A Petri Nets Formalism for Modeling and Simulation of DC Motor Drives Components

Mircea Adrian DRIGHICIU
University of Craiova
str.A.I.Cuza nr.13, RO-200585 Craiova
adrighiciu@em.ucv.ro

Abstract— **In this paper we deal with the problem of modeling and simulation a DC electrical drive, considered as a hybrid system which switches between its states in response to generic and non-generic external events. The model we use, called Modified Hybrid Petri Nets, is a hybrid model that combines differentials and discrete event dynamics and enables us to simulate the behavior of DC electrical drive in different service operations. A real application case was finally considered; simulation is carried out with an available software Visual Object Net ++.**

Index Terms—**DC motor drives, Load modeling, Modeling, Petri nets, Object oriented methods.**

I. INTRODUCTION

Hybrid systems have known recently considerable attention and many developments regarding their modeling and control have appeared in the literature. The most usual paradigm of a hybrid system is a supervisory control architecture, whose fundamental structure consists of a high level (supervision) where a discrete – event system models decision making, and a low level (process) where a continuous system models the process and its local control loops [2], [3], [4], [5], [6].

The behavior of the system between its states (the passage from an old to a new operating regime) is induced by the occurrence of the external asynchronous events. The higher level of the whole model treats all sequences of events (starts and ends of operations, occurrences of the unforeseen situations, internal or external disturbances etc.) without complete information regarding the evolution of the process in the elapsed time between events.

For our studies, the basic idea was to consider a general hybrid system (i.e. an DC electrical drive) such a dynamic structure, which can change it behavior in response to generic events (created when the state variable reaches specific values and integrated by the feeding back into the continuous system) and non – generic events (external events from the outside of the system and possible generic events of other systems). Thus, it is advantageous, if not indispensable, to be able to represent both continuous and discrete parts of the hybrid system in the same context, by a unique formalism, as far as possible [3], [4], [5], [6], [8].

In order to do the framework for hybrid systems, different approaches of modeling are used and at present is already an abundance of such paradigms, including Petri nets. Having a dual nature of a graphical tool and a mathematical object they can serve in both the theoretical and practical applications. At present, these models are frequently used for real time control and performance evaluation in a hybrid context and have become a favorite topic for the dynamic analysis of automation and production in modern technology. In this context, developed by David and Alla in the 1990s, Continuous Petri Nets, Hybrid Petri Nets and their numerous extensions, offer powerful concepts and formal techniques for expressing fundamental discrete events and continuous time behavior [2], [3], [4], [5], [13], [14], [15].

II. BACKGROUND ON HYBRID PETRI NETS

An exhaustive presentation of this formalism is out of purpose of the present work. Our approach is focused on the specific representations of one particularly class of Hybrid Petri Nets used for modeling of electric drives structure, assuming that the main definitions and rules of this formalism are known [2], [3], [4], [5], [6], [10], [11].

However, we must note here that among the main discrete event dynamic systems models, Petri net techniques have a special position because they seem to be more accepted in industry and the applications domain, in general.

It is known that the major step in the effort to enhance the modeling power of discrete Petri Nets (PN) has been their extensions, named Continuous Petri Nets (CPNs). In this case, the basic idea was to consider that the marking of places can be a real number instead of an integer. Hence, Continuous Petri nets are thus approximations to discrete – event systems allowing, basically, faster simulation of the latter without scarifying accuracy. Various timed CPNs models have been defined and they correspond to different calculation of the firing speeds associated to the transitions [4], [5], [13], [14].

A Hybrid Petri Net (HPN) is a combination of ordinary and continuous Petri nets (Fig.1). This model can treat integer variables together with real variables and symbolic variables usually encountered in other models of hybrid systems.

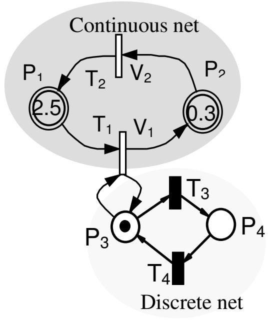


Fig. 1 A Hybrid Petri net structure

It inherits all the modeling facilities of Petri nets such as the ability to capture concurrency, synchronizations and conflicts, allowing to model systems with continuous flows and linear evolutions in an intuitive way [2], [3], [4], [5], [8].

There are two parts in a classical HPN (Fig.1): *a discrete part* and *a continuous part*, both interconnected thanks to arc linking a discrete node (pace or transition) to a continuous node (transition or place). In some cases, one part can influence the behavior of the other part without changing it own marking. In other cases, the firing of a D – transition can modify both the discrete and the continuous marking. Usually, the firing speeds – associated to the continuous transitions of the net – are constants and theirs values can be the same, or different. Moreover, using basic HPN models it is not possible to represent negative continuous variables.

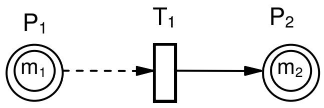
Differential Petri Nets (DPNs) constitute an extension of HPNs and can represent simultaneously continuous dynamic systems, modeled as systems of ordinary differential equations, and discrete event driven systems [6], [7], [12]. DPNs extend and combine the advantages of continuous-type Petri nets models and those of PN. The novel features of DPNs are the negative real values accepted for place markings and the use of an integration mechanism for the approximate representation of the continuous systems. Under the assumption that the continuous system can be represented by a *n* linear first order difference state equation, they are powerful enough to model in a single graph an hybrid system.

Also, similar to HPNs, a fundamental relation can be formulated, from which it is possible to deduce from a given marking at time t_i the reachable marking at the date t_k:

$$\boldsymbol{M}(t_k) = \boldsymbol{M}(t_i) + \boldsymbol{U} \cdot \left(\boldsymbol{\sigma}(t_k) + \int_{t_i}^{t_k} \boldsymbol{v}(u)du\right), \quad (1)$$

where $\boldsymbol{U}$ is the incidence matrix, the characteristic vector $\boldsymbol{\sigma}(t)$ represents the firing sequence for the discrete transitions and the speeds vector $\boldsymbol{v}(t)$ contains the instantaneous firing speeds of differential transitions.

Another powerful extinction of the classical structures of HPNs consist by the assignation to every continuous transition a firing speed as a function, whose arguments can be the token quantities of arbitrary places of the net (Fig. 2), [8], [9], [10], [11], [12]. Also, it is possible that some places are linked by test arcs – which not change at all the marking of connected places - to the output transitions, authorizing in this way their firing. The result is named Modified Petri Nets – (MPNs).



$v = f(m_1, m_2, ..., m_i, m_{i+1}, ... m_n)$

$m_1(t)$, $m_2(t)$ – quantity of tokens in P_1, P_2

$m_i(t)$, $m_{i+1}(t)$ – quantity of tokens in P_i, P

v – continuous transition firing speed

Fig. 2 A MPN topology

The firing speed *v* is the speed of token flow into the place P_2 [8], [9]. The transition T_1 is always active; it can be inactivated only by the empty discrete input places. The test arc (dotted line represented) P_1 – T_1 does not allow token flow.

Usually, if a single input arc is directed from P_i to P_{i+1}, the firing speed – *v* - of continuous connection transition between those ones is given by:

$$v = \frac{dm_{i+1}(t)}{dt} = -\frac{dm_i(t)}{dt} \quad (2)$$

For a place P $_i$ with „ j ” input arcs (Fig.3):

$$\dot{m}_i(t) = \frac{dm_i(t)}{dt} = \sum_j v_j \quad (3)$$

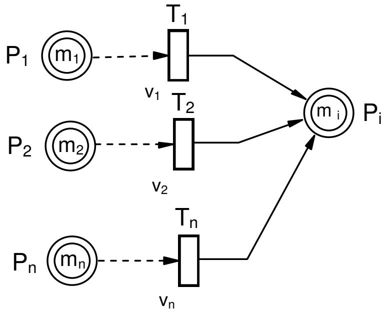


Fig. 3 Continuous place with „ j” input transitions

In order to check all of goods properties of the models, algebraic techniques can be used in addition with various software tools, which assures in the mean time different simulation scenarios in many different initial conditions [8], [9], [10], [11]. Also, all proposed models can be verified comparing theirs simulation scenarios results with others, offered by the classical known technical tools, like Matlab – Simulink, Modelica etc. or with laboratory experimental test.

III. CASE STUDY

In order to prove the resources of MPNs formalism in application of various synthesis techniques of the models, a DC electrical drive system was considered. Starting from structural scheme of the physical system, and then using a linear mathematical model of the electric drive (with several usual simplifications) a basic topology of the Petri net model was achieved. Then, according to different functional conditions of the physical system, additional structural elements were added to the main model, in order to obtain a modular and flexible framework – a Object Petri Net (OPN).

Mathematical model

Usually, the initial mathematical model of an DC electrical drive (Fig.4) consist of a linear differential equations set, which emphasizes the main system structure and stipulate the evolution rules for this one.

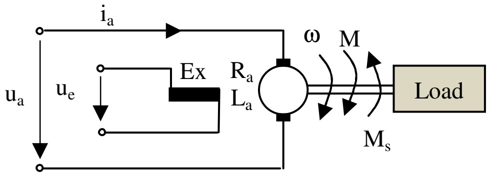


Fig. 4 A DC motor drive structure

If the inductances of the armature circuits are constant values (L_a = ct., L_e = ct.), by neglecting the rotor reaction effect, also the breaking brushless voltage ($\Delta u_p = 0$), the friction in the bearings and, moreover, considering that the motor is commanded by the rotor circuit (Φ = ct.), the equations set can be written in relative coordinates [1], [11]:

$$u = \rho \cdot i + \rho \frac{T_a}{T_m} \frac{di}{d\tau} + \varphi \cdot \nu \quad (4)$$

$$\varphi \cdot i = \mu_s + \frac{d\nu}{d\tau} \quad (5)$$

The main idea used to achieving the Petri Net model was to associate the input, state or output variables of the mathematical model with continuous places of an MPN model. Also, to the continuous transitions of the model will be associated variable firing speeds in accordance with (2) and (3), having as result a modular and flexible model topology with a variable structure.

The MPN model

Following the above idea, a first structure of the model (Fig.5) was synthesized starting from (4) and (5), using the facilities offered by DPNs and MPNs formalism.

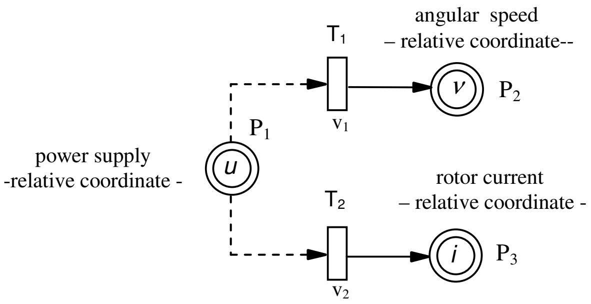


Fig. 5 MPN model of an DC electrical drive for $\mu_s = 0$

The P_1 place contains the power supply variable (input variable), written in relative coordinates. Its marking (1 for rated value) authorizes the continuous firing, with v_1 and v_2 speed values of two continuous transitions T_1 and T_2. In accordance with evolution rules of MPNs, during continuous firing of transitions, the P_1 marking is not modified. The P_2 place contains the angular speed variable (state/output variable), in relative coordinates. Similarly, in P_3 place the relative coordinate of rotor current variable (state variable) is indicated. The expressions of the two continuous transitions firing speeds v_1 and v_2 respectively were obtained directly from (4) and (5), according (2) and (3) rules:

$$v_1 = \frac{dn}{d\tau} = \varphi \cdot i \quad (6)$$

$$v_2 = \frac{di}{d\tau} = -\frac{T_m}{T_a} \cdot i - \frac{T_m}{T_a} \cdot \frac{\varphi}{\rho} \cdot \nu + \frac{T_m}{T_a} \cdot u \quad (7)$$

The basic topology of the MPN model is a modular structure, which can be developed by adding others functional Petri nets elements, in accordance with the real physical system represented. Hence, for illustrate the behavior of the DC motor in presence of the static torque M_s, a complementary sub – net can be connected, in addition at the initial model. The new MPN model obtained is a flexible framework, which allows analyzing and simulating the behavior of entire electrical drive in various situations and service operations (starting, regulating speed, electrical breaks) as well for many static torque expressions. (Fig.6) [8], [9], [10], [11], [12].

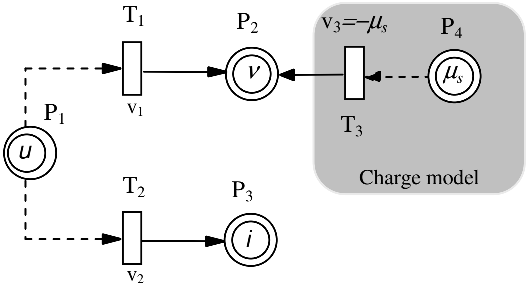


Fig. 6 MPN model of an DC electrical drive for $\mu_s = 1$

Each continuous transition of the model it's always active; it can be inactivated only by empty discrete input places. The test arc $P_1 - T_1$, $P_2 - T_2$ and $P_4 - T_3$ does not allow token flow. This makes it possible to model various kinds of systems without feedback: the token quantity of P_1 and P_4 is not influenced.

The DC electrical drive MPN model was synthesized and then its behavioral properties were verified by on-line simulation using Visual Object Net++ software tool [9], [10]. Also, the simulation results were compared with the laboratory experimental results or other sets of Matlab-Simulink models simulations, for the same scenarios and initial conditions. Hence, Fig.7 shows the evolution of the speed and rotor current, during the starting process by connecting directly of the motor at the rated power supply (the initial marking into P_1 is $u = 1$) for $M_s = M_N$ ($\mu_s = 1$), using an MPN model, for anDC motor drive with P_N = 2,64 kW; U_N = 110 V; I_N = 30 A; n_N = 1500 rot/min; R_a = 0,36 ohms; L_a = 0,125 H; J = 1,2 kgm^2.

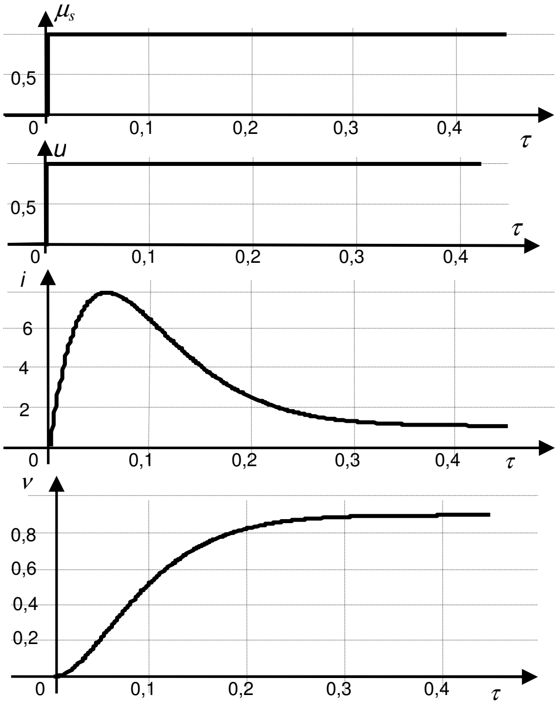


Fig. 7 On-line simulation results of the MPN model

For a linear dependence between the static torque of the motor and its angular speed ($\mu_s = k\cdot\nu$) the above structure (Fig.6) can be completed with two discrete transitions (T_4 and T_5), which induces minimal changes in the MPN topology (Fig.8). The T_4 transition is permanently enabled, and the weight of the $T_4 - P_4$ arc – $k\cdot\nu$ – ensures a continuous variation of the P_4 marking.

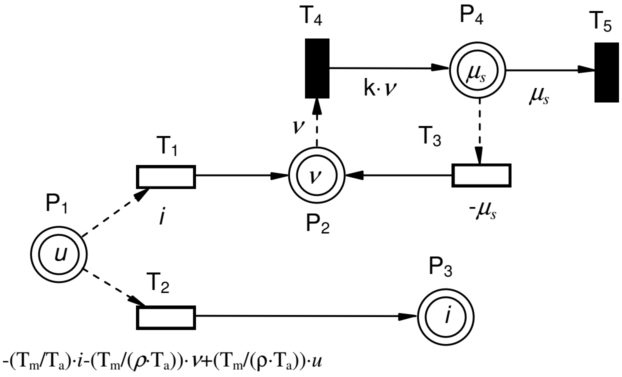


Fig. 8 DC electrical drive MPN model for $\mu_s = k\cdot\nu$

Object Petri Net paradigm

The most mathematical, textual or graphical approaches to describe hybrid systems are currently usable for small examples. Models of complex systems are unwieldy. Therefore a hierarchical concept to structure a model is needed. In order to solve the mentioned handling problems arising from the system complexity, in [8], [9], [10] is proposed a Object Oriented paradigm. The object-oriented concept unites the advantages of the modules and hierarchies and adds useful concepts like *reuse*, *encapsulation* and *information hiding*. Information hiding is realized by encapsulation the detail topology of the net and by publishing selected places, using an interface. In this way we get more flexibility of the whole model.

Using this concept, also the software tool facilities Fig.9 shows the Object Petri Net hierarchical structure of the electrical drive model (Fig.5); the marking of published interface places can be modified from the environment, or by interaction between many object nets.

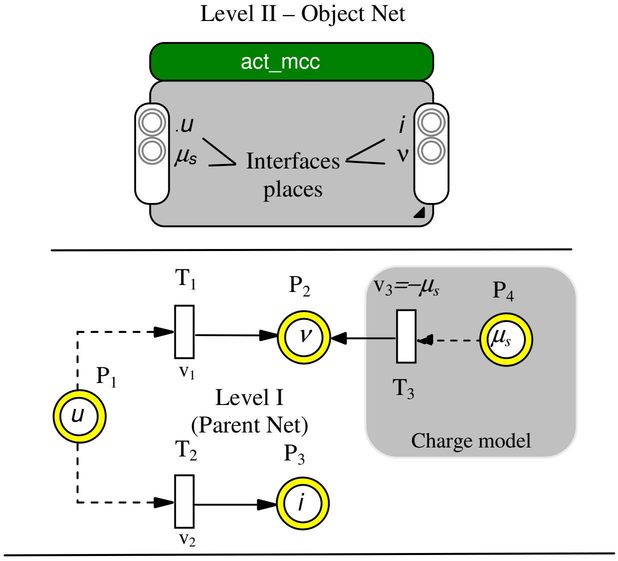


Fig. 9 The Object Net of the parent MPN model

IV. CONCLUSION

PNs are known to be powerful tools for modeling and analysis of discrete systems. The CPNs and HPNs, then DPNs and MPNs extensions of the basic models, allow modeling and analysis of continuous and hybrid systems on the same conceptual basis. Electrical drive systems involving both continuous and discrete variables, with an evolution often described by continuous models are such examples. The continuous dynamic is generally given by differential – algebraic equations and the discrete dynamic is generally modeled by automata or input – output transitions systems with a countable number of states.

Combining various synthesizing techniques, MPNs models of the DC electrical drives are a new alternative to the classical approaches based on Matlab – Simulink frameworks, not so accurate, but providing encouraging results in the future.

REFERENCES

[1] L. Cantemir, “Some principles problems concerning the electromechanical drive”, Proceedings of the 9th National Conference on Electrical Drives, Craiova, 8-9 October, pp.13-19, 1998.

[2] M. S. Branicky , “A Unified Framework for Hybrid Control□, Proc. 33rd Conference on Decision and Control, Lake Buena Vista, Fl., USA, pp. 4228 – 4234, 1994.

[3] R. David, H. Alla. Du Grafcet aux Réseaux de Petri, Hermes , Paris, 1992.

[4] R. David, H. Alla, “On Hybrid Petri Nets”, Discrete Event Dynamic Systems: Theory and Applications, No.11, pp.9 – 40, 2001.

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

[6] G. N. Davrazos, N. T. Koussoulas, “A General Methodology for Stability Analysis of Differential Petri Nets”, Proceedings of the 10th Mediteraneean Conference on Control and Automation – MED2002, Lisbon, Portugal, July 9-12, 2002.

[7] Isabel Demongodin, N.T. Koussoulas, “Differential Petri Net Models for Industrial Automation and Supervisory Control”, IEEE Transactions on Systems, Man and Cybernetics – Part C:Applications and Reviews, Vol.36, No.4, pp.543-553, 2006.

[8] R. Drath, “Hybrid Object Nets: An Object Oriented Concept for Modelling Complex Hybrid Systems”, Dynamical Hybrid Systems, ADPM98, Reims, 1998.

[9] R. Drath. Tool Visual Object Net ++, Available: http:// www.daimi.au.dk/ PetriNets/tools, section “Visual Object Net++”.

[10] M. A. Drighiciu, “Petri Net Formalism for a Possible Hierarchical Structure of Complex Hybrid Systems”, Proceedings of 6th International Carpathian Control Conference – ICCC 2005, Vol. II, Miskolc – Lillafured, Hungary, May 24 – 27, pp. 371 – 376, 2005.

[11] M. A. Drighiciu, Gh. Manolea, D. C. Cismaru, Anca Petrişor, “Hybrid Petri Nets as a New Formalism for Modeling Electrical Drives”, Proceedings of International Symposium on Power Electronics, Electrical Drives, Automation and Motion – SPEEDAM 2008, ISBN: 978-1-4244-1664-6, IEEE Catalog Number: CFP0848A-CDR, Ischia, Italy, 11th – 13th June, pp.626-631, 2008.

[12] M. A. Drighiciu, Gh. Manolea, Anca Petrisor, M. C. Popescu, “On Hybrid Systems Modeling with Petri Nets”, Recent Advances in System Science and Simulation in Engineering, Proceedings of the 7th WSEAS International Conference on System Science and Simulation in Engineering (ICOSSSE’08), Venice , 21- 23 November, pp.73-78, 2008.

[13] S. Engell, G. Frehse, E. Schneider (Coordinators). Modelling, Analysis, and Design of Hybrid Systems, Lecture Notes in Control and Information Sciences 279, Springer-Verlag, ISBN: 3-540-43812-2, pp.3-36, 2002.

[14] Emilia Villani, P. E. Miyagi, R. Valette. Modelling and Analysis of Hybrid Supervisory Systems; a Petri Net Approach, Springer-Verlag, ISBN 978-1-84628-650-6, pp. 1-13, 2007.

[15] J. Zaytoon (Coordinator). Systèmes dynamiques hybrides, Hermes-Science,2001.