Deadlock-Free Control of Automated Manufacturing Systems With Flexible Routes and Assembly Operations Using Petri Nets

Hesuan Hu, *Member, IEEE*, MengChu Zhou, *Fellow, IEEE*, Zhiwu Li, *Senior Member, IEEE*, and Ying Tang, *Senior Member, IEEE*

***Abstract*—In the context of automated manufacturing systems (AMS), Petri nets are widely adopted to solve the modeling, analysis, and control problems. So far, nearly all known approaches to liveness enforcing supervisory control investigate AMS with either flexible routes or assembly operations, whereas little work investigates them with both. In this paper, we propose a novel class of systems, which can well deal with both features so as to facilitate the control of more complex AMS. Using structural analysis, we show that liveness of their Petri net model can be attributed to the absence of undermarked siphons, which is realizable by synthesizing a proper supervisory controller. Moreover, an efficient method is developed and verified via AMS examples.**

***Index Terms*—Automated manufacturing systems (AMS), discrete event systems, Petri nets, supervisory control.**

I. Introduction

IN automated manufacturing systems (AMS), different types of products are produced through different routes by computer-controlled machines [1], [2], [9], [12], [16]. Deadlock may arise owing to the competition among the concurrent production processes for a limited number of shared resources, e.g., machines and robots [3], [4], [10], [11], [13], [14], [29]–[33], [37], [38], [42]. It deteriorates the system performance as no process can proceed to the end. Apparently, it is imperative for engineers to avoid such deadlock in AMS. In this paper, Petri nets are utilized as a mathematical tool to model, analyze, and control AMS.

An AMS is categorized as a type of resource allocation systems (RAS) as well as discrete event systems [5]–[8], [15], [17]–[28], [32], [34]–[36], [39], [41]–[53]. In AMS, a number of resources are shared and competed by several processes. From the resource or process perspectives, we can further divide AMS into different classes that can be treated differently. From the resource perspective, systems can be distinguished as single-unit and multiple-unit systems. The former permits only one resource used by each operation step. The latter permits an arbitrary number of same or different resources during each operation step. From the process perspective, the existing approaches deal with AMS with either flexible routes, e.g., [12], [16], [31], [33], [37], or assembly/disassebly operations [9]. An AMS owning both features is not seriously investigated owing to its complexity. By assembly/disassembly operations, we mean that: 1) synchronization is required and 2) a part can be split into different subparts and processed in parallel and these subparts are then used to compose the final product.

The seminal work in [12] recognizes that the deadlock states can be forbidden by introducing a monitor to each siphon to prevent it from being empty. This condition implies that the sum of tokens in a siphon must be greater than one; or equivalently, the sum of tokens in the complementary places of a siphon must be less than the initial token count in the siphon. Despite its importance, the work in [12] is far from perfection. It requires at least four improvements. First, the approach considers quite limited systems. At each operation step, only a unit of resources is allowed. In each process, only flexible routes are allowed while the assembly and disassembly operations are prohibited. This seriously constrains its applicability to more practical systems. Second, the approach is quite computationally demanding even for systems with moderate scales. This is because it requires the complete enumeration of all strict minimal siphons whose quantity increases exponentially with the size of a system. Third, the approach can induce an overly complex supervisor in its structure. This is because it requires to add one monitor to each identified siphon. Fourth, the approach cannot promise the maximal permissive behavior of the controlled systems. Many approaches are proposed by emphasizing the improvement of one or more aspects, e.g., Li and Zhou [29] derive one of the simplest supervisors without improving the other aspects.

In [9], deadlock problems are investigated in the systems with assembly/disassembly operations where the process operations are described by marked graphs. The approach establishes the relation between deadlock and the undermarkedness of siphons. As known, marked graphs cannot describe flexible routes that are inherent in modern AMS. Synchronization and flexibility requirements together pose a new challenge to designers. In this

Manuscript received August 19, 2011; revised February 20, 2012; accepted February 28, 2012. Date of publication May 10, 2012; date of current version December 19, 2012. This work was supported by the Natural Science Foundation of China under Grants 60474018, 51105395, 61034004, and 60773001.

Paper no. TII-11-427.

H. Hu is with the School of Electro-mechanical Engineering, Xidian University, Xi'an, Shaanxi 710071, China (e-mail: huhesuan@gmail.com).

M. C. Zhou is with the Key Laboratory of Embedded System and Service Computing, Ministry of Education, Tongji University, Shanghai 201804, China, and also with the Department of Electrical and Computer Engineering, New Jersey Institute of Technology, Newark, NJ 07102 USA (e-mail: zhou@njit.edu).

Z. Li is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an, Shaanxi 710071, China, and also with the Department of Computer Science, Martin-Luther University, Halle 06120, Germany (e-mail: zhwli@xidian.edu.cn).

Y. Tang is with the Department of Electrical and Computer Engineering, Rowan University, Glassboro, NJ 08028 USA (e-mail: tang@rowan.edu).

Digital Object Identifier 10.1109/TII.2012.2198661

paper, we allow the existence of flexible routes within each synchronized process.

Apparently, the proposed approach lifts some serious restrictions on the structure of a system. We attempt to establish the relationship between the liveness and the absence of undermarked siphons. A mathematical programming approach is developed to identify the deadlock states and remove them iteratively. Eventually, a live controlled net is obtained without enumerating all the reachable states and siphons in a Petri net model.

In the context of the abovementioned approaches, our contributions in this work can be summarized as follows.

First, we propose a novel class of systems, which can deal with both features of flexible routes and assembly/disassembly operations. Second, we establish the relationship between the nonliveness and the absence of undermarked siphons. Third, an algebraic method is proposed such that the siphons can be derived and controlled in an iterative way.

Section II reviews the basic definitions and notations of Petri nets used throughout this paper. Section III is devoted to a special class of Petri nets. In Section IV, following the theoretical work in [25], our method is proposed to derive a liveness-enforcing supervisor without enumerating either states or siphons. Section V illustrates an example to verify its effectiveness. Experimental results are provided in Section VI. Section VII concludes this paper.

II. Preliminaries

A Petri net is $N = (P, T, F, W)$, where P is a set of places, T is a set of transitions, $F \subseteq (P\times T)\cup(T\times P)$ is a set of directed arcs, and $W : (P\times T)\cup(T\times P) \rightarrow \mathbb{N} = \{0, 1, 2, \ldots\}$ such that $P\cup T \neq \emptyset$, $P\cap T = \emptyset$, and $W(x, y) = 0$ if $(x, y) \notin F$. The preset of a node $x \in P\cup T$ is defined as ${}^\bullet x = \{y \in P\cup T|(y, x) \in F\}$. Its postset $x^\bullet = \{y \in P\cup T|(x, y) \in F\}$. N is a state machine if $W : F \rightarrow \{1\}$ and $|{}^\bullet t| = |t^\bullet| = 1$. It is a marked graph if $W : F \rightarrow \{1\}$ and $|{}^\bullet p| = |p^\bullet| = 1$. N's input incidence matrix is $[N^-]$ so that $[N^-](i, j) = W(p_i, t_j)$ and output one is $[N^+]$ so that $[N^+](i, j) = W(t_j, p_i)$. Its incidence matrix $[N] = [N^+] - [N^-]$. $[N_{p_i}]$ (resp., $[N^-_{p_i}]$, $[N^+_{p_i}]$) is the i-th row of $[N]$ (resp., $[N^-]$, $[N^+]$). A Petri net is said to be an ordinary one if $\forall(x, y) \in F$, $W(x, y) = 1$. It is said to be a PT-ordinary one if $\forall p \in P$, $\forall t \in T$, if $(p, t) \in F$ then $W(p, t) = 1$. It is said to be a general one if $\exists(x, y) \in F$, $W(x, y) \geq 1$.

A marking of N is a mapping $M : P \rightarrow \mathbb{N}$. (N, M_0) is a net system with an initial marking M_0. t is enabled at M, denoted by $M[t\rangle$, if $\forall p \in^\bullet t$, $M(p) \geq W(p, t)$. M' is reachable from M, denoted by $M[\sigma\rangle M'$, if there exists a firing sequence $\sigma =< t_1 \ldots t_n >$ such that $M[t_1\rangle M_1 \ldots [t_n\rangle M'$. $\overrightarrow{\sigma}$ is a $|\ T\ |$-dimensional firing vector of σ, where $\overrightarrow{\sigma}(t)$ states the number of t's appearances in σ. The set of all markings reachable from M_0 is denoted by $R(N, M_0)$. Given (N, M_0), $t \in T$ is live under M_0 if $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $\ni M'[t\rangle$ holds. (N, M_0) is live if $\forall t \in T$, t is live. (N, M_0) is pure if $\forall(x, y) \in (P\times T)\cup(T\times P)$, $W(x, y) > 0 \Rightarrow W(y, x) = 0$.

A P- (resp., T-) vector is a column vector $I : P$ (resp., $J : T$) $\rightarrow \mathbb{Z}$ indexed by P (resp., T), where $\mathbb{Z}$ is the set of integers. A P-vector $I \neq \mathbf{0}$ becomes a P-invariant if $[N]^T \cdot I = \mathbf{0}$, where $\mathbf{0}$ means a vector of zeros. A P-invariant is called a P-semiflow if $I \geq \mathbf{0}$. $\|I\| = \{p \in P|I(p) \neq 0\}$ is called the support of I. For economy of space, $\sum_{p\in P} M(p)\cdot p$ (resp., $\sum_{p\in P} I(p)\cdot p$) is used to denote vector M (resp., I).

A nonempty set $S \subseteq P$ is a siphon if ${}^\bullet S \subseteq S^\bullet$. $Q \subseteq P$ is a trap if $Q^\bullet \subseteq {}^\bullet Q$. A strict minimal siphon is a siphon containing neither other siphon nor trap. Π means the set of siphons. $M(p)$ indicates the number of tokens in p under M. p is marked by M if $M(p) > 0$. The sum of tokens in S is denoted by $M(S)$, where $M(S) = \sum_{p\in S} M(p)$. A subset $S \subseteq P$ is marked by M if $M(S) > 0$. A siphon is undermarked if $\nexists t \in S^\bullet$ can fire. A string $x_1, x_2, \ldots$, and x_n is called a path of N if $\forall i \in \mathbb{N}_{n-1} = 1, 2, \ldots, n-1$, $x_{i+1} \in x_i^\bullet$. An elementary path is a path whose nodes are all different (except for, perhaps, x_1 and x_n), which is denoted by $EP(x_1, x_n)$. It is called a circuit if it is an elementary path and $x_1 = x_n$.

III. Petri Net Modeling of AMS

An approach is proposed in [9] to model systems with assembly processes. Its resulting models are called Augmented Marked Graphs (AMG). Their major feature is that their processes are described by marked graphs.

Definition 1: A simple marked graph (SMG) is a strongly connected marked graph $N = (P, T, F)$, where $P = \{p_0\}\cup P_A$, p_0 is called the idle place, $p \in P_A$ is called an activity place and each circuit in N contains p_0.

Tokens in an idle place represent the availability of its corresponding raw parts. Marked graphs can describe the assembly operation; however, it cannot model flexible routes for each activity in a process. We address this issue on the basis of the following notion.

Definition 2: A state machine block, called SM-block for short, $\mathcal{B}(p_s, p_e)$ is a Petri net where: 1) ${}^\bullet p_s = \emptyset$, $|\ p_s^\bullet\ | \geq 1$; 2) $p_e^\bullet = \emptyset$, $|^\bullet\ p_e\ | \geq 1$; and 3) the subnet between p_s and p_e is a state machine.

Fig. 1 shows three SM-blocks where $p_s = p_{11}$ and $p_e = p_{16}$. They are composed of multiple routes. Fig. 1(c) contains a rework path, i.e., $p_{15} - t_{16} - p_{14}$. An SM-block denotes that a job can be performed through several routes. They start at p_s and end at p_e. This is of significance. First, the efficiency can be improved since multiple resources are allowed to manufacture the same products in parallel. Second, a system becomes more reliable since the failure of some resources cannot necessarily paralyze the entire one compared with a rigid production line.

Definition 3: An Extended MG (EMG) is a Petri net $N = (P, T, F)$ derived from an SMG by iteratively replacing place p with an SM-block $\mathcal{B}(p_s, p_e)$ such that ${}^\bullet p_s = {}^\bullet p$ and $p_e^\bullet = p^\bullet$.

Fig. 2 shows the construction process of an EMG in Fig. 2(b) from SMG in Fig. 2(a), where p_3 is substituted by the SM-block in Fig. 1(a).

Apparently, an EMG can accommodate both assembly operations and flexible routes. This overcomes the defect that an SMG cannot describe the latter. As known, the involvement of flexible routes can greatly complicate the structural analysis of a system. To characterize the cyclic production feature of an AMS, p_0 is introduced to denote the initialization and termination of a process. Subsequently, we introduce Augmented EMG (AEMG) to define the interaction of a set of EMG through shared resources. To make the model as generic as possible, each step permits the use of multiple resources. Multiple same

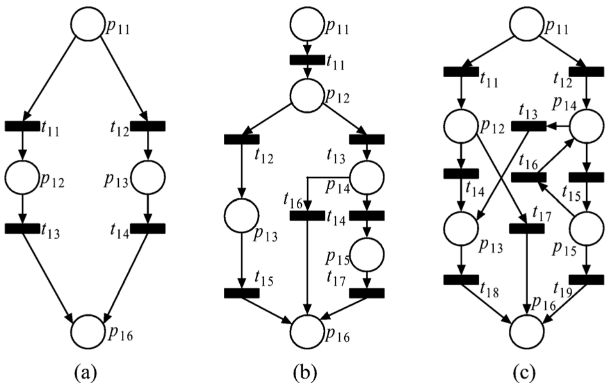


Fig. 1. Three SM-blocks.

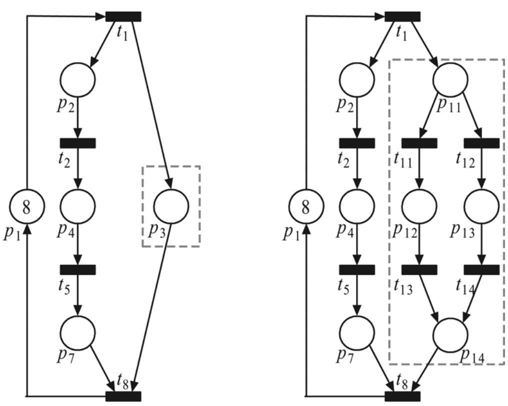


Fig. 2. The construction of EMG.

resources are also allowed to be engaged or released at the same time. Let $\mathbb{N}_K = \{1, 2, \ldots, K\}$.

Definition 4: An AEMG is a Petri net $N = (P, T, F, W)$ such that:

1) $P = P_0 \cup P_A \cup P_R$, where P_0 is a set of idle places such that $P_0 = \cup_{i \in \mathbb{N}_K} \{p_{0_i}\}$; P_A is a set of activity places such that $P_A = \cup_{i \in \mathbb{N}_K} P_{A_i}$. For $i, j \in \mathbb{N}_K$, $i \neq j$, $P_{A_i} \cap P_{A_j} = \emptyset$; and $P_R = \cup_{i \in \mathbb{N}_L} \{r_i\}$ is a set of resource places;
2) $T = \cup_{i \in \mathbb{N}_K} T_i$, where for each $i \in \mathbb{N}_K$, $T_i \neq \emptyset$, and for i, $j \in \mathbb{N}_K$, $i \neq j$, $T_i \cap T_j = \emptyset$;
3) For each $i \in \mathbb{N}_K$, $\overline{N}_i = N \mid (\{p_{0_i}\} \cup P_{A_i}, T_i, F_i)$ is an EMG; and
4) For each $r \in P_R$, $\exists$ a unique minimal P-semiflow $X_r \in \mathbb{N}^{|P|}$ such that $\{r\} = \|X_r\| \cap P_R$, $P_0 \cap \|X_r\| = \emptyset$, $P_A \cap \|X_r\| \neq \emptyset$, and $X_r(r) = 1$.

The structure of AEMG allows one to model the systems with both assembly operations and flexible routes. $\overline{N}_i$, $i \in \mathbb{N}_K$, means the manufacturing process of the ith product. It can be decomposed as an idle place p_{0_i}, a set of operation places P_{A_i}, and transitions T_i. A token in p_{0_i} represents a raw part of the ith product. The completion of a product signals the system to introduce a new part and this describes the feature of cyclic production. The places in P_{A_i} describe the various activities, some or all of which are needed to make the ith product. At a reachable marking $M \in R(N, M_0)$, a token in $p \in P_{A_i}$ means that the activity in p is ongoing. A transition $t \in T_i$ means an event whose occurrence represents that the ith job is processed by its previous operation and can be transferred to its subsequent one for further processing. If multiple places join at a transition, its firing means the occurrence of an assembly operation. The tokens in $r \in P_R$ denote the number of available resources. The fourth item imposes a P-semiflow X_r to each resource $r \in P_R$. $X_r(r) = 1$ means that one token in r denotes one copy of available resource r. For $p \in P_A$, $X_r(p) = k$ means that k copies of resources are required when a product is processed at the stage represented by p, where $0 \leq k \leq M_0(r)$. Also, $X_r^T \cdot M = X_r^T \cdot M_0$. This means that neither the resources can be created nor destroyed. Note that AEMG contains S^4R [37] (or S^3PGR [33]) and AMG [9]. The former one further contains S^3PR [12], LS^3PR [13], and PPN [40].

Definition 5: Given an AEMG $N = (P_0 \cup P_A \cup P_R, T, F, W)$, (N, M_0) is acceptably marked if: 1) $M_0(p_0) \geq 1$, $\forall p_0 \in P_0$; 2) $M_0(p) = 0$, $\forall p \in P_A$; 3) $M_0(r) \geq X_r(p)$, $\forall r \in P_R$, $\forall p \in P_A$, where X_r is r's minimal P-semiflow defined in Definition 4; 4) $\forall t \in T$ if $|\ t^\bullet \cap P_A\ | \geq 2$, $M_0(r) \geq \sum_{p \in P_A \cap t^\bullet} X_r(p)$; and 5) $\forall t \in T$ if $|{}^\bullet t \cap P_A\ | \geq 2$, $M_0(r) \geq \sum_{p \in P_A \cap {}^\bullet t} X_r(p)$.

Given an arbitrary marking $M \in R(N, M_0)$, t is M-process-enabled by $p \in P_A$ if $\exists p \in {}^\bullet t \cap \{P_0 \cup P_A\}$ such that $M(p) > 0$. Correspondingly, t is M-resource-enabled by $r \in P_R$ if $\forall r \in {}^\bullet t \cap P_R$ such that $M(r) \geq W(r,\ t)$. In the rest of this paper, unless otherwise stated, (N, M_0) is an acceptably marked AEMG, i.e., N is an AEMG and M_0 is an acceptable initial marking.

Definition 5 specifies an initial marking such that the whole system is meaningful, i.e., any type of jobs can be finished if all resources are dedicated to their production. If this condition is not satisfied, the production of certain products is doomed to failure owing to the insufficiency of resources. Deadlock-free control of such a system would be impossible due to such defect.

Definition 6: The set of holders of $r \in P_R$ is the support of a P-semiflow X_r without r, i.e., $H(r) = \|X(r)\| \setminus \{r\}$.

$H(r)$ contains only activity places due to $\|X(r)\| \cap P_R = \{r\}$. Let $S_R = S \cap P_R$, $S_A = S \cap P_A$, and $H_{S_R} = \cup_{r \in SR} H(r)$, where S is a siphon.

Definition 7: Token takers of S is the places that correspond to the holders of the resources in S but do not belong to S, i.e., $\widetilde{H}(S) = H_{S_R} \setminus S = (H_{S_R} \cap P_A) \setminus S_A$.

By assuming that $P_0 = \{p_1, p_9\}$, $P_{A_1} = \{p_2 - p_8\}$, $P_{A_2} = \{p_{10} - p_{12}\}$, $P_R = \{p_{13} - p_{17}\}$, $T_1 = \{t_1 - t_8\}$, and $T_2 = \{t_9 - t_{12}\}$, Fig. 3 shows the AEMG of an AMS with five resource types $\mathcal{R}_1 - \mathcal{R}_5$. They support two job types $\mathcal{J}_1$ and $\mathcal{J}_2$. Their capacities are $C_1 = C_2 = C_4 = C_5 = 2$, $C_3 = 1$. Job type $\mathcal{J}_1$ (resp., $\mathcal{J}_2$) is defined by the set of partially ordered job stages represented by $\{p_1 - p_8\}$ (resp., $\{p_9 - p_{12}\}$). A stage's resource requirement is given as a vector $a_{p_i} = (a_{i1}, a_{i2}, \ldots, a_{i|P_R|})^T$, where a_{ij} means the quantity of jth type of resources that is required by stage i. The resource requirements in this example are as follows: $a_{p_2} = [1\ 0\ 0\ 0\ 0]^T$, $a_{p_3} = [1\ 0\ 0\ 0\ 0]^T$, $a_{p_4} = [0\ 2\ 0\ 0\ 0]^T$, $a_{p_5} = [0\ 0\ 1\ 0\ 0]^T$, $a_{p_6} = [0\ 0\ 0\ 1\ 0]^T$, $a_{p_7} = [0\ 0\ 0\ 0\ 1]^T$, $a_{p_8} = [0\ 0\ 0\ 0\ 1]^T$, $a_{p_{10}} = [0\ 0\ 0\ 1\ 1]^T$, $a_{p_{11}} = [0\ 0\ 1\ 0\ 0]^T$, and $a_{p_{12}} = [1\ 1\ 0\ 0\ 0]^T$. The P-semiflows corresponding to resources are: $I_1 = p_2 + p_3 + p_{12} + p_{13}$, $I_2 = 2p_4 + p_{12} + p_{14}$, $I_3 = p_5 + p_{11} + p_{15}$, $I_4 = p_6 + p_{10} + p_{16}$, and $I_5 = p_7 + p_8 + p_{10} + p_{17}$.

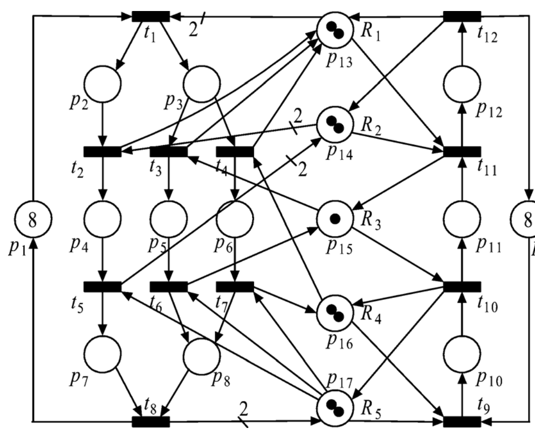


Fig. 3. An example AEMG.

IV. Liveness Analysis of AEMG

In ordinary nets, it is well known that their liveness is strongly related to their siphons. The nonexistence of empty siphons, at any marking $M \in R(N, M_0)$, is a necessary and sufficient condition for the liveness of some classes of Petri nets. In general nets, the concept of empty siphons is extended to the notion of undermarked siphons. A siphon S is undermarked at a marking M if $\forall t \in S^{\bullet}$, its firing is prevented by a set of resource places belonging to S. A deadlock marking in an ordinary net implies an empty siphon, while in a general net implies an undermarked one.

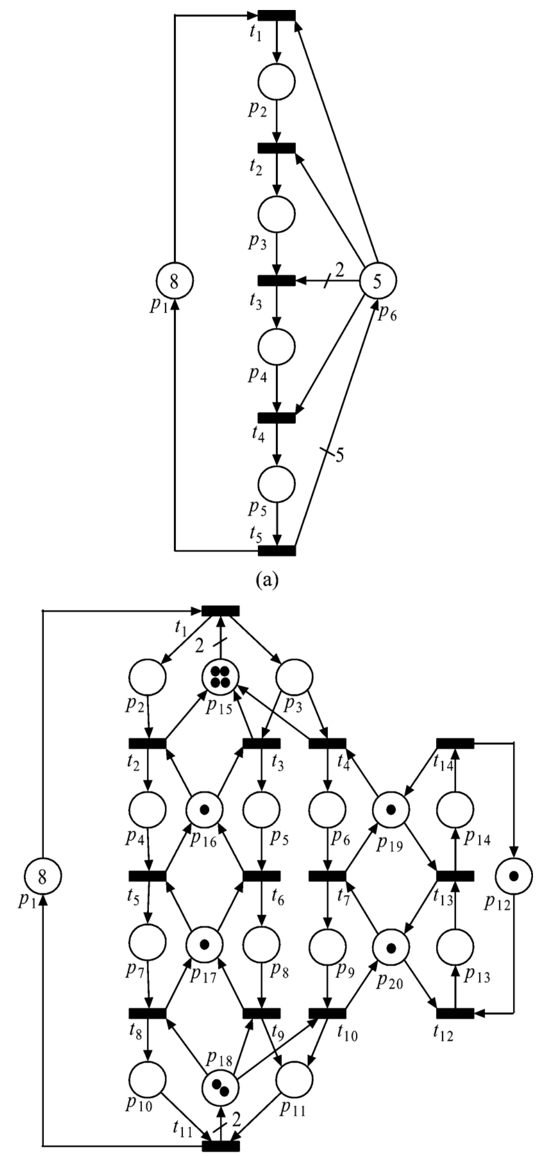
To establish the equivalence relation between undermarkedness of siphons and nonliveness of AEMG, a prerequisite is that each transition is of a chance to be enabled. This point can be obviously guaranteed since (N, M_0) is acceptably marked.

Suppose that a net is dead at a reachable marking. The initial marking is not reachable any more. Some transitions, not necessarily all, of some processes stagnate owing to the gradual emergence of a set of dead transitions. Fig. 4(a) shows an AEMG, which is acceptably marked. After a firing sequence $< t_1 t_1 >$, a marking $M' = 6p_1 + 2p_2 + 3p_6$ is reached. Although no siphon is undermarked, it is evident that t_4 is actually dead at M' since $\nexists M'' \in R(N,\ M')$ such that $M''[t_2\rangle$. On the basis of M', the occurrence of $< t_2 t_3 >$ can lead to a marking $M = 6p_1 + p_4$, under which all the transitions are disabled.

This scenario is somewhat similar to the results in [12]. We present it since the similar result in [12] cannot apply to AEMG. Our investigation shows that only partial transitions are dead. This is owing to the existence of flexible routes. We illustrate this scenario through the example in Fig. 4(b). After the firings of t_1, t_4, and t_{12}, in one of the two processes, t_4, t_7, and t_{10} become dead. However, all the other transitions in the same process remain live. In fact, this result not only holds for AEMG but also for the net class considered in [12].

The special structural properties of AEMG motivate us to prove that there exists an undermarked siphon S at $M \in R(N, M')$ when a transition is dead at M'. The basic idea behind this proof is to build an undermarked siphon that is composed of two different types of places: the undermarked resource places, i.e., S_R, and their unmarked holders, i.e., S_A.

Theorem 1: (N, M_0) is non-live iff there exist a marking $M \in R(N,\ M_0)$ and a siphon S such that $M(P_A) > 0$ and S is undermarked. Moreover, S satisfies: 1) $S_R = S \cap P_R = \{r \in P_R\colon \exists t \in r^{\bullet}$ such that $M(r) < W(r,t)$ and $M({}^{\bullet}t \cap P_A) > 0\} \neq \emptyset$ and 2) $S_A = S \cap P_A = \{p \in H_{S_R} \mid M(p) = 0\} \neq \emptyset$.



(a)

(b)

Fig. 4. Illustrative examples.

Proof: We first show the necessity part. Since (N, M_0) is non-live, there exists at least one dead transition t at a deadlock marking $M \in R(N, M_0)$. The structure of AEMG determines that a nonempty siphon S can only be built by $S = S_A \cup S_R$.

The statement that $S \neq \emptyset$ can be verified owing to the fact that $S_R \neq \emptyset$ and $S_A \neq \emptyset$.

To prove $S_R \neq \emptyset$, we must in advance show that $M(P_A) > 0$ if t is dead at M, which means $\exists p \in P_A$ such that $M(p) > 0$. To verify this point, we contradictorily assume $\forall p \in P_A$, $M(p) = 0$. According to Definitions 4 and 5, this implies $M = M_0$, under which no transition is dead. This contradicts our hypothesis that t is dead at M. Subsequently, without loss of generality, we suppose $p \in P_A$ such that $M(p) > 0$ and $t \in p^{\bullet}$. There must exist at least one $r \in P_R$ such that $t \in r^{\bullet}$ and $M(r) < W(r,t)$. Otherwise, t is fireable since it is both M-process-enabled and

M-resource-enabled. This also is in contradiction with M being dead. Therefore, $S_R \neq \emptyset$ thanks to the existence of p.

To prove $S_A \neq \emptyset$, we again suppose $S_A = \emptyset$ by contradiction. This implies $\forall p \in H_{S_R}, M(p) > 0$. Without loss of generality, consider a transition t such that $t^\bullet \cap S_R \neq \emptyset$ and ${}^\bullet t \cap S_R = \emptyset$. On account of the speciality of AEMG, such a transition trivially exists for each siphon S. Moreover, we have ${}^\bullet t \subseteq S_A \subseteq H_{S_R}$. Therefore, t is M-process-enabled since we assume $\forall p \in H_{S_R}, M(p) > 0$. Under this situation, the fireability of t can be established by considering the following two cases: 1) $t^\bullet \cap P_0 = \emptyset$, in this case, no resource can prevent its firing. Thus, t is fireable although it is only M-process-enabled and 2) $t^\bullet \cap P_0 \neq \emptyset$, then $\exists r' \in P_R$ thanks to the structure of AEMG. We immediately have $M(r') > W(r', t)$ since otherwise $r' \in S_R$ and $t \in S_R^\bullet$, which contradicts our hypothesis that ${}^\bullet t \cap S_R = \emptyset$. In the case, t is also fireable since it is both M-process-enabled and M-resource-enabled.

Next, we show that $S = S_A \cup S_R$ is a siphon by assuming $t \in {}^\bullet S$ is dead. The following cases should be considered.

First, we consider the case that $t \in {}^\bullet S_R$. Let $r \in t^\bullet \cap S_R$. The special structure of AEMG also implies that ${}^\bullet t \cap P_0 = \emptyset$ and $\exists p \in {}^\bullet t \cap P_A$ such that $p \in H(r)$. Two subcases should be further considered. (i) $M(p) = 0$. Then, $p \in S_A$ such that $t \in S_A^\bullet \subset S^\bullet$ and t is M-process-disabled by p. (ii) $M(p) > 0$. The assumption that t is dead leads to the fact that $\exists r' \in S_R$ such that $M(r') < W(r', t)$, $t \in r'^\bullet \subseteq S_R^\bullet \subseteq S^\bullet$ and t is M-resource-disabled by r'.

Second, we consider the case that $t \in {}^\bullet S_A$. Let $p \in P_A \cup P_0$. Two subcases should be further considered. (i) $\exists r \in S_R$ such that $t \in r^\bullet$ and $M(r) < W(r, t)$. Then $t \in S_R^\bullet \subseteq S^\bullet$. (ii) $\nexists r \in S_R$ such that $t \in r^\bullet$. Then $\exists p' \in S_A$ such that $M(p') = 0$ and $t \in p'^\bullet \subseteq S_A^\bullet \subseteq S^\bullet$. t is disabled by $p' \in S$. Thus, the nonliveness of (N, M_0) necessitates the existence of such a siphon.

To show the sufficiency part, suppose that (N, M_0) is live by contradiction. Since there exists a marking $M \in R(N, M_0)$ and a siphon $S \in \Pi$ such that $M(P_A) > 0$ and the firing of each M-process-enabled transition is prevented by a set of resource places belonging to S, $M_0 \notin R(N, M)$, which contradicts the assumption. Thus, it is not live. ■

V. A Deadlock Resolution Policy for AEMG

As stated in the previous section, the nonliveness of a Petri net is determined by the undermarkedness of some of its siphons. An intuitive idea is to detect these undermarked siphons and control them such that they are not undermarked. Compared with emptiness, it is quite difficult to probe these siphons together with their corresponding deadlock markings. Much work is performed to address such an issue. In [3], the concept of max-marked siphons is proposed. Given a Petri net, a siphon is said to be max-marked if $\forall M \in R(N, M_0)$, there exists $p \in S$ such that $M(p) > \max_{p^\bullet}$, where $\max_{p^\bullet} = \max_{t \in p^\bullet} W(p, t)$. A Petri net is said to be max-marked if every siphon is max-marked. Obviously, this novel concept greatly simplify the characterization of undermarkedness in general nets. However, it provides a sufficient conditions. This can greatly deteriorate the performance of a system since we need to impose a quite strong requirement upon the controlled system. To relax the condition in [3], the work in [37] develops another characterization. More importantly, a set of mathematical formulation is developed to derive siphons. However, such formulation involves too many variables due to the complexity of undermarkedness in general nets.

As shown in [12], emptiness of all siphons is a necessary and sufficient characterization for certain special classes of ordinary nets. In fact, this result can be easily extended to PT-ordinary nets in which $\forall p \in P, t \in T, W(p, t) \leq 1$. Compared with undermarkedness in general nets, it is quite easy to characterize emptiness in PT-ordinary nets. Following the research in [28], we convert a general AEMG to its corresponding PT-ordinary counterpart. After this manipulation, we can convert all the undermarked siphons to emptied ones. The latter ones can be identified more easily than the former.

A. Transformation of AEMG to PT-ordinary Nets

The transformation from general nets to the PT-ordinary ones is initially investigated in [28]. To facilitate our approach in this paper, we provide it in a simple way. In [28], if $\{p_i, p_j\} \subseteq {}^\bullet t$, where $i, j \subseteq \mathbb{N}^+$ and $i \neq j$, $W(p_i, t) > 1$ and $W(p_j, t) > 1$, the two weighted arcs (p_i, t) and (p_j, t) can be split in an interconnected way. However, our method can split each arc in an independent way [18].

Let (N, M_0) where $N = (P, T, F, W)$, be a Petri net system, $\exists t_j \in T, p \in P, f \in F$ such that $f = (p, t_j)$ and $W(f) = m (m \geq 2)$. We show the algorithm as follows.

Algorithm 1 [18]: PT-Transformation of (p, t_j)

Input: An arc $f = (p, t_j)$ with $W(f) = m$
Output: A subnet (P_X, T_X, F_X, W_X)
Begin
1. $i = 0, t_{j,0} = t_j, P_X = \emptyset, T_X = \emptyset, F_X = f, W_X = \emptyset$;
2. **while** $(i < m)$ **do**
   begin
3. $i = i + 1$;
4. $t_{j,i}$ and $p_{j,i}$ are added such that ${}^\bullet p_{j,i} = \{t_{j,i}\}$, $p_{j,i}^\bullet = \{t_{j,i-1}\}$, and ${}^\bullet t_{j,i} = \{p\}$;
5. $P_X = P_X \cup \{p_{j,i}\}$, $T_X = T_X \cup \{t_{j,i}\}$, $F_X = F_X \cup \{(p, t_{j,i})\} \cup \{(t_{j,i}, p_{j,i})\} \cup \{(p_{j,i}, t_{j,i-1})\}$;
   end
6. $W_X : F_X \to \{1\}$;
7. $t_{j,0} = t$;

End

Another issue is that new siphons might be introduced in the PT-ordinary net transformed from an original one. However, we can prove that they can be easily identified. We can mark them out and need not to consider them during siphon-computation to be shown next. We show such results through the following two propositions.

Lemma 1: Let N be an AEMG and $\widetilde{N}$ be its corresponding PT-ordinary one. S is a siphon in $\widetilde{N}$ if so in N.

Proof: Let S be a siphon in N. In N, we have ${}^\bullet S \subseteq S^\bullet$. We suppose that in $\widetilde{N}$, the preset of S is ${}^\bullet\widetilde{S}$ and the postset of S is $\widetilde{S}^\bullet$. Suppose the $p^\bullet$ and ${}^\bullet p$ represent the postset and preset of each place in N. Correspondingly, suppose that $\widetilde{p}^\bullet$ and ${}^\bullet\widetilde{p}$ represent the postset and preset of each place in $\widetilde{N}$. For the procedure of Algorithm 1, we know that for $\forall p \in P$, $\widetilde{p}^\bullet = {}^\bullet p$ whereas $\widetilde{p}^\bullet \supseteq {}^\bullet p$. As a result, we have ${}^\bullet\widetilde{S} = {}^\bullet S$ while $\widetilde{S}^\bullet \supseteq S^\bullet$. Therefore, we have ${}^\bullet\widetilde{S} \subseteq S^\bullet$. This implies that S remains to be a siphon in $\widetilde{N}$ if so in N. ∎

A siphon in N remains to be a siphon but not *vice versa*. This is owing to that new siphons might be introduced during the transformation process. We show next that these newly introduced siphons can be quite easily identified.

Lemma 2: Let $\|X_r\|$ be the support of a P-semiflow, which corresponds to a resource $r \in P_R$ in N. $\|X_r\|$ becomes a newly induced siphon in $\widetilde{N}$ if there exists a weighted arc from r to a transition t.

Proof: In N, we have ${}^\bullet\|X_r\| = \|X_r\|^\bullet$ immediately. This is equivalent to ${}^\bullet\{\|X_r\|\backslash\{r\}\} \cup {}^\bullet r = \{\|X_r\|\backslash\{r\}\}^\bullet \cup r^\bullet$. Let ${}^\bullet\widetilde{\{\|X_r\|\backslash\{r\}\}}$ and $\widetilde{\{\|X_r\|\backslash\{r\}\}}^\bullet$ denote the preset and postset of $\|X_r\|\backslash\{r\}$ in $\widetilde{N}$, respectively. After the transformation from N to $\widetilde{N}$, we know that ${}^\bullet\widetilde{\{\|X_r\|\backslash\{r\}\}} = {}^\bullet\{\|X_r\|\backslash\{r\}\}$, ${}^\bullet\widetilde{r} = {}^\bullet r$, $\widetilde{\{\|X_r\|\backslash\{r\}\}}^\bullet = \{\|X_r\|\backslash\{r\}\}^\bullet$. However, $\widetilde{r}^\bullet = r^\bullet$. As a result, we have ${}^\bullet\|X\| \subset \|X\|^\bullet$. This implies that $\|X_r\|$ becomes a siphon in $\widetilde{N}$. ∎

Theorem 2: A siphon in $\widetilde{N}$ corresponds to a minimal siphon in N or the support of a P-semiflow $\|X_r\|$ in N.

Proof: By contradiction, we suppose that S is a newly generated siphon in $\widetilde{N}$. Two cases must be considered independently.

First, S does not contain any newly added places. This means that ${}^\bullet S \subseteq S^\bullet$ holds in $\widetilde{N}$. According to the transformation process in Algorithm 1 and Lemma 1, ${}^\bullet S \subseteq S^\bullet$ must also hold in N. This contradicts the assumption that S is a newly introduced siphon.

Second, S contains the newly introduced places. This makes it different from the ones in N. According to the procedure in Algorithm 1, to maintain the relationship ${}^\bullet S \subseteq S^\bullet$ after such an involvement, we also have to involve the corresponding resource of such newly added places. To involve the resource, we also have to involve the operation place of such a resource. This implies we have to involve $\|X_r\|$. According to Lemma 2, $\|X_r\|$ has already become a siphon in $\widetilde{N}$. This means S cannot be a strict minimal siphon as we assume. ∎

Corollary 1: The number of newly induced strict and minimal siphon in $\widetilde{N}$ is bounded by the number of resource places, i.e., $\mid P_R \mid$.

Proof: According to Lemma 1, this is obvious since one resource can only introduce one siphon. ∎

Theorem 2 and Corollary 1 are of significance since we can identify and remove these newly induced siphons before the computation of all other siphons. This is useful since we are concerned about only the siphons in N.

Fig. 5 shows the corresponding PT-ordinary net of Fig. 3. It can be easily verified that p_{13} and p_{14} introduce two new siphons, i.e., $\{p_2, p_3, p_{12}, p_{13}\}$ and $\{p_4, p_{12}, p_{14}\}$. They should be identified and removed before the computation of other siphons.

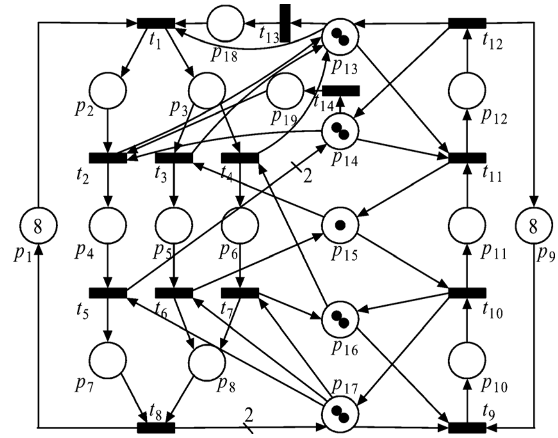


Fig. 5. The PT-ordinary AEMG of the net in Fig. 3.

B. Algebraic Characterization of Empty Siphons

It is an NP-hard problem to enumerate all the siphons. Moreover, the research in [11], [29], and [30] shows that it is not necessary to identify and control all these siphons. Most of them can be implicitly controlled after a few ones are supervised. Thus, detect one siphon at each time and control it immediately.

In ordinary Petri nets, the equivalence relationship holds between undermarkedness and emptiness of each siphon. Thus it is quite easy to derive siphons in such nets. The general idea is to characterize a siphon with a vector. Then we can evaluate whether the algebraic sum of markings can reach zero. Such an algorithm is initially investigated in [9], [37]. However, their research involves many variables, which significantly reduces the computational efficiency. In [4], a new algorithm is developed to solve it. However, it can only be applied to ordinary nets. It cannot detect the siphons in general nets.

In the previous subsection, this work shows that any general nets can be easily converted into PT-ordinary ones. This brings about a chance to apply the algorithm in [4] to our concerned systems. Another noticeable issue is that our transformation may introduce new siphons. These siphons should be removed from being detected by the algorithm. This is because we will eventually make an inverse conversion such that the controlled system remains to be a general net. As a result, we need to improve the algorithm of [4] significantly. Finally, to determine an empty siphon, we can utilize the following integer programming (IP) formulation:

$$\min \ \mathbf{1}^T \cdot \lambda$$

subject to

$$K_1 \cdot [N^-]^T \cdot \lambda \geq [N^+]^T \cdot \lambda \tag{1}$$

$$K_2 \cdot \lambda + M \leq K_2 \cdot \mathbf{1} \tag{2}$$

$$\mathbf{1}^T \cdot \lambda \geq 1 \tag{3}$$

$$\sum_{p \in \|X_r\|} \lambda(p) \leq \mid \|X_r\| \mid -1 \tag{4}$$

$$M = M_0 + [N] \cdot y \tag{5}$$

where $\lambda(p) \in \{0,1\}$, $M \in \mathbb{N}^{|P|}$, $y \in \mathbb{N}^{|T|}$, $K_1 = \max_{t \in T}[N^+](\cdot, t) \cdot \mathbf{1}$, and $K_2 = \max_{p \in P} M(p)$.

To facilitate the understanding of these equations, we make the following comments.

Equation (1) is equivalent to the inequality: $\forall t \in T$, $sign([N^-]^T \cdot \lambda) \geq sign([N^+]^T \cdot \lambda)$, where $sign(x)$ is a vector stating that the ith component of x is 1 (resp., 0, -1) if it is positive (resp., 0, negative). The physical meaning can be interpreted as follows. For a transition t, $sign([N^+]^T(\cdot, t) \cdot \lambda)$ means that t is the ingoing transition of a siphon $\|\lambda\|$. Correspondingly, $sign([N^-]^T(\cdot, t) \cdot \lambda)$ means that t is its outgoing transition. The inequality means that t must be the outgoing transition if it is the ingoing one of a siphon. Apparently, this implies that $\|\lambda\|$ is a siphon.

Equation (2) is equivalent to $\lambda^T \cdot M = 0$. As K_2 is sufficiently larger than $M(p), \forall p \in P, \forall M \in R(N, M_0)$. If $\lambda(p) = 1$, we conclude that M must be equivalent to 0; otherwise, it can lead to $K_2 \cdot \lambda + M \geq K_2 \cdot \mathbf{1}$.

Equation (3) implies that a siphon must contain at least one place. In other words, it cannot be empty.

Equation (4) is used to remove the newly introduced siphons after the transformation from a general net to a PT-ordinary one.

Equation (5) represents the reachable space. Spurious solution may be involved as some states can be the solutions whereas they can never be reached in a net's evolution. How to avoid the emergence of this phenomenon requires our further investigation.

The above formulation is inspired by the approach in [4]. However, it has distinct difference since the one in [4] is developed to derive only the dead markings. It requires that for each transition t there must exist at least one ingoing place that is empty at a dead marking M. This means that all the transitions are dead. We call it a total deadlock. However, to attain a live Petri net, we also need to identify the so-called partial deadlocks at which some transitions are dead while others are live. Obviously, their approach cannot identify the partial deadlock markings. We improve their formulation by focusing on the identification of empty siphons. As the emptiness of siphons is equivalent to the deadlock in AEMG, our improved formulation can identify both the total and partial deadlocks.

The effectiveness and efficiency of IP formulations are explained as follows. First, they can avoid the explicit enumeration of siphons in AEMG, which is well known to have exponential computational complexity. Second, they only concern about the empty siphons instead of the undermarked ones. The latter are of high difficulty to detect because of the difference in the least number of tokens in different undermarked siphons to mark them. Third, they involve less variables since it is easy to describe an empty siphon compared with an undermarked one. As for an IP problem, its efficiency largely depends on the number of its involved variables. Also, all the existing papers have supported that the use of IP outperforms the siphon enumeration and related methods for the same problem [9], [16], [33], [37].

C. Liveness Enforcement Through Linear Marking Constraint

As stated in the above section, the nonliveness of an AEMG can be attributed to the existence of an undermarked siphon at a deadlock marking $M_D \in R(N,\ M_0)$. Therefore, the basic idea to enforce the liveness of an AEMG is to prevent siphons from becoming undermarked by the addition of monitors.

Our approach is to identify these siphons and the representative markings under which these siphons are undermarked. As observed, the IP formulation in the above section can provide both of these two items simultaneously.

Clearly, the IP formulation contains three unknown vectors, i.e., λ, M, and y. Once a feasible solution is found, $\|\lambda\|$ represents an undermarked siphon, i.e., S; M represents the deadlock marking under which S is undermarked; and y corresponds to the firing count vector such that M is reached from M_0. Take a siphon $S = \{p_7, p_{11}, p_{15}, p_{17}\}$ for example. According to the IP formulation, when S is emptied, i.e., $M(S) = 0$, we have $M = 6p_1 + p_2 + p_3 + p_4 + p_5 + 6p_9 + 2p_{10}$. Moreover, $y = 2t_1 + t_2 + t_3 + 2t_9 + 2t_{13} + t_{14}$, which means that for the system to evolve from M_0 to M, transitions t_1, t_2, t_3, t_9, t_{13}, and t_{14} fires 2, 1, 1, 2, 2, and 1 time(s), respectively. Such a result is informative to identify deadlock issues and inspire a way to handle these deadlocks.

From the process perspective, we can ascertain that deadlock occurs because many processes are acquiring the resources originally in a siphon. Take $S = \{p_7, p_{11}, p_{15}, p_{17}\}$ in Fig. 3 for example again. There are two types of resources r_3 and r_5, which are denoted by p_{15} and p_{17}, respectively. At the deadlock marking M_D, 1 copy of r_1 is occupied by p_5 and two copies of r_2 are occupied by p_{10}. One resource is available for the further proceeding of other process stages involving in this siphon. To resolve such deadlocks, it is necessary to limit the number of resources being acquired by these processes. Mathematically, this can be implemented by a monitor by assuming $M(\overline{S}) \leq M_D(\overline{S}) - 1$, where $\overline{S} = \bigcup_{r \in S} \|X_r\| - S$ means the complementary set of S and M_D is a deadlock marking. Apparently, this requirement can be realized by a general exclusive mutual constraint (GMEC) which produces a monitor p_c. Assume l is a $|\ P\ |$-dimensional vector such that $l(p) = 1$, $p \in P_A \cap \overline{S}$; otherwise, $l(p) = 0$. Meanwhile, $b = M_D(\overline{S}) - 1$. This GMEC, denoted by a pair (l, b), can be realized by p_c as follows:

$$\begin{aligned}[N_{p_c}] &= -\ l^T \cdot [N], \\ M_0(p_c) &= b.\end{aligned}$$

To avoid the generation of new siphons, (l, b) must be substituted such that the outgoing arcs of each monitor are connected to the source transitions, i.e., $t \in P_0^{\bullet} \cap T$. To do so, we assume only source transitions are controllable, i.e., $T \backslash \{P^{\bullet} \cap T\} \subseteq T_{uc}$, where T_{uc} means the uncontrollable transitions that cannot be prevented from being fired by a supervisor. (l, b) can be substituted by a more restrictive pair $(\widehat{l}, \widehat{b})$ such that $\widehat{l}^T \cdot M < \widehat{b} \Rightarrow l^T \cdot M < b$, where $\widehat{l} = \zeta + a \cdot l$ and $\widehat{b} = a \cdot\ (b + 1) - 1$, where ζ is a $|P|$-dimensional positive vector and a is a positive scalar. This substitution also accelerates the iteration convergence and leads to a structurally simple supervisor with less computation.

TABLE I
GENERATED MONITORS FOR THE NET IN FIG. 3

i	${}^{\bullet}p_{c_i}$	$p_{c_i}^{\bullet}$	$M(p_{c_i})$
1	$\{t_8\}$	$\{t_1\}$	1
2	$\{t_8\}$	$\{t_1\}$	1
3	$\{t_2, t_8, t_{10}\}$	$\{2t_1, t_7, t_9\}$	2
4	$\{t_4, t_6, t_8, t_{10}\}$	$\{2t_1, t_9\}$	2
5	$\{t_6, t_7, t_{11}\}$	$\{t_1, t_9\}$	2
6	$\{t_5, t_8, t_{11}\}$	$\{2t_1, t_9\}$	2

Algorithm 2: Liveness-Enforcing Supervisor Synthesis

Input: A plant or uncontrolled AEMG net (N_0^*, M_0^*)
Output: Control places $\{p_{c_i}\}$ with $\{M(p_{c_i})\}$

1. Let $i = 0$. Transform (N_0^*, M_0^*) into a PT-ordinary one (N_i, M_i) with n places and m transitions;
2. Check whether an empty siphon exists in (N_i, M_i). If "yes," go to Step 3. If "not," exit with outputs;
3. Find an empty siphon S_i and dead marking M_{D_i} under which S_i is emptied. Derive $M_{D_i}(\overline{S}_i) = \sum_{p \in S_i} M_{D_i}(p)$;
4. Establish an inequality $l_{S_i}^T \cdot M \leq b_i$ where $l_{S_i}(p) = 1$ if $p \in P_A \cap \overline{S}_i$; otherwise, $l_{S_i}(p) = 0$. $b_i = M_{D_i}(\overline{S}_i) - 1$;
5. Adjust $l_{S_i}^T \cdot M \leq b_i$ so as to obtain a more restrictive inequality $\widehat{l}_{S_i}^T \cdot M \leq \widehat{b}_i$, which ensures that no new siphons can be introduced;
6. Control $\widehat{l}_{S_i}^T \cdot M \leq \widehat{b}_i$ with $p_{c_{i+1}}$ (or p_{n+1}) such that $[N_{p_{c_{i+1}}}] = -\widehat{l}_{S_i}^T \cdot [N_i]$, and $M(p_{c_{i+1}}) = \widehat{b}_i - \widehat{l}_{S_i}^T \cdot M_i$. Establish a new net model (N_{i+1}, M_{i+1}) such that $[N_{i+1}] = [[N_i]^T \mid [N_{p_{c_{i+1}}}]^T]^T$ and $M_{i+1} = [M_i^T \ M(p_{c_{i+1}})]^T$;
7. Let $i = i + 1$, $n = n + 1$. Go to Step 2.

After an AEMG is controlled, there is a reverse operation from a PT-ordinary net to its general counterpart. Owing to its simplicity, the details are omitted. Now, we apply the above algorithm to the Petri net model in Fig. 3, whose PT-ordinary version is shown in Fig. 5.

First, the IP formulation leads to a siphon $S_0 = \{p_7, p_{10}, p_{17}\}$ with $M_{D_0} = 5p_1 + 2p_2 + p_4 + p_6 + 2p_8 + 7p_9 + p_{11} + p_{16}$. As $\overline{S}_0 = \{p_8\}$, we have $M_{D_0}(\overline{S}_0) = 2$. Moreover, $l_{S_0} = p_8$ and $\widehat{l}_{S_0} = p_3 + p_5 + p_6 + p_8$. Hence, we have $[N_{p_{c_1}}] = -t_1 + t_8$, $M(p_{c_1}) = 1$, $[N_1] = [[N_0]^T \mid [N_{p_{c_1}}]^T]^T$, and $M_1 = [M_0^T \ M(p_{c_1})]^T$.

In the similar way, we can obtain $S_1 = \{p_8, p_{10}, p_{17}\}$, $S_2 = \{p_7, p_{11}, p_{15}, p_{17}\}$, $S_3 = \{p_8, p_{11}, p_{15}, p_{17}\}$, $S_4 = \{p_7, p_{12}, p_{14}, p_{15}, p_{17}\}$, and $S_5 = \{p_8, p_{12}, p_{14}, p_{15}, p_{17}\}$. Their corresponding monitors can be calculated in a similar way as we do in the first step. Finally, these monitors are shown in Table I. Further analysis shows that the original Petri net produces 174 states with 10 deadlock states. With these monitors, the controlled system is live with 40 states. Clearly, the controllers are not maximally permissive, implying that more research must be done to achieve maximal permissiveness.

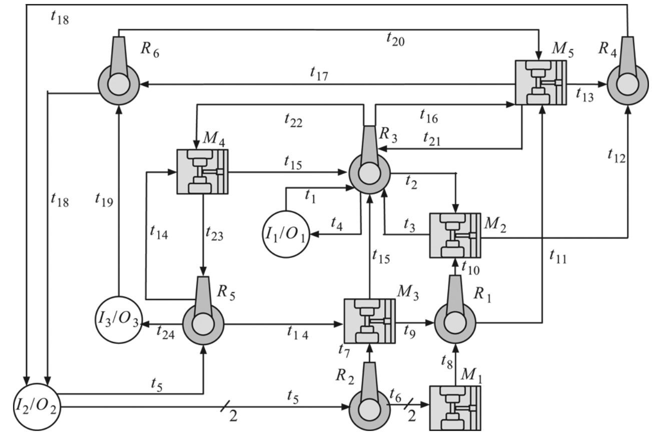


Fig. 6. The block diagram of an example AMS.

It is of importance to evaluate the system performance after liveness-enforcing supervision. This approach is always realized by introducing either deterministic and stochastic time information to each transitions or places. An additional control mechanism is introduced to keep a balance between the control cost and the system performance. We believe that this work offers a good basis for such further exploration. Due to the space limit, we have to leave this interesting topic as our future work.

VI. ILLUSTRATIVE EXAMPLE

Fig. 6 shows the block diagram of an AMS where three product types, i.e., $\mathcal{J}_1 - \mathcal{J}_3$, are manufactured. Obviously, this system is composed of six robots $\mathbf{R}_1 - \mathbf{R}_6$ and five machines $\mathbf{M}_1 - \mathbf{M}_5$. Each of $\mathbf{R}_1$ and $\mathbf{R}_3 - \mathbf{R}_6$ can hold one product only, while $\mathbf{R}_2$ can hold two. For machines $\mathbf{M}_2 - \mathbf{M}_5$, each of them can deal with one product at a time. $\mathbf{M}_1$ can deal with two. There are three loading buffers $\mathbf{I}_1 - \mathbf{I}_3$ and three unloading buffers $\mathbf{O}_1 - \mathbf{O}_3$ to load and unload the AMS. The action area for $\mathbf{R}_1$ is $\mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_5$; for robot $\mathbf{R}_2$ is $\mathbf{I}_2, \mathbf{M}_1$, and $\mathbf{M}_3$; for robot $\mathbf{R}_3$ is $\mathbf{I}_1, \mathbf{M}_2, \mathbf{M}_3, \mathbf{M}_4$, and $\mathbf{M}_5$; for robot $\mathbf{R}_4$ is $\mathbf{O}_2$, $\mathbf{M}_2$, and $\mathbf{M}_5$; for robot $\mathbf{R}_5$ is $\mathbf{I}_2, \mathbf{O}_3, \mathbf{M}_3$, and $\mathbf{M}_4$. By these resources, products $\mathcal{J}_1 - \mathcal{J}_3$ can be concurrently manufactured. Every arriving raw product belongs to one of these three products. According to the predefined routes, a raw product $\mathcal{J}_1$ is taken from $\mathbf{I}_1$ by $\mathbf{R}_3$. After being processed by $\mathbf{M}_2$, it is moved to $\mathbf{O}_1$ by $\mathbf{R}_3$. A raw product $\mathcal{J}_2$ is divided into two halves. Each one can be independently processed before they are combined together. One half is taken from $\mathbf{I}_2$ by $\mathbf{R}_2$. Subsequently, two flexible routes are available for its further treatment. First, it is manufactured in $\mathbf{M}_1$ and then moved to $\mathbf{M}_2$ by $\mathbf{R}_1$. Second, it is manufactured in $\mathbf{M}_3$ and then moved to $\mathbf{M}_5$ by $\mathbf{R}_1$. After its process in either $\mathbf{M}_2$ or $\mathbf{M}_5$, this half is finished and moved to $\mathbf{O}_2$ by $\mathbf{R}_4$. Another half is taken from $\mathbf{I}_2$ by $\mathbf{R}_5$. Then, it is processed in $\mathbf{M}_3$ and $\mathbf{M}_4$. After being processed by $\mathbf{M}_3$ and $\mathbf{M}_4$, it is moved to $\mathbf{M}_5$ by $\mathbf{R}_3$. After being processed by $\mathbf{M}_5$, it is moved to $\mathbf{O}_2$ by $\mathbf{R}_6$. At $\mathbf{O}_2$, an assembly operation is made so that a final product of $\mathcal{J}_2$ is obtained. A raw product of $\mathcal{J}_3$ is taken from $\mathbf{I}_3$ by $\mathbf{R}_6$. After being processed by $\mathbf{M}_5$, it is moved to $\mathbf{M}_4$ by $\mathbf{R}_3$. After being processed by $\mathbf{M}_4$, it is moved to $\mathbf{O}_3$ by $\mathbf{R}_5$. Noticeably, two copies of resources are required when the process of $\mathcal{J}_2$ is on either $\mathbf{M}_1$ or $\mathbf{R}_2$. In Fig. 6, the labeled

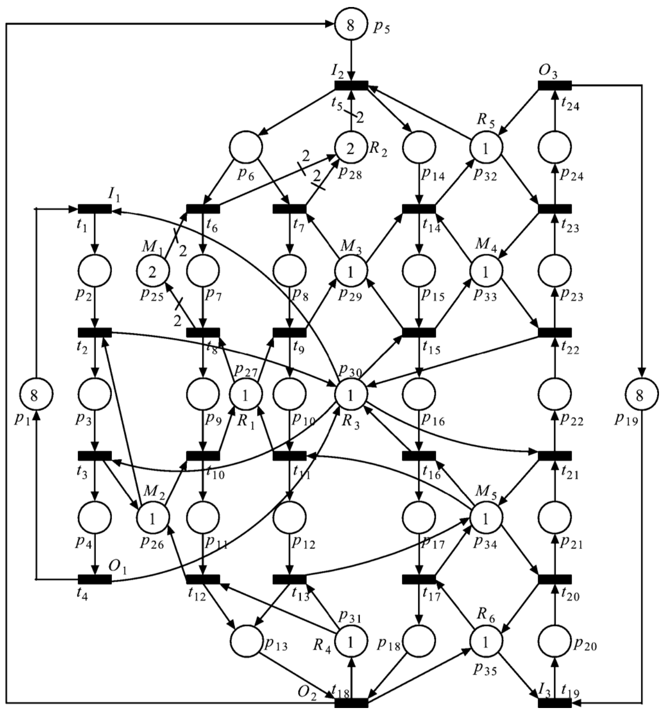


Fig. 7. The Petri net model of the AMS in Fig. 6.

notation, namely $t_1 - t_{24}$, beside each arc and leading to a resource $\mathbf{I}_i/\mathbf{O}_i$, $\mathbf{R}_i$, or $\mathbf{M}_i$ indicates that $\mathbf{I}_i/\mathbf{O}_i$, $\mathbf{R}_i$, or $\mathbf{M}_i$ is required in order to trigger the corresponding event represented by t_j. The required quantity of a resource is represented by the weight of the ingoing arc.

Fig. 7 shows the net model of this AMS, which allows multiple resource acquisitions, assembly operations, and flexible routes. The system is an AEMG where $P_0 = \{p_1, p_5, p_{19}\}$, $P_{A_1} = \{p_2 - p_4\}$, $P_{A_2} = \{p_6 - p_{18}\}$, $P_{A_3} = \{p_{20} - p_{24}\}$, $P_R = \{p_{25} - p_{35}\}$, $t_0^1 = t_1$, $t_0^2 = t_5$, and $t_0^3 = t_{19}$. Places $p_{25} - p_{35}$ denote $\mathbf{M}_1$, $\mathbf{M}_2$, $\mathbf{R}_1$, $\mathbf{R}_2$, $\mathbf{M}_3$, $\mathbf{R}_3$, $\mathbf{R}_4$, $\mathbf{R}_5$, $\mathbf{M}_4$, $\mathbf{M}_5$, and $\mathbf{R}_6$, respectively. Initially, it is assumed that there are no parts in the system. $M(p_1) = M(p_5) = M(p_{19}) = 8$ represents that the maximum number of job instances that are allowed for part types $\mathcal{J}_1 - \mathcal{J}_3$ at a time, respectively.

This net is deadlock-prone since some siphons can be eventually undermarked during the evolution. Our analysis shows that there are 49 siphons, $S_0 = \{p_{15}, p_{24}, p_{32}, p_{33}\}$, $S_1 = \{p_{18}, p_{21}, p_{31}, p_{34}, p_{35}\}$, $S_2 = \{p_{17}, p_{18}, p_{21}, p_{31}, p_{34}\}$, $S_3 = \{p_{12}, p_{18}, p_{21}, p_{34}, p_{35}\}$, $S_4 = \{p_9, p_{11} - p_{13}, p_{21}, p_{27}, p_{34}, p_{35}\}$, $S_5 = \{p_5 - p_7, p_9, p_{11} - p_{13}, p_{21}, p_{34}, p_{35}\}$, $S_6 = \{p_4, p_{18}, p_{24}, p_{26}, p_{30}, p_{31} - p_{35}\}$, $S_7 = \{p_4, p_{17}, p_{18}, p_{24}, p_{26}, p_{30} - p_{34}\}$, $S_8 = \{p_4, p_{18}, p_{23}, p_{26}, p_{30}, p_{31}, p_{33} - p_{35}\}$, $S_9 = \{p_4, p_{17}, p_{18}, p_{23}, p_{26}, p_{30}, p_{31}, p_{33}, p_{34}\}$, $S_{10} = \{p_4, p_{18}, p_{22}, p_{26}, p_{30}, p_{31}, p_{34}, p_{35}\}$, $S_{11} = \{p_4, p_{17}, p_{18}, p_{22}, p_{26}, p_{30}, p_{31}, p_{34}\}$, $S_{12} = \{p_4, p_{16} - p_{18}, p_{24}, p_{26}, p_{30} - p_{33}\}$, $S_{13} = \{p_4, p_{16} - p_{18}, p_{23}, p_{26}, p_{30}, p_{31}, p_{33}\}$, $S_{14} = \{p_4, p_{16} - p_{18}, p_{22}, p_{26}, p_{30}, p_{31}\}$, $S_{15} = \{p_3, p_{10}, p_{15}, p_{16}, p_{17}, p_{18}, p_{26}, p_{27}, p_{29}, p_{31}\}$, $S_{16} = \{p_4, p_{11} - p_{13}, p_{24}, p_{26}, p_{30}, p_{32} - p_{35}\}$, $S_{17} = \{p_4, p_{11} - p_{13}, p_{23}, p_{26}, p_{30}, p_{33} - p_{35}\}$, $S_{18} = \{p_2, p_4, p_6, p_7, p_9, p_{11} - p_{13}, p_{24}, p_{30}, p_{32}, p_{33} - p_{35}\}$, $S_{19} = \{p_4, p_{11} - p_{13}, p_{22}, p_{26}, p_{30}, p_{34}, p_{35}\}$, $S_{20} = \{p_2, p_4, p_9, p_{11} - p_{13}, p_{22}, p_{27}, p_{30}, p_{34}, p_{35}\}$, $S_{21} = \{p_4, p_{11}, p_{12}, p_{18}, p_{24}, p_{26}, p_{30}, p_{32} - p_{35}\}$, $S_{22} = \{p_4, p_{11}, p_{12}, p_{17}, p_{24}, p_{26}, p_{30}, p_{32} - p_{34}\}$, $S_{23} = \{p_4, p_{11}, p_{12}, p_{18}, p_{23}, p_{26}, p_{30}, p_{33} - p_{35}\}$, $S_{24} = \{p_4, p_{11}, p_{12}, p_{17}, p_{23}, p_{26}, p_{30}, p_{33}, p_{34}\}$, $S_{25} = \{p_4, p_{11}, p_{12}, p_{18}, p_{22}, p_{26}, p_{30}, p_{34}, p_{35}\}$, $S_{26} = \{p_4, p_{11}, p_{12}, p_{17}, p_{22}, p_{26}, p_{30}, p_{34}\}$, $S_{27} = \{p_4, p_{11}, p_{16}, p_{24}, p_{26}, p_{30}, p_{32}, p_{33}\}$, $S_{28} = \{p_4, p_{11}, p_{16}, p_{23}, p_{26}, p_{30}, p_{33}\}$, $S_{29} = \{p_4, p_{11}, p_{16}, p_{22}, p_{26}, p_{30}\}$, $S_{30} = \{p_2, p_4, p_{18}, p_{24}, p_{30} - p_{35}\}$, $S_{31} = \{p_2, p_4, p_{17}, p_{18}, p_{24}, p_{30} - p_{34}\}$, $S_{32} = \{p_2, p_4, p_{12}, p_{18}, p_{24}, p_{30}, p_{32} - p_{35}\}$, $S_{33} = \{p_2, p_4, p_9, p_{11} - p_{13}, p_{24}, p_{27}, p_{30}, p_{32} - p_{35}\}$, $S_{34} = \{p_2, p_4, p_{12}, p_{17}, p_{24}, p_{30}, p_{32} - p_{34}\}$, $S_{35} = \{p_2, p_4, p_{18}, p_{23}, p_{30}, p_{31}, p_{33} - p_{35}\}$, $S_{36} = \{p_2, p_4, p_{17}, p_{18}, p_{23}, p_{30}, p_{31}, p_{33}, p_{34}\}$, $S_{37} = \{p_2, p_4, p_{12}, p_{18}, p_{23}, p_{30}, p_{33} - p_{35}\}$, $S_{38} = \{p_2, p_4, p_9, p_{11} - p_{13}, p_{23}, p_{27}, p_{30}, p_{33} - p_{35}\}$, $S_{39} = \{p_2, p_4 - p_7, p_9, p_{11} - p_{13}, p_{23}, p_{30}, p_{33} - p_{35}\}$, $S_{40} = \{p_2, p_4, p_{12}, p_{17}, p_{23}, p_{30}, p_{33}, p_{34}\}$, $S_{41} = \{p_2, p_4, p_{18}, p_{22}, p_{30}, p_{31}, p_{34}, p_{35}\}$, $S_{42} = \{p_2, p_4, p_{17}, p_{18}, p_{22}, p_{30}, p_{31}, p_{34}\}$, $S_{43} = \{p_2, p_4, p_{12}, p_{18}, p_{22}, p_{30}, p_{34}, p_{35}\}$, $S_{44} = \{p_2, p_4, p_6 - p_9, p_{11} - p_{13}, p_{22}, p_{24}, p_{29}, p_{30}, p_{32}, p_{34}, p_{35}\}$, $S_{45} = \{p_2, p_4 - p_7, p_9, p_{11} - p_{13}, p_{22}, p_{30}, p_{34}, p_{35}\}$, $S_{46} = \{p_2, p_4, p_{12}, p_{17}, p_{22}, p_{30}, p_{34}\}$, $S_{47} = \{p_2, p_4, p_{16}, p_{24}, p_{30}, p_{32}, p_{33}\}$, and $S_{48} = \{p_2, p_4, p_{16}, p_{23}, p_{30}, p_{33}\}$.

Next, our proposed algorithm is used to iteratively derive the empty siphons and make them controlled until a live system results. Also, the remaining part is changed to be a PT-ordinary one as done in Fig. 5.

First, from the IP formulation, we obtain $S_0 = \{p_{15}, p_{24}, p_{32}, p_{33}\}$ with $M_{D_0} = 8p_1 + 6p_5 + p_9 + p_{13} + p_{14} + p_{16} + 7p_{19} + p_{23} + 2p_{25} + p_{26} + 3p_{28} + p_{29} + p_{34} + p_{35} + 8p_{37}$ and $\overline{S}_0 = \{p_{14}, p_{23}\}$, we have $M_{D_0}(\overline{S}_0) = 2$. Moreover, $l_{S_0} = p_{14} + p_{23}$ and $\widehat{l}_{S_0} = p_{14} + p_{20} + p_{21} + p_{22} + p_{23}$. Hence, we have $[N_{p_{c_1}}] = +t_5 - t_{14} + t_{19} - t_{23}$, $M(p_{c_1}) = 1$, $[N_1] = [[N_0]^T \mid [N_{p_{c_1}}]^T]^T$, and $M_1 = [M_0^T\ M(p_{c_1})]^T$.

Second, from the IP formulation we obtain $S_1 = \{p_{18}, p_{21}, p_{31}, p_{34}, p_{35}\}$ with $M_{D_1} = 8p_1 + 5p_5 + p_{10} + p_{11} + p_{13} + p_{15} + p_{16} + p_{17} + 6p_{19} + p_{20} + p_{24} + 2p_{25} + 6p_{28} + 8p_{37}$ and $\overline{S}_1 = \{p_{12}, p_{13}, p_{20}\}$, we have $M_{D_1}(\overline{S}_1) = 2$. Moreover, $l_{S_1} = p_{12} + p_{13} + p_{20}$ and $\widehat{l}_{S_1} = p_6 + p_7 + p_8 + p_9 + p_{10} + p_{11} + p_{12} + p_{13} + p_{20}$. Hence, we have $[N_{p_{c_2}}] = -2t_5 + t_{17} + t_{18} - t_{19} + t_{20}$, $M(p_{c_2}) = 2$, $[N_2] = [[N_1]^T \mid [N_{p_{c_2}}]^T]^T$, and $M_2 = [M_1^T\ M(p_{c_2})]^T$. $M(p_{c_2})$ is set to be 2 since the weight of outgoing arc $W(p_{c_2}, t_5) = 2$. After adding p_{c_2}, a PT-ordinary transformation is needed owing to the existence of a weighted arc.

Third, from the IP formulation, we obtain $S_2 = \{p_2, p_4, p_{16}, p_{23}, p_{30}, p_{33}\}$ with $M_{D_2} = 8p_1 + 7p_5 + p_{10} + p_{15} + 6p_{19} + p_{22} + p_{24} + 2p_{25} + p_{26} + p_{31} + p_{34} + p_{35} + p_{37}$ and $\overline{S}_2 = \{p_{15}, p_{22}\}$, we have $M_{D_2}(\overline{S}_2) = 2$. Moreover, $l_{S_2} = p_{15} + p_{22}$ and $\widehat{l}_{S_2} = p_{14} + p_{15} + p_{10} + p_{21} + p_{22}$. Hence, we have $[N_{p_{c_3}}] = -t_5 + t_{15} - t_{19} + t_{22}$, $M(p_{c_3}) = 1$, $[N_3] = [[N_2]^T \mid [N_{p_{c_3}}]^T]^T$, and $M_3 = [M_2^T\ M(p_{c_3})]^T$.

Forth, from the IP formulation we obtain $S_3 = \{p_4, p_{11}, p_{16}, p_{22}, p_{26}, p_{30}\}$ with $M_{D_3} = 6p_1 + p_2 + p_3 + 7p_5 + p_{10} + p_{17} + 8p_{19} + 2p_{25} + p_{29} + p_{31} + p_{32} + p_{33} + p_{35} + p_{38} + p_{41}$ and $\overline{S}_3 = \{p_2, p_3\}$, we have $M_{D_3}(\overline{S}_3) = 1$. Moreover, $l_{S_3} = p_2 + p_3$ and $\widehat{l}_{S_3} = p_2 + p_3$. Hence, we have $[N_{p_{c_4}}] = -t_1 + t_3$, $M(p_{c_4}) = 1$, $[N_4] = [[N_3]^T \mid [N_{p_{c_4}}]^T]^T$, and $M_4 = [M_3^T\ M(p_{c_4})]^T$.

TABLE II
GENERATED MONITORS FOR THE NET IN FIG. 7 DUE TO [3]

i	${}^\bullet p_{c_i}$	$p_{c_i}^\bullet$	i	${}^\bullet p_{c_i}$	$p_{c_i}^\bullet$	i	${}^\bullet p_{c_i}$	$p_{c_i}^\bullet$
1	$\{t_{14}, t_{23}\}$	$\{t_5, t_{19}\}$	18	$\{t_3, t_{18}, t_{22}\}$	$\{t_1, t_5, t_{19}\}$	35	$\{t_{16}, t_{23}\}$	$\{t_5, t_{19}\}$
2	$\{t_{17}, t_{18}, t_{20}\}$	$\{2t_5\ t_{19}\}$	19	$\{t_6, t_{13}, t_{23}\}$	$\{t_5, t_{19}\}$	36	$\{t_{17}, t_{18}, t_{22}\}$	$\{2t_5, t_{19}\}$
3	$\{t_{18}\}$	$\{t_5\}$	20	$\{t_3, t_{16}, t_{21}\}$	$\{t_1, t_5, t_{19}\}$	37	$\{t_{16}, t_{18}, t_{21}\}$	$\{2t_5, t_{19}\}$
4	$\{t_{17}, t_{20}\}$	$\{t_5, t_{19}\ \}$	21	$\{t_6, t_{11}, t_{18}, t_{21}\}$	$\{2t_5, t_{19}\}$	38	$\{t_{17}, t_{22}\}$	$\{t_5, t_{19}\}$
5	$\{t_6, t_{11}, t_{18}, t_{20}\}$	$\{2t_5, t_{19}\}$	22	$\{t_3, t_{17}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	39	$\{t_6, t_{11}, t_{18}, t_{21}\}$	$\{2t_5, t_{19}\}$
6	$\{t_6, t_{13}, t_{18}, t_{20}\}$	$\{2t_5, t_{19}\ \}$	23	$\{t_3, t_{16}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	40	$\{t_6, t_{13}, t_{18}, t_{22}\}$	$\{2t_5, t_{19}\}$
7	$\{t_3, t_{17}, t_{18}, t_{23}\}$	$\{t_1, 2t_5, t_{19}\}$	24	$\{t_3, t_{17}, t_{22}\}$	$\{t_1, t_5, t_{19}\}$	41	$\{t_{16}, t_{22}\}$	$\{t_5, t_{19}\}$
8	$\{t_3, t_{15}, t_{18}, t_{21}, t_{23}\}$	$\{t_1, 2t_5, 2t_{19}\}$	25	$\{t_3, t_{16}, t_{22}\}$	$\{t_1, t_5, t_{19}\}$	42	$\{t_{17}, t_{18}, t_{21}\}$	$\{2t_5, t_{19}\}$
9	$\{t_3, t_{17}, t_{18}, t_{22}\}$	$\{t_1, 2t_5, t_{19}\}$	26	$\{t_3, t_{17}, t_{21}\}$	$\{t_1, t_5, t_{19}\}$	43	$\{t_{16}, t_{18}, t_{21}\}$	$\{2t_5, t_{19}\}$
10	$\{t_3, t_{16}, t_{18}, t_{21}\}$	$\{t_1, 2t_5, t_{19}\}$	27	$\{t_3, t_{16}, t_{21}\}$	$\{t_1, t_5, t_{19}\}$	44	$\{t_{17}, t_{21}\}$	$\{t_5, t_{19}\}$
11	$\{t_3, t_{17}, t_{18}, t_{21}\}$	$\{t_1, 2t_5, t_{19}\}$	28	$\{t_3, t_{15}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	45	$\{t_6, t_{13}, t_{21}\}$	$\{t_5, t_{19}\}$
12	$\{\ t_3, t_{16}, t_{18}, t_{21}\}$	$\{t_1, 2t_5, t_{19}\}$	29	$\{t_3, t_{15}, t_{22}\}$	$\{t_1, t_5, t_{19}\}$	46	$\{t_6, t_{13}, t_{18}, t_{21}\}$	$\{2t_5, t_{19}\}$
13	$\{t_3, t_{18}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	30	$\{t_3\}$	$\{t_1\}$	47	$\{t_{16}, t_{21}\}$	$\{t_5, t_{19}\}$
14	$\{t_3, t_{15}, t_{18}\}$	$\{t_1, 2t_5\}$	31	$\{t_{17}, t_{18}, t_{23}\}$	$\{2t_5, t_{19}\}$	48	$\{t_{15}, t_{23}\}$	$\{t_5, t_{19}\}$
15	$\{t_3, t_{18}, t_{21}\}$	$\{t_1, t_5, t_{19}\}$	32	$\{t_{16}, t_{18}, t_{23}\}$	$\{2t_5, t_{19}\}$	49	$\{t_{15}, t_{22}\}$	$\{t_5, t_{19}\}$
16	$\{t_{18}\}$	$\{t_5\}$	33	$\{t_{17}, t_{23}\}$	$\{t_5, t_{19}\}$	50	–	–
17	$\{t_3, t_{12}, t_{13}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	34	$\{t_6, t_{11}, t_{16}, t_{23}\}$	$\{t_{5:2}\ t_{19}\}$	51	–	–

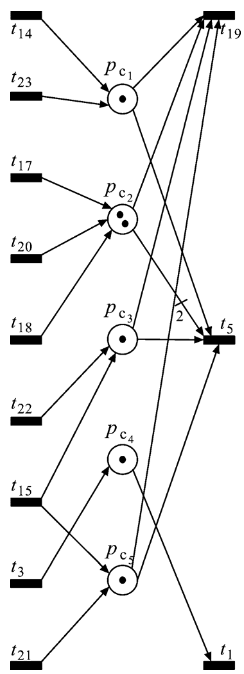


Fig. 8. The supervisor.

TABLE III
GENERATED MONITORS FOR THE NET IN FIG. 7 DUE TO [30]

i	${}^\bullet p_{c_i}$	$p_{c_i}^\bullet$	$M(p_{c_i})$
1	$\{t_{14}, t_{23}\}$	$\{t_5, t_{19}\}$	1
2	$\{t_{17}, t_{18}, t_{20}\}$	$\{2t_5, t_{19}\}$	2
3	$\{t_{18}\}$	$\{t_5\}$	1
4	$\{t_{17}, t_{20}\}$	$\{t_5, t_{19}\}$	1
5	$\{t_6, t_{11}, t_{18}, t_{20}\}$	$\{2t_5, t_{19}\}$	2
6	$\{t_6, t_{13}, t_{18}, t_{20}\}$	$\{2t_5, t_{19}\}$	2
7	$\{t_3, t_{17}, t_{18}, t_{23}\}$	$\{t_1, 2t_5, t_{19}\}$	2
8	$\{t_3, t_{17}, t_{18}, t_{21}\}$	$\{t_1, 2t_5, t_{15}\}$	2
9	$\{t_{13}, t_{18}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	1
10	$\{t_{18}\}$	$\{t_5\}$	1
11	$\{t_3, t_{12}, t_{13}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	1
12	$\{t_6, t_{13}, t_{23}\}$	$\{t_5, t_{19}\}$	1
13	$\{t_6, t_{11}, t_{18}, t_{21}\}$	$\{2t_5, t_{19}\}$	2
14	$\{t_{13}, t_{17}, t_{23}\}$	$\{t_1, t_5, t_{19}\}$	1
15	$\{t_6, t_{13}, t_{21}\}$	$\{t_5, t_{19}\}$	4

Fifth, from the IP formulation we obtain $S_4 = \{p_2, p_4, p_{12}, p_{17}, p_{22}, p_{30}, p_{34}\}$ with $M_{D_4} = 8p_1 + 7p_5 + p_6 + p_{16} + 7p_{19} + p_{21} + 2p_{25} + p_{26} + p_{27} + p_{29} + p_{31} + p_{32} + p_{33} + p_{35} + p_{37} + p_{42}$ and $\overline{S}_4 = \{p_{16}, p_{21}\}$, we have $M_{D_4}(\overline{S}_3) = 2$. Moreover, $l_{S_4} = p_{16} + p_{21}$ and $\widehat{l}_{S_4} = p_{14} + p_{15} + p_{16} + p_{20} + p_{21}$. Hence, we have $[N_{p_{c_5}}] = -t_5 + t_{16} - t_{19} + t_{21}$, $M(p_{c_5}) = 1, [N_5] = [[N_4]^T \mid [N_{p_{c_5}}]^T]^T$, and $M_5 = [M_4^T\ M(p_{c_5})]^T$.

After p_{c_5} is added, (N_5, M_5) is found to be live since no more empty siphons can be detected with the IP algorithm. Fig. 8 shows the derived supervisor.

For an AEMG, the above results indicate that the liveness enforcing supervisor can be derived in an iterative way, thus avoiding the enumeration of all the siphons or states. To the best of our knowledge, such a type of systems is quite general in both the process structure and resource allocation requirement. Most of the existing approaches, such as [12] and [16], are not directly applicable to the new class of net models. Our approach establishes the equivalence relationship between the absence of undermarked siphons and the absence of deadlocks. This makes feasible the application of some existing approaches, such as [2] and [30], because they also consider general nets. Also, they are representative because the former one is the first research to cope with siphon-induced deadlock problems in general Petri nets while the latter aims to derive the simplest supervisors. A comparison is thus conducted only between our method and the ones in [2] and [30] via the example in Fig. 7.

For the net in Fig. 7, the method in [2] leads to 49 monitors, namely, $p_{c_1} - p_{c_{49}}$, respectively, corresponding to $S_0 - S_{48}$, as shown in Table II. When these 49 monitors are added, a liveness-enforcing supervisor is obtained.

Among the above-derived 49 siphons $S_0 - S_{48}$, 15 of them, i.e., $S_0 - S_6, S_{10}, S_{12}, S_{15}, S_{16}, S_{18}, S_{20}, S_{21}, S_{44}$, are found to be elementary ones according to [30]. Thus, only 15 monitors are necessary to constitute a supervisor, as shown in Table III.

Further analysis shows that the permissible states produced by our approach and the ones in [2] and [30] are 167, 167, and 73, respectively. Compared with [2], our method can implement a supervisor with much fewer monitors. Compared with [30], our method can produce much more reachable states. Note that the methods in [2] and [30] need to conduct the enumeration of all the siphons, which has the exponential complexity. Different from theirs, our approach does not need to enumerate all siphons. This means that much less constraints are introduced to synthesize the final supervisor. As it is well known, each constraint may reduce the reachable states. As a result, our approach needs less constraints and produces more permissive behavior.

VII. Conclusion

This work focuses on a novel type of Petri nets, which can model automated manufacturing systems admitting both flexible routes and assembly operations. It is proved that the presence of deadlock can be related to the emergence of undermarked siphons during the evolution of such net models. To identify such siphons, we propose an algebraic method such that they can be derived in an iterative way. Compared with the existing approach, our model is more general as it allows both flexible routes and assembly operations. Furthermore, our method is more efficient as no siphon enumeration algorithm is required. Numerical results show that our resulting supervisor is simple in its structure and can ensure more permissive behaviors. Although our approach is shown via an example representing manufacturing systems, it is applicable to other domains, e.g., workflows, business processes, transportation systems, communication protocols, and logistic (service) systems. The generalization of the research results in this paper will be conducted. More research is needed to study the efficiency of the proposed method. Another direction is how to ensure the maximal permissiveness and performance of the controlled net.

References

[1] I. B. Abdallah and H. A. ElMaraghy, "Deadlock prevention and avoidance in FMS: A Petri net based approach," *Int. J. Adv. Manuf. Tech.*, vol. 14, no. 10, pp. 704–715, Sep. 1998.

[2] K. Barkaoui and I. B. Abdallah, "A deadlock prevention method for a class of FMS," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Vancouver, BC, Canada, Oct. 1995, pp. 4119–4124.

[3] K. Barkaoui and J. F. Peyre, "On liveness and controlled siphons in Petri nets," in *Lecture Notes in Computer Science*. Berlin, Germany: Springer-Verlag, 1996, vol. 1091, pp. 57–72.

[4] F. Basile, P. Chiacchio, A. Giua, and C. Seatzu, "Deadlock recovery of Petri net models controlled using observers," in *Proc. IEEE Symp. Emerg. Tech. Fact. Auto., Antibes-Juan les pins*, France, Oct. 2001, pp. 441–449.

[5] F. Basile, P. Chiacchio, and A. Giua, "An optimizaiton approach to Petri net monitor design," *IEEE Trans. Automat. Contr.*, vol. 52, no. 2, pp. 306–311, Feb. 2007.

[6] F. Basile, L. Recalde, P. Chiacchio, and M. Silva, "Closed-loop live marked graphs under generalized mutual exclusion constraint enforcement," *Discrete Event Dyn. Syst.*, vol. 19, no. 1, pp. 1–30, Mar. 2009.

[7] S. Bernardi, J. Campos, and J. Merseguer, "Timing-failure risk assessment of UML design using time Petri net bound techniques," *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 90–104, Feb. 2011.

[8] S. Bernardi and J. Campos, "Computation of performance bounds for real-time systems using time Petri nets," *IEEE Trans. Ind. Informat.*, vol. 5, no. 2, pp. 168–180, May 2009.

[9] F. Chu and X. L. Xie, "Deadlock analysis of Petri nets using siphons and mathematical programming," *IEEE Trans. Robot. Autom.*, vol. 13, no. 6, pp. 793–804, Dec. 1997.

[10] D. Y. Chao, "Maximal class of weakly live ordinary Petri nets without emptiable siphons," *IEEE Trans. Syst., Man, Cybern. B. Cybern.*, vol. 36, no. 6, pp. 1332–1341, Dec. 2006.

[11] D. Y. Chao, "Incremental approach to computation of elementary siphons for arbitrary simple sequential processes with resources," *IET Proc. Contr. Theor. Appl.*, vol. 2, no. 2, pp. 168–179, 2008.

[12] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995.

[13] J. Ezpeleta and R. Valk, "A polynomial deadlock avoidance method for a class of nonsequential resource allocation systems," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 36, no. 6, pp. 1234–1243, Nov. 2006.

[14] J. Ezpeleta, F. García-Valles, and J. M. Colom, "A class of well structured Petri nets for flexible manufacturing systems," in *Int. Conf. Application and Theory of Petri Nets*, J. Desel and M. Silva, Eds. Berlin, Germany: Springer-Verlag, Jun. 1998, vol. 1420, Lecture Notes in Computer Science, pp. 64–83.

[15] A. Ghaffari, N. Rezg, and X. L. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 137–142, Feb. 2003.

[16] Y. S. Huang, M. D. Jeng, X. L. Xie, and S. L. Chung, "Deadlock prevention policy based on Petri nets and siphons," *Int. J. Prod. Res.*, vol. 39, no. 2, pp. 283–305, Jan. 2001.

[17] B. Hruz and M. C. Zhou, *Modeling and Control of Discrete Event Dynamic Systems*. New York: Springer, 2007.

[18] H. S. Hu, M. C. Zhou, and Z. W. Li, "Supervisor design to enforce production ratio and absence of deadlock in automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 41, no. 2, pp. 201–212, Mar. 2011.

[19] H. S. Hu and Z. W. Li, "Clarification on the computation of liveness-enforcing supervisor for resource allocation systems with uncontrollable behavior and forbidden states," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 3, pp. 557–558, Jul. 2009.

[20] H. S. Hu, M. C. Zhou, and Z. W. Li, "Liveness enforcing supervision of video streaming systems using non-sequential Petri nets," *IEEE Trans. Multi.*, vol. 11, no. 8, pp. 1457–1465, Nov. 2009.

[21] H. S. Hu, M. C. Zhou, and Z. W. Li, "Algebraic synthesis of timed supervisor for automated manufacturing systems using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 3, pp. 549–557, Jul. 2010.

[22] H. S. Hu, M. C. Zhou, and Z. W. Li, "Low-cost and high-performance supervision in ratio-enforced automated manufacturing systems using timed Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 7, no. 4, pp. 933–944, Oct. 2010.

[23] S. Hur, R. Katebi, and A. Taylor, "Modeling and control of a plastic film manufacturing web process," *IEEE Trans. Ind. Informat.*, vol. 7, no. 2, pp. 171–178, May 2011.

[24] D. Herrero-Perez and H. Martinez-Barbera, "Modeling distributed transportation systems composed of flexible automated guided vehicles in flexible manufacturing systems," *IEEE Trans. Ind. Informat.*, vol. 6, no. 2, pp. 166–180, May 2010.

[25] M. V. Iordache and P. J. Antsaklis, "Synthesis of supervisor enforcing general linear constraints in Petri nets," *IEEE Trans. Autom. Control*, vol. 48, no. 11, pp. 2036–2039, Nov. 2003.

[26] M. V. Iordache and P. J. Antsaklis, "Design of τ-liveness enforcing supervisors in Petri nets," *IEEE Trans. Autom. Contr.*, vol. 48, no. 11, pp. 1962–1974, Nov. 2003.

[27] M. Kloetzer, C. Mahulea, C. Belta, and M. Silva, "An automated framework for formal verification of timed continuous Petri nets," *IEEE Trans. Ind. Informat.*, vol. 6, no. 3, pp. 460–471, Aug. 2010.

[28] K. Lautenbach and H. Ridder, *The Linear Algebra of Deadlock Avoidance—A PN Approach*. Landau, Germany: Inst. Comput. Sci., Univ. Koblenz, 1996.

[29] Z. W. Li and M. C. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 34, no. 1, pp. 38–51, Jan. 2004.

[30] Z. W. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 38, no. 2, pp. 1083–4427, Mar. 2008.

[31] Z. W. Li, M. C. Zhou, and N. Q. Wu, "A survey and comparison of Petri net-based deadlock prevention policies for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. B. Appl. Rev.*, vol. 38, no. 2, pp. 173–188, Mar. 2008.

[32] J. O. Moody and P. J. Antsaklis, "Petri net supervisors for DES with uncontrollable and unobservable transitions," *IEEE Trans. Autom. Control*, vol. 45, no. 3, pp. 462–476, Mar. 2000.

[33] J. Park and S. A. Reveliotis, "Deadlock avoidance in sequential resource allocation systems with multiple resource acquisitions and flexible routings," *IEEE Trans. Autom. Control*, vol. 46, no. 10, pp. 1572–1583, Oct. 2001.

[34] L. Piroddi, R. Cordone, and I. Fumagalli, "Combined siphon and marking generation for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 39, no. 3, pp. 650–661, May 2009.

[35] L. Piroddi, R. Cordone, and I. Fumagalli, "Selective siphon control for deadlock prevention in Petri nets," *IEEE Trans. Syst., Man,Cybern. A. Syst., Humans*, vol. 38, no. 6, pp. 1337–1348, Nov. 2008.
[36] I. Sindicic, S. Bogdan, and T. Petrovic, "Resource allocation in free-choice multiple reentrant manufacturing systems based on machine-job incidence matrix," *IEEE Trans. Ind. Informat.*, vol. 7, no. 1, pp. 105–114, Feb. 2011.
[37] F. Tricas, F. Garcia-Valles, J. M. Colom, and J. Ezpeleta, "A Petri net structure-based deadlock prevention solution for sequential resource allocation systems," in *Proc. Int. Conf. Robot. Autom.*, Barcelona, Spain, Apr. 2005, pp. 272–278.
[38] M. Uzam and M. C. Zhou, "An improved iterative synthesis method for liveness enforcing supervisors of flexible manufacturing systems," *Int. J. Prod. Res.*, vol. 44, no. 10, pp. 1987–2030, May 2006.
[39] M. Vallee, M. Merdan, W. Lepuschitz, and G. Koppensteiner, "Decentralized reconfiguration of a flexible transportation system," *IEEE Trans. Ind. Informat.*, vol. 7, no. 3, pp. 505–516, Aug. 2011.
[40] K. Y. Xing, B. S. Hu, and H. X. Chen, "Deadlock avoidance policy for Petri-net modeling of flexible manufacturing systems with shared resources," *IEEE Trans. Autom. Control*, vol. 41, no. 2, pp. 289–295, Feb. 1996.
[41] K. Yamalidou, J. O. Moody, M. Lemmon, and P. J. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, pp. 15–18, Jan. 1996.
[42] *Deadlock Resolution in Computer-Integrated Systems*, M. C. Zhou and M. P. Fanti, Eds. New York: Marcel Dekker, 2004.
[43] M. C. Zhou and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. London, U.K.: Kluwer, 1993.
[44] M. C. Zhou and F. DiCesare, "Parallel and sequential mutual exclusions for Petri net modeling for manufacturing systems with shared resources," *IEEE Trans. Robot. Autom.*, vol. 7, no. 4, pp. 515–527, Aug. 1991.
[45] M. C. Zhou and M. C. Leu, "Modeling and performance analysis of a flexible PCB assembly station using Petri nets," *IEEE Trans. ASME, J. Electron. Packag.*, vol. 113, no. 4, pp. 410–416, 1991.
[46] M. C. Zhou, F. DiCesare, and A. A. Desrochers, "A hybrid methodology for synthesis of Petri nets for manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 8, no. 3, pp. 350–361, 1992.
[47] M. C. Zhou, F. DiCesare, and D. Rudolph, "Design and implementation of a Petri net based supervisor for a flexible manufacturing system," *IFAC J. Automatica*, vol. 28, no. 6, pp. 1199–1208, 1992.
[48] M. C. Zhou and M. D. Jeng, "Modeling, analysis, simulation, scheduling, and control of semiconductor manufacturing systems: A Petri net approach," *IEEE Trans. Semicond. Manuf.*, vol. 11, no. 3, pp. 333–357, 1998.
[49] M. C. Zhou and K. Venkatesh, *Modeling, Simulation and Control of Flexible Manufacturing Systems: A Petri Net Approach*. Singapore: World Scientific, 1998.
[50] Z. W. Li and M. C. Zhou, "Two-stage method for synthesizing liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Ind. Informat.*, vol. 2, no. 4, pp. 313–325, Nov. 2006.
[51] Z. W. Li and M. C. Zhou, "Clarifications on the definitions of elementary siphons in Petri nets," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 36, no. 6, pp. 1227–1229, Nov. 2006.
[52] Z. W. Li, H. S. Hu, and A. R. Wang, "Design of liveness-enforcing supervisors for flexible manufacturing systems using Petri nets," *IEEE Trans. Syst., Man, Cybern. C. Appl. and Rev.*, vol. 37, no. 4, pp. 517–526, Jul. 2007.
[53] N. Q. Wu, M. C. Zhou, and Z. W. Li, "Resource-oriented Petri net for deadlock avoidance in flexible assembly systems," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 38, no. 1, pp. 56–69, Jan. 2008.
[54] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons in Petri nets and their application," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 38, no. 1, pp. 133–148, Jan. 2008.
[55] Z. W. Li, M. C. Zhou, and M. D. Jeng, "A maximally permissive deadlock prevention policy for FMS based on Petri net siphon control and the theory of regions," *IEEE Trans. Autom. Sci. Eng.*, vol. 5, no. 1, pp. 182–188, Jan. 2008.
[56] N. Wu, C. Chu, F. Chu, and M. C. Zhou, "A Petri net method for schedulability and scheduling problems in single-arm cluster tools with wafer residency time constraints," *IEEE Trans. Semicond. Manuf.*, vol. 21, no. 2, pp. 224–237, May 2008.
[57] Z. W. Li and M. C. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. London, U.K.: Springer-Verlag.
[58] K. Xing, M. C. Zhou, X. Yang, and F. Tian, "Optimal Petri net-based polynomial-complexity deadlock avoidance policies for automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern., Part A*, vol. 39, no. 1, pp. 188–199, Jan. 2009.
[59] Y. F. Chen and Z. W. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 45, no. 5, pp. 1028–1034, May 2011.
[60] Y. F. Chen, Z. W. Li, and M. Khalgui, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 374–393, Apr. 2011.
[61] Z. W. Li, G. Y. Liu, M.-H. Hanisch, and M. C. Zhou, "Deadlock prevention based on structure reuse of Petri net supervisors for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 42, no. 1, pp. 178–191, Jan. 2012.
[62] Y.-S. Huang, Y.-L. Pan, and M. C. Zhou, "Computationally improved optimal deadlock control policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern.–Part A. Syst. and Humans*, vol. 42, no. 2, pp. 401–415, Mar. 2012.
[63] Z. W. Li, N. Q. Wu, and M. C. Zhou, "Deadlock control of automated manufacturing systems based on Petri nets - A literature review," *IEEE Trans. Syst., Man, Cybern. C. Appl. and Rev.*, vol. 42, no. 4, pp. 437–462, Jul. 2012.
[64] N. Wu and W. Q. Zeng, "Deadlock avoidance in AGV system using colored Petri net model," *Int. J. Prod. Res.*, vol. 40, no. 1, pp. 223–238, 2008.
[65] N. Wu, "Necessary and sufficient conditions for deadlock-free operation in flexible manufacturing systems using a colored Petri net model," *IEEE Trans. Syst., Man, Cybern.–Part C*, vol. 29, no. 2, pp. 192–204, 1999.

Hesuan Hu (M'11) received the B.S. degree in computer engineering, and the M.S., and Ph.D. degrees in electromechanical engineering from Xidian University, Shaanxi, China, in 2003, 2005, and 2010, respectively.

He is currently affiliated with Xidian University as a faculty member. From March 2008 to March 2011, he worked at New Jersey Institute of Technology as a research scholar. His research interests include Petri nets, discrete event systems, automated manufacturing systems, multimedia streaming systems, and artificial intelligence. In the above areas, he has more than 50 publications in journals, book chapters, and conference proceedings.

Dr. Hu received the Franklin V. Taylor Outstanding Paper Award from the IEEE Systems, Man, and Cybernetics Society in 2010. He has been a reviewer for many journals and a program committee member for numerous conferences. He is listed in *Who's Who in America*. He is the Associate-Editor-at-Large of the *Journal of Intelligent Manufacturing*.

MengChu Zhou (S'88-M'90-SM'93-F'03) received the B.S. degree in electrical engineering from the Nanjing University of Science and Technology, Nanjing, China, in 1983, the M.S. degree in automatic control from the Beijing Institute of Technology, Beijing, China, in 1986, and the Ph.D. degree in computer and systems engineering from the Rensselaer Polytechnic Institute, Troy, NY, in 1990.

He joined the New Jersey Institute of Technology (NJIT), Newark, in 1990, and is a Professor of Electrical and Computer Engineering. He is presently a Professor with Tongji University, Shanghai, China. He has over 460 publications including 10 books, over 200 journal papers, (majority in IEEE TRANSACTIONS), and 18 book-chapters. His research interests are in intelligent automation, life cycle engineering and sustainability evaluation, Petri nets, sensor networks, energy systems, semiconductor manufacturing, and transportation.

Dr. Zhou is a Life Member of the Chinese Association for Science and Technology-USA and served as its President in 1999. He is a Fellow of the American Association for the Advancement of Science (AAAS). He was the recipient of NSF's Research Initiation Award, the CIM University-LEAD Award by the Society of Manufacturing Engineers, the Perlis Research Award and the Fenster Innovation in Engineering Education Award by NJIT, the Humboldt Research

Award for U.S. Senior Scientists, the Leadership Award and Academic Achievement Award by the Chinese Association for Science and Technology-USA, the Asian American Achievement Award by the Asian American Heritage Council of New Jersey, and the Outstanding Contributions Award, the Distinguished Lecturership and Franklin V. Taylor Memorial Award of the IEEE SMC Society. He is the founding Editor of the IEEE Press Book Series on Systems Science and Engineering, Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING, and Associate Editor of the IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS-PART A, SYSTEMS AND HUMANS and the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS. He was General Chair of the IEEE Conference on Automation Science and Engineering, Washington DC, August 23-26, 2008, General Co-Chair of the 2003 IEEE International Conference on System, Man and Cybernetics (SMC), Washington DC, October 5-8, 2003, Founding General Co-Chair of the 2004 IEEE International Conference on Networking, Sensing and Control, Taipei, March 21-23, 2004, and General Chair of the 2006 IEEE International Conference on Networking, Sensing and Control, Ft. Lauderdale, FL, April 23-25, 2006. He was the founding chair of Discrete Event Systems Technical Committee and founding Co-Chair of Enterprise Information Systems Technical Committee of IEEE SMC Society, and Chair (founding) of Semiconductor Manufacturing Automation Technical Committee of IEEE Robotics and Automation Society.

Zhiwu Li (M'06–SM'07) received the B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering, respectively, from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively.

In 1992, he joined Xidian University, where he is currently a Professor and Dean of School of Electro-Mechanical Engineering. From 2002 to 2003, he was a Visiting Professor at the Department of Electrical and Computer Engineering, University of Toronto, Toronto, ON, Canada. From 2007 to 2008, he was a Visiting Scientist at the Department of Mechanical Engineering, Technion-Israel Institute of Technology, Technion City, Haifa, Israel. Since 2008, he has been a Visiting Professor in the Automation Technology Laboratory, Institute of Computer Science, Martin-Luther University of Halle-Wittenburg, Halle, Germany. His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, and systems integration.

Ying Tang (S'99–M'02–SM'07) received the B.S. degree in electrical engineering and the M.S. degree in computer engineering from Northeastern University, China, in 1996 and 1998, respectively, and the Ph.D. degree in electrical engineering from the New Jersey Institute of Technology, Newark, in 2001.

She is currently an Associate Professor of Electrical and Computer Engineering at Rowan University, Glassboro, NJ. She has published 4 book chapters, and over 80 journal and conference proceedings articles. Her research interests include discrete event systems and visualization, Petri nets and applications, virtual/augmented reality, green manufacturing design and optimization, intelligent learning game systems and artificial intelligence.

Dr. Tang is an Associate Editor of the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING.