Proceedings of the 2006 IEEE
Conference on Computer Aided Control Systems Design
Munich, Germany, October 4-6, 2006

Petri Net Toolbox in Control Engineering Education

M. H. Matcovschi, C. Mahulea, C. Lefter, and O. Pastravanu

***Abstract*—The Petri Net Toolbox (*PN Toolbox*) for MATLAB is a software package that offers instruments for the simulation, analysis and design of untimed, deterministic and stochastic P-/T-timed and stochastic *Petri nets* (PNs). The facilities available in this toolbox are appropriate for studying the dynamics of many classes of discrete-event systems. The Petri Net Simulink Block (*PNSB*) allows the modeling and analysis of hybrid systems whose event-driven part(s) is (are) modeled based on the PN formalism. The current paper focuses on the exploitation of the *PN Toolbox* for illustrating the usage of the PN theory in Control Engineering from the pedagogic point of view, the discussion being supported by example problems and comments on the teaching goals. The *PN Toolbox* is included in the Connections Program of The MathWorks Inc., as a third party product.**

I. Introduction

DURING the last decade, many universities offering education programs in *Control Engineering* (CE) included in their curricula courses on *Petri nets* (PNs) as a consequence of the impact of the development of the PN field on research and technology. Such courses on PNs should be complemented by relevant computer exercises for laboratory classes. This requires adequate software for analyzing the properties of different types of PNs. Even though many academic or research groups created software tools for studying PNs [1], most of these packages either deal only with particular types of PNs, or offer a limited set of instruments for PN analysis. On the other hand, the regular training of CE students focuses on the exploitation of the generous resources provided by MATLAB and its toolboxes. The idea of bridging the PN formalism with the widely spread usage of MATLAB represented the starting point for the design and implementation of the ***Petri Net Toolbox*** (***PN Toolbox***) at the Department of Automatic Control and Applied Informatics of the Technical University "Gh. Asachi" of Iasi, Romania.

The development of the ***PN Toolbox*** for MATLAB commenced in 2000, as a software product for internal usage at the laboratory classes accompanying the course "Applications of Petri nets in Control Engineering". The instruments implemented in the ***PN Toolbox*** were thought to cover the following syllabus: 1. Untimed PNs, 2. Analysis of behavioral properties, 3. Analysis of structural properties, 4. Procedural control based on PN models, 5. Deterministic P- and T-timed PNs, 6. Stochastic P- and T-timed PNs, 7. Generalized Stochastic PNs, 8. Max-plus representations, 9. Synchronized PNs and hybrid systems. The construction of this syllabus (see also [2]) relied on information carefully selected from well-known bibliographic sources, such as [3] -[9]. The ***PN Toolbox*** provides instruments for simulation, analysis and design, corresponding to the key issues in chapters 1-8 of the above mentioned course. In mid 2003 we released Version 2.0 of the ***PN Toolbox*** [10], [11] that was included, at the beginning of 2004, in the Connections Program of The MathWorks Inc. In 2005, in order to ensure the integration of the simulation facilities available in the ***PN Toolbox*** with Simulink, we started developing the prototype of a ***Petri Net Simulink Block*** (***PNSB***) for storing PN models [12]. This extension permits the simulation of hybrid systems when the event-driven part of their dynamics is described via the PN formalism. Referring to our teaching objectives, we have thus created effective tools for the computer experiments that cover the topics in chapter 9 of our syllabus. The current Version 2.2 of the ***PN Toolbox*** is fully compatible with the latest MATLAB Release 14 SP3.

Thus, the ***PN Toolbox*** broadens the utilization domain of MATLAB toward the area of discrete-event systems, for which the offer of The MathWorks Inc. is limited to the State-Flow product, based on finite state machines, and to SimEvents which extends Simulink with tools for modeling and simulating discrete-event systems using queues and servers. With respect to other PN software for MATLAB, it is worth mentioning the existence of products [13] and [14] that use only the algebraic resources of MATLAB for PN investigation, without incorporating MATLAB-based graphical editors, and which, at this time, are not included in the MATLAB Connections Program.

The current paper presents the ***PN Toolbox*** from the point of view of an educator who is interested in organizing laboratory classes and homeworks / projects for supporting students' insight into the applications of PN in CE. Two important advantages qualify the ***PN Toolbox*** as an appropriate educational aid. (a) It offers a large panel of general purpose instruments for analysis and synthesis of PN models, which, on the best of our knowledge, cannot be found, all together, in any other software package. (b) The user can integrate PN models into Simulink diagrams for modeling complex hybrid systems. Aspects related to (a) are detailed in Section II, whereas aspects related to (b) are illustrated by Section III. Relevant examples are included in the text for emphasizing concrete teaching goals.

II. Simulation, Analysis and Design with PN Toolbox

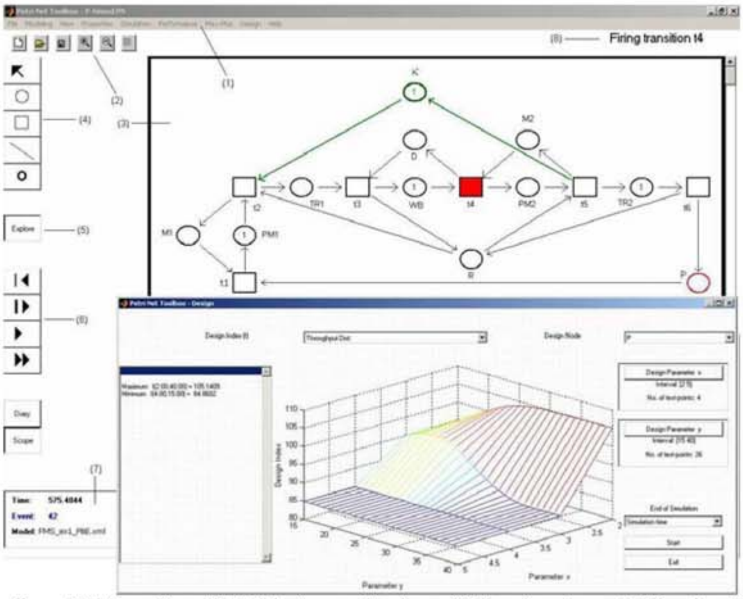
The first part of the current section creates an overview of the training facilities available in the ***PN Toolbox***, whereas the second part discusses a series of teaching objectives in solving problems, assisted by the ***PN Toolbox***.

The authors are with the Department of Automatic Control and Applied Informatics, Technical University "Gh. Asachi" of Iasi, Blvd. Mangeron 53A, 700050 Iasi, Romania (e-mail: pntool@delta.ac.tuiasi.ro).

C. Mahulea is now a PhD student at the University of Zaragoza, Spain.

C. Lefter is now a PhD student at Boston University, MA ,USA.

0-7803-9797-5/06/$20.00 ©2006 IEEE



Time: 575.4844
Event: 42
Model: FMS_ex1_PBE.xml

Fig. 1. The GUI of the *PN Toolbox*: (1) Menu Bar, (2) Quick Access Toolbar, (3) Drawing Area, (4) Drawing Panel, (5) Draw / Explore Switch, (6) Simulation Panel and (7) Status Panel.

A. *PN Toolbox at a First Glance*

In the current version of the ***PN Toolbox***, five types of classic PN models are accepted, namely: untimed, place-timed, transition-timed, stochastic and generalized stochastic. The timed nets can be deterministic or stochastic, and the stochastic case allows using appropriate functions to generate random sequences corresponding to probability distributions with positive support. The places can have finite or infinite capacities. Besides regular arcs, the PN models can contain double or inhibitor arcs. In addition, the ***PN Toolbox*** allows the assignment of priorities and/or probabilities to conflicting transitions.

The ***PN Toolbox*** has an easy to exploit *Graphical User Interface* (GUI) (fig. 1) that offers a great maneuverability and helps the user concentrate strictly on the specific problems of discrete event dynamic systems (by circumventing the need for writing codes). A complete description of the GUI and its usage can be found in the online help available on the website have created for the ***PN Toolbox*** [15]. For the sake of brevity, we only present here some types of analysis that the user can perform after drawing a PN model, simply by accessing the menus of the GUI: (i) visualization of the *Incidence Matrix*, which is automatically built from the net topology; (ii) exploration of the *Behavioral Properties* (such as liveness, boundedness etc.) by consulting the *Coverability Tree*, which is automatically built from the net topology and initial marking [2], [3]; (iii) identification of special *Topologies*, i.e. marked graph (MG), state machine (SM), free-choice net (FCN), extended free-choice net (EFCN) and asymmetric choice net (ACN) [2], [3], based on the algorithm described in [16]; (iv) computation of *Siphons* and *Traps* for ordinary PNs [3] and investigation of liveness for FCNs and ACNs [16]; (v) exploration of *Structural Properties* (structural boundedness, repetitiveness, conservativeness and consistency) [2]; (vi) calculation of *P*- and *T-Invariants* based on an algorithm adapted from the theory in [3], [17]; (vii) running a *Simulation* experiment; (viii) displaying current results of the simulation using the *Scope* and *Diary* facilities; (ix) evaluation of the global *Performance Indices* (such as average marking of places, average firing delay of transitions, etc.); (x) *Max-Plus Analysis* (restricted to *P*-timed event-graphs) [4], [18]; (xi) *Design* of a configuration with suitable dynamics (via automated iterative simulations).

In order to enrich the educational value of the ***PN Toolbox***, on the dedicated website [15] we have also posted some animated demos whose purpose is to present specific sequences of operations in handling the GUI and to guide the user in the interpretation of the numerical results. By watching these demos, the user learns how to handle the key problems of discrete event systems within a PN framework. At the same time, the ***PN Toolbox*** was meant to illustrate, by short movies, behaviors that are typical for discrete event systems, such as, for instance, sequential / parallel sharing of resources, routing policies, services in queuing networks, etc. Each movie shows the physical motion of a real-life system synchronized with the token dynamics in the associated PN model.

B. *Objectives of the **PN Toolbox** - Assisted Training*

The students' training relies on various types of problems that can be solved by the appropriate usage of the instruments available from the GUI of the ***PN Toolbox***. For a concrete discussion of some important teaching goals, below, we refer to a semester project sequencing ten problems (**P**_***n***, $n = 1,\ldots,10$), with numerous connections between them, and, concomitantly, we point out the objectives (**O**_***n***, $n = 1,\ldots,10$), of the training assisted by the ***PN Toolbox***. Consider a *flexible manufacturing system* (FMS) that consists of two different machines (a lathe (M1) and a drilling machine (M2)), a robot (R) and a buffer (B) with two slots between the two machines, as described in [19] (fig. 2). Every input part must be processed by M1 first and then by M2 in order to get the final product. Both machines are automatically loaded and are unloaded by the robot. A variable number of pallets can be used to fix on the processed parts. At the initial moment, all resources are idle.

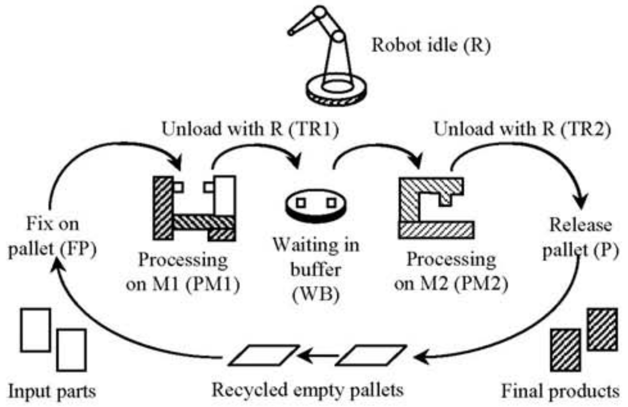


Fig. 2. Schematic representation of the FMS.

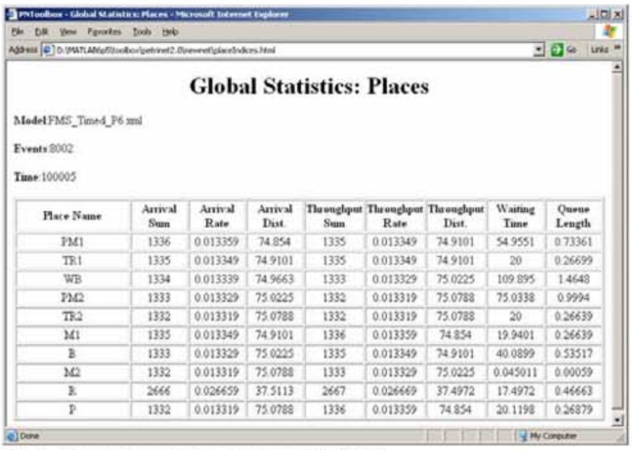
(**P_1**) Construct the untimed PN model of the FMS and simulate it (*Run Slow*) in the cases of using 2 to 5 pallets. Explain the simulation results in terms of the behavioral properties of the PN model and in terms of the shared usage of the robot. (**O_1**) The simulation based on untimed PNs is recommended for testing the logical functioning of the FMSs, independent of the durations of operations. In our case (see the PN model in fig. 1, but without place K and its input and output arcs), for 2 or 3 pallets the simulation runs for an arbitrarily large number of events, but for 4 or 5 pallets it reaches a deadlock corresponding to a circular blocking of the resources [20]. The coverability tree built by the ***PN Toolbox*** confirms that for 2 or 3 pallets the PN model is live and reversible, whereas for more than 3 pallets it contains a deadlock marking.

(**P_2**) Determine the minimal support P- and T- invariants of the PN model built at (**P_1**) and explain their physical meaning. (**O_2**) For PNs modeling real-life systems the P-invariants are related to the status of the resources, whereas the T-invariants reflect the sequencing of the operations. The student will note that each minimal support P-invariant (computed by the ***PN Toolbox***) is associated with a resource in the sense that each minimal support contains the place corresponding to the idle resource as well as all the place(s) corresponding to the operation(s) performed by the considered resource. There exists a single basic T-invariant showing that the FMS returns to its initial state if each operation is started (and completed) once.

(**P_3**) Analyze the structural properties of the PN model built at (**P_1**) and discuss their connections with the basic P- and T- invariants determined above. (**O_3**) The ***PN Toolbox*** supports the analysis of the structural properties of PNs by handling the algebraic equalities and inequalities characterizing them. The student should notice that the PN is covered by P- / T-invariants, therefore it is conservative / consistent; consequently it is also bounded / repetitive [3].

(**P_4**) Considering the buffer B allocated in advance (concomitantly with the robot for unloading M1), redraw the PN model of the FMS with 4 pallets and analyze its liveness relying on the topology and the token contents of the siphons and traps. (**O_4**) The ***PN Toolbox*** can be used (a) to classify the PN model from the topological point of view (in this case, it is an asymmetric choice net), (b) to compute its siphons and traps and (c) to check if each siphon contains traps marked by the initial marking. In our case, the particular results of (a), (b), (c) ensure the liveness [3].

(**P_5**) Consider a kanban control [19] with 3 cards for limiting the number of parts (pallets) that may be in one of the following stages: unloading from M1, waiting in buffer B or being processed on M2. Redraw the PN model of the FMS with 4 pallets and analyze its liveness by (i) the same procedure as in (**P_4**) and (ii) the coverability tree. (**O_5**) By using the **PN Toolbox** for the model represented in the upper side of the background window from fig. 1, the student will see that the liveness criterion based on facts (a), (b), (c) in (**O_4**) is just sufficient and it does not offer a universal instrument for testing liveness.


PNToolbox - Global Statistics: Places - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Address D:\MATLAB6p5\toolbox\petrinet2.0\newnet\placeIndices.html
Go
Links
Done
My Computer

Global Statistics: Places

Model FMS_Timed_P6.xml

Events 8002

Time 100005

Place Name	Arrival Sum	Arrival Rate	Arrival Dist.	Throughput Sum	Throughput Rate	Throughput Dist.	Waiting Time	Queue Length
PM1	1336	0.013359	74.854	1335	0.013349	74.9101	54.9551	0.73361
TR1	1335	0.013349	74.9101	1335	0.013349	74.9101	20	0.26699
WB	1334	0.013339	74.9663	1333	0.013329	75.0225	109.895	1.4648
PM2	1333	0.013329	75.0225	1332	0.013319	75.0788	75.0338	0.9994
TR2	1332	0.013319	75.0788	1332	0.013319	75.0788	20	0.26639
M1	1335	0.013349	74.9101	1336	0.013359	74.854	19.9401	0.26639
B	1333	0.013329	75.0225	1335	0.013349	74.9101	40.0899	0.53517
M2	1332	0.013319	75.0788	1333	0.013329	75.0225	0.045011	0.00059
R	2666	0.026659	37.5113	2667	0.026669	37.4972	17.4972	0.46663
P	1332	0.013319	75.0788	1336	0.013359	74.854	20.1198	0.26879

Fig. 3. Simulation results commented in (**P_6**).

(**P_6**) Consider the FMS with 4 pallets and without kanban control, as in (**P_1**), with the following durations of the operations: dPM1 = 40 time units (t.u.), dPM2 = 75 t.u. for processing on M1, respectively on M2, dTR1 = dTR2 = 20 t.u. for unloading the two machines and dp = 20 t.u. for releasing the pallets. Construct the deterministic P-timed PN model and simulate the FMS for 100,000 t.u. Determine the number of processed parts, the duration of a production cycle, the utilization of M1, M2 and R. (**O_6**) The student learns to evaluate the technological performances of manufacturing systems by exploring the dynamics of the corresponding timed PN models. The simulation results presented in fig. 3 show that the FMS processes 1336 parts (*Throughput Sum* for place P), a production cycle lasts for about 75 t.u. (*Arrival Distance* for place P) and the utilization of the resources is about 0.73 for M1 (*Queue Length* for place PM1), 1 for M2 (*Queue Length* for place PM2) and 0.53 for R (sum of *Queue Length* indices for places TR1 and TR2).

(**P_7**) Repeat the simulation at (**P_6**) for the case when the duration for processing on M2 is dPM2 = 85 t.u. (**O_7**) The simulation ends in a deadlock that does not occur at

(**P_6**), due to a favorable matching of the durations considered for the operations. Since in a real FMS the durations of the operations can vary from a processed part to another, the deadlock problem needs an adequate analysis based on the untimed PN model, as discussed at (**P_1**).

(**P_8**) For the FMS with 4 pallets and kanban control consider the same durations for the operations as in (**P**_7). Simulate the FMS for 100,000 t.u. and comment on the changes of the performance indices with respect to (**P_6**). (**O_8**) The student has to understand the role of the bottleneck machine M2 that imposes the duration of a production cycle.

(**P_9**) Consider the FMS with 4 pallets and kanban control. The processing times on M1 and M2 are uniformly distributed in the intervals [35, 45] t.u. and, respectively [80, 90] t.u., while each unloading operation takes 20 t.u. Find the optimal number of pallets and the optimal duration (considered constant but unknown) for releasing and recycling a pallet so as to ensure the best value for the mean production cycle time. Give an intuitive interpretation of the result. (**O_9**) A design experiment can be performed in the **PN Toolbox**, by considering two design parameters, namely the number x of pallets in the system (varying from 2 to 5), and the time y for releasing a pallet (varying from 15 to 40 t.u.). The Design window is shown in the lower part of fig. 1 and presents the dependence of the two parameters for the mean production cycle time. The minimum value of the mean production cycle time is 85 t.u. and it can be obtained when using 4 or 5 pallets with arbitrary release time in the considered interval. However the same minimal value can be achieved for 3 pallets and a releasing time between 25 t.u. and 40 t.u. because, when the releasing time is smaller than 25 t.u., a pallet remains on M2 for more than the mean processing time of 85 t.u. (waiting to be unloaded by R, which is busy unloading M1 when the processing on M2 finishes). Intuitively reasoning, the mean production cycle time cannot be reduced below the value of 85 t.u. because this value represents the mean processing time on the bottleneck machine M2.

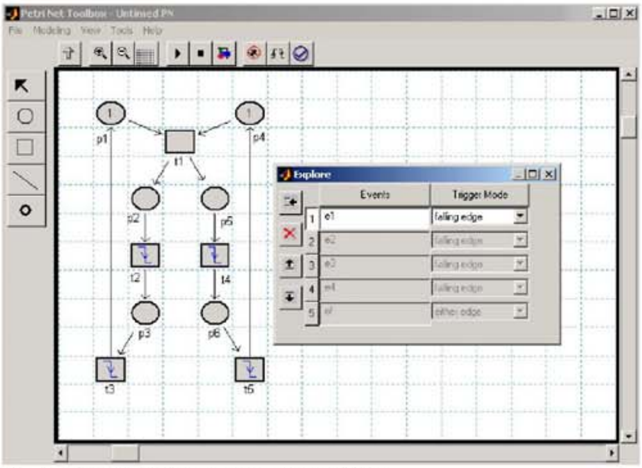
III. Integration of PN Models with Simulink

The first part of the current section presents the recently created ***Petri Net Simulink Block*** (PNSB) that allows the integration of the PN formalism with the simulation philosophy of the Simulink environment [12]. The second part discusses some important training objectives envisaged by the simulation of hybrid systems whose discrete-event driven parts are modelled by PNs.

A. Brief Description of PNSB

This version of the PNSB has been developed to allow the connection of untimed/P-timed/T-timed PNs with other blocks, available in the standard Simulink libraries. The PN model stored in a PNSB should contain synchronized (triggered) transitions [5] whose firing depends on the occurrence of external (triggering) events. A *synchronized transition* is fired whenever (i) it is enabled by the net marking and (ii) one of its associated triggering events, defined at the PN level, occurs. Both finite and infinite server semantics [5] can be used for transition firings. A Simulink block diagram can contain any number of PNSBs needed to model a complex system.

The PNSB accepts, as input, a set of signals in a multiplexed form; the evolution of each signal can generate Simulink events. The number of input signals has to be equal to the number of events defined for the PN. Once an input signal generates an event, the simulation time of Simulink "freezes", the PNSB identifies the generated event and fires all the enabled transitions. After that, the simulation is resumed until a new event is generated. At each simulation step, the PNSB outputs the net marking as a vector of n integer values, where n represents the number of places in the net and broadcasts internal events generated by firing certain transitions in the PN model.



	Events	Trigger Mode
1	e1	falling edge
2	e2	falling edge
3	e3	falling edge
4	e4	falling edge
5	e5	either edge

Fig. 4. The main window of the PNSB Editor.

The PNSB is equipped with a GUI that allows a user to draw a PN model (PNSB Editor - see fig. 4), define the triggering events and debug the Simulink model. The functions of the PNSB Editor are similar to the editing facilities available in the GUI of the ***PN Toolbox***. A complete description of the PNSB and its usage can be found in the online help available on the website [21].

B. Training Objectives for Simulink-Based Analysis of Hybrid Systems

By enriching the collection of Simulink blocks with the PNSB, we aimed to offer our students the possibility of studying hybrid systems exhibiting event-driven dynamics modeled by the PN formalism. For a concrete discussion of some important teaching goals we refer below to a laboratory work sequencing seven *problems* (**PH_*n***, $n = 1,\ldots,7$), with connections between them, and, concomitantly, we point out the *objectives* (**OH_*n***, $n = 1,\ldots,7$), of the training that relies on Simulink experiments.

Consider the hydraulic system in fig. 5 consisting of two cylindrical tanks T*i*, $i = 1,2$, with the cross section areas $A_1 = 1.7671$ m^2, $A_2 = 0.7854$ m^2. The tanks are filled with liquid (ρ = 1000 kg/m^3), by a common pressure source PS = 10^4 N/m^2 through two on/off admission valves V$_i$,

$i = 1,2$ (with the hydraulic resistances R_{vi}, $i = 1,2$). When the liquid level h_i in tank Ti, $i = 1,2$, reaches a given maximum height $h_{1max} = h_{2max} = 0.9$ m, valve V$_i$ is closed and, concomitantly, the evacuation valve W$_i$ (with the hydraulic resistances R_{wi}, $i = 1,2$) is opened and it remains opened until the liquid level in tank T$_i$, $i = 1,2$, reaches a given minimum height $h_{1min} = h_{2min} = 0.1$ m. The admission valves V$_i$, $i = 1,2$, are opened simultaneously once both evacuation valves are closed. Initially, the two tanks are empty and the four valves are off. By pushing the START button, valves V$_1$, V$_2$ are turned on and the system begins to operate.

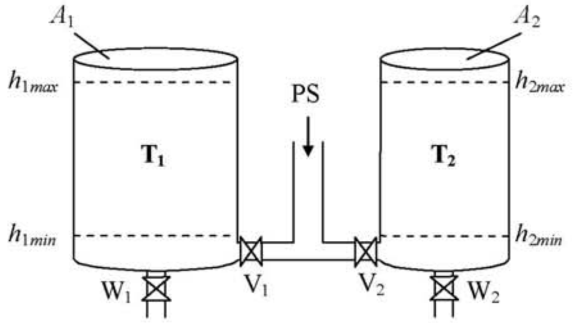


Fig. 5. Schematic representation of the hydraulic system.

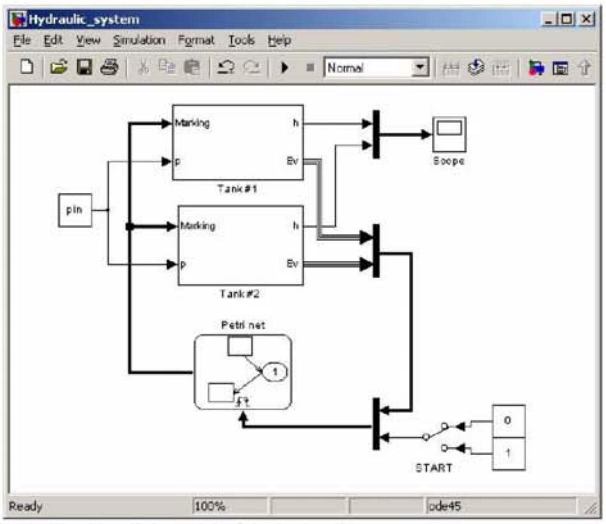
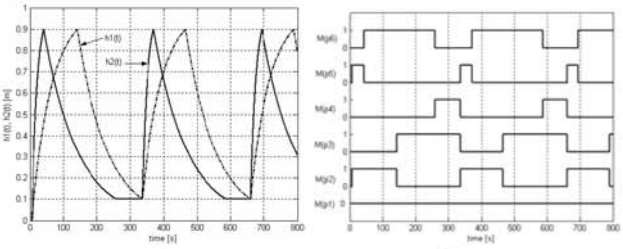


Fig. 6. The Simulink model of the hydraulic system.

(**PH_1**) Construct a Simulink model of the hydraulic system. (**OH_1**) The student should consider two blocks for modeling the continuous time evolution of the liquid level in each tank and a PNSB for modeling the event-driven part of the dynamics (see fig. 6). The PNSB storing the PN model in fig. 4 receives the events corresponding to the situations when h_i reaches the limits h_{imax} and h_{imin}, $i = 1,2$ (for the external synchronization of transitions t2, t3, t4 and t5) and delivers its current marking which drives the control logic for switching the valves. A manual switch block simulates the action of the START button.

(**PH_2**) Consider the following values of the hydraulic resistances R_{v1}=333,330 kg/sm^4, R_{w1}=500,000 kg/sm^4, R_{v2}=200,000 kg/sm^4, R_{w2}=1,250,000 kg/sm^4. Simulate the behavior of the system for 800 seconds and use the simulation results to comment on the interaction between the two continuous-time blocks and the PNSB. (**OH_2**) By displaying the simulation results the student should understand the role of the triggering events in driving the dynamics of the PN and the role of the marking in switching the continuous-time models. The graphical plots in fig. 7 correspond to the case when the START button is pressed 7 s after the simulation commences.

(**PH_3**) Use the simulation results at (**PH_2**) to analyze the transient- and steady-state operation of the hydraulic system. (**OH_3**) The initial condition and the simulation horizon are chosen so that the student can observe the transient state (lasting for 330 s in fig. 7) and a cyclic steady-state (lasting for 324 s in fig. 7). Such periodic dynamics are typical for some classes of hybrid systems.



(a) (b)

Fig 7. The graphical plots (a) of h$_1$(t), h$_2$(t) and (b) of the net marking.

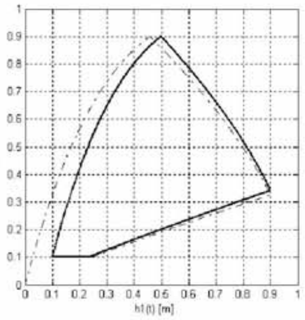


Fig. 8. The state space portrait commented in (**PH_4**).

(**PH_4**) Based on the simulation results at (**PH_2**), draw a state-space portrait in the (h_1, h_2) plane and use it to comment on the transient- and steady-state operation of the hydraulic system. (**OH_4**) The state-space portrait is a usual analysis instrument for 2 states systems and the integration of our PNSB with Simulink facilitates its construction. The student can easily identify an open curve corresponding to the transient state (dashed line in fig. 9) and a closed curve corresponding to the cyclic evolution (solid line in fig. 9).

(**PH_5**) If the hydraulic resistances R$_{v1}$, R$_{w1}$ have the same values as used at (**PH_2**), find the values of R$_{v2}$, R$_{w2}$ such that the state-space portrait is a segment belonging to the bisector of the first quadrant. Specify the coordinates of the two end-points of this segment. (**OH_5**) The student will better understand the parameterization (with respect to time) of the state-space portrait, since the requirement in (**PH_5**) can be fulfilled if and only if $h_1(t) = h_2(t)$ for all t, meaning $R_{v2} = A_1R_{v1}/A_2$ and $R_{w2} = A_1R_{w1}/A_2$. In this case, the student can also notice that for the PN model in fig. 5 the time-evolutions of the markings of places p1, p2 and p3 are similar to the ones of places p4, p5 and p6, respectively.

(PH_6) For the hydraulic resistances R_{v1}, R_{w1}, R_{v2}, R_{w2} consider the same values as given at **(PH_2)**. Construct a P-timed PN that models the steady-state behavior of the hydraulic system. **(OH_6)** Student should notice that the qualitative aspects of the cyclic behavior can be captured by a P-timed PN with the same topology as in fig. 5 and the following timing of the places: d(p1) = d(p4) = 0 s, d(p2) = 130 s, d(p3) = 194 s, d(p5) = 34 s, d(p6) = 214 s. The simulation under ***PN Toolbox*** yields 324 s for the duration of a cycle, value which can also be calculated by means of the formula available for P-timed marked graphs (e.g. [3]). Nevertheless, this model gives no quantitative information about the liquid levels in the tanks at arbitrary times.

(PH_7) Under the hypotheses of **(PH_6)**, is it possible to construct a PN that models both the transient and the steady state? **(OH_7)** The timing of the P-timed PN differs from the transient to the steady state, meaning that the usage of a single PN model is not possible. This together with the remarks at **(OH_6)** proves the usefulness of investigating the hydraulic system as hybrid system, and, consequently shows the importance of the PNSB as teachware for CE education.

IV. Conclusions

The newly created software tools presented in this paper allow developing complex and attractive computer exercises for illustrating the applications of PNs in CE. The effectiveness of these tools as educational aids is strongly supported by their full compatibility with MATLAB and Simulink, which represent the most popular products used in CE training. The examples of problems and teaching objectives taken into discussion by the previous sections have been selected so as to reflect the importance (from the pedagogic point of view) of the software instruments designed and implemented by the authors. Obviously, by organizing laboratory classes and homeworks based on the facilities of the ***PN Toolbox*** and ***PNSB***, educators can approach much larger areas of applications than considered by the examples included in our text.

Another type of benefits, resulting from the exploitation of the ***PN Toolbox*** and deserving the educators' attention, refers to the construction of MATLAB specialized procedures, where the users borrow functions, data structures or programming techniques from the ***PN Toolbox***. Thus, with a reduced programming effort, the educator can build his/her own instruments responding to a particular set of training needs, which is not completely covered by the facilities available in the ***PN Toolbox***.

References

[1] K.H. Mortensen, *Petri Nets Tool Database*, http://www.informatik.uni-hamburg.de/TGI/PetriNets/tools/, 2006.

[2] O. Pastravanu, M. H. Matcovschi and C. Mahulea, *Applications of Petri Nets in Studying Discrete Event Systems* (in Romanian), Iasi: Ed. Gh. Asachi, 2002.

[3] T. Murata, “Petri Nets: Properties, Analysis and Applications”, *Proc. IEEE*, 1989, pp. 541-580.

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

[5] R. David and H. Alla, *Discrete, Continuous, and Hybrid Petri Nets*, Berlin Heidelberg: Springer-Verlag, 2005.

[6] A. A. Desrochers and R. Y. Al-Jaar, *Petri Nets for Automated Manufacturing Systems: Modeling, Control and Performance Analysis*, Rensselaer Polytechnic Institute, 1993.

[7] M. C. Zhou, and F. DiCesare, *Petri Net Synthesis for Discrete Event Control of Manufacturing Systems*, Boston: Kluwer, 1993.

[8] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems*, Norwell, MA: Kluwer Academic Publishers, 1999.

[9] G.Nenninger and V.Krebs, “Modeling and Analysis of Hybrid Systems: A New Approach Integrating Petri Nets and Differential Equations”, *Proc. of the 1997 Joint Wksp. on Parallel and Distributed Real-Time Systems (WPDRTS/OORTS)*, Geneva, Switzerland, 1997, pp. 234-243.

[10] M. H. Matcovschi, C. Mahulea and O. Pastravanu, “Petri Net Toolbox for MATLAB”, *11th Mediterranean Conference on Control and Automation MED'03*, Rhodes, Greece, 2003 (CD-ROM).

[11] O. Pastravanu, M. H. Matcovschi and C. Mahulea, “Petri Net Toolbox – Teaching Discrete Event Systems under Matlab”, In M. Voicu (Ed.), *Advances in Automatic Control*, Boston: Kluwer Academic Publishers, 2004, pp. 247-256.

[12] M. H. Matcovschi, C. Popescu, O. Pastravanu, "A new approach to hybrid system simulation: Development of a Simulink library for Petri Net models", *Control Engineering and Applied Informatics*, vol. 7, nr. 4, pp. 55-62, 2005.

[13] M.V. Iordache and P.J. Antsaklis, “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, 2002.

[14] M. Svádová and Z. Hanzálek, “PN Matlab Toolbox 2.0”, In *MATLAB 2004 - Sborník příspěvků 12. ročníku konference*. Praha: VŠCHT, 2004, Vol. 2, s, pp. 546-552, 2004.

[15] C. Mahulea, M. H. Matcovschi and O. Pastravanu, *Home Page of the Petri Net Toolbox*, http://www.ac.tuiasi.ro/pntool, 2005.

[16] C. Lefter, C. Mahulea, M. H. Matcovschi and O. Pastravanu, “Instruments for topology-based analysis in Petri Net Toolbox”, *Periodica Politechnica Timisoara, Transactions on Automatic Control and Computer Science*, Vol.49 (63), pp. 153-158, 2004.

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

[18] M. H. Matcovschi, C. Mahulea and O. Pastravanu. “Computer Tools for Linear Systems over Max-Plus Algebra”, *Periodica Politechnica Timisoara, Transactions on Automatic Control and Computer Science*, Vol. 47 (61), pp. 97-102, 2002.

[19] F. Lewis, H. H. Huang, O. Pastravanu and A. Gurel, “Control Systems Design for Flexible Manufacturing Systems”, In A. Raouf and M. Ben-Daya (Eds.), *Flexible Manufacturing Systems: Recent Developments*, Elsevier Science, 1995, pp. 259-290.

[20] F. Lewis, A. Gürel, S. Bogdan, A. Doganalp and O. Pastravanu, “Analysis of Deadlock and Circular Waits Using a Matrix Model for Flexible Manufacturing Systems”, *Automatica*, Vol. 34, No. 9, pp. 1083-1100, 1998.

[21] M. H. Matcovschi and O. Pastravanu, *Home Page of the Petri Net Simulink Block*, http://www.ac.tuiasi.ro/~pnsb, 2006.