systems does not interest us, because only the occurrence of events matters in our case studies. The appearance of a single event causes the generation of a new state.

If there is no occurrence of an event, the system remains unchanged (i.e., there is no state change). An event can correspond to the transmission of a message in a communication network, to the occurrence of a disturbance in a transport network, or to the arrival of a part in a production cell.

Comparing them to automata, PNs [1–3] have the advantage of being much more general models, benefiting from much richer structures. Thus, PNs fit perfectly into the description of certain types of DESs. An automaton is nothing other than a marking graph generated from a Petri net.

The use of symmetry concepts between known mathematical properties and Petri nets constitutes a strong point to respond to key issues in the field of control synthesis.

Nowadays, the use of Petri nets is almost always necessary when it comes to complex systems requiring a high level of precision and safety, such as in the case of the petroleum industry or nuclear power plants. Thus, the use of our methodology in symmetry with the notions of mathematics applied to concrete examples will make it possible to ensure these kinds of systems perform a desired behavior that satisfies specific industrial constraints.

The methodology presented in this work is based on the theory of regions [4,5] in control synthesis presented by Ghaffari et al. [6–9]. This theory can be presented and summarized by a linear system to solve, and then one can calculate the Petri net supervisor offline.

The calculation of this PN controller is useful for preventing the production system from blocking, as well as preventing dangerous markings and forbidden states prohibited

by the control specifications. Moreover, the combinatorial explosion of states on the generated reachability graph increases the computational task of the PN monitor. In this paper, the theory of regions is not applied on the whole state space, but in a specific region of the reachability graph. Therefore, the PN supervisor is activated/deactivated online if the process enters or exits in/from the chosen region (specific zone (SZ)). All our experiments were applied in a flexible manufacturing system already installed in our research laboratory.

Several research projects have been established in the field of control synthesis to reduce the computational load of the Petri net supervisor. Among these works, one can cite the work of Bashir et al. and Der Jeng [10,11]; this study proposed a new approach to calculating a supervisory structure using combined control places and control transitions to ensure flexible manufacturing systems' smooth operation. A place-transition monitor was designed for each concurrent process of the systems. Thus, three algorithms were proposed. The first one computes the loop markings at each process of the PN model. Then, algorithm II is used to sort the deadlock markings based on the concurrent processes of the PN model. The transition-place controller and place-transition supervisor are designed using algorithm III.

In the literature, several other studies focused on reachability graph analysis to determine a near-optimal Petri net controller [12–16]. Their approaches are all based on the exploration of the whole graph, or on dividing the graph into a deadlock zone and a live zone. Additionally, Chen et al. [17] studied a vector covering method to reduce the number of markings/states to study. All these approaches can decrease the number of Petri net controllers. Nonetheless, they have a major drawback: the calculation burden is extremely difficult, especially for big models.

In our previous studies, we addressed the design of a PN controller for forbidden transitions/states presented as general mutual exclusion constraints (GMECs). Therefore, as with any supervisory control method based on the whole reachability graph, the use of the graph is a fundamental step in the theory of regions. In addition, like any method based on the reachability graph, which may contain many markings, with respect to the structural size of the system, the design of the Petri net controller becomes harder and even impossible. The main contribution of those studies is the development of a control synthesis without using the state space. Therefore, based on mathematical concepts and Petri net properties, the proposed approach provides a maximally permissive Petri net controller for bounded Petri nets following the theory of regions interpretation. However, these methods are not canonical for some models, and assumptions have been considered to maintain the efficiency of the methodology.

The remainder of this work is presented as follows: Tools used in this work, such as the theory of regions or Petri nets [18], are discussed in Section 2. Our methodology is explained in Section 3. The experiments in Section 4 were applied in a flexible manufacturing system to illustrate the effectiveness of our method. Finally, comparison results with examples taken from [9,19] are described in Section 5, with our conclusion and perspectives drawn in Section 6.

## 2. Tools and Methods

The Petri net supervisor in our methodology can be defined by a set of control places  $\sum P_c$ . The control place  $P_c$  can be calculated by solving the linear system of the theory of regions [18–22]. This linear system is composed of three types of conditions. The reachability conditions can be expressed by the following equation:

$$M_0(P_c) + C(P_c, \cdot) \vec{\Gamma}_M \geq 0 \quad (1)$$

where  $\vec{\Gamma}_M$  is the path between  $M_0$  and  $M$ .

The second type of conditions in the linear system of the theory of regions is represented by the cycle equation as follows:

$$\sum_{t \in T} C(P_c, t) \cdot \vec{\sigma}[t] = 0, \forall \sigma \in \Delta \quad (2)$$

where  $\vec{\sigma}[t]$  represents the algebraic sum of occurrences of  $t$  in  $\sigma$ , and  $\Delta$  represents the set of existing cycles on the graph.

Finally, the control place  $P_c$  has to satisfy the marking/transition separation instance (MTSI) equation:

$$M_0(P_c) + C(P_c, \cdot) \vec{\Gamma}_M + C(P_c, t) < 0 \quad (3)$$

Each MTSI condition is relative to one forbidden event  $(M, t)$ ; the marking  $M$  is a reachable marking and  $t$  is a prohibited transition. Remarkably, the MTSI conditions can have the same solutions by solving Equations (1)–(3). Consequently, the control places are smaller than the set of MTSI equations.

By resolving this linear system, one can calculate the Petri net supervisor composed of a set of control places  $\sum P_c$  characterized by its initial marking  $M_0(P_c)$  and its incidence vectors  $C(P_c, \cdot)$ .

So, as stated at the beginning, Petri nets are the manufacturing systems on which the work is modeled. A PN is a mathematical modeling tool for discrete event systems and many other systems. From an informal point of view, this modeling is represented by two types of nodes: the transitions  $T$  and the places  $P$ . These places and transitions are related by oriented arcs. A place can never be connected to another place; a transition cannot be connected to another transition. Moreover, from a formal point of view, a PN is a bipartite graph:  $PN = \langle P, T, Pre, Post \rangle$ , where  $T$  is a set of controllable and uncontrollable transitions [23].  $Pre : P \times T \rightarrow \mathbb{N}$  represents a pre-incidence function that specifies weighted arcs from  $P$  to  $T$ .  $Post : P \times T \rightarrow \mathbb{N}$  is a post-incidence function that specifies weighted arcs from  $T$  to  $P$  ( $\mathbb{N}$  is a set of nonnegative integers). Let the sets  $p^{(t)}$  and  ${}^{(t)}p$  be the output transitions and the input transitions, respectively, of a place  $p$ . A Petri net can be expressed by an indexed matrix  $C$  such that  $C(p_i, t_j) = w(t_j, p_i)$  if  $t_j \in {}^{(t)}p_i$  and  $C(p_i, t_j) = -w(p_i, t_j)$  if  $t_j \in p_i^{(t)}$ , else 0, where  $w : F \rightarrow \mathbb{N}$  is a valuation function of arcs (the finite set of arcs  $F \subseteq (P \times T) \cup (T \times P)$ ). Moreover, let  $t^{(p)}$  and  ${}^{(p)}t$  be the sets of output places and input places, respectively, of a transition  $t$ . The graph constructed from the initial marking  $M_0$  is denoted by  $G(N, M_0)$ . The set of generated markings in RG is denoted by  $M$ . A transition is enabled from a marking  $M \in M$  (denoted by  $M[t >]$ ) if and only if  $M \geq Pre(\cdot, t)$ . If loops are not considered, an enabled transition may fire-yield a new marking  $M'$  such that  $M' = M + C(\cdot, t)$ . This expression can be defined by  $M[t > M']$ . A new marking  $M'$  is reachable from a marking  $M$  if a firing sequence  $\sigma = t_1, t_2, \dots, t_n$  exists by firing  $\sigma$ , and any marking  $M'$  attainable from the initial marking  $M_0$  satisfies the following PN state equation:  $M' = M_0 + C \cdot \vec{\sigma}$ , where  $\vec{\sigma} : T \rightarrow \mathbb{N}$  is a vector of non-negative integers called the occurrence of  $t_i$  in  $\sigma$ .

The tools are presented in Section 2 by giving an example of the application of the theory of regions with a Petri net (i.e., see Figures 1 and 2). One can calculate the PN supervisor with the classical method of the theory of regions.

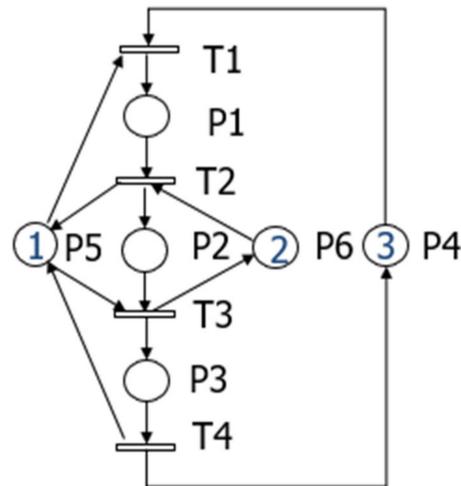


Figure 1. The Petri net example.

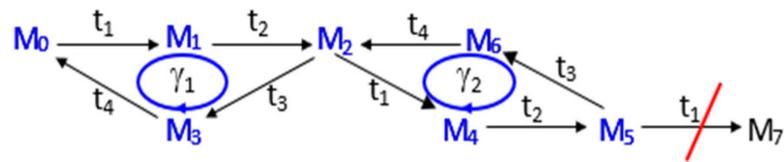


Figure 2. The generated reachability graph.

The application of the classic theory of regions in supervisory control can give us the following linear system (Equations (1)–(3)):

$$M_0(P_c) \geq 0$$

$$M_1(P_c) = M_0(P_c) + C(P_c, T_1) \geq 0$$

$$M_2(P_c) = M_0(P_c) + C(P_c, T_1) + C(P_c, T_2) \geq 0$$

$$M_3(P_c) = M_0(P_c) + C(P_c, T_1) + C(P_c, T_2) + C(P_c, T_3) \geq 0$$

$$M_4(P_c) = M_0(P_c) + 2C(P_c, T_1) + 2C(P_c, T_2) + C(P_c, T_3) \geq 0$$

$$M_5(P_c) = M_0(P_c) + 2C(P_c, T_1) + 2C(P_c, T_2) \geq 0$$

$$M_6(P_c) = M_0(P_c) + 2C(P_c, T_1) + C(P_c, T_2) \geq 0$$

State number 7 is a blocking state [21] and will lead to a deadlock. So, it is considered a forbidden state. The MTSI is  $(M_5 \xrightarrow{T_1} M_7)$ :

$$C(P_c, T_1) + C(P_c, T_2) + C(P_c, T_3) + C(P_c, T_4) = 0$$

$$M_7(P_c) = M_0(P_c) + 2C(P_c, T_1) + 2C(P_c, T_2) + C(P_c, T_1) < 0$$

All these equations constitute the linear system of the theory of regions to solve in order to calculate the control place  $P_c$ , characterized by its initial marking and incidence vector. Consequently, the linear system is composed of nine conditions, and its resolution allows one to determine this Petri net controller:  $M_0(P_c) = 2; C(P_c, \cdot) = (-1, 0, 0, 1)$ .

Therefore, the Petri net controller is given in the following Figure 3.

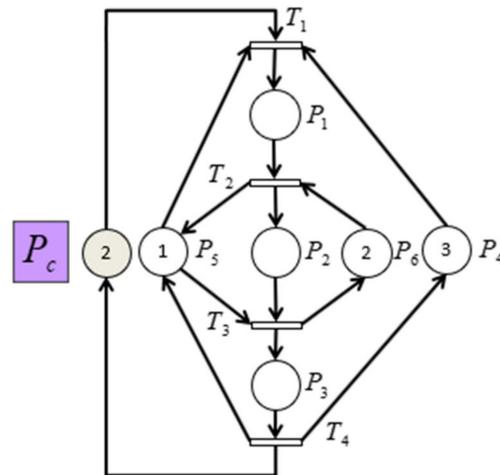


Figure 3. The controlled Petri net.

In order to better understand our methodology and our control algorithm, some concepts are defined as follows:

**Definition 1.** A forbidden state ( $M \in MI$ ) is a state that does not respect a control specification modeled by the following:  $GMEC(\vec{w}, K): MI = \left\{ M \in M \mid \vec{w}^t \cdot M \geq K \right\}$ , where  $\vec{w}$  is a weight vector of non-negative integers and  $K$  is a positive integer.

**Definition 2.**  $\Omega$  is the set of event separation instances that the PN supervisor must prohibit:  $\Omega = \{(M, t) / \exists M[t > M', M \in G(N, M_0) M' \in MI]\}$

**Definition 3.** A blocking state ( $M \in MB$ ) is a marking from which there is no transition validated by it, i.e., there is no path that can return us to the initial state  $M_0$

**Definition 4.** A state ( $M \in MS$ ) is said to be a source state if the process can enter from this marking to the specific zone on which the theory of regions is applied:

$$MS = \{M \in G(N, M_0) \mid \exists M[t > M', M \in \Omega M' \in MI \cup MB]\}$$

**Definition 5.** A frontier state ( $M \in MF$ ) is a marking from which one can activate and deactivate the Petri net supervisor.

### 3. Methodology and Algorithm

This section focuses on the reduction in the computational burden on the Petri net supervisor by using a different application of the theory of regions on the reachability graph. Equations (1)–(3) are employed on specific regions and not on the whole graph. Therefore, the computation time decreases significantly.

**Definition 6.** A specific zone,  $SZ_i$ , is a region on the reachability graph  $G(N, M_0)$  with a certain number of states and correlated MTSIs. Each specific zone  $SZ_i$  corresponds to a control place  $Pc_i$

**Definition 7.** Two MTSIs,  $\Omega_1$  and  $\Omega_2$ , are correlated if the corresponding forbidden transitions are the same.

**Proposition 1.** The application of the theory of regions in our methodology begin from the initial source state ( $MS_{0i} \in SZ_i$ ). This initial source state is determined by firing the smallest transition sequence  $\sigma$  [24].

**Proof.** Let  $v$  and  $v'$  be the respected transition vectors of  $M$  and  $M'$ , respectively, such as  $M, M' \in G(N, M_0)$ ; and let  $C$  be the incidence matrix of a given Petri net.  $\sigma$  and  $\sigma'$  are the associated transition firing vectors of  $M$  and  $M'$ , respectively, such that  $M = M_0 + C \cdot \vec{\sigma}$  and  $M' = M_0 + C \cdot \vec{\sigma}'$ . From [24], if  $\sigma' \geq \sigma$ , there exists a transition firing sequence:  $\sigma'' = \sigma' - \sigma \mid M[\sigma'' > M'$  then  $v$  is smaller than  $v'$ .  $\square$

### 3.1. Control Policy Algorithm

The supervisory control policy using the new interpretation of the theory of regions is summarized in the flowchart below (see Figure 4). The main idea of the proposed method is to calculate the set of control places (i.e.,  $\sum_i P_{ci}$ ) that connect the initial state  $MS_0$  to the other markings (i.e.,  $MI, MF,$  and  $MB$ ). Then, one can resolve the linear system of the theory of regions to synthesize the PN controllers offline using CPLEX.

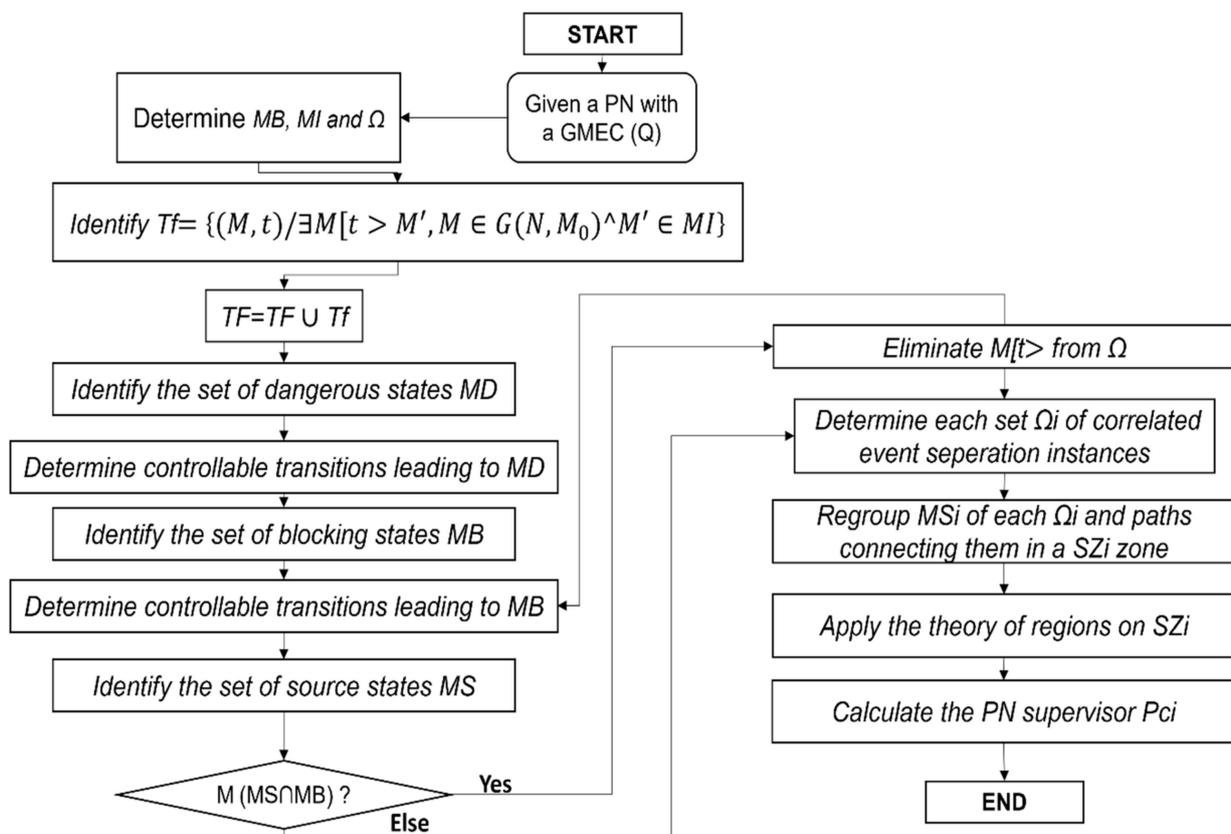


Figure 4. Flowchart of the control method.

After having determined the reachability graph, the monitor imposes a GMEC control constraint on the production system. In order to act as quickly as possible and to respond effectively to industrial constraints, the algorithm presented in Figure 4 first identifies the blocking and prohibited states that do not respect the GMEC. This identification allows us to determine the set of forbidden transitions  $Tf$ . If the uncontrollable transitions exist, other dangerous states  $MD$  are to be identified as well. Based on the set of all prohibited transitions, specific zones ( $SZs$ ) will be grouped according to the correlation criterion of  $MTSi$ .

Once the area is detected, the theory of regions is applied over the entire zone  $SZi$  to synthesize the corresponding controller.

This PN controller is activated if the process enters  $SZi$  and is deactivated if the process is no longer there. Finally, all the control places constitute our PN supervisor.

### 3.2. Boolean Control Function Using the Activation/Deactivation Vector

After calculating the Petri net supervisors  $P_{ci}$  for each  $SZ_i$ , one can calculate the Boolean control function  $FBC$  that can deactivate and activate the PN controllers. Consequently, the role of each monitor is to control its zone ( $SZ_i$ ) on the graph.

The online function can allow the switch/alternation between the PN supervisors on the reachability graph by activating or deactivating each Petri net controller  $P_{ci}$ .

So, after identifying the set of specific zones ( $SZs$ ) of the reachability graph, one can proceed to the activation/deactivation step. Once a marking  $M_i$  enters or exits a  $SZ_i$ , each controller ( $P_{ci}$ ) must control its  $SZ_i$ . Initially, the controller is inactive until a marking enters a  $SZ_i$  using the deactivation/activation vector.

**Definition 8.** For each state  $M$  in the graph, an activation vector is calculated. The activation/deactivation vector is a path between the initial state of the graph  $M_0$  and  $M$ .

The activation vector  $\sigma_A = \{\sigma \in T^* / \exists M_0[\sigma > M', M' \in SZ_i]\}$ .

The deactivation vector  $\sigma_D = \{\sigma \in T^* / \exists M_0[\sigma > M'] \wedge M'[t >, M' \in SZ_i, t \notin SZ_i]\}$ .

The PN supervisors can control the system either alternately or simultaneously depending on the process state. Consequently, the necessary PN supervisor is activated or deactivated.

$$FBC = F(P_{c1}, P_{c2}, \dots, P_{cn})$$

$$\Leftrightarrow FBC = P_{c1} + P_{c2} + \dots + P_{cn}$$

$$\Rightarrow FBC = \sum P_{cn}$$

An OR function (logical function) characterizes the union of the PN supervisors and takes the following values:

$$FBC = \begin{cases} 1 & \text{If the system is controlled} \\ 0 & \text{Otherwise} \end{cases}$$

## 4. PN Example

A flexible manufacturing system implemented in our laboratory (LGIPM) in France is presented in Figure 5. This production cell produces engraved glass cubes in different colors with digital models that can be chosen. Each engraved piece is stored, assembled with a base, and then packaged as a finished product.



**Figure 5.** Flexible manufacturing system in UFR MIM.

In our study, only two assembly stations (station no. 3 + station no. 4) of the FMS could be considered. The modeling of this physical system is given in the following Figure 6. The first assembly station is modeled by P2, while the second station is represented by P3.

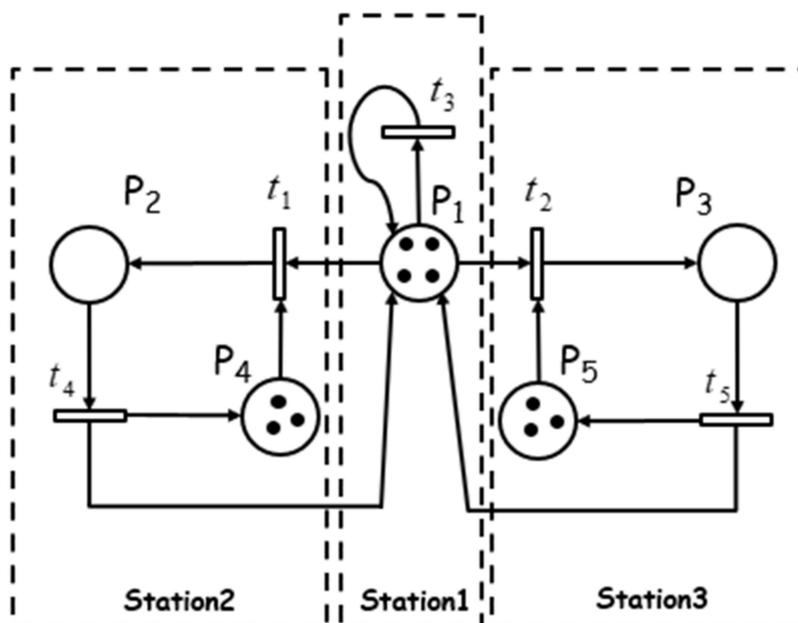


Figure 6. The PN modeling.

In a particular context of changing the work rate, P2 models our self-production station, while P3 represents a subcontractor. The capacity of each station is three pieces (three tokens in P4 and P5 each). P1 is the first station of our FMS and models stock production.

The two transitions ( $t_1$  and  $t_2$ ) represent the events of pallet input to the assembly stations, while  $t_4$  and  $t_5$  model the output events through their annex conveyors. Otherwise, the product or the pallet can continue its way to the main conveyor by  $t_3$ .

In an optimization context, our station 2 can at most work twice the rate compared to the subcontractor (station 3). This constraint can be expressed using the GMEC [23]:

$$M(P_2) + 2M(P_3) \leq 3$$

By using CPN Tools software, one can generate the state space. This reachability graph contains 13 markings (i.e., Figure 7).

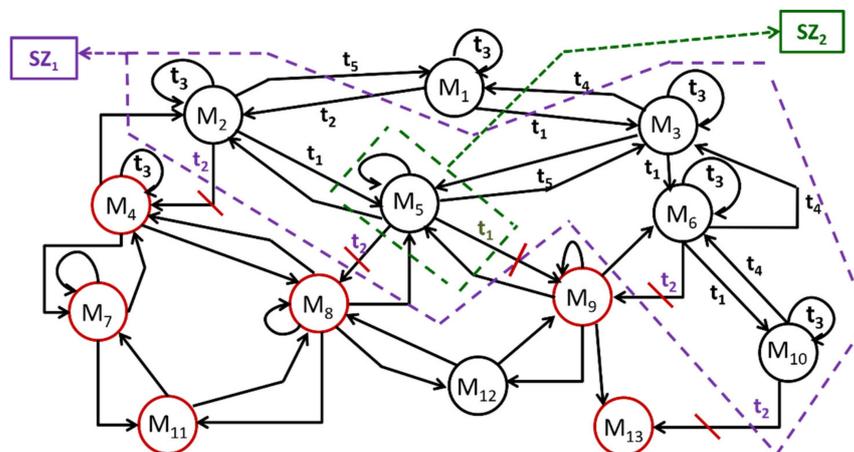


Figure 7. The reachability graph/the state space.

The application of our algorithm can allow us to determine the specific zones (**SZs**) with a minimal number of markings. In each  $SZ_i$ , the theory of regions is applied to calculate the corresponding  $P_{ci}$  offline.

However, the activation and deactivation of PN supervisors via the Boolean function  $FB_c$  is performed online.

In our example, and according to Definition 2, the set of event separation instances contains five MTSIs correlated by  $t_2$  and  $t_1$ :

$$\Omega_1 = \{(M_2[t_2 > M_4]); (M_6[t_2 > M_9]); (M_5[t_2 > M_8]); (M_{10}[t_2 > M_{13}])\}$$

$$\Omega_2 = \{(M_5[t_1 > M_9])\}$$

Then, by applying the first six steps of our algorithm, one can identify the following specific zones:

$$SZ_1 = \{(M_2[t_2 >]); (M_2[t_1 >]); (M_3[t_1 >]); (M_5[t_5 >]); (M_6[t_2 >]); (M_5[t_2 >]); (M_{10}[t_2 >])\}$$

$$SZ_2\{(M_5[t_1 >])\}$$

Therefore, the detailed information of **SZ** markings is listed in Table 1.

**Table 1.** Information of **SZ** markings.

<b>SZ<sub>i</sub></b>	<b>Marking No.</b>	<b>Categorization</b>	<b>Marking (P<sub>1</sub>, P<sub>2</sub>, P<sub>3</sub>)</b>
<b>SZ<sub>1</sub></b>	M2	MS1	(3,0,1)
	M4	MI1	(2,0,2)
	M5	MS1	(2,1,1)
	M8	MI2	(1,1,2)
	M6	MS1	(2,2,0)
	M9	MI3	(1,2,1)
	M10	MS1	(1,3,0)
	M13	MI4	(0,3,1)
	M3	MI5	(3,1,0)
	<b>SZ<sub>2</sub></b>	M5	MS2
M9		MI2	(1,2,1)

Owing to space limitations, not all equations can be listed. Only equations of the algorithm application are considered:

$$M_{01} = M_2 \text{ and } M_{02} = M_5 \rightarrow \text{The initial states of } SZ_1 \text{ and } SZ_2$$

By applying the algorithm of our methodology, one can obtain two Petri net supervisors:  $P_{c1}$  and  $P_{c2}$  when Equations (1)–(3) for each  $SZ_i$  are solved.

$$P_{c1} : M_2(P_{c1}) = M_{01} = 1; C(P_{c1}, \cdot) = (-1, -2, 0, 1, 2)$$

$$P_{c2} : M_5(P_{c2}) = M_{02} = 1; C(P_{c2}, \cdot) = (-2, 0, 0, 0, 0)$$

$$\text{The frontier states : } MF_1 = (M_2, M_3); MF_2 = (M_5)$$

One can deduce that it is only in the marking  $M_5$  that the PN controllers are involved simultaneously. Otherwise, the PN supervisors act alternatively:

$$\Leftrightarrow FB_c = P_{c1} + P_{c2}$$

To switch between  $P_{c1}$  and  $P_{c2}$ , detailed information about deactivation and activation is presented in Table 2.

**Table 2.** Information about activation/deactivation vectors.

<i>PN Supervisor</i>	<b>Marking No.</b>	<b>Deactivation Vector</b>	<b>Activation Vector</b>
$P_{c1}$	$M2$	[0,1,0,0,1]	[0,1,0,0,0]
	$M3$	[1,0,0,1,0]	[1,0,0,0,0]
$P_{c2}$	$M5$	[0,0,0,1,1]	[1,1,0,0,0]

## 5. PN Comparisons

In this part, a comparison of our new control synthesis method with the work of Ghaffari et al. [7] and the work of Huang et al. [19] is given using three examples taken from the literature.

For convenience, abbreviations are used to name the three approaches. Our new approach is called RS1. The method of Ghaffari et al. is called Gh2, and the method of Huang et al. is noted as Hu3.

All Petri net supervisors presented in these examples were calculated using CPLEX.

### 5.1. Comparison with FMS Example

Table 3 presents the experimental results of calculating the Petri net supervisors in comparison with the work of Gh2. By observing the results obtained in Table 3, one can infer that for the same number of legal markings and MTSIs, one can determine the same number of PN controllers with the reduced number of equations (N) of the theory of regions. In Gh2, 40 equations were solved to calculate two control places in 693 ms. The reduction technology was involved in RS1 by decreasing the calculation time and the number of equations (N) to solve. The computation time was divided by at least two (319 ms). The obtained controlled system is presented by Table 4.

**Table 3.** Table of comparisons with FMS example.

<i>Methodology</i>	<b>Number of Supervisors</b>	<b>N</b>	<b>Calculation Time (ms)</b>
RS1	2	15	319
Gh2	2	40	693

**Table 4.** The controlled system with FMS example.

<i>Methodology</i>	<b>Control Place</b>	$M_{0i}(P_{ci})$	$C(P_{ci}, .)$
RS1	$P_{c1}$	1	(−1,−2,0,1,2)
	$P_{c2}$	1	(−2,0,0,0,0)
Gh2	$P_{c1}$	1	(0,−1,0,0,1)
	$P_{c2}$	1	(−1,−2,0,1,2)

### 5.2. Comparison with Hu3 Example

The second example is taken from the work of Huang et al. [19]. It deals with the deadlock prevention policy. Table 5 presents the experimental results of calculating the Petri net supervisors in comparison with the work of Hu3. By obtaining the results presented in Table 5, one can deduce that four event separation instances and 46 equations are needed if Hu3 is used. Our method RS1 requires 10 event separation instances and 22 equations to solve. The Hu3 method needs three control places and three PN controllers if the RS1 approach is used.

**Table 5.** Table of comparisons with Hu3 example.

<i>Methodology</i>	<b>Number of Supervisors</b>	<b>N</b>	<b>Calculation Time (ms)</b>
RS1	3	22	334
Gh2	3	46	589

The main advantage of the RS1 method is the gain of calculation time, which was reduced from 589 to 334 ms. Then, for the same number of control places and legal markings, 22 equations (N) needed to be solved with our method, as opposed to 46 equations for the Hu3 approach. The controlled system of this example is given in Table 6.

**Table 6.** The controlled system with Hu3 example.

Methodology	Control Place	$M_{0i}(Pci)$	$C(Pci, .)$
RS1	Pc1	0	(−2,0,0,0,0,0,0,0,0)
	Pc2	0	(0,−2,0,0,0,0,0,0,0)
	Pc3	0	(0,0,0,0,0,−2,0,0,0)
Gh2	Pc1	2	(0,0,0,0,1,−1,0,1,−1,0)
	Pc2	3	(0,0,0,1,−1,0,0,1,−1,0)
	Pc3	2	(0,0,0,1,−1,0,0,1,−1)

## 6. Discussion

Tables 1–3 show the comparisons of our methodology with previous work according to criteria related to the calculation burden of Petri net controllers.

In the first example, one can compare our new supervisory control approach with the classical method of the theory of regions developed by Ghaffari et al. [7] using example 1 of flexible manufacturing systems. The RS1 method is confronted with a deadlock prevention approach determined by Huang et al. in example 2 taken from [19], which has many advantages and resolves problems of forbidden state transitions. The experiment results of the Hu3 approach imply that its control policy is more powerful than existing deadlock prevention policies such as those presented in the works of Li et al. [24], Park et al. [25], and Uzam et al. [18].

Based on comparisons, one in all cited examples, the computational load of the PN supervisors was almost divided by two. The computation time using CPLEX for all the methods was reduced by applying our control approach.

Moreover, the number of equations to solve was reduced using our methodology. As a result, our new approach translated by RS1 can drastically decrease the challenges encountered in supervisory control by using a new interpretation of the theory of regions.

## 7. Conclusions and Perspectives

Finally, the specific use of symmetry concepts, between mathematics and Petri nets in supervisory control, was a major asset to us in mapping the mathematical core of such a practical problem. This finally allowed us to solve our problem and control our FMS, as stated from the start of the project.

The proposed control method based on the SZ method using the theory of regions was tested in a flexible manufacturing system implemented in our laboratory. Due to our previous research, the calculated Petri net controllers can be implemented on programmable logic controllers such as Siemens and Schneider. The results obtained either by programming or by experiments comparisons with the methods (Hu3 and Gh2) clearly show that our method is very efficient.

For the same number of Petri net controllers, one can calculate the PN monitors using few equations and short calculation times. The application of the theory of regions is restricted to specific zones of the graph and not on all states.

For this reason, our supervisors were first calculated offline, then activated and deactivated according to the situation or the state in which the system was located via the defined Boolean function. This step was then performed online.

Based on experiment results, it is obvious to remark on the considerable reduction in the computational cost of the supervisory control using our new approach to synthesize a maximally permissive Petri net controller.

In the near future, we are interested in developing other methods to solve problems that involve a combinatorial explosion of states by generating a reachability graph of a Petri net, which is an important issue in the supervisory control domain.

**Author Contributions:** In this work, the authors are specialists in the industrial engineering field and, more precisely, in model-based systems engineering and the supervision of DES. Conceptualization, S.R.; methodology, S.R.; software, S.R.; validation, N.R.; formal analysis, S.R.; investigation, S.R.; resources, N.R.; data curation, Z.H.; writing—original draft preparation, S.R.; writing—review and editing, S.R.; visualization, Z.H.; supervision, N.R.; project administration, N.R.; funding acquisition, N.R. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Acknowledgments:** We used our equipment in the laboratory with all the appropriate accessories and software, such as Siemens STEP7 and SCHNEIDER Control Expert control software; the simulation and modeling software CPN Tools; and supervision software such as WINCC FLEXIBLE.

**Conflicts of Interest:** The authors Sadok Rezig, Nidhal Rezg, and Zied Hajej declare no conflict of interest.

## References

- Murata, T.; Subrahmanian, V.; Wakayama, T. A Petri net model for reasoning in the presence of inconsistency. *IEEE Trans. Knowl. Data Eng.* **1991**, *3*, 281–292. [[CrossRef](#)]
- Murata, T. Petri nets: Properties, analysis and applications. *Proc. IEEE* **1989**, *77*, 541–580. [[CrossRef](#)]
- Chen, T.; Sanchez-Aarnoutse, J.C.; Buford, J. Petri Net Modeling of Cyber-Physical Attacks on Smart Grid. *IEEE Trans. Smart Grid* **2011**, *2*, 741–749. [[CrossRef](#)]
- Ramadge, P.J.; Wonham, W.M. The control of discrete event systems. *Proc. IEEE* **1989**, *77*, 81–98. [[CrossRef](#)]
- Wonham, W.M.; Ramadge, P.J. On the Supremal Controllable Sublanguage of a Given Language. *SIAM J. Control. Optim.* **1987**, *25*, 637–659. [[CrossRef](#)]
- Ghaffari, A.; Rezg, N.; Xie, X. Design of a live and maximally permissive petri net controller using the theory of regions. *IEEE Trans. Robot. Autom.* **2003**, *19*, 137–141. [[CrossRef](#)]
- Ghaffari, A.; Rezg, N.; Xie, X. Feedback control logic for forbidden-state problems of marked graphs: Application to a real manufacturing system. *IEEE Trans. Autom. Control* **2003**, *48*, 18–29. [[CrossRef](#)]
- Ghaffari, A.; Rezg, N.; Xie, X. Live and Maximally Permissive Controller Synthesis Using Theory of Regions. In *Synthesis and Control of Discrete Event Systems*; Springer: Singapore, 2002; pp. 155–166.
- Ghaffari, A.; Rezg, N.; Xie, X. Algebraic and geometric characterization of Petri net controllers using the theory of regions. In *Proceedings of the Sixth International Workshop on Discrete Event Systems, Zaragoza, Spain, 4 October 2002*; pp. 219–224.
- Bashir, M.; Zhou, J.; Muhammad, B.B. Optimal Supervisory Control for Flexible Manufacturing Systems Model with Petri Nets: A Place-Transition Control. *IEEE Access* **2021**, *9*, 58566–58578. [[CrossRef](#)]
- Der Jeng, M. A Petri net synthesis theory for modeling flexible manufacturing systems. *IEEE Trans. Syst. Man Cybern. Part B* **1997**, *27*, 169–183. [[CrossRef](#)] [[PubMed](#)]
- Xing, K.-Y.; Hu, B.-S.; Chen, H.-X. Deadlock avoidance policy for Petri-net modeling of flexible manufacturing systems with shared resources. *IEEE Trans. Autom. Control* **1996**, *41*, 289–295. [[CrossRef](#)]
- Ji, Y.; Yin, X.; Lafortune, S. Optimal supervisory control with mean payoff objectives and under partial observation. *Automatica* **2021**, *123*, 109359. [[CrossRef](#)]
- Uzam, M.; Wonham, W. A hybrid approach to supervisory control of discrete event systems coupling RW supervisors to Petri nets. *Int. J. Adv. Manuf. Technol.* **2006**, *28*, 747–760. [[CrossRef](#)]
- Zhao, M.; Uzam, M.; Hou, Y. Near-optimal supervisory control of flexible manufacturing systems using divide-and-conquer iterative method. *Adv. Mech. Eng.* **2016**, *8*, 1687814016639823. [[CrossRef](#)]
- Abubakar, U.S.; Liu, G.; Uzam, M. Petri Net-Based Robust Supervisory Control of Automated Manufacturing Systems with Multiple Unreliable Resources. *IEEE Access* **2021**, *9*, 100264–100278. [[CrossRef](#)]
- Chen, Y.; Li, Z.; Zhou, M. Optimal Supervisory Control of Flexible Manufacturing Systems by Petri Nets: A Set Classification Approach. *IEEE Trans. Autom. Sci. Eng.* **2013**, *11*, 549–563. [[CrossRef](#)]
- Uzam, M.; Zhou, M. An Iterative Synthesis Approach to Petri Net-Based Deadlock Prevention Policy for Flexible Manufacturing Systems. *IEEE Trans. Syst. Man Cybern. Part A Syst. Humans* **2007**, *37*, 362–371. [[CrossRef](#)]
- Huang, Y.-S.; Pan, Y.-L. An improved maximally permissive deadlock prevention policy based on the theory of regions and reduction approach. *IET Control Theory Appl.* **2011**, *5*, 1069–1078. [[CrossRef](#)]

20. Rezig, S. *Approches Canoniques Pour la Synthèse des Contrôleurs Réseaux de Petri*. Ph.D. Thesis, Université de Lorraine, Lorraine, French, 2016.
21. Rezig, S.; Ezzeddine, W.; Turki, S.; Rezg, N. Mathematical Model for Production Plan Optimization—A Case Study of Discrete Event Systems. *Mathematics* **2020**, *8*, 955. [[CrossRef](#)]
22. Darondeau, P.; Xie, X. Linear control of live marked graphs. *Automatica* **2003**, *39*, 429–440. [[CrossRef](#)]
23. Giua, A.; DiCesare, F.; Silva, M. Generalized mutual exclusion constraints on nets with uncontrollable transitions. In Proceedings of the 1992 IEEE International Conference on Systems, Man, and Cybernetics, Chicago, IL, USA, 18–21 October 1992; pp. 974–979.
24. Li, Z.; Zhou, M. *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2009.
25. Park, E.; Tilbury, D.M.; Khargonekar, P.P. A modeling and analysis methodology for modular logic controllers of machining systems using Petri net formalism. *IEEE Trans. Syst. Man Cybern. Part C* **2011**, *31*, 168–188. [[CrossRef](#)]