EXPLORING STRUCTURAL PROPERTIES OF PETRI NETS IN MATLAB

Mihaela Hanako Matcovschi, Cristian Mahulea and Octavian Păstrăvanu

Department of Automatic Control and Industrial Informatics
Technical University "Gh. Asachi" of Iasi, Blvd. Mangeron 53A, 6600 Iasi, Romania
Phone: +40-32-230751, *Fax*: +40-32-214290
E-Mail: opastrav@delta.ac.tuiasi.ro

Abstract: The standard approach to structural properties of Petri nets (such as boundedness, conservativeness, repetitiveness and consistency) is based on the compatibility of different systems of linear inequalities. Unfortunately the direct usage of this approach is unsuitable for implementation in the ***Petri Net Toolbox*** (recently developed as an analysis and design instrument integrated with MATLAB), because MATLAB offers no facilities for inequality resolution. The paper proposes a technique that allows converting the standard formulation referred to above into a MATLAB tractable problem. This technique exploits the systems of linear inequalities as constraints for an adequate objective function, minimised by a linear programming routine. Both theoretical background and the implementation in the Petri Net Toolbox are presented. Two examples illustrate the effectiveness of the proposed technique as well as its usefulness for computer-based training.

Keywords: Petri nets, Structural properties, Discrete event systems, Linear inequalities, MATLAB, Computer-based training.

1. INTRODUCTION

The successful study of *Petri nets* (PNs) *by Control Engineering* (CE) students is highly dependent on the organization of computer experiments for laboratory classes and, consequently, requires specific software tools to deal with the complexity of analysis and design problems. During the last decade, many academic or research groups developed PN simulators equipped with various supplementary facilities for efficient approaches to PN properties (see, for instance, (Feldbrugge, 1993) and the brief list given in (Păstrăvanu, 1997, pp. 195)). Unfortunately, such software platforms are not familiar for CE students, whose regular practical training focuses on the exploitation of the generous resources provided by MATLAB and its specialized toolboxes.

Under these circumstances, a *Petri Net Toolbox* (***PN Toolbox***), embedded in the MATLAB environment, was designed and implemented at the Department of Automatic Control and Industrial Informatics of the Technical University "Gh. Asachi" of Iasi, whose skeleton and functionality were briefly presented in (Mahulea *et al*, 2001). The integration with the MATLAB philosophy presents the considerable advantage (with respect to other PN software) of creating powerful algebraic, statistical and graphical instruments, which exploit the high quality routines available in MATLAB.

The MATLAB orientation of the ***PN Toolbox*** also ensures the necessary flexibility for further improvement, by upgrading the tools that already exist and by adding new ones. Along these lines, new facilities have been recently developed for the analysis of the structural properties and it is the purpose of the current paper to give an overview of their theoretical background and software implementation. The main idea consists in deriving appropriate formulations for the analysis of the structural properties so as to allow using the computational resources offered by MATLAB.

The text is organised according to the following plan: Section 2 briefly presents the fundamental results that are used by the standard approach to structural properties, based on systems of linear inequalities. Section 3 exposes a technique for testing the compatibility of such systems of linear inequalities, which is suitable to MATLAB capabilities. The implementation of this technique in the ***PN Toolbox*** is discussed in Section 4. Section 5 considers two relevant examples to demonstrate the validity of the proposed methodology for analysing the structural properties, as well as to illustrate its exploitation as an instrument incorporated into the ***PN Toolbox***. Some concluding remarks are formulated in the last section.

2. STANDARD APPROACH TO STRUCTURAL PROPERTIES BASED ON LINEAR INEQUALITIES

A Petri net consists of a particular type of directed weighted bipartite graph, denoted by ***N***, and an initial state called *initial marking*, $\mathbf{M}_0$. The underlying graph ***N*** of a PN contains two kinds of nodes, called *places* and *transitions*. Its weighted arcs connect a place to a transition or a transition to a place. It is assumed that all PNs considered in this paper are *pure*, that is they have no self-loops. The topological structure of a pure Petri net ***N*** with *n* transitions and *m* places is completely described by an $n \times m$ matrix of integers $\mathbf{A}$, called *incidence matrix*. A *marking* (state) $\mathbf{M}$ assigns to each position *p* a nonnegative integer $\mathbf{M}(p)$ (*p* is marked with $\mathbf{M}(p)$ tokens). The firing of an enabled transition will change the token distribution in a net. A marking $\mathbf{M}$ is said to be *reachable* from $\mathbf{M}_0$ if there exists a sequence of firings that transforms $\mathbf{M}_0$ into $\mathbf{M}$.

Structural properties of PNs are those that depend only on their topological structure and are independent of the initial marking. Thus, these properties may be characterised in terms of the incidence matrix $\mathbf{A}$ and its associated homogenous equations or inequalities. In (Murata, 1989) necessary and sufficient conditions for structural boundedness, conservativeness, repetitiveness and consistency of a PN are provided. A brief overview is presented in the sequel.

Throughout this paper, $\mathbb{Z}^m$ ($\mathbb{R}^m$) and $\mathbb{Z}^{n\times m}$, for $n, m \in \mathbb{N}^*$, denote the set of integer (real) vectors with *m* elements, respectively that of $n \times m$ integer matrices. The null vector is denoted by $\mathbf{0}$. For a given matrix $\mathbf{A} \in \mathbb{Z}^{n\times m}$, $\mathbf{A}^T$ stands for its transposed matrix. Let $\mathbf{x}, \mathbf{y} \in \mathbb{Z}^m$, be two *m*-vectors, $\mathbf{x} = [x_1, x_2, \ldots, x_m]^T$, $\mathbf{y} = [y_1, y_2, \ldots, y_m]^T$. The following inequality type notations are used:

$$\mathbf{x} < \mathbf{y} \iff x_i < y_i,\ \forall i = \overline{1,m};$$

$$\mathbf{x} \le \mathbf{y} \iff x_i \le y_i,\ \forall i = \overline{1,m};$$

$$\mathbf{x} <\neq \mathbf{y} \iff \mathbf{x} \le \mathbf{y} \text{ and } \exists i \in \{1,2,\ldots,m\},\ x_i \ne y_i.$$

Structural Boundedness. A Petri net ***N*** with an initial marking $\mathbf{M}_0$ is said to be *bounded* if the number of tokens in each place does not exceed a finite number $k \in \mathbb{N}$ for any marking reachable from $\mathbf{M}_0$. A PN that is bounded for any finite initial marking $\mathbf{M}_0$ is said to be *structurally bounded.*

Theorem 1. A Petri net ***N*** is structurally bounded iff there exists an *m*-vector of positive integers, $\mathbf{y} \in \mathbb{Z}^m$, $\mathbf{y} > \mathbf{0}$, such that

$$\mathbf{A}\mathbf{y} \le \mathbf{0}. \tag{1}$$

Proof. See (Murata, 1989). ■

Conservativeness. A Petri net ***N*** is said to be *conservative* if there exists an *m*-vector of positive integers, $\mathbf{y} \in \mathbb{Z}^m$, $\mathbf{y} > \mathbf{0}$, such that for any initial marking $\mathbf{M}_0$ and for every marking $\mathbf{M}$ reachable from $\mathbf{M}_0$ the following identity holds

$$\mathbf{M}^T\mathbf{y} = \mathbf{M}_0^T\mathbf{y} = \text{a constant.} \tag{2}$$

In case that equality (2) holds for an *m*-vector of nonnegative integers $\mathbf{y} \in \mathbb{Z}^m$, $\mathbf{y} >\neq \mathbf{0}$, then the net is said to be *partially conservative*.

Theorem 2. A Petri net ***N*** is (partially) conservative iff there exists an *m*-vector of integers, $\mathbf{y} \in \mathbb{Z}^m$, $\mathbf{y} > \mathbf{0}$ ($\mathbf{y} >\neq \mathbf{0}$), such that

$$\mathbf{A}\mathbf{y} = \mathbf{0}. \tag{2}$$

Proof. See (Murata, 1989). ■

Repetitiveness. A Petri net ***N*** is said to be *repetitive* if there exists an initial marking $\mathbf{M}_0$ and a firing sequence σ such that every transition occurs infinitely often in σ. In case that there exists an initial marking $\mathbf{M}_0$ and a firing sequence σ such that some transitions (not all) occur infinitely often in σ, the net is said to be *partially repetitive*.

Theorem 3. A Petri net ***N*** is (partially) repetitive iff there exists an *n*-vector of integers, $\mathbf{x} \in \mathbb{Z}^n$, $\mathbf{x} > \mathbf{0}$ ($\mathbf{x} >\neq \mathbf{0}$), such that

$$\mathbf{A}^T\mathbf{x} \ge \mathbf{0}. \tag{3}$$

Proof. See (Murata, 1989). ■

Consistency. A Petri net ***N*** is said to be *consistent* if there exists an initial marking $\mathbf{M}_0$ and a firing sequence σ from $\mathbf{M}_0$ back to $\mathbf{M}_0$ such that every transition occurs at least once in σ. In case that there exists an initial marking $\mathbf{M}_0$ and a firing sequence σ from $\mathbf{M}_0$ back to $\mathbf{M}_0$ such that some transitions (not all) occur at least once in σ, the net is said to be *partially consistent*.

Theorem 4. A Petri net ***N*** is (partially) consistent iff there exists an *n*-vector of integers, $\mathbf{x} \in \mathbb{Z}^n$, $\mathbf{x} > \mathbf{0}$ ($\mathbf{x} >\neq \mathbf{0}$), such that

$$\mathbf{A}^T\mathbf{x} = \mathbf{0}. \qquad (4)$$

Proof. See (Murata, 1989). ■

P- and T-Invariants. An *m*-vector $\mathbf{y} >\neq \mathbf{0}$ of integers is called a *P-invariant* if $\mathbf{Ay} = \mathbf{0}$. An *n*-vector $\mathbf{x} >\neq \mathbf{0}$ of integers is called a *T-invariant* if $\mathbf{A}^T\mathbf{x} = \mathbf{0}$. There is, obviously, a direct connection between the P-invariants and the conservativeness of a PN on the one side, and between the T-invariants and the consistency on the other side.

The set of places (transitions) corresponding to nonzero entries in a P-invariant **y** (T-invariant **x**) is called the *support* of the invariant. A support is said to be *minimal* if no proper nonempty subset of the support is also a support of an invariant. An invariant **z** is said to be *minimal* if there is no other invariant $\mathbf{z}^1$ (of the same type) such that $\mathbf{z} >\neq \mathbf{z}^1$. A minimal invariant with minimal support is called a *basic* invariant. Any invariant may be written as a linear combination of basic invariants.

A Petri net ***N*** is said to be *covered* with P-invariants (T-invariants) if all its positions (transitions) belong to the support of a basic invariant. If a PN is covered with P-invariants, then it is conservative. If a PN is covered with T-invariants, then it is consistent.

3. CONVERTING STANDARD APPROACH TO A MATLAB TRACTABLE PROBLEM

As presented in the previous section, the study of the structural properties of PNs leads to the study on the compatibility of some systems of linear inequalities. Since MATLAB provides no function for the direct study of such systems, there is the need of another approach that exploits MATLAB capabilities.

Let $\mathcal{K}$ be the positive hiperoctant of $\mathbb{R}^m$, $\mathcal{K} = \left\{\mathbf{y} \in \mathbb{R}^m \middle| \mathbf{y} \geq \mathbf{0}\right\}$, whose interior and boundary are denoted by $\mathcal{K}_i = \left\{\mathbf{y} \in \mathcal{K} \middle| \mathbf{y} > \mathbf{0}\right\}$, respectively by $\mathcal{K}_b = \left\{\mathbf{y} \in \mathcal{K} \middle| \exists i \in \{1,2,\ldots,m\},\, y_i = \mathbf{0}\right\}$.
The study of the structural properties of PNs may be reduced to deciding whether a homogenous system of linear equations or inequalities has any solution in $\mathcal{K}_i$ or all its solutions are included in $\mathcal{K}_b$, as stated by the following theorems.

Theorem 5. Let $\mathbf{A}$ be an $n \times m$ integer matrix. If the system of linear inequalities

$$\mathbf{Ay} \leq \mathbf{0},\ \mathbf{y} \in \mathbb{R}^m, \qquad (5)$$

has a solution $\mathbf{y}_p$ in $\mathcal{K}_i$, then there is an *m*-vector of integers, $\mathbf{y}_z \in \mathbb{Z}^m \cap \mathcal{K}_i$, that satisfies (5).
Proof. Two distinct cases may be studied: solution $\mathbf{y}_p$ strictly satisfies all the inequalities in (5), $\mathbf{Ay}_p < \mathbf{0}$, or at least one of the *m* elements of vector $\mathbf{Ay}_p$ is equal to zero. In both cases, due to the particular form of matrix $\mathbf{A}$, in any neighbourhood of $\mathbf{y}_p$ there can be found a vector $\mathbf{y}_q \in \mathcal{K}_i$, having as elements rational numbers, which satisfies (5). By writing all the elements of $\mathbf{y}_q$ in simplified fraction form and choosing α as the least common multiple of their denominators, the vector $\mathbf{y}_z = \alpha\mathbf{y}_q$ is a vector of integers and it is also a solution to (5) that belongs to $\mathcal{K}_i$. ■

Theorem 6. If for a given $n \times m$ integer matrix $\mathbf{A}$ the system of linear inequalities (5) has a nontrivial solution $\mathbf{y}_p \in \mathcal{K}_b$, then there is an *m*-vector of integers, $\mathbf{y}_z \in \mathbb{Z}^m \cap \mathcal{K}_b$, $\mathbf{y}_z \neq \mathbf{0}$, that satisfies (5).
Proof. The demonstration to this theorem is similar to the previous one. A solution to (5), $\mathbf{y}_q \in \mathcal{K}_b$, having as elements rational numbers, can be found in any neighbourhood of $\mathbf{y}_p$. Based on $\mathbf{y}_q$, there can be constructed an *m*-vector of integers, $\mathbf{y}_z \in \mathbb{Z}^m \cap \mathcal{K}_b$, $\mathbf{y}_z \neq \mathbf{0}$, that also satisfies (5). ■

The same way, for a homogenous system of linear equations the following results may be demonstrated.

Theorem 7. If $\mathbf{A} \in \mathbb{Z}^{n\times m}$ and the system of linear equations

$$\mathbf{Ay} = \mathbf{0},\ \mathbf{y} \in \mathbb{R}^m, \qquad (6)$$

has a solution $\mathbf{y}_p$ in $\mathcal{K}_i$, then there is an *m*-vector of integers, $\mathbf{y}_z \in \mathbb{Z}^m \cap \mathcal{K}_i$, that satisfies (6).
Proof. The proof to this theorem is the counterpart of the one to theorem 5. ■

Theorem 8. If the system of linear equations (6) has a nontrivial solution in $\mathcal{K}_b$, then there is an *m*-vector of integers, $\mathbf{y}_z \in \mathbb{Z}^m \cap \mathcal{K}_b$, $\mathbf{y}_z \neq \mathbf{0}$, that satisfies system (6).
Proof. This theorem may be proven likewise theorem 6. ■

Further on, the decision on the solutions to a linear system of inequalities (or equations) may be settled on by considering the linear programming problem (LPP):

$$\min_{\mathbf{y}} \mathbf{f}^T\mathbf{y} \quad \text{such that} \qquad (7)$$

$$\begin{aligned} \mathbf{Ay} &\leq \mathbf{0}, \\ \mathbf{y} &\geq \mathbf{0}, \end{aligned} \qquad (8)$$

respectively

$$\begin{aligned} \mathbf{Ay} &= \mathbf{0}, \\ \mathbf{y} &\geq \mathbf{0}, \end{aligned} \qquad (9)$$

where $\mathbf{f} = \left[-1,-1,\ldots,-1\right]^T$. The minimisation problem (7) under constraint (8) (or (9)) has always a solution. The position of this solution in $\mathcal{K}_i$ or in $\mathcal{K}_b$ specifies in fact whether the linear system (8) (respectively (9)) has any solution in $\mathcal{K}_i$ or all its solutions are included in $\mathcal{K}_b$.

The MATLAB implementation of this approach utilises the function **linprog** that solves the LPP

$$\min_{\mathbf{x}} \mathbf{f}^T\mathbf{x} \quad \text{such that} \tag{10}$$

$$\begin{aligned} &\mathbf{A}_{in}\mathbf{x} \le \mathbf{b}_{in}, \\ &\mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq}, \\ &\mathbf{l} \le \mathbf{x} \le \mathbf{u}, \end{aligned} \tag{11}$$

where $\mathbf{x}, \mathbf{f}, \mathbf{b}_{in}, \mathbf{b}_{eq}, \mathbf{l}, \mathbf{u}$ are vectors and $\mathbf{A}_{in}, \mathbf{A}_{eq}$ are matrices of appropriate dimensions.

The systems of linear equations or inequalities involved in the characterisation theorems of the structural properties of PNs may be regarded as constraints of such an LPP. The third relation in (11) is used to define the positive hiperoctant. The lower bound, **l**, is set to **0**. The existence of the built-in MATLAB function **Inf**, which returns the IEEE arithmetic representation for positive infinity, allows the elements of vector **x** to be superiorly unbounded by choosing all the elements of the upper bound vector, **u**, equal to **Inf**. This way, the numerical optimisation algorithm will evenly try to move off zero all the elements of **x**, according to the maximum number of iterations allowed by the call to **linprog** function.

The conclusion on the position of the solutions to the systems of linear equations or inequalities is finally achieved by comparing to 1 all the elements of the solution vector provided by the **linprog** function call. In case that the structural boundedness of a PN is under study, the elements of the solution vector that are smaller then 1 correspond to the unbounded positions in the net. The same way, in case of conservativeness (consistency) the elements smaller then 1 of the solution vector correspond to the positions (transitions) that do not belong to any P-invariant (T-invariant). Four new MATLAB functions implement this approach to the study of the structural properties of PNs and have been successfully integrated into ***PN Toolbox***.

4. IMPLEMENTATION AS AN INSTRUMENT AVAILABLE IN PN TOOLBOX

The ***PN Toolbox*** (Mahulea *et al*, 2001) was designed to offer specific instruments for the simulation, analysis and synthesis of discrete event systems modelled by PNs. Its embedding in the MATLAB environment presents considerable advantages with respect to other PN software by exploiting the high quality routines available in MATLAB.

In the present version of ***PN Toolbox*** *untimed*, *transition-timed* and *place-timed* PN models are accepted. The timed nets can be *deterministic* or *stochastic*. Priorities or probabilities can be assigned to conflicting transitions. Since MATLAB includes the built-in function **Inf**, our toolbox is able to operate with nets having *infinite capacity* places, unlike other PN software where places are meant as having finite capacity. User interaction with PN graphs is allowed by an easy to exploit ***Graphical User Interface*** (GUI) (MathWorks Inc., 1997a, b) presented in figure 1.

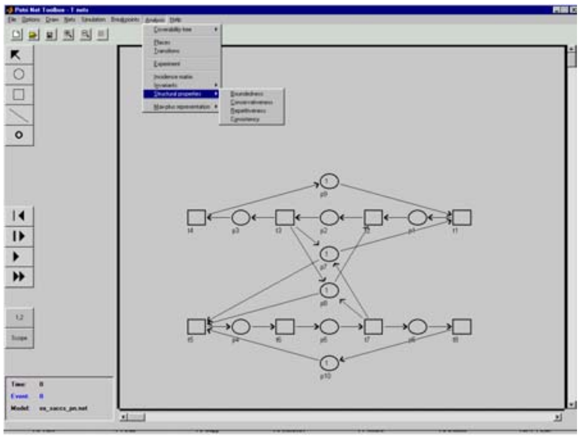


Figure 1. Screen capture of the main window of ***PN Toolbox***

The ***Menu bar*** displays a set of eight drop-down menus that offer facilities for file handling, provides tools for graphical editing of PNs in the ***Drawing area*** and gives the user the possibility to control the simulation progress. In addition, the ***Analysis menu*** makes available some computational tools for investigating both *behavioral* and *structural* properties, generating *P-* or *T-invariants*, calculating global performance indices for timed nets, constructing *max-plus* state-space descriptions for marked-graph topologies etc.

In particular, the ***Structural properties*** submenu contains four pop-up menus (***Boundedness***, ***Conservativeness***, ***Repetitiveness*** and ***Consistency***) corresponding to the structural properties under discussion in this paper, based on the incidence matrix of a given graphical model. The results of the analysis are displayed in related dialogue windows. In case that the studied PN is not structurally bounded, the unbounded places are displayed. Also, for partially conservative (consistent) PNs in the related window there are displayed the places (transitions) that do not belong to any P-invariant (respectively, T-invariant) support (see figure 5).

5. ILLUSTRATIVE EXAMPLES

A large number of FMSs modeled by PNs has been investigated in order to assess the facilities developed in this toolbox. For these tests to be conclusive, the chosen examples had different levels of complexity.

The current paper presents two examples illustrating different situations that may occur.

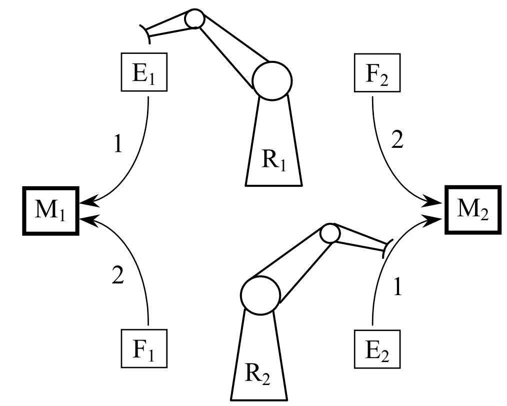


Figure 2. Sketch of the FMS used in Example 1

Example 1 refers to the flexible manufacturing system (FMS) sketched in figure 2. It consists of two robots (R_1 and R_2) used to place two types of parts, (E_1 and E_2), on two types of plates (F_1 and F_2). The assembling of part (E_i) on plate (F_i) is carried out on machine (M_i) ($i=\overline{1,2}$). First, the right side robot is uses to place the part on the corresponding machine; then, the left side robot is used to transport the plate on the same machine. Next, the robots are released and the assembling is started. After this operation is completed, the machine is also released.

The synthesis of the corresponding PN model, presented in figure 1, has been made according to the *hybrid synthesis* proposed in (Zhou and DiCesare, 1993) in order to avoid the apparition of dead-lock. The net has $n=8$ transitions and $m=10$ places. Its incidence matrix, computed directly from the graphical model, is

$$\mathbf{A}=\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & -1 & 0 \\ -1 & 1 & 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -1 & -1 & 0 & -1 \\ 0 & 0 & 0 & -1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \end{bmatrix}. \quad (12)$$

The procedure for generating *P-* or *T-invariants* starts by calling the MATLAB function **null**, to determine a basis of integer vectors for the null space of the incidence matrix (case of *P-invariants*), or that of the transposed incidence matrix (case of *T-invariants*). Linear combinations constructed with these vectors provide invariants (not necessarily minimal or with minimal support).

The net has four basic P-invariants (figure 3) denoted below by $\mathbf{y}^k$, $k=\overline{1,4}$, as well as two basic T-invariants, $\mathbf{x}^i$, $i=\overline{1,2}$,

$$\begin{aligned} \mathbf{y}^1 &= [\,1,1,0,1,1,0,1,0,0,0]^T, \\ \mathbf{y}^2 &= [\,0,1,0,1,1,0,0,1,0,0]^T, \\ \mathbf{y}^3 &= [\,1,1,1,0,0,0,0,0,1,0]^T, \\ \mathbf{y}^4 &= [\,0,0,0,1,1,1,0,0,0,1]^T, \end{aligned} \quad (13)$$

$$\begin{aligned} \mathbf{x}^1 &= [\,1,1,1,1,0,0,0,0]^T, \\ \mathbf{x}^2 &= [\,0,0,0,0,1,1,1,1]^T. \end{aligned} \quad (14)$$

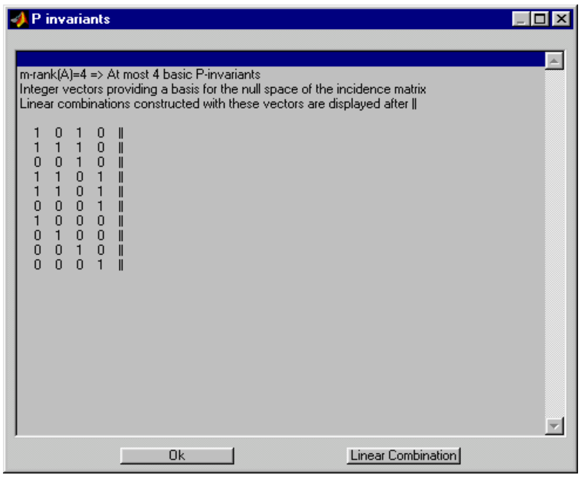


Figure 3. Screen capture of the ***P-Invariants*** window

It can be noticed that the net is covered by P-invariants, therefore it is structurally bounded and conservative. Since it is also covered by T-invariants, the net is repetitive and consistent as well. The same results are obtained by calling the four newly implemented functions dedicated to the study of structural properties of PNs.

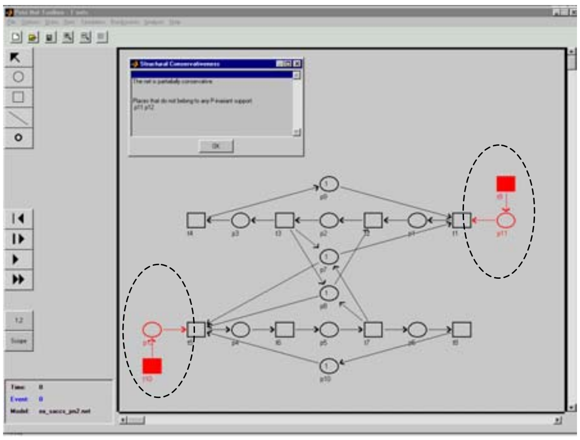


Figure 4. The PN model used in Example 2

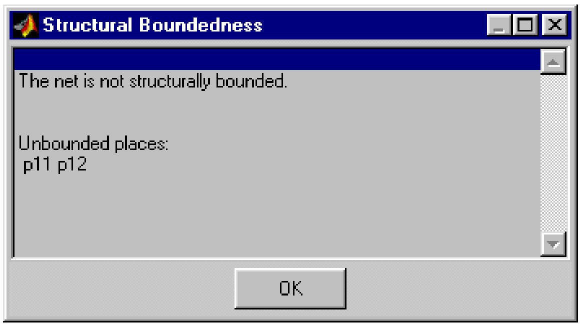
Example 2 is obtained by adding to the previous model two transitions (denoted by t9 and t10) and two places (denoted by p11 and p12), circled in figure 4. Their physical meaning is that of storage spaces for the pieces to be worked on the machines. The incidence matrix of the PN model has 10 rows and 12 columns. The P-invariants are given by

$$\mathbf{y}^1 = [\ 1,\ 1,\ 0,\ 1,\ 1,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0]^T,$$
$$\mathbf{y}^2 = [\ 0,\ 1,\ 0,\ 1,\ 1,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0,\ 0]^T,$$
$$\mathbf{y}^3 = [\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0,\ 0]^T,\qquad (15)$$
$$\mathbf{y}^4 = [\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 1,\ 0,\ 0]^T,$$

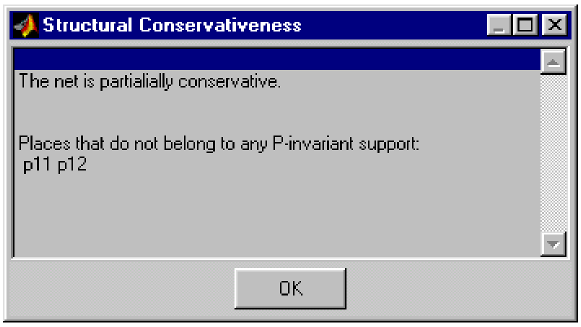
and the T-invariants are

$$\mathbf{x}^1 = [\ 1,\ 1,\ 1,\ 1,\ 0,\ 0,\ 0,\ 0,\ 1,\ 0]^T,\qquad (16)$$
$$\mathbf{x}^2 = [\ 0,\ 0,\ 0,\ 0,\ 1,\ 1,\ 1,\ 1,\ 0,\ 1]^T.$$

This net is not structurally bounded; also, it is partially conservative. The newly added positions are unbounded and do not belong to any P-invariant support (figure 5). The net is repetitive and consistent. That results also from its being covered by T-invariants.



a.



b.

Figure 5. Screen capture of the ***Structural Boundedness*** (a) and ***Structural Conservativeness*** (b) windows associated with the PN model in figure 4

6. CONCLUSIONS

The standard approach to boundedness, conservativeness, repetitiveness and consistency of Petri nets, which relies on the compatibility of systems of linear inequalities, has been converted into a MATLAB tractable problem. This allowed the MATLAB implementation of novel software modules devoted to the analysis of structural properties, which have been successfully incorporated into the ***PN Toolbox***. The newly created instruments are able to test the compatibility of systems of linear inequalities, although the MATLAB environment does not offer proper functions for computing the solutions to such systems.

Two PN models have been considered as illustrative examples to demonstrate the validity of the proposed methodology for analysing the structural properties, as well as to show how ***PN Toolbox*** can be exploited for such complex tasks.

7. REFERENCES

Feldbrugge, F. (1993). Petri net tool overview In: *Advances in Petri Nets* (G. Rozenberg, Ed.), pp.169-209. Springer-Verlag, Berlin Heidelberg.

Mahulea, C., Bârsan, L. and Păstrăvanu, O. (2001). MATLAB tools for Petri-net-based approaches to flexible manufacturing systems. In: *Prep. of the 9th IFAC Symposium on Large Scale Systems LSS 2001* (Filip, F.G., Dumitrache, I. and Iliescu, S.S., Eds.), pp.184-189, Bucharest.

Murata, T. (1989). Petri nets: properties, analysis and application. In: *Proc. of the IEEE*, **77**, pp.541-580.

Păstrăvanu, O. (1997). *Discrete Event System - Qualitative Techniques in a Petri Net Framework*, MatrixRom, Bucharest (in Romanian).

The MathWorks, Inc. (1997a). *Building GUIs with MATLAB*. Natick, Massachusetts.

The MathWorks, Inc. (1997b). *Using MATLAB Graphics*. Natick, Massachusetts.

Zhou, M.C. and DiCesare, F. (1993). *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*. Kluwer Academic Publisher, Norwell, Massachusetts.

8. ABREVIATIONS

CE – Control Engineering
FMS – Flexible Manufacturing System
GUI – Graphical User Interface
LPP – Linear Programming Problem
PN – Petri Net