

Article

Global Supervisory Structure for Decentralized Systems of Flexible Manufacturing Systems Using Petri Nets

Muhammad Bashir * and Liang Hong

College of Electronics and Information, Xi'an Polytechnic University, Xi'an 710048, China

* Correspondence: m.bashir71@yahoo.com; Tel.: +86-150-9132-7525

Received: 27 July 2019; Accepted: 26 August 2019; Published: 4 September 2019



Abstract: Decentralized supervisory structure has drawn much attention in recent years to address the computational complexity in designing supervisory structures for large Petri net model. Many studies are reported in the paradigm of automata while few can be found in the Petri net paradigm. The decentralized supervisory structure can address the computational complexity, but it adds the structural complexity of supervisory structure. This paper proposed a new method of designing a global controller for decentralized systems of a large Petri net model for flexible manufacturing systems. The proposed method can both reduce the computational complexity by decomposition of large Petri net models into several subnets and structural complexity by designing a global supervisory structure that can greatly reduce the cost at the implementation stage. Two efficient algorithms are developed in the proposed method. Algorithm 1 is used to compute decentralized working zones from the given Petri net model for flexible manufacturing systems. Algorithm 2 is used to compute the global controller that enforces the liveness to the decentralized working zones. The ring assembling method is used to reconnect and controlled the working zones via a global controller. The proposed method can be applied to large Petri nets size and, in general, it has less computational and structural complexity. Experimental examples are presented to explore the applicability of the proposed method.

Keywords: decentralized system; working zone; flexible manufacturing system; Petri nets; global controller

1. Introduction

Due to the complexity of flexible manufacturing systems in the recent years, decentralized systems control has drawn much attention worldwide by the researchers in the area of the discrete event systems. Large-scale discrete event systems (DES) are modeled through the composition of many smaller subsystems representing shared resources by the concurrent operations. The controlled of large-size flexible manufacturing systems is tedious and time-consuming, perhaps it required huge memory space due to a large number of its state space. Moreover, the analysis of complex systems is very complicated and sometimes remains non-deterministic polynomial time (NP-hard). To reduce the computational complexity and required memory space for the analysis and control of large flexible manufacturing systems, we decomposed the systems into several working zones (i.e., several sub-problems).

Several methods are developed in the literature to analyze the decentralized systems of DES-based on the system applications. Most of the methods focus on the design of a decentralized controller (i.e., modular controller) to supervise the operations of the DES. The DES find application in the areas of automation [1,2], software development [3], computer engineering [4,5], communication

systems [6], flexible manufacturing system [7,8] and transportation networks [9]. The main challenges of the study of DES are the occurrence of deadlocks. Deadlocks can occur due to excessive use of shared resources in the systems, which degrade the system performance of DES [10–17]. Three main tools exist to design a decentralized supervisory structure for DES, namely, they are; graph theory [2,5,15,16], automata [18–22] and Petri nets [14,16,23–27]. This paper focuses on the design of a global supervisory structure for decentralized systems using the Petri nets paradigm.

In a Petri net paradigm, forbidden states are often specified by linear constraints. The concept of generalized mutual exclusion constraints (GMECs) is often used to analyze and construct a supervisory structure that enforces the liveness of the DES [25,27,28]. The emergence of large Petri nets model for flexible manufacturing systems motivates the research to decompose the systems in a systematic method. The work of Li in [29] developed a divide and conquer method to control the deadlocks. The developed method is used to decompose the Petri net model into smaller subnets, in such a way that the decomposition is done via resource transition circuits. Control places are designed using siphon techniques for the decomposed subnets. This adds the structural complexity to the size of the supervisory structure. Uzam et al. modified the divide and conquer method in [30] via an iterative decomposition of the Petri net model. At each iteration, control places are designed for the decomposed subnets using reachability techniques. The method suffers from the computational complexity since it involved the computation of the reachability graph at each iteration.

A modular supervisory scheme for modeling and synthesis of supervisory agents using a Petri net model is developed in [4]. These modular supervisors are conceived as operating independently, each exercising control to satisfy its own specification. However, when all the modular supervisors operate concurrently, they may result in conflict. To reduce the computational complexity for the design of the supervisory structure, the work presented in [31] proposed two methods for the decentralized supervisory structure. The first method considers the design of decentralized supervisory structure using communication while assuming that all the transitions in the Petri net model are observable and controllable. The second method considers the transformation of constraints that is not d-admissible in the design of the decentralized supervisory structure with no communications. The solution is obtained by solving integer linear programming (ILPP). The first method allows more reachable states than the second method due to the advantage of unrestricted communications. Generating and solving integer linear programming problem (ILPP) is computationally complex [32–35] and can limit the application of the developed methods.

It is generally known that computation of a reachability graph is computationally complex and sometimes NP-hard. To avoid the computation of the reachability graph, the authors in [17] developed a new method of decentralized supervisory control for communication systems using a Petri nets paradigm. The communication among the decentralized subnets are enforced in a centralized environment. Ye et al. in [28] extended the works to flexible manufacturing systems that emphasized the efficiency of the decomposition of Petri nets for flexible manufacturing systems. The developed method designs a decentralized supervisory structure without transforming the inadmissible constraints to admissible constraints. The communication constraints are incorporated between the decentralized subnets. The decentralized supervisory structure is usually associated with structural complexity, which require huge costs in the implementation stage. The computational and structural complexity is addressed in the proposed paper. The former is achieved by decomposition of Petri nets into a several subnets model while the latter is addressed using a global supervisory structure.

The study of [36] proposed a decomposition of a Petri net model through the use of local invariants (place and transition invariants) computation since it is well-known that the invariants can help to prove structural properties of Petri nets, i.e., boundedness, reversibility, liveness, and safeness. This proposed paper utilized the use of P-invariants in the decomposition of a large Petri net model and the operation for each decentralized subnet can be monitored using the global controller unlike the work of Bourjij et al. in [36] that the global solution can only be found by the concatenation of the subnets that has the same coupling power.

In this paper, we proposed a new method to design a global supervisory structure of the decentralized systems using Petri nets for flexible manufacturing systems. First, a subnet is computed from the given Petri net model for a flexible manufacturing system. Second, the subnet model is split into several decentralized working zones. The main contributions of the proposed paper are summarized as follows: (i) two efficient algorithms are developed, the first algorithm is used to compute the decentralized working zones from the subnet model, while the second algorithm is used to compute the global controller; (ii) the proposed method can reduce the computational and structural complexity through the use of global supervisory structure and (iii) the proposed method can be applied to a complex system for the computation of the supervisory structure.

The ring assembling is used to reconnect and control the working zones via a global controller. The global controller is computed and added to the decentralized working zones. When all the decentralized working zones are reconnected via a global controller, the resulting Petri net model is live maximally permissive behavior. Several experimental examples are used to explore the applicability of the proposed method.

The remainder of this paper is organized as follows. Formal descriptions of Petri nets and notations used in this paper are presented in Section 2. Section 3 provides the detail ideas for the computation of decentralized working zones. Design of a global controller is presented in Section 4. Experimental examples are provided in Section 5. Discussions are presented in Section 6. Finally, Section 7 concludes this paper.

2. Preliminaries

A Petri net is a four-tuple $N = (P, T, F, W)$, where P and T are finite and non-empty sets. P is a set of places and T is a set of transitions with $P \cap T = \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called a flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. Places are graphically represented by circles while transitions by bars or square boxes. $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(x, y) > 0$ if $(x, y) \in F$, and $W(x, y) = 0$, otherwise, where $(x, y) \in (P \times T) \cup (T \times P)$ and \mathbb{N} is the set of non-negative integers. $N = (P, T, F, W)$ is said to be *ordinary*, denoted as $N = (P, T, F)$, if $\forall f \in F, W(f) = 1$. Let $x \in P \cup T$ be a node in $N = (P, T, F, W)$.

A marking of a Petri nets N describes the current state of the system. $M(p)$ denotes the number of tokens in place p . The presence of token in the Petri net structure makes the system to be dynamic. It is essential to track the places that are marked at each reachable marking. A place p is said to be marked at a marking M if $M(p) > 0$. Usually, we express the markings and vectors as a formal sum notation or multiset for easy representation i.e., $\sum_{p \in P} M(p)p$. Let us consider a simple example, assuming we have a net system with places p_1 to p_7 formally represented as $P = \{p_1, \dots, p_7\}$. If we have a marking that has one token in place p_3 and two tokens in places p_2 and p_5 , respectively, using multiset notation, the marking can be written as $2p_2 + p_3 + 2p_5$ instead of $(0 \ 2 \ 1 \ 0 \ 2 \ 0 \ 0)$. In summary, a marking (M) of (N, M_0) is a mapping from P to \mathbb{N} .

Let us assume $t \in T$ to be a transition in the Petri nets N . At any marking M , transition t is enabled if $\forall p \in \bullet t, M(p) \geq W(p, t)$. For easier representation and without any ambiguity, it can be denoted as $M[t)$. If the enable transition is fired, it leads the system into new reachable marking M' . Formally, the tokens at each place in the new reachable marking can be tract as $\forall p \in P, M'(p) = M(p) - W(p, t) + W(t, p)$ and this can be denoted as $M[t)M'$ in short.

Suppose that x is a node in the Petri net model i.e., $x \in P \cup T$. The input (preset) of x is expressed as $\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$ and its output (postset) is expressed as $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$. The Petri net structure can be represented in matrix form and is called an incidence matrix. An incidence matrix is obtained from the input and output matrices. The input incidence matrix is defined as $[D^-] = W(p, t)$ and the output incidence matrix is defined as $[D^+] = W(t, p)$. Therefore, the incidence matrix of the net is expressed as $[D] = [D^+] - [D^-]$. The incidence vector for a place p (transition t) in the row (column) incidence matrix can be represented as $[D](p, \cdot)$ ($[D](\cdot, t)$).

A Petri net model (N, M_0) is said to be in a deadlock state when none of its transitions can be enabled at a marking M , i.e., $\nexists t \in T, M[t]$ holds. A Petri net model is deadlock free when some of the reachable states can not be observed. In other words, a deadlock free is called live-lock and can be expressed as $\forall M \in R(N, M_0), \exists t \in T, M_0[t]$. (N, M_0) is said to be live if all its reachable states can return to its initial marking M_0 . Mathematically, it can be expressed as (N, M_0) being live if $\forall M \in R(N, M_0), \forall t \in T, \exists M' \in R(N, M), M'[t]M_0$. Let M and M' be two markings in $R(N, M_0)$. M A-covers M' if $\forall p \in P_A, M(p) \geq M'(p)$, which is denoted as $M(p) \geq_A M'(p)$.

Let σ be a transition sequence. The Parikh vector of σ , denoted by $\vec{\sigma}$, is a column vector, represented by $\vec{\sigma} = [\#\sigma(t_1), \#\sigma(t_2), \dots, \#\sigma(t_{|T|})]^T$, where $\#\sigma(t_i)$ denotes the number of occurrences of t_i in σ . σ^* denote the set of all firing sequence in (N, M_0) . The notation $\xi(\sigma^*)$ represents the vector firing sequence σ^* in (N, M_0) mapping from (N, M_0) to its decentralized subnet. A P-vector is a column vector $I : P \rightarrow \mathbb{Z}$, indexed by P , where $\mathbb{Z} = \{\dots, -2, -1, 0, 1, 2, \dots\}$. A P-vector I is a place invariant if $I \neq \mathbf{0}$ and $I^T[D] = \mathbf{0}^T$. A T-vector is a column vector $H : T \rightarrow \mathbb{Z}$ indexed by T . A T-vector H is a transition invariant if $[D]H = \mathbf{0}$ and $H \neq \mathbf{0}$. The support of a place (transition) invariant $I(H)$ is denoted by $\|I\| = \{p | I(p) \neq 0\}$ ($\|H\| = \{t | H(t) \neq 0\}$). Let \mathbb{N}^+ denote the set of positive integers.

3. Petri Net Model with S³PR

Systems of simple sequential processes with resources (S³PR) are extensively used and adopted in the analysis of FMSs [27,37,38]. This paper considers an S³PR class of Petri nets for flexible manufacturing systems.

Definition 1. A system of simple sequential processes with resources (S³PR) is a Petri net $N = (P, T, F)$, satisfying [37]:

1. $P = P_A \cup P_R \cup P^0$ is a partition of places with $P_A \cap P_R = \emptyset, P_A \cap P^0 = \emptyset, \text{ and } P_R \cap P^0 = \emptyset$;
 - (a) p_i^0 is called the process idle place of N_i . Elements in P_A^i and P_R^i are called operation and resource places;
 - (b) $P_A^i \neq \emptyset; P_R^i \neq \emptyset; p_i^0 \in P^0; (P_A^i \cup \{p_i^0\}) \cap P_R^i = \emptyset$;
 - (c) $\forall p \in P_A^i, \forall t \in \bullet p, \forall t' \in p \bullet, \exists r_p \in P_R^i, \bullet t \cap P_R^i = t' \bullet \cap P_R^i = \{r_p\}$;
 - (d) $\forall r \in P_R^i, \bullet \bullet r \cap P_A^i = r \bullet \bullet \cap P_A^i \neq \emptyset, \bullet r \cap r \bullet = \emptyset$;
 - (e) $\bullet \bullet (p_i^0) \cap P_R^i = (p_i^0) \bullet \bullet \cap P_R^i = \emptyset$;
2. $T = \bigcup_{i=1}^m T_i$, with $T_i \cap T_j = \emptyset$ for all $i \neq j$, is a set of transitions;
3. N_i is a strongly connected state machine, where $N_i = (P_A^i \cup \{p_i^0\}, T_i, F_i)$ is the resulting net after the places in P_R^i and related arcs are removed from N_i ;
4. Any two N_i are composable when they share a set of common places. Every shared place must be a resource place.
5. For $p \in P_A, \bullet \bullet (p) \cap P_R = \{r_p\}$ where resource place r_p is called the resource used by p .

Definition 2. Let (N, M_0) be the Petri net model with $N = (P, T, F)$, where $P = P_A \cup P_R \cup P^0$. (N', M_0) is said to be a subnet of (N, M_0) with $N' = (P', T', F')$ if $P' = P \setminus P^0, T' = T, F' = F \setminus Q$. $Q \subseteq (P^0 \times T) \cup (T \times P^0)$ is called a flow relation of the net, represented by an arc with an arrow from idle places to transitions or transitions to idle places.

Definition 3. Resource r is said to be shared if $\exists p, p' \in (\|I_r\| \setminus \{r\}), p \in P_{A_i}, p' \in P_{A_j}, i \neq j$. Let the shared resource place denoted by r in (N, M_0) .

Definition 4. Let $t \in T$ be the transition in (N', M_0) . Transition t is said to be source transition if $|t^\bullet| = 1$ and is said to be sink transition if $|\bullet t| = 1$. Let t^s denote the source transition and t^c denote the sink transition. The sets of source and sink transition are denoted as $\Theta = \{t | t \in T, |t^\bullet| = 1\}$ and $\Psi = \{t | t \in T, |\bullet t| = 1\}$, respectively.

Property 1. Every concurrent process has at least one source and sink transitions in (N, M_0) .

3.1. Decentralization of FMS into Working Zones

The analysis of a complex Petri net model for a flexible manufacturing system is hard and time-consuming if it's not NP-hard. The complex system can be addressed by dividing it into several decentralized problems (subnet).

Definition 5. A subnet of a Petri net model is called a working zone denoted by $Z = (P^q, T^q, F^q)$ if it satisfies the following:

- $P^q = \|I_r\|, P^q = P_A^q \cup P_R^q$ is a partition of places with $P_A^q \cap P_R^q = \emptyset$. P_A^q and P_R^q is the set of activity and resource places, respectively;
- $T^q = \{t \mid t \in T', \bullet\|I_r\| \cup \|I_r\|\bullet\}$ is the set of transition associated with the minimal P-semiflow;
- $\forall t \in T', |\bullet t| = |t\bullet| = 1$ is a state machine;
- $F^q \rightarrow (P^q \times T^q) \cup (T^q \times P^q)$ is a flow relation simply represented by directed arcs;
- $\exists r \in P_R^q$, then $M(r) \geq 1$;
- Z is strongly bounded.

Suppose that we have $N' = (P', T', F')$ as a subnet from (N, M_0) with $N = (P, T, F)$. N' consists of a set of a decentralized working zone $\mathfrak{S}_1 = (P_1^q, T_1^q, F_1^q), \mathfrak{S}_2 = (P_2^q, T_2^q, F_2^q), \dots$, and $\mathfrak{S}_\ell = (P_\ell^q, T_\ell^q, F_\ell^q)$ such that $P' = P_1^q \cup P_2^q \cup \dots \cup P_\ell^q, T' = T_1^q \cup T_2^q \cup \dots \cup T_\ell^q$, and $F' = F_1^q \cup F_2^q \cup \dots \cup F_\ell^q$. A transition $t \in T'$ is a common transition if t belongs to \mathfrak{S}_i and \mathfrak{S}_j in $N', i, j \in \{1, 2, \dots, \ell\}$ and $i \neq j$. Let $T_d \subseteq T'$ denote a set of common transitions from N' . With no ambiguity and easy representation, $\forall t \in T_d, t$ can be rewritten as t' belongs to \mathfrak{S}_i and t'' belongs to \mathfrak{S}_j , where t', t'' s.t. $[t]$ condition. Suppose that $\mathfrak{S}_z = (P_z^q, T_z^q, F_z^q)$ is a decentralized working zone from N' , where $z \in \{1, \dots, \ell\}$ and ℓ is the number of working zones. D_z is the incidence matrix of \mathfrak{S}_z and each working zone can be represented as:

$$\forall t_j \in T_z^q, \left| \sum_{1 \leq i \leq n, 1 \leq j \leq m} D_z(p_i, t_j) \right| \leq 1. \tag{1}$$

The places of each decentralized working zone form the support of P-invariants and can be represented as a multiset

$$\|I\| = \sum_{i=1}^n p_i \geq 1, \tag{2}$$

where $n = |P_z^q|$ and $m = |T_z^q|$. Let $t_i \in T_z^q, i \in \{1, \dots, m\}$ be the transitions associated with the places in the P-invariants. All places in the support of P-invariants are connected to its corresponding transitions as

$$\forall i \in \{1, \dots, n\}, \forall j \in \{1, \dots, m\}, \exists p_i \in P_z^q, \exists t_j \in T_z^q, (p_i \times t_j) \cup (t_j \times p_i). \tag{3}$$

Let D' be the incidence matrix of the subnet model, and I be the P-invariant in the decentralized working zone. Each decentralized working zone should satisfy Equation (4),

$$\bullet\|I\| = \|I\|\bullet, I^T[D'] = 0^T. \tag{4}$$

Every working zone should contain at least one token after the decentralization, which implies that there exists at least one resource place in each decentralized working zone being marked at the initial marking M_0 , i.e., for every $\mathfrak{S}_z, z \in \{1, \dots, \ell\}$

$$\exists r^s \in P_z^q, M(r^s) \neq 0. \tag{5}$$

Theorem 1. Let $N' = \{P', T', F'\}$ be a subnet of (N, M_0) . N' can be decentralized into minimal working zones via Algorithm 1.

Algorithm 1 Decentralization of an FMS into working zones

Require: Petri net model of an FMS structure suffering from deadlock

Ensure: Decentralized working zones

- 1: Compute the subnet model (N', M_0) from (N, M_0) using Definition 2
 - 2: Compute the minimal P-invariant I_i in (N', M_0)
 - 3: Compute the support of minimal P-invariant $\|I_i\|$ from (N', M_0)
 - 4: $i \in \{1, 2, \dots, n\}$, /* n is the number of $\|I_i\|$ in (N', M_0) */
 - 5: **while** $i \leq n$ **do**
 - 6: **if** ($\|I_i\|$ satisfied equ. 1–5) **then**
 - 7: Compute T_d^q from N' using Definition 5
 - 8: Compute $T_d^i = T_d^q \cap T'$
 - 9: Remove $Z_i = (P_i^q, T_d^q, F_i^q)$ from N' , $P_i^q \subseteq \|I_i\|$
 - 10: $P' = P' \setminus \|I_i\|$
 - 11: $T' \cup T_d^i = T' \setminus T_d^q$
 - 12: $F' = F' \setminus F_i^q$
 - /* T_d is the set of common transition between the two successive Z_i */
 - 13: **endif**
 - 14: $N' = (P', T', F')$
 - 15: $i = i + 1$
 - 16: **end**
 - 17: Output the decentralized working zones
-

Proof. According to Definition 5, if $P' \setminus P_A' \neq \emptyset$, at least one share resource place (i.e., $\exists r \in P_R'$) therefore exists in (N', M_0) . It is obvious that any shared resource place for a Petri net model of an FMS forms a minimal P-semiflow $\|I_r\|$ such that $\{r\} = \|I_r\| \setminus P_A'$, $P_A' \in P'$. Therefore, each minimal P-semiflow can satisfy Equations (1)–(5) of Algorithm 1 and hence the theorem holds. \square

Theorem 2. Let (N', M_0) be a system with $N' = (P', T', F')$ and \mathfrak{S}_i be decentralized subnet from N' , $i \in \{1, \dots, \ell\}$, where ℓ is the number of decentralized working zones. Let σ^* and σ_i^* denote all the transition firing sequences in N' and \mathfrak{S}_i , respectively. $\zeta_i(\sigma^*) = \sigma_i^*$ if and only if for any two working zones \mathfrak{S}_i and \mathfrak{S}_j satisfied $T_i' \cap T_j' \neq \emptyset$, $\sigma_i^* |_{T_i' \cap T_j'} = \sigma_j^* |_{T_i' \cap T_j'}$, for $i, j \in \{1, \dots, \ell\}$ [28].

To minimize space and redundancy, we can refer the reader to the work of [28] for the proof of Theorem 2. According to Theorem 2, both the structural and dynamic properties of subnet with that of decentralized systems are preserved and equivalence for any transition firing sequence from decentralized working zones that can be mapped into a subnet model.

3.2. Demonstrated Example

Let us assume that we consider a flexible manufacturing system shown in Figure 1a that produced two different part types (goods). Machines M1, M2 and M3 can produce two different types of parts (P1 and P2) in parallel. A robot R1 can load (unload) parts P1 (P2) from M1 to M2 (M2 to M1), respectively. Similarly, the robot R2 can load part P2 from M2 to M1 and unload part P1 from M1 to M2. The machines M1, M2 and M3 can only process one part at a time. Similarly, the robots R1 and R2 can only process one part at a time. Parts P1 and P2 are considered in the production sequence for a flexible manufacturing system through input/output buffers I1/O1 and I2/O2. Initially, it is assumed that there are no parts in the system.

The layout of flexible manufacturing systems shown in Figure 1b can be equivalently represented using Petri nets as shown in Figure 2a. Places in the Petri nets shown in Figure 2a are partitioned as $P = P^0 \cup P_A \cup P_R$, where $P^0 = \{p_{16}, p_{17}\}$, $P_R = \{p_{11}, \dots, p_{15}\}$ and $P_A = \{p_1, \dots, p_{10}\}$. Places p_1, p_2, p_3, p_4 and p_5 represent the operations of M1, R1, M2, R2 and M3 for the production sequence of part P1, respectively. Similarly, for the production sequence part P2, places p_6, p_7, p_8, p_9 and p_{10} represent the operations of M3, R2, M2, R1 and M1, respectively. Places p_{16} and p_{17} represent the I1/O1 and I2/O2 buffers, respectively, while places $p_{11}, p_{12}, p_{13}, p_{14}$ and p_{15} represent the shared resources of M1, R1, M2, R2 and M3, respectively.

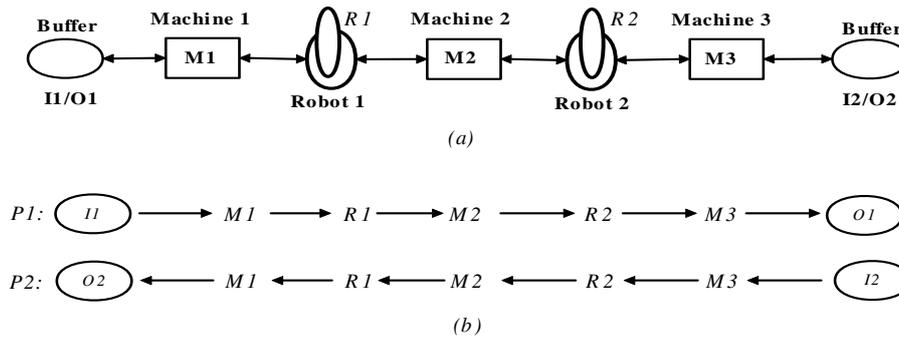


Figure 1. Flexible manufacturing layout.

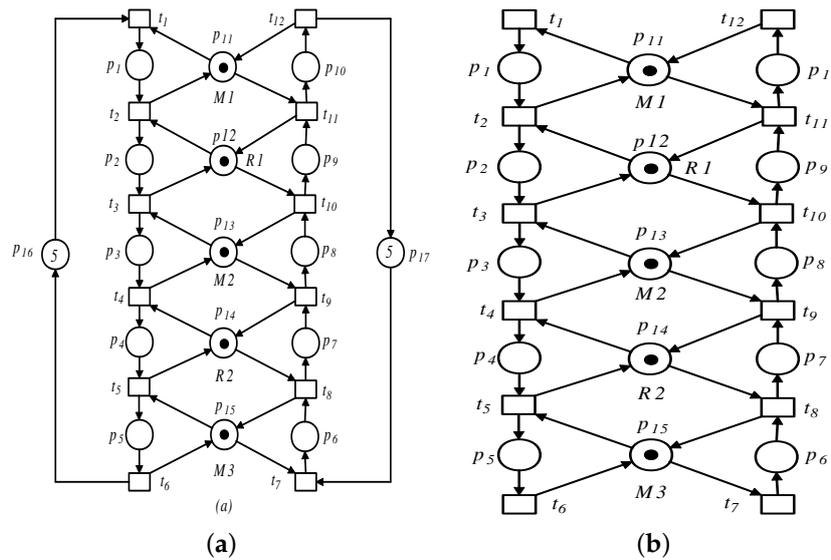


Figure 2. (a) Petri net model (N, M_0) ; (b) a subnet model $N = (N', M_0)$ with $N' = (P', T', F')$.

The subnet N' shown in Figure 2b has five minimal P -invariants: $I_1 = (100000000110000)^T$, $I_2 = (010000001001000)^T$, $I_3 = (001000010000100)^T$, $I_4 = (000100100000010)^T$, and $I_5 = (000011000000001)^T$. The support of minimal P -invariants are $\|I_1\| = \{p_1, p_{10}, p_{11}\}$, $\|I_2\| = \{p_2, p_9, p_{12}\}$, $\|I_3\| = \{p_3, p_8, p_{13}\}$, $\|I_4\| = \{p_4, p_7, p_{14}\}$ and $\|I_5\| = \{p_5, p_6, p_{15}\}$. The preset and posset of P -invariants are $\bullet\|I_1\| = \|I_1\|\bullet = \{t_1, t_2, t_{11}, t_{12}\}$, $\bullet\|I_2\| = \|I_2\|\bullet = \{t_2, t_3, t_{10}, t_{11}\}$, $\bullet\|I_3\| = \|I_3\|\bullet = \{t_3, t_4, t_9, t_{10}\}$, $\bullet\|I_4\| = \|I_4\|\bullet = \{t_4, t_5, t_8, t_9\}$ and $\bullet\|I_5\| = \|I_5\|\bullet = \{t_5, t_6, t_7, t_8\}$. The set of common transition between P -invariants are $T_d^1 = \|I_1\|\bullet \cap \|I_2\|\bullet = \{t_2, t_{11}\}$, $T_d^2 = \|I_2\|\bullet \cap \|I_3\|\bullet = \{t_3, t_{10}\}$, $T_d^3 = \|I_3\|\bullet \cap \|I_4\|\bullet = \{t_4, t_9\}$ and $T_d^4 = \|I_4\|\bullet \cap \|I_5\|\bullet = \{t_5, t_8\}$. The working zones are computed as shown in Figure 3, i.e., $\mathfrak{S}_1 = (P_1^q, T_1^q, F_1^q)$ with $P_1^q = \{p_1, p_{10}, p_{11}\}$ and $T_1^q = \{t_1, t_2, t_{11}, t_{12}\}$, $\mathfrak{S}_2 = (P_2^q, T_2^q, F_2^q)$ with $P_2^q = \{p_2, p_9, p_{12}\}$ and $T_2^q = \{t_2, t_3, t_{10}, t_{11}\}$, $\mathfrak{S}_3 = (P_3^q, T_3^q, F_3^q)$ with $P_3^q = \{p_3, p_8, p_{13}\}$ and $T_3^q = \{t_3, t_4, t_9, t_{10}\}$, $\mathfrak{S}_4 = (P_4^q, T_4^q, F_4^q)$ with $P_4^q = \{p_4, p_7, p_{14}\}$ and $T_4^q = \{t_4, t_5, t_8, t_9\}$ and $\mathfrak{S}_5 = (P_5^q, T_5^q, F_5^q)$ with $P_5^q = \{p_5, p_6, p_{15}\}$ and $T_5^q = \{t_5, t_6, t_7, t_8\}$.

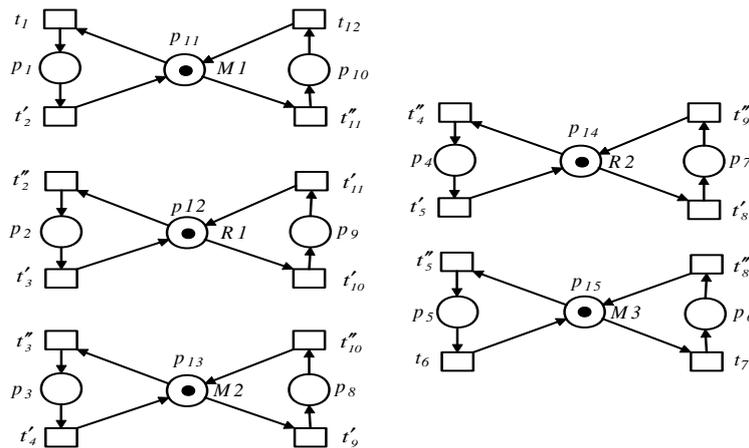


Figure 3. Decentralized working zones.

4. Design of a Global Controller

The global controller aims to reconnect and controlled the working zones of an FMS such that it can enforce the liveness of the decentralized working zones. Moreover, the proposed method can provide the liveness of the Petri net model while preserving more reachable states of the system.

4.1. Global Controller Using Ring Assembling

The decentralized working zones can be reconnected and controlled through the global controller in a ring manner and we called such a process ring assembling. Ring assembling consists of a single global controller that oversees the operations of all the working zones in an FMS. The controller can perform the following functions: (i) it reconnects the working zones after decentralization, and (ii) it enforces liveness to the overall working zones in an FMS. The global controller connecting the working zones of an FMS decreases the structural complexity of the supervisory structure. The block diagram for the supervisory control using ring assembling is shown in Figure 4, and its equivalent representation using a Petri net model is depicted as shown in Figure 5.

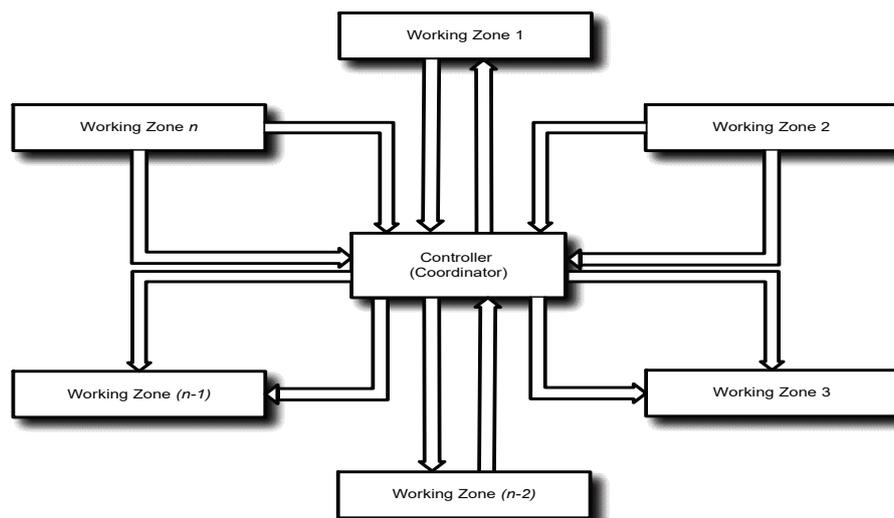


Figure 4. Block diagram of global controllers using ring assembling.

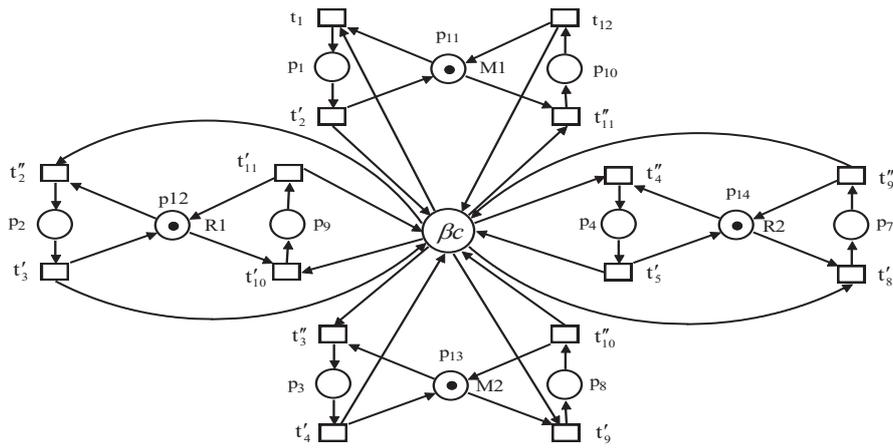


Figure 5. Petri net model with a global controller.

Let us consider the design of a global controller that controlled all the decentralized working zones. Let $\mathfrak{S}_i = (P'_i, T'_i, F'_i), i \in \{1, \dots, \ell\}$ be decentralized working zones from $N' = (P', T', F')$. Let P'_R be the set of resource places in N' and P'_i be the set of place for the decentralized system \mathfrak{S}_i . Let λ_i denote the characteristic activity place for each decentralized zone and can be represented as

$$\forall i \in \{1, \dots, \ell\}, \lambda_i = \sum_{j=1}^x M(p_j)p_j \setminus M(r_j)r_j \leq \beta_G, \tag{6}$$

where $p_j \in P_i^q$ and $r_j \in P_R$ be the activity and resource place associated with the working zones \mathfrak{S}_i , i.e., $i \in \{1, \dots, \ell\}$ respectively. β_i is the initial marking of the decentralized zones. $x = |P_i^q|$ is the number of the activity places in \mathfrak{S}_i . The overall characteristic activity place of the global controller is

$$\sum_{i=1}^{\ell} \lambda_i \leq \beta_i. \tag{7}$$

Let L_i be the marking of places in the characteristic activity place of \mathfrak{S}_i and D_{L_i} denote the incidence matrix of \mathfrak{S}_i . Let \mathcal{C} denote the global controller connected between the working zones (i.e., $\mathfrak{S}_1, \dots, \mathfrak{S}_\ell$) and D_{C_i} be the incidence matrix of \mathfrak{S}_i connected to the global controller. D_{C_i} can be computed as

$$D_{C_i} = -L_i D_{L_i}. \tag{8}$$

The initial number of tokens in the controller can be determined by the initial number of tokens in the characteristic activity place of the working zones associated with a decentralized controller i.e., if β_i represents the initial marking of the decentralized working zone for $i \in \{1, \dots, \ell\}$, and β_C denote the initial marking of \mathcal{C} , then the initial marking of the decentralized controller is

$$\beta_C = \sum_{i=1}^{\ell} \beta_i. \tag{9}$$

Finally, let D'_C be the incidence matrix of the global controller. The incidence matrix of the global controller connecting all the decentralized working zones is

$$D'_C = [D_{C_1}] + \dots + [D_{C_\ell}], \tag{10}$$

where D'_C has a dimension of $1 \times (2 * |T'| - *|\Psi| - *|\vartheta|)$, and $|T'|$ is the number of transitions in the subnet N' , while $|\Psi|$ and $|\vartheta|$ are the number of sink and source transition, respectively.

Theorem 3. *The decentralized controller computed using Algorithm 2 can enforce liveness of the Petri net model (N^c, M_0^c) .*

Algorithm 2 Computation of decentralized controller

Require: Petri net model (N, M_0) of FMS structure suffering from deadlock

Ensure: Controlled Petri nets (N^c, M_0^c)

- 1: compute the subnet N' of the Petri net model
 - 2: compute the decentralized working zones using Algorithm 1 i.e., \mathfrak{S}_i using Algorithm 1
 - 3: **for** $(1, \ell, i++)$ **do**
 - 4: compute λ_i
 - 5: compute β_i
 - 6: **end**
 - 7: computes the overall characteristic activity place for the global controller
 - 8: computes the overall incidence matrix of the global controller D'_C using Equation (10)
 - 9: computes the initial marking of the global controller β_C using Equation (9)
 - 10: add the global controller to the working zones \mathfrak{S}_i
 - 11: Output the controlled Petri net model (N^c, M_0^c)
-

Proof. Since all the working zones forming a minimal P-semiflow with the global controller C exist, and the places in the working zones constitute a minimal P-semiflow in (N^c, M_0^c) , at each reachable marking of (N^c, M_0^c) , each working is marked and can't lead the system into a deadlock region. \square

Proposition 1. *Let ϑ and ϑ' be the set of source transitions in (N', M_0) and (N^c, M_0^c) , respectively. The set of sink transition in (N', M_0) is equal to the set of sink transition in (N^c, M_0^c) i.e., $\vartheta = \vartheta'$.*

Proof. Since the source transitions from (N', M_0) that are part of the common transition of the decentralized working zones exist, i.e., $\exists t \in \vartheta, t \notin T_d$. Moreover, both the P-invariants in (N', M_0) and that of (N^c, M_0^c) are preserved, which lead the systems to have equal firing condition at the initial markings. Perhaps, all of the transitions that can be enabled at the initial marking are source transitions. Hence, we conclude that $\vartheta \subseteq \vartheta'$ or $\vartheta' \subseteq \vartheta$ holds. \square

Theorem 4. *Let \mathfrak{S}_i and \mathfrak{S}_j be the decentralized working zones from N' with $\mathfrak{S}_i = (P_i^q, T_i^q, F_i^q)$ and $\mathfrak{S}_j = (P_j^q, T_j^q, F_j^q)$, $\forall i, j \in \{1, \dots, \ell\}$. Assume t is a common transition between the working zones \mathfrak{S}_i and \mathfrak{S}_j ; t can be expressed as t'_i and t'_j belongs to \mathfrak{S}_i and \mathfrak{S}_j , respectively, if $[t] \equiv [t'_i], T_i^q \cap T_j^q \neq \emptyset$.*

Proof. Suppose M^* and M^{**} are the markings generated from (N^c, M_0^c) at markings M_i and M_j by firing transitions t' and t'' , respectively. Let M'^* be a marking after transition t fired from a marking M' of (N', M_0) . We prove that $M^* \geq_A M^{**}$ is true.

Let M_0 be the initial marking in (N', M_0) with $N' = (P', T', F')$ and M_0^c be the initial marking of (N^c, M_0^c) with $N^c = (P_A \cup P_R \cup \{P^0\} \cup C, T, F)$. Since at their initial markings $\sum_{(p_i \in P)} M_0(p_i) \equiv \sum_{(p_i \in P)} M_0^c(p_i) \setminus M(C)$, from Proposition 1, we know that the controlled Petri net model (N^c, M_0^c) and the subnet model (N', M_0) contain the same set of sink (ϑ) and source (ϕ) transitions. It implies that $M_0[t^s]M' = M_0^c[t^s]M_i$ is true. Similarly, it is also true for $M_0[t^s]M' = M_0^c[t^s]M_j$.

We proceed to show that it can also be true at a certain marking by firing a sequence of transitions for (N', M_0) that is mapped in (N^c, M_0^c) . Let σ be the set of a transition sequence that is mapped from (N', M_0) to (N^c, M_0^c) . The firing of σ from the initial markings can lead to $M_0[t^s]M_i[\sigma] \dots [t]M^*$ and $M_0^c[t^s]M'[\sigma] \dots [t]M^{**}$, since all the minimal P-invariants of (N', M_0) and (N^c, M_0^c) are preserved and are equal. This implies that the entire transition firing sequence σ of (N', M_0) can be traced in (N^c, M_0^c) , and subsequently leads $M'^* \geq_A M^*$ to hold. Similarly, it is true for $M'^* \geq_A M^{**}$. Consequently, if $M'^* \geq_A M^*$ and $M'^* \geq_A M^{**}$ is true, it is definitely true for $M^* \geq_A M^{**}$. \square

Let us consider the Petri net model of FMS shown in Figure 2. The Petri net model of FMS consists of five working zones as computed by Algorithm 1. The characteristic activity places in \mathfrak{S}_1 is $p_1 + p_{10} \leq 1$. Similarly, the characteristic activity places in $\mathfrak{S}_2, \mathfrak{S}_3, \mathfrak{S}_4$ and \mathfrak{S}_5 are $p_2 + p_9 \leq 1, p_3 + p_8 \leq 1, p_4 + p_7 \leq 1$ and $p_5 + p_6 \leq 1$, respectively. The overall characteristic activity place for the global controller is $p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9 + p_{10} \leq 5$. The incidence matrix of the decentralized working zones are:

$$D_{C_1} = - \begin{bmatrix} p_1 & p_{10} \\ 1 & 1 \end{bmatrix} \begin{bmatrix} t_1 & t'_2 & t'_{11} & t_{12} \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} p_1 \\ p_{10}, \end{matrix}$$

$$D_{C_2} = - \begin{bmatrix} p_2 & p_9 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} t''_2 & t'_3 & t'_{10} & t'_{11} \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} p_2 \\ p_9, \end{matrix}$$

$$D_{C_3} = - \begin{bmatrix} p_3 & p_8 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} t'_3 & t'_4 & t'_9 & t'_{10} \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} p_3 \\ p_8, \end{matrix}$$

$$D_{C_4} = - \begin{bmatrix} p_4 & p_7 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} t''_4 & t'_5 & t'_8 & t''_9 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} p_4 \\ p_7, \end{matrix}$$

$$D_{C_5} = - \begin{bmatrix} p_5 & p_6 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} t''_5 & t_6 & t_7 & t''_8 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix} \begin{matrix} p_5 \\ p_6. \end{matrix}$$

The constraints of the global controller connecting all the working zones are summarized as shown in Table 1. The global controller \mathcal{C} is responsible for connecting and controlled the operations of the working zones \mathfrak{S}_i . When all the decentralized working zones are connected via a global controller, the overall Petri net model is live with maximally permissive behaviour. The advantage of the proposed method is that it can be applied to complex systems. The complex systems can be easily addressed by dividing the system into various subproblems. The overall controlled systems are shown in Figure 6.

Table 1. Overall decentralized controllers for the Petri net model are shown in Figure 2.

Characteristic Activity Place	\mathcal{C}_i	Pre	Post	$\beta_{\mathcal{C}}$
$p_1 + p_2 + p_3 + p_4 + p_5 + p_6 + p_7 + p_8 + p_9 + p_{10} \leq 5$	\mathcal{C}	$t'_2, t'_3, t'_4, t'_5, t_6, t''_8, t'_9, t'_{11}, t_{12}$	$t_1, t''_2, t''_3, t''_4, t''_5, t_7, t'_8, t'_9, t'_{10}, t'_{11}$	5

4.2. Computational Complexity

Algorithm 1 computes the decentralized working zones from (N', M_0) with $N = (P', T', F')$. The algorithm is used to search and split up the P-invariant from (N, M_0) . Obviously, each P-invariant is associated with the resource place in (N, M_0) . Let the number of resource places be n i.e., $|P_R| = n$. The “While” loop is executed n times to search for the P-invariants in (N, M_0) . Since the number of resource places is less than the number of activity places in (N', M_0) , therefore, in the worst case scenario, we have n times to search the P-invariants that satisfy Equations (1)–(5). The complexity of Algorithm 1 to decentralize the working zones from (N', M_0) in the worst case is $\mathcal{O}(n)$.

Algorithm 2 is used to compute the global controller for all the decentralized zones of the subnet model. The “For loop” is executed ℓ times to compute the characteristic activity places for the decentralized working zones. Based on the fact that each decentralized working zone is associated with the resource place of the Petri net model, the computational complexity of Algorithm 2 in the worst case is therefore $\mathcal{O}(n\ell)$. In general, the computational complexity of the proposed method has polynomial time complexity.

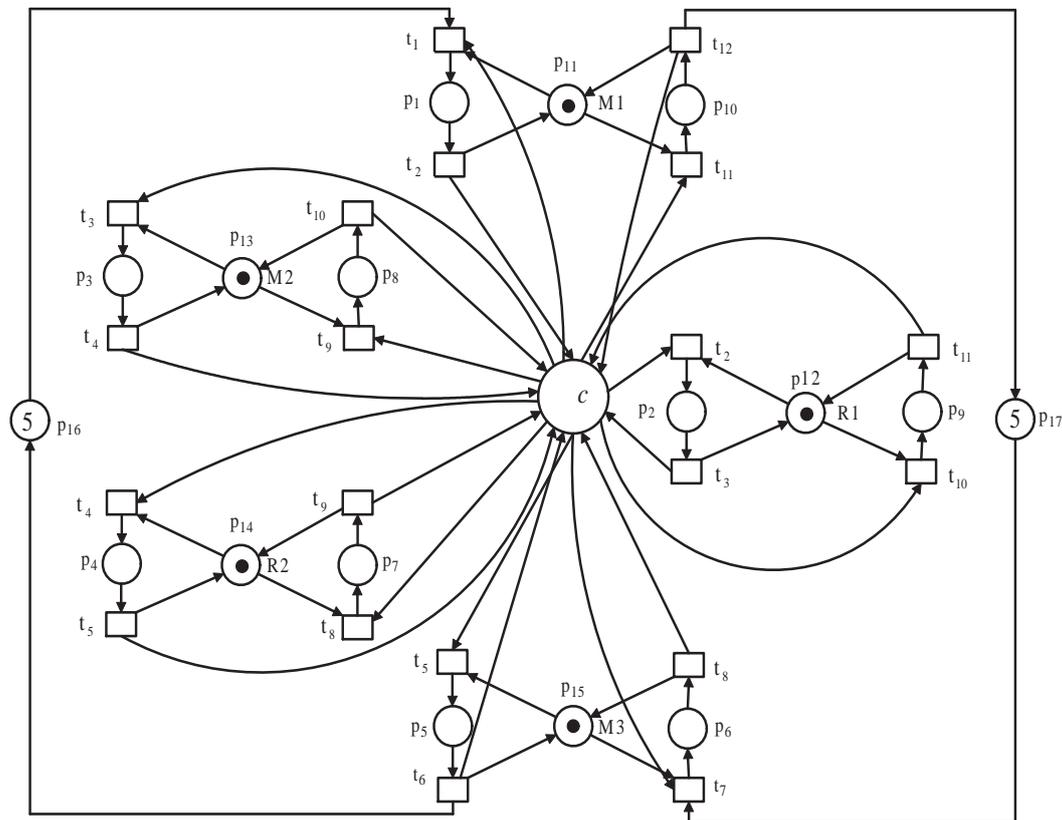


Figure 6. Controlled Petri net model.

5. Experimental Examples

This section presents some experimental examples to show the applicability of the proposed method. In each experimental example, the global controller can enforce the liveness of the Petri net model with maximally permissive behavior.

Example 1. Let us consider the Petri net model of a flexible manufacturing system (FMS) shown in Figure 7. The FMS consists of two concurrent processes. The places in the Petri net model are partitioned into three sets as $P = P_A \cup P_R \cup P^0$, where $P^0 = \{p_1, p_8\}$, $P_R = \{p_{13}, \dots, p_{16}\}$ and $P_A = \{p_2, \dots, p_7, p_9, \dots, p_{12}\}$. Places p_2, p_3, p_4, p_5, p_6 and p_7 represent the operations of the production sequence of part P1, respectively. Similarly, places p_9, p_{10}, p_{11} and p_{12} represent that of P2. Places p_{13}, p_{14}, p_{15} and p_{16} represent the shared resources of the FMS. The Petri net model for an FMS shown in Figure 7 is prompt with deadlocks.

First, the set of sink and source transitions from the Petri net model shown in Figure 7 are $\Psi = \{t_8, t_{13}\}$ and $\vartheta = \{t_1, t_4, t_9\}$, respectively. The subnet model is computed by removing all the I/O places in the Petri net model as shown in Figure 8 and then by applying Algorithm 1 to the subnet model shown in Figure 8. The minimal place invariants in the subnet model shown in Figure 8 are $I_1 = (010100000011000)$, $I_2 = (0010100000100100)$, $I_3 = (0000010000000010)$ and $I_4 = (0000001010000001)$

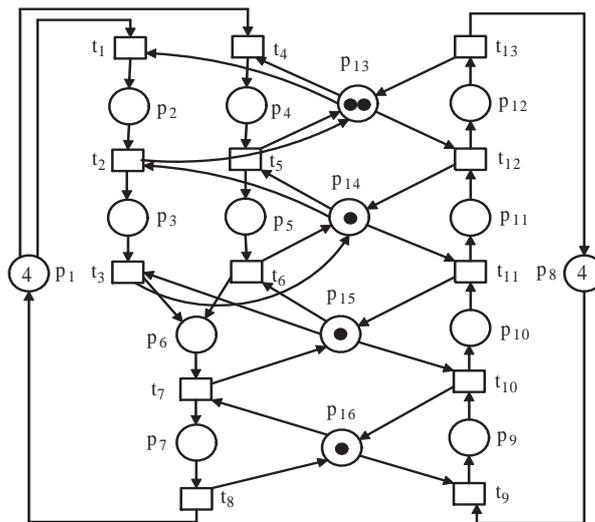


Figure 7. Petri net model for an FMS.

With their support, $\|I_1\| = \{p_2, p_4, p_{12}, p_{13}\}$, $\|I_2\| = \{p_3, p_5, p_{11}, p_{14}\}$, $\|I_3\| = \{p_6, p_{10}\}$, and $\|I_4\| = \{p_7, p_9, p_{16}\}$, respectively. Using Equations (1)–(5), the working zones are $\mathfrak{S}_1 = (P_1^q, T_1^q, F_1^q)$ with $P_1^q = \{p_2, p_4, p_{12}, p_{13}\}$, $T_1^q = \{t_1, t_2, t_4, t_5\}$, $F_1^q \subseteq (P_1^q \times T_1^q) \cup (T_1^q \times P_1^q)$; $\mathfrak{S}_2 = (P_2^q, T_2^q, F_2^q)$ with $P_2^q = \{p_3, p_5, p_{11}, p_{14}\}$, $T_2^q = \{t_2, t_3, t_5, t_6\}$, $F_2^q \subseteq (P_2^q \times T_2^q) \cup (T_2^q \times P_2^q)$; $\mathfrak{S}_3 = (P_3^q, T_3^q, F_3^q)$ with $P_3^q = \{p_6, p_{10}, p_{15}\}$, $T_3^q = \{t_3, t_6, t_7, t_{10}, t_{11}\}$, $F_3^q \subseteq (P_3^q \times T_3^q) \cup (T_3^q \times P_3^q)$ and $\mathfrak{S}_4 = (P_4^q, T_4^q, F_4^q)$ with $P_4^q = \{p_7, p_9, p_{16}\}$, $T_4^q = \{t_7, t_8, t_9, t_{10}\}$, $F_4^q \subseteq (P_4^q \times T_4^q) \cup (T_4^q \times P_4^q)$. The decentralized working zones are shown in Figure 9.

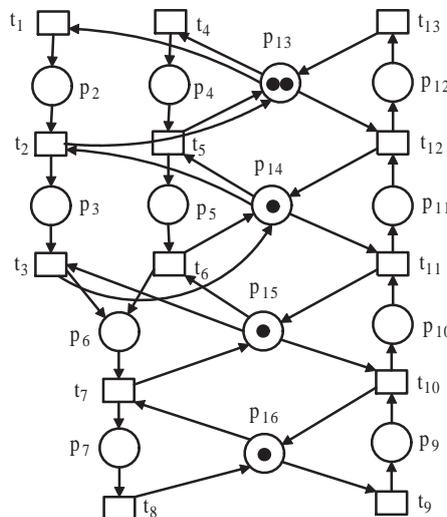


Figure 8. A subnet model (N', M_0) for an FMS.

Third, the characteristic activity place for the working zones $\mathfrak{S}_1, \mathfrak{S}_2, \mathfrak{S}_3$ and \mathfrak{S}_4 are $\lambda_1 = p_2 + p_4 + p_{12} \leq 2$, $\lambda_2 = p_3 + p_{11} + p_{14} \leq 1$, $\lambda_3 = p_6 + p_{10} + p_{15} \leq 1$ and $\lambda_4 = p_7 + p_9 + p_{16} \leq 1$, respectively. The global controller is designed using the characteristic activity place for the working zones.

When the decentralized systems are connected to the global controller, the resultant Petri net model shown in Figure 10 is live with maximally permissive behavior of 360 states as shown in Table 2. Table 3 provides a comparison between the proposed method with the previous methods in the literature. From the comparison Table 3, our proposed method can obtain minimal supervisory structure.

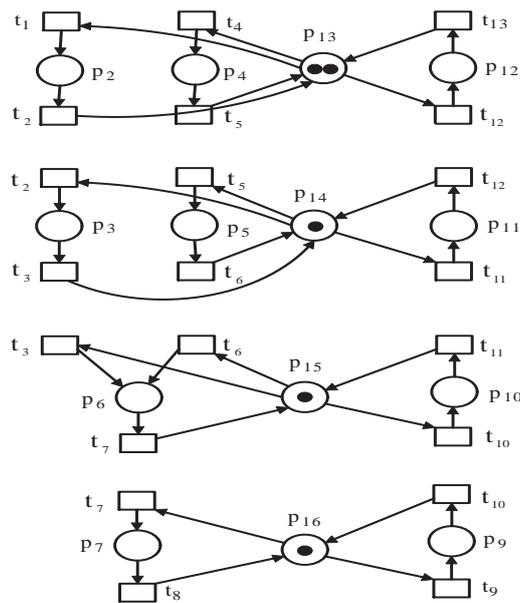


Figure 9. A subnet model $N = (N', M_0)$ with $N' = (P', T', F')$.

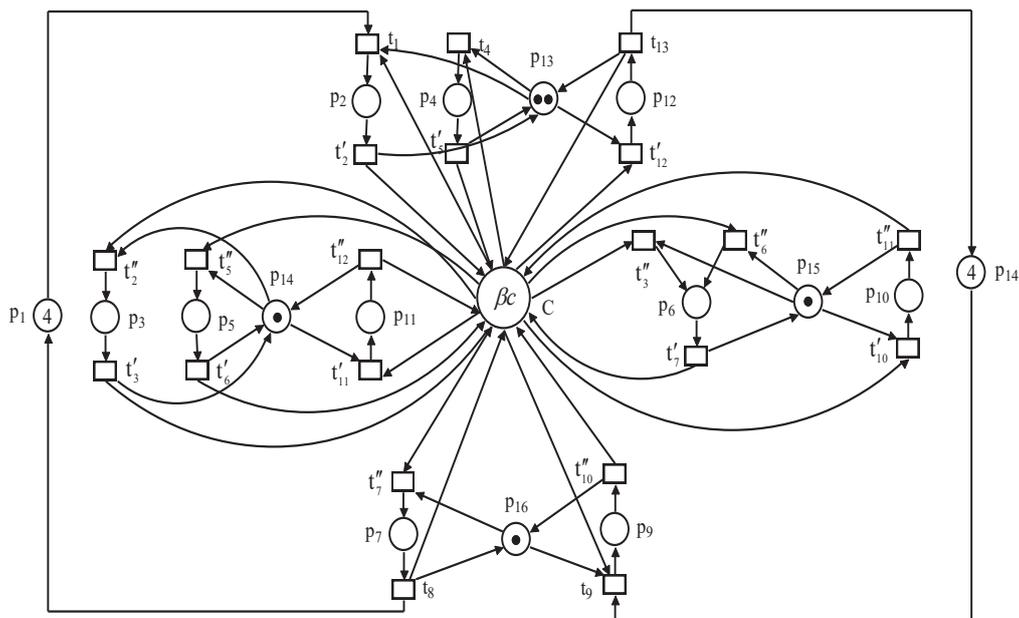


Figure 10. A controlled Petri net model (N^c, M_0^c) .

Table 2. Behavior of the controlled Petri net model from the minimal to maximum initial marking.

k	Reachable States	Is It Optimal?	Is It Live?
1	11	No	Yes
$\beta_c = 5$	54	No	Yes
$1 \leq k \leq \beta_c$	156	No	Yes
4	288	No	Yes
5	360	Yes	Yes

Table 3. Performance comparison of the proposed method with some of the available methods in the literature.

Methods	No. of Controllers	No. of Control Arcs	No. of Reachable States
[29]	5	28	151
[27]	5	28	151
[26]	6	30	360
Proposed method	1	32	360

Example 2. Suppose we consider a Petri net model for a flexible manufacturing system (FMS) shown in Figure 11. The FMS consists of three concurrent processes. The places in the Petri net model can be represented as $P = P_A \cup P_R \cup P^0$, where $P^0 = \{p_1, p_{13}, p_{14}\}$, $P_R = \{p_{20}, \dots, p_{27}\}$ and $P_A = \{p_2, \dots, p_{12}, p_{15}, \dots, p_{19}\}$. Places p_2, p_3 , and p_4 represent the operations of the production sequence of part P1, respectively. Similarly, places $p_5, p_6, p_7, p_8, p_9, p_{10}, p_{11}$ and p_{12} represent that of P2. The places $p_{15}, p_{16}, p_{17}, p_{18}$ and p_{19} represent the operations of the production sequence of part P3, respectively. Places $p_{20}, p_{21}, p_{22}, p_{23}, p_{24}, p_{25}, p_{26}$ and p_{27} represent the shared resources. The Petri net model for an FMS shown in Figure 11 is prompt with deadlocks.

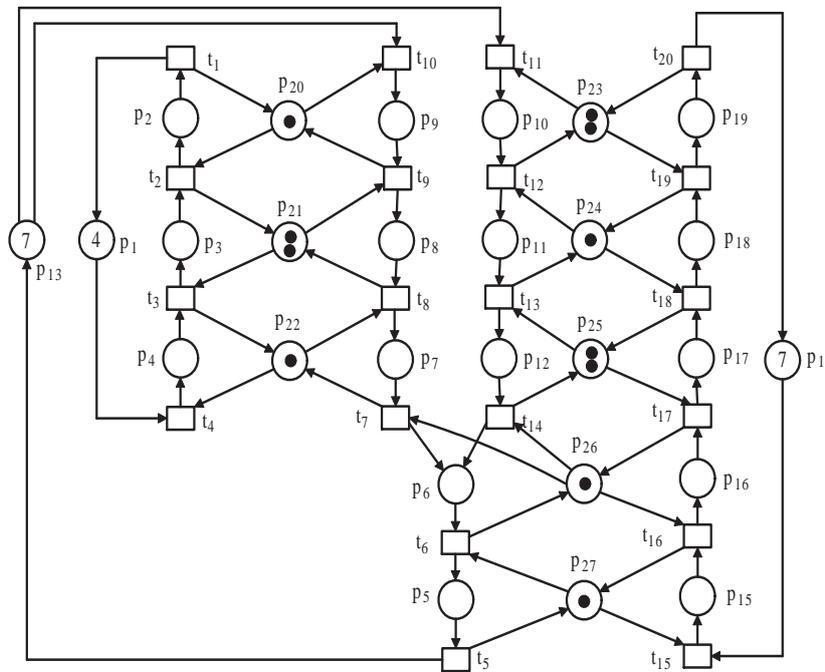


Figure 11. Petri net model for an FMS.

First, the set of the sink and source transition from the Petri net model shown in Figure 11 are $\Psi = \{t_1, t_5, t_{20}\}$ and $\vartheta = \{t_4, t_{10}, t_{11}, t_{15}\}$, respectively. The subnet model is computed by removing all the I/O places from the sink and source transitions in the Petri net model as shown in Figure 11 and then, by applying Algorithm 1 on the Petri net model shown in Figure 12, the subnet model has eight minimal place invariants.

The support of the minimal P-invariants are $\|I_1\| = \{p_2, p_9, p_{20}\}$, $\|I_2\| = \{p_3, p_8, p_{21}\}$, $\|I_3\| = \{p_4, p_7, p_{22}\}$, $\|I_4\| = \{p_6, p_{16}, p_{26}\}$, $\|I_5\| = \{p_5, p_{15}, p_{27}\}$, $\|I_6\| = \{p_{12}, p_{17}, p_{25}\}$, $\|I_7\| = \{p_{11}, p_{18}, p_{24}\}$ and $\|I_8\| = \{p_{10}, p_{19}, p_{23}\}$. Using Equations (1)–(5), the working zones are $\mathfrak{S}_1 = (P_1^q, T_1^q, F_1^q)$ with $P_1^q = \{p_4, p_7, p_{22}\}$, $T_1^q = \{t_1, t_2, t_9, t_{10}\}$, $F_1^q \subseteq (P_1^q \times T_1^q) \cup (T_1^q \times P_1^q)$; $\mathfrak{S}_2 = (P_2^q, T_2^q, F_2^q)$ with $P_2^q = \{p_3, p_8, p_{21}\}$, $T_2^q = \{t_2, t_3, t_8, t_9\}$, $F_2^q \subseteq (P_2^q \times T_2^q) \cup (T_2^q \times P_2^q)$; $\mathfrak{S}_3 = (P_3^q, T_3^q, F_3^q)$ with $P_3^q = \{p_4, p_7, p_{22}\}$, $T_3^q = \{t_3, t_4, t_7, t_8\}$, $F_3^q \subseteq (P_3^q \times T_3^q) \cup (T_3^q \times P_3^q)$; $\mathfrak{S}_4 = (P_4^q, T_4^q, F_4^q)$ with $P_4^q = \{p_6, p_{16}, p_{26}\}$, $T_4^q = \{t_6, t_7, t_{14}, t_{16}, t_{17}\}$, $F_4^q \subseteq (P_4^q \times T_4^q) \cup (T_4^q \times P_4^q)$; $\mathfrak{S}_5 = (P_5^q, T_5^q, F_5^q)$ with $P_5^q = \{p_5, p_{15}, p_{27}\}$, $T_5^q = \{t_5, t_6, t_{15}, t_{16}\}$, $F_5^q \subseteq (P_5^q \times T_5^q) \cup (T_5^q \times P_5^q)$; $\mathfrak{S}_6 = (P_6^q, T_6^q, F_6^q)$ with $P_6^q = \{p_{12}, p_{17}, p_{25}\}$,

$T_6^q = \{t_{13}, t_{14}, t_{17}, t_{18}\}$, $F_6^q \subseteq (P_6^q \times T_6^q) \cup (T_6^q \times P_6^q)$; $\mathfrak{S}_7 = (P_7^q, T_7^q, F_7^q)$ with $P_7^q = \{p_{11}, p_{18}, p_{24}\}$, $T_7^q = \{t_{12}, t_{13}, t_{18}, t_{19}\}$, $F_7^q \subseteq (P_7^q \times T_7^q) \cup (T_7^q \times P_7^q)$ and $\mathfrak{S}_8 = (P_8^q, T_8^q, F_8^q)$ with $P_8^q = \{p_{10}, p_{19}, p_{23}\}$, $T_8^q = \{t_{11}, t_{12}, t_{19}, t_{20}\}$, $F_8^q \subseteq (P_8^q \times T_8^q) \cup (T_8^q \times P_8^q)$. The working zones are shown in Figure 13.

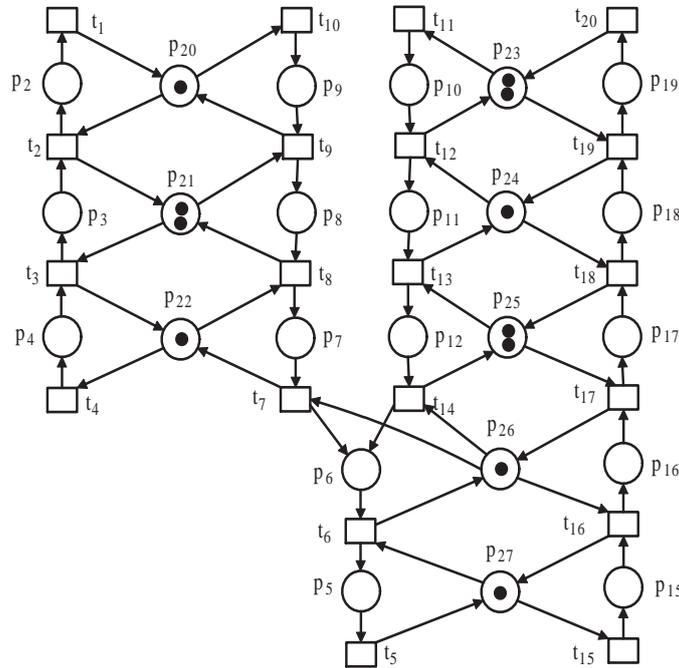


Figure 12. A subnet model for (N', M_0) .

Third, the characteristic activity place associated with the working zones $\mathfrak{S}_1, \mathfrak{S}_2, \mathfrak{S}_3, \mathfrak{S}_4, \mathfrak{S}_5, \mathfrak{S}_6, \mathfrak{S}_7$ and \mathfrak{S}_8 , respectively, are $p_2 + p_9 + p_{20} \leq 1$, $p_3 + p_8 + p_{21} \leq 2$, $p_4 + p_7 + p_{22} \leq 1$, $p_6 + p_{16} + p_{26} \leq 1$, $p_5 + p_{15} + p_{27} \leq 1$, $p_{12} + p_{17} + p_{25} \leq 2$, $p_{11} + p_{18} + p_{24} \leq 2$ and $p_{10} + p_{19} + p_{23} \leq 1$. The global controller is computed using the characteristic activity places, and its detailed behaviour of the global controller is presented in Table 4.

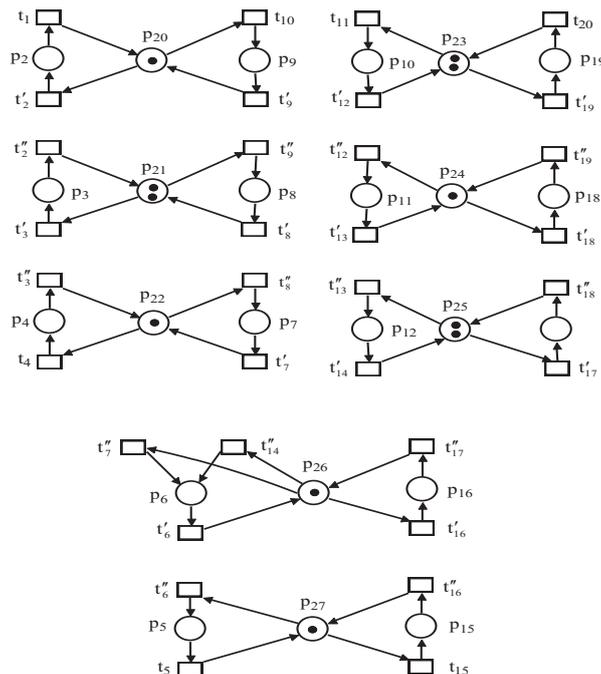


Figure 13. Decentralized subnets of Petri net model for FMS.

Table 4. Behaviour of the controlled Petri net model from the minimal to maximum initial marking.

k	Reachable States	Is It Optimal?	Is It Live?
1	17	No	Yes
2	138	No	Yes
$\beta_C = 11$	712	No	Yes
$1 \leq k \leq \beta_C$	2615	No	Yes
	7251	No	Yes
	15,730	No	Yes
	27,392	No	Yes
	39,240	No	Yes
	47,736	No	Yes
	51,624	No	Yes
	52,488	Yes	Yes

When the decentralized systems are connected to the global controller, the resultant Petri net model shown in Figure 12 is live with maximally permissive behavior of 52,488 states. Table 5 provides a comparison between our proposed method with the previous methods in the literature. According to the table shown in 5, our proposed method can obtain only one controlled place to enforce the liveness of the Petri net model shown in Figure 12. The controlled Petri nets are presented in Figure 14.

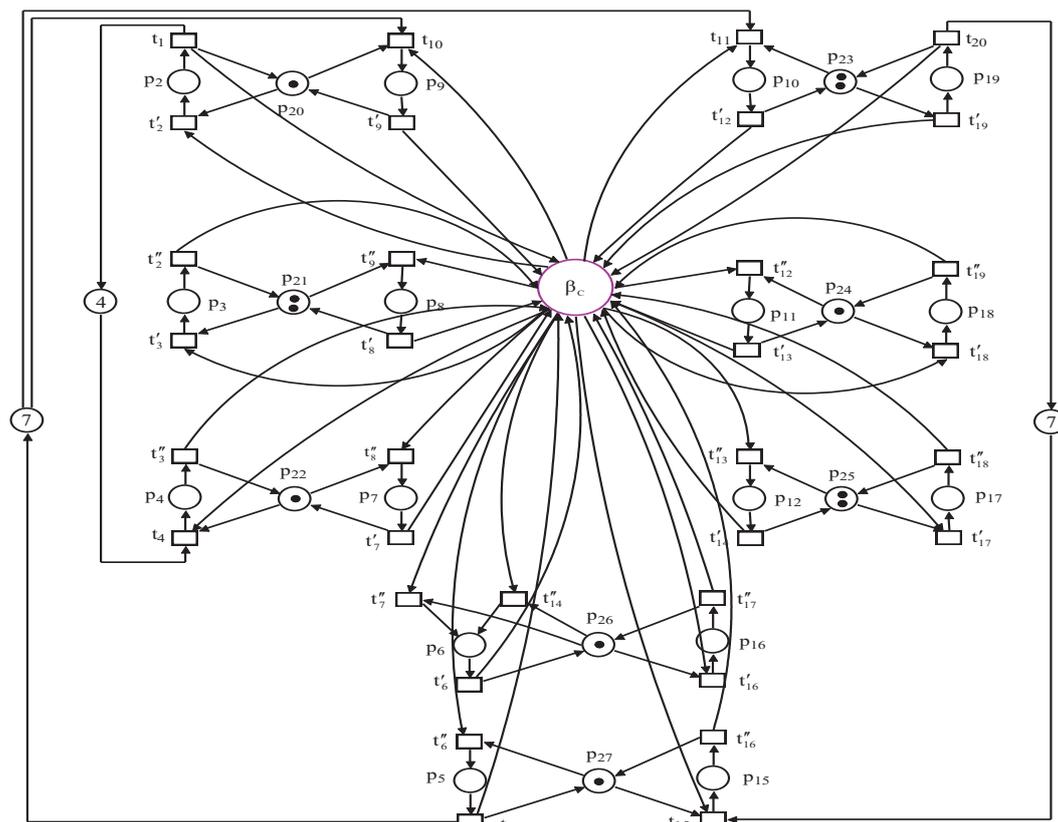


Figure 14. Controlled Petri net (N^c, M_0^c) model using a global controller.

Table 5. Performance comparison of the proposed method with some of the available methods in the literature.

Methods	No. of Controllers	No. of Control Arcs	No. Reachable States
[27]	15	45	35,254
[29]	12	56	35,254
[26]	6	56	52,488
Proposed method	1	54	52,488

6. Discussion

The proposed method reduces the computational complexity when compared with the traditional method available in the literature as it avoids the computation of the reachability graph that leads to state explosion problems. Our proposed method utilized the structural analysis to decompose the Petri net model into working zones and later design a global controller for the decentralized working zones without solving integer linear programming problems (ILLPs) as adopted mostly in the literature. The ILLP method cannot solve the large sized Petri nets because it generates a lot of variables and it is NP-hard.

A supervisor is said to have less structural complexity when it has less control places that can be added to the uncontrolled Petri net model. From Table 3 and 5 of our proposed manuscript, our proposed method only has one control place with 32 and 54 control arcs, respectively, when compared with the existing policy in the literature. It shows that our proposed method has less structural complexity. The advantage derived from having a single controller is to reduce the structural complexity and henceforth reduce the cost of the implementation and maintenance when validated. The proposed method can work on an S³PR Petri net for flexible manufacturing systems more specifically for the robotic assembling line cell and transportation systems.

An assembling line cell manufacturing usually processed parts in different work zones. Parts are added through the input/output (buffer) to the first work zone in the concurrent process to add a semi-finished good (processed the part) and then move it to the next immediate next working zone for further processing. Similarly, this procedure continues until all the work zones have processed the parts in sequence along the concurrent process of the Petri net model. However, a system with multiple concurrent processes exists in real assembling cells such as the one we model in this proposed paper. The activity places from different processes of the Petri net model can process parts simultaneously from the first work zone to the last working zone in the assembling line cell.

Our proposed method can work on intelligent transportation systems such as an automatic guided vehicle (AGV), modern railways, traffic signal control systems, and parking guidance and information systems. The AGVs work based on working zone areas to avoid collision and deadlocks and to enhance the effectiveness of the systems. In a system of multiple AGVs, each AGV has its assigned working zones that can move to deliver goods to the specified station freely without coincidence to another AGV in the systems. Modern railway stations have multiple railway lines with various trains in the systems. To avoid deadlocks in the systems, each train has assigned working zone areas that can operate from one station to the next available station until it reaches a final destination. Similarly, traffic signal control systems and parking guidance systems work based on working zone areas. In traffic control, the working zones are the lanes of the road control by the traffic lamp while, in the car parking systems, the working zones are on the floor level of the systems.

7. Conclusions

This paper presents a new method for designing a global supervisory structure for decentralized systems using a Petri net model with an S³PR structure. The proposed method developed two algorithms. First, Algorithm 1 is used to compute the decentralized working zones from the subnet model N' . Algorithm 2 is used to design the global controller for the decentralized working zones computed by Algorithm 1. A ring assembling method is presented to design the global controller for

the decentralized working zones that are responsible for reconnecting and controlling the operations of all the decentralized systems. The proposed method has these advantages: (i) it can be applied to a complex Petri net model for flexible manufacturing systems based on the fact that the proposed method can divide the system into several sub-problems; (ii) it has less computational complexity for the computation of the global controller; and (iii) the proposed method can obtain a minimal number of decentralized controllers to enforce liveness of the uncontrolled Petri net model.

Author Contributions: Conceptualization, M.B.; Methodology, M.B.; Formal Analysis, M.B.; Writing—Original Draft Preparation, M.B.; Writing—Review & Editing, L.H.; Supervision, L.H.; Project Administration, L.H.; Funding Acquisition, L.H.

Funding: This research was funded by the National Science Basic Research Plan in Shaanxi Province of China under Grant No. 2018JM6089.

Acknowledgments: The authors acknowledge the School of Electronics and Information of Xi'an Polytechnic University, 710048, China for the support given to conduct this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Lacerda, B.; Lima, P.U. LTL-Based Decentralized Supervisory Control of Multi-Robot Tasks Modelled as Petri Nets. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, San Francisco, CA, USA, 25–30 September 2011; pp. 3081–3086.
2. Palomeras, N.; Ridao, P.; Silvestre, C.; El-fakdi, A. Multiple vehicles mission coordination using Petri nets. In Proceedings of the IEEE International Conference on Robotics and Automation, Anchorage, AK, USA, 3–8 May 2010; pp. 3531–3536.
3. Lee, J.S.; Wang, Y.M. A Preliminary Application of Petri Nets to the Supervision of Remotely Operated Systems. In Proceedings of the IECON 2010-36th Annual Conference on IEEE Industrial Electronics Society, Glendale, AZ, USA, 7–10 November 2010; pp. 2145–2149.
4. Lee, J.S.; Zhou, M.C.; Hsu, P.L. A Petri-Net Approach to Modular Supervision With Conflict Resolution for Semiconductor Manufacturing Systems. *IEEE Trans. Autom. Sci. Eng.* **2007**, *4*, 584–588. [[CrossRef](#)]
5. Lee, J.S.; Zhou, M.C.; Hsu, P.L. An application of Petri nets to supervisory control for human-computer interactive systems. *IEEE Trans. Ind. Electr.* **2005**, *52*, 1220–1226. [[CrossRef](#)]
6. Dou, C.; Lv, M.F.; Zhao, T.Y.; Ji, Y.P.; Li, H. Decentralised coordinated control of micro-grid based on multi-agent system. *IET Gener. Transm. Distrib.* **2015**, *9*, 2474–2484. [[CrossRef](#)]
7. Hei, X.; Takahashi, S.; Hideo, N. Toward developing a Decentralized Railway Signalling System Using Petri Nets. In Proceedings of the 2008 IEEE Conference on Robotics, Automation and Mechatronics, Chengdu, China, 21–24 September 2008; pp. 851–855.
8. Loures, E.R.; dos Santos, E.A.P.; Buseti de Paula, M.A. Maintenance integration in a modular supervision framework based on Petri net with objects. Application to a robot-driven flexible cell. In Proceedings of the 2006 IEEE International Conference on Information Reuse & Integration, Waikoloa Village, HI, USA, 16–18 September 2006; pp. 69–74.
9. Herrero-Pérez, D.; Martínez-Barberá, H. Modeling Distributed Transportation Systems Composed of Flexible Automated Guided Vehicles in Flexible Manufacturing Systems. *IEEE Trans. Ind. Inf.* **2010**, *6*, 166–180. [[CrossRef](#)]
10. Hou, Y.; Zhao, M.; Liu, D.; Hong, L. An Efficient Siphon-Based Deadlock Prevention Policy for a Class of Generalized Petri Nets. *Discret. Dyn. Nat. Soc.* **2016**, *2016*. [[CrossRef](#)]
11. Safae, C.; Salma, M. Modelling and Simulation of Biochemical Processes Using Petri Nets. *Processes* **2018**, *6*, 97. [[CrossRef](#)]
12. Dimitri, L. Dynamical Scheduling and Robust Control in Uncertain Environments with Petri Nets for DESs. *Processes* **2017**, *5*, 54. [[CrossRef](#)]
13. Gu, C.; Li, Z.W.; Al-Ahmari, A. A Multistep Look-Ahead Deadlock Avoidance Policy for Automated Manufacturing Systems. *Discret. Dyn. Nat. Soc.* **2017**, *2017*. [[CrossRef](#)]

14. Wang, S.G.; Wang, C.Y.; Zhou, M.C. A transformation algorithm for optimal admissible generalized mutual exclusion constraints on Petri nets with uncontrollable transitions. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 3745–3750.
15. Huang, Y.S.; Pan, Y.L.; Su, P.J. Transition-based deadlock detection and recovery policy for FMSs using graph technique. *ACM Trans. Embed. Comput. Syst.* **2013**, *12*. [[CrossRef](#)]
16. Li, Z.W.; Wu, N.Q.; Zhou, M.C. Deadlock control of automated manufacturing system based on Petri nets—A literature review. *IEEE Trans. Syst. Man Cybern. Part C* **2012**, *42*, 437–460.
17. Ye, J.H.; Li, Z.W.; Giua, A. Decentralized Supervision of Petri Nets With a Coordinator. *IEEE Trans. Syst. Man Cybern. Syst.* **2015**, *45*, 955–966.
18. Yoo, T.S.; Lafortune, S. Decentralized Supervisory Control With Conditional Decisions: Supervisor Realization. *IEEE Trans. Autom. Control* **2005**, *50*, 1205–1211.
19. Cai, K.; Wonham, W.M. Supervisor Localization for Large-Scale Discrete-Event Systems. In Proceedings of the Joint 48th IEEE Conference on Decision and Control and 28th Chinese Control Conference, Shanghai, China, 16–18 December 2009; pp. 3099–3105.
20. Komenda, J.; van Schuppen, J.H. Modular control of discrete-event systems with coalgebra. *IEEE Trans. Autom. Control* **2008**, *53*, 447–460. [[CrossRef](#)]
21. Wong, K.C.; Lee, S. Structural decentralized control of concurrent discrete-event systems. *Eur. J. Control* **2002**, *8*, 477–491.
22. Mannani, A.; Gohari, P. Decentralized supervisory control of discrete-event systems over communication networks. *IEEE Trans. Autom. Control* **2008**, *53*, 547–559. [[CrossRef](#)]
23. Bashir, M.; Muhammad, B.B.; Li, Z.W. Minimal supervisory structure for flexible manufacturing system using Petri nets. In Proceedings of the IEEE International Conference on Control, Automation and Robotics, Hongkong, China, 28–30 April 2016; pp. 291–296.
24. Bashir, M.; Li, Z.W.; Uzam, M.; Wu, N.Q.; Al-Ahmari, A. On structural reduction of liveness-enforcing Petri net supervisors for flexible manufacturing systems: An algebraic approach. *IMA J. Math. Control Inf.* **2017**. [[CrossRef](#)]
25. Bashir, M.; Li, Z.W.; Uzam, M.; Al-Ahmari, A.; Wu, N.Q.; Liu, D.; Qu, T. A Minimal Supervisory Structure to Optimally Enforce Liveness on Petri Net Models for Flexible Manufacturing Systems. *IEEE Access* **2017**, *5*, 15731–15749. [[CrossRef](#)]
26. Bashir, M.; Liu, D.; Uzam, M.; Al-Ahmari, A.; Wu, N.Q.; Li, Z.W. Optimal Enforcement of Liveness to Flexible Manufacturing Systems Modeled with Petri Nets via Transition-based Controllers. *SAGE J. Adv. Mech. Eng. (AIME)* **2018**, *10*. [[CrossRef](#)]
27. Chen, Y.F.; Li, Z.W.; Barkaoui, K.; Giua, A. On the enforcement of a class of nonlinear constraints on Petri nets. *Automatica* **2015**, *55*, 116–124. [[CrossRef](#)]
28. Ye, J.; Zhou, M.C.; Li, Z.W.; Al-Ahmari, A. Structural Decomposition and Decentralized Control of Petri Nets. *IEEE Trans. Syst. Man Cybern. Syst.* **2017**./TSMC.2017.2703950. [[CrossRef](#)]
29. Li, Z.W.; Zhu, S.; Zhou, M.C. A Divide-and-Conquer Strategy to Deadlock Prevention in Flexible Manufacturing Systems. *IEEE Trans. Syst. Man Cybern. Part C* **2009**, *39*, 156–169.
30. Uzam, M.; Li, Z.W.; Gelen, G.; Zakariyya, R.S. A divide-and-conquer-method for the synthesis of liveness enforcing supervisors for flexible manufacturing systems. *J. Intell. Manuf.* **2016**, *27*, 1111–1129. [[CrossRef](#)]
31. Iordache, M.V.; Antsaklis, P.J. Decentralized Supervision of Petri Nets. *IEEE Trans. Autom. Control* **2006**, *51*, 376–381. [[CrossRef](#)]
32. Lee, E.J.; Toguyeni, A.; Dangoumau, N. A Petri Net based Decentralized Synthesis Approach for the Control of Flexible Manufacturing Systems. In Proceedings of the IMACS Multiconference on “Computational Engineering in Systems Applications”(CESA), Beijing, China, 4–6 October 2006; pp. 1497–1503.
33. Gasparri, A.; Paola, D.D.; Giua, A.; Ulivi, G.; Naso, D. Consensus-Based Decentralized Supervision of Petri nets. In Proceedings of the 50th IEEE Conference on Decision and Control and European Control Conference (CDC-ECC), Orlando, FL, USA, 12–15 December 2011; pp. 1128–1135.
34. Basile, F.; Cordone, R.; Piroddi, L. A branch and bound approach for the design of decentralized supervisors in Petri net models. *Automatica* **2015**, *52*, 322–333. [[CrossRef](#)]
35. Luo, J.; Zhou, M.C. Petri-net controller synthesis for partially controllable and observable discrete event systems. *IEEE Trans. Autom. Control* **2017**, *62*, 1301–1313. [[CrossRef](#)]

36. Bourjij, A.; Boutayeb, M.; Cecchin, T. A Decentralized Approach for Computing Invariants in Large Scale and Interconnected Petri Nets. In Proceedings of the 1997 IEEE International Conference on Systems, Man, and Cybernetics, Computational Cybernetics and Simulation, Orlando, FL, USA, 12–15 October 1997; pp. 1741–1746.
37. Ezpeleta, J.; Colom, J.M.; Martinez, J. A Petri net based deadlock prevention policy for flexible manufacturing systems. *IEEE Trans. Robot. Autom.* **1995**, *11*, 173–184. [[CrossRef](#)]
38. Chu, F.; Xie, X.L. Deadlock analysis of Petri nets using siphons and mathematical programming. *IEEE Trans. Robot. Autom.* **1997**, *13*, 793–804.



© 2019 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 (<http://creativecommons.org/licenses/by/4.0/>).