IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received January 13, 2019, accepted February 4, 2019, date of publication February 13, 2019, date of current version March 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2899190

Time-Varying Automated Manufacturing Systems and Their Invariant-Based Control: A Petri Net Approach

CHEN CHEN[1], (Student Member, IEEE), AND HESUAN HU[1,2,3], (Senior Member, IEEE)
[1]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[2]School of Computer Science and Engineering, College of Engineering, Nanyang Technological University, Singapore 639798
[3]State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710054, China

Corresponding author: Hesuan Hu (huhesuan@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61203037, Grant 51305321, and Grant 61573265, in part by the New Century Excellent Talents in University under Grant NCET-12-0921, in part by the Fundamental Research Funds for the Central Universities under Grant K5051304021, Grant K7215581201, and Grant K5051304004, in part by the Major Fundamental Research Program of the Natural Science Foundation of Shaanxi Province under Grant 2017ZDJC-34, in part by the Academic Research Fund Tier 1 by the Ministry of Education in Singapore under Grant 2014-T1-001-147, and in part by the Academic Research Fund Tier 2 by the Ministry of Education in Singapore under Grant MOE2015-T2-2-049.

ABSTRACT In the context of automated manufacturing systems, we propose a class of time-varying systems which enable different working modes at different time intervals. This paper provides a comprehensive and comparative study on invariant-based linear constraints in supervisory control of time-varying systems. First, generalized linear constraints (GLCs) which contain both marking and firing vectors are used to prevent the illegal event from firing at the critical good marking, so as not to reach any deadlock. Supervisor simplification on GLCs is presented in order to remove the dependent ones while retaining the independent ones. Then, weight coefficients are introduced to linear constraints to enhance our specifications' expressivity capability in these iterative approaches. It shows that such specifications have an obvious advantage in improving permissiveness for the systems whose processes can accommodate more tokens. Above linear constraints can be implemented via place invariants in Petri net, and solve the forbidden state problem and event/state separation problem. The experimental studies illustrate the application of time-varying systems and the effectiveness of invariant-based control methods.

INDEX TERMS Automated manufacturing systems, Petri nets, invariant-based control, supervisor simplification.

I. INTRODUCTION

Automated manufacturing systems (AMSs) are usually used to automatically and simultaneously accomplish different kinds of jobs by using a limited number of shared resources such as machines, robots, automated guided vehicles, and buffers. They always appear as a highly-complex interaction among a set of processes and a set of resources. Petri nets (PNs) are widely used to model, analyze, and control AMSs, since they provide not only the abundant structural information but also the rigorous mathematical expressivity. There are plenty of literatures in this field to solve various problems, including deadlock problem [10], [12]–[14], robust deadlock problem [29], [30], fault diagnosis problem [6], [51], process scheduling problem [1], [46]–[50], [53], [59], [60], dynamic reconfiguration problem [57], [58], and control implementation problem [34], [35]. Among these studies, liveness which ensures all events can finally fire from any one in the reachable markings is the most fundamental requirement. On the contrary, a deadlock is a highly undesirable situation in which each of a set of two or more jobs (or events) keeps waiting indefinitely for the other jobs (or events) in the set to release resources [8], [43]. The occurrence of a deadlock can cripple the entire system and render automated operations impossible. To prevent deadlocks, numerous control strategies are designed, resulting in supervisory control techniques (SCTs) [2]–[4], [15]–[17], [23]–[26], [28], [33], [37], [39], [40], [45], [55], [57].

In the scenario of PNs, place invariants (*P*-invariants) are considered as an important tool to analyze structural properties and help enforce specifications [18], [20]–[22],

The associate editor coordinating the review of this manuscript and approving it for publication was Shouguang Wang.

[27], [31], [32], [52]. They only depend on the topological structure while not on the initial markings. In the support of a P-invariant, the weighted sum of tokens remains to be constant. In supervisory control, specifications in form of linear inequalities can be enforced by constructing a P-invariant structure superimposed on the original net. Generalized mutual exclusion constraints (GMECs) are a typical kind of such specifications which limit the weighted sum of tokens [18]. The work in [52] addresses the logical and algebraic constraints containing elements of the marking and firing vectors. In [22], in order to be more expressive, the class of constraints are generalized to linear constraints containing marking terms, firing vector terms, and Parikh vector terms. It shows that the extended constraints can describe any supervisor consisting of control places arbitrarily connected to the transitions of a plant PN. Thus, above work shows the generality of linear constraints which can be enforced using P-invariants.

The specifications of linear constraints are widely used in liveness-enforcing supervisory control. From the perspective of structure, as a kind of special structural objects, siphons are closely related to deadlocks. In the existing typical classes of PNs, such as Systems of Simple Sequential Processes with Resources (S^3PR) [15], System of Sequential Systems with Shared Processes Resources (S^4PR) [41], Hierarchical Augmented Marked Graphs (HAMGs) [8], the PN is not live if and only if there is at least a siphon that is insufficiently marked. The condition and approach to enumerate minimal siphons in S^4PR are studied in [44] and [56]. Given a set of siphons, GMECs are constructed to ensure the sufficient markedness of siphons, leading to a volume of control strategies [8], [15], [20], [36], [41]. Admittedly, the number of siphons is exponential with the size of PNs. Furthermore, the control places that enforce the sufficient markedness of siphons would bring in new siphons [15]. From reachability perspective, supervisory control is desired to retain the good markings while remove the bad ones. Some typical approaches are also based on linear constraints to generate supervisors. In [42], an iterative method is proposed. At each iteration, the first-met bad marking (FBM) is derived from the reachability graph. It represents the very first entry from the live zone (LZ) to the dead zone (DZ). Then, a control place is designed to prevent the FBM from being reached by using a P-invariant. This method is more suitable for ordinary PNs, and the GMECs constructed to prevent the system from reaching FBMs are simple since all the coefficients are unitary. The work in [11] uses the vector covering approach to reduce the sets of legal markings and FBMs respectively, resulting in the maximally permissive control policy that can be achieved by designing control places such that all the markings in the minimal covered set of FBMs are forbidden and no marking in the minimal covering set of legal markings is forbidden. Furthermore, the work in [13] takes advantage of weighted and data inhibitor arcs to design the optimal supervisors that maximally preserve the legal markings. In these work, integer programming is utilized to design the control places that separate the good and bad markings. The work in [5] addresses the design of compact and maximally permissive decentralized supervisors for PNs, based on GMECs. Maximal permissiveness is ensured by efficiently exploring the solution space using a branch and bound method that operates on the reachable states.

Besides the liveness-enforcing supervisory control of AMSs, the process scheduling is also an important problem when considering time information. For example, the deterministic timed weighted marked graphs are used to solve cycle time optimization problem that finds an initial marking to minimize the average cycle time [19]. Cluster tools, as a kind of automated robotic manufacturing systems containing multiple computer-controlled process modules, are also widely used to study the cyclic scheduling problem [1], [48], [53]. For the time-related constraints, such as wafer residency time constraint, wafer sojourn time, and activity time variation, are studied in single- or dual-arms cluster tool to schedule processes [38], [46], [47], [54], [60]. Although there are plenty of works on process scheduling problem, there is litter work on time-varying specifications which imply events are disabled to fire at some certain time intervals. In practice, time-varying specifications can increase the flexibility ability while decrease the configuration cost. Then, one may concern the liveness-enforcing supervisory control of such systems with time-varying specifications. In this work, the time-varying specifications and linear constraints are studied respectively. It discovers that they can be enforced independently. Thus, this would greatly improve the modeling ability via PNs and decrease the complexity of the liveness enforcement for such kind of systems.

Specifically, a special class of PNs, namely Time-varying System of Sequential Systems with Shared Resources (Tv-S^4R) is proposed in [8]. Generalized linear constraints (GLCs) containing marking and firing vectors are used to solve the deadlock prevention problem. This paper supplements the work in two aspects. On one hand, we address the time-varying specifications in a different way, such that an event is fireable only when it is process-, resource-, and time-enabled at a marking. In this circumstance, job types can be prohibited or allowed at different time intervals. On the other hand, forbidden state problem and event/state separation problem are studied via linear constraints which can be implemented by P-invariants in PNs. Especially, weight coefficients are introduced in linear constraints, so as to enhance specifications' expressivity and improve system's permissiveness. The advantage of the weighted linear constraints is embodied in the system whose processes can accommodate more tokens. Therefore, this paper provides a comprehensive and comparative study on invariant-based supervisory control methods as well as their supervisor simplifications. This derives the unified algorithm which combines both supervisor synthesis and supervisor simplification. It is straightforward and easy to use in application, although it has the exponential complexity and does not achieve the maximal permissiveness in general PNs. It concludes that event-based ones present a new perspective to deal with event/state separation problem while the weighted state-based method is superior to others in terms of permissiveness.

The remainder of this paper is structured as follows. Section II reviews PN's basic definitions and notations. Section III is devoted to a special class of PNs as well as its time-varying specifications. Section IV deals with the event-based control method and its supervisor simplification in liveness-enforcing supervisory control. The weighted invariant-based control methods are presented in Section V. It also provides the unified algorithm to generate the simplified liveness-enforcing supervisors. Section VI concludes this paper.

II. PRELIMINARIES

In this section, we present some notations which are used in the following discussions.

A. PETRI NETS (PNS) DEFINITIONS

A PN 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 $W : (P \times T) \cup (T \times P) \to \mathbb{N} = \{0, 1, 2, \ldots\}$ such that $P \cup T \neq \emptyset$, $P \cap T = \emptyset$, and $W(x, y) = 0$ if $(x, y) \notin F$. $W(x, y)$ is undefined if $x, y \in P$ or $x, y \in T$. Specially, N is said to be ordinary when $W : F \to \{1\}$. Otherwise, it is general. A marking of N is a mapping $M : P \to \mathbb{N}$. It can be represented by tokens located at various places. $M(p)$ (resp., $M_0(p)$) indicates the number of tokens in p at M (resp., M_0). M_0 denotes the initial marking. (N, M_0) is a net system with an initial marking M_0. Graphically, places, transitions, and tokens are represented by circles, bars, and dots, respectively. As far as a W is nonzero, it is depicted by an arc bridging a pair of place and transition. Its value is labeled by a number namely weight which assigns to each arc a nonnegative integer arc multiplicity; nevertheless, no arc may connect two places or two transitions. By default, absence of a label for an arc implies that its weight is unity. A PN system's size is defined by $|N| = |P| + |T| + \sum_{p\in P} M_0(p)$.

B. STRUCTURE PROPERTIES

A PN 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 $W : F \to \{1\}$ and $\forall t \in T$, $|{}^\bullet t| = |t^\bullet| = 1$. It is a marked graph if $W : F \to \{1\}$ and $\forall p \in P$, $|{}^\bullet p| = |p^\bullet| = 1$. N's input incidence matrix $[N^-]_{|P|\times|T|} = [W(p_i, t_j)]$ and output one $[N^+]_{|P|\times|T|} = [W(t_j, p_i)]$. Its incidence matrix $[N]_{|P|\times|T|} = [N^+]_{|P|\times|T|} - [N^-]_{|P|\times|T|}$. $[N_{p_i}]_{1\times|T|}$ (resp., $[N^-_{p_i}]_{1\times|T|}$, $[N^+_{p_i}]_{1\times|T|}$) is the i-th row of $[N]_{|P|\times|T|}$ (resp., $[N^-]_{|P|\times|T|}$, $[N^+]_{|P|\times|T|}$). A path is an ordered string $\langle x_1, x_2, \ldots, x_n\rangle$ such that: 1) $\{x_1, x_2, \ldots, x_n\} \subseteq P \cup T$; 2) $\forall i \in \mathbb{N}_{n-1} = \{1, 2, \ldots, n-1\}$, $x_{i+1} \in x_i^\bullet$. A simple path is an ordered string whose all nodes are different. A circuit is a path in which the first and last nodes are identical, i.e., $x_1 = x_n$, while others are different. A PN is strongly-connected if there exists a directed path from every node to every other node in $P \cup T$.

C. DYNAMIC PROPERTIES

t is enabled at M, denoted by $M\,[t\rangle$, if $\forall p \in {}^\bullet t$, $M(p) \geq W(p, t)$. Given a marking M, t can fire if it is enabled at M. M' is reachable from M, denoted by $M\,[\sigma\rangle\,M'$, if there exists a firing sequence $\sigma = \langle t_1, t_2, \ldots, t_n\rangle$ such that $M\,[t_1\rangle\,M_1 \ldots [t_n\rangle\,M'$. $\overrightarrow{\sigma}$ is a $|T|$-dimensional firing count vector where $\overrightarrow{\sigma}(t)$ states the number of t's appearances in σ. Precisely, this evolution can be described by $M' = M + [N] \cdot \overrightarrow{\sigma}$. The set of all markings reachable from M_0 is denoted by $R(N, M_0)$. It follows a necessary reachability condition, i.e., $M = M_0 + [N] \cdot \overrightarrow{\sigma}$. When $|\sigma| = 1$, we have $M\,[t\rangle\,M'$, implying t's firing at M can lead to M'. (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 $\forall M \in R(N, M_0)$, $\exists M' \in R(N, M)$, $M'\,[t\rangle$ holds. t is dead at $M \in R(N, M_0)$ if $\nexists M' \in R(N, M)$ so that $M'\,[t\rangle$ holds. (N, M_0) is deadlock-free if $\forall M \in R(N, M_0)$, $\exists t \in T$, $M\,[t\rangle$. It is livelock if it is deadlock-free and $\exists t \in T$ so that t is dead at $M \in R(N, M_0)$. (N, M_0) is live if $\forall t \in T$, t is live under M_0.

A Timed Place Petri net (TPPN) is a tuple $N = (P, T, F, W, M_0, D)$, where the definitions on P, T, F, W, and M_0 are the same as the ones in general PNs. $D = [d_1, d_2, \ldots, d_n]^T$ is a delay function mapping from P to $\mathbb{R}^+$, where d_i is the delay of a place p_i, and $\mathbb{R}^+$ is the set of positive real numbers. A token in a place p_i is not available until d_i units of time have elapsed after its arrival.

D. FUNDAMENTAL OBJECTS

A nonempty set $S \subseteq P$ (resp., $Q \subseteq P$) is a siphon (resp., trap) if ${}^\bullet S \subseteq S^\bullet$ (resp., $Q^\bullet \subseteq {}^\bullet Q$). A strict minimal siphon is a siphon containing neither other siphon nor trap except itself. p is marked by M if $M(p) > 0$ where $M(p)$ means the number of tokens in p at M. The sum of token numbers 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 $\forall t \in S^\bullet$ such that t is dead at M.

A P- (resp., T-) vector is a column vector $I : P$ (resp., $J : T$) $\to \mathbb{Z}$ 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. By $I \geq \mathbf{0}$, we mean that $\forall p \in P$, $I(p) \geq 0$ and $\exists p \in P$, $I(p) > 0$. 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. $\|I\|^+ = \{p \in P \mid I(p) > 0\}$ (resp., $\|I\|^- = \{p \in P \mid I(p) < 0\}$) is called the positive (resp., negative) support of I. A P-semiflow I (resp., T-semiflow J) is said to be minimal if there exists no other P-semiflow I' (resp., T-semiflow J') such that $\|I\| \supset \|I'\|$ (resp., $\|J\| \supset \|J'\|$). 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., $[N] \cdot J = \mathbf{0}$).

III. PN MODELING OF AMS

In order to improve the flexility of AMS model, we consider a kind of models, i.e., Time-varying System of Sequential Systems with Shared Resources (Tv-S^4R), which possesses different operation modes at different time intervals. The AMS discussed in this work is composed of two parts. One is the virtual model which is a circuit representing the sequential time intervals, while another is the physical model which allows flexible routes and multiple-resource acquisitions. The former is usually realized by buffers, while the latter consists of a set of resources $\mathcal{R} = \{\mathcal{R}_i, i \in \mathbb{N}_L = \{1, 2, \ldots, L\}\}$ and a set of job processes $\mathcal{J} = \{\mathcal{J}_j, j \in \mathbb{N}_K = \{1, 2, \ldots, K\}\}$, where $L, K \in \mathbb{N}^+$. A natural number $C_i \in \mathbb{N}^+$ is associated to each resource as its capacity. Each process $\mathcal{J}_j$ is denoted as a series of parallel and/or sequential stages. In the j-th process, a stage, say k, is represented by a place p_{jk} whose resource requirement is expressed by an L-dimensional vector $\alpha_{p_{jk}}$ with $\alpha_{p_{jk}}[i]$, where $i \in \mathbb{N}_L$, indicating how many units of resource $\mathcal{R}_i$ are required to support the execution of the stage denoted by p_{jk}. In the sequel, $\alpha_{p_{jk}}$ is denoted as $\sum_{i=1}^{L}\mathcal{R}_i$. This model is characterized as System of Sequential Systems with Shared Resources (S^4R), which can represent a wide range of systems since it has no limitation on the utilization of resource type and quantity. Combined with the time information, Tv-S^4R can represent systems with time-varying specifications.

A. TV-S^4R MODEL

Definition 1: An S^4R is a strongly, general, and pure PN $N = (P, T, F, W)$ 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 (activity), and resource places, respectively; b) $P_0 = \cup_{i\in\mathbb{N}_K}\{p_{0_i}\}$, where $\mathbb{N}_K = \{1, 2, \ldots, K\}$; c) $P_A = \cup_{i\in\mathbb{N}_K} P_{A_i}$, where $\forall i \in \mathbb{N}_K$, $P_{A_i} \neq \emptyset$, and $\forall i, j \in \mathbb{N}_K$, $i \neq j$, $P_{A_i} \cap P_{A_j} = \emptyset$; and d) $P_R = \{r_1, r_2, \ldots, r_L\}$;
2) $T = \cup_{i\in\mathbb{N}_K} T_i$, where $\forall i \in \mathbb{N}_K$, $T_i \neq \emptyset$, and $\forall i, j \in \mathbb{N}_K$, $i \neq j$, $T_i \cap T_j = \emptyset$;
3) $\forall i \in \mathbb{N}_K$, subnet $\overline{N}_i = N \mid (\{p_{0_i}\} \cup P_{A_i}, T_i)$ is a strongly-connected state machine such that every circuit contains p_{0_i}; and
4) $\forall r \in P_R$, $\exists$ a unique minimal P-invariant $X_r \in \mathbb{N}^{|P|}$ such that $\{r\} = \|X_r\| \cap P_R$, $P_0 \cap \|X_r\| = \emptyset$, $P_A \cap \|X_r\| \neq \emptyset$, $X_r(r) = 1$, and $P_A = \cup_{r\in P_R}(\|X_r\| \setminus \{r\})$.

Systems modeled by S^4Rs are characterized by high efficiency and concurrency to manufacturing. Then, time variable is incorporated in S^4Rs, resulting in Tv-S^4R.

Definition 2: A Tv-S^4R is a TPPN $\widehat{N} = (\widehat{P}, \widehat{T}, \widehat{F}, \widehat{W}, \widehat{D})$ which composed of two independent subsets $N = (P, T, F, W)$ and $N_{Tv} = (P_{Tv}, T_{Tv}, F_{Tv}, W_{Tv}, D_{Tv})$ such that $P \cap P_{Tv} = \emptyset$, $T \cap T_{Tv} = \emptyset$, and $F \cap F_{Tv} = \emptyset$. Moreover, we have:

1) $N = (P, T, F, W)$ is an S^4R;
2) $N_{Tv} = (P_{Tv}, T_{Tv}, F_{Tv}, W_{Tv})$ is a one-way loop TPPN such that $\forall t \in T_{Tv}$, $|{}^\bullet t| = |t^\bullet| = 1$, $\forall p \in P_{Tv}$, $|{}^\bullet p| = |p^\bullet| = 1$, and $W_{Tv} : F_{Tv} \to \{0, 1\}$;
3) $D_{Tv} = [d_{Tv_1}, d_{Tv_2}, \ldots, d_{Tv_l}]^T$ such that $\forall p_{Tv_i} \in P_{Tv}$ is associated with a delay $d_{Tv_i} \in \mathbb{R}^+$, where $l = |P_{Tv}|$; and
4) $\widehat{P} = P \cup P_{Tv}$, $\widehat{T} = T \cup T_{Tv}$, $\widehat{F} = F \cup F_{Tv}$, $\widehat{W}(x, y) = W(x, y)$ if $x, y \in P \cup T$, $\widehat{W}(x, y) = W_{Tv}(x, y)$ if $x, y \in P_{Tv} \cup T_{Tv}$, and $\widehat{D} = [D^T, D_{Tv}^T]^T$ with $D = [\varepsilon_1\ \varepsilon_2 \ldots \varepsilon_{|P|}]^T$ where $\varepsilon \in \mathbb{R}^+ \cup \{0\}$.

Definition 3: Marking $\widehat{M}_0$ is an acceptable initial marking of $\widehat{N}$ if

1) $\widehat{M}_0(p_0) \geq 1$, $\forall p_0 \in P_0$;
2) $\widehat{M}_0(p) = 0$, $\forall p \in P_A$;
3) $\widehat{M}_0(r) \geq X_r(p)$, $\forall r \in P_R$, $\forall p \in P_A$; and
4) there exists a unique $p_{Tv_i} \in P_{Tv}$ such that $\widehat{M}_0(p_{Tv_i}) = 1$, and $\forall j \neq i, j \in \mathbb{N}_{|P_{Tv}|} = \{1, 2, \ldots, |P_{Tv}|\}$, $M_0(p_{Tv_j}) = 0$.

In Tv-S^4R, $\widehat{M} = [M; M_{Tv}]$ where $M \in \mathbb{N}^{|P|}$ and $M_{Tv} \in \mathbb{N}^{|P_{Tv}|}$. M corresponds to the manufacturing state while M_{Tv} indicates which time interval the corresponding state belongs to. Thus, events can be stipulated to fire or not fire at specific time intervals. This is the time-varying feature of Tv-S^4R. In this circumstance, it has the ability to schedule job types at different time intervals for the benefit of high flexibility and productivity. Note that they are different from timed Petri nets. The former puts emphasis on the different operation modes of an event at a time, especially the time-varying feature. While the latter focuses on time delay associated to each event so as to deal with performance evaluations, safety determination, or behavioral properties [7]. In the sequential, $(\widehat{N}, \widehat{M}_0)$ is a well-marked Tv-S^4R.

For example, Fig. 1 is a PN of Tv-S^4R, where $P_0 = \{p_1, p_7\}$, $P_A = \{p_2\text{ - }p_6, p_8\text{ - }p_{10}\}$, $P_{A_1} = \{p_2\text{ - }p_6\}$, $P_{A_2} = \{p_8\text{ - }p_{10}\}$, $P_R = \{p_{11}\text{ - }p_{15}\}$, $P_{Tv} = \{p_{16}, p_{17}, p_{18}, p_{19}\}$, and $D_{Tv} = [6, 6, 6, 6]^T$. Hereby, D_{Tv} means that the four places in P_{Tv} have a time delay of 6 hours, respectively. Initially, there is one token in p_{16}, which indicates the clock is initialized at 00:00 AM; this token stays there until 6:00 AM before moving to p_{17}, where it stays until 12:00 AM; from 00:00 PM to 12:00 PM, the token will stay at p_{18} and p_{19} for 6 hours respectively. Job types $\mathcal{J}_1$ (resp., $\mathcal{J}_2$) is defined by the set of partially ordered job stages $\{p_2\text{ - }p_6\}$ (resp., $\{p_8\text{ - }p_{10}\}$). The conjunctive resource requirements associated with various job stages in the net in Fig. 1 are as follows: $\alpha_{p_2} = 1 \cdot p_{11} + 0 \cdot p_{12} + 0 \cdot p_{13} + 0 \cdot p_{14} + 0 \cdot p_{15} = p_{11}$, $\alpha_{p_3} = p_{12}$, $\alpha_{p_4} = p_{13}$, $\alpha_{p_5} = p_{14}$, $\alpha_{p_6} = 2 \cdot p_{15}$, $\alpha_{p_8} = p_{15}$, $\alpha_{p_9} = p_{14}$, and $\alpha_{p_{10}} = 2 \cdot p_{13}$.

B. TIME-VARYING SPECIFICATIONS ON TV-S^4R

In Tv-S^4R, each transition in N has two operation modes during a time interval, i.e., being allowed or prohibited. Thereby, Tv-S^4R can realize time-varying job processing without reconfigure manufacturing infrastructure. This comes into being the time-varying specifications. First, we define a subset of time intervals $\widetilde{P}_{Tv} \subseteq P_{Tv}$ which is composed of k time intervals, i.e., $\widetilde{P}_{Tv} = \{p_{Tv_{\beta_1}}, p_{Tv_{\beta_2}}, \ldots, p_{Tv_{\beta_k}}\}$, where $k \in \mathbb{N}^+$, $k \leq |P_{Tv}|$, and $\beta_1, \ldots, \beta_k \in \mathbb{N}_k$. Note that $p_{Tv_{\beta_1}}, p_{Tv_{\beta_2}}, \ldots, p_{Tv_{\beta_k}}$ can represent either continuous or discontinuous k time intervals. In the sequel, $M(\widetilde{P}_{Tv}) = \sum_{i=1}^{k} M(p_{Tv_{\beta_i}})$. Suppose that $t \in T$ is only allowed to fire at the time intervals $\widetilde{P}_{Tv}$.

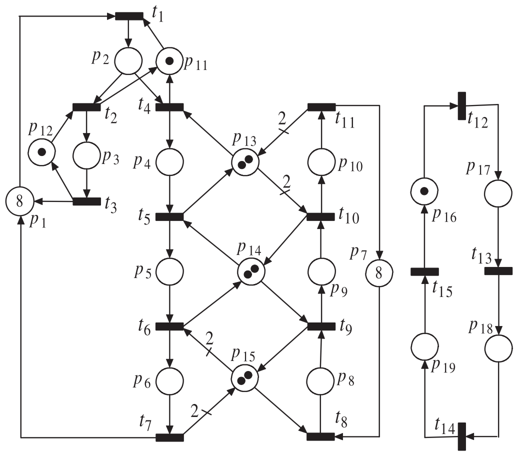

FIGURE 1. An example of Tv-S^4R net.

Definition 4: Given $(\widehat{N}, \widehat{M}_0)$, $M \in R(\widehat{N}, \widehat{M}_0)$ is a reachable marking. A transition t is M-process-enabled iff ${}^\bullet t \cap (P_0 \cup P_A) \neq \emptyset$, and $\forall p \in {}^\bullet t \cap (P_0 \cup P_A)$, $M(p) \neq 0$. A transition t is M-resource-enabled iff ${}^\bullet t \cap P_R \neq \emptyset$, and $\forall r \in {}^\bullet t \cap P_R$, $M(r) \geq W(r, t)$. A transition t is M-time-enabled iff $M(\widetilde{P}_{Tv}) = 1$.

According to Definition 4, a transition t can fire at M if and only if it is M-process-, -resource-, and -time-enabled. It can be seen that the M-process- and -resource-enabledness of t can be detected via PN structure, while the time-enabledness cannot be. Thus, a monitor $\tilde{p}_c$ is designed to time-enable t from structural perspective.

Theorem 1: Let ${}^\bullet\tilde{p}_c = \{t\} \cup {}^\bullet\widetilde{P}_{Tv}$, $\forall t_i \in {}^\bullet\tilde{p}_c$, $W(t_i, \tilde{p}_c) = 1$; $\tilde{p}_c^\bullet = \{t\} \cup \widetilde{P}_{Tv}^\bullet$, $\forall t_j \in \tilde{p}_c^\bullet$, $W(\tilde{p}_c, t_j) = 1$; and $M_0(\tilde{p}_c) = 0$. By implementing $\tilde{p}_c$, t is M-time-enabled when $M(\tilde{p}_c) \neq 0$.

Proof: Because of ${}^\bullet\widetilde{P}_{Tv} \subseteq {}^\bullet\tilde{p}_c$ and $\widetilde{P}_{Tv}^\bullet \subseteq \tilde{p}_c^\bullet$, when $M(\widetilde{P}_{Tv}) = 1$, it follows $M(\tilde{p}_c) = 1$. Then, due to $t \in {}^\bullet\tilde{p}_c$ and $t \in \tilde{p}_c^\bullet$, when $M(\tilde{p}_c) = 1$, t is M-time-enabled. Thus, t is allowed to fire at time intervals $\widetilde{P}_{Tv}$. ■

According to Theorem 1, the construction of $\tilde{p}_c$ is composed of two parts. The first is the parallelization of $\widetilde{P}_{Tv}$, i.e., ${}^\bullet\widetilde{P}_{Tv} \subseteq {}^\bullet\tilde{p}_c$ and $\widetilde{P}_{Tv}^\bullet \subseteq \tilde{p}_c^\bullet$. This ensures $M(\tilde{p}_c) = 1$ when $M(\widetilde{P}_{Tv}) = 1$ while $M(\tilde{p}_c) = 0$ when $M(\widetilde{P}_{Tv}) = 0$. The second is the self-loop between $\tilde{p}_c$ and t. This ensures the M-time-enabledness of t when $M(\tilde{p}_c) = 1$. Note that when $|\widetilde{P}_{Tv}| = 1$, implying t is time-enabled at only one time interval $\widetilde{P}_{Tv}$, $\tilde{p}_c$ is exactly $\widetilde{P}_{Tv}$. In this case, we only need to add a self-loop between t and $\widetilde{P}_{Tv}$. Newly added control place $\tilde{p}_c$ is applicable to the case that t is time-enabled at several continuous or discontinuous time intervals.

Take the PN in Fig. 1 as an example. There are two branches for process $\mathcal{J}_1$, i.e., $\langle t_2, p_3, t_3\rangle$ and $\langle t_4, p_4, \ldots, p_6, t_7\rangle$. Here, the time-varying specifications stipulate that these two branches are working alternatively. To be specifically, the left one is working during the first and third time intervals, while the right one at the second and fourth time intervals. To eliminate confusion, a branch stopping working at a time interval means it does not accept new parts from its previous stage during this time interval. If there have been some parts in this branch, they are allowed to continue processing. According

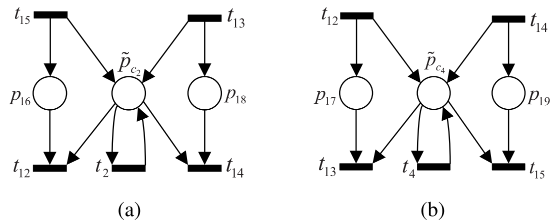

FIGURE 2. Time-varying specification monitors. (a) Monitor $\tilde{p}_{c_2}$ to control t_2. (b) Monitor $\tilde{p}_{c_4}$ to control t_4.

to above specifications, it is equivalent that t_2 is time-enabled when $M(p_{16}) = 1$ or $M(p_{18}) = 1$, while t_4 is time-enabled when $M(p_{17}) = 1$ or $M(p_{19}) = 1$. Consequently, two monitors $\tilde{p}_{c2}$ and $\tilde{p}_{c4}$ are designed to control t_2 and t_4 respectively, as shown in Fig. 2. For $\tilde{p}_{c2}$, it follows that ${}^\bullet\tilde{p}_{c2} = \{t_2, t_{13}, t_{15}\}$, $\tilde{p}_{c2}^\bullet = \{t_2, t_{12}, t_{14}\}$, and $M_0(\tilde{p}_{c2}) = 0$. For $\tilde{p}_{c4}$, it follows that ${}^\bullet\tilde{p}_{c4} = \{t_4, t_{12}, t_{14}\}$, $\tilde{p}_{c4}^\bullet = \{t_4, t_{13}, t_{15}\}$, and $M_0(\tilde{p}_{c4}) = 0$.

The system initially has 1384 states. By adding the time-varying specifications, the system still has 1384 states. Nevertheless, the evolution of reachable markings has been changed. Consider a reachable marking $M_0\ [\sigma\rangle\ M$, where $\sigma = \langle 6t_1 5t_4 3t_5 t_6 t_{12}\rangle$ and $M = [2\ 1\ 0\ 2\ 2\ 1\ 8\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$. Without time-varying specification, t_2 and t_7 are fireable at M and their firings lead to M^1 and M^2 respectively, where $M^1 = [2\ 0\ 1\ 2\ 2\ 1\ 8\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 0\ 1\ 0\ 0]^T$ and $M^2 = [3\ 1\ 0\ 2\ 2\ 0\ 8\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 2\ 0\ 1\ 0\ 0]^T$. Comparatively, with time-varying specification, t_7 is fireable leading to M^2 while t_2 is prevented from firing at M because of $M(p_{17}) = 1$. However, this does not mean M^1 is not reachable. Consider another reachable marking $M_0\ [\sigma'\rangle\ M'$, where $\sigma' = \langle 6t_1 5t_4 3t_5 t_6\rangle$ and $M' = [2\ 1\ 0\ 2\ 2\ 1\ 8\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$. t_2 is M'-process-, -resource-, and -time-enabled. It follows $M'\ [t_2\rangle\ M'^1$, where $M'^1 = [2\ 0\ 1\ 2\ 2\ 1\ 8\ 0\ 0\ 0\ 1\ 0\ 0\ 0\ 0\ 1\ 0\ 0\ 0]^T$. Then, t_{12} is time to fire, leading to $M'^1\ [t_{12}\rangle\ M^1$. Thus, we conclude that the time-varying specification does not restrict any reachable marking while can change the evolution of reachable markings.

C. LIVENESS ANALYSIS OF TV-S^4R

Given the PN model of an AMS, we are often required to ensure liveness with respect to any transition that can eventually fire from any state reachable from M_0. Siphons, as a special structural object, are closely related to liveness. In Tv-S^4R, let $\widehat{\mathcal{S}} = \{\widehat{S}_1, \widehat{S}_2, \cdots, \widehat{S}_{l_1}\}$ be the set of siphons of $\widehat{N}$ and $\mathcal{S} = \{S_1, S_2, \cdots, S_{l_2}\}$ be the set of siphons of N, where $l_1, l_2 \in \mathbb{N}^+$. Then, it follows Lemma 1.

Lemma 1: In Tv-S^4R, we have $\widehat{\mathcal{S}} = \mathcal{S}$.

Proof: Proof by contradiction. N has a set of siphons $\mathcal{S}$ whereas N_{T_v} contains no siphon because it is a one-way loop net which constitutes the support of a P-invariant. Suppose that there is a new siphon S'. It must contain components some of which belong to N and some of which belong to N_{T_v}, respectively. In the initial net, because of $F \cap F_{T_v} = \emptyset$, such that S' does not existent. Therefore, $\widehat{\mathcal{S}} = \mathcal{S}$. ■

Lemma 1 indicates that all the siphons of Tv-S^4R are located in the S^4R part. Theorem 2 discusses the liveness property after adding time-varying specifications.

Theorem 2: By adding monitors enforcing time-varying specifications, the liveness property of the Tv-S^4R does not change.

Proof: Suppose that $\tilde{p}_c$ is a monitor to enforce a time-varying specification on t. According to Theorem 1, $\tilde{p}_c$ is composed of two parts. For the self-loop between $\tilde{p}_c$ and t, when analyzing siphons, these arcs can be removed. For the parallelization with $\widetilde{P}_{Tv}$, according to Lemma 1, adding $\tilde{p}_c$ will not bring in new siphons. Thus, it does not change the liveness property of the Tv-S^4R. ■

Theorem 2 indicates that time-varying specifications and liveness analysis can be implemented independently. Thus, we only need to consider the liveness of S^4R part. The work in [2] pioneers in addressing the liveness enforcement issues for S^4R through the controllability of all siphons during system evolution. Monitors are appropriately associated to the plant model in order to avoid any siphon's undermarkedness. To be self-contained, we cite the following results from [2], it is noted that a siphon refers to a strict minimal one.

Definition 5: 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: 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: (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: 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.

Lemma 2: Given (N, M_0) with monitor p_c, I_S is a P-invariant in $\widehat{N}$. S is max-controlled if $\sum_{p \in P} I_S(p) \cdot M_0(p) > \sum_{p \in P} I_S(p) \cdot (max_{p^\bullet} - 1)$ holds.

For the sake of a Tv-S^4R's liveness-enforcement, each detected potentially undermarked siphon can be associated with a monitor such that all the siphons are max-controlled.

Theorem 3: Given a Tv-S^4R, i.e., $\widehat{N}$, it is live if all siphons in its S^4R counterpart, i.e., N, are max-cs controlled.

Proof: According to Lemma 1, the liveness of Tv-S^4R is equivalent to that of its S^4R part. Thus, according to the work [41], when all the siphons in its S^4R counterpart are max-cs controlled, the S^4R counterpart N is live. Then, the Tv-S^4R net $\widehat{N}$ is live as well. ■

For Tv-S^4R, the time-varying specifications will not influence the liveness analysis. In the sequel, when discussing the liveness of Tv-S^4R, it usually refers to the liveness of the S^4R part.

IV. EVENT-BASED LIVENESS-ENFORCING SUPERVISOR FOR Tv-S^4R AND ITS SIMPLIFICATION

Supervisory control based on P-invariants are widely used in PNs. One typical form is GMECs $L^T \cdot M \leq B$, where $L \in \mathbb{Z}^{n \times n_c}$, $B \in \mathbb{Z}^{n_c}$, n_c is the number of constraints, and n is the number of places. A set of monitors can be designed with incidence matrix $[N_c] = -L^T \cdot [N]$ and $M_0(p_c) = B - L^T \cdot M_0$. Another typical form is GLCs which contain both marking and firing vectors [27], [52]. They are expressed as $L^T \cdot M + H^T \cdot q \leq B$, where $L \in \mathbb{Z}^{n \times n_c}$, $H \in \mathbb{Z}^{m \times n_c}$, $q = [q_1, \ldots, q_m]^T$, for each element $q_i \in \{0, 1\}$, $B \in \mathbb{Z}^{n_c}$, n, m, and n_c are the number of places, transitions, and constraints, respectively. Without loss of generality, L and H are assumed to have only nonnegative elements. Note that q is a vector of Boolean variables. It represents transitions' firing admissibility at current marking M. If t_i is fireable at M, we have $q_i = 1$; otherwise, $q_i = 0$. Consequently, $L^T \cdot M + H^T \cdot q \leq B$ has two meanings: 1) all markings M must satisfy $L^T \cdot M \leq B$, 2) if $M\ [t_i\rangle\ M'$, t_i is allowed to fire only if $L^T \cdot M + H^T \cdot q_i \leq B$ and $L^T \cdot M' \leq B$.

The implementation of GLCs can be realized through proper supervision, resulting in a supervisor. If all the transitions are controllable and observable, the supervisor with regard to a set of inequalities $L^T \cdot M + H^T \cdot q \leq B$ can be enforced as follows [22]:

$$[N_{l_c}^+] = max(\mathbf{0}^T, -L^T \cdot [N]),$$
$$[N_{l_c}^-] = max(\mathbf{0}^T, L^T \cdot [N]);$$

and

$$[N_{p_c}^+] = [N_{l_c}^+] + max(\mathbf{0}^T, H^T - [N_{l_c}^-]),$$
$$[N_{p_c}^-] = max(H^T, [N_{l_c}^-]).$$

The operation max for matrices is defined as follows. For two matrices A and B with the same dimension, $C = max(A, B)$ is the matrix of elements $C_{ij} = max(A_{ij}, B_{ij})$. The structure of a desired supervisor is described by $[N_{p_c}^+]$ and $[N_{p_c}^-]$. Its initial marking $M_0(p_c)$ depends on M_0 of the plant PN as follows:

$$M_0(p_c) = B - L^T \cdot M_0.$$

A. EVENT-BASED LIVENESS-ENFORCING SUPERVISOR FOR Tv-S^4R

When a Tv-S^4R is not live, its reachability graph is divided into LZ and DZ. The former consists of all good markings while the latter includes bad markings and deadlocks. In liveness-enforcing supervisory control, there are two perspectives to synthesize monitors. From state perspective, forbidden state problem is proposed to restrict system's behaviors within LZ. From event perspective, event/state separation problem is proposed, so as to prevent the illegal transitions from firing at good states to bad ones or deadlocks.

To solve the forbidden state problem, the work in [42] proposes an iterative approach. An FBM is a marking residing within DZ and it represents the very first entry from LZ to DZ. At each iteration, an FBM is singled out from the RG of a given PN model. The objective is to prevent this marking from being reached via a P-invariant of the PN. A well-established invariant-based control method is used to derive a control place. This process is carried out until the net model becomes live. The proposed method is generally applicable, easy to use, effective, and straightforward. Inspired by this method, we propose a more precise control method called event-based control, which is dedicated to the event/state separation problem.

In reachability graph, compared with FBM, we define a set of critical good markings (CGMs) which are found in the LZ and considered to be the exits of LZ to the FBM. In other words, for each CGM, there exists a transition $t_i \in T$ such that CGM $[t_i\rangle$ FBM. Such transitions are denoted as illegal events or transitions. Then, our objective is to prevent the illegal transition from firing at corresponding CGM, so as to realize event/state separation. This can be realized by event-based control in terms of a GLC. Thus, at each iteration, we are going to identify the set of CGMs and their illegal transitions, and establish a set of GLCs to prevent the system from reaching DZ. According to the work in [42], we know that the iterations terminate in a finite number of times for a bounded PN for a flexible manufacturing system, leading to a live controlled PN. Similarly, we can also claim that our method can terminate in a finite number of iterations.

Suppose $M\ [t_i\rangle\ M^*$, where M and M^* are CGM and FBM respectively, and t_i is the illegal transition. The number of tokens in the marked subset of the activity places at M represents the last exit into DZ. If t_i is supposed to fire, the token distributions of the activity places at M^* reflect the first entry into DZ. Thus, t is enabled at M and should be prevented by the monitor. Then, we establish a GLC $l^T \cdot M + h^T \cdot q \leq b$, where $\forall p \in P_A$ and $M(p) > 0$, $l(p) = 1$, $h(t_i) = 1$, and $b = \sum_{p \in P_A} M(p)$. In other words, $l^T \cdot M$ characterizes the marked subset of activity places at M; $h^T \cdot q$ reflects the forbidden transition at M; and b is the threshold of the sum of tokens in the marked activity places. The GLC constructs a generalized P-invariant for the PN [22]. We can find that M is retained in LZ by the GLC, while M^* is eliminated from reachability graph.

To formalize our method, we first define some symbols. The original model is denoted as (N^0, M_0^0), whose reachability graph, CGMs and FBMs are denoted as RG^0, CGM^0, and FBM^0, respectively. The monitors for the control of CGM^0 is denoted as p_c^0. p_c^0 will be superimposed on the original system, composing the first-controlled system (N^1, M_0^1). Similarly, in the i-th iteration, the controlled system, its reachability graph, CGMs, FBMs, and generated monitors are denoted as (N^i, M_0^i), RG^i, CGM^i, FBM^i, and p_c^i, respectively. The iterative processes are executed until the system is live. According to above description, Algorithm 1 is designed. Since our PN is bounded, Algorithm 1 can terminate in finite steps. The computation complexity of Algorithm 1 is exponential since the reachability graph is required here.

Algorithm 1 is used for the PN in Fig. 1. We only consider the liveness of the S^4R part. Initially, N is not live with 348 states including 264 good states in LZ and 84 bad states in DZ. After 5 iterations resulting in 10 monitors, the controlled system is live with 220 ($220/264 = 83.33\%$) states.

Algorithm 1 Liveness Enforcement Based on Event-Based Control

```
Input: Original system (N^0, M_0^0);
Output: Liveness-enforcing supervisor;
i := 0;
while the system (N^i, M_0^i) is not live do
    Generate the reachability graph RG^i;
    Find FBM^i, CGM^i, and corresponding illegal
    transitions;
    Build the event-based control l_i^T · M + h_i^T · q ≤ b_i;
    Compute supervisor p_c^i for system (N^i, M_0^i) and
    superimpose it on (N^i, M_0^i);
    i := i + 1;
```

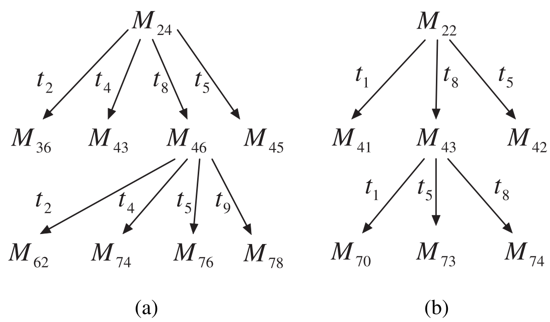

FIGURE 3. Partial reachability graphs of FBM and CGMs of PN^0. (a) Example CGM #1. (b) Example CGM #2.

Table 1 shows the results of all 5 iterations. Note that we only present the state of initial and activity places here. Take the first iteration as an example. The CGM^0, FBM^0, and corresponding illegal transitions are M_{43} and M_{46}, M_{74}, and t_1 and t_4, respectively. Their partial reachability graphs are shown in Fig. 3. In Fig. 3(a), M_{24} is a good marking and all the transitions can fire if they are enabled. Thereby, we have $M_{24}\ [t_4\rangle\ M_{46}$. $M_{46} = [6\ 1\ 0\ 1\ 0\ 0\ 6\ 2\ 0\ 0\ 0\ 1\ 1\ 2\ 0]^T$ is a CGM because of $M_{46}\ [t_4\rangle\ M_{74}$. Then, t_4 should be prevented at M_{46}. We have $M(p_2) + M(p_4) + M(p_8) + q_4 \leq 4$, resulting in the monitor $[N^+_{p_c^{1_1}}] = t_1 + t_4 + t_8$, $[N^-_{p_c^{1_1}}] = t_2 + t_4 + t_5 + t_9$, and $M_0(p_c^{1_1}) = 4$. Fig. 3(b) shows another route to the FBM, i.e., $M_{22}\ [t_8\rangle\ M_{43}\ [t_8\rangle\ M_{74}$. Then we have $M(p_4) + M(p_8) + q_8 \leq 3$, leading to the monitor $[N^+_{p_c^{1_2}}] = t_4 + t_8$, $[N^-_{p_c^{1_2}}] = t_5 + t_9$, and $M_0(p_c^{1_2}) = 3$.

Hereby, we also use state-based control for comparison [42]. It can be realized in form of $L^T \cdot M \leq B$. After 6 iterations, we can get 6 monitors achieving 208 ($208/264 = 78.79\%$) good states. For example, in (N^0, M_0^0), according to $M_{74} = [6\ 0\ 0\ 2\ 0\ 0\ 6\ 2\ 0\ 0\ 1\ 1\ 0\ 2\ 0]^T$, a GMEC is obtained: $M(p_4) + M(p_8) \leq 3$. Then we have $[N^+_{p_c^1}] = t_5 + t_9$, $[N^-_{p_c^1}] = t_4 + t_8$, and $M_0(p_c^1) = 3$. Although it obtains a smaller supervisor, this method preserves less good markings compared with the event-based method.

TABLE 1. Results of iterations using event-based control for PN in Fig. 1.

i	States	FBM^i	$CGM^i\ [q^i\rangle$	Event-based control constraints	M_0	${}^\bullet p_{c_i}$	$p_{c_i}^\bullet$
1	348	M_{74}	$M_{43}=[6\,0\,0\,2\,0\,0\,7\,1\,0\,0]\ [t_8\rangle$	$p_4+p_8+q_8\le 3$	3	$\{t_5,t_9\}$	$\{t_4,t_8\}$
			$M_{46}=[6\,1\,0\,1\,0\,0\,6\,2\,0\,0]\ [t_4\rangle$	$p_2+p_4+p_8+q_4\le 4$	4	$\{t_2,t_4,t_5,t_9\}$	$\{t_1,t_4,t_8\}$
2	324	M_{82}	$M_{50}=[7\,0\,0\,1\,0\,0\,6\,1\,1\,0]\ [t_9\rangle$	$p_4+p_8+p_9+q_9\le 3$	3	$\{t_5,t_9,t_{10}\}$	$\{t_4,t_8,t_9\}$
			$M_{52}=[7\,1\,0\,0\,0\,0\,6\,0\,2\,0]\ [t_4\rangle$	$p_2+p_9+q_4\le 3$	3	$\{t_2,2\cdot t_4,t_{10}\}$	$\{t_1,t_4,t_9\}$
3	280	M_{106}	$M_{71}=[6\,0\,0\,0\,2\,0\,8\,0\,0\,0]\ [t_8\rangle$	$p_5+q_8\le 2$	2	$\{t_6,t_8\}$	$\{t_5,t_8\}$
			$M_{73}=[6\,0\,0\,1\,1\,0\,7\,1\,0\,0]\ [t_5\rangle$	$p_4+p_5+p_8+q_5\le 3$	3	$\{t_5,t_6,t_9\}$	$\{t_4,t_5,t_8\}$
4	244	M_{27}	$M_{13}=[7\,0\,0\,1\,0\,0\,7\,1\,0\,0]\ [t_8\rangle$	$p_4+p_8+q_8\le 2$	2	$\{t_5,t_9\}$	$\{t_4,t_8\}$
			$M_{14}=[7\,1\,0\,0\,0\,0\,6\,2\,0\,0]\ [t_4\rangle$	$p_2+p_8+q_4\le 3$	3	$\{t_2,2\cdot t_4,t_9\}$	$\{t_1,t_4,t_8\}$
5	236	M_{131}	$M_{99}=[6\,0\,0\,1\,1\,0\,7\,0\,1\,0]\ [t_8\rangle$	$p_4+p_5+p_9+q_8\le 3$	3	$\{t_6,t_8,t_{10}\}$	$\{t_4,t_8,t_9\}$
			$M_{100}=[6\,1\,0\,0\,1\,0\,6\,1\,1\,0]\ [t_4\rangle$	$p_2+p_5+p_8+p_9+q_4\le 4$	4	$\{t_2,2\cdot t_4,t_6,t_9,t_{10}\}$	$\{t_1,t_4,t_5,t_8,t_9\}$
6	220						

B. EVENT-BASED SUPERVISOR SIMPLIFICATION

For the inequalities $L^T \cdot M + H^T \cdot q \le B$, they can be written in another way: $[L^T, H^T] \cdot [M; q] \le B$, where $[L^T, H^T]$ and $[M; q]$ are composition matrices. However, $[M; q]$ cannot be considered as M in $L^T \cdot M \le B$ but a special situation of M, because q is a binary vector while M is a nonnegative integer vector. Here we extend the meaning of the theorem in [20].

Definition 8: Let $L^T \cdot M + H^T \cdot q \le B$ be a set of inequalities, $\Omega = \{[M; q] \mid l_i^T \cdot M + h_i^T \cdot q \le b_i, \forall i \in \mathbb{N}_n\}$ and $\Omega_{\mathbb{N}_n\backslash\{k\}} = \{[M; q] \mid l_i^T \cdot M + h_i^T \cdot q \le b_i, \forall i \in \mathbb{N}_n - \{k\}\}$. $l_k^T \cdot M + h_k^T \cdot q \le b_k$ is said to be dependent on other inequalities iff $\Omega = \Omega_{\mathbb{N}_n\backslash\{k\}}$.

Proposition 2: An inequality $l_k^T \cdot M + h_k^T \cdot q \le b_k$ is dependent on the others iff $min\{b_k - l_k^T \cdot M - h_k^T \cdot q, [M; q] \in \Omega_{\mathbb{N}_n\backslash\{k\}}\} \ge 0$.

Proof: For the necessity part, according to Definition 8, because $l_k^T \cdot M + h_k^T \cdot q \le b_k$ is dependent, we have $\Omega = \Omega_{\mathbb{N}_n\backslash\{k\}}$. Thus, $min\{b_k - l_k^T \cdot M - h_k^T \cdot q, [M; q] \in \Omega\} \ge 0$ holds, implying $min\{b_k - l_k^T \cdot M - h_k^T \cdot q, [M; q] \in \Omega_{\mathbb{N}_n\backslash\{k\}}\} \ge 0$.

For the sufficiency part, $min\{b_k - l_k^T \cdot M - h_k^T \cdot q, [M; q] \in \Omega_{\mathbb{N}_n\backslash\{k\}}\} \ge 0$ means that for all $[M; q] \in \Omega_{\mathbb{N}_n\backslash\{k\}}$, $l_k^T \cdot M + h_k^T \cdot q \le b_k$ holds. Thus, we have $\Omega = \Omega_{\mathbb{N}_n\backslash\{k\}}$. According to Definition 8, $l_k^T \cdot M + h_k^T \cdot q \le b_k$ is dependent. ∎

Theorem 4: Let $L^T \cdot M + H^T \cdot q \le B$ be a set of inequalities. For $k \in \mathbb{N}_n$, inequality $l_k^T \cdot M + h_k^T \cdot q \le b_k$ is dependent on the others iff there exist nonnegative coefficients α_i, $i \in \mathbb{N}_n - \{k\}$ such that $[l_k^T, h_k^T] \le \sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot [l_i^T, h_i^T]$ and $b_k \ge \sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot b_i$.

Proof: For the necessity part, we assume that $l_k^T \cdot M + h_k^T \cdot q \le b_k$ is dependent on the others. According to Proposition 2, we have $min\{b_k - l_k^T \cdot M - h_k^T \cdot q, [M; q] \in \Omega_{\mathbb{N}_n\backslash\{k\}}\} \ge 0$. Based on the duality theorem, this is equivalent to $max\{l_k^T \cdot M + h_k^T \cdot q - b_k, [M; q] \in \Omega_{\mathbb{N}_n\backslash\{k\}}\} \le 0$. This implies that there exists an optimal solution $[M^*; q^*]$ such that $l_k^T \cdot M^* + h_k^T \cdot q^* \le b_k$. The dual mathematical programming is $min\{\sum_{i\in\mathbb{N}_n-\{k\}}\alpha_i \cdot b_i, \sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot [l_i^T, h_i^T] \ge [l_k^T, h_k^T], \alpha_i \ge 0, \forall i \in \mathbb{N}_n\backslash\{k\}\}$. As a result, there exists an optimal solution α_i, $i \in \mathbb{N}_n\backslash\{k\}$ such that $\sum_{i\in\mathbb{N}_n-\{k\}}\alpha_i \cdot b_i = l_k^T \cdot M + h_k^T \cdot q \le b_k$, $\sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot [l_i^T, h_i^T] \ge [l_k^T, h_k^T]$.

For the sufficiency part, we have $[l_k^T, h_k^T] \le \sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot [l_i^T, h_i^T]$ and $b_k \ge \sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot b_i$ according to the hypothesis. For any $[M; q] \ge \mathbf{0}$, we have $l_k^T \cdot M + h_k^T \cdot q \le \sum_{i\in\mathbb{N}_n-\{k\}}\alpha_i \cdot (l_i^T \cdot M + h_i^T \cdot q) \le \sum_{i\in\mathbb{N}_n-\{k\}}\alpha_i \cdot b_i$. According to Definition 7, we know that $l_k^T \cdot M + h_k^T \cdot q \le b_k$ is dependent on other inequalities, i.e., $l_i^T \cdot M + h_i^T \cdot q \le b_i$, where $i \in \mathbb{N}_n - \{k\}$. ∎

TABLE 2. Division of 10 inequalities synthesized by event-based control.

	q_i	Inequalities	Indep.
1	q_4	$p_2 + p_4 + p_8 + q_4 \le 4$	Yes
		$p_2 + p_9 + q_4 \le 3$	Yes
		$p_2 + p_8 + q_4 \le 3$	Yes
		$p_2 + p_5 + p_8 + p_9 + q_4 \le 4$	Yes
2	q_5	$p_4 + p_5 + p_8 + q_5 \le 3$	Yes
3	q_8	$p_4 + p_8 + q_8 \le 3$	No
		$p_5 + q_8 \le 2$	Yes
		$p_4 + p_8 + q_8 \le 2$	Yes
		$p_4 + p_5 + p_9 + q_8 \le 3$	Yes
4	q_9	$p_4 + p_8 + p_9 + q_9 \le 3$	Yes

Therefore, to reduce the supervisor size, Algorithm 2 is programmed to identify these dependent and independent GLCs. Suppose that there are K GLCs where $K \in \mathbb{N}^+$. The identification of the set of nonnegative coefficients α_i can be realized by linear programming $min \sum_{i=1}^{n-1} \alpha_i$, subject to $[l_k^T, h_k^T] \le \sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot [l_i^T, h_i^T]$, $b_k \ge \sum_{i\in\mathbb{N}_n-\{k\}} \alpha_i \cdot b_i$, $\alpha_i \ge 0$. Thus, the computation complexity of Algorithm 2 is polynomial with regard to the number of inequalities.

$[l^T, h^T]$ and $[M; q]$ are $(|P| + |T|)$ dimensional vectors. If the vector dimension can be decreased, the computational complexity of the algorithm can be decreased as well. Considering that the firing vectors in these inequalities only have one element in our circumstance, we can divide these inequalities into several sets according to the same firing vector support.

For example, in the preceding analysis, we obtain 10 supervisors. They are divided into 4 sets, as shown in Table 2. For the 3-th set, q_8 is the same firing vector element, and there are 4 inequalities: $M(p_4) + M(p_8) + q_8 \le 3$, $M(p_5) + q_8 \le 2$, $M(p_4) + M(p_8) + q_8 \le 2$, and $M(p_4) + M(p_5) + M(p_9) + q_8 \le 3$. $M(p_4) + M(p_8) + q_8 \le 3$ is dependent on the other 3 ones. This is because $(M(p_4) + M(p_8) + q_8) \le 0 \cdot (M(p_5) + q_8) + 1 \cdot (M(p_4) + M(p_8) + q_8) + 0 \cdot (M(p_4) + M(p_5) + M(p_9) + q_8) + 0 \cdot (M(p_2) + M(p_3) + M(p_5) + M(p_8) + q_8)$

Algorithm 2 Identification of Independent and Dependent GLCs

Input: $L^T \cdot M + H^T \cdot q \leq B$, or equivalently, $[L^T, H^T] \cdot [M; q] \leq B$;
Output: Independent GLCs $L_{\mathcal{I}}^T \cdot M + H_{\mathcal{I}}^T \cdot q \leq B_{\mathcal{I}}$ and dependent GLCs $L_{\mathcal{D}}^T \cdot M + H_{\mathcal{D}}^T \cdot q \leq B_{\mathcal{D}}$;
$i := 1$, $m := 0$, $\mathcal{L}'_{\mathcal{I}} := \emptyset$, $\mathcal{L}'_{\mathcal{D}} := \mathcal{L}'$, $\mathcal{B}_{\mathcal{I}} := \emptyset$, and $\mathcal{B}_{\mathcal{D}} := \mathcal{B}$;
Arrange all the GLCs according to the ascending (resp., descending) order of $|l_i|$;
while $i \leq K$ **do**
  if $i = 1$ **then**
    $m := m + 1$, $\mathcal{L}'_{\mathcal{I}} := \mathcal{L}'_{\mathcal{I}} \cup \{[l_i^T, h_i^T]\}$, $\mathcal{L}'_{\mathcal{D}} := \mathcal{L}'_{\mathcal{D}} \setminus \{[l_i^T, h_i^T]\}$, $\mathcal{B}_{\mathcal{I}} := \mathcal{B}_{\mathcal{I}} \cup \{b_i\}$, $\mathcal{B}_{\mathcal{D}} := \mathcal{B}_{\mathcal{D}} \setminus \{b_i\}$;
  else if $\exists \alpha_j \geq 0$ && $[l_i^T, h_i^T] \leq \sum_{j=1}^{m} \alpha_j \cdot [l_j^T, h_j^T]$ && $b_i \geq \sum_{j=1}^{m} \alpha_j \cdot b_j$ **then**
    $m := m + 1$, $\mathcal{L}'_{\mathcal{I}} := \mathcal{L}'_{\mathcal{I}} \cup \{[l_i^T, h_i^T]\}$, $\mathcal{L}'_{\mathcal{D}} := \mathcal{L}'_{\mathcal{D}} \setminus \{[l_i^T, h_i^T]\}$, $\mathcal{B}_{\mathcal{I}} := \mathcal{B}_{\mathcal{I}} \cup \{b_i\}$, $\mathcal{B}_{\mathcal{D}} := \mathcal{B}_{\mathcal{D}} \setminus \{b_i\}$;
  $i := i + 1$;

and $3 \geq 0 \cdot 2 + 1 \cdot 2 + 0 \cdot 3 + 0 \cdot 4$. The Sets 2 and 4 have only one inequality, respectively; thus, these inequalities are independent. Set 1 has no dependent inequality. At last, there are 9 monitors retained, preserving 220 states.

According to above analysis, the GLCs provide an event perspective to deal with liveness-enforcing supervisory control problem via constructing P-invariants. In event-based control, the illegal transition is prevented from firing at the CGM, resulting in an event/state separation instance. However, each instance requires a GLC and leads to a monitor. Thus, the event-based control enhances our specifications' expressivity capability while suffers from a huge size supervisor. Comparatively, the state-based control in terms of GMECs has a more compact supervisor. For the permissiveness, an FBM is optimal if and only if none of the good markings in the RG marks equally the activity places at this FBM [42]. From the illustrative examples in [42], we can find that the state-based control is optimal or almost optimal for most Systems of Simple Sequential Processes with Resources (S^3PR) nets. For general net, it cannot guarantee the optimality in general. Some good states may violate the constraint which is going to prohibit FBM. For the event-based control, on one hand, some good states may violate the state specification part, i.e., $l^T \cdot M$. On the other hand, the transition to be prevented at CGM M^* is also prevented at a good marking, thus removing the following good markings. Thus, no matter in GMECs or GLCs, good markings may be unintentionally removed.

V. WEIGHTED INVARIANT-BASED LIVENESS-ENFORCING SUPERVISOR FOR TV-S^4R

Note that both GMECs for FBMs and GLCs for CGMs as well as illegal transitions concentrate on the subset of activity places which are marked at FBMs or CGMs respectively. We can find that their weight coefficients are unitary for these activity places. In this section, we introduce weight coefficients in invariant-based liveness-enforcing supervisors, to explore more concise specification expressions.

A. MATHEMATICAL ANALYSIS

When establishing a linear constraint for a marking, our principle is that the activity place which has more tokens should possess the higher weight coefficients. Based on this principle, a weight coefficient of an activity place is designed based on the proportion of the number of the tokens in this activity place to the sum of those in the subset of activity places.

In the state-based control, suppose that M^* is an FBM. The sum of tokens in the set of the activity places is denoted as $\Phi_{M^*} = \sum_{i=1}^{|P_A|} M^*(p_i)$. Then, the weight coefficient of an activity place p_i is represented as $\frac{M^*(p_i)}{\Phi_{M^*}}$. Thus, the weighted sum of tokens in the set of activity places at a marking M is represented as $\sum_{i=1}^{|P_A|} \frac{M^*(p_i)}{\Phi_{M^*}} \cdot M(p_i)$. It should be less than the weighted sum of tokens within the subset of the activity places at M^*. Since $\frac{1}{\Phi_{M^*}}$ is the least weight coefficient, we set the scalar as $\sum_{i=1}^{|P_A|} \frac{M^*(p_i)}{\Phi_{M^*}} \cdot M^*(p_i) - \frac{1}{\Phi_{M^*}}$. Thus, the weighted inequality is expressed as follows.

$$\sum_{i=1}^{|P_A|} \frac{M^*(p_i)}{\Phi_{M^*}} \cdot M(p_i) \leq \sum_{i=1}^{|P_A|} \frac{M^*(p_i)}{\Phi_{M^*}} \cdot M^*(p_i) - \frac{1}{\Phi_{M^*}}. \quad (1)$$

By multiplying Φ_{M^*} on both sides, it implies that

$$\sum_{i=1}^{|P_A|} M^*(p_i) \cdot M(p_i) \leq \sum_{i=1}^{|P_A|} M^*(p_i) \cdot M^*(p_i) - 1. \quad (2)$$

It can be easily checked that M^*, as an FBM, is prevented when enforcing Inequality (2). Consequently, the bad markings reachable from M^* are prevented as well.

Similarly, in the event-based control, we suppose that M^* is a CGM and t_i is the illegal transition. To prevent the firing of t_i at M^*, the weighted inequality is stated as:

$$\sum_{i=1}^{|P_A|} \frac{M^*(p_i)}{\Phi_{M^*}} \cdot M(p_i) + \frac{1}{\Phi_{M^*}} \cdot q_i \leq \sum_{i=1}^{|P_A|} \frac{M^*(p_i)}{\Phi_{M^*}} \cdot M^*(p_i). \quad (3)$$

By multiplying Φ_{M^*} on both sides, it implies that

$$\sum_{i=1}^{|P_A|} M^*(p_i) \cdot M(p_i) + q_i \leq \sum_{i=1}^{|P_A|} M^*(p_i) \cdot M^*(p_i). \quad (4)$$

In Inequality (4), when the system is at the CGM $M = M^*$, it can be easily verified that q_i is prevented from firing at this moment. Therefore, the sequential bad markings would not be reachable.

As we know, some good markings may be unintentionally removed by the inequalities in the state- and event-based control. To decrease the probability of removing the good markings, the weighted linear constraints are developed. Until

TABLE 3. **Four types of inequalities to prevent the system from reaching the FBM.**

Type	Inequality for CGM as well as t_i or FBM
I	$\sum_{i=1}^{K} M(p_{\varepsilon_i}) \leq \Phi_{M^*} - 1$
II	$\sum_{i=1}^{K} M(p_{\varepsilon_i}) + q_i \leq \Phi_{M^*}$
III	$\sum_{i=1}^{\lvert P_A \rvert} M^*(p_i) \cdot M(p_i) \leq \sum_{i=1}^{\lvert P_A \rvert} M^*(p_i) \cdot M^*(p_i) - 1$
IV	$\sum_{i=1}^{\lvert P_A \rvert} M^*(p_i) \cdot M(p_i) + q_i \leq \sum_{i=1}^{\lvert P_A \rvert} M^*(p_i) \cdot M^*(p_i)$

now, we develop three kinds of linear constraints to separate the good and bad markings. Note that when enforcing the non-weighted linear constraints, we first need to identify the marked activity places at M^*, while this step can be omitted when designing the weighted linear constraints, since the empty places have the zero weights automatically. Suppose that $\{p_{\varepsilon_1}, \ldots, p_{\varepsilon_K}\} \subseteq P_A$, where $\{\varepsilon_1, \ldots, \varepsilon_K\} \subseteq \mathbb{N}^+$ and $K \leq |P_A|$, is the subset of activity places that are marked at M^*. For better comparison, combined with the state-based control, these four linear constraints are presented in Table 3. The state-based, event-based, weighted state-based, and weighted event-based are denoted as Types I, II, III, and IV, respectively.

B. LIVENESS-ENFORCING SUPERVISORY CONTROL

Then, the weighted inequalities are applied to their iterative processes respectively. For the event-based control, the weighted GLCs are applied to Algorithm 1. The supervisor simplification on GLCs is also suitable for the weighted GLCs. For the state-based control, when obtaining a set of GMECs, an algebraic method can deal with supervisor simplification work, as addressed in [20].

Definition 9: Let $L^T \cdot M \leq B$ be a set of inequalities, $\mathcal{M} = \{M \mid l_i^T \cdot M \leq b_i, \forall i \in \mathbb{N}_n\}$, and $\mathcal{M}_{\mathbb{N}_n \setminus \{k\}} = \{M \mid l_i^T \cdot M \leq b_i, \forall i \in \mathbb{N}_n - \{k\}\}$. $l_k^T \cdot M \leq b_k$ is said to be dependent on other inequalities iff $\mathcal{M} = \mathcal{M}_{\mathbb{N}_n \setminus \{k\}}$.

Theorem 5: 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 $b_k \geq \sum_{i \in \mathbb{N}_n \setminus \{k\}} \alpha_i \cdot b_i$.

For the proof, please refer to [20]. By this method, given any set of inequalities, we can identify the independent and dependent ones by comparing their coefficients. Consequently, we present the unification of the state-based, event-based, weighted state-based, and weighted event-based control methods as well as their supervisor simplifications, which is in form of linear constraints and implemented by enforcing P-invariants.

For unification, suppose that the inequality $\xi^T \cdot \Upsilon \leq b$ represents any type of inequality in Table 3. To be specific, when representing the one of Type I or III, ξ is equal to l and Υ to M, while when representing the one of Type II or IV, ξ is equal to $[l; h]$ and Υ to $[M; q]$. Then, we establish a unified algorithm to combine the supervisor synthesis and simplification. Given a set of inequalities $\Xi^T \cdot \Upsilon \leq B$, i.e., $\{\xi_i^T \cdot \Upsilon \leq b_i, i \in \mathbb{N}_K\}$, let $\Xi_{\mathcal{I}}^T \cdot \Upsilon \leq B_{\mathcal{I}}$ and $\Xi_{\mathcal{D}}^T \cdot \Upsilon \leq B_{\mathcal{D}}$ be the sets of independent and dependent inequalities respectively. The unified and combined procedures are addressed in Algorithm 3. It consists of two stages. The former containing Steps 4 – 9 focuses on supervisor synthesis, while the latter containing Steps 10 – 16 solves supervisor simplification problem. Note that $\xi^T \cdot \Upsilon \leq b$ generated in each iteration is of the same type.

Algorithm 3 Simplified Liveness-enforcing Supervisor Synthesis

1 Input: Original system (N^0, M_0^0);
2 Output: Simplified liveness-enforcing supervisor $\Xi_{\mathcal{I}}^T \cdot \Upsilon \leq B_{\mathcal{I}}$;
3 $i := 0$, $K := 0$, $m := 0$, $\Xi_{\mathcal{I}} := \emptyset$, $\Xi_{\mathcal{D}} := \Xi$, $B_{\mathcal{I}} := \emptyset$, and $B_{\mathcal{D}} := B$;
4 **while** the system (N^i, M_0^i) is not live **do**
5   Generate the reachability graph RG^i;
6   Find FBM^i, CGM^i, and corresponding illegal transitions;
7   Build an inequality $\xi_i^T \cdot \Upsilon \leq b_i$ which is of one type of Table 3 to prevent the system from reaching the FBM^i;
8   Compute supervisor p_c^i for system (N^i, M_0^i) and superimpose it on (N^i, M_0^i);
9     $i := i + 1$, $K := K + 1$;
10 Arrange all the K inequalities $\Xi^T \cdot \Upsilon \leq B$ according to the ascending (resp., descending) order of $|\xi_i|$;
11 **while** $k \leq K$ **do**
12   **if** $k = 1$ **then**
13         $m := m + 1$, $\Xi_{\mathcal{I}} := \Xi_{\mathcal{I}} \cup \{\xi_k\}$, $\Xi_{\mathcal{D}} := \Xi_{\mathcal{D}} \setminus \{\xi_k\}$, $B_{\mathcal{I}} := B_{\mathcal{I}} \cup \{b_k\}$, $B_{\mathcal{D}} := B_{\mathcal{D}} \setminus \{b_k\}$;
14   **else if** $\exists \alpha_j \geq 0$ && $\xi_k \leq \sum_{j=1}^{m} \alpha_j \cdot \xi_j$ && $b_k \geq \sum_{j=1}^{m} \alpha_j \cdot b_j$ **then**
15         $m := m + 1$, $\Xi_{\mathcal{I}} := \Xi_{\mathcal{I}} \cup \{\xi_k\}$, $\Xi_{\mathcal{D}} := \Xi_{\mathcal{D}} \setminus \{\xi_k\}$, $B_{\mathcal{I}} := B_{\mathcal{I}} \cup \{b_k\}$, $B_{\mathcal{D}} := B_{\mathcal{D}} \setminus \{b_k\}$;
16     $k := k + 1$;

Then, we take the PN in Fig. 1 as an example. By using the weighted state-based method, the controlled net is live with 224 (224/264 = 84.85%) states after 6 iterations. For example, consider the FBM $M_{74} = [6\,0\,0\,2\,0\,0\,6\,2\,0\,0\,1\,1\,0\,2\,0]^T$ in the first iteration. In state-based method, the GMEC is established as $M(p_4) + M(p_8) \leq 3$. While in weighted state-based method, it is prevented by the inequality $2 \cdot M(p_4) + 2 \cdot M(p_8) \leq 7$. For comparison, the results are shown in Table 4. Also, all the weighted linear constraints are independent. Finally, it can be seen that it has a better control result in permissiveness compared with the state-based and event-based methods. Then, the weighted event-based method is used for this PN. After 5 iterations, the controlled net is live with 220 states, as the same result with the event-based method. The results are shown in Table 5. For this example, the weighted state-based method is priority to the others since it has the smallest supervisor while retains the most good states.

To evaluate the performance of the four P-invariant-based control methods, we provide two more examples.

TABLE 4. Results of weighted state-based control for PN in Fig. 1.

i	#States	FBM	Inequality	Weighted Inequality	M_0	$^\bullet p_{c_i}$	$p_{c_i}^\bullet$	Indep.
1	348	$M_{74}=[6\,0\,0\,2\,0\,0\,6\,2\,0\,0]$	$p_4{+}p_8{\le}3$	$2{\cdot}p_4{+}2{\cdot}p_8{\le}7$	7	$\{2{\cdot}t_5,2{\cdot}t_9\}$	$\{2{\cdot}t_4,2{\cdot}t_8\}$	Yes
2	324	$M_{82}=[7\,0\,0\,1\,0\,0\,6\,0\,2\,0]$	$p_4{+}p_9\le 2$	$p_4{+}2{\cdot}p_9{\le}\,4$	4	$\{t_5,2{\cdot}t_{10}\}$	$\{t_4,2{\cdot}t_9\}$	Yes
3	304	$M_{109}=[6\,0\,0\,0\,2\,0\,7\,1\,0\,0]$	$p_5{+}p_8{\le}2$	$2{\cdot}p_5{+}p_8{\le}4$	4	$\{2{\cdot}t_6,t_9\}$	$\{2{\cdot}t_5,t_8\}$	Yes
4	284	$M_{109}=[6\,0\,0\,1\,1\,0\,6\,2\,0\,0]$	$p_4{+}p_5{+}p_8{\le}3$	$p_4{+}p_5{+}2{\cdot}p_8{\le}5$	5	$\{t_6,2{\cdot}t_9\}$	$\{t_4,2{\cdot}t_8\}$	Yes
5	276	$M_{110}=[6\,0\,0\,2\,0\,0\,6\,1\,1\,0]$	$p_4{+}p_8{+}p_9{\le}3$	$2{\cdot}p_4{+}p_8{+}p_9{\le}5$	5	$\{2{\cdot}t_5,t_{10}\}$	$\{2{\cdot}t_4,t_8\}$	Yes
6	268	$M_{146}=[6\,0\,0\,1\,1\,0\,6\,1\,1\,0]$	$p_4{+}p_5{+}p_8{+}p_9{\le}3$	$p_4{+}p_5{+}p_8{+}p_9{\le}3$	3	$\{t_6,t_{10}\}$	$\{t_4,t_8\}$	Yes
7	224							

TABLE 5. Results of weighted event-based control for PN in Fig. 1.

i	#States	GCM $[t_i\rangle$	Weighted Inequality	M_0	$^\bullet p_{c_i}$	$p_{c_i}^\bullet$	Indep.
1	348	$M_{43}=[6\,0\,0\,2\,0\,0\,7\,1\,0\,0]\ [t_8\rangle$	$2{\cdot}p_4{+}p_8{+}q_8{\le}5$	5	$\{2{\cdot}t_5,t_9\}$	$\{2{\cdot}t_4,t_8\}$	Yes
		$M_{46}=[6\,1\,0\,1\,0\,0\,6\,2\,0\,0]\ [t_4\rangle$	$p_2{+}p_4{+}2{\cdot}p_8{+}q_4{\le}6$	6	$\{t_2,t_4,t_5 2{\cdot}t_9\}$	$\{t_1,t_4 2{\cdot}t_8\}$	Yes
2	324	$M_{50}=[7\,0\,0\,1\,0\,0\,6\,1\,1\,0]\ [t_9\rangle$	$p_4{+}p_8{+}p_9{+}q_9{\le}3$	3	$\{t_5,t_9,t_{10}\}$	$\{t_4,t_8,t_9\}$	Yes
		$M_{52}=[7\,1\,0\,0\,0\,0\,6\,0\,2\,0]\ [t_4\rangle$	$p_2{+}2{\cdot}p_9{+}q_4{\le}5$	5	$\{t_2,2{\cdot}t_4,2{\cdot}t_{10}\}$	$\{t_1,t_4,2{\cdot}t_9\}$	Yes
3	280	$M_{70}=[6\,0\,0\,0\,2\,0\,8\,0\,0\,0]\ [t_8\rangle$	$2{\cdot}p_5{+}q_8{\le}4$	4	$\{2{\cdot}t_6,t_8\}$	$\{2{\cdot}t_5,t_8\}$	Yes
		$M_{72}=[6\,0\,0\,1\,1\,0\,7\,1\,0\,0]\ [t_5\rangle$	$p_4{+}p_5{+}p_8{+}q_5{\le}3$	3	$\{t_5,t_6,t_9\}$	$\{t_4,t_5,t_8\}$	Yes
4	244	$M_{13}=[7\,0\,0\,1\,0\,0\,7\,1\,0\,0]\ [t_8\rangle$	$p_4{+}p_8{+}q_8{\le}2$	2	$\{t_5,t_8,t_9\}$	$\{t_4,2{\cdot}t_8\}$	Yes
		$M_{14}=[7\,1\,0\,0\,0\,0\,6\,2\,0\,0]\ [t_4\rangle$	$p_2{+}2{\cdot}p_8{+}q_4{\le}5$	5	$\{t_2,2{\cdot}t_4,2{\cdot}t_9\}$	$\{t_1,t_4,2{\cdot}t_8\}$	Yes
5	236	$M_{99}=[6\,0\,0\,1\,1\,0\,7\,0\,1\,0]\ [t_8\rangle$	$p_4{+}p_5{+}p_9{+}q_8{\le}3$	3	$\{t_6,t_8,t_{10}\}$	$\{t_4,t_8,t_9\}$	Yes
		$M_{100}=[6\,1\,0\,0\,1\,0\,6\,1\,1\,0]\ [t_4\rangle$	$p_2{+}p_5{+}p_8{+}p_9{+}q_4{\le}4$	4	$\{t_2,2{\cdot}t_4,t_6,t_9,t_{10}\}$	$\{t_1,t_4,t_5,t_8,t_9\}$	Yes
6	220						

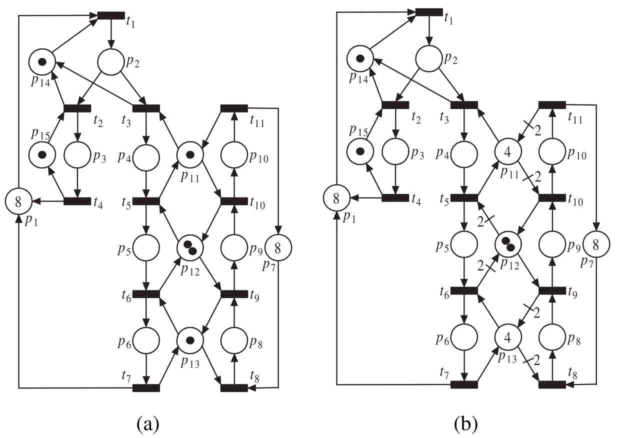

FIGURE 4. Two example PNs. (a) An S^3PR. (b) An S^4R.

Fig. 4 shows two PNs, among which the former is an S^3PR net while the latter an S^4R net. It is known that each stage in S^3PR requires only one unit of single-type resource while there is no restriction on the resource usage in type and quantity in S^4R. They are both typical classes of PNs in AMSs.

In Fig. 4(a), there are 188 states including 168 good states and 20 bad states. In Table 6, we can find that these four methods obtain four different supervisors while have the same control results. Also, they enforce the liveness of this PN with maximal permissiveness. The state-based methods derive the least number of monitors compared with the event-based ones. Furthermore, the supervisor of the state-based method has less arcs compared with the weighted state-based one, thus being the simplest supervisor structure. Therefore, in this example, the state-based control methods are recommended. This also verifies that the state-based methods are more suitable for ordinary resource allocation system. Their monitors are presented in Table 6.

In Fig. 4(b), the S^4R is prone to deadlocks with 936 good states and 152 bad states. The control results of four methods are shown in Table 7. We can find that the event-based and weighted event-based methods have the same control results. The weighted event-based method has one more monitor compared with the event-based one. Furthermore, these two methods have the least number of iterations, while preserve the least number of states ($802/936 = 85.68\%$). The state-based control enforces the net to be live with 829 states ($829/936 = 88.57\%$). While the weighted state-based control can preserves 885 states ($885/936 = 94.55\%$). Both of them have three iterations, leading to three monitors. This example shows that the weighted state-based method is priority to other ones in permissiveness.

C. COMPARISONS AND COMPLEXITY ANALYSIS

First, we discuss the permissiveness of the proposed method. From analysis above, we can find that the proposed method in Algorithm 3 is not maximally permissive in all the examples. This is because some good markings may be unintentionally removed by an inequality $\xi_i^T \cdot \Upsilon \le b_i$. If none of the good markings violates $\Xi^T \cdot \Upsilon \le B$, the maximal permissiveness is achieved. Nevertheless, there are also some merits. First, it synthesizes the maximally or almost-maximally permissive supervisors for some structurally simple classes of PNs, for example, S^3PR PNs. Second, such kind of supervisor, as a conjunctive set of linear constraints, can preserve more good

TABLE 6. Control results of four methods for the PN in Fig. 4(a).

Type	i	#Sta.	FBM (*CGM $[t_i\rangle$)	Inequalities	M_0	${}^\bullet p_c$	$p_c^\bullet$	Indep.
I	1	188	M_{62} = [7 0 0 1 0 0 6 0 2 0]	$p_4+p_9\leq 2$	2	$\{t_5,t_{10}\}$	$\{t_3,t_9\}$	Yes
	2	180	M_{79} = [6 0 0 0 2 0 7 1 0 0]	$p_5+p_8\leq 2$	2	$\{t_6,t_9\}$	$\{t_5,t_8\}$	Yes
	3	172	M_{103} = [6 0 0 1 1 0 6 1 1 0]	$p_4+p_5+p_8+p_9\leq 3$	3	$\{t_6,t_{10}\}$	$\{t_3,t_8\}$	Yes
	4	168						
II	1	188	$^*M_{39}$ = [7 0 0 1 0 0 6 1 1 0] $[t_9\rangle$	$p_4+p_8+p_9+q_9\leq 3$	3	$\{t_5,t_9,t_{10}\}$	$\{t_3,t_8,t_9\}$	Yes
			$^*M_{40}$ = [7 1 0 0 0 0 6 0 2 0] $[t_3\rangle$	$p_2+p_9+q_3\leq 3$	3	$\{t_2,2{\cdot}t_3,t_{10}\}$	$\{t_1,t_3,t_9\}$	Yes
	2	180	$^*M_{54}$ = [6 0 0 0 2 0 8 0 0 0] $[t_8\rangle$	$p_5+q_8\leq 2$	2	$\{t_6,t_8\}$	$\{t_5,t_8\}$	Yes
			$^*M_{56}$ = [6 0 0 1 1 0 7 1 0 0] $[t_5\rangle$	$p_4+p_5+p_8+q_5\leq 3$	3	$\{t_5,t_6,t_9\}$	$\{t_3,t_5,t_8\}$	Yes
	3	172	$^*M_{79}$ = [6 0 0 1 1 0 7 0 1 0] $[t_8\rangle$	$p_4+p_5+p_9+q_8\leq 3$	3	$\{t_6,t_8,t_{10}\}$	$\{t_3,t_8,t_9\}$	Yes
			$^*M_{81}$ = [6 1 0 0 1 0 6 1 1 0] $[t_3\rangle$	$p_2+p_5+p_8+p_9+q_3\leq 4$	4	$\{t_2,2{\cdot}t_3,t_6,t_{10}\}$	$\{t_1,t_3,t_5,t_8\}$	Yes
	4	168						
III	1	188	M_{62} = [7 0 0 1 0 0 6 0 2 0]	$p_4+2{\cdot}p_9\leq 4$	4	$\{t_5,2{\cdot}t_{10}\}$	$\{t_3,2{\cdot}t_9\}$	Yes
	2	180	M_{79} = [6 0 0 0 2 0 7 1 0 0]	$2{\cdot}p_5+p_8\leq 4$	4	$\{2{\cdot}t_6,t_9\}$	$\{2{\cdot}t_5,t_8\}$	Yes
	3	172	M_{103} = [6 0 0 1 1 0 6 1 1 0]	$p_4+p_5+p_8+p_9\leq 3$	3	$\{t_6,t_{10}\}$	$\{t_3,t_8\}$	Yes
	4	168						
IV	1	188	$^*M_{39}$ = [7 0 0 1 0 0 6 1 1 0] $[t_9\rangle$	$p_4+p_8+p_9+q_9\leq 3$	3	$\{t_5,t_9,t_{10}\}$	$\{t_3,t_8,t_9\}$	Yes
			$^*M_{40}$ = [7 1 0 0 0 0 6 0 2 0] $[t_3\rangle$	$p_2+2{\cdot}p_9+q_3\leq 5$	5	$\{t_2,2{\cdot}t_3,2{\cdot}t_{10}\}$	$\{t_1,t_3,2{\cdot}t_9\}$	Yes
	2	180	$^*M_{54}$ = [6 0 0 0 2 0 8 0 0 0] $[t_8\rangle$	$2{\cdot}p_5+q_8\leq 4$	4	$\{2{\cdot}t_6,t_8\}$	$\{2{\cdot}t_5,t_8\}$	Yes
			$^*M_{56}$ = [6 0 0 1 1 0 7 1 0 0] $[t_5\rangle$	$p_4+p_5+p_8+q_5\leq 3$	3	$\{t_5,t_6,t_9\}$	$\{t_3,t_5,t_8\}$	Yes
	3	172	$^*M_{79}$ = [6 0 0 1 1 0 7 0 1 0] $[t_8\rangle$	$p_4+p_5+p_9+q_8\leq 3$	3	$\{t_6,t_8,t_{10}\}$	$\{t_3,t_8,t_9\}$	Yes
			$^*M_{81}$ = [6 1 0 0 1 0 6 1 1 0] $[t_3\rangle$	$p_2+p_5+p_8+p_9+q_3\leq 4$	4	$\{t_2,2{\cdot}t_3,t_6,t_{10}\}$	$\{t_1,t_3,t_5,t_8\}$	Yes
	4	168						

TABLE 7. Control results of four methods for the PN in Fig. 4(b).

Type	i	#Sta.	FBM (*CGM $[t_i\rangle$)	Inequalities	M_0	${}^\bullet p_c$	$p_c^\bullet$	Indep.
I	1	1088	M_{49} = [7 0 0 0 1 0 6 2 0 0]	$p_5+p_8\leq 2$	2	$\{t_6,t_9\}$	$\{t_5,t_8\}$	Yes
	2	1056	M_{160} = [5 0 0 3 0 0 6 2 0 0]	$p_4+p_8\leq 4$	4	$\{t_5,t_9\}$	$\{t_3,t_8\}$	Yes
	3	985	M_{160} = [5 0 0 3 0 0 7 0 1 0]	$p_4+p_9\leq 3$	3	$\{t_5,t_{10}\}$	$\{t_3,t_9\}$	Yes
	4	829						
II	1	1088	$^*M_{26}$ = [7 0 0 0 1 0 7 1 0 0] $[t_8\rangle$	$p_5+p_8+q_8\leq 2$	2	$\{t_6,t_8,t_9\}$	$\{t_5,2{\cdot}t_8\}$	Yes
			$^*M_{27}$ = [7 1 0 0 0 0 6 0 2 0] $[t_5\rangle$	$p_4+p_8+q_5\leq 3$	3	$\{2{\cdot}t_5,t_9\}$	$\{t_3,t_5,t_8\}$	Yes
	2	861	$^*M_{69}$ = [5 1 0 2 0 0 8 0 0 0] $[t_3\rangle$	$p_2+p_4+q_3\leq 3$	3	$\{t_2,t_3,t_5\}$	$\{t_1,t_3\}$	Yes
	3	802						
III	1	1088	M_{49} = [7 0 0 0 1 0 6 2 0 0]	$p_5+2{\cdot}p_8\leq 4$	4	$\{t_6,2{\cdot}t_9\}$	$\{t_5,2{\cdot}t_8\}$	Yes
	2	1056	M_{160} = [5 0 0 3 0 0 6 2 0 0]	$3{\cdot}p_4+2{\cdot}p_8\leq 12$	12	$\{3{\cdot}t_5,2{\cdot}t_9\}$	$\{3{\cdot}t_5,2{\cdot}t_8\}$	Yes
	3	985	M_{160} = [5 0 0 3 0 0 7 0 1 0]	$3{\cdot}p_4+p_9\leq 9$	9	$\{3{\cdot}t_5,t_{10}\}$	$\{3{\cdot}t_3,t_9\}$	Yes
	4	885						
IV	1	1088	$^*M_{26}$ = [7 0 0 0 1 0 7 1 0 0] $[t_8\rangle$	$p_5+p_8+q_8\leq 2$	2	$\{t_6,t_8,t_9\}$	$\{t_5,2{\cdot}t_8\}$	Yes
			$^*M_{27}$ = [7 1 0 0 0 0 6 0 2 0] $[t_5\rangle$	$p_4+2{\cdot}p_8+q_5\leq 5$	5	$\{2{\cdot}t_5,2{\cdot}t_9\}$	$\{t_3,t_5,2{\cdot}t_8\}$	Yes
	2	961	$^*M_{69}$ = [5 0 0 3 0 0 8 0 0 0] $[t_8\rangle$	$3{\cdot}p_4+q_8\leq 9$	9	$\{3{\cdot}t_5,t_8\}$	$\{3{\cdot}t_3,t_8\}$	Yes
			$^*M_{71}$ = [5 1 0 2 0 0 8 0 0 0] $[t_3\rangle$	$p_2+2{\cdot}p_4+q_3\leq 5$	5	$\{t_2,2{\cdot}t_5\}$	$\{t_1,t_3\}$	Yes
	3	802						

markings as much as possible. Especially, when introducing weights in the inequality expression, it further improves the permissiveness in some extent.

Specifically, according to our illustrative examples, we derive four qualitative conclusions on the comparison of their permissiveness. First, there is no theoretical and practical evidence that shows which has a higher permissiveness between the event-based and state-based methods. For illustration, in Fig. 1, event-based method has the better result; in Fig. 4(a), they have the same results; while in Fig. 4(b), the state-based method has a better result. Second, the weighted state-based and event-based methods can enhance permissiveness compared with their non-weighted methods respectively. At least, they will not be worse than the non-weighted methods. In the three examples, we can see that the event-based and weighted event-based methods have the same results while the weighted state-based method has better results compared with the state-based one. Third, if we compare the performance and structural simplicity of the four methods, the weighted state-based one is recommendable. From our examples, the supervisors of weighted state-based control usually possess the least number of control places while preserve the most number of good states. Four, the weighted state-based method has an obvious advantage in improving permissiveness for the systems whose processes can accommodate more tokens. In such systems, there are more resources to support raw parts being manufactured in their processes. This situation can increase the disparity among the weight coefficients. Compared with the PN in Fig. 4(a), the PN in Figs. 1 and 4(b) can accommodate more tokens in their processes. Thus, the weighted state-based method has the better control results in the latter examples.

Second, we discuss the computation complexity of the unified algorithm. As we know, it consists of two stages. The first one is to synthesize a supervisor. No matter for any type of linear constraint, it is based on the reachability graph analysis, which suffers from the state explosion problem. Thus, the supervisor synthesis is of exponential complexity. The second one is to simplify the supervisor. The simplification theories in Theorems 4 and 5 imply that it can be transformed into linear programming, whose computational complexity is polynomial with regard to the number of inequalities. Thus, Algorithm 3 is of exponential complexity. However, admittedly, since there is no integer programming or exploration techniques for the reachability graph, the proposed method is easy to use and straightforward in application.

VI. CONCLUSION

This paper proposes a class of PNs which is featured with time-varying specifications. Time-varying specifications and liveness-enforcing supervisory control can be enforced independently. The former is realized by designing monitors which time-enable transitions at stipulated time intervals, while the latter is implemented by invariant-based methods. For the invariant-based methods, we present a comprehensive and comparative study on the state-based, event-based, weighted state-based, and weighted event-based methods, and provide a unified and combined algorithm to derive the simplified liveness-enforcing supervisors based on place invariants. It concludes that the state-based methods which solve the forbidden state problem produce more compact supervisors while the event-based ones which solve the event/state separation problem generate more concise supervisors, in general case. The weighted inequalities are proposed to improve the specifications' expressivity and enhance system permissiveness. It shows that the weighted state-based method has an obvious advantage in improving permissiveness for the systems whose processes can accommodate more tokens. Thus, the future work should extend the current results to more complex systems, especially those whose processes accommodate more tokens or where there are parallel processes, and find supervisor simplification technique which can reduce the size of monitors by changing some parameters.

REFERENCES

[1] L. Bai, N. Wu, Z. Li, and M. Zhou, "Optimal one-wafer cyclic scheduling and buffer space configuration for single-arm multicluster tools with linear topology," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 46, no. 10, pp. 1456–1467, Oct. 2016.

[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] F. Basile, R. Cordone, and L. Piroddi, "Integrated design of optimal supervisors for the enforcement of static and behavioral specifications in Petri net models," *Automatica*, vol. 49, no. 11, pp. 3432–3439, Nov. 2013.

[4] 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.

[5] F. Basile, R. Cordone, and L. Piroddi, "A branch and bound approach for the design of decentralized supervisors in Petri net models," *Automatica*, vol. 52, pp. 322–333, Feb. 2015.

[6] M. P. Cabasino, A. Giua, C. N. Hadjicostis, and C. Seatzu, "Fault model identification and synthesis in Petri nets," *Discrete Event Dyn. Syst.*, vol. 25, no. 3, pp. 419–440, 2015.

[7] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*. New York, NY, USA: Springer, 2008.

[8] C. Chen and H. Hu, "Time-varying automated manufacturing systems and their event-based control: A Petri net approach," in *Proc. IEEE Int. Conf. Robot. Automat.*, Seattle, WA, USA, May 2015, pp. 3833–3840.

[9] C. Chen and H. Hu, "Liveness-enforcing supervision in AMS-oriented HAMGs: An approach based on new characterization of siphons using Petri nets," *IEEE Trans. Autom. Control*, vol. AC-63, no. 7, pp. 1987–2002, Jul. 2018.

[10] H. Chen, N. Wu, Z. Li, and T. Qu, "On a maximally permissive deadlock prevention policy for automated manufacturing systems by using resource-oriented Petri nets," *ISA Trans.*, to be published. doi: 10.1016/j.isatra.2018.11.025.

[11] Y. Chen and Z. Li, "Design of a maximally permissive liveness-enforcing supervisor with a compressed supervisory structure for flexible manufacturing systems," *Automatica*, vol. 47, no. 5, pp. 1028–1034, May 2011.

[12] Y. Chen, Z. Li, K. Barkaoui, N. Wu, and M. Zhou, "Compact supervisory control of discrete event systems by Petri nets with data inhibitor arcs," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 2, pp. 364–379, Feb. 2017.

[13] X. Cong, Y. Chen, Z. Li, N. Wu, E. A. Nasr, and A. M. El-Tamim, "Optimal Petri net supervisors of discrete event systems via weighted and data inhibitor arcs," *IEEE Access*, vol. 6, no. 1, pp. 8245–8257, 2018.

[14] X. Cong *et al.*, "Design of optimal Petri net supervisors for flexible manufacturing systems via weighted inhibitor arcs," *Asian J. Control*, vol. 20, no. 1, pp. 511–530, Jan. 2018.

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

[16] A. Ghaffari, N. Rezg, and X. 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] A. Ghaffari, N. Rezg, and X. Xie, "Feedback control logic for forbidden-state problems of marked graphs: Application to a real manufacturing system," *IEEE Trans. Autom. Control*, vol. 48, no. 1, pp. 18–29, Jan. 2003.

[18] A. Giua, F. DiCesare, and M. Silva, "Generalized mutual exclusion contraints on nets with uncontrollable transitions," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Chicago, IL, USA, Oct. 1992, pp. 974–979.

[19] Z. He, Z. Li, and A. Giua, "Optimization of deterministic timed weighted marked graphs," *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 2, pp. 1084–1095, Feb. 2017.

[20] H. Hu and Y. Liu, "Supervisor simplification for AMS based on Petri nets and inequality analysis," *IEEE Trans. Autom. Sci. Eng.*, vol. 11, no. 1, pp. 66–77, Jan. 2014.

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

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

[24] M. Iordache and P. Antsaklis, *Supervisory Control of Concurrent Systems: A Petri Net Structural Approach*. Boston, MA, USA: Kluwer, 2006.

[25] F. Liu and H. Lin, "Reliable supervisory control for general architecture of decentralized discrete event systems," *Automatica*, vol. 46, no. 9, pp. 1510–1516, Sep. 2010.

[26] M. Liu, Z. He, N. Wu, A. Al-Ahmari, and Z. Li, "Resource configuration analysis for a class of Petri nets based on strongly connected characteristic resource subnets," *IEEE Access*, vol. 5, pp. 26376–26386, 2017.

[27] 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.

[28] C. Li, Y. Chen, Z. Li, and K. Barkaoui, "Synthesis of liveness-enforcing Petri net supervisors based on a think-globally-act-locally approach and vector covering for flexible manufacturing systems," *IEEE Access*, vol. 5, pp. 16349–16358, 2017.
[29] G. Liu, P. Li, N. Wu, and L. Yin, "Two-step approach to robust deadlock control in automated manufacturing systems with multiple resource failures," *J. Chin. Inst. Eng.*, vol. 41, no. 4, pp. 484–494, Oct. 2018.
[30] G. Liu, P. Li, Z. Li, and N. Wu, "Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs," *IEEE Trans. Syst., Man, Cybern. Syst.*, to be published. doi: 10.1109/TSMC.2018.2815618.
[31] J. Luo and K. Nonami, "Approach for transforming linear constraints on Petri nets," *IEEE Trans. Autom. Control*, vol. 56, no. 12, pp. 2751–2765, Dec. 2011.
[32] J. Luo, H. Ni, W. Wu, S. Wang, and M. Zhou, "Simultaneous reduction of Petri nets and linear constraints for efficient supervisor synthesis," *IEEE Trans. Autom. Control*, vol. 60, no. 1, pp. 88–103, Jan. 2015.
[33] J. Luo and M. Zhou, "Petri-net controller synthesis for partially controllable and observable discrete event systems," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1301–1313, Mar. 2017.
[34] Z. Ma, Z. Li, and A. Giua, "Characterization of admissible marking sets in Petri nets with conflicts and synchronizations," *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, Mar. 2017.
[35] Z. Ma, Z. Li, and A. Giua, "Design of optimal Petri net controllers for disjunctive generalized mutual exclusion constraints," *IEEE Trans. Autom. Control*, vol. 60, no. 7, pp. 1774–1785, Jul. 2015.
[36] 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.
[37] Y. Qiang, G. Tian, Y. Liu, and Z. Li, "Energy-efficiency models of sustainable urban transportation structure optimization," *IEEE Access*, vol. 6, pp. 18192–18199, 2018.
[38] Y. Qiao, N. Wu, F. Yang, M. Zhou, and Q. Zhu, "Wafer sojourn time fluctuation analysis of time-constrained dual-arm cluster tools with wafer revisiting and activity time variation," *IEEE Trans. Syst., Man, Cybern. Syst.*, vol. 48, no. 4, pp. 622–636, Apr. 2018.
[39] E. Salimi and R. S. Sreenivas, "On invariant-based monitors that enforce liveness in a class of partially controlled general Petri nets," *IEEE Trans. Autom. Control*, vol. 60, no. 10, pp. 2825–2830, Oct. 2015.
[40] R. S. Sreenivas, "On the existence of supervisory policies that enforce liveness in partially controlled free-choice Petri nets," *IEEE Trans. Autom. Control*, vol. 57, no. 2, pp. 435–449, Feb. 2012.
[41] 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. IEEE Int. Conf. Robot. Automat.*, Barcelona, Spain, Apr. 2005, pp. 272–278.
[42] M. Uzam and M. Zhou, "An iterative synthesis approach to Petri net-based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 37, no. 3, pp. 362–371, May 2007.
[43] N. Viswanadham, Y. Narahari, and T. L. Johnson, "Deadlock prevention and deadlock avoidance in flexible manufacturing systems using Petri net models," *IEEE Trans. Robot. Autom.*, vol. 6, no. 6, pp. 713–723, Dec. 1990.
[44] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S^4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.
[45] N. Wu, M. Zhou, and G. Hu, "One-step look-ahead maximally permissive deadlock control of AMS by using Petri net," *ACM Trans. Embedded Comput. Syst.*, vol. 12, no. 1, pp. 1–23, Jan. 2013.
[46] N. Q. Wu and M. C. Zhou, "Schedulability analysis and optimal scheduling of dual-arm cluster tools with residency time constraint and activity time variation," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 1, pp. 203–209, Jan. 2012.
[47] N. Q. Wu and M. C. Zhou, "Modeling, analysis and control of dual-arm cluster tools with residency time constraint and activity time variation based on Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 9, no. 2, pp. 446–454, Apr. 2012.
[48] N. Q. Wu, F. Chu, C. Chu, and M. C. Zhou, "Petri net modeling and cycle-time analysis of dual-arm cluster tools with wafer revisiting," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 43, no. 1, pp. 196–207, Jan. 2013.
[49] N. Wu, M. Zhou, and Z. Li, "Short-term scheduling of crude-oil operations: Enhancement of crude-oil operations scheduling using a Petri net-based control-theoretic approach," *IEEE Robot. Autom. Mag*, vol. 22, no. 2, pp. 64–76, Jun. 2015.
[50] N. Wu, M. Zhu, L. Bai, and Z. Li, "Short-term scheduling of crude oil operations in refinery with high-fusion-point oil and two transportation pipelines," *Enterprise Inf. Syst.*, vol. 10, no. 6, pp. 581–610, May 2016.
[51] D. Xiang, G. Liu, C. Yan, and C. Jiang, "Detecting data-flow errors based on Petri nets with data operations," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 251–260, Jan. 2018.
[52] K. Yamalidou, J. Moody, M. Lemmon, and P. Antsaklis, "Feedback control of Petri nets based on place invariants," *Automatica*, vol. 32, pp. 15–18, Jan. 1996.
[53] F. Yang, N. Q. Wu, Y. Qiao, and R. Su, "Polynomial approach to optimal one-wafer cyclic scheduling of treelike hybrid multi-cluster tools via Petri nets," *IEEE/CAA J. Autom. Sinica*, vol. 5, no. 1, pp. 270–280, Jan. 2018.
[54] F. Yang, N. Wu, Y. Qiao, M. Zhou, and Z. Li, "Scheduling of single-arm cluster tools for an atomic layer deposition process with residency time constraints," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 3, pp. 502–516, Mar. 2017.
[55] J. Ye, Z. Li, and A. Giua, "Decentralized supervision of Petri nets with a coordinator," *IEEE Trans. Syst., Man, Cybernetics, Syst.*, vol. 45, no. 6, pp. 955–966, Jun. 2015.
[56] D. You, S. Wang, W. Dai, W. Wu, and Y. Jia, "An approach for enumerating minimal siphons in a subclass of Petri nets," *IEEE Access*, vol. 6, pp. 4255–4265, 2018.
[57] J. Zhang, G. Frey, A. Al-Ahmari, T. Qu, N. Wu, and Z. Li, "Analysis and control of dynamic reconfiguration processes of manufacturing systems," *IEEE Access*, vol. 6, pp. 28028–28040, 2018.
[58] J. Zhang, M. Khalgui, Z. Li, G. Frey, O. Mosbahi, and H. Ben Salah, "Reconfigurable coordination of distributed discrete event control systems," *IEEE Trans. Control Syst. Technol.*, vol. 23, no. 1, pp. 323–330, Jan. 2015.
[59] S. Zhang, N. Wu, Z. Li, T. Qu, and C. Li, "Petri net-based approach to short-term scheduling of crude oil operations with less tank requirement," *Inf. Sci.*, vol. 417, pp. 247–261, Nov. 2017.
[60] Q. Zhu, M. Zhou, Y. Qiao, and N. Wu, "Petri net modeling and scheduling of a close-down process for time-constrained single-arm cluster tools," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 48, no. 3, pp. 389–400, Mar. 2018.

CHEN CHEN received the B.S. degree in automation from Zhoukou Normal University, Zhoukou, China, in 2012. She is currently pursuing the Ph.D. degree in control theory and control engineering with Xidian University. Her research interests include discrete event systems and their supervisory control techniques, Petri nets, automated manufacturing systems, and supervisor simplification.

HESUAN HU (M'11–SM'12) received the B.S. degree in computer engineering and the M.S. and Ph.D. degrees in electro-mechanical engineering from Xidian University, Xi'an, China, in 2003, 2005, and 2010, respectively, where he is currently a Full Professor. He is also with the School of Computer Science and Engineering, College of Engineering, Nanyang Technological University, Singapore, and the State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an. His research interests include discrete event systems and their supervisory control techniques, Petri nets, automated manufacturing systems, multimedia streaming systems, and artificial intelligence. In the aforementioned areas, he has over 120 publications in journals, book chapters, and conference proceedings. He was a recipient of many national and international awards, including the Franklin V. Taylor Outstanding Paper Award from the IEEE SMC Society, in 2010, and the finalists of the Best Automation Paper from the IEEE RAS Society, in 2013, 2016, and 2017. He is an Associate Editor of the *IEEE Control Systems Magazine*, the *IEEE Robotics and Automation Magazine*, the IEEE Transactions on Automation Science and Engineering, and the *Journal of Intelligent Manufacturing*. He was on the Editorial Board of over 10 international journals.