

Supervisor simplification for AMS based on petri nets and inequality analysis

Hu, Hesuan; Liu, Yang

2014

Hu, H., & Liu, Y. (2014). Supervisor simplification for ams based on petri nets and inequality analysis. *IEEE Transactions on Automation Science and Engineering*, 11(1), 66-77.

<https://hdl.handle.net/10356/102332>

<https://doi.org/10.1109/TASE.2013.2288645>

© 2014 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works. Published version of this article is available at [DOI: <http://dx.doi.org/10.1109/TASE.2013.2288645>]

Downloaded on 24 Aug 2022 22:50:19 SGT

Supervisor Simplification for AMS Based on Petri Nets and Inequality Analysis

Hesuan Hu, *Senior Member, IEEE*, and Yang Liu

Abstract—In the framework of automated manufacturing systems (AMS), Petri nets are widely used to model, analyze, and control them. Resolving deadlocks is of paramount significance because their emergence may zero a system's throughput. Supervisory control technique is the most widely adopted method to resolve them. A control policy can be converted into satisfying a set of inequalities, each of which corresponds to a siphon in a Petri net structure. The number of siphons can be exponential in the worst case, so does the number of inequalities. Taking into account the independent and dependent inequalities, this paper proposes a method to remove all the dependent inequalities while preserving only the independent ones. This method can significantly reduce the size of a supervisory controller. Examples are presented to illustrate the effectiveness and efficiency of this method.

Note To Practitioners—Owing to its importance in practice, deadlock arouses interest. Various resolution methods have been proposed by researchers and practitioners. The recent advance is to associate various sensors and actuators to certain predetermined working nodes for the sake of management and coordination of limited resources among concurrent operations. This uses a supervisory controller to a plant model. Nevertheless, existing approaches suffer from structural complexity due to their failure to reduce the sophisticated supervisor structures, which impedes their implementation in practice. Without a simplified control network, the communication cost among heterogeneous nodes increases while the responsive time decreases. Through the analysis on the dependency relationship among different inequalities, this paper presents a reduction technique to significantly simplify the supervisor structures. Without any extra investment, our approach betters a controlled system's performance.

Index Terms—Liveness enforcing supervision, Petri nets, automated manufacturing systems, supervisor simplification, linear programming.

I. INTRODUCTION

Thanks to the strong desire to reduce manufacturing cost, improve product quality, and ensure workplace safety, automated manufacturing systems (AMS) are developed to automatically schedule manufacturing tasks, assign production resources, and handle mechanical work [1], [2], [8], [12], [13], [17]–[21], [25], [26], [31]. Sophisticated software is involved

Manuscript received xxxxxx xx, 200x; revised xxxxxx xx, 200x. *This work was supported by the Natural Science Foundation of China under Grant No 61203037 and 51305321, Fundamental Research Funds for the Central Universities K5051304004 and K5051304021, New Century Excellent Talents in University under Grant No NCET-12-0921, and NAP under Grant No M4081155.020 and M4080996.020.

Hesuan Hu is with School of Electro-mechanical Engineering, Xidian University, Xi'an, Shaanxi 710071, P. R. China. He is also with School of Computer Engineering, Nanyang Technological University, Singapore 639798. Yang Liu is with School of Computer Engineering, Nanyang Technological University, Singapore 639798 (email: hshu@mail.xidian.edu.cn; yangliu@ntu.edu.sg).

to diagnose the malfunction, inspect the product quality, and monitor a production line. In a fully automated facility, any human activity is excluded to minimize the cost. Its dynamics is featured by a series of activities driven by the asynchronous occurrences of discrete events. Differential and difference equations become inapplicable despite their wide application to time-driven systems [23], [24], [28], [38]. Petri nets prove to be one of the most powerful and popular mathematical tools to tackle such event-driven systems owing to their formalism and compactness [3], [4], [6], [7], [9]–[11], [14], [16], [22], [27], [29], [30], [32]–[35], [37].

To fulfill specifications, either by intrinsic default or by external imposition, feedback control technique must be developed for such systems [28], [38]. A plant model that induces undesirable behavior must be controlled [28], [38]. In order to change its original behavior, one must design a supervisor, thereby leading to a so-called supervisory control technique (SCT). In the spirit of SCT, a supervisor must constrain the behavior of systems into the legal or admissible domain by forbidding their illegal one [38]. As an elegant supervisor, a premise is the separation between the plant model from the supervisor. Some formal methods, such as automata, are of difficulty to achieve so owing to their language-based paradigm. Petri nets prove to be promising as they can represent both of them in a compact and split way [23], [24], [28], [38].

Unlike automata, Petri nets provide abundant structure information with regard to a modeled system. Through capturing such information, one can synthesize the supervisor in a linear-algebraic way. In [15], the concept of generalized mutual exclusion constraints (GMEC) is proposed to limit the weighted sum of tokens in a subset of places. Each GMEC corresponds to a linear supervisory specification which can be implemented by a control place (monitor, in short) and its related flows to some controllable transitions. In terms of SCT, each GMEC determines a set of forbidden states while each monitor specifies the control mechanism deciding which transition to fire and which one to forbid at each state. An optimal supervisor should avoid the occurrence of all these forbidden states without intervening other good ones.

In the cases of modeling an AMS with Petri nets, it is not surprising that the feature of liveness is critical because it guarantees that all the processes can be executed to their termination [1]–[5], [12], [13], [16], [17], [19]–[21], [24]–[26], [31], [33]. Siphons are a structural object in Petri nets to characterize these deadlock states. To avoid such states, siphons cannot be undermarked [2], [3], [12], [17], [25], [26]. Alternatively, they must be always sufficiently marked during the system evolution, which leads to a typical GMEC problem

[15], [28], [38]. Unfortunately, the number of siphons grows exponentially with regard to the net size. This implies that the number of monitors can be astronomical even for a moderate-sized system. Using inequality analysis, this paper proposes a method to identify and remove the redundant inequalities corresponding to certain GMEC. As a result, the number of monitors can decrease significantly, thus leading to a reduced liveness-enforcing supervisor.

Our approach makes key contributions in the following respects. First, it is applicable to any Petri net models where the GMEC technique can be used. For clarification, our experiments are conducted on some special classes of Petri nets; however, there is actually no limitation upon its applicability. Second, the behavior of the controlled system can be optimal if one removes only the redundant inequalities without influencing the others. Third, it requires only a procedure to solve a list of linear homogeneous systems, which proves to be polynomial in its computational complexity.

This paper is structured as follows. 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. Some novel approach is presented based upon their properties for the supervisory control purpose. In Section IV, with the aid of inequality analysis, our method is proposed to distinguish independent and dependent inequalities and show their use in simplifying a supervisor. Section V illustrates an example to verify the effectiveness of our approach. Section VI concludes the 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 $W : (P \times T) \cup (T \times P) \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$ such that $P \cup T \neq \emptyset$, $P \cap T = \emptyset$, and $W(x, y) = 0$ if $(x, y) \notin F$. A Petri net is said to be pure if $\forall x, y \in P \cup T : W(x, y) \neq 0 \Rightarrow W(y, x) = 0$. The preset of a node $x \in P \cup T$ is defined as $\bullet x = \{y \in P \cup T \mid (y, x) \in F\}$. Its postset $x^\bullet = \{y \in P \cup T \mid (x, y) \in F\}$. N is a state machine if $W : F \rightarrow \{1\}$ and $\forall t \in T, |\bullet t| = |t^\bullet| = 1$. It is a marked graph if $W : F \rightarrow \{1\}$ and $\forall p \in P, |\bullet p| = |p^\bullet| = 1$. N 's input incidence matrix $[N^-] = [W(p_i, t_j)]$ and output one $[N^+] = [W(t_j, p_i)]$. 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 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 $M \llbracket t \rrbracket$, if $\forall p \in \bullet t, M(p) \geq W(p, t)$. M' is reachable from M , denoted by $M \llbracket \sigma \rrbracket M'$, if there exists a firing sequence $\sigma = \langle t_1 t_2 \dots t_n \rangle$ such that $M \llbracket t_1 \rrbracket M_1 \dots \llbracket t_n \rrbracket M'$. $\vec{\sigma}$ is a $|T|$ -dimensional firing count vector where $\vec{\sigma}(t)$ states the number of t 's appearances in σ . The set of all markings reachable from M_0 is denoted by $R(N, M_0)$. (N, M_0) is bounded if $\exists k \in \mathbb{N}^+ = \mathbb{N} \setminus \{0\}, \forall M \in R(N, M_0), \forall p \in P, M(p) \leq k$. $t \in T$ is live under M_0 if $\forall M \in R(N, M_0), \exists M' \in R(N, M), M' \llbracket t \rrbracket$ holds. t is dead at $M \in R(N, M_0)$ if $\nexists M' \in R(N, M), \exists M' \llbracket t \rrbracket$ holds. (N, M_0) is deadlock-free if $\forall M \in R(N, M_0), \exists t \in T, M \llbracket t \rrbracket$. It is

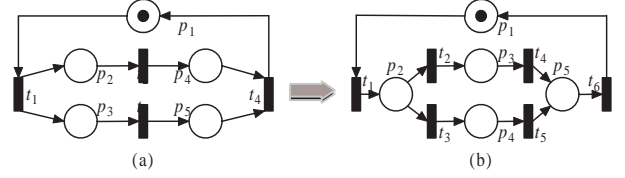


Fig. 1. The conversion from a marked graph to a state machine.

livelock if it is deadlock-free and $\exists t \in T$ so that t is dead. (N, M_0) is live if $\forall t \in T, t$ is live under M_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 $I \geq \mathbf{0}$. $\|I\| = \{p \in P \mid I(p) \neq 0\}$ is called the support of I . For economy of space, $\sum_{p \in P} M(p) \cdot p$ (resp., $\sum_{p \in P} I(p) \cdot p, \sum_{t \in T} J(t) \cdot t$) is used to denote vector M (resp., I, J). (N, M_0) is conservative (resp., consistent) if $\exists I > \mathbf{0}$ (resp., $\exists J > \mathbf{0}$) so that $I^T \cdot [N] = \mathbf{0}^T$ (resp., $[N] \cdot J^T = \mathbf{0}$). A circuit is an ordered set $\langle x_1, x_2, \dots, x_n \rangle$ such that: 1) $\{x_1, x_2, \dots, x_n\} \subseteq P \cup T$; 2) $\forall i \in \mathbb{N}_{n-1} = \{1, 2, \dots, n-1\}, x_{i+1} \in x_i^\bullet$; 3) $\forall \{i, j\} \subseteq \mathbb{N}_n$ except $\{i, j\} = \{1, n\}, x_i \neq x_j$; and 4) $x_1 = x_n$.

A nonempty set $S \subseteq P$ (resp., $Q \subseteq P$) is a siphon (resp., trap) if $\bullet S \subseteq S^\bullet$ (resp., $Q^\bullet \subseteq Q$). A strict minimal siphon is a siphon containing neither other siphon nor trap. $M(p)$ indicates the number of tokens in p at 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.

III. PETRI NET MODELING OF AMS

For better understanding, we focus throughout this paper on a special class of Petri nets, namely, System of Sequential Systems with Shared Resources (S^4R). Nevertheless, this does not necessarily mean the applicability limitation of our proposed method. In fact, it can be used in more general systems without an extension. In such kind of systems, various job types are modeled by state machines. The availability of various resources is modeled by resource places. Since there is no special limitation upon the resource quantity and types at each operation stage, S^4R models a general resource allocation mechanism. Although job routes in S^4R are constrained by a state machine, this does not mitigate its modeling capability because any net model can be converted to a state machine with its reachability graph. Fig. 1 shows the conversion process from a marked graph to a state machine, where the net in Fig. 1(a) denotes a marked graph while the net in Fig. 1(b) denotes a state machine. Compared with the latter, the former is assumed to be able to model operations like disassembly and assembly. Specifically, places of net in Fig. 1(b), i.e., p_1, p_2, p_3, p_4 , and p_5 , correspond to the markings of the net in Fig. 1(a), i.e., $M_0 = p_1, M_1 = p_2 + p_3, M_2 = p_3 + p_4, M_3 = p_2 + p_5$, and $M_4 = p_4 + p_5$. Specifically, an AMS can be partitioned into a set of resource types $\mathbb{R} = \{\mathcal{R}_i, i = 1, 2, \dots, L\}$ and a set of process types $\mathbb{J} = \{\mathcal{J}_j, j = 1, 2, \dots, K\}$. Every resource type \mathcal{R}_i is further characterized by

its capacity $C_i \in \mathbb{N}^+$. Processing requirements of process type \mathcal{J}_j are defined by a set of concurrent and/or sequential stages. Each process stage, say k , modeled by a place p_{jk} is associated with a conjunctive resource requirement, expressed by an L -dimensional vector $a_{p_{jk}}$ with $a_{p_{jk}}[i]$, $i \in \mathbb{N}_L = \{1, 2, \dots, L\}$, indicating how many units of resource \mathcal{R}_i are required to support the execution of the stage denoted by p_{jk} . For systems whose each process can be converted to state machines, S^4R can well handle them. Nevertheless, for more complex scenarios, such as free choice multiple reentrant flowlines, they are beyond S^4R 's description capability and deserve further investigation. Also, our method is not appropriate to tackle with systems exhibiting fast reconfiguration and unliable resources.

A. S^4R Models

Definition 1: An S^4R is a strongly-connected generalized pure Petri net $N = (P, T, F, W)$ where: 1) $P = P_0 \cup P_A \cup P_R$ is a partition such that: a) P_0 , P_A , and P_R are called idle, operation (or activity), and resource places, respectively; b) $P_0 = \cup_{i \in \mathbb{N}_K} \{p_{0i}\}$; c) $P_A = \cup_{i \in \mathbb{N}_K} P_{A_i}$, where for each $i \in \mathbb{N}_K$, $P_{A_i} \neq \emptyset$, and for each $i, j \in \mathbb{N}_K$, $i \neq j$, $P_{A_i} \cap P_{A_j} = \emptyset$; and d) $P_R = \{r_1, r_2, \dots, r_n\}$, $n > 0$. 2) $T = \cup_{i \in \mathbb{N}_K} T_i$, where for each $i \in \mathbb{N}_K$, $T_i \neq \emptyset$, and for each $i, j \in \mathbb{N}_K$, $i \neq j$, $T_i \cap T_j = \emptyset$. 3) For each $i \in \mathbb{N}_K$, subnet $\bar{N}_i = N | (\{p_{0i}\} \cup P_{A_i}, T_i)$ is a strongly connected state machine such that every cycle contains p_{0i} . 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 $X_r(r) = 1$ where $\mathbb{N}^{|P|}$ means $|P|$ -dimensional vectors whose each component belongs to \mathbb{N} . 5) $P_A = \cup_{r \in P_R} (\|X_r\| \setminus \{r\})$.

In N , $\forall p_0 \in P_0$, $M_0(p_0)$ indicates the upper bound of the maximum number of products that are allowed to be concurrently manufactured in a process initialized by p_0 . $\forall p \in P_A$, $M(p) > 0$ means ongoing operations modeled by p . $\forall r \in P_R$, $M_0(r)$ denotes the capacity of a resource r . From Definition 1, S^4R is evidently conservative and consistent. S^4R can model a set of concurrently-progressing types of parts. For products with the same type, they share the same processing route. The whole model is a composition of different processes through their sharing resources. In S^4R , each process exhibits routing flexibility as well as each stage allows multiple-resource acquisition. For more details, please refer to [31] and [33].

Definition 2: M_0 is an acceptable initial marking in N if (1) $M_0(p_0) \geq 1$, $\forall p_0 \in P_0$; (2) $M_0(p) = 0$, $\forall p \in P_A$; and (3) $M_0(r) \geq X_r(p)$, $\forall r \in P_R$, $\forall p \in P_A$.

Given an arbitrary marking $M \in R(N, M_0)$, a transition t is process-enabled if $M(\bullet t \cap P_A) > 0$. Note that $|\bullet t \cap P_A| = 1$ by definition. Correspondingly, t is resource-enabled by $\forall r \in \bullet t \cap P_R$ if $M(r) \geq W(r, t)$. In the rest of this paper, (N, M_0) is an acceptably marked S^4R .

Definition 3: Let $r \in P_R$ be a resource place in (N, M_0) . The set of holders of r is the support of a minimal P -semiflow X_r without r , i.e., $H(r) = \|X_r\| \setminus \{r\}$. Clearly, $H(r)$ contains only operation places due to $\|X_r\| \cap P_R = \{r\}$.

Let $S_R = S \cap P_R$ and $S_A = S \cap P_A$.

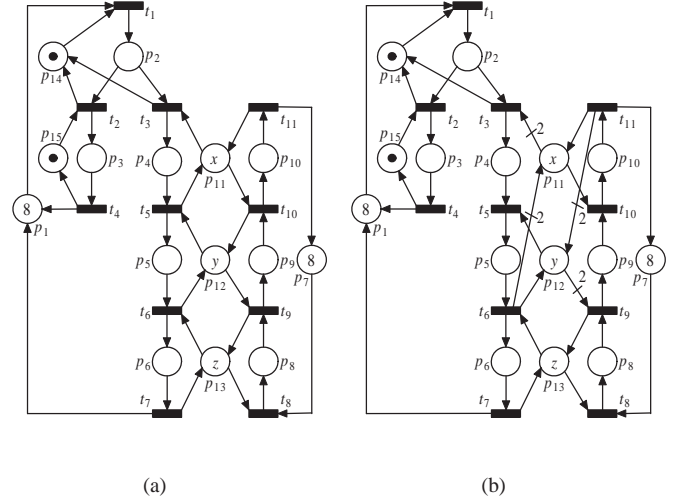


Fig. 2. Two example S^4R nets. (a) Net model where $x = z = 2$ and $y = 1$; (b) Net model where $x = z = 4$ and $y = 2$.

Definition 4: Let S be a siphon that can be undermarked in (N, M_0) . Token takers, denoted by $\tilde{H}(S)$, are the places that correspond to the holders of the resources in S but do not belong to S .

Suppose $H_{S_R} = \cup_{r \in S_R} H(r)$. We have $\tilde{H}(S) = H_{S_R} \setminus S = (H_{S_R} \cap P_A) \setminus S_A$. The following condition for liveness of S^4R is presented in [33]. (N, M_0) is live if $\nexists M \in R(N, M_0)$ and an undermarked siphon S such that 1) $\forall r \in S_R$, $M(r) < W(r, t)$; 2) $\forall p \in S_A$, $M(p) = 0$; and 3) $\forall p \in \tilde{H}(S)$, $M(p) > 0$.

Thanks to their special structure, S^4R can describe AMS in which each product is manufactured via sequential and/or concurrent manufacturing processes. It is composed of a set of subsets \bar{N}_i , $i \in \mathbb{N}_K$, which are in one-to-one correspondence with a product and its related manufacturing processes. More specifically, each \bar{N}_i can be decomposed into an acyclic graph and an idle place p_{0i} . The operations together with their interactions required by a process are represented by the activity places and transitions involved in the respective acyclic graph of \bar{N}_i . A set of activity places with a same ingoing transition correspond to the initialization of a number of concurrently executed processes, while the ones with a same outgoing transition correspond to an assembly operation. The initial marking of an idle place p_{0i} corresponds to the number of products that are allowed in the system at a time. As a convention, p_{0i} is also designated to model repetitive production. Places in P_R are used to model various resource types. Their marking during the evolution of a Petri net corresponds to the number of available resources in the modeled AMS. In particular, their initial markings define the capacities of the corresponding resource types.

In the sequel, when mentioning (N, M_0) , we refer to an acceptably marked S^4R . Fig. 2 shows two typical S^4R nets. Obviously, these two Petri nets share the same process nets whereas their resource allocation mechanisms are different. The former allows only single unit of resource acquisition at each operation stage whereas the latter allows multiple ones. Therefore, the latter is more general. Owing to its popularity,

the former is presented as a benchmark.

Under the assumption that $P_0 = \{p_1, p_7\}$, $P_{A_1} = \{p_2 - p_6\}$, $P_{A_2} = \{p_8 - p_{10}\}$, $P_R = \{p_{11} - p_{15}\}$, Fig. 2 shows two S^4R representing AMS consisting of five resource types $\mathcal{R}_1 - \mathcal{R}_5$ with capacities $C_1 = C_3 = 2$, $C_2 = C_4 = C_5 = 1$, and supporting two job types \mathcal{J}_1 and \mathcal{J}_2 . Job type \mathcal{J}_1 (resp., \mathcal{J}_2) is defined by the set of partially ordered job stages $\{p_1 - p_6\}$ (resp., $\{p_7 - p_{10}\}$). The conjunctive resource requirements associated with various job stages in the net in Fig. 2(a) (resp., Fig. 2(b)) are as follows: $a_{p_2} = p_4$, $a_{p_3} = p_5$, $a_{p_4} = p_{11}$, $a_{p_5} = p_{12}$, $a_{p_6} = p_{13}$, $a_{p_8} = p_{13}$, $a_{p_9} = p_{12}$, and $a_{p_{10}} = p_{11}$ (resp., $a_{p_2} = p_{14}$, $a_{p_3} = p_{15}$, $a_{p_4} = 2p_{11}$, $a_{p_5} = 2p_{11} + p_{12}$, $a_{p_6} = p_{13}$, $a_{p_8} = p_{13}$, $a_{p_9} = 2p_{12}$, and $a_{p_{10}} = p_{11} + 2p_{12}$). The P -semiflows corresponding to the resources are: $I_1 = p_4 + p_{10} + p_{11}$, $I_2 = p_5 + p_9 + p_{12}$, $I_3 = p_6 + p_8 + p_{13}$, $I_4 = p_2 + p_{14}$, and $I_5 = p_3 + p_{15}$ (resp., $I_1 = 2p_4 + 2p_5 + p_{10} + p_{11}$, $I_2 = p_5 + 2p_9 + 2p_{10} + p_{12}$, $I_3 = p_6 + p_8 + p_{13}$, $I_4 = p_2 + p_{14}$, and $I_5 = p_3 + p_{15}$).

B. Liveness-enforcing Supervisory Control Using GMEC

In Petri nets, many approaches have been proposed to forbid inadmissible states such that the remaining states satisfy a constraint $l^T \cdot M \leq b$, where $l^T \in \mathbb{N}^{1 \times |P|}$, $b \in \mathbb{N}^+$.

A control specification on a net system (N, M_0) can be expressed in the form of $l^T \cdot M \leq b$, where M means a reachable marking, while l and b are an integer vector and a scalar, respectively. To implement it, a monitor p_c should be superimposed on the net structure according to the following incidence matrix $[N_{p_c}] = -l^T \cdot [N]$. Correspondingly, the initial marking of p_c must be set as $M_0(p_c) = b - l^T \cdot M_0$. Obviously, the problem is feasible only if $b - l^T \cdot M_0 > 0$.

According to [2], the liveness of an S^4R can be ensured when all of its siphons are properly controlled during the system evolution. Therefore, the basic idea to enforce its liveness is to prevent siphons from becoming undermarked by the addition of monitors. Hereby, we adopt max -control, which is a special SCT initially proposed in [2]. For the self-completeness of this paper, we cite the following definitions in [2]. In the sequel, when talking about a siphon, we mean a strict minimal one.

Definition 5: [2] A siphon S is said to be max -marked at $M \in R(N, M_0)$ if $\exists p \in S$ such that $M(p) \geq max_{p \bullet}$, where $max_{p \bullet} = \max_{t \in p \bullet} W(p, t)$.

Definition 6: [2] A siphon S is said to be max -controlled if S is max -marked at any reachable marking, i.e., $\forall M \in R(N, M_0)$, $\exists p \in S$ such that $M(p) \geq max_{p \bullet}$.

Definition 7: [2] (N, M_0) is said to be satisfying the max controlled-siphon property (max cs-property, for short) if each siphon of (N, M_0) is max -controlled.

Proposition 1: [2] S is a siphon. If there exists a P -invariant I such that $\forall p \in \|I\|^- \cap S$, $max_{p \bullet} = 1$, $\|I\|^+ \subseteq S$, and $\sum_{p \in P} I(p) \cdot M_0(p) > \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1)$, then S is max -controlled.

S is an undermarked siphon at $M \in R(N, M_0)$. (N^*, M_0^*) , $N^* = (P_0 \cup P_A \cup P_R \cup \{p_c\}, T, F^*, W^*)$ is an augmented net to control S through the addition of a monitor p_c such that $F^* = F \cup (\{p_c\} \times p_c^\bullet) \cup (\bullet p_c \times \{p_c\})$, and $\forall p \in P_0 \cup P_A \cup P_R$, $M_0^*(p) = M_0(p)$.

From Proposition 1 and the aforementioned statements, we should construct a P -invariant (not necessarily a P -semiflow). First, we establish two P -semiflows. One is $g_S = \sum_{r \in S_R} X_r$. Another one is introduced by assuming a monitor p_c which leads to another P -semiflow such that $\forall p \in \|g_S\| \setminus S$, $h_S(p) = g_S(p)$, $h_S(p_c) = 1$, and $\|h_S\| \setminus \{p_c\} \subseteq P_A$.

Lemma 1: Given an augmented net (N^*, M_0^*) with monitor p_c . $I_S = g_S - h_S$ is a P -invariant in N^* .

Proof: Since g_S and h_S are P -semiflows, we have $g_S^T \cdot [N^*] = \mathbf{0}^T$ and $h_S^T \cdot [N^*] = \mathbf{0}^T$. Therefore, $I_S^T \cdot [N^*] = (g_S^T - h_S^T) \cdot [N^*] = g_S^T \cdot [N^*] - h_S^T \cdot [N^*] = \mathbf{0}^T$. ■

Lemma 2: Let $I_S = g_S - h_S$. S is max -controlled if $\sum_{p \in P} I_S(p) \cdot M_0(p) > \sum_{p \in S} I_S(p) \cdot (max_{p \bullet} - 1)$ holds.

Proof: Thanks to the construction of I_S , $\|I_S\|^+ = \|g_S\| \setminus \|h_S\| \subseteq S$, $\|I_S\|^- = \{p_c\}$, and $\|I_S\|^- \cap S = \emptyset$. Based on Proposition 1, S is max -controlled when $\sum_{p \in P} I_S(p) \cdot M_0(p) > \sum_{p \in S} I_S(p) \cdot (max_{p \bullet} - 1)$ holds. ■

In order to develop a liveness-enforcing supervisor for S^4R , one can add a monitor for each detected undermarked siphon S .

Definition 8: ϑ is called S 's control P -vector if: 1) ϑ is a $|P|$ -vector; 2) $\|\vartheta\| = \|g_S\| - \|g_S\| \cap \|h_S\|$ such that $\vartheta(p) = g_S(p)$, $\forall p \in \|g_S\| - \|g_S\| \cap \|h_S\|$; $\vartheta(p) = 0$, otherwise.

Definition 9: l is called S 's monitor P -vector if: 1) l is a $|P|$ -vector; and 2) $\|l\| = \|h_S\| - \{p_c\}$ such that $l(p) = h_S(p)$, $\forall p \in \|h_S\| - \{p_c\}$; $l(p) = 0$, otherwise.

Proposition 2: For ϑ and l , we have 1) $\|\vartheta\| + \|l\| = \|g_S\|$; 2) $\|\vartheta\| \cap \|l\| = \emptyset$; and 3) $\vartheta + l = g_S$.

Proof: 1) $\|\vartheta\| + \|l\| = \|g_S\| - (\|g_S\| \cap \|h_S\|) + (\|g_S\| \cap \|h_S\|) = \|g_S\|$.

2) By contradiction, we suppose that $\exists p$, such that $p \in \|\vartheta\| \wedge p \in \|l\|$. This means $p \in \|h_S\|$ and $p \in \|g_S\| - \|h_S\|$. This is impossible because $p \in \|h_S\| \cap (\|g_S\| - \|h_S\|) = \emptyset$.

3) This point is due to the construction of l and ϑ . ■

The above lemmas along with propositions suffice to convert the control of a siphon to the enforcement of a GMEC, which is actually in the form of an inequality.

Theorem 1: Given (N, M_0) , S is max -controlled if $\forall M \in R(N, M_0)$, $\vartheta^T \cdot M \geq \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1) + 1$.

Proof: Based on Proposition 1, S is max -controlled if $\sum_{p \in P} I_S(p) \cdot M_0(p) > \sum_{p \in S} I_S(p) \cdot (max_{p \bullet} - 1)$. Since $I_S = g_S - h_S$, we have $\sum_{p \in \|g_S\| - \|g_S\| \cap \|h_S\|} g_S(p) \cdot M_0(p) - M_0(p_c) > \sum_{p \in S} I_S(p) \cdot (max_{p \bullet} - 1)$. Consider Definition 8. We have $\vartheta^T \cdot M_0 - M_0(p_c) > \sum_{p \in S} I_S(p) \cdot (max_{p \bullet} - 1)$. According to [28], this implies a ‘‘greater than and equal to’’ inequality which is implemented by a monitor p_c . That is $\vartheta^T \cdot M \geq \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1) + 1$. ■

Corollary 1: Given (N, M_0) , $\vartheta^T \cdot M \geq \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1) + 1$ is equivalent to $l^T \cdot M \leq \sum_{p \in S} M_0(S_R) - \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1) - 1$.

Proof: According to Proposition 2, we have $\vartheta + l = g_S$ where $g_S = \sum_{r \in S_R} X_r$ is a P -invariant. Therefore, we have $\vartheta^T \cdot M + l^T \cdot M = (\vartheta^T + l^T) \cdot M = g_S^T \cdot M = g_S^T \cdot M_0 = M_0(S_R)$. As a result, $\vartheta^T \cdot M = M_0(S_R) - l^T \cdot M$. $\vartheta^T \cdot M \geq \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1) + 1$ is thus equivalent to $M_0(S_R) - l^T \cdot M \geq \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1) + 1$. Through transformation, we have $l^T \cdot M \leq M_0(S_R) - \sum_{p \in S} I(p) \cdot (max_{p \bullet} - 1) - 1$. ■

TABLE I
MONITOR PARAMETERS FOR PETRI NET IN FIG. 2(a)

i	S_i	g_{S_i}	h_{S_i}	l_i	b_i
1	$\{p_5, p_{10}, p_{11}, p_{12}\}$	$p_4 + p_5 + p_9 + p_{10} + p_{11} + p_{12}$	$p_4 + p_9 + p_c$	$p_4 + p_9$	2
2	$\{p_6, p_9, p_{12}, p_{13}\}$	$p_5 + p_6 + p_8 + p_9 + p_{12} + p_{13}$	$p_5 + p_8 + p_c$	$p_5 + p_8$	2
3	$\{p_6, p_{10}, p_{11}, p_{12}, p_{13}\}$	$p_4 + p_5 + p_6 + p_8 + p_9 + p_{10} + p_{11} + p_{12} + p_{13}$	$p_4 + p_5 + p_8 + p_9 + p_c$	$p_4 + p_5 + p_8 + p_9$	4

TABLE II
MONITOR PARAMETERS FOR PETRI NET IN FIG. 2(b)

i	S_i	g_{S_i}	h_{S_i}	l_i	b_i
1	$\{p_5, p_{10}, p_{11}, p_{12}\}$	$2p_4 + 3p_5 + 2p_9 + 3p_{10} + p_{11} + p_{12}$	$2p_4 + 2p_9 + p_c$	$2p_4 + 2p_9$	3
2	$\{p_6, p_9, p_{10}, p_{12}, p_{13}\}$	$p_5 + p_6 + p_8 + 2p_9 + 2p_{10} + p_{12} + p_{13}$	$p_5 + p_8 + p_c$	$p_5 + p_8$	4
3	$\{p_6, p_{10}, p_{11}, p_{12}, p_{13}\}$	$2p_4 + p_5 + p_6 + p_8 + 2p_9 + 3p_{10} + p_{11} + p_{12} + p_{13}$	$2p_4 + 3p_5 + p_8 + 2p_9 + p_c$	$2p_4 + 3p_5 + p_8 + 2p_9$	7

Evidently, a siphon can be prevented from being undermarked by simply applying a specification $l^T \cdot M \leq M_0(S_R) - \sum_{p \in S} I(p) \cdot (\max_{p \bullet} - 1) - 1$ to the net. For the sake of brevity, we assume a scalar $b = M_0(S_R) - \sum_{p \in S} I(p) \cdot (\max_{p \bullet} - 1) - 1$ in the sequel.

Based on Corollary 1, each siphon can be *max*-controlled through a GMEC denoted by a pair (l, b) . The latter can be easily identified through the structural analysis of an S^4R . An S^4R system becomes live when all siphons are *max*-controlled with (l, b) . For more detail, please refer to [2], [31], [33]. In the sequel, we assume all siphons are produced with an enumeration method.

Take the net shown in Fig. 2(a) as an example. $S = \{p_5, p_{10} - p_{12}\}$ is a siphon. Evidently, we have $g_S = p_4 + p_5 + p_9 + p_{10} + p_{11} + p_{12}$, $h_S = p_4 + p_9 + p_c$, and $I_S = g_S - h_S = p_5 + p_{10} + p_{11} + p_{12} - p_c$. As a result, we have $\vartheta_S = p_5 + p_{10} + p_{11} + p_{12}$ and $l_S = p_4 + p_9$. $M_0(S_R) = M_0(p_{11}) + M_0(p_{12}) = 2 + 1 = 3$. $\sum_{p \in S} I(p) \cdot (\max_{p \bullet} - 1) = (\max_{p_5 \bullet} - 1) + (\max_{p_{10} \bullet} - 1) + (\max_{p_{11} \bullet} - 1) + (\max_{p_{12} \bullet} - 1) = (1 - 1) + (1 - 1) + (1 - 1) + (1 - 1) = 0$. Therefore, $b = M_0(S_R) - \sum_{p \in S} I(p) \cdot (\max_{p \bullet} - 1) - 1 = 3 - 0 - 1 = 2$. After a pair (l, b) is obtained, a monitor p_c along with its outgoing and ingoing arcs can be calculated with the technique in [28]. Tables I and II show the corresponding S , g_S , h_S , I_S , ϑ , l , and b in the Petri nets shown in Figs. 2(a) and (b), respectively.

IV. SUPERVISOR SIMPLIFICATION VIA INEQUALITY ANALYSIS

The above analysis shows that the liveness supervision in the framework of Petri nets can be converted to the satisfaction of a set of inequalities. These inequalities constitute a linear system that restricts the behavior of the plant model. However, one must notice that such a set of inequalities is not minimal. In other words, some inequalities may be redundant, implying that some inequalities are dependent on the others. As one inequality corresponds to one monitor, the removal of those dependent ones can reduce the size of the final supervisor. Therefore, it is attractive to identify a way to remove them while preserving the independent ones. The separation between them might not be unique. Some selection techniques should be developed to properly make their identification and distinction.

A. Identification of Independent and Dependent Inequalities

Suppose $L = [l_1 \ l_2 \ \dots \ l_n]$ and $B = [b_1 \ b_2 \ \dots \ b_n]^T$. $L^T \cdot M \leq B$ means n GMECs. Among them, some are dependent on others.

Definition 10: Let $L^T \cdot M \leq B$ be a set of inequalities, $\mathcal{M} = \{M | l_i^T \cdot M \leq b_i, \forall i \in \mathbb{N}_n\}$, and $\mathcal{M}_{\mathbb{N}_n \setminus \{k\}} = \{M | l_i^T \cdot M \leq b_i, \forall i \in \mathbb{N}_n - \{k\}\}$. $l_k^T \cdot M \leq b_k$ is said to be dependent on other inequalities iff $\mathcal{M} = \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}$.

Proposition 3: An inequality $l_k^T \cdot M \leq b_k$ is dependent on the others iff $\min\{b_k - l_k^T \cdot M, M \in \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}\} \geq 0$.

Proof: For the necessary part, we have $\mathcal{M} = \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}$ according to Definition 10. Since $l_k^T \cdot M \leq b_k$ holds in the space determined by \mathcal{M} according to the hypothesis of this proposition, it is apparent that $l_k^T \cdot M \leq b_k$ holds in $\mathcal{M}_{\mathbb{N}_n \setminus \{k\}}$. This means that $\min\{b_k - l_k^T \cdot M, M \in \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}\} \geq 0$.

For the sufficient part, we have $l_k^T \cdot M \leq b_k$ because of $\min\{b_k - l_k^T \cdot M, M \in \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}\} \geq 0$. This means $\mathcal{M}_{\mathbb{N}_n \setminus \{k\}}$ implicitly ensures $l_k^T \cdot M \leq b_k$. Moreover, the combination of $\mathcal{M}_{\mathbb{N}_n \setminus \{k\}}$ and $l_k^T \cdot M \leq b_k$ is equivalent to \mathcal{M} . As a result, we have $\mathcal{M} = \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}$. ■

Proposition 3 implies that the dependance of one inequality upon others can be determined by solving a mathematical programming problem:

$$\min \quad b_k - l_k^T \cdot M \quad (1)$$

subject to

$$l_i^T \cdot M \leq b_i, \quad i \in \mathbb{N}_n - \{k\} \quad (2)$$

Corollary 2: An inequality $l_k^T \cdot M \leq b_k$ is independent iff $\min\{b_k - l_k^T \cdot M, M \in \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}\} < 0$.

Proof: Owing to its duality to Proposition 3, this statement holds obviously. ■

Consider the three inequalities in Fig. 2(a). To verify one's dependency upon the other two, we can have the following three mathematical programming formulations.

For $M(p_4) + M(p_9) \leq 2$, we have

$$\min \quad 2 - M(p_4) - M(p_9)$$

subject to

$$M(p_5) + M(p_8) \leq 2$$

$$M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$$

For $M(p_2) + M(p_8) \leq 2$, we have

$$\min 2 - M(p_5) - M(p_8)$$

subject to

$$M(p_4) + M(p_9) \leq 2$$

$$M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$$

For $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$, we have

$$\min 4 - M(p_4) - M(p_5) - M(p_8) - M(p_9)$$

subject to

$$M(p_4) + M(p_9) \leq 2$$

$$M(p_5) + M(p_8) \leq 2$$

Their solutions are -2 , -2 , and 0 , respectively. This implies that $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$ are independent of the remaining inequalities whereas $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ is dependent on $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$. However, for the first two sets of formulations, we can substitute $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ by $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 2$. Note that when the latter is satisfied, the former is of course satisfied. The objectives become 0 , which implies that $M(p_4) + M(p_9) \leq 2$ and $M(p_2) + M(p_8) \leq 2$ can become dependent constraints if we reduce the right-hand scalar of their independent one, i.e., $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 2$. In general, the right-hand-side scalar reduction results a more restrictive constraint that realizes the original one in a less permissive way. On the other hand, this can be well used to reduce the size of a supervisor at the likely sacrifice of behavior permissiveness.

According to the above statement, we need to establish and solve a set of mathematical programming formulations to verify the dependency of each inequality with respect to others. As known, it can be quite time-consuming to solve them. Moreover, in the case that the dependency of some inequalities does not hold, one may resort to decreasing the right-hand scalars with regard to the independent inequalities. Nevertheless, there is no unified principle to follow. In the following, we present a quite efficient method to tackle such issue.

Theorem 2: Let $L^T \cdot M \leq B$ be a set of inequalities and $k \in \mathbb{N}_n$. $l_k^T \cdot M \leq b_k$ is dependent on the others iff there exist $n - 1$ nonnegative coefficients α_i , $i \in \mathbb{N}_n \setminus \{k\}$ such that $l_k \leq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot l_i$ and $b_k \geq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot b_i$.

Proof: For the necessary part, we assume that $l_k^T \cdot M \leq b_k$ is dependent on the others. According to Proposition 3, we have $\min\{b_k - l_k^T \cdot M, M \in \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}\} \geq 0$. Based on the duality theorem, this is equivalent to the statement $\max\{l_k^T \cdot M - b_k, M \in \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}\} \leq 0$. This implies that there exists an optimal solution M^* such that $l_k^T \cdot M^* \leq b_k$. The dual mathematical programming is $\min\{\sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot b_i, \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot l_i \geq l_k, \alpha_i \geq 0, \forall i \in \mathbb{N}_n \setminus \{k\}\}$. As a result, there exists an optimal solution α_i , $i \in \mathbb{N}_n \setminus \{k\}$ such that $\sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot b_i = l_k^T \cdot M \leq b_k$, $\sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot l_i \geq l_k$.

For the sufficient part, we have $l_k \leq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot l_i$ and $b_k \geq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot b_i$ according to the hypothesis. For

any $M \geq 0$, we have $l_k^T \cdot M \leq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot l_i^T \cdot M \leq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot b_i$. According to Definition 10, we know that $l_k^T \cdot M \leq b_k$ is dependent on other inequalities, i.e., $l_k^T \cdot M \leq b_k$, where $i \in \mathbb{N}_n \setminus \{k\}$. ■

In Fig. 2(a), $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ is dependent on the other two, i.e., $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$. This is because $(M(p_4) + M(p_9)) + (M(p_5) + M(p_8)) \leq M(p_4) + M(p_5) + M(p_8) + M(p_9)$ and $4 \geq 2 + 2$. $M(p_4) + M(p_9) \leq 2$ is not dependent on $M(p_5) + M(p_8) \leq 2$ and $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$. This is because, despite the fact that $M(p_5) + M(p_8) \leq M(p_4) + M(p_5) + M(p_8) + M(p_9)$, we have $2 \leq 4$. An important and interesting issue is that we can replace the scalar 4 with 2 so that $2 \leq 2$ holds, which makes $M(p_4) + M(p_9) \leq 2$ dependent on $M(p_5) + M(p_8) \leq 2$ and $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$. The same analysis applies to the dependency of $M(p_5) + M(p_8) \leq 2$ on the other.

In Fig. 2(b), $2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9) \leq 7$ is independent of the other two, i.e., $2M(p_4) + 2M(p_9) \leq 3$ and $M(p_5) + M(p_8) \leq 4$. This is because, although $(2M(p_4) + 2M(p_9)) + 3(M(p_5) + M(p_8)) \geq 2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9)$, we have $7 \leq 3 \times 4 + 3$. To make $2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9) \leq 7$ dependent, we can decrease 4 to 1 such that $7 \leq 3 \times 1 + 3$. $2M(p_4) + 2M(p_9) \leq 3$ is not dependent on $M(p_5) + M(p_8) \leq 4$ and $2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9) \leq 7$. This is because, despite the fact that $2p_4 + 2p_9 \leq 2p_4 + 3p_5 + p_8 + 2p_9$, we have $3 \leq 7$. Similarly, we can decrease the right-hand scalar 7 to 3 such that $3 \leq 3$ holds, which makes $2M(p_4) + 2M(p_9) \leq 3$ dependent on $M(p_5) + M(p_8) \leq 4$ and $2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9) \leq 3$. The same analysis applies to the dependency of $M(p_5) + M(p_8) \leq 4$ on the other.

B. Analysis of Independent and Dependent Inequalities

It is noticed that the above analysis presents a simple procedure to derive the dependency relationship between one inequality and the others. An interesting issue is that the independent inequalities are actually not unique. As shown by the examples, two independent inequalities may result when different sets of inequalities are determined as the dependent ones. With the aid of a multiset technique, we herein present some structural analysis of these inequalities and their dependency relationship.

Proposition 4: Let $L^T \cdot M \leq B$ be a set of inequalities and $k \in \mathbb{N}_n$. If $l_k^T \cdot M \leq b_k$ is dependent on other inequalities, $\|l_k\| \subseteq \cup_{i \in \mathbb{N}_n \setminus \{k\}} \|l_i\|$.

Proof: We can prove it by contradiction. Suppose that l_k is dependent on l_i where $i \in \mathbb{N}_n \setminus \{k\}$ and $\exists p \in \|l_k\|$ so that $t \notin \cup_{i \in \mathbb{N}_n \setminus \{k\}} \|l_i\|$. Then, we have $l_k(p) \geq 1$ and $\sum_{i \in \mathbb{N}_n} l_i(p) = 0$. As a result, we have $l_k(p) > \sum_{i \in \mathbb{N}_n} l_i(p)$, leading to the fact that $l_k \not\leq \sum_{i \in \mathbb{N}_n} l_i(p)$. This is in contradiction with the hypothesis that l_k is dependent on others, which requires that $l_k \leq \sum_{i \in \mathbb{N}_n} l_i(p)$. ■

In Fig. 2(a), $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$ cannot be dependent on each other because neither $\{p_4, p_9\} \subseteq \{p_5, p_8\}$ nor $\{p_5, p_8\} \subseteq \{p_4, p_9\}$ is true. Both $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$ are dependent on $M(p_4)$

$+ M(p_5) + M(p_8) + M(p_9) \leq 4$ because $\{p_4, p_9\} \subseteq \{p_4, p_5, p_8, p_9\}$ and $\{p_5, p_8\} \subseteq \{p_4, p_5, p_8, p_9\}$. $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ is not dependent on either $M(p_4) + M(p_9) \leq 2$ or $M(p_5) + M(p_8) \leq 2$ because $\{p_4, p_5, p_8, p_9\} \not\subseteq \{p_4, p_9\}$ and $\{p_4, p_5, p_8, p_9\} \not\subseteq \{p_5, p_8\}$. Nevertheless, $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ is dependent on the union of $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$ because $\{p_4, p_5, p_8, p_9\} \subseteq \{p_4, p_9\} \cup \{p_4, p_5, p_8, p_9\}$. To be concise, the analysis upon the Petri net model in Fig. 2(b) is omitted owing to its similarity to the case in Fig. 2(a).

Theorem 3: Let $L^T \cdot M \leq B$ be a set of inequalities and $k \in \mathbb{N}_n$. $l_k^T \cdot M \leq b_k$ is dependent on other inequalities iff there exist $n - 1$ nonnegative coefficients α_i , $i \in \mathbb{N}_n \setminus \{k\}$ such that $\|l_k\| \leq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot \|l_i\|$ and $b_k \geq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot b_i$.

Proof: This proof is a multiset version of the proof of Theorem 2. Owing to their similarity, it is omitted. ■

C. Supervisor Simplification

Suppose $\mathcal{L} = \{\|l_i\|\}$, where $i \in \mathbb{N}_n$. From the above analysis, the key issue is to identify $\Omega = \{\omega_1, \omega_2, \dots, \omega_m\} \subseteq \mathbb{N}_n$ leading to two sets, i.e., $\mathcal{L}_{\mathcal{I}} = \{\|l_i\|\}_{i \in \Omega} \subseteq \mathcal{L}$ and $\mathcal{L}_{\mathcal{D}} = \mathcal{L} \setminus \mathcal{L}_{\mathcal{I}}$ such that: 1) $\forall \omega_k \in \Omega$, $\nexists \alpha_i \geq 0$ such that $l_{\omega_k} = \sum_{i \in \Omega \setminus \{\omega_k\}} \alpha_{\omega_i} \cdot l_{\omega_i}$; and 2) $\forall \mu_k \in \mathbb{N}_n \setminus \Omega$, $\exists \alpha_i \geq 0$, $l_{\mu_k} \leq \sum_{i \in \mathbb{N}_m} \alpha_i \cdot l_{\omega_i}$. Obviously, $\mathcal{L}_{\mathcal{I}}$ is not unique.

Definition 11: Let $i \in \Omega$, $j \in \mathbb{N}_n \setminus \Omega$. $\{\|l_{\omega_1}\|, \|l_{\omega_2}\|, \dots, \|l_{\omega_m}\|\}$ is called a *max- $\mathcal{L}_{\mathcal{I}}$* (resp., *min- $\mathcal{L}_{\mathcal{I}}$*) if $\forall i, j$, $\|l_{\omega_i}\| \geq \|l_{\omega_j}\|$ (resp., $\|l_{\omega_i}\| \leq \|l_{\omega_j}\|$).

As opposed to \mathcal{L} , we have $\mathcal{B} = \{b_i\}$, where $i \in \mathbb{N}_n$; $\mathcal{B}_{\mathcal{I}} = \{b_i\}$, where $i \in \Omega$; and $\mathcal{B}_{\mathcal{D}} = \{b_i\}$, where $i \in \mathbb{N}_n \setminus \Omega$. $L_{\mathcal{D}}$, $B_{\mathcal{D}}$, and $B_{\mathcal{I}}$ denote vectors while their corresponding multisets are represented by $\mathcal{L}_{\mathcal{D}}$, $\mathcal{B}_{\mathcal{D}}$, $\mathcal{B}_{\mathcal{I}}$.

Take the Petri net model in Fig. 2(a) as an example. $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ is a *max- $\mathcal{L}_{\mathcal{I}}$* while $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$ are a *min- $\mathcal{L}_{\mathcal{I}}$* .

Therefore, to reduce the supervisor size, our first step is to identify these dependent inequalities, which can be realized by the following algorithm.

Algorithm 1: Identification of Independent and Dependent Inequalities

Input: $L^T \cdot M \leq B$

Output: Independent Inequalities $L_{\mathcal{I}}^T \cdot M \leq B_{\mathcal{I}}$ and Dependent Inequalities $L_{\mathcal{D}}^T \cdot M \leq B_{\mathcal{D}}$ with $L_{\mathcal{I}} = [l_{\omega_1} \ l_{\omega_2} \ \dots \ l_{\omega_m}]$, $B_{\mathcal{I}} = [b_{\omega_1} \ b_{\omega_2} \ \dots \ b_{\omega_m}]$, $L_{\mathcal{D}} = [l_{\mu_1} \ l_{\mu_2} \ \dots \ l_{\mu_{n-m}}]$, and $B_{\mathcal{D}} = [b_{\mu_1} \ b_{\mu_2} \ \dots \ b_{\mu_{n-m}}]$ where $\{\mu_1, \mu_2, \dots, \mu_{n-m}\} \subseteq \mathbb{N}_n \setminus \{\omega_1, \omega_2, \dots, \omega_m\}$.

- 1) $i = 1$, $m = 1$, $\mathcal{L}_{\mathcal{I}} := \emptyset$, $\mathcal{L}_{\mathcal{D}} := \mathcal{L}$, $\mathcal{B}_{\mathcal{I}} := \emptyset$, and $\mathcal{B}_{\mathcal{D}} := \mathcal{B}$;
- 2) Arrange all the elements in \mathcal{L} according to the ascending (resp., descending) order of $\|l\|$ for the *min- $\mathcal{L}_{\mathcal{I}}$* (resp., *max- $\mathcal{L}_{\mathcal{I}}$*), respectively;
- 3) **while** ($i \leq |\mathcal{L}|$) **do**
begin
- 4) Check whether $\exists \alpha_j \geq 0$ so that $l_i \leq \sum_{j=1}^m \alpha_j \cdot l_{\omega_j}$. If so, go to Step 6). Otherwise, go to Step 5);
- 5) $m := m + 1$, $\mathcal{L}_{\mathcal{I}} := \mathcal{L}_{\mathcal{I}} \cup \{l_i\}$, $\mathcal{L}_{\mathcal{D}} := \mathcal{L}_{\mathcal{D}} \setminus \{l_i\}$, $\mathcal{B}_{\mathcal{I}} := \mathcal{B}_{\mathcal{I}} \cup \{b_i\}$, and $\mathcal{B}_{\mathcal{D}} := \mathcal{B}_{\mathcal{D}} \setminus \{b_i\}$;

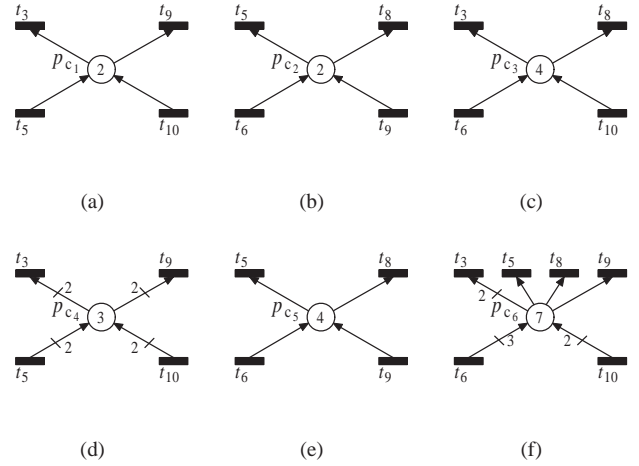


Fig. 3. The synthesized supervisors.

6) $i := i + 1$;
end

Based on Algorithm 1, the inequality set is divided into two disjoint sets, i.e., $L_{\mathcal{I}}$ and $L_{\mathcal{D}}$. The computational complexity is polynomial with regard to the number of inequalities. Using Theorem 3, we can decrease the right-hand scalars such that the ones in $L_{\mathcal{D}}$ can be ignored during the supervisor synthesis process. Algorithm 1 distinguishes the independent and dependant inequalities without resorting to integer programming techniques whose computational complexity proves to be exponential. Its execution involves only a limited number of comparison between some coefficient vectors and scalars.

Algorithm 2: Supervisor Simplification

Input: $L^T \cdot M \leq B$

Output: $L_{\mathcal{I}} \cdot M \leq B_{\mathcal{I}}$

- 1) Identify $\mathcal{L}_{\mathcal{I}}$ and $\mathcal{L}_{\mathcal{D}}$ using Algorithm 1;
- 2) $i = 0$;
- 3) **while** ($i < n - m$) **do**
begin
- 4) $i := i + 1$;
- 5) Find $\alpha_{\omega_j} \geq 0$, $j \in \mathbb{N}_m$, such that $l_{\mu_i} \leq \sum_{j=1}^m \alpha_{\omega_j} \cdot l_{\omega_j}$;
- 6) Decrease b_{ω_j} , $j \in \mathbb{N}_m$, such that $b_{\mu_i} \geq \sum_{j=1}^m \alpha_{\omega_j} \cdot b_{\omega_j}$.
end

n , m , and $n - m$ denote the numbers of all, independent, and dependent inequalities, respectively. l_{μ_i} and b_{μ_i} (resp., l_{ω_j} and b_{ω_j}) are the coefficient vectors and scalars of the dependent (resp., independent) inequalities. In Fig. 2(a), $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$ are identified as *min- $\mathcal{L}_{\mathcal{I}}$* while $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ is identified as *max- $\mathcal{L}_{\mathcal{I}}$* . For Fig. 2(b), $2M(p_4) + 2M(p_9) \leq 3$ and $M(p_5) + M(p_8) \leq 4$ are identified as *min- $\mathcal{L}_{\mathcal{I}}$* while $2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9) \leq 7$ is identified as *max- $\mathcal{L}_{\mathcal{I}}$* . Fig. 3 represents all these monitors where p_{c_1} and p_{c_2} realizes $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$, p_{c_3} realizes $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$; $p_{c_4} - p_{c_6}$ realize $2M(p_4) + 2M(p_9) \leq 3$, $M(p_5) + M(p_8) \leq 4$, and $2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9) \leq 7$.

min- $\mathcal{L}_{\mathcal{I}}$ may lead to more, if not the most, permissive control control whereas *max- $\mathcal{L}_{\mathcal{I}}$* can result in a concise, if not the simplest, supervisor. For the Petri net model in

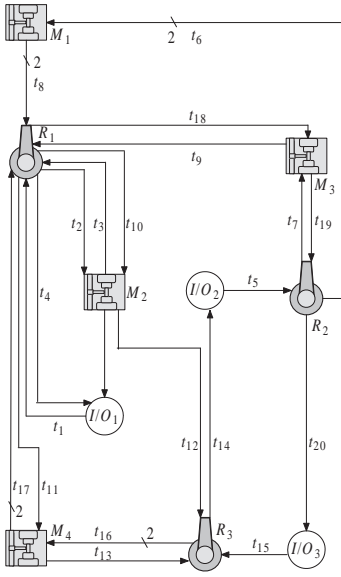


Fig. 4. The diagram block of an AMS.

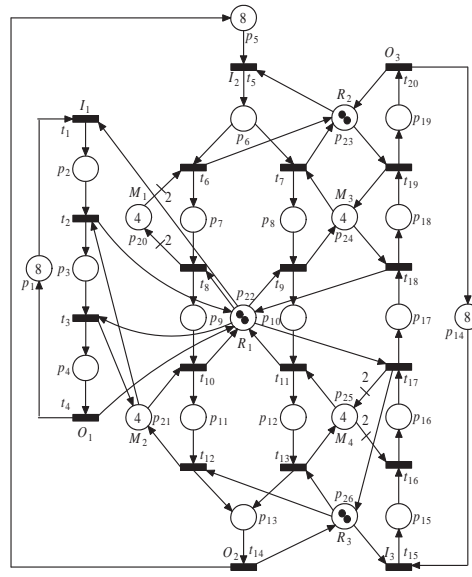


Fig. 5. The Petri net model of the AMS in Fig. 4.

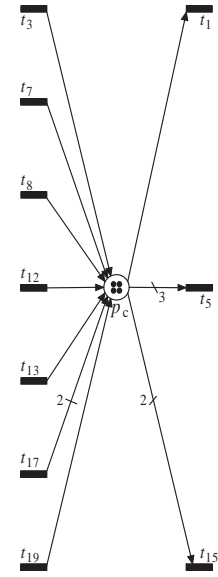


Fig. 6. The supervisor.

TABLE III
SUPERVISOR FOR PETRI NET IN FIG. 2(a)

Method	i	$M_0(p_{c_i})$	$\bullet p_{c_i}$	$p_{c_i}^{\bullet}$
Our Approach	1	2	$\{t_6, t_{10}\}$	$\{t_3, t_8\}$
	2	2	$\{t_6, t_{10}\}$	$\{t_3, t_8\}$
Approach in [26]	1	2	$\{t_5, t_{10}\}$	$\{t_3, t_9\}$
	2	2	$\{t_6, t_9\}$	$\{t_5, t_8\}$
Approach in [2]	1	2	$\{t_6, t_{10}\}$	$\{t_3, t_8\}$
	3	4	$\{t_6, t_{10}\}$	$\{t_3, t_8\}$

TABLE IV
SUPERVISOR FOR PETRI NET IN FIG. 2(b)

Method	i	$M_0(p_{c_i})$	$\bullet p_{c_i}$	$p_{c_i}^{\bullet}$
Our Approach	1	3	$\{3t_6, 2t_{10}\}$	$\{2t_3, t_5, t_8, t_9\}$
	2	3	$\{2t_5, 2t_{10}\}$	$\{2t_3, 2t_9\}$
Other Approach	2	4	$\{t_6, t_9\}$	$\{t_5, t_8\}$
	3	7	$\{2t_6, 2t_{10}\}$	$\{2t_3, t_5, t_8, t_9\}$

Fig. 2(a), the same behavior can be obtained whether the system is controlled by the first two inequalities or all the three ones. After the imposition of the first two inequalities, the third one's implicitness is obvious and its removal has no influence upon the system, thus leading to a maximal permissiveness with a simplified supervisor structure. If one only removes the inequalities whose removal has no influence upon the system behavior, the controlled system definitely can achieve the maximal permissiveness with a structurally compact supervisor.

For an S^4R , the above results indicate that the liveness enforcing supervisor can be synthesized in a quite compact way, thus avoiding astronomical number of monitors. Most of the existing approaches, such as [12], [17], are not applicable. Approaches in [2], [26] are feasible since they consider general nets. Moreover, 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 [26].

For both nets in Fig. 2, the method in [2] leads to 3 monitors, namely, $p_{c_1} - p_{c_3}$ since there are 3 siphons. These monitors constitute supervisors as shown in Tables III and IV, respectively.

In Fig. 2(a), among 3 siphons, 2 of them, are found to be elementary siphons according to [26]. Thus, 2 monitors are necessary to constitute a supervisor, as shown in Table III. In Fig. 2(b), no siphons are found to be elementary siphons. Therefore, the supervisor remains the same as in [2] and is shown in Table IV.

By contrast, our approach shows that only one monitor is necessary for each Petri net because inequalities $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 2$ and $2M(p_4) + 3M(p_5) + M(p_8) + 2M(p_9) \leq 3$ prove to be independent in Figs. 2(a) and (b), respectively. It shows that our approach can reduce the supervisor size more compared to approaches in [2] and [26]. For all supervisors, the net liveness can be achieved after their outgoing transitions move to p_0^{\bullet} as done in [12].

V. ILLUSTRATIVE EXAMPLE

Fig. 4 shows the block diagram of an AMS where three product types, i.e., $\mathcal{J}_1 - \mathcal{J}_3$, are manufactured. It is composed of three robots $\mathbf{R}_1 - \mathbf{R}_3$ and four machines $\mathbf{M}_1 - \mathbf{M}_4$. Each robot can hold two products. Each machine can deal with four products at a time. 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 $\mathbf{I}_1, \mathbf{O}_1, \mathbf{M}_1, \mathbf{M}_2, \mathbf{M}_3$, and \mathbf{M}_4 ; for robot \mathbf{R}_2 is $\mathbf{I}_2, \mathbf{O}_3, \mathbf{M}_1$, and \mathbf{M}_3 ; for robot \mathbf{R}_3 is $\mathbf{I}_3, \mathbf{O}_2, \mathbf{M}_2$, 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}_1 . After being processed by \mathbf{M}_2 , it is moved to \mathbf{O}_1 by \mathbf{R}_1 . A raw product \mathcal{J}_2 is taken from \mathbf{I}_2 by \mathbf{R}_2 . 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 M_3 and then moved to M_5 by R_1 . After its process in either M_2 or M_5 , it is moved to O_2 by R_3 so that a final product of \mathcal{J}_2 is obtained. A raw product of \mathcal{J}_3 is taken from I_3 by R_3 . After being processed by M_4 , it is moved to M_3 by R_1 . After that, it is moved to O_3 by R_2 . Noticeably, two copies of resources are required when \mathcal{J}_2 is processed by M_1 or \mathcal{J}_3 is processed by M_4 . In Fig. 4, resource allocation is indicated by a directed arc as labeled by t_j , $j \in \mathbb{N}_{20}$. A number across the arc indicates the resource quantity while the default value is one.

Fig. 5 shows the net model of this AMS, which allows multiple resource acquisitions and flexible routes. The system is an S^4R where $P_0 = \{p_1, p_5, p_{14}\}$, $P_{A_1} = \{p_2 - p_4\}$, $P_{A_2} = \{p_6 - p_{13}\}$, $P_{A_3} = \{p_{15} - p_{19}\}$, $P_R = \{p_{20} - p_{26}\}$, $t_0^1 = t_1$, $t_0^2 = t_5$, and $t_0^3 = t_{15}$. Places $p_{20} - p_{26}$ denote M_1 , M_2 , R_1 , R_2 , M_3 , R_4 , and R_3 , respectively. Initially, it is assumed that there are no parts in process. $M(p_1) = M(p_5) = M(p_{19}) = 8$ represents that the maximum job instances that are allowed for part types $\mathcal{J}_1 - \mathcal{J}_3$ at a time, respectively.

TABLE VI

 GENERATED MONITORS FOR THE NET IN FIG. 5 DUE TO $\min\text{-}\mathcal{L}_{\mathcal{I}}$

i	$\bullet p_{c_i}$	$p_{c_i}^\bullet$	$M(p_{c_i})$
1	$\{t_6, t_9, t_{18}\}$	$\{t_5, t_{15}\}$	4
2	$\{t_6, t_{11}, 2t_{17}\}$	$\{t_5, 2t_{15}\}$	2
3	$\{t_3, t_7, t_{10}\}$	$\{t_1, t_5\}$	5
4	$\{t_6, t_{13}, 2t_{17}\}$	$\{t_5, 2t_{15}\}$	4
5	$\{t_7, 2t_8, t_9, t_{19}\}$	$\{2t_5, t_{15}\}$	2
6	$\{t_3, t_7, t_{12}, t_{17}\}$	$\{t_1, t_5, t_{15}\}$	3

This net is deadlock-prone since some siphons can be eventually undermarked during system evolution. Our analysis shows that there are 21 siphons, i.e., $S_1 = \{p_{13}, p_{16}, p_{25}, p_{26}\}$, $S_2 = \{p_4, p_{13}, p_{19} - p_{26}\}$, $S_3 = \{p_4, p_{13}, p_{18}, p_{21}, p_{22}, p_{24} - p_{26}\}$, $S_4 = \{p_4, p_{13}, p_{17}, p_{21}, p_{22}, p_{25}, p_{26}\}$, $S_5 = \{p_4, p_{11}, p_{12}, p_{19} - p_{25}\}$, $S_6 = \{p_4, p_{11}, p_{12}, p_{18}, p_{21}, p_{22}, p_{24}, p_{25}\}$, $S_7 = \{p_4, p_{11}, p_{12}, p_{17}, p_{21}, p_{22}, p_{25}\}$, $S_8 = \{p_4, p_{10}, p_{13}, p_{19} - p_{24}, p_{26}\}$, $S_9 = \{p_4, p_{10}, p_{13}, p_{18}, p_{21}, p_{22}, p_{24}, p_{26}\}$, $S_{10} = \{p_4, p_{10}, p_{11}, p_{19} - p_{24}\}$, $S_{11} = \{p_4, p_{10}, p_{11}, p_{18}, p_{21}, p_{22}, p_{24}\}$, $S_{12} = \{p_4, p_{10}, p_{13}, p_{17}, p_{21}, p_{22}, p_{26}\}$, $S_{13} = \{p_4, p_{10}, p_{11}, p_{17}, p_{21}, p_{22}\}$, $S_{14} = \{p_2, p_4, p_9, p_{13}, p_{19}, p_{20}, p_{22} - p_{26}\}$, $S_{15} = \{p_2, p_4, p_9, p_{12}, p_{19}, p_{20}, p_{22} - p_{25}\}$, $S_{16} = \{p_2, p_4, p_9, p_{13}, p_{18}, p_{22}, p_{24} - p_{26}\}$, $S_{17} = \{p_2, p_4, p_9, p_{12}, p_{18}, p_{22}, p_{24}, p_{25}\}$, $S_{18} = \{p_2, p_4, p_9, p_{13}, p_{17}, p_{22}, p_{25}, p_{26}\}$, $S_{19} = \{p_2, p_4, p_9, p_{12}, p_{17}, p_{22}, p_{25}\}$, $S_{20} = \{p_2, p_4, p_9, p_{10}, p_{19}, p_{20}, p_{22} - p_{24}\}$, and $S_{21} = \{p_2, p_4, p_9, p_{10}, p_{18}, p_{22}, p_{24}\}$.

According to our approach, these siphons lead to 21 inequalities as shown in Table V. Using Algorithms 1 and 2, the second inequality, $l_2^T \cdot M \leq b_2$, is an independent inequality and also $\max\text{-}\mathcal{L}_{\mathcal{I}}$ after we decrease b_2 from 19 to 4. This is because, $\forall i \in \mathbb{N}_{21} \setminus \{2\}$, we have $l_i^T \leq l_2$ and $b_i \geq b_2$. Based on this inequality, we can produce the corresponding monitor as shown in Fig. 6. Our analysis shows that the controlled system is live with 6970 reachable states. For the sake of brevity, $p_{c_i}^\bullet$ is assumed to be t_0 as done in [12].

As $\max\text{-}\mathcal{L}_{\mathcal{I}}$ is quite conservative and the system behavior can be greatly restricted. We also can resort to $\min\text{-}\mathcal{L}_{\mathcal{I}}$, which shows a set of dependent inequalities, i.e., inequalities 1, 12, 13, and 19 – 21. One can easily verify that any other

inequality is dependent on these ones. This set of inequalities produce monitors as shown in Table VI. The corresponding controlled system is live with 13960 reachable states.

For the net in Fig. 5, the method in [2] leads to 21 monitors, namely, $p_{c_1} - p_{c_{21}}$ respectively corresponding to $S_1 - S_{21}$, as shown in Table VII. When these 21 monitors are added, a liveness-enforcing supervisor is obtained.

Among the above-derived 21 siphons $S_1 - S_{21}$, 8 of them, i.e., $S_1 - S_5$, S_8 , S_{10} , and S_{14} , are found to be elementary ones according to [26]. Thus, only 8 monitors are necessary to constitute a supervisor, as shown in Table VIII.

Further analysis shows that the permissible states produced by the approaches in [2] and [26] are 642743 and 986, respectively. The work in [2] is the first approach on the liveness supervisory control in terms of general Petri nets. Its solution requires that all siphons satisfy the *max-cs* property. Therefore, the size of a supervisor is proportional to the number of siphons, which proves to be in an exponential relationship with the size of the plant model. Compared with [2], our method can implement a supervisor with much fewer monitors. The work in [26] distinguishes siphons as elementary and dependent ones. Only the elementary ones are required to be controlled. However, this method does not really provide the lower bound of the size of the liveness-supervising supervisor, particularly in cases where some weakly dependent siphons exist. This defect can be exemplified by the net shown in Fig. 2(a). According to the approach in [26], S_1 can be a weakly dependent siphon with regard to the corresponding elementary siphons, i.e., S_2 and S_3 , because $\eta_1 = \eta_3 - \eta_2$. The theory in [26] requires two monitors to be imposed on S_2 and S_3 , respectively. Comparatively, our approach shows that only one monitor is necessary because both $M(p_4) + M(p_9) \leq 2$ and $M(p_5) + M(p_8) \leq 2$ are dependent on $M(p_4) + M(p_5) + M(p_8) + M(p_9) \leq 4$ after we decrease the right-hand scalar 4 to 2. Therefore, compared with [26], our method can produce more concise supervisor structure and syntactically explain the principle behind such a reduction. Since their applicability is validated through P -invariants, elementary siphons must involve some characteristic vectors so as to establish supervisor simplification techniques. Their computation is unnecessarily complicated compared with our work which distinguishes the independent and dependent inequalities in a quite straightforward way. Thus, our method outperforms existing strategies based on elementary siphons.

TABLE VIII

GENERATED MONITORS FOR THE NET IN FIG. 5 DUE TO [26]

i	$\bullet p_{c_i}$	$p_{c_i}^\bullet$	$M(p_{c_i})$
1	$\{t_6, t_{13}, 2t_{17}\}$	$\{t_5, 2t_{15}\}$	4
2	$\{t_3, t_7, t_8, t_{12}, t_{13}, 2t_{17}, t_{19}\}$	$\{t_1, 2t_5, 3t_{15}\}$	10
3	$\{t_3, t_{12}, t_{13}, 2t_{17}, t_{18}\}$	$\{t_1, t_5, 3t_{15}\}$	4
4	$\{t_3, t_{12}, t_{13}, 3t_{17}\}$	$\{t_1, t_5, 3t_{15}\}$	3
5	$\{t_3, t_7, t_8, t_{10}, t_{11}, t_{17}, t_{19}\}$	$\{t_1, 2t_5, 2t_{15}\}$	2
6	$\{t_3, t_7, t_8, t_9, t_{12}, t_{19}\}$	$\{t_1, 2t_5, t_{15}\}$	3
7	$\{t_3, t_7, t_8, t_9, t_{10}, t_{19}\}$	$\{t_1, 2t_5, t_{15}\}$	2
8	$\{t_7, 2t_8, t_{13}, 2t_{17}, t_{19}\}$	$\{2t_5, 3t_{15}\}$	3

VI. CONCLUSION

This work focuses on the synthesis of liveness enforcing supervisors of automated manufacturing systems allowing both

TABLE V
GENERATED INEQUALITIES CORRESPONDING TO SIPHONS

i	Inequalities ($l_i \cdot M \leq b_i$)	Independent
1	$M(p_{12})+M(p_{15})+2M(p_{16}) \leq 4$	No
2	$M(p_2)+M(p_3)+M(p_6)+2M(p_7)+M(p_8)+M(p_9)+M(p_{10})+M(p_{11})+M(p_{12})+M(p_{15})+3M(p_{16})+M(p_{17})+M(p_{18}) \leq 19$	Yes
3	$M(p_2)+M(p_3)+M(p_8)+M(p_9)+M(p_{10})+M(p_{11})+M(p_{12})+M(p_{15})+3M(p_{16})+M(p_{17}) \leq 14$	No
4	$M(p_2)+M(p_3)+M(p_9)+M(p_{10})+M(p_{11})+M(p_{12})+M(p_{15})+3M(p_{16}) \leq 10$	No
5	$M(p_2)+M(p_3)+M(p_6)+2M(p_7)+M(p_8)+M(p_9)+M(p_{10})+3M(p_{16})+M(p_{17})+M(p_{18}) \leq 17$	No
6	$M(p_2)+M(p_3)+M(p_8)+M(p_9)+M(p_{10})+2M(p_{16})+M(p_{17}) \leq 12$	No
7	$M(p_2)+M(p_3)+M(p_9)+M(p_{10})+2M(p_{16}) \leq 8$	No
8	$M(p_2)+M(p_3)+M(p_6)+2M(p_7)+M(p_8)+M(p_9)+M(p_{11})+M(p_{15})+M(p_{16})+M(p_{17})+M(p_{18}) \leq 16$	No
9	$M(p_2)+M(p_3)+M(p_8)+M(p_9)+M(p_{11})+M(p_{15})+M(p_{16})+M(p_{17}) \leq 7$	No
10	$M(p_2)+M(p_3)+M(p_6)+2M(p_7)+M(p_8)+M(p_9)+M(p_{17})+M(p_{18}) \leq 14$	No
11	$M(p_2)+M(p_3)+M(p_8)+M(p_9)+M(p_{17}) \leq 9$	No
12	$M(p_2)+M(p_3)+M(p_9)+M(p_{11})+M(p_{15})+M(p_{16}) \leq 7$	No
13	$M(p_2)+M(p_3)+M(p_9) \leq 5$	No
14	$M(p_6)+2M(p_7)+M(p_8)+M(p_{10})+M(p_{12})+2M(p_{16})+M(p_{17})+M(p_{18}) \leq 15$	No
15	$M(p_6)+2M(p_7)+M(p_8)+M(p_{10})+2M(p_{16})+M(p_{17})+M(p_{18}) \leq 13$	No
16	$M(p_8)+M(p_{10})+M(p_{12})+M(p_{15})+3M(p_{16})+M(p_{17}) \leq 10$	No
17	$M(p_8)+M(p_{10})+2M(p_{16})+M(p_{17}) \leq 8$	No
18	$M(p_{10})+M(p_{12})+M(p_{15})+3M(p_{16}) \leq 6$	No
19	$M(p_{10})+2M(p_{16}) \leq 4$	No
20	$M(p_6)+2M(p_7)+M(p_8)+M(p_{17})+M(p_{18}) \leq 10$	No
21	$M(p_8)+M(p_{17}) \leq 5$	No

TABLE VII
GENERATED MONITORS FOR THE NET IN FIG. 5 DUE TO [3]

i	$\bullet p_{c_i}$	$\overset{\circ}{p}_{c_i}$	$M(p_i)$	i	$\bullet p_{c_i}$	$\overset{\circ}{p}_{c_i}$	$M(p_i)$
1	$\{t_6, t_{13}, 2t_{17}\}$	$\{t_5, 2t_{15}\}$	4	12	$\{t_3, t_7, t_{12}, t_{17}\}$	$\{t_1, t_5, t_{15}\}$	7
2	$\{t_3, t_7, t_8, t_{12}, t_{13}, 2t_{17}, t_{19}\}$	$\{t_1, 2t_5, 3t_{15}\}$	19	13	$\{t_3, t_7, t_{10}\}$	$\{t_1, t_5\}$	5
3	$\{t_3, t_{12}, t_{13}, 2t_{17}, t_{18}\}$	$\{t_1, t_5, 3t_{15}\}$	14	14	$\{t_7, 2t_8, t_{13}, 2t_{17}, t_{19}\}$	$\{2t_5, 3t_{15}\}$	15
4	$\{t_3, t_{12}, t_{13}, 3t_{17}\}$	$\{t_1, t_5, 3t_{15}\}$	10	15	$\{t_7, 2t_8, t_{11}, t_{17}, t_{19}\}$	$\{2t_5, 2t_{15}\}$	13
5	$\{t_3, t_7, t_8, t_{10}, t_{11}, t_{17}, t_{19}\}$	$\{t_1, 2t_5, 2t_{15}\}$	17	16	$\{t_6, t_{13}, 2t_{17}, t_{18}\}$	$\{t_5, 3t_{15}\}$	10
6	$\{t_3, t_{10}, t_{11}, t_{17}, t_{18}\}$	$\{t_1, t_5, 2t_{15}\}$	12	17	$\{t_6, t_{11}, t_{17}, t_{18}\}$	$\{t_5, 2t_{15}\}$	8
7	$\{t_3, t_{10}, t_{11}, 2t_{17}\}$	$\{t_1, t_5, 2t_{15}\}$	8	18	$\{t_6, t_{13}, 3t_{17}\}$	$\{t_5, 3t_{15}\}$	6
8	$\{t_3, t_7, t_8, t_9, t_{12}, t_{19}\}$	$\{t_1, 2t_5, t_{15}\}$	16	19	$\{t_6, t_{11}, 2t_{17}\}$	$\{t_5, 2t_{15}\}$	4
9	$\{t_3, t_9, t_{12}, t_{18}\}$	$\{t_1, t_5, t_{15}\}$	7	20	$\{t_7, 2t_8, t_9, t_{19}\}$	$\{2t_5, t_{15}\}$	10
10	$\{t_3, t_7, t_8, t_9, t_{10}, t_{19}\}$	$\{t_1, 2t_5, t_{15}\}$	14	21	$\{t_6, t_9, t_{18}\}$	$\{t_5, t_{15}\}$	5
11	$\{t_3, t_9, t_{10}, t_{18}\}$	$\{t_1, t_5, t_{15}\}$	9	—	—	—	—

flexible routes and multiple resource acquisition operations. Deadlocks are related to the emergence of a class of Petri net objects, namely undermarked siphons. To avoid their undermarkedness, the number of tokens in these siphons' complementary places must be strictly limited, thus leading to a set of general mutual exclusive constraints. They can be distinguished as the independent and dependent ones. The dependent ones can be implicitly controlled after these independent ones are properly controlled. This technique can reduce the number of needed monitors. Numerical results show that our resulting supervisor is simple in its structure and can ensure more permissive behavior. The generalization of the research results in this paper will be conducted for more complex systems.

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, Sept. 1998.
- [2] K. Barkaoui and I. B. Abdallah, "A deadlock prevention method for a class of FMS," in *Proc. IEEE Int. Conf. Syst., Man, and Cybern.*, pp. 4119–4124, Vancouver, British Columbia, Canada, Oct. 1995.
- [3] K. Barkaoui and J. F. Peyre, "On liveness and controlled siphons in Petri nets," in *Lect. Not. in Comput. Sci.*, vol. 1091, pp. 57–72, Berlin: Springer-Verlag, Jun. 1996.
- [4] F. Basile, P. Chiacchio, A. Giua, and C. Seatzu, "Deadlock recovery of Petri net models controlled using observers," in *Proc. IEEE Sym. Emerg. Tech. Fact. Auto.*, Antibes-Juan les pins, France, Oct. 2001, pp. 441–449.
- [5] F. Basile, P. Chiacchio, and A. Giua, "An optimization approach to Petri net monitor design," *IEEE Trans. Automat. Contr.*, vol. 52, no. 2, pp. 306–311, Feb. 2007.
- [6] F. Basile, P. Chiacchio, and J. Coppola, "A hybrid model of complex automated warehouse systems part I: Modeling and simulation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 4, pp. 640–653, Oct. 2012.
- [7] F. Basile, P. Chiacchio, and J. Coppola, "A hybrid model of complex automated warehouse systems part II: Analysis and experimental results," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 4, pp. 654–668, Oct. 2012.
- [8] 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.
- [9] C. B. Chu, F. Chu, M. C. Zhou, H. X. Chen, and Q. N. Shen, "A polynomial dynamic programming algorithm for crude oil transportation planning," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 42–55, Jan. 2012.
- [10] D. Y. Chao, "Improvement of suboptimal siphon- and FBM-based control model of a well-known S^3PR ," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 404–411, Apr. 2011.
- [11] Y. H. Du, X. T. Li, and P. C. Xiong, "A Petri net approach to mediation-aided composition of web services," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 429–435, Apr. 2012.
- [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] M. P. Fanti, A. M. Mangini, and W. Ukovich, "Fault detection by labeled Petri nets in centralized and distributed approaches," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 392–404, Apr. 2013.
- [15] A. Giua, F. DiCesare, M. Silva, "Generalized mutual exclusion constraints on nets with uncontrollable systems," in *Proc. IEEE Int. Conf. Syst., Man, and Cyber.*, Chicago, IL, Oct. 1992, pp. 974–979.
- [16] 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.
- [17] Y. S. Huang, M. D. Jeng, X. L. Xie X, 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.
- [18] B. Hruz and M. C. Zhou, *Modeling and Control of Discrete Event Dynamic Systems*, Springer, NYC, NY, 2007.
- [19] 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.
- [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, Z. W. Li, and Y. Tang, "Deadlock-free control of automated manufacturing systems with flexible routes and assembly operations using Petri nets," *IEEE Trans. Ind. Inform.*, vol. 9, no. 1, pp. 109–121, Feb. 2013.
- [23] 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.
- [24] 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.
- [25] 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.
- [26] 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.
- [27] L. X. Li and C. N. Hadjicostis, "Least-cost transition firing sequence estimation in labeled Petri nets with unobservable transitions," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 394–403, Apr. 2011.
- [28] 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.
- [29] S. Miremadi, K. Åkesson, and B. Lennartson, "Symbolic computation of reduced guards in supervisory control," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 4, pp. 754–765, Oct. 2011.
- [30] Y. Mati and X. L. Xie, "Multiresource shop scheduling with resource flexibility and blocking," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 175–189, Jan. 2011.
- [31] 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.
- [32] F. Qiao, Y. M. Ma, L. Li, and H. X. Yu, "A Petri net and extended genetic algorithm combined scheduling method for wafer fabrication," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 1, pp. 197–204, Jan. 2013.
- [33] F. Tricas, F. García-Vallés, 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.
- [34] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Petri net-based scheduling of single-arm cluster tools with reentrant atomic layer deposition processes," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 42–55, Jan. 2011.
- [35] N. Q. Wu, F. Chu, C. B. Chu, and M. C. Zhou, "Schedulability analysis of short-term scheduling for crude oil operations in refinery with oil residency time and charging-tank-switch-overlap constraints," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 1, pp. 190–204, Jan. 2011.
- [36] K. Y. Xing, M. C. Zhou, F. Wang, H. X. Liu, and F. Tian, "Resource-transition circuits and siphons for deadlock control of automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 41, no. 1, pp. 74–84, Jan. 2011.
- [37] K. Y. Xing, M. C. Zhou, F. Wang, H. X. Liu, and F. Tian, "Resource-transition circuits and siphons for deadlock control of automated manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A. Syst., Humans*, vol. 41, no. 1, pp. 74–84, Jan. 2011.
- [38] 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.



Hesuan Hu (M'11-SM'12) received the B.S. degree in computer engineering and the M.S. and Ph.D. degrees in electromechanical engineering from Xidian University, Xi'an, China, in 2003, 2005, and 2010, respectively.

From 1997 to 2000, he was an Assistant Engineer with the Metrology Department, Xi'an Aircraft Engine Control Engineering Company, Ltd., where he served as the Director of the Electrical and Electronic Laboratory from 2001 to 2002 and the Leader for several innovation projects. From 2008

to 2011, he worked as a Research Scholar in the Department of Electrical and Computer Engineering at New Jersey Institute of Technology, Newark, NJ, USA. At the same institute, he also served as an Adjunct Instructor in charge of several courses in the field of electrical engineering at the graduate level. He is currently a Ph. D advisor at Xidian University. His research interests include Petri nets, discrete event systems, automated manufacturing systems, multimedia streaming systems, and artificial intelligence. In the aforementioned areas, he has over 60 publications in journals, book chapters, and conference proceedings.

Dr. Hu was a Program Committee member for the numerous conferences including the 2008 International Conference on Industrial Engineering and Other Applications of Applied Intelligent Systems, the 2008 IEEE International Conference on Automation Science and Engineering, the 2008–2013 IEEE International Conference on Networking, Sensing and Control, the 2010 IEEE International Conference on Systems, Man, and Cybernetics (SMC), and the 2010 IEEE International Conference on Mechatronics and Automation. He serves as a frequent Reviewer for more than 30 international journals, including the IEEE TRANSACTIONS ON SYSTEMS, MAN, AND CYBERNETICS-PART A: SYSTEMS AND HUMANS; the IEEE TRANSACTIONS ON INDUSTRIAL INFORMATICS; the IEEE TRANSACTIONS ON AUTOMATION SCIENCE AND ENGINEERING; the IEEE COMMUNICATIONS LETTERS; the IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY; the IEEE TRANSACTIONS ON COMPUTERS; the ACM TRANSACTIONS ON EMBEDDED COMPUTING SYSTEMS; and DISCRETE EVENT DYNAMIC SYSTEMS. He is on the Editorial Board of 10+ international journals. He is listed in Who's Who in North American Education, Who's Who in America, Who's Who in the World. He was the recipient of many national and international awards including the Franklin V. Taylor Outstanding Paper Award from the IEEE SMC Society in 2010 and the finalist of the Best Automation Paper from the IEEE ICRA Society in 2013.



Yang Liu graduated in 2005 with a Bachelor of Computing in the National University of Singapore (NUS). In 2010, he obtained his PhD and continued with his post doctoral work in NUS. Since 2012, he joined Nanyang Technological University as a Nanyang Assistant Professor. His research focuses on software engineering, formal methods and security. Particularly, he specializes in software verification using model checking techniques. This work led to the development of a state-of-the-art model checker, Process Analysis Toolkit.