

PROCEEDINGS

CSCS-14

14th INTERNATIONAL CONFERENCE
ON CONTROL SYSTEMS
AND COMPUTER SCIENCE

2-5 July 2003

1
VOL

editura
POLITEHNICA
PRESS



POLITEHNICA University of Bucharest



MODELING, SIMULATION AND ANALYSIS OF PETRI NETS IN MATLAB

Mihaela-Hanako Matcovschi, Cristian Mahulea, Octavian Pastravanu

*Department of Automatic Control and Industrial Informatics
Technical University "Gh. Asachi" of Iasi, Blvd. Mangeron 53A, Iasi, 6600, Romania
Tel./Fax: +40-232-230.751, E-Mail: {mhanako, cmahulea, opastrav}@delta.ac.tuiasi.ro*

Abstract: Despite the large offer of software products available for MATLAB, none of its toolboxes provides instruments able to handle Petri net models. It is the purpose of our paper to present several types of problems formulated for such models, which are approached in MATLAB. The treatment does not use the primary resources of the MATLAB environment, but exploits the facilities of a toolbox with a fully coherent structure. Thus, the user does not need to develop codes and his/her attention can integrally focus on modeling, simulation, analysis and design.

Keywords: Petri nets, modeling and simulation, flexible manufacturing cells, discrete event systems, education and training.

1. INTRODUCTION

Despite the numerous toolboxes and facilities offered by MATLAB, as an environment specialized in scientific computation, no instrument is available for dealing with Discrete Event Systems (DES) modeled by Petri nets (PNs) in general, except for two packages focusing on restricted areas of interest (Svádová and Hanzálek, 2001) and (Iordache and Antsaklis, 2002). Within the mentioned context, the initiative of developing instruments for simulation, analysis and design of PNs under MATLAB brought remarkable benefits for training and research because Control Engineering people are familiar with the exploitation of *Graphical User Interfaces* (GUIs) based on this popular software. Although a recent list of the programs developed for PNs includes many resources (Mortensen, 2003) running under different operating systems, our initiative was successful due to the large preference shown for MATLAB.

Therefore, the work invested in producing version 1 of the *Petri Net Toolbox* (Mahulea *et al.*, 2001) was

enthusiastically continued at the Department of Automatic Control and Industrial Informatics of the Technical University "Gh. Asachi", yielding a more reliable and complex product able to cover various types of applications. It is worth separately mentioning that the overall design and implementation philosophy that sustains the *PN Toolbox*, as well as the integration with MATLAB, allow further developments in the modern direction of studying hybrid dynamics involving both DES and ODE models.

The *PN Toolbox* served during the last two academic years as a support for laboratory experiments accompanying the courses on *Discrete Event Dynamical Systems* and *Queueing Systems* given at the Technical University "Gh. Asachi". Its testing as a teaching aid used by a large number of students permitted a careful brush up of the inherent software bugs. Moreover, this toolbox was used for practical demonstrations at the University of Sheffield during an ERASMUS / SOCRATES mobility (Mahulea, 2002).

This paper places emphasis on the diversity of instruments newly created in MATLAB for studying DES based on PN models. Therefore, Section 2 gives only general information about the toolbox and advises the interested reader how to get a complete description of this software package. Instead, Section 3 provides numerous details regarding its concrete usage for addressing typical problems in DES simulation, analysis and design. The discussion in this section is organized so as to cover the most relevant topics specific to different classes of PNs. Some concluding remarks on the effectiveness of the presented software are formulated in Section 4.

2. BRIEF OVERVIEW OF THE SOFTWARE

Three main objectives were envisaged by the organization of the *PN Toolbox*: (i) creation of powerful graphical instruments to support the intuitive understanding of the model usage, (ii) implementation of efficient algorithms for simulation, analysis and synthesis and (iii) organization of a set of demonstrative examples showing the usage of the software accompanied by a comprehensive online help.

The GUI gives the possibility to draw PNs in a natural fashion and allows a straightforward access to various commands starting adequate procedures for exploiting the PN models. A description of the role played by the GUI and of the key algorithms used by the *PN Toolbox* can be found in (Matcovschi *et al.*, 2003). However, part of the facilities as well as the basis of the computational approaches are referred to and commented in the next section, concomitantly with their illustration by examples.

To ensure a quick understanding of the exploitation for all the available facilities we have developed a web site devoted to this software (Mahulea *et al.*, 2003). There, the interested user can run several demonstrative movies presenting specific sequences of operations in handling the GUI and interpreting the results of the commands. For details, one can read the adequate instructions supplied by the user's guide.

Compared to the previous version of the *PN Toolbox*, version 2 brings the following noticeable improvements: • a redesigned GUI; • a reorganization of the existing menus and the introduction of some new commands; • a redesigned structure for the MATLAB object defining the generic PN model; • an extension to stochastic and generalized stochastic PNs of the analysis and simulation procedures; • upgraded algorithms for simulation in the Run Fast mode, analysis of the structural properties, analysis and simulation of max-plus models; • new facilities for parameter-based design; • usage of an xml file for model saving; • usage of an html file for saving the numerical values of the global performance indices resulting from a simulation experiment; • extensive help support and animated demos.

3. FACILITIES AND DEMOS

There are several possibilities for analyzing discrete event systems modeled by PNs, which are briefly presented in the sequel depending on the type of the model.

3.1. Untimed Models

The *behavioral properties* of untimed PNs (e.g. boundedness, liveness, reversibility, etc.) may be approached through the *coverability tree* automatically constructed by the *PN Toolbox* based on the topology and initial marking of the net. For unbounded nets, the coverability tree is built applying the ω -convention, which means the usage of a generic symbol (herein denoted by " ω ") for referring to unbounded markings (Murata, 1989).

The *structural properties* of untimed PNs (e.g. structural boundedness, repetitiveness, consistency and conservativeness) are approached as integer programming problems formulated in terms of the *incidence matrix* of the net (Matscovschi *et al.*, 2001). Also, the algorithm for computing the minimal support P- and T-invariants presented in (Martinez and Silva, 1982) is implemented; these invariants may be displayed, on request, in separate windows.

Example 1. For illustrating these facilities, the flexible manufacturing system (FMS) sketched in fig. 1 was selected. It consists of two robots (R_1 and R_2) used to place two types of parts, (A_1 and A_2), on two types of plates (B_1 and B_2). The assembling of part (A_i) on plate (B_i) is carried out on machine (M_i) ($i = \overline{1,2}$) as follows. First, the right side robot is used to place the part on the machine; then, the left side robot is used to transport the plate on the same machine. After the assembling is completed, the machine and the robots are released.

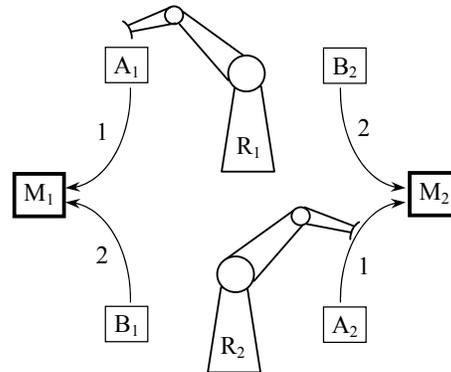


Fig. 1. Sketch of the FMS used in Example 1.

The untimed PN model of this FMS is presented in fig. 2. The places of the net are labeled according to their physical meaning. The initial marking corresponds to the situation when all the resources are idle.

The analysis of the behavioral properties of the PN model starts with consulting its coverability tree (fig. 3). The dead marking $M_4 = [1, 0, 1, 0, 0, 0, 0, 0]$ matches

up the situation when robot R_1 is used to place part A_1 on M_1 the same time when robot R_2 is used to place part A_2 on M_2 ; consequently, none of the machines can complete its task.

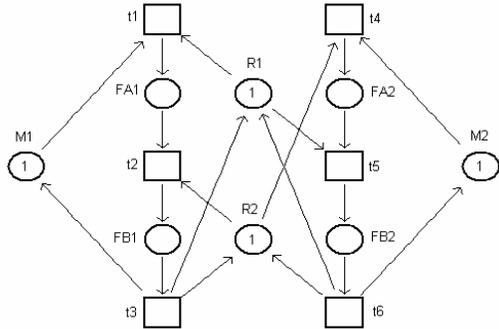


Fig. 2. The untimed PN model of the FMS used in Example 1.

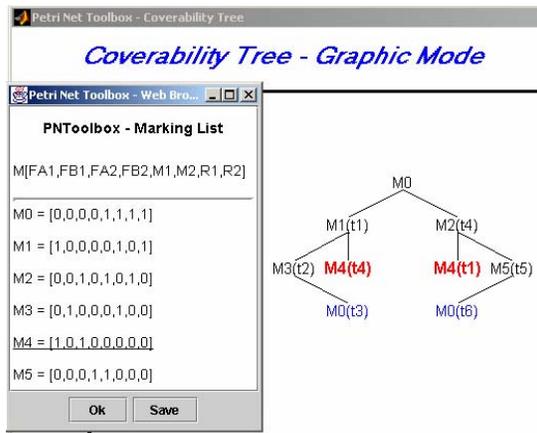


Fig. 3. The coverability tree of the net in fig. 2.

In order to ensure the mutual parallel exclusion of the shared resources (R_1 and R_2) and avoid deadlock, a lookahead feedback controller (Lewis *et al.*, 1995) must be used. The resulting PN model is presented in fig. 4 and its incidence matrix is displayed in fig. 5.

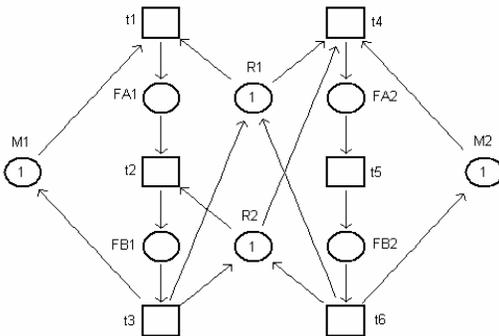


Fig. 4. The PN model of the FMS with lookahead feedback.

For the analysis of the structural properties of the new net in fig. 4, the minimal support P- and T-invariants can be consulted (fig. 6).

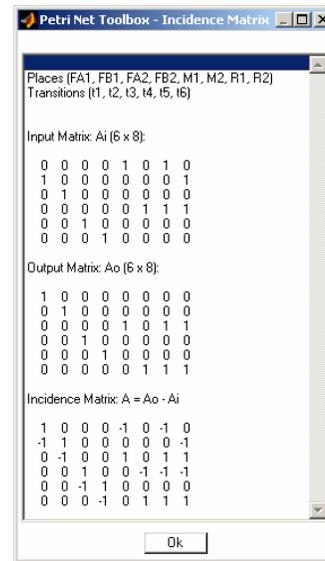
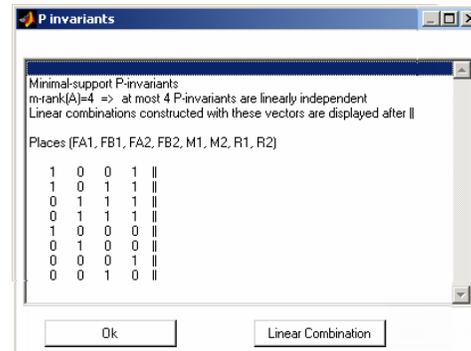
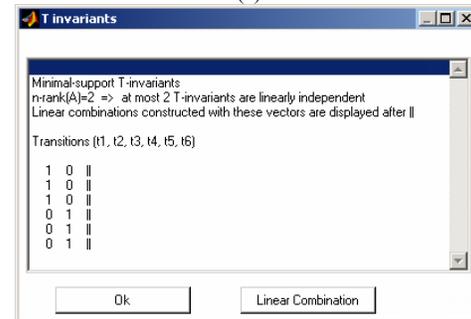


Fig. 5. The incidence matrix of the PN in fig. 4.

It can be noticed that this net is covered both by P- and T-invariants, therefore it is structurally bounded, repetitive, conservative and consistent (fig. 7).



(a)



(b)

Fig. 6. The minimal support (a) P- and (b) T-invariants of the PN in fig. 4.

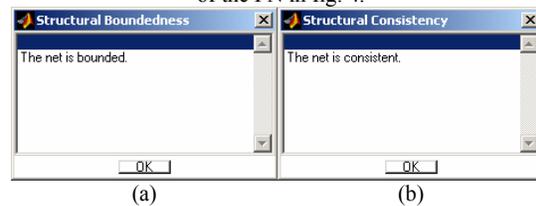


Fig. 7. The results for the analysis of (a) structural boundedness and (b) consistency of the PN in fig. 4.

3.2. Timed Models

In the *PN Toolbox*, both place timed and transition timed models are accepted. The timing may be deterministic and/or stochastic. Besides running a simulation and analyzing the results (facilities that are presented in Section 3.3), the minimum cycle time for the steady state dynamics of a (partially) consistent PN covered by P-invariants (Murata, 1989), (David and Alla, 1992) may be displayed.

For a deterministically place-timed event graph a max-plus simulation can be also performed. The max-plus state-space representation (Bacelli *et al.*, 1992) in implicit form

$$\begin{aligned} \mathbf{x}(k) &= \bigoplus_{i=0}^M [A_i \otimes \mathbf{x}(k-i) \oplus B_i \otimes \mathbf{u}(k-i)], \\ \mathbf{y}(k) &= \bigoplus_{i=0}^M [C_i \otimes \mathbf{x}(k-i) \oplus D_i \otimes \mathbf{u}(k-i)], \end{aligned} \quad (1)$$

for $k = \overline{1, N}$, is derived by the *PN Toolbox* based on the topology and initial marking of the net. The numerical values of the matrices involved in these equations may be displayed in a separate window.

The max-plus simulation utilizes properly chosen initial conditions for the explicit form of the state equation

$$\mathbf{x}(k) = A_0^* \left[\bigoplus_{i=1}^M A_i \mathbf{x}(k-i) \oplus \bigoplus_{i=0}^M B_i \mathbf{u}(k-i) \right], \quad (2)$$

where matrix A_0^* is associated to A_0 by means of the Kleene-star operator (Bacelli *et al.*, 1992). The user can graphically visualize the results of the simulation.

Example 2. The following FMS, adapted from (Bacelli *et al.*, 1992), comprises three machines, denoted by M_i , $i = 1, 2, 3$, and is supposed to produce three kinds of parts, namely P_i , $i = 1, 2, 3$, according to a certain product mix. The routes to be followed by each part on the machines are depicted in the upper side of fig. 8 and the processing times are given in Table 1. For carrying the parts only one pallet is available for each part type. The sequencing of part types on the machines is known and it is (P_2, P_3) on M_1 , (P_1, P_2, P_3) on M_2 and (P_1, P_2) on M_3 . Each time a part is completely processed, the corresponding pallet is released and another part of the same type is automatically fed into the system.

Table 1.

Processing times of parts on machines in Example 2.			
	P_1	P_2	P_3
M_1	-	1	5
M_2	3	2	3
M_3	4	3	-

The place-timed PN model is presented in the lower side of fig. 8 that presents a screen capture of the corresponding animated movie accompanying the *PN Toolbox*. The implicit form of the max-plus state-space representation, automatically derived by the *PN Toolbox*, is presented in fig. 9.

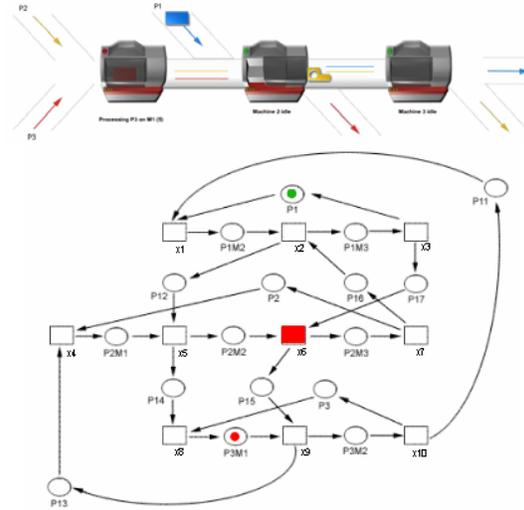


Fig. 8. Routing of parts along the machines in Example 2 and the corresponding PN model.

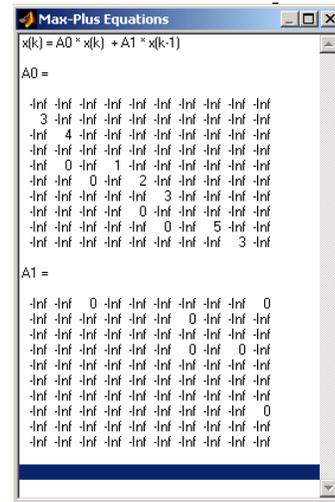


Fig. 9. The max-plus state space representation of the FMS used in Example 2.

Fig. 10 presents the graphical plot of firing time vs. firing count for transitions x_1 , x_2 and x_3 , associated to the occurrences of events driving the progress in processing a part of type P_1 , obtained through max-plus simulation. It can be noticed that the cycle time is equal to 11.

Since the PN model of this FMS is consistent and covered by P-invariants, the minimum cycle time of the FMS can also be directly computed (fig. 11), obtaining the same value.

3.3. Generalized Stochastic Petri Nets

For all types of PN models, three modes of simulation are implemented in the *PN Toolbox*, namely: *Step*, *Run Slow* and *Run Fast*. The *Step* and *Run Slow* simulation modes are accompanied by animation; the user can record the progress of the simulation in a log file with HTML format.

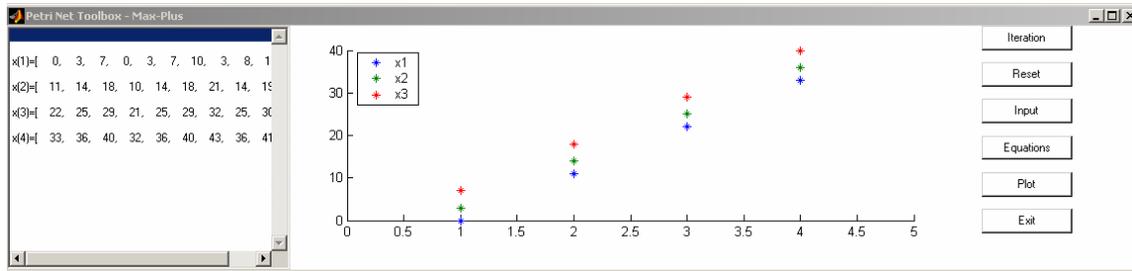


Fig. 10. Graphical plot of firing time vs. firing count for transitions x1, x2 and x3, associated to the occurrences of events driving the progress in processing a part of type P1.

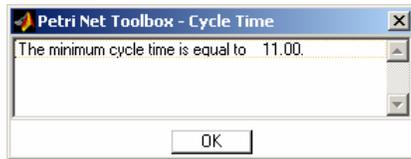


Fig. 11. The cycle time of the FMS used in Example 2.

After ending a simulation experiment, several *Performance Indices* are available to globally characterize the simulated dynamics. Some of the indices recorded for the transitions of the net refer to: the total number of firings during the simulation (*Service Sum*), the mean frequency of firings (*Service Rate*), the mean time between two successive firings (*Service Distance*), the fraction of time when server is busy (*Utilization*).

For the places of the net, the recorded indices refer to: the total number of arrived (*Arrival Sum*) and departed (*Throughput Sum*) tokens, the mean time between two successive instants when tokens arrive in (*Arrival Distance*) and depart from (*Throughput Distance*) the place, the mean time a token spends in a place (*Waiting Time*), the average number of tokens weighted by time (*Queue Length*).

Only for timed or (generalized) stochastic PN, the time evolution for both current and global values of a *Performance Index* may be displayed dynamically while in the *Step* and *Run Slow* simulation modes by means of the *Scope* command.

Another facility available only for timed or (generalized) stochastic PN is *Design*, which can be used for the synthesis of the models. One or two *Design Parameters* varying within intervals defined by the user can be included in the model. For each test-point belonging to this (these) interval(s) a simulation experiment is performed in the *Run Fast* mode. The dependence of a *Design Index* on the *Design Parameter(s)* can be visualized as a graphical plot (2-D or 3-D, respectively).

Example 3. The manufacturing system presented in fig. 12 was selected for illustrating the usage of the **PN Toolbox** in the analysis of Generalized Stochastic Petri nets (GSPNs). The system consists of a machine (server – S) that can process one part (client) at a time and a queue (Q). The capacity of the system is limited to 2 clients. In case there are available slots in

the system, the clients arrive according to a Poisson process of rate $\lambda = 1 \text{ s}^{-1}$. The processing time is exponentially distributed with the mean service rate equal to $\mu = 2 \text{ s}^{-1}$.

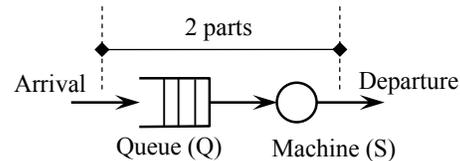


Fig. 12. The manufacturing system used in Example 3.

The corresponding GSPN model is presented in fig. 13, where transitions t1 and t2 are exponentially timed and t3 is untimed.

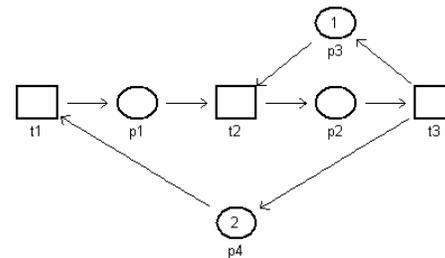


Fig. 13. The GSPN model used in Example 3.

The performances of this system can be obtained from the global performance indices with respect to the places in the net, which are depicted in fig. 14 after simulating the operation of the system for 25,000 s. For example, the utilization of the server is given by the *Queue Length* for p2 (0.429) and the waiting time of a client in the queue corresponds to the *Waiting Time* for p1 (0.167). These values may be compared to the ones obtained analytically according to the theory of the M/M/1/K systems.

For investigating the dependence of the waiting time of a client on the capacity of the system and the mean service time, the *Design* facility can be used. Fig. 15 presents the results of a set of parameterized experiments performed for the capacity of the system (parameter x) ranging between 1 and 10 and the mean service time (parameter y) varying from 0.1 to 10 s. The maximum value of 80.3 s is attained when both independent parameters reach their greatest values in the permitted domain.

Model: ex_3_4_1.xml
 Events: 64195
 Time: 25000.4076

Place Name	Arrival Sum	Arrival Rate	Arrival Dist.	Throughput Sum	Throughput Rate	Throughput Dist.	Waiting Time	Queue Length
p1	21399	0.85595	1.1683	21398	0.85594	1.1683	0.16705	0.14298
p2	21398	0.85591	1.1684	21398	0.85594	1.1683	0.50142	0.42916
p3	21398	0.85591	1.1684	21398	0.85594	1.1683	0.66694	0.57084
p4	21398	0.85591	1.1684	21399	0.85598	1.1683	1.6682	1.4279

Fig. 14. Screen capture of the window presenting the global performance indices with respect to the places in the net, corresponding to the complete processing of 5,000 parts.

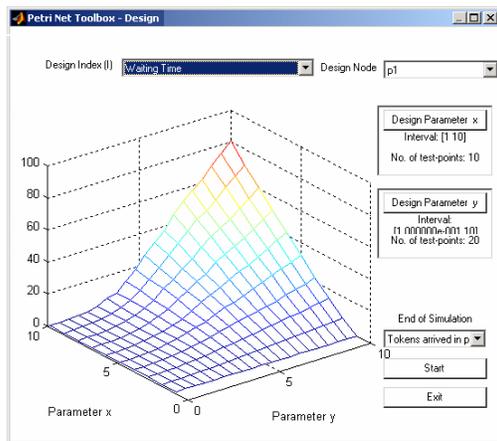


Fig. 15. Waiting Time of a client vs. the capacity of the system (parameter x) and mean service time (parameter y).

4. CONCLUSIONS

Despite the large offer of software products available for MATLAB, none of its toolboxes provides instruments able to handle Petri net models. This fact has motivated the development of the *PN Toolbox* whose usage allows focusing the attention on the proper topics of control engineering by getting rid of the tasks related to code writing. The facilities created for simulation, analysis and design prove useful in many types of applications including a wide range of event-driven dynamics, as illustrated by the three examples presented in the text.

REFERENCES

Bacelli, F., G. Cohen, G.J. Olsder and J.P. Quadrat (1992). *Synchronization and Linearity, An Algebra for Discrete Event Systems*, Wiley, New York.

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

Lewis, F., H.H. Huang, O. Pastravanu and A. Gurel (1995). Control Systems Design for Flexible Manufacturing Systems, In: *Flexible Manufacturing Systems: Recent Developments* (A. Raouf, and M.

Ben-Daya (Eds.)), Elsevier Science, pp. 259-290.

Iordache, M.V. and P.J. Antsaklis (2002). *Software Tools for the Supervisory Control of Petri Nets Based on Place Invariants*, Technical Report ISIS-2002-003, University of Notre Dame, IN, USA, <http://www.nd.edu/~isis/techreports/isis-2002-003.pdf>.

Mahulea, C., L. Bârsan and O. Pastravanu (2001). Matlab Tools for Petri-Net-Based Approaches to Flexible Manufacturing Systems, In: *9th IFAC Symposium on Large Scale Systems LSS 2001* (F.G. Filip, I. Dumitrache and S. Iliescu (Eds.)), Bucharest, Romania, pp. 184-189.

Mahulea, C. (2002). *Petri Net Toolbox – A MATLAB Library for Discrete Event Systems*, MS thesis, The Sheffield University, Department of Automatic Control & System Engineering, UK.

Mahulea, C., M.H. Matcovschi and O. Pastravanu (2003). *Home Page of the Petri Net Toolbox*, <http://www.ac.tuiasi.ro/pntool>.

Martinez, J. and M. Silva (1982). A Simple and Fast Algorithm to Obtain All Invariants of a Generalized Petri Net, In: *Application and Theory of Petri Nets* (C. Girault and W. Reisig (Eds.)), Informatik Fachberichte 52, Springer, pp. 301-310.

Matcovschi, M.H., C. Mahulea and O. Pastravanu (2001). Exploring Structural Properties of Petri Nets in MATLAB, *7th International Symposium on Automatic Control and Computer Science SACCs 2001*, Iasi, Romania, CD Rom.

Matcovschi, M.H., C. Mahulea and O. Pastravanu (2003). Petri Net Toolbox for MATLAB, *11th IEEE Mediterranean Conference on Control and Automation MED'03*, Rhodes, Greece (to appear).

Mortensen, K.H. (2003). *Petri Nets Tools and Software*, <http://www.daimi.au.dk/PetriNets/tools>.

Murata, T. (1989). Petri Nets: Properties, Analysis and Applications, *Proc. of the IEEE*, vol. 77, pp. 541-580.

Svádová, M. and Z. Hanzálek (2001), Matlab Toolbox for Petri Nets, *22nd International Conference ICATPN 2001*, Newcastle, UK, pp. 32-36.