Proceedings of the 2016 Industrial and Systems Engineering Research Conference
H. Yang, Z. Kong, and MD Sarder, eds.

On the Algebraic Structure of Solutions to Some Livelock-Avoidance Problems in Manufacturing Systems Modeled by Petri Nets

Rakesh Nagi
Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

R.S. Sreenivas
Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign

Ehsan Salimi
Daniel J. Epstein Department of Industrial and Systems Engineering
University of Southern California

Abstract

If at least one of the activities of an automated manufacturing system enters a state of suspended-animation for perpetuity, we say the system is experiencing *livelock*. We consider *Petri Net* (PN) models of manufacturing systems that are prone to livelocks, which can be made livelock-free by a *supervisory policy*. Specifically, we consider supervisory policies that are represented by structural additions to the original PN, known as *monitors*. The role of the monitor parallels that of a traffic-cop; it regulates the activities of a system in a manner that guarantees the supervised system is livelock-free. In this paper we relate the qualitative property of livelock-freedom and the quantitative property of maximizing throughput in a class of PN models. It identifies a necessary and sufficient condition for class of systems where the objective of achieving livelock-freedom in a minimally-restrictive manner, also accomplishes the objective of maximizing its throughput.

Keywords
Manufacturing Systems. Livelock-Freedom. Throughput-Maximization. Supervisory Control. Monitors.

1. Introduction

A system is *livelocked* if there is at least one activity that has entered into a state of suspended animation for perpetuity. In contrast, a *deadlock* occurs when *none* of the activities of the system can proceed to completion. For example, in a flexible manufacturing system many operations are running at the same time: conveyors, robots and machines. The limiteded amout of available resources, brings operation into a fierce competition causing operations waiting for resources while others using them. In extrem cases, this cause system to move into deadlock situation, halting the manufacturing process [15]. A livelock-free system cannot experience deadlocks. However, a deadlock-free system can experience livelocks. A livelock-prone system can be made livelock-free through the process of *supervision*. A *supervisory policy* regulates the activities of a system in such a manner that the resulting supervised-system is livelock-free. The desiderata for the supervisory policy is that it should be *minimally restrictive*; that is, it stops the occurrence of "undesirable" activities *only* when it is absolutely necessary.

In the general setting, the *qualitative* objective of livelock-freedom is not equivalent to the *quantitative* objective of increasing a system's throughput. That said, there are special cases where the minimally restrictive policy for livelock-freedom also maximizes the system throughput. This paper identifies one such a family of instances using the algebraic characterization of the minimally restrictive livelock avoidance policy. Our discourse centers around *Petri net* (PN) models of manufacturing systems [1, 2], and *liveness enforcing supervisory policies* (LESPs) for these PN models [3–6].

The rest of the paper is organized as follows. Section 2 introduces the notation and definitions used in this paper. This section also introduces the paradigm of Supervisory Control, and performance models that use Generalized Semi-Markov Processes (GSMPs). The main result is presented via an illustrative example in Section 3.

2. Notation and Definition

We use $\mathcal{N}$ ($\mathcal{N}^+$) to denote the set of non-negative (positive) integers. The term $card(\bullet)$ denotes the cardinality of a set. A *Petri net structure* $N = (\Pi, T, \Theta, \Gamma)$ is an ordered 4-tuple, where $\Pi = \{p_1, \ldots, p_n\}$ is a set of *n places*, $T = \{t_1, \ldots, t_m\}$ is a collection of *m transitions*, $\Theta \subseteq (\Pi \times T) \cup (T \times \Pi)$ is a set of *arcs*, and $\Gamma : \Theta \rightarrow \mathcal{N}^+$ is the *weight* associated with each arc. In a manufacturing system, places might signify a machine buffer while transitions signify operations. Places might also signify a certain conditions. A place might have one or more "tokens" that correspondingly mean the number of parts in the machine buffer or presence of the condition. Note that a PN has a bipartite structure in that the arcs are only from places to transitions or transitions to places (and not from transitions to transitions or places to places). A PN structure is said to be *ordinary* (*general*) if the weight associated with an arc is (not necessarily) unitary. The weights signify the quantity per items required for an operation if the arc is from a place to a transition, and quantity produced per operation when the arc is from a transition to a place. When a transition "fires" a number of tokens equal to the weight of each input arc are removed from the corresponding input places and a number of tokens equal to the weight of each output arc are placed in the corresponding output places. A transition cannot fire if the requisite number of tokens are not present in the input place(s) (see later for formal definition of *enabled*). The *initial marking function* (or the *initial marking*) of a PN structure N is a function $\mathbf{m}^0 : \Pi \rightarrow \mathcal{N}$, which identifies the number of *tokens* in each place. We will use the term *Petri net* (PN) and the symbol $N(\mathbf{m}^0)$ to denote a PN structure N along with its initial marking $\mathbf{m}^0$. The marking signifies the state of the manufacturing system being modeled by the PN.

A marking $\mathbf{m} : \Pi \rightarrow \mathcal{N}$ is sometimes represented by an integer-valued vector $\mathbf{m} \in \mathcal{N}^n$, where the i-th component $\mathbf{m}_i$ represents the token load ($\mathbf{m}(p_i)$) of the i-th place. In a graphical view, the weight of an arc is represented by an integer that is placed along side the arc. For brevity, we refrain from denoting the weight of those arcs $\theta \in \Theta$ where $\Gamma(\theta) = 1$.

For a string of transitions $\sigma \in T^*$, we use $\mathbf{x}(\sigma)$ to denote the *Parikh vector* of σ. That is, the i-th entry, $\mathbf{x}_i(\sigma)$, corresponds to the number of occurrences of transition t_i in σ. σ signifies the sequence of operations performed (tranistions fired) in a manufacturing systems (PN model).

We define the sets ${}^\bullet x := \{y \mid (y,x) \in \Theta\}$ and $x^\bullet := \{y \mid (x,y) \in \Theta\}$. A transition $t \in T$ is said to be *enabled* at a marking $\mathbf{m}^i$ if $\forall p \in {}^\bullet t, \mathbf{m}^i(p) \geq \Gamma((p,t))$. The set of enabled transitions at marking $\mathbf{m}^i$ is denoted by the symbol $T_e(N, \mathbf{m}^i)$. An enabled transition $t \in T_e(N, \mathbf{m}^i)$ can *fire*, which changes the marking $\mathbf{m}^i$ to $\mathbf{m}^{i+1}$ according to $\mathbf{m}^{i+1}(p) = \mathbf{m}^i(p) - \Gamma(p,t) + \Gamma(t,p)$.

In those contexts where the marking is interpreted as a nonnegative integer-valued vector, it is useful to define the *input matrix* $\mathbf{IN}$ and *output matrix* $\mathbf{OUT}$ as two $n \times m$ matrices, whose (i,j)-th entry is defined as follows: $\mathbf{IN}_{i,j} = \Gamma((p_i, t_j))$ if $p_i \in {}^\bullet t_j$, and $\mathbf{IN}_{i,j}$ is zero otherwise; likewise, $\mathbf{OUT}_{i,j} = \Gamma((t_j, p_i))$ if $p_i \in t_j^\bullet$, and is zero otherwise. The *incidence matrix* $\mathbf{C}$ of the PN N is an $n \times m$ matrix, where $\mathbf{C} = \mathbf{OUT} - \mathbf{IN}$. A marking $\mathbf{m}^i$ is called *potentially reachable* from $\mathbf{m}^0$ if $\exists \mathbf{y} \in \mathcal{N}^m$ such that the equation $\mathbf{Cy} = (\mathbf{m}^i - \mathbf{m}^0)$ is satisfied. While every reachable marking is also potentially reachable, there can be potentially reachable markings that cannot be reachable.

An integral vector $\mathbf{y}$ where $\mathbf{y}^T\mathbf{C} = \mathbf{0}, \mathbf{y} \geq \mathbf{0}$ is said to be a *place-invariant* of the PN. A place invariant indicates that the number of tokens in all reachable markings satisfies some linear invariant. To be precise,if the firing of a transition $t \in T$ at a marking $\mathbf{m}^1$ resulted in a marking $\mathbf{m}^2$, then it follows that $\mathbf{y}^T\mathbf{m}^1 = \mathbf{y}^T\mathbf{m}^2$.

2.1 Supervisory Control of PNs

Petri Nets are regulated by a supervisory policy, which determines which event is to be permitted at each state, in such a manner that some behavioral specification is satisfied. Some of the transitions represent local activities that can be prevented, while others that are external to system cannot be prevented by the supervisor. As an illustration, transitions that represent failures cannot be prevented from ring, but transitions that represent admission of an entity into a system, can be prevented when the system is near full-capacity. Consequently, the paradigm of supervisory control assumes a subset of *controllable transitions*, denoted by $T_c \subseteq T$, which can be prevented from firing by an external agent called the *supervisor*. The set of *uncontrollable transitions*, denoted by $T_u \subseteq T$, is given by $T_u = T - T_c$. The controllable (resp. uncontrollable) transitions are represented as filled (resp. unfilled) boxes in graphical representation of PNs.

A *supervisory policy* $\mathcal{P} : \mathcal{N}^n \times T \rightarrow \{0,1\}$, is a function that returns a 0 or 1 for each transition and each reachable marking. The supervisory policy $\mathcal{P}$ permits the firing of transition t_j at marking $\mathbf{m}^i$, only if $\mathcal{P}(\mathbf{m}^i, t_j) = 1$. We use the symbol $T_{\mathcal{P}}(N, \mathbf{m}^i)$ to denote the set of transitions permitted by $\mathcal{P}$ in $N(\mathbf{m}^i)$. That is, $T_{\mathcal{P}}(N, \mathbf{m}^i) := \{t_j \in T \mid \mathcal{P}(\mathbf{m}^i, t_j) = 1\}$.

If $t_j \in T_e(N, \mathbf{m}^i)$ for some marking $\mathbf{m}^i$, we say the transition t_j is *state-enabled* at $\mathbf{m}^i$. If $\mathcal{P}(\mathbf{m}^i, t_j) = 1$, we say the transition t_j is *control-enabled* at $\mathbf{m}^i$. A transition has to be state- and control-enabled before it can fire. The fact that uncontrollable transitions cannot be prevented from firing by the supervisory policy is captured by the requirement that $\forall \mathbf{m}^i \in \mathcal{N}^n, \mathcal{P}(\mathbf{m}^i, t_j) = 1$, if $t_j \in T_u$. This is implicitly assumed of any supervisory policy in this paper.

A string of transitions $\sigma = < t_1 t_2 \cdots t_k >$, where $t_j \in T \, (j \in \{1, 2, \ldots, k\})$ is said to be a *valid firing string* starting from the marking $\mathbf{m}^i$, if, (1) $t_1 \in T_e(N, \mathbf{m}^i), \mathcal{P}(\mathbf{m}^i, t_1) = 1$, and (2) for $j \in \{1, 2, \ldots, k-1\}$ the firing of the transition t_j produces a marking $\mathbf{m}^{i+j}$ and $t_{j+1} \in T_e(N, \mathbf{m}^{i+j})$ and $\mathcal{P}(\mathbf{m}^{i+j}, t_{j+1}) = 1$.

The set of reachable markings under the supervision of $\mathcal{P}$ in N from the initial marking $\mathbf{m}^0$ is denoted by $\Re(N, \mathbf{m}^0, \mathcal{P})$. If $\mathbf{m}^{i+k}$ results from the firing of $\sigma \in T^*$ starting from the initial marking $\mathbf{m}^i$, we represent it symbolically as $\mathbf{m}^i \stackrel{\sigma}{\rightarrow} \mathbf{m}^{i+k}$.

A transition t_k is *live* under the supervision of $\mathcal{P}$ if

$$\forall \mathbf{m}^i \in \Re(N, \mathbf{m}^0, \mathcal{P}), \exists \mathbf{m}^j \in \Re(N, \mathbf{m}^i, \mathcal{P}) \text{ such that } t_k \in T_e(N, \mathbf{m}^j) \text{ and } \mathcal{P}(\mathbf{m}^j, t_k) = 1.$$

A policy $\mathcal{P}$ is a *liveness enforcing supervisory policy* (LESP) for $N(\mathbf{m}^0)$ if all transitions in $N(\mathbf{m}^0)$ are live under $\mathcal{P}$. The policy $\mathcal{P}$ is said to be *minimally restrictive* if for every LESP $\widehat{\mathcal{P}} : \mathcal{N}^n \times T \rightarrow \{0,1\}$ for $N(\mathbf{m}^0)$, the following condition holds: $\forall \mathbf{m}^i \in \mathcal{N}^n, \forall t \in T, \mathcal{P}(\mathbf{m}^i, t) \geq \widehat{\mathcal{P}}(\mathbf{m}^i, t)$. If an LESP exists for a PN, then PN can have finitely many LESPs. On the other hand, if a PN has an LESP, then it has a unique minimally restrictive LESP (cf. theorem 6.1, [7]).

The *language*, $L(N, \mathbf{m}^0, \mathcal{P})$, generated by a PN $N(\mathbf{m}^0)$ under the a supervisory policy $\mathcal{P}$ is defined as $L(N, \mathbf{m}^0, \mathcal{P}) = \{\sigma \in T^* \mid \mathbf{m}^0 \stackrel{\sigma}{\rightarrow} \mathbf{m}^1 \text{ under } \mathcal{P} \text{ in } N(\mathbf{m}^0)\}$, which is the set of all firing strings that are valid in $N(\mathbf{m}^0)$ under the supervision of $\mathcal{P}$.

Following [8], we say a language $L(N, \mathbf{m}^0, \mathcal{P})$ is *fair* if $\forall \sigma, \widehat{\sigma} \in L(N, \mathbf{m}^0, \mathcal{P}), (\mathbf{x}(\sigma) \leq \mathbf{x}(\widehat{\sigma})) \Rightarrow (\sigma \trianglelefteq \widehat{\sigma})$, where

$$(\sigma \trianglelefteq \widehat{\sigma}) \Leftrightarrow \forall t_i \in T, \{(\sigma t_i \in L(N, \mathbf{m}^0, \mathcal{P})) \wedge (\mathbf{x}(\sigma)_i = \mathbf{x}(\widehat{\sigma})_i) \Rightarrow (\widehat{\sigma} t_i \in L(N, \mathbf{m}^0, \mathcal{P}))\}.$$

That is, a fair language is characterized by the property that for any two firing strings $\sigma, \widehat{\sigma}$ where the number of occurrences of a transition in σ is less than or equal to the number of occurrences of the same transition in $\widehat{\sigma}$ (i.e. $(\mathbf{x}(\sigma) \leq \mathbf{x}(\widehat{\sigma}))$), if transition t_i appears an equal number of times in σ and $\widehat{\sigma}$ (i.e. $(\mathbf{x}(\sigma)_i = \mathbf{x}(\widehat{\sigma})_i)$), and it is permitted to fire after σ (i.e. $(\sigma t_i \in L(N, \mathbf{m}^0, \mathcal{P}))$), then it is also permitted to fire after $\widehat{\sigma}$ (i.e. $(\widehat{\sigma} t_i \in L(N, \mathbf{m}^0, \mathcal{P}))$).

2.2 Quantitative Models: Stochastic PNs

Stochastic features can be incorporated into PNs using *Generalized Semi-Markov Process* (GSMP) processes [9]. We associate a point-process that describes the execution-time of each transition and the timed-version of the PN operates as follows. For each $t_i \in T$ at the n-th occurrence, we have a stack of firing-times $\{\omega_i(n)\}_{n=1}^{\infty}$ which is a collection of realizations from a predetermined probability distribution indexed to each transition in the PN.

At initialization, the *simulation-clock* (τ) is reset (i.e. $\tau = 0$); and, we schedule the firing of $t_j \in T_e(N, \mathbf{m}^0) \cap T_{\mathcal{P}}(N, \mathbf{m}^0)$ at time $\tau = \tau + \omega_j(1)$, where $j = argmin_i\{\omega_i(1), t_i \in T_e(N, \mathbf{m}^0) \cap T_{\mathcal{P}}(N, \mathbf{m}^0)\}$. Let $\mathbf{m}^0 \stackrel{t_j}{\rightarrow} \mathbf{m}^1$, then $\forall t_k \in (T_e(N, \mathbf{m}^0) \cap T_{\mathcal{P}}(N, \mathbf{m}^0) \cap T_e(N, \mathbf{m}^1) \cap T_{\mathcal{P}}(N, \mathbf{m}^1)) - \{t_j\}$, $\omega_k(1) \leftarrow (\omega_k(1) - \omega_j(1))$; $\{\omega_j(n)\}_{n=1}^{\infty} \leftarrow \{\omega_j(n)\}_{n=2}^{\infty}$. This process is repeated as often as necessary to generate a sample-path of transition-firings over the simulation-interval $[0, \tau]$. We use the symbol $\sigma(N, \mathbf{m}^0, \mathcal{P}, \tau, \{\{\omega_i(n)\}_{n=1}^{\infty}\}_{t_i \in T}\})$ to denote the valid firing-string in $N(\mathbf{m}^0)$ under the supervision of $\mathcal{P}$ that occurs over the interval $[0, \tau]$ under the above interpretation.

The following result, which follows directly from the proof of Theorem 1 of reference [8] establishes the fact that if we have two LESPs $\mathcal{P}_1$ and $\mathcal{P}_2$ for a PN $N(\mathbf{m}^0)$, that is driven by a common set of transition firing-times $\{\{\omega_i(n)\}_{n=1}^{\infty}\}_{t_i \in T}$, if $L(N, \mathbf{m}^0, \mathcal{P}_1) \subseteq L(N, \mathbf{m}^0, \mathcal{P}_2)$ and $L(N, \mathbf{m}^0, \mathcal{P}_2)$ is fair, then the PN $N(\mathbf{m}^0)$ supervised by $\mathcal{P}_2$ will run faster. That is, its throughput (measured in terms of the average number of transition-firings per unit-time) will be larger.

Theorem 2.1. *(Theorem 1, [8]) Let $\mathcal{P}_i : \mathcal{N}^n \times T \rightarrow \{0,1\}(i=1,2)$ be two LESPs for a PN $N(\mathbf{m}^0)$. If $L(N,\mathbf{m}^0,\mathcal{P}_1) \subseteq L(N,\mathbf{m}^0,\mathcal{P}_2)$, and $L(N,\mathbf{m}^0,\mathcal{P}_2)$ is fair, then*
$$\mathbf{x}(\sigma(N,\mathbf{m}^0,\mathcal{P}_1,\tau,\{\{\omega_i(n)\}_{n=1}^{\infty}\}_{t_i\in T}\})) \leq \mathbf{x}(\sigma(N,\mathbf{m}^0,\mathcal{P}_2,\tau,\{\{\omega_i(n)\}_{n=1}^{\infty}\}_{t_i\in T}\})).$$

Proof. Follows directly from the proof of Theorem 1 in [8]. □

A language $L(N,\mathbf{m}^0,\mathcal{P})$ is *non-preemptive* [8] (or, *non-interruptive* [10]) if $\mathbf{m}^0 \stackrel{\sigma}{\rightarrow} \mathbf{m}^1 \stackrel{t}{\rightarrow} \mathbf{m}^2$ under $\mathcal{P}$ at $N(\mathbf{m}^0)$, then

$$(T_e(N,\mathbf{m}^1) \cap T_{\mathcal{P}}(N,\mathbf{m}^1)) - \{t\} \subseteq T_e(N,\mathbf{m}^2) \cap T_{\mathcal{P}}(N,\mathbf{m}^2)$$

That is, if a transition $\widehat{t}(\neq t)$ is state- and control-enabled at $\mathbf{m}^1$, it remains state- and control-enabled at $\mathbf{m}^2$, as well. Alternately, the firing of transition t at $\mathbf{m}^1$ does not prevent the firing of $\widehat{t}$. The following result that $L(N,\mathbf{m}^0,\mathcal{P})$ is fair if and only if it is non-preemptive.

Theorem 2.2. *(Proposition 3 and 4, [8]) ($L(N,\mathbf{m}^0,\mathcal{P})$ is fair) $\Leftrightarrow$ ($L(N,\mathbf{m}^0,\mathcal{P})$ is non-preemptive)*

As a consequence of these results, if $\mathcal{P}^*$ is the minimally restrictive LESP for $N(\mathbf{m}^0)$, and $L(N,\mathbf{m}^0,\mathcal{P}^*)$ is fair (or, equivalently non-preemptive), then the minimally restrictive LESP $\mathcal{P}^*$ also guarantees maximization of throughput, as well. What remains is a test for fairness (or equivalently, non-preemption) of the supervised system. By using an illustrative example, in the next section we present a necessary and sufficient condition for fairness of the maximally permissive LESP for a class of PNs in the literature. In the interest of readability, we present our observations through an illustrative example.

3. Main Results

The literature contains several references to LESPs that are effectively obtained by augmenting the original PN model with extra places, or *monitors*, along with extra arcs between monitors and the existing transitions. The initial token-load of the monitors are determined uniquely by the initial marking of the original PN model. An *invariant-based monitor* is an instance of monitor based supervision that ensures the markings of the original PN stay within an appropriately defined polyhedron. There are necessary and sufficient conditions for the existence of a minimally restrictive, invariant-based monitor that guarantees livelock-freedom [4]. In cases where the minimally restrictive LESP can be represented by an invariant-based monitor, the supervised PN (and consequently $L(N,\mathbf{m}^0,\mathcal{P})$) is represented by (another) PN.

More specifically, let $N(\mathbf{m}^0)$ be a PN, where $N = (\Pi,T,\Phi,\Gamma)$. *Monitors* are extra places $\Pi_c = \{c_1,c_2,\ldots,c_k\}$ ($\Pi \cap \Pi_c = \emptyset$) that are added to the PN structure N using extra arcs $\Phi_c \subseteq (\Pi_c \times T) \cup (T \times \Pi_c)$ along with their associated weights $\widehat{\Gamma} : \Phi_c \rightarrow \mathcal{N}^+$, to form a new structure $N_c = (\Pi \cup \Pi_c, T, \Phi \cup \Phi_c, \Gamma_c)$, where

$$\Gamma_c(\phi) = \begin{cases} \Gamma(\phi) & \text{if } \phi \in \Phi, \\ \widehat{\Gamma}(\phi) & \text{if } \phi \in \Phi_c. \end{cases}$$

The initial token load of the monitors in Π_c are determined from the initial marking $\mathbf{m}^0$, according to $\Theta : \mathcal{N}^n \rightarrow \mathcal{N}^k$. The PN structure N_c with an initial marking of $((\mathbf{m}^0)^T \ \Theta(\mathbf{m}^0)^T)^T$ is represented as $N_c(\mathbf{m}^0,\Theta(\mathbf{m}^0))$. The set of markings that can be reached from the initial marking $((\mathbf{m}^0)^T \ \Theta(\mathbf{m}^0)^T)^T$ in N_c is denoted by $\Re(N_c,\mathbf{m}^0,\Theta(\mathbf{m}^0))$.

Since there might be arcs in Φ_c that originate from some $c_i \in \Pi_c$ to some uncontrollable transition $t_u \in T_u$, we must require $\forall \mathbf{m} \in \Re(N_c,\mathbf{m}^0,\Theta(\mathbf{m}^0))$,

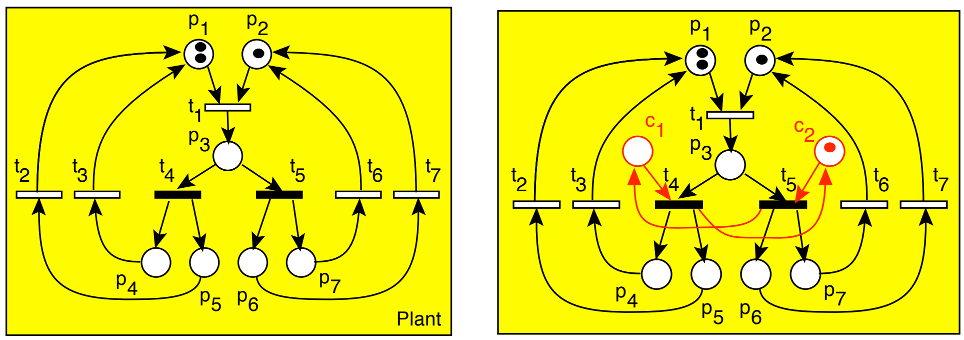
$$(\forall p \in (^{\bullet}t_u \cap \Pi), \ \mathbf{m}(p) \geq \Gamma_c((p,t_u))) \Rightarrow (\forall c \in (^{\bullet}t_u \cap \Pi_c), \mathbf{m}(c) \geq \Gamma_c((c,t_u)))\,.$$

That is, no uncontrollable transition is prevented from firing at some marking that is reachable in $N_c(\mathbf{m}^0,\Theta(\mathbf{m}^0))$ due to a lack of sufficient tokens in the monitors. The requirement, $(\Pi_c \times T_u) \cap \Phi_c = \emptyset$, which supposes that there is no arc from a monitor to an uncontrollable transition in N_c, is sufficient but not necessary, for the above condition to be true.

For $\mathbf{A} \in \mathcal{N}^{k\times n}$, $\mathbf{b} \in \mathcal{N}^k$, an initial marking $\mathbf{m}^0 \in \mathcal{N}^n$ where $\mathbf{Am}^0 \geq \mathbf{b}$, and $\Theta(\mathbf{m}^0) = \mathbf{Am}^0 - \mathbf{b}$, an *invariant-based* monitor ensures $\forall(\mathbf{m}_1^T \ \mathbf{m}_2^T)^T \in \Re(N_c,\mathbf{m}^0,\Theta(\mathbf{m}^0)), \mathbf{Am}_1 \geq \mathbf{b}$ and $\mathbf{m}_2 = \mathbf{Am}_1 - \mathbf{b} \geq \mathbf{0}$ [11, 12]. That is, $\forall(\mathbf{m}_1^T \ \mathbf{m}_2^T)^T \in \Re(N_c,\mathbf{m}^0,\Theta(\mathbf{m}^0))$, the property $\mathbf{Am}_1 \geq \mathbf{b}$, remains *invariant* for all reachable markings.

Liveness enforcement using invariant-based monitors seeks to augment the PN $N(\mathbf{m}^0)$ as described above, such that $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$ is live. When this objective is achieved, the influence of the monitors can be interpreted as an implicit definition of an LESP for the PN $N(\mathbf{m}^0)$.

Consider the PN $N(\mathbf{m}^0)$ shown in Figure 1(a). This PN is not free from livelocks. The minimally restrictive LESP for this PN can be represented using an invariant-based monitor construction, which is shown in Figure 1(b). The monitor places and the extra arcs are shown in red in this figure.



(a) $N(\mathbf{m}^0)$ (b) Invariant-Based Monitor

Figure 1: (a) A PN $N(\mathbf{m}^0)$ that is livelock-prone. The LESP is required to regulate the firing of the controllable transitions t_4 and t_5 in such a manner that the resulting supervised PN is livelock-free. (b) The minimally restrictive LESP, represented as a monitor-based supervisor $\widehat{N}_c(\mathbf{m}^0, \widehat{\Theta}(\mathbf{m}^0))$. The initial token load of the monitor place is given by $\Theta(\mathbf{m}^0) = ((\mathbf{m}^0(p_2) + \mathbf{m}^0(p_3) + \mathbf{m}^0(p_6) + \mathbf{m}^0(p_7) - 1)^T \ (\mathbf{m}^0(p_1) + \mathbf{m}^0(p_3) + \mathbf{m}^0(p_4) + \mathbf{m}^0(p_5) - 1)^T)^T$.

The formal theory behind the construction of invariant-based monitors that are equivalent to the minimally restrictive LESP for a large class of PNs can be found in reference [4]. In this construction, the details of which are skipped for brevity, each monitor-places will serve as input places to some controllable transition. For example, there is an input arc from the monitor place c_1 (resp. c_2) to controllable transition t_4 (resp. t_5). The input arcs to the monitor places can be from any transition in the PN. From the structure of the invariant-based monitor shown in Figure 1(b), we can conclude that the set of monitor places form a place-invariant. Therefore, for any reachable marking of the PN shown in Figure 1(b) only one of the two monitor places would be marked. This in turn would mean that the supervised system of Figure 1(b) is non-preemptive, and from Theorems 2.2 and 2.1, we infer that, in addition to guaranteeing livelock-freedom, this minimally restrictive LESP also maximizes throughput for any distribution of firing-times of the transitions in the PN $N(\mathbf{m}^0)$ of Figure 1(a).

A place $p \in \Pi$ in a PN $N(\mathbf{m}^0)$ is said to be essentially decision free (EDF) [13] if and only if $\forall \mathbf{m} \in \Re(N, \mathbf{m}^0)$ there is at most one transition enabled in $p^\bullet$. That is, $card(T_e(N.\mathbf{m}) \cap p^\bullet) \leq 1$. The following result presents a test for fairness of an invariant-based monitor construction for livelock-avoidance.

Theorem 3.1. *Let $N(\mathbf{m}^0)$ be a PN, where $N = (\Pi, T, \Phi, \Gamma)$. Further, let us suppose that $N_c = (\Pi \cup \Pi_c, T, \Phi \cup \Phi_c, \Gamma_c)$ represents the invariant-based monitor that is equivalent to the minimally restrictive LESP $\mathcal{P}^*$ for $N(\mathbf{m}^0)$. Then $L(N, \mathbf{m}^0, \mathcal{P})$ is fair if and only if every place in $N_c(\mathbf{m}^0, \Theta(\mathbf{m}^0))$ is essentially decision free (EDF).*

The proof of this claim follows directly from the definition of the EDF property and the previous results. The EDF property of places is decidable [13], which in turn implies that it is possible to check if the minimally restrictive LESP also maximizes throughput for the large class of PNs identified in reference [4].

4. Conclusion

We considered supervisory policies that enforce the qualitative property of livelock-freedom in models of manufacturing systems, and we identified a necessary and sufficient algebraic structural condition that ensures that the same policy also satisfies the quantitative requirement of maximizing the throughput. It is very important to mention that most of the *flexible manufacturing* systems and specially *flexible assembly* systems (FAS), comply with the non-preemptive property; FAS consists of controlled assembly stations connected by automated material handling system, which gives the system the flexibility of simultaneously assembling different products at the same time. Translating to the language of PNs, these systems are capable of firing different transitions at the same time; To be precise, firing any enabled transition at a marking, keeps the remaining transitions still enabled at the newly reached state. Therefore, regulating such a system with a minimally restrictive LESP not only makes the system live-lock free, but also improves the throughput rate of the system, speed in which the system produces products in general, regardless of their type. A good examples of such systems can be found in electronic and semiconductor industry [14]. On the other hand, most of the manufacturing systems do not comply with such a flexibility and it becomes an interesting avenue to explore: Is there any algebraic structure for the underlying PN of these systems that suggest a trade off between the minimally restrictive LESP and improving performance measures such as throughput. That being said, our future work will be directed to studying tradeoffs between throughput and controllability when some livelock situations can be tolerated.

References

[1] J. Peterson, Petri net theory and the modeling of systems. Englewood Cliffs, NJ: Prentice-Hall, 1981.

[2] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, no. 4, pp. 541?580, 1989.

[3] V. Deverakonda and R.S. Sreenivas, "On a Sufficient Information Structure for Supervisory Policies that Enforce Liveness in a Class of General Petri Nets," *IEEE Transactions on Automatic Control*, Vol. 60, No. 7, July 2015, 1915-1921.

[4] E. Salimi and R.S. Sreenivas, "On Invariant-Based Monitors that Enforce Liveness in a Class of Partially Controlled General Petri Nets," *IEEE Transactions on Automatic Control*, Vol. 60, No. 10, October 2015, 2825-2830.

[5] S. Chandrasekaran, N. Somnath and R.S. Sreenivas, "A software tool for the automatic synthesis of minimally restrictive liveness enforcing supervisory policies for a class of general Petri net models of manufacturing- and service-systems," *Journal of Intelligent Manufacturing*, Volume 26, No. 5, October, 2015, 945-958.

[6] R.S. Sreenivas, "On the Existence of Supervisory Policies that Enforce Liveness in Partially Controlled Free-Choice Petri Nets," *IEEE Transactions on Automatic Control*, 435-449, Vol. 57, No. 2, February, 2012.

[7] R.Sreenivas, "On the existence of supervisory policies that enforce liveness in discrete-event dynamic systems modeled by controlled Petri nets," *IEEE Transactions on Automatic Control*, vol. 42, no. 7, pp. 928?945, July 1997.

[8] F. Lin, "A linguistic approach to stochastic comparison of queuing systems," *European Journal of Operational Research*, 77 (1994), 27-39.

[9] P.W. Glynn, "A GSMP formalism for discrete event systems," *Proceedings of the IEEE*, 77:1 (1989), 14-23.

[10] P. Glasserman and D. Yao, "Algebraic Structure of Some Stochastic Discrete Event Systems, with Applications," *Discrete Event Dynamic Systems: Theory and Applications*, 1 (1991), 7-35.

[11] A. Giua, "Petri nets as discrete event models for supervisory control," Ph.D. dissertation, ECSE Dept., Rensselaer Polytechnic Institute, Troy, NY., 1992.

[12] J. Moody and P. Antsaklis, *Supervisory Control of Discrete Event Systems using Petri Nets*. MA: Kluwer Academic Publishers, 1998.

[13] B.H. Krogh and R.S. Sreenivas, "Essentially-Decision-Free Petri nets for Real-Time Resource Allocation," IEEE International Conference on Robotics and Automation, Raleigh, North Carolina, 1987.

[14] Sawik, Tadeusz, "Production Planning and Scheduling in Flexible Assembly Systems: With 66 Figures and 51 Tables," Springer Science and Business Media, 1999.

[15] Hou, C. Q., S. Y. Li, Y. Cai, H. M. Wu, A. M. An, and Y. Wang, "A livelock control policy for a flexible manufacturing system modeling with a subclass of generalized Petri nets." Cogent Engineering 1, no. 1 (2014): 944766.