A PETRI NET APPROACH TO THE MODELLING AND ANALYSIS OF FLEXIBLE MANUFACTURING SYSTEMS*

Y. NARAHARI and N. VISWANADHAM

School of Automation, Indian Institute of Science, Bangalore 560012, India

Abstract

In this paper we present an approach for modelling and analyzing flexible manufacturing systems (FMSs) using Petri nets. In this approach, we first build a Petri net model (PNM) of the given FMS in a bottom-up fashion and then analyze important qualitative aspects of FMS behaviour such as existence/absence of deadlocks and buffer overflows. The basis for our approach is a theorem we state and prove for computing the invariants of the union of a finite number of Petri nets when the invariants of the individual nets are known. We illustrate our approach using two typical manufacturing systems: an automated transfer line and a simple FMS.

Keywords and phrases

Flexible manufacturing systems, concurrency, Petri net modelling, union of Petri nets, place invariants, deadlocks.

1. Introduction

In recent times, *flexible manufacturing systems* (FMSs) have become very popular on account of their high levels of productivity and low levels of inventory costs. A typical FMS comprises 5 – 20 machines and 10 – 50 part types and involves numerous, complex interactions. Much of the research in FMSs has sought to enable an easier understanding of these interactions and to this end, many modelling schemes

*A shorter version of this paper was presented at the 1st ORSA/TIMS Special Interest Conference on FMSs, University of Michigan, Ann Arbor, August 1984.

[3] have been proposed. One of the recent proposals is the *Petri net model* [4], the classical model of concurrent systems. The Petri net model (PNM) is graphically elegant and is supported by rich mathematical theory. It provides a compact model to capture the intricacies of FMS interactions.

In literature, Petri net models (PNMs) fall into one of two classes: *Timed PNMs* [10] and *Untimed PNMs* [1,9]. A timed PNM of an FMS captures the actual physical behaviour of the FMS by assuming specific durations for various activities in the FMS. An untimed PNM of an FMS does not associate any times with the activities and so models the physical behaviour of the FMS under all possible time dependencies. Hence, analysis using untimed PNMs yields more conservative results compared to those obtained using timed PNMs. Most of the useful theoretical results in Petri net theory pertain to untimed Petri nets. Moreover, the results that have been derived for timed Petri nets are valid only for a restricted class of nets and there is not much hope of deriving the results for more generalized timed Petri nets [10]. Untimed PNMs are mainly used for understanding qualitative aspects of the modelled system, such as existence/absence of deadlocks, buffer overflows, and mutual exclusion [1,9]. Timed PNMs are useful in computing quantitative performance measures such as throughput rate and processing times [10]. In this paper, our main objective is to gain insights into the qualitative behaviour of a given FMS with regard to deadlocks, buffer overflows, invariance of number of jobs, etc. Hence, we use untimed Petri nets to model FMSs. In contrast to our using untimed Petri nets for qualitative analysis of FMSs, Dubois and Stecke [4] use timed Petri nets for real-time control and performance evaluation of FMSs. In the sequel of this paper, a PNM refers to an untimed PNM.

The contribution of this paper is twofold. First, we show how PNMs can be systematically built for given FMSs. The construction of PNMs follows a *bottom-up* approach in the sense that the given FMS is decomposed into functional sub-units and the PNM of the FMS is obtained by coalescing the PNMs of the functional sub-units. Secondly, we develop two useful theorems that enable computation of *invariants* of the *union* of a *finite* number of Petri nets when the invariants of the individual nets are known and the nets satisfy certain conditions. We use the above theorem to compute the invariants of the overall PNM of the given FMS. We then use the classical invariant analysis of Petri net theory [2,6] to determine potential qualitative properties of the FMS such as existence/absence of deadlocks, buffer overflows, invariance of number of jobs in the system, and recoverability from failures. Thus our approach helps in building PNMs for FMSs and in the analysis/verification of the qualitative properties of the FMSs.

The rest of the paper is organized as follows. In sect. 2, we present a comprehensive review of relevant Petri net theory and bring out the relevance of Petri net modelling of FMSs. In sect. 3, we review Petri net invariants and prove the main results of the paper. We show, in sect. 4, how a PNM can be systematically constructed for a given FMS and how the results of sect. 3 become useful in the qualitative analysis of the FMS. In sect. 5, we present the application of our approach to two illustrative manufacturing systems: A transfer line with three machines and two buffers, and a simple FMS with three machines and two part types.

2. Introduction to Petri nets

In this section, we give an introduction to Petri net theory through some definitions and examples. These definitions are quite standard and are mostly taken from [1] and [9]. We have slightly altered the standard notation to suit our subsequent discussions. We also introduce certain potential properties of a Petri net and discuss their significance in the FMS context.

2.1. BASIC DEFINITIONS

Definition 2.1. A Petri net G is a four-tuple $(P, T, \mathrm{IN}, \mathrm{OUT})$ where

$P = \{p_1, p_2, p_3, \ldots, p_n\}$ is a set of places,

$T = \{t_1, t_2, t_3, \ldots, t_m\}$ is a set of transitions,

$P \cup T \neq \emptyset,\ P \cap T = \emptyset,$

IN: $(P \times T) \to N$ is an input function that defines directed arcs from places to transitions (N is the set of all non-negative integers), and

OUT: $(P \times T) \to N$ is an output function that defines directed arcs from transitions to places. □

Pictorially, places are represented by circles and transitions by horizontal bars. If $\mathrm{IN}(p_i, t_j) = k$, where $k > 1$ is an integer, we include a directed arc from place p_i to transition t_j and label it by k. If $\mathrm{IN}(p_i, t_j) = 1$, we include a directed arc but without any label. If $\mathrm{IN}(p_i, t_j) = 0$, we do not include any directed arc from p_i to t_j.

Example 2.1. Let us consider a machine that processes one job at a time. As soon as the processing is over, another job is made available and the machine starts processing again. Figure 1(a) depicts a Petri net model (PNM) of the above system. The places and the transitions have the following interpretation.

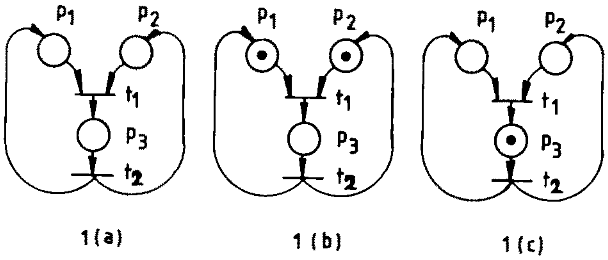


Fig. 1. (a) A PNM that represents the processing of a job on a machine. (b) Initial marking M_0 of the PNM. (c) Marking M_1 reached after firing t_1 in (b).

p_1 : machine ready to process;

p_2 : job waiting for processing;

p_3 : job undergoing machining;

t_1 : machining commences;

t_2 : machining concludes.

In the above example, it may be noted that the places represent various conditions in the system, whereas transitions represent the commencement or conclusion of events. We have assumed that the machine, if it breaks down, will be repaired and will resume its operation on the job. For the above PNM,

$$P = \{p_1, p_2, p_3\}, \qquad T = \{t_1, t_2\}, \quad \text{and}$$

$$\mathrm{IN}\ (p_1, t_1) = \mathrm{IN}\ (p_2, t_1) = \mathrm{IN}\ (p_3, t_2) = 1$$

$$\mathrm{IN}\ (p_3, t_1) = \mathrm{IN}\ (p_1, t_2) = \mathrm{IN}\ (p_2, t_2) = 0$$

$$\mathrm{OUT}(p_3, t_1) = \mathrm{OUT}(p_1, t_2) = \mathrm{OUT}(p_2, t_2) = 1$$

$$\mathrm{OUT}(p_1, t_1) = \mathrm{OUT}(p_2, t_1) = \mathrm{OUT}(p_3, t_2) = 0.$$

Definition 2.2. Let 2^P be the powerset of P. We then define functions $\mathrm{IP}: T \to 2^P$ and $\mathrm{OP}: T \to 2^P$ as follows:

$$\mathrm{IP}\ (t_j) = \{p_i \in P: \mathrm{IN}\ (p_i, t_j) \neq 0\} \quad \forall t_j \in T,$$

$$\mathrm{OP}(t_j) = \{p_i \in P: \mathrm{OUT}(p_i, t_j) \neq 0\} \quad \forall t_j \in T.$$

IP (t_j) is called the set of *input places* of t_j, and

OP(t_j) the set of *output places* of t_j. □

For the PNM of fig 1(a),

$$\mathrm{IP}(t_1) = \mathrm{OP}(t_2) = \{p_1, p_2\} \text{ and}$$

$$\mathrm{OP}(t_1) = \mathrm{IP}(t_2) = \{p_3\} .$$

Definition 2.3. A *marking* M of a Petri net G is a function $M: P \to N$, where N is the set of all non-negative integers. □ A *marked Petri net* (MPN) W is a Petri net G together with a marking defined on it. We denote an MPN by (G, M) and write $W = (G, M)$. We generally associate an *initial marking* M_0 with a given PNM. M_0 will represent the initial state of the system which the PNM is modelling.

A marking of a Petri net with n places in an $(n \times 1)$ vector and associates with each place a certain number of *tokens* which are represented by means of dots inside the places. Figure 1(b) gives a marked Petri net with marking M_0 given by

$$M_0 = (M_0(p_1), M_0(p_2), M_0(p_3))^{\mathrm{T}} = (1, 1, 0)^{\mathrm{T}} .$$

In the above, the superscript T denotes transpose operation.

The marking M_1 of the PNM of fig. 1(c) is given by

$$M_1 = (M_1(p_1), M_1(p_2), M_1(p_3))^{\mathrm{T}} = (0, 0, 1)^{\mathrm{T}} .$$

Definition 2.4. A transition t_j of a Petri net is said to be *enabled* in a marking M iff

$$M(p_i) \geqslant \mathrm{IN}(p_i, t_j) \ \forall \ p_i \in \mathrm{IP}(t_j) .$$

An *enabled* transition t_j can *fire* anytime. When a transition t_j, enabled in a marking M, fires, a new marking M' is reached according to the equation

$$M'(p_i) = M(p_i) + \mathrm{OUT}(p_i, t_j) - \mathrm{IN}(p_i, t_j) \ \forall \ p_i \in P .$$

We say marking M' is *reachable* from M. □

In fig. 1(b), transition t_1 is enabled in marking M_0. When t_1 fires, the marking M_1 is reached. Transition t_2 is enabled in M_1 and when t_2 fires, the new marking is M_0.

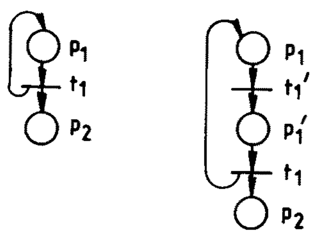
Definition 2.5. Reachability of markings is a *reflexive* and *transitive* relation on the set of all markings. The set of all markings reachable from an initial marking M_0 of a Petri net is said to be the *reachability set* of M_0 and is denoted by $R[M_0]$. □
It can be seen from figs. 1(b) and 1(c) that

$$R[M_0] = R[M_1] = \{M_0, M_1\}.$$

Definition 2.6. A Petri net $G = (P, T, \mathrm{IN}, \mathrm{OUT})$ is said to be *pure* or *self-loop free* iff there exists no pair $(p_i, t_j) \in P \times T$ for which

$$\mathrm{IN}(p_i, t_j) \neq 0 \quad \text{and} \quad \mathrm{OUT}(p_i, t_j) \neq 0. \quad \square$$

In a pure Petri net, there exists no place which is an input place and an output place of the same transition. Given a Petri net that is *not* pure, we can always get an equivalent *pure* Petri net by introducing dummy places and dummy transitions, as shown in figs. 2(a) and 2(b).



2 (a) 2 (b)

Fig. 2. (a) An impure Petri net. (b) A pure Petri net equivalent of the above Petri net. p_1' is a dummy place, t_1' is a dummy transition.

Definition 2.7. The incidence matrix C of a *pure* Petri net is an $(n \times m)$ matrix (n is the number of places and m is the number of transitions) defined by

$$C(i, j) = -\mathrm{IN}(p_i, t_j) \qquad \text{if } \mathrm{IN}\ (p_i, t_j) \neq 0$$

$$= \mathrm{OUT}(p_i, t_j) \qquad \text{if } \mathrm{OUT}(p_i, t_j) \neq 0$$

$$= 0 \qquad \text{otherwise}. \qquad \square$$

In the above definitions, $C(i, j)$ is the (i, j)th element of C, where $i = 1, 2, 3, \ldots, n$ and $j = 1, 2, 3, \ldots, m$.

A pure Petri net is completely described by its incidence matrix. For the Petri net of fig. 1(a),

$$C(1,1) = -1, \quad C(1,2) = 1, \quad C(2,1) = -1,$$

$$C(2,2) = 1, \quad C(3,1) = 1, \text{ and } C(3,2) = -1.$$

Definition 2.8. Let $G_1 = (P_1, T_1, \mathrm{IN}_1, \mathrm{OUT}_1)$ and $G_2 = (P_2, T_2, \mathrm{IN}_2, \mathrm{OUT}_2)$ be two *pure* Petri nets such that there exists no pair, $p \in P_1 \cap P_2$ and $t \in T_1 \cap T_2$, satisfying

$$\text{either } \mathrm{IN}_1 \ (p, t) \neq 0 \text{ and } \mathrm{IN}_2 \ (p, t) \neq 0,$$

$$\text{or } \mathrm{OUT}_1(p, t) \neq 0 \text{ and } \mathrm{OUT}_2(p, t) \neq 0.$$

We define their union as the Petri net $G = (P, T, \mathrm{IN}, \mathrm{OUT})$ where

$$P = P_1 \cup P_2, \quad T = T_1 \cup T_2,$$

$$\mathrm{IN} = \mathrm{IN}_1 \cup \mathrm{IN}_2, \text{ and } \mathrm{OUT} = \mathrm{OUT}_1 \cup \mathrm{OUT}_2.$$

The union of any finite number of Petri nets is also defined likewise. $\square$

For example, the Petri net of fig. 3(c) is the union of the Petri nets in figs. 3(a) and 3(b).

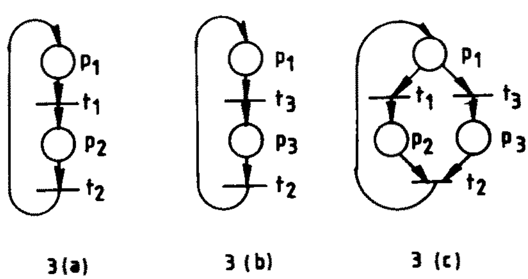


Fig. 3. (a) and (b) Two Petri nets. (c) Union of the Petri nets in (a) and (b).

2.2. RELEVANCE OF PETRI NET MODELLING TO FMSs

In our approach of modelling FMSs by Petri nets, we use the following interpretation for places, transitions, and tokens:

(1) Places represent conditions or resources (pallets or machines) or parts (buffer).
(2) If a place represents a condition, a token in the place indicates that the condition is true and no token indicates that the condition is false. If a place represents resources, a token in it represents a single resource and if a place represents a buffer, a token in it stands for a part.
(3) Transitions represent either commencement or conclusion of events.

We assume the above interpretation for all PNMs in the rest of the paper. We now define certain potential properties of a Petri net and discuss their relevance in the FMS context.

Definition 2.9. Safeness and boundedness. A place p_i of a Petri net is said to be *bounded* in a marking M_0 iff there exists a positive integer B such that

$$M(p_i) \leqslant B \ \forall \ M \in R\,[M_0] \,.$$

If $B = 1$, we say the place is *safe*. If all places of a Petri net are bounded (safe) in a marking M_0, the Petri net itself is said to be bounded (safe) in that marking. □

Boundedness of a PNM refers to absence of overflows in the modelled system. If the PNM of an FMS is bounded, we can say that all buffers in the FMS are finite. In such a case, buffer sizes can be estimated by finding the bounds on the places that represent the buffers.

Definition 2.10. Conservativeness. A marked Petri net $W = (G, M_0)$ is said to be *conservative* iff

$$\sum_{i=1}^{n} M(p_i) = \text{constant} \ \forall \ M \in R\,[M_0] \,. \ \square$$

If a marked Petri net is conservative, then the sum of all tokens will remain a constant in all reachable markings. Such a PNM will represent a system with constant number of resources/jobs, for example, a closed queueing system.

Definition 2.11. Properness. A marked Petri net $W = (G, M_0)$ is said to be *proper* iff

$$M_0 \in R[M] \; \forall \; M \in R[M_0] \,. \; \square$$

In a proper Petri net, the initial marking is reachable from all reachable markings. Thus, properness implies re-initializability. This assumes much significance in the context of fully automated manufacturing systems. For, properness in such systems will ensure that the system will eventually re-initialize itself from any current state (which could be an illegal state or a failure state). If the PNM of an FMS is not proper, then we can say that manual intervention is necessary to restore the system to its initial state from certain states.

Definition 2.12. Liveness. A transition t_j of a marked Petri net $W = (G, M_0)$ is said to be *live* iff, for all reachable markings M, there exists a sequence of transition firings which results in a marking in which t_j is enabled. A Petri net is said to be *live* if all its transitions are live. □

Liveness of a PNM implies absence of deadlocks in the modelled system. Deadlocks occur in a system when processes, which want to run, hold insufficient resources, as a result of which the system comes to a standstill. Deadlocks could occur in an FMS because of the complex nature of the interactions in the FMS. Using the PNM of an FMS, we can determine if deadlocks exist and take the necessary action for their prevention.

Examples. The marked Petri net of fig. 1(b) is safe, bounded, live, proper, but not conservative. Figures 4(a)–4(d) depict PNMs satisfying different sets of these properties.

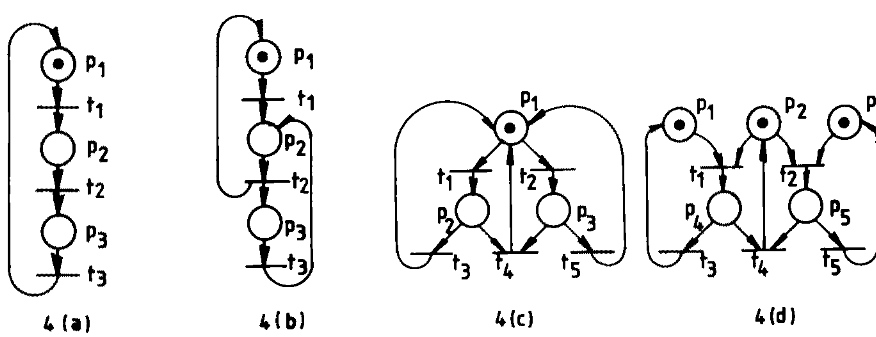


Fig. 4. (a) An MPN (marked Petri net) that is safe, bounded, conservative, proper and live. (b) An MPN that is not safe, not bounded, not conservative, not proper but live. (c) An MPN that is safe, bounded, conservative, proper but not live. t_4 is not live. (d) An MPN that is safe, bounded, not conservative, not proper, and not live.

3. Two theorems on the invariants of a Petri net

In this section, we review the important concept of Petri net invariants [6] and present our results which form a basis of the approach, to be discussed in sect. 4, for the modelling and analysis of FMSs. The main idea for the development of these results has been taken from [2].

3.1. PETRI NET INVARIANTS

We introduce in the following definition, a slight change in standard notation for ease of presentation.

Definition 3.1. Let $G = (P, T, \mathrm{IN}, \mathrm{OUT})$ be a *pure* Petri net with incidence matrix C. Let

$$P = \{p_1, p_2, \ldots, p_n\} \quad \text{and} \quad T = \{t_1, t_2, \ldots, t_m\} .$$

A $(1 \times n)$ row-vector X is said to be a *place invariant* or *p-invariant* of G iff $XC = 0$ and an $(m \times 1)$ column-vector Y is said to be a *transition invariant* or *t-invariant* iff $CY = 0$. □

Theoretically, X and Y could consist of real numbers as components. However, we find it more than adequate to consider only integers as the components.

3.1.1. Significance of invariants. Let (G, M_0) be a pure marked Petri net. If M is some reachable marking, then it may be shown that

$$M = M_0 + CY, \tag{3.1}$$

where C is the incidence matrix of G and Y is an $(m \times 1)$ vector with non-negative integer entries. For $j = 1, 2, 3, \ldots, m$, the jth entry of Y gives the number of times transition t_j is fired in a firing sequence that leads to M starting from M_0. Multiplying both sides of eq. (3.1) by a $(1 \times n)$ row-vector X, we get $XM = XM_0 + XCY$. If X is a p-invariant, the above becomes $XM = XM_0$. This implies that, for all reachable markings, the weighted sum of tokens is a constant, the weights being given by the p-invariant.

If Y in eq. (3.1) is a t-invariant, then $CY = 0$ and hence $M = M_0$. This implies that a t-invariant, if it exists, will give the number of times different transitions should be fired in order that a particular marking may be *reproducible*.

3.1.2. Use of invariants. A knowledge of the invariants of a Petri net is very useful for establishing/disproving some important properties of Petri nets such as boundedness, conservativeness, properness, and liveness. We corroborate the above statement by stating below, without proof, the following theorems.

THEOREM 3.1

A Petri net G is bounded if there exists a p-invariant X all of whose entries are strictly positive. □

THEOREM 3.2

A Petri net G is conservative iff there exists a p-invariant X all of whose entries are equal to unity. □

THEOREM 3.3

A Petri net G is not proper if its only t-invariant is the trivial invariant, that is with all its components equal to zero. □

The above theorems give a sufficient condition for boundedness, a necessary and sufficient condition for conservativeness, and a necessary condition for properness, respectively. In most cases, a knowledge of the invariants together with some additional information about the modelled system will be sufficient to completely investigate these above properties and also liveness.

3.2. TWO THEOREMS ON PETRI NET INVARIANTS

We now develop two results on Petri net invariants. Theorem 1 facilitates computation of p-invariants of the union of two Petri nets when the p-invariants of the individual Petri nets are known. Theorem 2 is an identical result on t-invariants. These results form the nucleus of the FMS modelling and analysis approach subsequently discussed in this paper.

THEOREM 1

Let $G_1 = (P_1, T_1, \mathrm{IN}_1, \mathrm{OUT}_1)$ and $G_2 = (P_2, T_2, \mathrm{IN}_2, \mathrm{OUT}_2)$ be two pure Petri nets such that their union $G = (P, T, \mathrm{IN}, \mathrm{OUT})$ is also pure. Let r, s, and n be positive integers such that $r \leqslant s < n$. Also, let $T_1 \cap T_2 = \emptyset$ and

$$P_1 = \{p_1, p_2, \ldots, p_r, p_{r+1}, \ldots, p_s\},$$

$$P_2 = \{p_r, p_{r+1}, \ldots, p_{s+1}, \ldots, p_n\},$$

so that

$$P = \{p_1, p_2, \ldots, p_{r+1}, \ldots, p_s, p_{s+1}, \ldots, p_n\}.$$

Then the n-vector

$$X = (x_1, x_2, \ldots, x_r, x_{r+1}, \ldots, x_{s+1}, \ldots, x_n)$$

will be a p-invariant of G iff the s-vector

$$X_1 = (x_1, x_2, \ldots, x_r, x_{r+1}, \ldots, x_s)$$

is a p-invariant of G_1 and the $(n - r + 1)$-vector

$$X_2 = (x_r, x_{r+1}, \ldots, x_s, x_{s+1}, \ldots, x_n)$$

is a p-invariant of G_2. □

Proof

Let $T_1 = \{t_1, t_2, \ldots, t_k\}$ and $T_2 = \{t_{k+1}, t_{k+2}, \ldots, t_m\}$, where $k < m$. Let C_1, C_2 and C represent the incident matrices of G_1, G_2 and G, respectively. If $C_1(i_1, j_1)$, $C_2(i_2, j_2)$ and $C(i,j)$ represent typical elements of C_1, C_2 and C, respectively, then the ranges of the indices i_1, i_2, j_1, j_2, i and j are given by:

$$i_1 = 1, 2, 3, \ldots, s;$$

$$j_1 = 1, 2, 3, \ldots, k;$$

$$i_2 = r, r+1, r+2, \ldots, s, \ s+1, \ldots, n;$$

$$j_2 = k+1, k+2, \ldots, m;$$

$$i = 1, 2, 3, \ldots, n;$$

$$j = 1, 2, 3, \ldots, m.$$

Further, it can be seen from the definition of union of Petri nets (Def. 2.8) that

$$\begin{aligned} C(i,j) &= C_1(i,j) \quad i = 1, 2, \ldots, s; \quad j = 1, 2, \ldots, k \\ &= C_2(i,j) \quad i = r, r+1, \ldots, n; \quad j = k+1, \ldots, m \\ &= 0 \quad \text{otherwise}. \end{aligned} \tag{3.2}$$

We first prove the necessary part. The sufficiency part can be proved by reversing the arguments. To prove the necessary part, we are given that X_1 and X_2 are p-invariants of G_1 and G_2, respectively. This implies that

$$\sum_{i=1}^{s} x_i C_1(i,j) = 0 \quad \text{for} \quad j = 1, 2, \ldots, k \tag{3.3}$$

and

$$\sum_{i=r}^{n} x_i C_2(i,j) = 0 \quad \text{for} \quad j = k+1, \ldots, m. \tag{3.4}$$

For $j = 1, 2, \ldots, k$, (3.2) and (3.3) give

$$\sum_{i=1}^{n} x_i C(i,j) = \sum_{i=1}^{s} x_i C(i,j) + \sum_{i=s+1}^{n} x_i C(i,j) = 0.$$

For $j = k+1, \ldots, m$, (3.2) and (3.4) give

$$\sum_{i=1}^{n} x_i C(i,j) = \sum_{i=1}^{r-1} x_i C(i,j) + \sum_{i=r}^{n} x_i C(i,j) = 0.$$

Combining the above two equations, we obtain

$$\sum_{i=1}^{n} x_i C(i,j) = 0 \quad \text{for} \quad j = 1, 2, \ldots, m.$$

Thus, the vector $X = (x_1, x_2, \ldots, x_r, x_{r+1}, \ldots, x_s, \ldots, x_n)$ is a p-invariant of G. □

We now state an identical result on t-invariants. We have omitted the proof since it runs on identical lines as the previous proof.

THEOREM 2

Let $G_1 = (P_1, T_1, \mathrm{IN}_1, \mathrm{OUT}_1)$ and $G_2 = (P_2, T_2, \mathrm{IN}_2, \mathrm{OUT}_2)$ be two pure Petri nets such that their union $G = (P, T, \mathrm{IN}, \mathrm{OUT})$ is also pure. Let r, s, and m be integers such that $0 < r \leqslant s < m$ and let

$$T_1 = \left\{t_1, t_2, \ldots, t_r, t_{r+1}, \ldots, t_s\right\}$$

and

$$T_2 = \left\{t_r, t_{r+1}, \ldots, t_s, t_{s+1}, \ldots, t_m\right\}$$

so that

$$T = \left\{t_1, t_2, \ldots, t_r, t_{r+1}, \ldots, t_s, t_{s+1}, \ldots, t_m\right\} .$$

Also, let $P_1 \cap P_2 = \emptyset$. Then the m-vector $Y = (y_1, y_2, \ldots, y_m)^{\mathrm{T}}$ is a t-invariant of G iff the s-vector $Y_1 = (y_1, y_2, \ldots, y_r, y_{r+1}, \ldots, y_s)^{\mathrm{T}}$ is a t-invariant of G_1 and the $(m - r + 1)$-vector $Y_2 = (y_r, y_{r+1}, \ldots, y_s, y_{s+1}, \ldots, y_m)^{\mathrm{T}}$ is a t-invariant of G_2. □

Remarks on theorems 1 and 2

(1) In these theorems, the Petri nets G_1, G_2 and G have all got to be pure. This does not, however, restrict the use of the theorems since any Petri net that is not pure can be made equivalent to a pure Petri net as shown in figs. 2(a) and 2(b).

(2) Given a Petri net, all its p-invariants can be computed by decomposing it into simpler Petri nets and then applying theorem 1. Similarly, application of theorem 2 helps in computing all t-invariants of a given Petri net.

(3) The theorems can be generalized to any finite number of Petri nets by repeated application.

4. The approach

We now present our approach to the modelling and analysis of FMSs using Petri nets.

An FMS involves numerous concurrent interactions which can be viewed as the sum total of interactions on various part types in the FMS. The processing on each part type is composed of a sequence of operations on the part type. Each operation of the sequence will be performed on one of a finite set of machines since there will be a choice of machines for virtually every operation. To construct a PNM for the given FMS, we first represent each machine operation by a PNM. We then obtain the PNM for each specific operation as the *union* of the PNMs corresponding to the processing on the machine belonging to the choice list of machines of that operation. Next, we obtain the PNM of each part type as the *union* of the PNMs of the operations involved in the processing of that part type. Finally, the PNM of the FMS is obtained by constructing the *union* of the PNMs of all part types. Thus we have a systematic bottom-up scheme for building a PNM of the given FMS.

At each stage of building the PNM, we can invoke theorem 1 to compute the p-invariants of the PNM obtained by coalescing the PNMs of the previous stage. Thus we can obtain the p-invariants of the overall PNM of the FMS. Using the p-invariants, we can investigate the qualitative behaviour of the given FMS by evaluating the properties of the PNM. The next section presents two examples to illustrate this analysis procedure.

5. Examples

5.1. A TRANSFER LINE CONSISTING OF THREE MACHINES AND TWO BUFFERS

Figure 5(a) shows a transfer line that comprises three machines M_1, M_2 and M_3 and two buffers B_1 and B_2 with capacities n_1 and n_2, respectively, ($n_1 > 0, n_2 > 0$). We make the following assumptions on the above system:

(i) There is a perennial source of jobs and the flow control mechanism allows a job into the system whenever M_1 if free.

(ii) Machines M_1 and M_2 will process a job only if each of buffers B_1 and B_2, in that order, has at least one empty slot to accommodate the jobs processed by M_1 and M_2.

(iii) Jobs waiting in B_1 and B_2 are released for processing in a random order.

Let G_1, G_2, and G_3 be the PNMs depicted in figs. 5(b), 5(c) and 5(d), respectively. For i = 1, 2, 3, G_i represents the processing that takes place on M_i. If G is the union of G_1, G_2 and G_3, then G represents all the activities occurring in the transfer line. The interpretation of the places and transitions of the above PNMs is as follows. The index i takes values 1, 2 and 3 and j takes values 1, 2 in the following:

JR – a job ready for processing by M_1,

MR_i – machine M_i ready to process a job,

P_i – machine M_i processing a job,

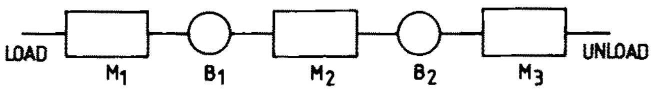
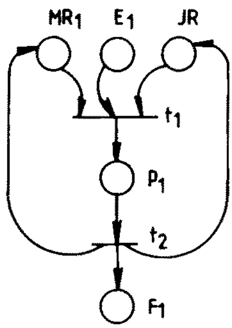
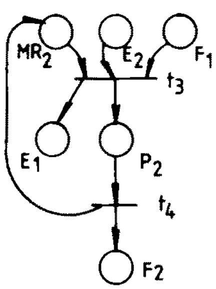


Fig. 5. (a) A transfer line with 3 machines and 2 buffers.



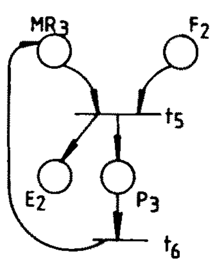
E_1	MR_1	JR	P_1	F_1
e_1	e_2	e_3	$e_1+e_2+e_3$	e_1

Fig. 5. (b) PNM and *p*-invariants for processing on M_1.



MR_2	E_2	F_1	E_1	P_2	F_2
e_4	e_5	e_1	e_1	e_4+e_5	e_5

Fig. 5. (c) PNM and *p*-invariants for processing on M_2.



F_2	MR_3	P_3	E_2
e_5	e_6	e_6	e_5

Fig. 5. (d) PNM and *p*-invariants for processing on M_3.

E_1	MR_1	JR	P_1	F_1	MR_2	F_2	E_2	P_2	MR_3	P_3
e_1	e_2	e_3	$e_1+e_2+e_3$	e_1	e_4	e_5	e_5	e_4+e_5	e_6	e_6

Fig. 5. (e) *p*-invariants of the PNM of the transfer line.

Fig. 5. Transfer line example.

E_j – empty slots in buffer B_j,
F_j – occupied slots in buffer B_j,
t_{2i-1} – machine i commences processing a job,
t_{2i} – machine i finishes processing a job.

The table in fig. 5(e) shows the p-invariants of G which are obtained from the p-invariants of G_1, G_2 and G_3 shown in figs. 5(b), 5(c) and 5(d), respectively. It may be noted that theorem 1 has been used in writing down these p-invariants. In the above p-invariants, the quantities e_1, e_2, e_3, e_4, e_5 and e_6 are any real numbers but, for the sake of convenience, we take them as integers.

Let us assume that the above system has the following initial state: All machines ready to process, all buffers empty, and a solitary job waiting to be processed by M_1. If M_0 is the initial marking of G, corresponding to the above initial state, we have

$$M_0(MR_1) = M_0(MR_2) = M_0(MR_3) = M_0(JR) = 1$$

$$M_0(E_1) = n_1 ; M_0(E_2) = n_2$$

$$M_0(P_1) = M_0(P_2) = M_0(F_1) = M_0(F_2) = M_0(P_3) = 0 .$$

Let M be any reachable marking of the marked Petri net (G, M_0). If $S(M)$ denotes the weighted sum of tokens in marking M, taking the weights as the components of the p-invariant shown in fig. 5(e), we get

$$\begin{aligned} S(M) = {} & e_1 M(E_1) + e_2 M(MR_1) + e_3 M(JR) \\ & + (e_1 + e_2 + e_3) M(P_1) + e_1 M(F_1) + e_4 M(MR_2) \\ & + e_5 M(F_2) + e_5 M(E_2) + (e_4 + e_5) M(P_2) \\ & + e_6 M(MR_3) + e_6 M(P_3) \end{aligned} \tag{5.1}$$

and

$$S(M_0) = e_1 n_1 + e_5 n_2 + e_2 + e_3 + e_4 + e_6 . \tag{5.2}$$

By the property of p-invariants, we have

$$S(M) = S(M_0) \ \forall\, M \in R\,[M_0] \ . \tag{5.3}$$

Equations (5.1)–(5.3) imply the following equations:

$$M(\mathrm{E}_1) + M(\mathrm{P}_1) + M(\mathrm{F}_1) = n_1 \ , \tag{5.4}$$

$$M(\mathrm{MR}_1) + M(\mathrm{P}_1) = 1 \ , \tag{5.5}$$

$$M(\mathrm{JR}) + M(\mathrm{P}_1) = 1 \ , \tag{5.6}$$

$$M(\mathrm{MR}_2) + M(\mathrm{P}_2) = 1 \ , \tag{5.7}$$

$$M(\mathrm{E}_2) + M(\mathrm{F}_2) + M(\mathrm{P}_2) = n_2 \ , \tag{5.8}$$

$$M(\mathrm{MR}_3) + M(\mathrm{P}_3) = 1 \ . \tag{5.9}$$

We can now make the following observations about the system under consideration.

(i) The PNM G is bounded and non-conservative. This follows as a direct application of theorems 3.1 and 3.2. Thus there are no overflows in the system and also the number of jobs being processed does not remain a constant at all times.

(ii) In every state of the above system, at least one event is enabled. That is, the above system does not have a deadlocked state. This observation is proved in the following.

Property 1: In every reachable marking of the MPN (G, M_0), at least one transition is enabled.

Proof

Let us assume to the contrary. Then, there exists a reachable marking M such that none of the transitions t_i ($i = 1, 2, \ldots, 6$) are enabled in M. In particular, t_2, t_4, and t_6 are not enabled. Therefore, we get $M(\mathrm{P}_1) = M(\mathrm{P}_2) = M(\mathrm{P}_3) = 0$. From (5.4)–(5.9), we have

$$M(E_1) + M(F_1) = n_1 , \tag{5.10}$$

$$M(MR_1) = 1 , \tag{5.11}$$

$$M(JR) = 1 , \tag{5.12}$$

$$M(MR_2) = 1 , \tag{5.13}$$

$$M(E_2) + M(F_2) = n_2 , \tag{5.14}$$

$$M(MR_3) = 1 . \tag{5.15}$$

Since t_1 is also not enabled, we get, using (5.11) and (5.12), that $M(E_1) = 0$. Hence (5.10) implies that $M(F_1) = n_1$. This, together with (5.13) will imply that $M(E_2) = 0$ since t_3 is not enabled. Now (5.14) becomes $M(F_2) = n_2$ and by (5.15), t_5 is now enabled. This is a contradiction to our assumption that none of the transitions are enabled in M. This proves the property. □

(iii) The system is live since the MPN (G, M_0) is live. We prove the liveness of (G, M_0) in the following.

Property 2: The marked Petri net (G, M_0) is live.

Proof

Let M be any reachable marking. We have to show that each of t_i ($i = 1, 2, 3, 4, 5, 6$) is live. In this proof, we only show that t_1 and t_2 are live. Identical arguments can be used to prove the liveness of t_3, t_4, t_5 and t_6.

Equation (5.5) implies that $M(P_1) \leqslant 1$. Hence, $M(P_1)$ is either zero or unity. If $M(P_1) = 1$, then t_2 is enabled. If $M(P_1) = 0$, then we will show that we can reach a marking $M^\star$ in which t_1 is enabled. Thus, both t_1 and t_2 are live.

To show that, for $M(P_1) = 0$, we can reach a marking $M^\star$ in which t_1 is enabled, we proceed as follows. Since $M(P_1) = 0$, we have from (5.4), (5.5) and (5.6):

$$M(E_1) + M(F_1) = n_1 , \tag{5.16}$$

$$M(MR_1) = 1 , \tag{5.17}$$

$$M(JR) = 1 . \tag{5.18}$$

If $M(E_1) > 0$, then (5.17) and (5.18) imply that $M = M^\star$. If $M(E_1) = 0$, we can invoke property 1 and argue that a marking $M^\star$ is indeed reachable, for which transition t_1 is enabled. Thus, t_1 and t_2 are live. □

(iv) Blocking and starving of machines are possible in the above system. For example, if we put $M(F_1) = n_1$ in (5.16), we get $M(E_1) = 0$ and so transition t_1 is disabled as buffer B_1 is full and hence machine M_1 is blocked. Similarly, if we put $M(E_1) = n_1$, we get $M(F_1) = 0$ and so transition t_3 is disabled. Now machine M_2 is forced to starve.

(v) Evaluating the properties of G_1, G_2 and G_3 using their invariants provides some clues for designing the system. In the current example, G_1, G_2 and G_3 are all bounded, non-conservative, not live. Their union G is bounded, non-conservative but live. Thus, liveness can be introduced into the system by including additional interactions in the system. The same can not be said of boundedness and conservativeness. If one of G_1, G_2, G_3 were unbounded, G would be surely unbounded. Also, if one of G_1, G_2 and G_3 were non-conservative, then so would G be.

5.2. AN FMS WITH THREE MACHINES AND TWO PART TYPES

Figure 6(a) shows the operation sequence and choice of machines for an FMS with three machines M_1, M_2 and M_3 and two part types J_1 and J_2. We make the following assumptions about the system.

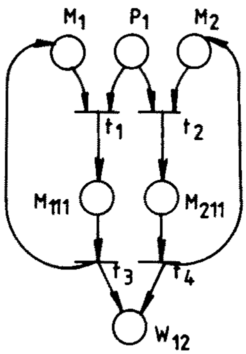
(i) Machine set-up times and transportation times are negligible.
(ii) There are n_1 pallets available for parts of type J_1 and n_2 pallets for parts of type J_2.
(iii) A part is let into the system whenever a pallet of its type is available.
(iv) When a part finds more than one machine free to process it, the machine to process it is chosen non-deterministically.
(v) The initial state of the system is: All pallets free and all machines free.

Let G_1, G_2 and G_3, G_4 be the PNMs shown in figs. 6(b), 6(c), 6(d) and 6(e), respectively, and let G be their union. G_1 and G_2 represent the PNMs, respectively, of the first and second operations on parts of type J_1. G_3 and G_4 correspond to the first and second operations on parts of type J_2. G will then give the PNM of the overall FMS. The interpretation of the places in the above PNMs is as follows ($i = 1, 2, 3$, $j = 1, 2$, $k = 1, 2$):

M_i – machine M_i free,
P_j – pallets of type j available,
W_{jk} – jobs of type j waiting for kth operation,
M_{ikj} – machine M_i performing the kth operation on a part of type j.

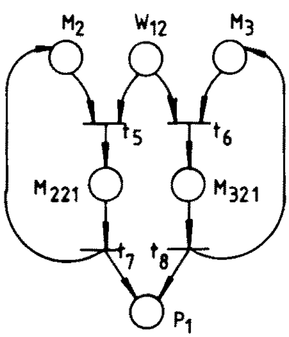
	op#1	op#2
J_1	M_1 / M_2	M_2 / M_3
J_2	M_1 / M_3	$M_1/M_2/M_3$

Fig. 6(a) Part types and operations of the FMS.



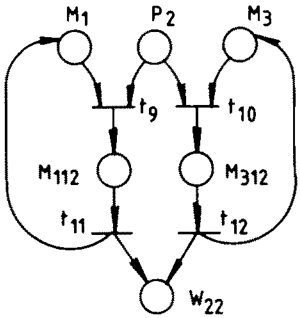
P_1	M_1	M_2	M_{111}	M_{211}	W_{12}
e_1	e_2	e_3	e_1+e_2	e_1+e_3	e_1

Fig. 6(b) PNM and p-invariants of the first operation on parts of type J_1.



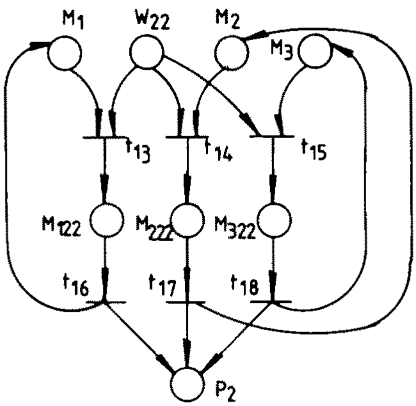
W_{12}	M_2	M_3	M_{221}	M_{321}	P_1
e_1	e_3	e_4	e_1+e_3	e_1+e_4	e_1

Fig. 6(c) PNM and p-invariants for the second operation on parts of type J_1.



P_2	M_1	M_3	M_{112}	M_{312}	W_{22}
e_5	e_2	e_4	e_2+e_5	e_4+e_5	e_5

Fig. 6(d) PNM and p-invariants for the first operation on parts of type J_2.



W_{22}	M_1	M_2	M_3	M_{122}	M_{222}	M_{322}	P_2
e_5	e_2	e_3	e_4	e_2+e_5	e_3+e_5	e_4+e_5	e_5

Fig. 6(e) PNM and p-invariants for the second operation on parts of type J_2.

place	P_1	M_1	M_2	M_{111}	M_{211}	W_{12}	M_3	M_{221}
inv. comp.	e_1	e_2	e_3	e_1+e_2	e_1+e_3	e_1	e_4	e_1+e_3

place	M_{321}	P_2	M_{112}	M_{312}	W_{22}	M_{122}	M_{222}	M_{322}
inv. comp.	e_1+e_4	e_5	e_2+e_5	e_4+e_5	e_5	e_2+e_5	e_3+e_5	e_4+e_5

Fig. 6(f) p-invariants of the overall PNM of the FMS.

Fig. 6. An FMS with 3 machines and 2 part types.

The interpretation of the transitions is simple because each transition flags the commencement or conclusion of a machine operation. Figure 6(f) shows the p-invariants of G, while figs. 6(b)–6(e) display the p-invariants of G_1, G_2, G_3 and G_4, respectively.

We can now make the following observations about the present system.

(i) G_1, G_2, G_3 and G_4 are bounded, non-conservative, and not live. G is bounded, non-conservative, and live. These can be shown using the corresponding p-invariants.

(ii) It can be shown from the p-invariants of G that the place W_{12} is bounded by n_1 and the place W_{22} by n_2. The place W_{12} represents jobs of type J_1 waiting for the second operation. So, it represents the input buffers of M_2 and M_3 and output buffers of M_1 and M_2. Similarly, place W_{22} represents the input buffers of M_1, M_2 and M_3 and output buffers of M_1 and M_3. Using this information, we can find upper bounds for the input and output buffers of the machines. If I_1, I_2 and I_3 and O_1, O_2 and O_3 are these upper bounds for the input and output buffers of M_1, M_2 and M_3 respectively, it can be shown that

$$I_1 = n_2,\ O_1 = n_1 + n_2,\ J_2 = n_1 + n_2,\ O_2 = n_1,\ I_3 = n_1 + n_2$$

$$\text{and } O_3 = n_2.$$

6. Concluding remarks

In this paper, we have shown how Petri net models can be systematically built for FMSs and how they can be analyzed to gain insights into the qualitative behaviour of FMSs. The two illustrative examples presented in sect. 5 embody all the essential principles of our approach. The same principles can be used to model and analyze any given FMS. However, if the FMS is complex, the PNM will be quite huge and analysis will not be easy. One way to overcome this problem is to use *coloured Petri nets* instead of ordinary Petri nets. Coloured Petri nets lead to compact models for even complex FMSs. Kamath [5] has discussed the use of coloured Petri nets in the modelling and simulation of FMSs. We have recently extended theorems 1 and 2 to coloured Petri nets [7].

Deadlocks are an important aspect of FMS behaviour and a future research direction would be to use the Petri net approach to detect possible deadlocks in a given FMS. This paper discusses how Petri net invariants are useful in showing absence of deadlocks in a given FMS. In [8], we consider the question of FMS deadlocks in greater detail and establish the presence of deadlocks in typical manufacturing systems.

Acknowledgements

Our sincere thanks are due to Professor Rajan Suri and two anonymous reviewers of an earlier version of this paper for their valuable suggestions and constructive criticism. Their comments have helped edify many parts of this paper. We also thank our colleague Manjunath Kamath for many active discussions.

References

[1] T. Agerwala, Putting Petri nets to work, IEEE Computer (December 1979) 85.

[2] T. Agerwala and Yong-Chai choed Amphai, A synthesis rule for concurrent systems, *Proc. IEEE Conf. Design Automation* (1978) p. 305.

[3] J.A. Buzacott and D.W. Yao, Flexible manufacturing systems: A review of models, Working Paper No. 82-007 (Dept. of Industrial Engineering, Univerity of Toronto, Canada, 1982).

[4] D. Dubois and K.E. Stecke, Using Petri nets to represent production processes, *Proc. IEEE Conf. Decisions and Control* (December 1983) p. 1062.

[5] M. Kamath, Modelling and simulation of flexible manufacturing systems using coloured Petri nets, M.E. Project Report (School of Automation, I.I.Sc., Bangalore, July 1984).

[6] G. Memmi and G. Roucairol, Linear algebra in net theory, in: *Net Theory and Applications*, ed. W. Brauer, Lecture Notes in Computer Science 84(1980)213.

[7] Y. Narahari and N. Viswanadham, On the invariants of coloured Petri nets, *Proc. 6th European Workshop on Petri Net Theory and Applications*, Espoo, Finland (June 1985) p. 277.

[8] Y. Narahari and N. Viswanadham, A Petri net based investigation of deadlocks in automated manufacturing systems, *Proc. CSI/IFIP Int. Conf. on CAD/CAM/CAE Systems*, Bangalore, India (June 1985) p. 9.1.

[9] J.L. Peterson, *Petri Net Theory and the Modelling of Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1981).

[10] C.V. Ramamoorthy and G.S. Ho, Performace evaluation of asynchronous concurrent systems by Petri nets, IEEE Trans. Software Engineering (September 1980) 440.