Modeling Continuous Systems Using Modified Petri Nets Model

Abbas Dideban, Alireza Ahangarani Farahani, and Mohammad Razavi

Abstract

Due to the changes which may occur in their parameters, systems are usually demonstrated by some subsystems for different conditions. This paper employs Modified Petri Nets (MPN) to model theses subsystems and makes it simple to analyze them. In this method, first, the continuous transfer function is converted to a discrete transfer function and then, by MPN, system is modeled and analyzed. All subsystems can be modeled and used in state control or cascade control loops. Here, the focus is mainly on the conception as well as the definitions of the new unified representation model for continuous control systems. Simulation results show that the new method for modeling continuous systems works effectively.

Keywords—Petri Net, Modified Petri Nets, Transfer function, Continuous Petri Nets.

I. INTRODUCTION

In the past few years, due to great advances in technology and computer, modeling based on Petri Nets have attracted researchers' attention. Automata and Petri Nets are the main modeling tools in the research area of control synthesis for discrete event system such as aviation, spaceflight, correspondence computer integrated manufacture system etc. Recently, many researchers study modeling based on Petri Nets because of the advantages of the graphical and distributed representations of the system states and the computational efficiencies [1].

Petri Nets were introduced in Carl A. Petri's 1962 Ph.D. dissertation [2]. Petri Nets have been used extensively as a tool for modeling, analysis and synthesis for discrete event systems [3]. It is usually interpreted as a control flow graph of the modeled system [4]. Petri Nets are an alternative tool for modeling of systems [5]. This tool is powerful and useful for study and analysis of discrete event systems. They are both a graphical and mathematical tool that can model deterministic or stochastic system behaviors and phenomena such as parallelism, asynchronous behavior, conflicts, resource sharing and mutual exclusion [2].

Modeling based on Petri Nets can describe system behaviors by linear algebraic equations [5].

The Continuous Petri Nets model is presented by R. David and H. Alla in [6]. These authors have obtained a continuous model by fluidization of a discrete Petri Net. Further Continuous Petri Nets constitute part of process modeling made by systematic procedure that is discussed in [7]. The continuous part can model systems with continuous flows. Autonomous Continuous Petri Nets and other models like Differential Algebraic Equations Petri Nets have been studied intensively since the advent of this research area [8, 9]. Digitized signals leaning on analogue signal and continuous approximated model is used in this paper and the important connection between mathematical transfer function and state event base model has been presented in [10]. This leads to mathematical model with very simple algorithm in contradiction to complexity of mathematical models.

In many systems, differential equations of systems change due to condition variation. Hence, there are different transfer functions and system can be divided into some subsystems; for instance some hybrid aeronautic systems tend to switch between different dynamic equations as a result of parameter or operation environment changes. Modeling and control of these systems for control is difficult but very important. MPN can model these systems. In MPN method, continuous transfer function is modeled by Petri Nets approach. For use of MPN some definitions for modeling are presented. In all Petri Nets modeling, event is the base of Petri Nets but in modified theory, delay presents the rule of event. Connection between discrete and continuous Petri Nets and lead to MPN is the principal element of this paper [11]. Additionally, a Petri Nets model can be described by a linear algebraic equation, or other mathematical models reflecting the behavior of the system. This opens a possibility for the formal analysis of the model. By this, modeling is represented that all of features of system is preserved in incidence matrix.

In this paper a continuous model has been derived from dynamics of system by this supposition that all signals are continuous and Petri Nets has been determined by a digitized transfer function. The goal of this document is to improve a new approach to continuous modeling.

Manuscript received July 14, 2014; revised December 29, 2014; accepted January 14, 2015.

A. Dideban, Department of Electrical and Computer Engineering Semnan University, Semnan, Iran (e-mail: adideban@semnan.ac.ir).

A. Ahangarani Farahani, Department of Electrical and Computer Engineering, Semnan University, Semnan, Iran (e-mail: a.ahangarani@students.semnan.ac.ir).

M. Razavi, Department of Electrical and Computer Engineering Semnan University, Semnan, Iran (e-mail:mohammad_rzv@yahoo.com).

The remainder of this paper is organized as follows: In Section 2, main concepts of discrete and continuous Petri Netsis presented. The basic definitions and method of MPN is explained in section 3.The advantage of continuous Petri Nets over MPN is also described in this section. To convince the reader about utility of MPN, basic examples are given in section 4.Section 5 is dedicated to simulation results and finally the conclusion is stated in section 6.

II. Basic Concepts and Notation

A. Ordinary Petri Nets

Petri Net is a directed net consisting of place, transition, directed arc and token [12].

A Petri Net is a 5-tuple $N = \{P, T, W^-(Pre), W^+(Post), M_0\}$ where

$P= \{p_1, p_2,...,p_n \}$ is a finite set of places, and $n>0$ is the number of places.

$T= \{t_1, t_2,..., t_m \}$ is a finite set of transitions, and $m>0$ is the number of transitions and $P \cap T=\emptyset$, i.e. the sets P and T are disjointed.

Pre or $W^-: (P\times T) \to N$ is the input function, *Post* or $W^+: (T\times P) \to N$ is the output function.

M_0 is the initial marking. The incidence matrix W is calculated by $W=W^+ - W^-$.

A transition can only be fired if each of the input places of this transition contains at least one token.

Here the following notations will be used [13]:

${}^{\circ}T_j = \left\{ P_i \in P \middle| \Pr e\left(P_i, T_j\right) > 0 \right\}$ = set of input places of T_j.

$T_j^{\circ} = \left\{ P_i \in P \middle| \mathrm{P}ost\left(P_i, T_j\right) > 0 \right\}$ = set of output places of T_j.

$P_i^{\circ} = \left\{ T_j \in T \middle| \Pr e\left(P_i, T_j\right) > 0 \right\}$ =set of input transitions of P_i. $P_i^{\circ} = \left\{ T_j \in T \middle| \mathrm{P}ost\left(P_i, T_j\right) > 0 \right\}$ = set of output transitions of P_i.

The dynamic facets of Petri Net models are characterized by certain markings. These markings are projects of token to the places of a Petri Net. Markings may alter during the execution of a Petri Net, which is controlled by the number and distribution of tokens. A transition is enabled if and only if each of its input places includes certain number of tokens. When a transition is enabled, it may fire. As soon as a transition fires, all enabling tokens are removed from its input places and then a token is transferred to each of its output places [14].

B. Continues Petri Nets

A marked Continuous Petri Nets is a set of

$R = \{P, T, \Pr e, Post, M_0\}$

Such that P, T, *Pre* and *Post* are the same that mentioned in earlier section and M_0 is the initial marking of all places knowing that $M(t)$ denotes the marking at time $\boldsymbol{t}$. It shall be mentioned that P is positive real number.

The important difference between ordinary Petri Nets and Continuous Petri Nets (CPN) is enabling degree. Enabling degree of a transition T_j for marking denotes by q or $q(T_j,m)$ is the real number that is shown in the equation below [6]:

$$q\left(T_j, m\right) = \min_{i:P_i \in {}^{\circ}T_j} \left(\frac{m\left(P_i\right)}{\Pr e\left(P_i, T_j\right)} \right) \tag{1}$$

If $q > 0$, transition T_j is enabled; it is said to be q-enabled. It is important to note that the marking of a Continuous Petri Nets can take real positive values, while in discrete Petri Nets only integer values are possible. In fact, this is the only difference between a continuous and a discrete Petri Nets [15].

Timed Petri Nets with constant times associated either with places or with transitions are used in order to model various systems. A timed Continuous Petri Net is a pair (R, *Spe*) such that: R is a marked autonomous Continuous Petri Net; *Spe* is a function from the set T. For T_j, $Spe(T_j)=V_j$ is the maximal speed associated with transition T_j. The values $v_j(t)$is called, the instantaneous firing speed of the transition T_j.

The concept of validation of a continuous transition is different from the traditional concept met in discrete Petri Nets. The fundamental equation for a timed Continuous Petri Nets between times (t and $t+dt$), the quantity of the firing T_j being $v_j.dt$ is as follows [6]:

$$\begin{aligned} m\left(t+dt\right) &= m\left(t\right) + W.v\left(t\right)dt \Rightarrow \\ m\left(t_2\right) &= m\left(t\right) + W.\int_{t_1}^{t_2} v\left(t\right)dt \end{aligned} \tag{2}$$

where W is the Petri Net incidence matrix, $v(t)$ is the characteristic vector of s The characteristic vector s of a firing sequence S is a vector that each component is an integer corresponding to the number of firings of the corresponding transition. $m(t+dt)$ and $m(t)$ are the corresponding new markings and previous markings respectively.

C. State Equation

Dynamic behavior of the system represented by the Petri Net can be expressed using the Petri Net incidence matrix W in which W is an $n\times m$ matrix.

It is desirable to have an equation to test if a given marking M_k is reachable from an initial marking M_0. Suppose that M_k is reachable from M_0 by successive firing of certain sequences. Then [16]:

$$\begin{aligned} &M_1 = M_0 + W.U_0 \\ &M_2 = M_1 + W.U_1 \Rightarrow \\ &M_2 = M_0 + W.U_0 + W.U_1 \Rightarrow \\ &M_2 = M_0 + W\left(U_0 + U_1\right) \\ &\vdots \\ &M_k = M_{k-1} + W.U_{k-1} \Rightarrow \\ &M_k = M_0 + W\left(U_0 + U_1 + \cdots + U_{k-1}\right) \end{aligned} \tag{3}$$

Using equations in (3) it is easy to show that the state equation is:

$$M_k = M_0 + W.U \tag{4}$$

here, U is the summation of all U_i ($i=m,1,\ldots,k-1$).

III. Modified Petri Nets (MPN)

The transfer function of systems can be determined by system identification techniques. The transfer functions for continuous systems are demonstrated by differential equations. It is very difficult to show differential equations with Petri Nets model. Implementation of derivation function with Petri Nets, results a very giant and complex model that spoil all advantages of Petri Nets like series, visual presentation, comprehensible model etc. To overcome this problem, difference equations can be used, instead of differential equations. Continuous equation shall be digitized with adequate sample time. In difference equation recursive function with delays used instead of derivative function.

In this method time delays will be implemented by transitions and places play the rule of input and output for systems and maybe its inner dynamics.

For modeling there are some new definitions:

Definition 1: Marking of places in Modified Petri Nets, can be negativeor non-negative real numbers at any time.

Definition 2: Transitions are enabled always, if $M(P_i)>0$ or $M(P_i)<0$.

Definition 3: Speeds associated with transitions are infinity.

Definition 4: After firing the transitions the marking of placesreach zero.

Consider a first-order transfer function with one pole and no zeros as

$$F(s) = \frac{A}{s+B} \tag{5}$$

Discrete-time model of the transfer function in (5) with sample time of T_S is

$$F(z) = \frac{a}{z-b} \tag{6}$$

The relationship between input-output of the (6) can be described by

$$y(n) = ax(n-1) + by(n-1) \tag{7}$$

Petri Nets model of (7) is shown in Figure 1. In Figure 1 places P_1 and P_2 depict input variable and output variable respectively. Moreover, in this figure state equations can be written as follows

$$m(n) = m_0 + \sum W.v \tag{8}$$

where

$$W = \begin{bmatrix} 0 & 0 \\ a & b-1 \end{bmatrix}$$

And

$$v = \begin{bmatrix} m(P_1(n-1)) \\ m(P_2(n-1)) \end{bmatrix}$$

IV. Simulation Results

In this section, performance of MPN for modeling continuous systems with differential equation is depicted. Simulation is carried out using MATLAB/SIMULINK version 7.12.0.635. Here, the start and end times of simulation are 0 and 100 (s) respectively. Also the following transfer function is used in simulation.

$$G(s) = \frac{s+1}{s^2 + 2s + 10} \tag{9}$$

After discretizing the system using the triangle (first-order hold) approximation with sample time $T_s=0.1(S)$, the resulted transfer function is

$$\frac{Y(z)}{U(z)} = g(z) = \frac{-0.08479z^{-2} + 0.09378z^{-1}}{1 - 1.729z^{-1} + 0.8187z^{-2}} \tag{10}$$

Then difference equation of (10) can be expressed as follows

$$\begin{aligned} y(k) = &-0.08479u(k-2) + 0.09378u(k-1) \\ &-0.8187y(k-2) + 1.729y(k-1) \end{aligned} \tag{11}$$

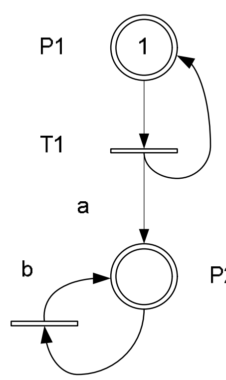

Fig. 1. Petri Nets model of equation (7).

Consequently, the state equation of (11) can be written

$$\begin{cases}\begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -0.8187 & 1.729 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u(k-1) \\ y(k) = \begin{bmatrix} -0.08479 & 0.09378 \end{bmatrix} x(k) \end{cases} \tag{12}$$

$$A = \begin{bmatrix} 0 & 1 \\ -0.8187 & 1.729 \end{bmatrix} \qquad B = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

$$C = \begin{bmatrix} -0.08479 & 0.09378 \end{bmatrix} \qquad D = 0$$

Finaly, Petri Nets of this model has been demonstrated in Figure 2.

In Figure 2P_1 and P_3 indicate input variable and output variable respectively. The incidence matrix W for Figure 2 can be obtained as follows

$$W^+ = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0.09378 & -0.08479 & 1.729 & -0.8187 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

$$W^- = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \Rightarrow$$

$$W = W^+ - W^- = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & -1 & 0 & 0 \\ 0.09378 & -0.08479 & 0.729 & -0.8187 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

And state equation is obtained by (13)

$$m = m_0 + Wv \tag{13}$$

And here

$$v = \begin{bmatrix} m(P_1(n-1)) \\ m(P_2(n-1)) \\ m(P_3(n-1)) \\ m(P_4(n-1)) \end{bmatrix}$$

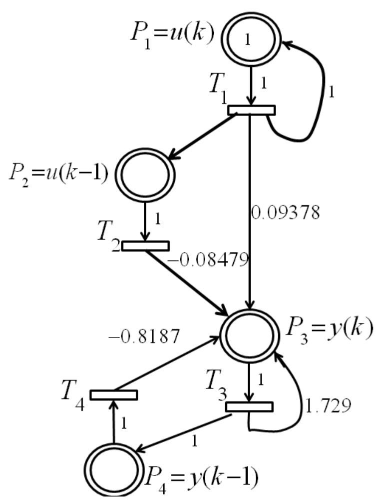

Fig. 2. Petri Nets Model of (11).

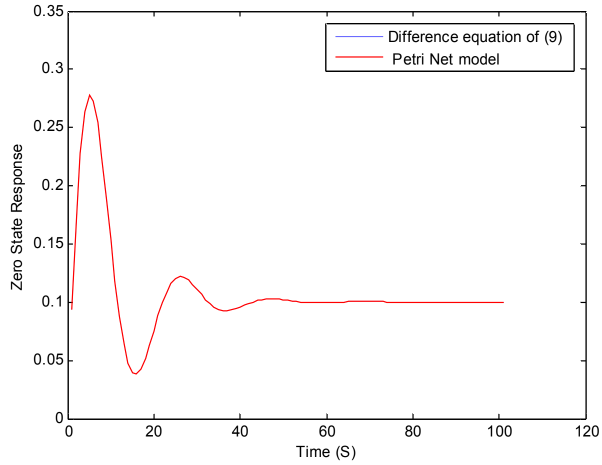

Fig. 3. Zero state response for difference equation of (10) and Petri Nets model

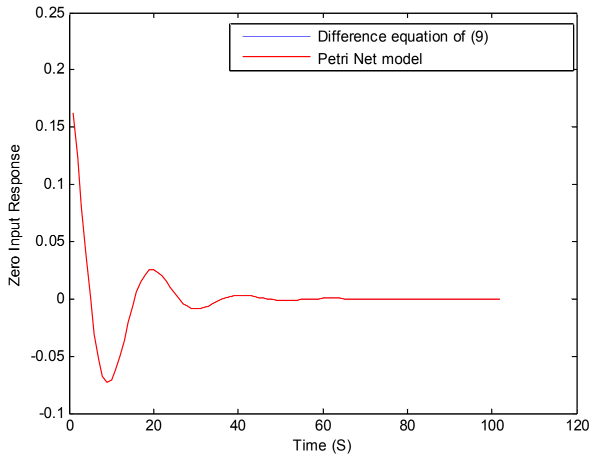

Fig. 4. Zero input response for difference equation of (10) and Petri Nets model.

Furthermore, model is simulated in zero state and zero input. In the zero state, input is step. Figure 3 shows zero state response for difference equation of (10) and Petri Nets model

In Figure 4 shows a simulation for difference equation of (10) and Petri Nets model based on the zero input.

Comparing responses in Figure 3 and 4 for difference equation and Petri Nets Model, it is obvious that Petri Nets model response is similar to difference equation. Therefore, the MPN can be substituted by difference equation.

V. CONCLUSION

Petri Nets makes it possible to model all continuous systems with very simple rules and prepare a visual model with its dynamics. Modified Petri Nets (MPN) can model systems with variable parameters or multiple subsystems easily. The system analysis and controller design using MPN is comfortable. In the resulted model, all eigenvalues can be extracted to be used in cascade control systems, although there is an error in calculation due to sample time selection an d estimation of equalization *P(n-1)* and *P(n)*. By using MPN, we present a visual and systematic method for dynamics system modeling. In hybrid systems with several dynamic behavior modes, the provided visual model gives a useful overview of the systems. Further works will focuses on controller design or fault detection that includes design; implement and analysis of controller for continuous system in Petri Nets environment.

REFERENCES

[1] T.Ze, X.Liyang and L.Di,"Controller Design of DES Petri Nets with Mixed Constraint," *Chinese Journal of Aeronautics*, vol. 18, no. 1, August 2005, pp. 283- 288.

[2] R. Zurawski and M. Zhou,"Petri Nets and Industrial Applications: A Tutorial, "*IEEE. Transactions on Industrial Electronics*, Vol. 41, No. 6, pp.567-583, December 1994.

[3] X.D. Koutsoukosand P.J. Antsaklis, "Hybrid Control Systems Using Timed Petri Nets: Supervisory Control Design Based on Invariant Properties," Hybrid Systems V , P. Antsaklis, W. Kohn, M. Lemmon, A. Nerode, S. Sastry Eds, Lecture Notes in Computer Science, LNCS 1567, Springer-Verlag, pp.142-162, 1999.

[4] Krzysztof Sacha," Fault Analysis Using Petri Nets," IEEE Real-time Embedded Systems Workshop, pp. 130-133, December 2001.

[5] A.Dideban, M.Zareiee and H.Alla,"Controller synthesis with very simplified linear constraints in PN model,"2nd IFAC Workshop on Dependable Control of discrete Systems, Bari, Italy, 2009.

[6] H. Alla and R.David,"A modeling and analysis tool for discrete events systems: continuous Petri net," *Performance Evaluation*, Vol.33, pp.175-199, 1998.

[7] Y.F.Wang and C.T.Chang,"A hierarchical approach to construct petri nets for modeling the fault propagation mechanisms in sequential operation," *Computer and Chemical Engineering*, pp. 259-280, 2003.

[8] L.Recalde, E.Teruel and M.Silva, "Autonomous C.: Application And Theory Of Petri Nets," 20thInternational Conference, Icatpn'99, Williamsburg, Virginia, USA,Vol. 1630, *Springer Verlag*, pp. 107–126, 1999.

[9] R.Champagnat, R.Valette, J.C.Hochon and H.Pingaud, "Modelling, Simulation And Analysis Of Batch Production Systems", *Discrete Event Dynamic Systems: Theory and Applications*, Vol. 11, pp. 118-136, January-April 2001.

[10] C.R.V´Azquez, L.Recalde and M.Silva, "Stochastic Continuous-State Approximation of Markovian Petri Net Systems," 47thIEEE Conference on Decision and Control Cancun, Mexico, Dec. 9-11, 2008.

[11] T.Gu and R.Dong," A Novel Continuous Model To Approximate Time Petri Nets: Modelling And Analysis," *Int. J. Appl. Math. Computer Science*, Vol.15, No.1, pp.141–150, 2005.

[12] R. David and H. Alla,"Discrete, Continuous, and Hybrid Petri Nets," *Springer-Verlag Berlin Heidelberg*, Printed in Germany, 2005.

[13] N. G. Leveson and J. L. Stolzy, "Safety Analysis Using Petri Nets," *IEEE Trans. on Software Engineering*, Vol. SE-13, NO. 3, March 1987.

[14] A. Dideban, M. Kiani and H. Alla, "Implementation of Petri Nets Based Controller using SFC," *Control Engineering and Applied Informatics*, Vol. 13, No. 4, pp. 82-92, 2011.

[15] M. Kloetzer, C. Mahulea, C. Belta, L. Recalde and M. Silva," Formal Analysis of Timed Continuous Petri Nets," 47th IEEE Conference on Decision and Control, Cancun, Mexico, pp. 245-250, December 2008.

[16] A.A. Desrochers and R.Y.Al-Jaar, "Application of petri nets in manufacturing system: *modeling, control, and performance analysis*," IEEE Press, January 1994.