MATLAB TOOLS FOR PETRI-NET-BASED APPROACHES TO FLEXIBLE MANUFACTURING SYSTEMS

Cristian Mahulea, Laura Barsan and Octavian Pastravanu

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 skeleton and the functionality of a Petri Net Toolbox, embedded in the MATLAB environment, are briefly presented, as offering a collection of instruments devoted to simulation, analysis and synthesis of discrete-event systems. The integration with the MATLAB philosophy responds to the general interest manifested by educators for enlarging the compatibility between the traditional background of Control Engineering students and the novelty of Discrete-Event-Systems scenarios. Two amply commented case studies illustrate the usage of some facilities available in this new toolbox and prove its capability to cover a wide range of teaching goals for the Petri-net-based approaches to Flexible Manufacturing Systems. *Copyright © IFAC 2001*

Keywords: Discrete event systems, Petri nets, Manufacturing systems, Large-scale systems, MATLAB, Educational aids.

1. INTRODUCTION

During the last decade, many universities offering education programs in Control Engineering (CE) included, in their curricula, courses on Flexible Manufacturing Systems (FMSs), as a consequence of the large impact on research and technology proved by the tremendous development of this field. A well-known trend in approaching FMSs relies on Petri-net (PN) models (see, for instance, (Desrochers and Al-Jaar, 1993; Lewis *et al*., 1988; Lewis *et al*., 1995; Zhou and DiCesare, 1993)), and the organization of computer experiments for laboratory classes requires specific software tools to deal with the complexity of analysis and design problems. However, despite the diversity of PN simulators developed by many research groups (see, for instance, (Feldbrugge, 1993) and the brief list given in (Pastravanu, 1997, pp. 195)), 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. It is the purpose of our work to present the skeleton and the functionality of a ***PN Toolbox*** integrated with the MATLAB philosophy, whose facilities are able to cover a wide range of topics in teaching FMSs. At the conceptual level, the paper reflects the general interest manifested by CE educators for enlarging the compatibility between the novelty of FMS scenarios and the traditional background and programming skills of CE students.

PN Toolbox was designed to offer specific instruments for simulation, analysis and synthesis of discrete event systems. Its embedding in the MATLAB environment 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. Moreover, this MATLAB orientation of the ***PN Toolbox*** was intended to permit further development, including tools devoted to hybrid systems, because MATLAB incorporates comprehensive software for studying continuous and discontinuous dynamics.

The paper is organized according to the following plan. Section 2 creates an overview of the main tools developed for handling PNs within the software framework specific to MATLAB applications. Programming techniques and algorithms used for implementation are briefly commented in Section 3. Section 4 illustrates the usage of the PN Toolbox for approaching several analysis and design problems issued by the study of two FMSs. Some concluding remarks are formulated in Section 5.

2. DESCRIPTION AND FACILITIES OF PN TOOLBOX

In the present version of ***PN Toolbox***, three types of PN models are accepted, namely *untimed*, *transition-timed* and *place-timed* nets. The timed nets can be *deterministic* or *stochastic*, and the stochastic case allows using all the distribution functions available in MATLAB Statistics Toolbox. Priorities or probabilities can be assigned to conflicting transitions. Unlike other PN software where places are meant as having finite capacity (because of the arithmetic representation used by the computational environment), our toolbox is able to operate with *infinite capacity*, since MATLAB includes the built-in function **Inf**, which returns the IEEE arithmetic representation for positive infinity. The user may also select the speed of simulation, which determines how long the simulator will pause before making the next set of transition firings. Thus, an animation effect can be obtained that creates a relevant picture for the overall dynamics of the PN governed by the transition firing rule. A reset feature allows the user to quickly return the design to its initial marking.

User interaction with PN graphs is allowed by an easy to exploit ***Graphical User Interface*** (GUI), (MathWorks Inc., 1997a, b) whose purpose is two-fold. First, it gives the user the possibility to draw Petri nets in a natural fashion, to store, retrieve and resize (by Zoom-In and Zoom-Out features) such drawings. Second, it permits the simulation and analysis of the Petri nets, by exploiting all the computational resources of the environment, via the global variables stored in the MATLAB's Workspace. This GUI consists of a MATLAB figure window, exhibiting a ***Menu bar***, four control panels: (*i*) ***File Management panel***, (*ii*) ***Drawing panel***, (*iii*) ***Simulation panel*** (*iv*) ***Status panel*** and a ***Drawing area***. Figure 1 presents a hard-copy of the main window opened by ***PN Toolbox***, where all the component parts of the GUI (that have been enumerated above) are visible.

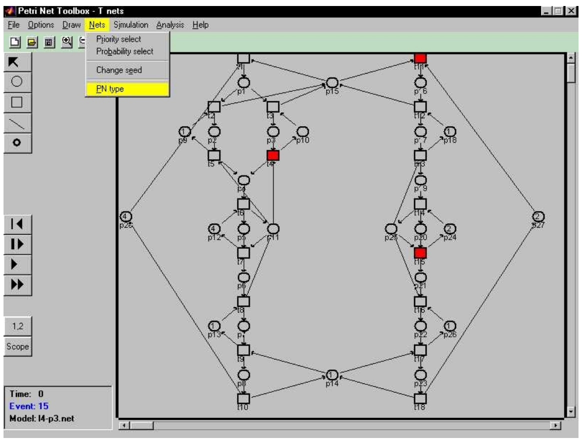


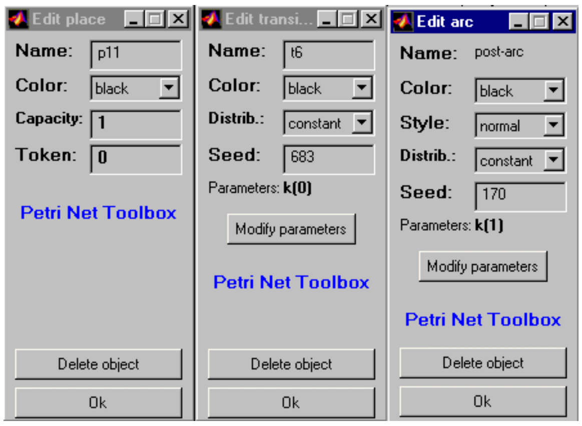
Fig. 1. Hard-copy of the main window of ***PN Toolbox***

The ***Menu bar*** (placed horizontally, on the top of the window in fig. 1) displays a set of seven drop-down menus at the top of the window, where the user can select all the facilities available in the ***PN Toolbox***. These menus are ***File***, ***Options***, ***Draw***, ***Nets***, ***Simulation***, ***Analysis***, and ***Help***. The ***File menu*** (containing the pop-up menus ***New***, ***Open***, ***Save*** and ***Exit***) offers facilities for file-handling and zoom operations. The ***Options menu*** (containing the pop-up menus ***Color***, ***Show grid***, ***Events - Run fast***, ***Delay – Run slow*** and ***View Arc Weight***) allows choosing specific conditions for visualization and simulation. The ***Draw menu*** (containing the pop-up menus ***Place draw***, ***Transition draw***, ***Arc draw*** and ***Token select***) provides tools for graphical editing (graph nodes, arcs, tokens, labels) in the ***Drawing area***. The ***Nets menu*** (containing the pop-up menus ***Priority select***, ***Probability select***, ***Change seed*** and ***PN type***) permits specifying additional information for the transition firing rule to be used in running the simulation. The ***Simulation menu*** (containing the pop-up menus ***Event***, ***Run slow***, ***Run fast***, ***Reset***, ***Marking color*** and ***Log File***) gives the user the possibility to control the simulation progress and to record the results. The ***Analysis menu*** (containing the pop-up menus ***Coverability tree***, ***Places***, ***Transitions***, ***Parameterized experiments***, ***Incidence matrix***, ***P-Invariants***, ***T-Invariants***, ***Max-plus***) provides computational tools for studying various aspects encountered in the PN dynamics. Thus, instruments have been developed to investigate both *behavioral and structural properties*, to generate *P-* or *T-invariants*, to calculate global performance indices for timed nets, to construct *max-plus* state-space descriptions for marked-graph topologies etc.

Buttons belonging to control panels (*i*)-(*iii*) duplicate part of the functionality available in the ***Menu bar***, aiming to ensure a more convenient access to the services which are most frequently requested.

The ***File Management panel*** (placed, as an horizontal bar, just below the ***Menu bar***) presents a number of image buttons, whose actions are identical to those controllable by the ***File menu***. The ***Drawing***

panel (placed vertically, in the left side of the window in fig. 1, just below the ***File Management panel***) presents a number of image buttons, whose actions are identical to those controllable by the ***Draw menu***. In addition, the ***Edit button*** of this panel permits a read/write access to the characteristics of the net nodes and arcs, by opening a dialog box associated with the object selected in the ***Drawing area***. These characteristics are: capacity, marking, timing information and labeling – for places; timing information and labeling – for transitions; weight and existence of inhibitory function – for arcs. Figure 2 reproduces the dialog boxes associated with the three aforementioned elements, in the case of a *T-timed* PN. The option ***Color*** (that appears in all these boxes, regardless of the nature of the element) creates the possibility of assigning different colors to the elements introduced in different phases of a multi-step design procedure.



Edit place
Name: p11
Color: black
Capacity: 1
Token: 0

Edit transi...
Name: t6
Color: black
Distrib.: constant
Seed: 683
Parameters: k(0)

Edit arc
Name: post-arc
Color: black
Style: normal
Distrib.: constant
Seed: 170
Parameters: k(1)

- a - - b - - c -

Fig. 2. Hard-copies of the dialog boxes controlled by ***Edit Button*** in ***Simulation panel***, associated with places (a), transitions (b) and arcs (c), in the case of a *T-timed* PN

The ***Simulation panel*** (placed vertically, in the left side of the window in fig. 1, just below the ***Drawing panel***) presents a number of image buttons, whose actions are identical to those controllable by the ***Simulation menu*** and provides supplementary facilities for visualizing the time-depending evolution of various *performance indices*.

The ***Status panel*** is a message board (placed in the bottom left corner of the window in fig. 1), where the ***PN Toolbox*** displays the current values for the simulation time and for the number of events.

The ***Drawing area*** (located in the central and right side of the window in fig. 1), is provided with a grid, where the nodes of the Petri net graph are to be placed and two scrollbars (on the right and bottom sides) to move the desired parts of the graph into view.

3. PROGRAMMING TECHNIQUES, ALGORITHMS AND IMPLEMENTATION

The development of GUI exploited the general philosophy of *object-oriented programming* and, in particular, the MATLAB's *object-oriented graphics system,* called Handle Graphics (MathWorks Inc., 1997a, b). This technique offers maximum flexibility to the codes and allows simple interventions for updating or further expanding of the toolbox. For controlling the properties of the objects located within the ***Drawing area***, special methods have been implemented, that rely on appropriate calls of the standard functions **get** and **set** of the Handle Graphics.

All the procedures available in ***PN Toolbox*** were implemented as **m** files (which can run in the MATLAB window) and they were designed in a modular fashion, especially to get started from the GUI. To ensure storing/retrieving services, four distinct types of *files* are used in the present version of ***PN Toolbox***, namely for the PN model, for the coverability tree associated with a model, for the complete history of a simulation experiment, and for the final values of the performance indices corresponding to a given simulation experiment.

The algorithms used by ***PN Toolbox*** in simulation and analysis have been designed to permit different implementations for the transition firing rules, in accordance with the specificity of the net: finite or infinite capacity, untimed, *T-timed* or *P-timed*. The progress of simulation is driven by an asynchronous clock corresponding to the occurrence of events (e.g. (Cassandras, 1993)). In the untimed case, the sequencing of the events is reduced to simply ordering their occurrence, without any temporal significance, unlike the timed case when simulation requires a continuous correlation with physical time (e.g. (Cassandras, 1993; David and Alla, 1992)). In simulation, the random number generator existing in MATLAB is used for several purposes (depending on the type of the PN) and, therefore, distinct values are requested for its seed. These values are automatically set by the PN Toolbox, or they can be chosen by the user, and their preservation from one simulation experiment to another means the repeatability of the simulated dynamics.

The construction of the coverability tree (as the basic element for the analysis of the *behavioral properties*) follows the lines of the algorithm given in (Murata, 1989). The analysis of the *structural properties* exploits necessary and sufficient conditions based on the compatibility of different systems of linear inequalities, built with the incidence matrix, e.g. (Murata, 1989). For implementation, the MATLAB function **lp** is used, and the systems of linear inequalities are regarded as constraints.

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 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 existence of the built-in function **Inf** allowed the development of new routines for matrix addition and multiplication in the semiring ($\mathbf{R}\cup\{-\infty\}$,max,+), representing the background for operating with *max/plus state-space models* (e.g. (Cohen *et al.*,1985; Cassandras, 1993; Desrochers and Al-Jaar, 1993)).

The visualization instrument called ***Scope*** (available in the Simulation panel) opens a supplementary *figure window*, which relies on the plotting functions of MATLAB (MathWorks Inc., 1997b) and serves to monitor the time evolution of selected *performance indices* (such as *service distance*, *idle time*, *service time*, *activity* – for transitions, and *arrival distance*, *passage distance*, *queue length* – for positions, etc).

4. CASE STUDIES FOR FMS ANALYSIS AND DESIGN

The facilities developed in this toolbox have been tested for studying a large number of FMSs modeled by PNs, most of these examples being chosen from literature so as to exhibit a high level of complexity. In the current paper, two illustrative examples are considered, each of them placing emphasis on different topics of FMS analysis.

Example 1 refers to the automated manufacturing system sketched in fig. 3 (Zhou and DiCesare, 1993, pp. 122), whose PN model is given in fig. 1.

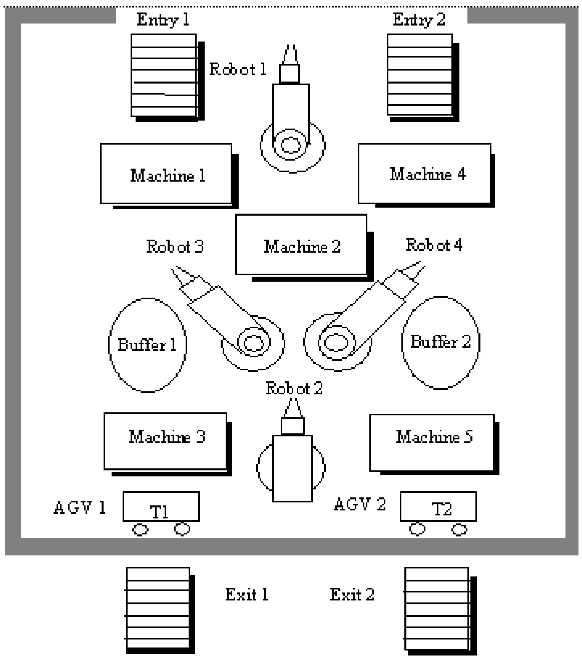
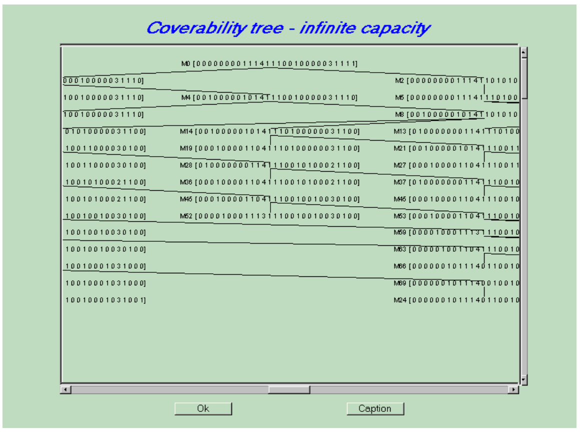


Fig. 3. Sketch of the FMS used in Example 1

If this PN is gradually built according to the *hybrid synthesis* proposed in (Zhou and DiCesare, 1993), the user can take advantage of the editing facilities provided by ***PN Toolbox***, in order to assign different colors to the PN elements added in different steps of the *bottom-up design* (as shown in fig. 1).

The net is initially meant as *untimed*, and the ***PN Toolbox*** is used to study the *behavioral properties* within the context of a variable number of pallets available for transportation. The coverability tree is automatically built, by selecting option ***Coverability tree / Graphics*** from the ***Analysis menu***, which yields the image reproduced in fig. 4. By repeating this construction for different number of pallets, one can see that the network is bounded, reversible and life if this number is less than 5 for the flow Entry 1- Exit 1 in fig. 3 (modeled in the left side of the PN given in fig. 1), and less than 4 for the flow Entry 2 – Exit 2 in fig. 3 (modeled in the right side of the PN given in fig. 1). Whenever the number of pallets used for transportation in one or both flows exceeds the values mentioned above, liveness and reversibility are lost, since a partial or total deadlock occurs (corresponding to a circular blocking of some resources – e.g. (Pastravanu, 1997; Lewis *et al.*, 1998)). These results of the analysis can be confirmed by simulation experiments with the ***Run fast*** option, if a sufficiently large number of events (i.e. thousands) is taken.

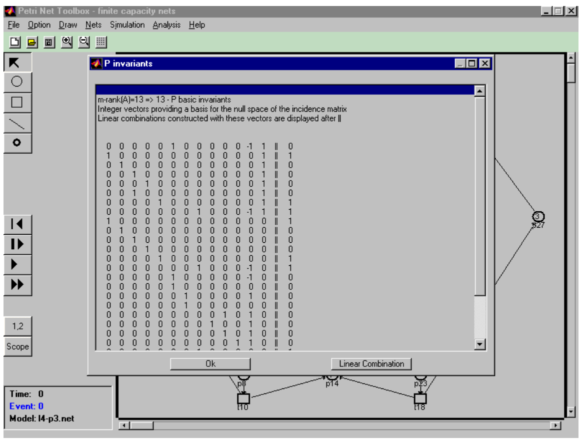


Coverability tree - infinite capacity
M0 [0 0 0 0 0 0 0 0 1 1 1 4 1 1 1 0 0 1 0 0 0 0 0 3 1 1 1 1]
M4 [0 0 1 0 0 0 0 0 1 0 1 4 1 1 1 0 0 1 0 0 0 0 0 3 1 1 1 0]
M14 [0 0 1 0 0 0 0 0 1 0 1 4 1 1 1 0 1 0 0 0 0 0 0 3 1 1 0 0]
M19 [0 0 0 1 0 0 0 0 1 1 0 4 1 1 1 0 1 0 0 0 0 0 0 3 1 1 0 0]
M28 [0 1 0 0 0 0 0 0 0 1 1 4 1 1 1 0 0 1 0 1 0 0 0 2 1 1 0 0]
M36 [0 0 0 1 0 0 0 0 1 1 0 4 1 1 1 0 0 1 0 1 0 0 0 2 1 1 0 0]
M45 [0 0 0 1 0 0 0 0 1 1 0 4 1 1 1 0 0 1 0 0 1 0 0 3 0 1 0 0]
M52 [0 0 0 0 1 0 0 0 1 1 1 3 1 1 1 0 0 1 0 0 1 0 0 3 0 1 0 0]
0 0 0 1 0 0 0 0 0 3 1 1 1 0]
1 0 0 1 0 0 0 0 0 3 1 1 1 0]
1 0 0 1 0 0 0 0 0 3 1 1 1 0]
0 1 0 1 0 0 0 0 0 3 1 1 0 0]
1 0 0 1 1 0 0 0 0 3 0 1 0 0]
1 0 0 1 1 0 0 0 0 3 0 1 0 0]
1 0 0 1 0 1 0 0 0 2 1 1 0 0]
1 0 0 1 0 1 0 0 0 2 1 1 0 0]
1 0 0 1 0 0 1 0 0 3 0 1 0 0]
1 0 0 1 0 0 1 0 0 3 0 1 0 0]
1 0 0 1 0 0 1 0 0 3 0 1 0 0]
1 0 0 1 0 0 0 1 0 3 1 0 0 0]
1 0 0 1 0 0 0 1 0 3 1 0 0 0]
1 0 0 1 0 0 0 1 0 3 1 0 0 1]
M2 [0 0 0 0 0 0 0 0 1 1 1 4 1 1 0 1 0 1 0
M5 [0 0 0 0 0 0 0 0 1 1 1 4 1 1 1 0 1 0 0
M8 [0 0 1 0 0 0 0 0 1 0 1 4 1 1 0 1 0 1 0
M13 [0 1 0 0 0 0 0 0 0 1 1 4 1 1 1 0 1 0 0
M21 [0 0 1 0 0 0 0 0 1 0 1 4 1 1 1 0 0 1 1
M27 [0 0 0 1 0 0 0 0 1 1 0 4 1 1 1 0 0 1 1
M37 [0 1 0 0 0 0 0 0 0 1 1 4 1 1 1 0 0 1 0
M46 [0 0 0 1 0 0 0 0 1 1 0 4 1 1 1 0 0 1 0
M53 [0 0 0 1 0 0 0 0 1 1 0 4 1 1 1 0 0 1 0
M59 [0 0 0 0 1 0 0 0 1 1 1 3 1 1 1 0 0 1 0
M63 [0 0 0 0 0 1 0 0 1 1 0 4 1 1 1 0 0 1 0
M66 [0 0 0 0 0 0 1 0 1 1 1 4 0 1 1 0 0 1 0
M69 [0 0 0 0 0 0 1 0 1 1 1 4 0 0 1 0 0 1 0
M24 [0 0 0 0 0 0 1 0 1 1 1 4 0 1 1 0 0 1 0
Ok
Caption

Fig. 4. Hard-copy of the ***Coverability tree*** window associated with the PN model in fig. 1

Within the window ***P-Invariants*** reproduced in fig. 5 (accessible from the ***Analysis menu***), user can automatically build integer vectors belonging to the null space of the incidence matrix, which represent invariants (not necessarily basic *P-invariants*), whenever all their components are non-negative. These invariants create a deeper insight into the internal mechanisms governing the allocation of non-shared and shared resources.

In a second stage, a *timed version* of the initial PN can be considered, to undertake various simulation experiments, able to reveal the timing effects upon the occurrence of *circular blocking of resources*, when the number of pallets used for transportation is inadequately chosen. Moreover, this timed PN allows a detailed analysis of efficiency in the FMS exploitation, based on different *performance indices*, computed during simulation and displayed by the visualization instrument ***Scope***. Both deterministic and stochastic cases present a great deal of interest for such studies. As an illustration, fig. 6 presents a hard-copy of the ***Scope*** window used with the option ***Service Distance*** to observe the time duration necessary for processing one part on the flow Entry 1 – Entry 2 in fig. 3 (modeled in the left side of the PN given in fig. 1), when the observation lasts about 200 minutes.


Petri Net Toolbox - finite capacity nets

P invariants

```
m-rank(A)=13 => 13 - P basic invariants
Integer vectors providing a basis for the null space of the incidence matrix
Linear combinations constructed with these vectors are displayed after ||

 0  0  0  0  0  1  0  0  0  0  0 -1  1  ||  0
 1  0  0  0  0  0  0  0  0  0  0  0  1  ||  1
 0  1  0  0  0  0  0  0  0  0  0  0  1  ||  0
 0  0  1  0  0  0  0  0  0  0  0  0  1  ||  0
 0  0  0  1  0  0  0  0  0  0  0  0  1  ||  0
 0  0  1  0  0  0  0  0  0  0  0  0  1  ||  0
 0  0  0  0  1  0  0  0  0  0  0  0  1  ||  1
 0  0  0  0  0  0  0  1  0  0  0 -1  1  ||  1
 1  0  0  0  0  0  0  0  0  0  0  0  0  ||  1
 0  1  0  0  0  0  0  0  0  0  0  0  0  ||  0
 0  0  1  0  0  0  0  0  0  0  0  0  0  ||  0
 0  0  0  1  0  0  0  0  0  0  0  0  0  ||  0
 0  0  0  0  1  0  0  0  0  0  0  0  0  ||  1
 0  0  0  0  0  0  0  1  0  0  0 -1  0  ||  1
 0  0  0  0  0  1  0  0  0  0  0 -1  0  ||  0
 0  0  0  0  0  1  0  0  0  0  0  0  0  ||  0
 0  0  0  0  0  0  1  0  0  0  0  1  0  ||  0
 0  0  0  0  0  0  1  0  0  0  0  0  0  ||  0
 0  0  0  0  0  0  0  0  0  1  0  1  0  ||  0
 0  0  0  0  0  0  0  0  1  0  0  1  0  ||  0
 0  0  0  0  0  0  0  0  0  1  0  1  0  ||  0
 0  0  0  0  0  0  0  0  0  0  1  1  0  ||  0
```

Fig. 5. Hard-copy of the ***P-Invariants*** window associated with the PN model in fig 1

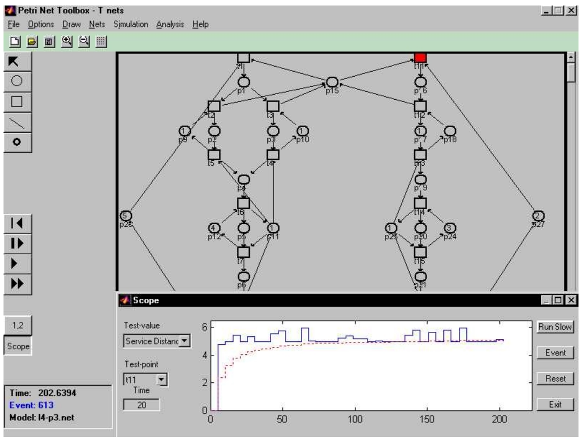
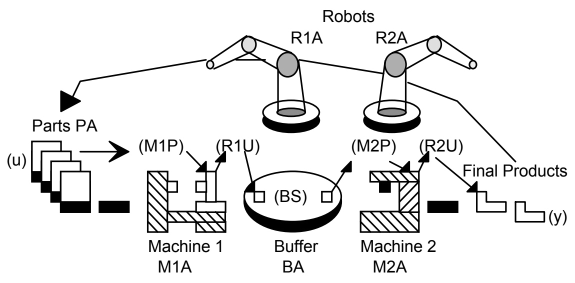


Fig. 6. Hard-copy of the ***Scope*** window

The simulation experiment considers a uniformly distributed duration in the release of the pallets (between 1 and 3 minutes), and deterministic durations (not detailed here, for sake of simplicity) allocated to the utilization and release of machines and robots. At the end of the simulation for the time horizon mentioned above, the selection of ***Transition / Service distance*** in the ***Analysis menu*** provides the average time of 5.14 minutes which is consumed to complete the processing of one part on the flow under discussion.

Example 2 refers to the flexible manufacturing workcell depicted in fig. 7 (Lewis *et al.*, 1995, pp. 262). which is modeled by the deterministic *P-timed* marked graph, shown in fig. 8.



	M1A	M2A	R1A	R2A	BA	PA
Processing time [min]	10	11	5	4	0	0
Setup time [min]	3	3	2	2	1	3

Fig. 7. Sketch of the workcell used in Example 2

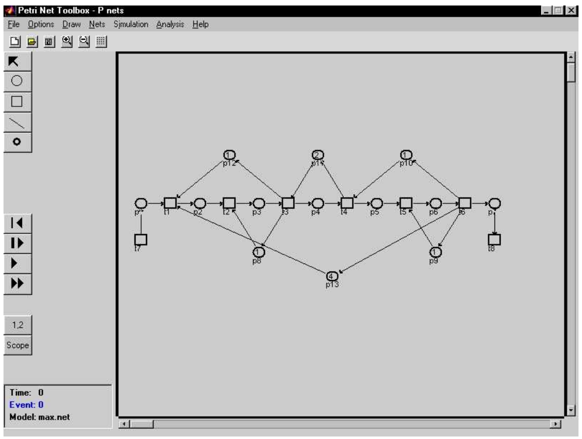

Petri Net Toolbox - P nets
File Options Draw Nets Simulation Analysis Help
Scope
Time: 0
Event: 0
Model: max.net

Fig. 8. Hard-copy of the main window displaying the PN model of the FMS sketched in fig. 7

Figure 9 presents the results of a ***parameterized experiment*** called from ***Analysis menu***, which plots the surface corresponding to service/time for transition t_8 in fig. 8, when the duration associated to places p_2 and p_6 varies from 5 to 20 minutes and from 10 to 30 minutes, respectively.

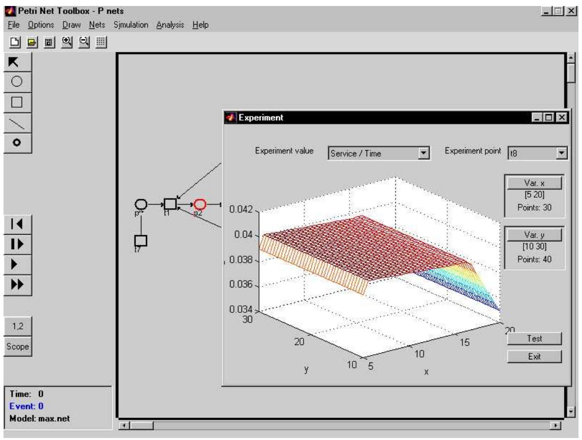


Fig. 9. Hard copy of the results corresponding to the ***Parameterized experiments***

For this net a *max/plus state-space representation* is constructed. Four pallets are considered for transportation. The notations ⊕ and ⊗ are used for '**max**' and '**+**' operations, respectively, yielding the equations:

$$\boldsymbol{x}(k) = \boldsymbol{A}_0 \otimes \boldsymbol{x}(k) \oplus \boldsymbol{A}_1 \otimes \boldsymbol{x}(k-1) \oplus \boldsymbol{A}_2 \otimes \boldsymbol{x}(k-2) \oplus \boldsymbol{A}_3 \otimes \boldsymbol{x}(k-3) \oplus \boldsymbol{A}_4 \otimes \boldsymbol{x}(k-4) \oplus \boldsymbol{B} \otimes u(k),$$

$$y(k) = \boldsymbol{C} \otimes \boldsymbol{x}(k),$$

where the state space vector $\boldsymbol{x}(k)$ collects the instants when the transitions of the PN model fire for the k-th time:

$$\boldsymbol{x}(k) = \left[x_k^1 \; x_k^2 \; x_k^3 \; x_k^3 \; x_k^5 \; x_k^6\right]^T,$$

the input $u(k)$ denotes the instant when the k-th part enters the workcell, and the output $y(k)$ denotes the time when the k-th part leaves the workcell. The entries of the matrices used in this state-space description are defined as follows:

$(\boldsymbol{A}_0)_{ij}=\varepsilon$ for all i,j=1,...,6, except for $(\boldsymbol{A}_0)_{21}=t_{\mathrm{M1P}}$, $(\boldsymbol{A}_0)_{32}=t_{\mathrm{R1U}}$, $(\boldsymbol{A}_0)_{43}=t_{\mathrm{BS}}$, $(\boldsymbol{A}_0)_{54}=t_{\mathrm{M2P}}$, $(\boldsymbol{A}_0)_{65}=t_{\mathrm{R2U}}$;

$(\boldsymbol{A}_1)_{ij}=\varepsilon$ for all i,j=1,...,6, except for $(\boldsymbol{A}_1)_{13}=t_{\mathrm{M1A}}$, $(\boldsymbol{A}_1)_{23}=t_{\mathrm{R1A}}$, $(\boldsymbol{A}_1)_{46}=t_{\mathrm{M2A}}$, $(\boldsymbol{A}_1)_{56}=t_{\mathrm{R2A}}$;

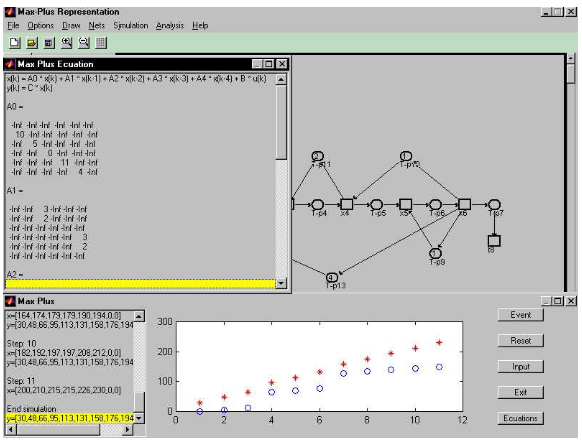
$(\boldsymbol{A}_2)_{ij}=\varepsilon$ for all i,j=1,...,6, except for $(\boldsymbol{A}_2)_{33}=t_{\mathrm{BA}}$;

$(\boldsymbol{A}_3)_{ij}=\varepsilon$ for all i,j=1,...,6;

$(\boldsymbol{A}_4)_{ij}=\varepsilon$ for all i,j=1,...,6, except for $(\boldsymbol{A}_4)_{16}=t_{\mathrm{PA}}$;

$$\boldsymbol{B}=[0 \; \varepsilon \; \varepsilon \; \varepsilon \; \varepsilon \; \varepsilon]^T; \; \boldsymbol{C}=[\varepsilon \; \varepsilon \; \varepsilon \; \varepsilon \; \varepsilon \; 0].$$

Using this representation, comprehensive tests can be performed to outline the influence of the duration corresponding to different activities, with a view to the optimization of machine utilization. Figure 10 presents the two windows pertaining to the *max-plus analysis* as follows: (*i*) a window containing global information – placed in the lower part of the screen, and (*ii*) a window displaying the max-plus equations – placed in the upper part of the screen. The visualization instrument available in window (*i*) plots the results of the max-plus simulation of the workcell operation for the activity times given in the lower part of fig. 7, using a non-uniform sequence of arrival times of the input parts feeding the FMS (0, 6, 11, 65, 70, 77, 128, 135, 139, 143, 150 minutes). The time intervals of starvation are simply detected from the graphical plots in fig 10, where the symbols 'o' and '*' are associated with the system input and output, respectively



```
x(k) = A0 * x(k) + A1 * x(k-1) + A2 * x(k-2) + A3 * x(k-3) + A4 * x(k-4) + B * u(k)
y(k) = C * x(k)

A0 =

 -Inf  -Inf -Inf  -Inf  -Inf -Inf
   10  -Inf -Inf  -Inf  -Inf  -Inf
 -Inf     5  -Inf -Inf  -Inf  -Inf
 -Inf  -Inf     0  -Inf  -Inf -Inf
 -Inf  -Inf  -Inf    11  -Inf -Inf
 -Inf  -Inf  -Inf  -Inf     4  -Inf

A1 =

 -Inf -Inf     3 -Inf -Inf -Inf
 -Inf -Inf     2 -Inf -Inf -Inf
 -Inf -Inf -Inf -Inf -Inf -Inf
 -Inf -Inf -Inf -Inf -Inf     3
 -Inf -Inf -Inf -Inf -Inf     2
 -Inf -Inf -Inf -Inf -Inf -Inf

A2 =
```

```
x=[164,174,179,179,190,194,0,0]
y=[30,48,66,95,113,131,158,176,194

Step: 10
x=[182,192,197,197,208,212,0,0]
y=[30,48,66,95,113,131,158,176,194

Step: 11
x=[200,210,215,215,226,230,0,0]

End simulation
y=[30,48,66,95,113,131,158,176,194
```

Fig. 10. Hard copy of the windows pertaining to the ***Max-plus*** analysis

5. CONCLUSIONS

The MATLAB tools newly created for exploring dynamics and properties of PNs represent valuable educational aids in organizing laboratory sessions devoted to the analysis and design of FMSs with high level of complexity. The considered case studies amply illustrate the role played by PN techniques, embedded in the powerful software environment offered by MATLAB, in approaching the performance evaluation for various structures of FMSs.

ABBREVIATION AND NOMENCLATURE

CE	– Control Engineering
FMS	– Flexible Manufacturing System
GUI	– Graphical User Interface
PN	– Petri Net

REFERENCES

Cassandras, C.G. (1993). *Discrete Event Systems: Modeling and Performance Analysis*, Irwin, Boston.

Cohen G.D., Dubois, J.P. Quadrat, and M. Viot (1985). A linear-system-theoretic view of discrete-event processes and its use for performance evaluation in manufacturing. *IEEE Trans. Automat. Control,* **30**, 210-220.

David, R. et Alla, H. (1992). *Du Grafcet aux réseaux de Petri* (2e édition), Hermes, Paris.

Desrocheres, A.A. and Al-Jaar, R.Y. (1993*). Modeling and Control of Automated Manufacturing Systems*. IEEE Computer Society Press, Rensselaer, Troy, New-York.

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

Lewis, F.L., Huang, H.H., Pastravanu, O. and Gűrel, A. (1995). Control systems design for flexible manufacturing systems In: *Flexible Manufacturing Systems: Recent Developments* (A. Raouf, and M. Ben-Daya, Eds.). pp. 253-290. Elsevier Science, Oxford.

Lewis, F.L., Gurel, A., Bogdan, S., Doganalp, A. and Pastravanu, O. (1998). Analysis of deadlock and circular waits using a matrix model for flexible manufacturing systems, *Automatica*, **34**, 1083-1100.

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

Pastravanu, 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.