

## Continuous-Time Delay-Petri Nets as A New Tool to Design State Space Controller

Alireza Ahangarani Farahani, Abbas Dideban

<sup>1</sup>*Semnan University, Department of Electrical and computer Engineering,  
Semnan, Iran  
e-mail: adideban@semnan.ac.ir*

**crossref** <http://dx.doi.org/10.5755/j01.itc.45.4.13665>

**Abstract.** In this paper a new method of modelling and controller design for continuous systems is introduced. Petri Net is a useful tool for modelling, analysis and controller synthesis in discrete event systems. Continuous-Time Delay-Petri Nets (CTDPN) are presented to challenge with other types of continuous dynamic system modelling tools. This article focuses on an approach to design a controller using Petri Net for continuous dynamic systems. It is shown here that this method simplifies system analysis and controller design by first, converting the continuous state space to a discrete state space and then applying CTDPN, to model and analyse it. Finally, a state feedback control algorithm is adapted to be used for models which are derived by CTDPN approach. Simulation results show the efficiency of the proposed approach comparing to state space feedback controller in terms of the simplicity of the system analysis and the time it takes to simulate the close-loop system.

**Keywords:** Continuous Petri Nets; Controller Design; State Feedback; Time Delay Petri Nets.

### 1. Introduction

Recently, due to great advances in technology and computer aided techniques, modelling based on Petri Nets has attracted researchers' attention. Automata and Petri Nets are the main modelling tools in the area of control synthesis for discrete event systems in computer integrated manufacturing systems, correspondence, aviation, spaceflight, etc. In the last decade, many researchers study modelling based on Petri Nets because of the advantages of the graphical and distributed representations of the system states and the computational efficiencies [1,2].

Petri Nets have been used extensively as a tool for modelling, analysis and synthesis of discrete event systems and usually interpreted as a control flow graph of the modelled system [3]. This method is an alternative tool for modelling event based systems [4]. Petri Nets are both a graphical and mathematical tool that can model deterministic or stochastic system behaviours and phenomena such as parallelism, asynchronous behaviour, conflicts, resource sharing and mutual exclusion [5].

Petri Nets were introduced in Carl A. Petri's 1962 Ph.D. dissertation [6]. Since then, researches have proved it to be a helpful graphical and mathematical modelling tool applicable to many systems. As a graphical tool, Petri Nets can be used as an effective

visual communication aid similar to flow charts, block diagrams, and networks [6]. As a mathematical tool, the modelling procedure based on Petri Nets can describe the system behaviour by linear algebraic equations [7].

Continuous Petri Nets model was presented by David and Alla in [8]. These authors have obtained a continuous model by fluidization of a discrete Petri Nets [9]. This approach is a useful method for modelling of many phenomena and systems. In [10], controllability of Timed Continuous Petri Nets, under infinite server semantics, with uncontrollable transitions is presented. Further, Continuous Petri Nets constitute a part of process of modelling by a systematic procedure that was discussed in [11]. 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 in this research area [12, 13]. Digitized signals leaning on analogue signal and continuous approximated model are used in this paper which presents important connection between mathematical transfer function and state event base model [14]. This leads to a mathematical model with very simple algorithm in contrast with the complexity of mathematical models.

The classical control theory is based on a simple input-output description of the plants which are

usually expressed as a transfer function and never use any knowledge of the interior structure of the plant.

Modern control theory solves many of the limitations by using a much richer description of the system dynamics. The so-called state space description represents the dynamics as a set of coupled first-order differential equations between a set of internal variables known as state variables, together with a set of algebraic equations that combine the state variables into physical output variables [15].

For many systems, the differential equations of the system change due to the change of conditions. Hence, there are different transfer functions and the system can be divided into a number of subsystems. Therefore, they are absolutely difficult to be modelled and controlled. One of the most powerful tools to model and control these types of systems is Petri Nets. Some previous attempts are done in order to model continuous dynamic systems using Petri Nets [16].

Here in this article a Continuous-Time Delay-Petri Nets (CTDPN) method is presented to solve the difficulty of modelling and control. Using CTDPN, first continuous state space is modelled by Petri Nets approach. After modelling step, a state feedback control algorithm based on CTDPN is introduced. One of the significant advantageous of this procedure is that all features of the system are preserved in incidence matrix.

The remainder of this paper is organized as follows: In Section 2, main concepts of discrete and continuous Petri Nets are presented. The basic definitions for Continuous-Time Delay-Petri Nets (CTDPN) are explained in Section 3. The differences between CTDPNs with PNs are also explained in this section. The control algorithm based on state space controller is presented in Section 4. Section 5 is dedicated to simulation results and finally the conclusion is stated in Section 6.

## 2. Basic Concepts and Notations

### 2.1. Ordinary Petri Nets

Petri Net is a directed net consisting of places, transitions, directed arcs and tokens [17].

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

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

$T = \{t_1, t_2, \dots, t_m\}$  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 disjoint;

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

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

Pictorially, places are represented by circles, transitions are represented by rectangles or bars, and arcs are depicted as arrows.

Here, the following notations will be used [18]:

${}^o t_j = \{p_i \in P \mid Pre(p_i, t_j) > 0\}$  = set of input places of  $t_j$ .

$t_j^\circ = \{p_i \in P \mid Post(p_i, t_j) > 0\}$  = set of output places of  $t_j$ .

${}^\circ p_i = \{t_j \in T \mid Pre(p_i, t_j) > 0\}$  = set of input transitions of  $p_i$ .

$p_i^\circ = \{t_j \in T \mid Post(p_i, t_j) > 0\}$  = set of output transitions of  $p_i$ .

The dynamic behaviour of Petri Net models is characterized by certain markings. Markings may be altered during the execution of a Petri Net, which are 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 the tokens are transferred to each of its output places [19]. When using Petri nets to model systems, places represent states and transitions are used to represent events [20].

Dynamic behaviour of the system represented by the Petri Nets can be expressed using the Petri Nets incidence matrix  $W$  where  $W$  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 [21]:

$$M_k = M_0 + W(U_0 + U_1 + \dots + U_{k-1}). \quad (1)$$

Where  $U_i$  is the firing transition of  $t_i$ . Using equation (1), it is easy to show that the state equation is:

$$M_k = M_0 + W.U \quad (2)$$

where,  $U$  is the firing count vector and is equal to summation of all  $U_i (i=0,1,2,\dots,k-1)$ .

### 2.2. Continuous Petri Nets

A marked Continuous Petri Net is a 5-uple  $R = \{P, T, Pre, Post, M_0\}$  such that  $P$ ,  $T$ ,  $Pre$  and  $Post$  are the same as mentioned in previous subsection, and  $M_0$  is the initial marking of all places knowing that  $M(t)$  denotes the vector marking at time  $t$ , including elements  $m(p_1), \dots, m(p_n)$  in which  $n$  shows the number of places. It shall be mentioned that  $m(p_i)$  is a positive real number tokens [21].

An important difference between the ordinary Petri Net and the Continuous Petri Net (CPN) is the enabling degree. The enabling degree of a transition  $t_j$  for a marking  $M$  is indicated by  $q$  or  $q(t_j, M)$  which is the real number as shown in the equation below [22]:

$$q(t_j, M) = \min_{i: p_i \in {}^o t_j} \left( \frac{m(p_i)}{Pre(p_i, t_j)} \right) \quad (3)$$

If  $q > 0$ , the transition  $t_j$  is enabled; it is said to be  $q$ -enabled. It is important to note that the marking of a Continuous Petri Net 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 Net [23].

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; and  $Spe$  is a function from the set  $T$ . For transition  $t_j$ , the values  $v_j(t)$  denote instantaneous firing speed and  $Spe(t_j) = V_j$  is the maximal speed. The instantaneous firing speed  $v_j(t)$  satisfying the following conditions:

$$v_j(t) \leq V_j(t).$$

The concept of the 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  $t+dt$ ) is as follows:

$$\begin{aligned} M(t+dt) &= M(t) + W v(t) dt \Rightarrow \\ M(t_2) &= M(t_1) + W \int_{t_1}^{t_2} v(t) dt \end{aligned} \quad (4)$$

where  $W$  is the Petri Nets incidence matrix,  $v(t)$  is the characteristic vector of  $s$  (firing sequence),  $M(t+dt)$  and  $M(t)$  are the new marking and the previous marking, respectively. Additionally, in equation (4), time  $t_1$  is smaller than time  $t_2$ .

### 3. Continuous-Time Delay-Petri Nets (CTDSPN)

The transfer function or state space of systems can be determined by the system identification techniques. The state space for a continuous system is indicated through differential equations. It is so difficult to show differential equations with ordinary Petri Nets model. To overcome the problem, CTDPN has been proposed. Continuous equation shall be digitized with adequate sample time. In discrete state space, a recursive function with delays is used instead of derivative function, making modelling much simpler. In CTDPN, transition firing plays the role of delay. For this innovation, there should be some new hypotheses:

**Hypothesis1.** Place tokens in CTDPN can be negative or non-negative real numbers at any time.  $\square$

**Hypothesis2.** A transition is enabled if  $m(p_i) > 0$  or  $m(p_i) < 0$ .  $\square$

**Hypothesis3.** The speed of transitions is infinity.  $\square$

**Hypothesis4.** When transitions are fired, values of tokens of input places tokens become zero.  $\square$

In this method, time delays will correspond to transitions, and places play the role of input variable, output variable and states for systems.

To illustrate this approach, consider a first-order state space as

$$\begin{aligned} x(k) &= Ax(k-1) + Bu(k-1) \\ y(k) &= x(k). \end{aligned} \quad (5)$$

The CTDPN model of equation (5) with a unit step input is shown in Fig. 1.

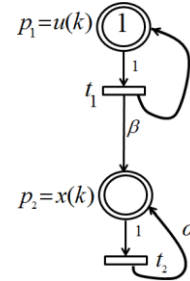


Figure 1. Petri Nets model of equation (5)

In Fig. 1, places  $p_1$  and  $p_2$  depict the input variable and the output variable, respectively.

Now consider strictly proper transfer function with  $n$  poles and  $m$  zeros as:

$$F(s) = \frac{b_m s^m + b_{m-1} s^{m-1} + \dots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \dots + a_1 s + a_0}. \quad (6)$$

Discrete-time model of the transfer function in equation (6) with sample time of  $T_s$  is:

$$F(z) = \frac{\beta_{n-1} z^{-1} + \dots + \beta_0 z^{-n}}{1 + \alpha_1 z^{-1} + \dots + \alpha_n z^{-n}}. \quad (7)$$

The discrete state space of the equation (7) is:

$$\begin{cases} x(k) = Ax(k-1) + Bu(k-1) \\ y(k-1) = Cx(k-1) \end{cases} \Rightarrow \quad (8)$$

$$\begin{aligned} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_{n-1}(k) \\ x_n(k) \end{bmatrix} &= \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \dots & 0 \\ \vdots & \vdots & \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 1 \\ -\alpha_1 & -\alpha_2 & -\alpha_3 & \dots & -\alpha_{n-1} & -\alpha_n \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ \vdots \\ x_{n-1}(k-1) \\ x_n(k-1) \end{bmatrix} \\ &+ \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} u(k-1) \\ y(k-1) &= [\beta_0 \quad \beta_1 \quad \beta_2 \quad \dots \quad \beta_{n-2} \quad \beta_{n-1}] \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ \vdots \\ x_{n-1}(k-1) \\ x_n(k-1) \end{bmatrix}. \end{aligned}$$

Petri Nets model of equation (8) is shown in Fig. 2.

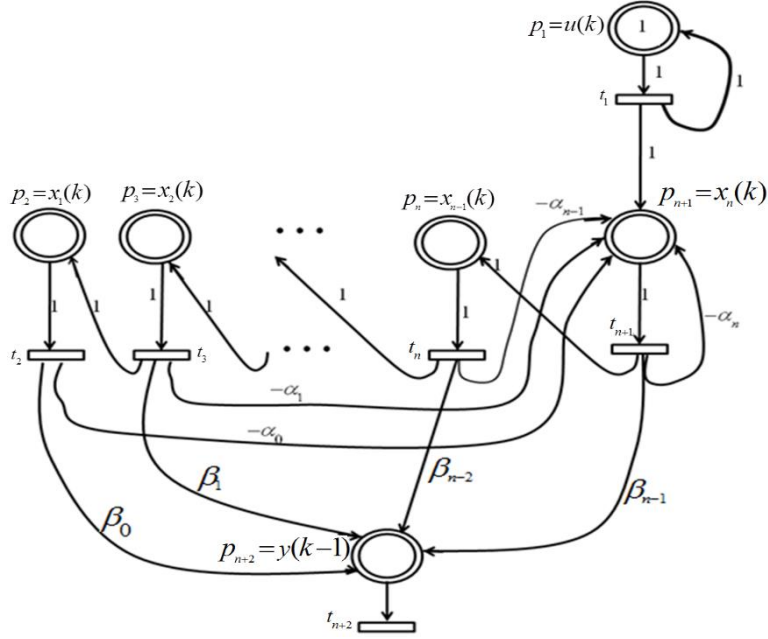


Figure 2. Petri Nets model of discrete state space described by equation (8)

In Fig. 2, places  $p_1$  and  $p_{n+1}$  depict input variable and output variable, respectively. Here,  $p_2$  to  $p_n$  correspond to  $x_1, \dots, x_n$ , respectively. Moreover, considering this figure, the state equation can be written as follows:

$$m(k) = m_0 + \sum W.v \quad (9)$$

where the dot signifies the multiplication between the matrix  $W$  and the vector  $v$ , also  $W$ ,  $W^+$ ,  $W^-$  and  $v$  are as follows:

$$v = \begin{bmatrix} m(p_1(k-1)) \\ m(p_2(k-1)) \\ \vdots \\ m(p_{n+1}(k-1)) \\ m(p_{n+2}(k-1)) \end{bmatrix}$$

$$W^+ = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \dots & 0 \\ \vdots & 0 & 0 & \dots & 1 & \vdots \\ 1 & -\alpha_1 & -\alpha_2 & \dots & -\alpha_n & 0 \\ 0 & \beta_0 & \beta_1 & \dots & \beta_{n-1} & 0 \end{bmatrix}$$

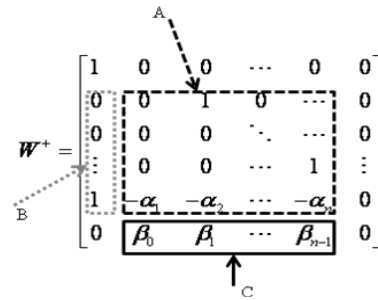
$$W^- = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & 0 & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$W = W^+ - W^- = \begin{bmatrix} 0 & 0 & 0 & \dots & 0 & 0 \\ 0 & -1 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \dots & 0 \\ \vdots & 0 & 0 & \dots & 1 & \vdots \\ 1 & -\alpha_1 & -\alpha_2 & \dots & -\alpha_n - 1 & 0 \\ 0 & \beta_0 & \beta_1 & \dots & \beta_{n-1} & -1 \end{bmatrix}$$

and

The role of principal equation of Petri net is the same as that of discrete state space equation.

It is obvious that the properties of the discrete state space are reserved in  $W^+$  matrix, because:



**Property 1.** The state space eigenvalues can be obtained by calculating the eigenvalues of a  $W^+$  matrix and omitting the values equal to 0 and 1 among them.  $\square$

**Proof.** The characteristic equations of  $W^+$  are  $\det(zI - W^+) = 0$ . This function is a polynomial of degree  $n+2$ .

$$\det(zI - W^+) = \det \begin{bmatrix} z-1 & 0_{1 \times n} & 0 \\ -B_{n \times 1} & zI_{n \times n} - A_{n \times n} & 0 \\ 0 & -C_{1 \times n} & z \end{bmatrix} \quad (10)$$

$$= (z-1) \times \det(zI_{n \times n} - A_{n \times n}) \times z = 0$$

Therefore, the eigenvalues of  $W^+$  are

$$\det(zI - W^+) = (z-1) \times \det(zI_{n \times n} - A_{n \times n}) \times z = 0 \Rightarrow$$

$$\begin{cases} z-1=0 \\ z=0 \end{cases} \Rightarrow \det(zI_{n \times n} - A_{n \times n}) = 0$$

$$\begin{cases} z=1 \\ z=0 \end{cases} \\ f_{system}(z) = \det(zI_{n \times n} - A_{n \times n}) = 0.$$

So, the properties of the discrete state space are kept in continuous -Time Delay- Petri Nets modelling.  $\square$

#### 4. State Space Controller Design Algorithm

The state of a dynamical system is the set of variables known as state variables that fully describe the system and its response to any given set of inputs [24].

State variable feedback allows the flexible selection of linear system dynamics. State feedback involves the use of the state vector to compute the control action for specified system dynamics.

The equations for the linear system and the feedback control law are, respectively, given by the following equations:

$$\begin{aligned} x(k+1) &= Ax(k) + Bu(k) \\ y(k) &= Cx(k) \\ u(k) &= -Kx(k) + r(k). \end{aligned} \quad (11)$$

The two equations can be combined to yield the closed-loop state equation:

$$\begin{aligned} x(k+1) &= Ax(k) + B[-Kx(k) + r(k)] \Rightarrow \\ x(k+1) &= (A - BK)x(k) + Br(k). \end{aligned} \quad (12)$$

The closed-loop state matrix is defined as:

$$A_{cl} = A - BK. \quad (13)$$

The dynamics of the closed-loop system depends on the eigenvalues and eigenvectors of the matrix  $A_{cl}$ . Thus, the desired system dynamics can be achieved with appropriate choice of the gain matrix  $K$ .

To design a feedback control law based on Petri Nets model using the proposed algorithm, following steps should be considered:

#### Algorithm 1.

**Step 1.** Calculate open loop poles of the system using  $W^+$  and Property 1.

$$\det(zI - W^+) = 0.$$

**Step 2.** Design a controller such that the closed loop poles are at certain desired locations. The desired pole locations are defined with the characteristic equation:

$$f_{desired}(z) = z^n + \gamma_n z^{n-1} + \gamma_{n-1} z^{n-2} + \dots + \gamma_2 z + \gamma_1. \quad (14)$$

If  $f_{desired}(z) = f_{system}(z)$ , there is no need to design controller; otherwise go to Step 3.

**Step 3.** The  $W^{+}_{newK}$  is structured as follows. The dimensions of  $W^{+}_{newK}$  are  $(n+2) \times (n+2)$ .

$$W^{+}_{newK} = W^+ - \begin{bmatrix} 0_{(n) \times 1} \\ 1_{1 \times 1} \\ 0_{1 \times 1} \end{bmatrix} \times \begin{bmatrix} 0_{1 \times 1} & K_{1 \times (n)} & 0_{1 \times 1} \end{bmatrix} \quad (15)$$

$$= \begin{bmatrix} 1 & 0 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 0 & \ddots & \dots & 0 \\ \vdots & 0 & 0 & \dots & 1 & \vdots \\ 1 & -\alpha_1 - k_1 & -\alpha_2 - k_2 & \dots & -\alpha_n - k_n & 0 \\ 0 & \beta_0 & \beta_1 & \dots & \beta_{n-1} & 0 \end{bmatrix}.$$

**Step 4.** The characteristic equation of  $W^{+}_{newK}$  is calculated as:

$$\begin{aligned} \det(zI - W^{+}_{newK}) &= (z-1) \times \det(zI - A_{cl}) \times z \\ &= (z-1) \times f_{newk}(z) \times z. \end{aligned} \quad (16)$$

where

$$\begin{aligned} f_{newk}(z) &= z^n + (\alpha_n + k_n)z^{n-1} \\ &\quad + (\alpha_{n-1} + k_{n-1})z^{n-2} + \dots \\ &\quad + (\alpha_2 + k_2)z + (\alpha_1 + k_1). \end{aligned} \quad (17)$$

**Step 5.** From equation  $f_{newk}(z) = f_{desired}(z)$ , the  $K$ -vector is obtained:

$$\begin{aligned} f_{desired}(z) &= f_{newk}(z) \Rightarrow \\ \gamma_1 &= (\alpha_1 + k_1) \Rightarrow k_1 = \gamma_1 - \alpha_1 \\ &\vdots \\ \gamma_{n-1} &= (\alpha_{n-1} + k_{n-1}) \Rightarrow k_{n-1} = \gamma_{n-1} - \alpha_{n-1} \\ \gamma_n &= (\alpha_n + k_n) \Rightarrow k_n = \gamma_n - \alpha_n. \end{aligned} \quad (18)$$

**Step 6.** End.

The flowchart of this algorithm is depicted in Fig. 3.

#### 5. Simulation Results

In this section, the performance of a feedback control law designed on the basis of Petri Net model is

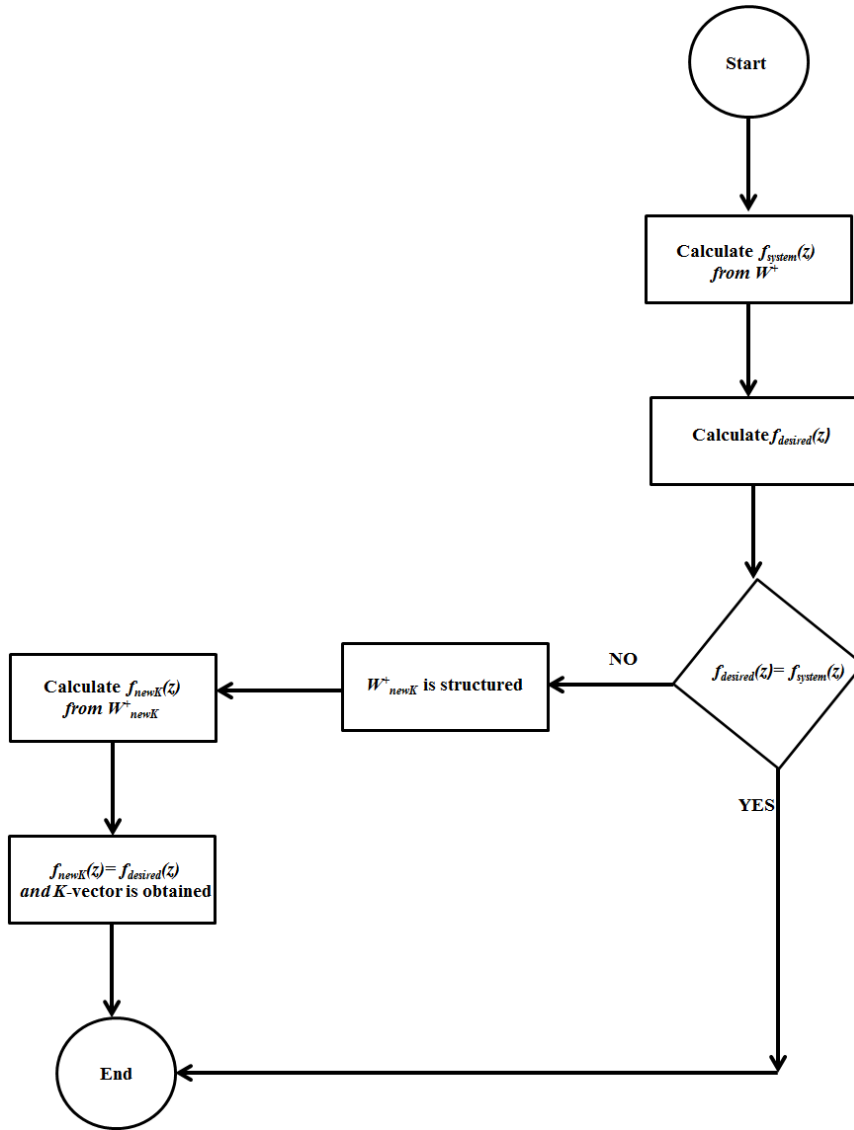


Figure 3. Flowchart of the algorithm to design a feedback control law based on Petri Nets model

presented. The simulation is carried out using MATLAB version 7.12.0.635. The inverted pendulum is an open loop unstable system. The system of interest is shown in Fig. 4, where  $F$  is the force in Newtons,  $m$  is the mass of the pendulum rod in kilograms,  $M$  is the mass of the moving cart in kilograms and  $\theta$  is the angle of the inverted pendulum measured from the vertical  $y$ -axis in radians.

A mathematical model of the pendulum system is derived below:

$$\begin{cases} \Delta \dot{x} = \Delta v \\ \Delta \dot{v} = -\frac{mg}{M} \Delta \theta + \frac{1}{M} \Delta F \\ \Delta \dot{\theta} = \Delta \omega \\ \Delta \dot{\omega} = -\frac{(m+M)g}{Ml} \Delta \theta - \frac{1}{ml} \Delta F \\ y = \Delta \theta. \end{cases} \quad (19)$$

The output variable is angle of the inverted pendulum.

The parameters of the inverted pendulum system used for simulations are given in Table 1.

Table 1. Parameters of the inverted pendulum system

| $M(\text{kg})$ | $m(\text{kg})$ | $g(\text{m/s}^2)$ | $l(\text{m})$ |
|----------------|----------------|-------------------|---------------|
| 0.792          | 0.231          | 9.8               | 0.305         |

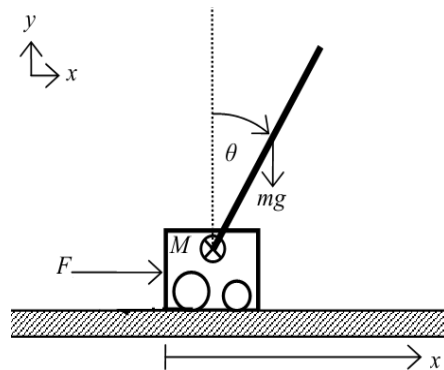


Figure 4. Schematic of the inverted pendulum system

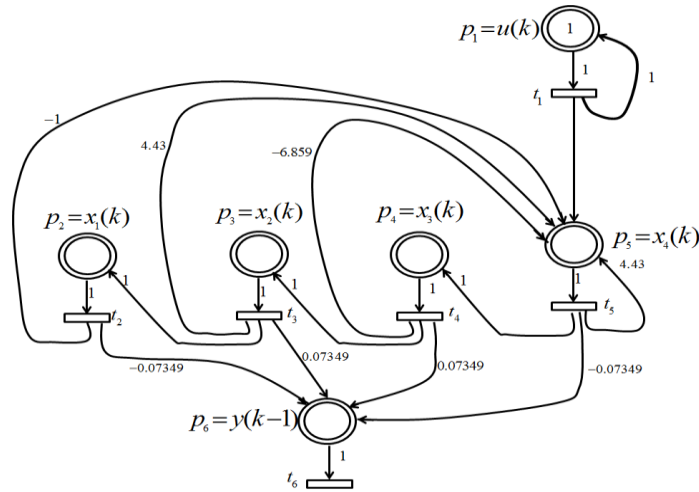


Figure 5. Petri Nets model of (20)

Let us define  $x_1$ ,  $x_2$ ,  $x_3$ , and  $x_4$  as the states of the system. Then the following state equations can be obtained:

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -2.86x_3(t) + 1.26u \\ \dot{x}_3(t) = x_4(t) \\ \dot{x}_4(t) = 41.5x_3(t) - 14.2u \\ y = x_3. \end{cases} \quad (20)$$

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

$$W = W^+ - W^- = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 & 1 & 0 \\ 1 & -1 & 4.43 & -6.859 & 3.43 & 0 \\ 0 & -0.07349 & 0.07349 & 0.07349 & -0.07349 & -1 \end{bmatrix}$$

After discretizing the system using the triangle (first-order hold) approximation with sample time  $T_s=0.1$ sec, the resulted state equation can be written:

$$\begin{cases} \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ x_4(k) \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 4.43 & -6.859 & 4.43 \end{bmatrix} \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ x_4(k-1) \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} u(k-1) \\ y(k) = [-0.07349 \quad 0.07349 \quad 0.07349 \quad -0.07349] \underline{x}(k) \end{cases}$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1 & 4.43 & -6.859 & 4.43 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\ C = [-0.07349 \quad 0.07349 \quad 0.07349 \quad -0.07349] \quad D = 0. \quad (21)$$

Finally, Petri Net for this model is demonstrated in Fig. 5.

In Fig. 5, places  $p_1$  and  $p_6$  indicate input variable and output variable, respectively. The place descriptions of the model in Fig. 5 are introduced in Table 2.

The incidence matrix  $W$  for Fig. 5 can be obtained as follows:

$$W^+ = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 4.43 & -6.859 & 4.43 & 0 \\ 0 & -0.07349 & 0.07349 & 0.07349 & -0.07349 & 0 \end{bmatrix} \quad \text{and}$$

Table 2. Place descriptions of Fig. 5

| Place | State    | Description                               |
|-------|----------|-------------------------------------------|
| $p_1$ | $u(k)$   | Input                                     |
| $p_2$ | $x_1(k)$ | Position of cart in $x$ direction         |
| $p_3$ | $x_2(k)$ | Velocity of cart in $x$ direction         |
| $p_4$ | $x_3(k)$ | Angle of the inverted pendulum            |
| $p_5$ | $x_4(k)$ | Angular velocity of the inverted pendulum |
| $p_6$ | $y(k-1)$ | Output                                    |

The eigenvalues of system can be obtained as:

$$\det(zI - W^+) = \det \begin{pmatrix} z-1 & 0 & 0 & 0 & 0 & 0 \\ 0 & z & -1 & 0 & 0 & 0 \\ 0 & 0 & z & -1 & 0 & 0 \\ 0 & 0 & 0 & z & -1 & 0 \\ -1 & 1 & -4.43 & 6.859 & z-4.43 & 0 \\ 0 & 0.07349 & -0.07349 & -0.07349 & 0.07349 & 0 \end{pmatrix} \\ = (z-1) \times \det \begin{pmatrix} z & -1 & 0 & 0 \\ 0 & z & -1 & 0 \\ 0 & 0 & z & -1 \\ 1 & -4.43 & 6.859 & z-4.43 \end{pmatrix} \times z=0 \Rightarrow$$

$$\det(zI - W^+) = (z-1) \times \det \begin{pmatrix} z & -1 & 0 & 0 \\ 0 & z & -1 & 0 \\ 0 & 0 & z & -1 \\ 1 & -4.43 & 6.859 & z-4.43 \end{pmatrix} \times z = 0 \Rightarrow$$

$$\det \begin{pmatrix} z & -1 & 0 & 0 \\ 0 & z & -1 & 0 \\ 0 & 0 & z & -1 \\ 1 & -4.43 & 6.859 & z-4.43 \end{pmatrix} = (z-1)^2(z-0.524)(z-1.91) = 0 \Rightarrow$$

$$\begin{cases} z_1=1 \\ z_2=1 \\ z_3=0.524 \\ z_4=1.9 \end{cases}$$

This system is unstable.

In this step by applying feedback control law based on the Petri Nets model a controller is designed and consequently the system begins to be stable. The following step should be taken to do so:

**Step 1.** The open loop poles of the system are calculated:

$$\begin{cases} z_1=1 \\ z_2=1 \\ z_3=0.524 \\ z_4=1.9 \end{cases}$$

**Step 2.** Define the desired pole locations with the characteristic equation.

$$f_{desired}(z) = (z - 0.525)(z + 0.7)(z + 0.8)(z - 0.2)$$

$$= z^4 + 0.775z^3 - 0.423z^2 - 0.249z + 0.059.$$

**Step 3.**  $W_{newK}^+$  is constructed as follows:

$$W_{newK}^+ = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1 & 4.43 & -6.859 & 4.43 & 0 \\ 0 & -0.07349 & 0.07349 & 0.07349 & -0.07349 & 0 \end{pmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \end{bmatrix} \times [0 \quad k_1 \quad k_2 \quad k_3 \quad k_4 \quad 0]$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1-k_1 & 4.43-k_2 & -6.859-k_3 & 4.43-k_4 & 0 \\ 0 & -0.07349 & 0.07349 & 0.07349 & -0.07349 & 0 \end{pmatrix}$$

$$W_{newK}^+ = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1-k_1 & 4.43-k_2 & -6.859-k_3 & 4.43-k_4 & 0 \\ 0 & -0.07349 & 0.07349 & 0.07349 & -0.07349 & 0 \end{pmatrix}$$

The Petri Nets model with the feedback control law is shown in Fig. 6.

**Step 4.** The characteristic equation of  $W_{newK}^+$  is obtained as:

$$\det(zI - W_{newK}^+) = \det \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 1 & -1-k_1 & 4.43-k_2 & -6.859-k_3 & 4.43-k_4 & 0 \\ 0 & -0.07349 & 0.07349 & 0.07349 & -0.07349 & 0 \end{pmatrix}$$

$$= (z-1) \times \det \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1-k_1 & 4.43-k_2 & -6.859-k_3 & 4.43-k_4 \end{pmatrix} \times z = 0 \Rightarrow$$

$$f_{newk} = \det \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -1-k_1 & 4.43-k_2 & -6.859-k_3 & 4.43-k_4 \end{pmatrix} = 0 \Rightarrow$$

$$f_{newk} = z^4 + (k_4 - 4.43)z^3 + (k_3 + 6.859)z^2 + (k_2 - 4.43)z + (k_1 + 1) = 0$$



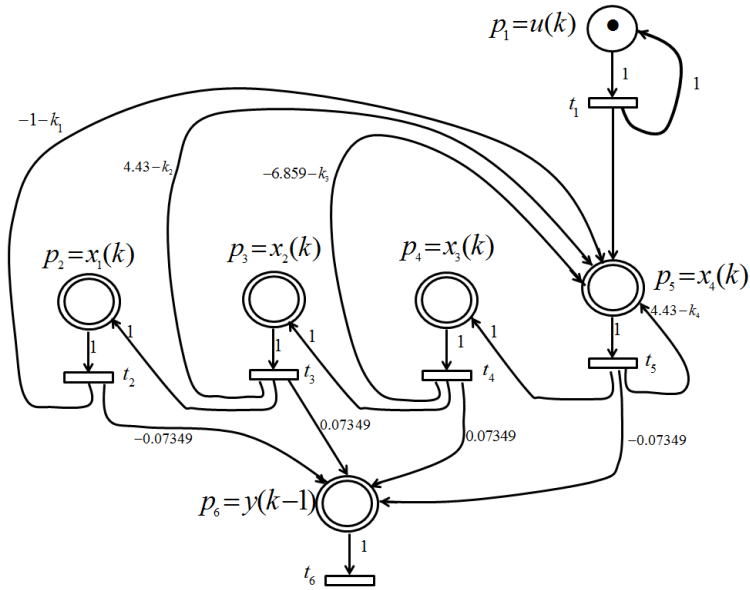


Figure 6. Petri Nets modelling with feedback control law

**Step 5.** In this Step the following equation should be satisfied:

$$f_{newk}(z) = f_{desired}(z) \Rightarrow$$

$$f_{newk}(z) = z^4 + (k_4 - 4.43)z^3 + (k_3 + 6.859)z^2 + (k_2 - 4.43)z + (k_1 + 1)$$

$$f_{desired}(z) = z^4 + 0.775z^3 - 0.423z^2 - 0.249z + 0.059$$

Consequently, the  $K$ -vector is:

$$K = [k_1, k_2, k_3, k_4] = [-0.941, 4.18, -7.28, 5.21]$$

This model has been simulated here. Fig. 7, shows the angle of the inverted pendulum step response without controller using Petri Nets model and state space model.

As it is shown above as the time increases, the deviation of pendulum angle increases too. Therefore, the system is unstable. The introduced Petri Nets method showed this fact clearly.

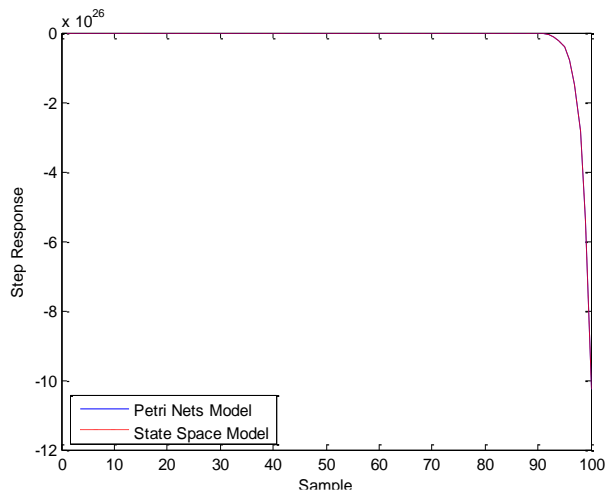


Figure 7. Step response of system with Petri Nets and state space model

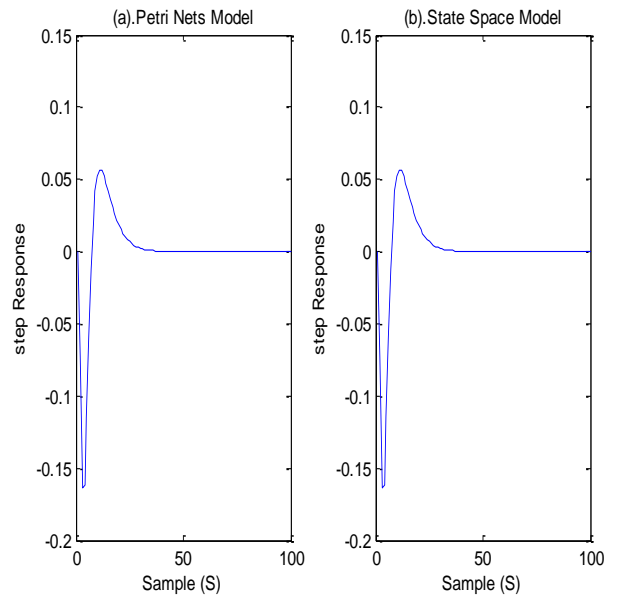
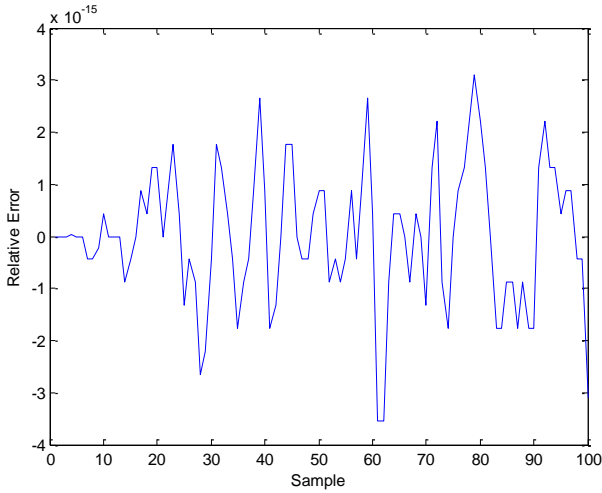


Figure 8. Output of system with state space and Petri Nets approach controller

Fig. 8, illustrates the instable system with the feedback control law based on the state space model and Petri Nets approach, and it is obvious that after applying the controller, system begins to be stable.

By comparing responses in Fig. 8, it is shown that the Petri Nets model response is the same as the state space model. The comparative accuracy of this approach can be determined by investigating relative error. The output relative error between state space and Petri Nets approach controller is illustrated in Fig. 9.



**Figure 9.** Output error between state space and Petri Nets approach controller

The simulation results show that the dynamic behaviour including transient response, steady state response and steady state error is the same in both methods. Hence, the validity of the presented approach can be verified. The simulation of the

**Table 3.** Simulation time results

| System                                               | Transfer function                        | Petri Nets                       | Conventional                    |
|------------------------------------------------------|------------------------------------------|----------------------------------|---------------------------------|
| Boring Machine                                       | $G(s) = \frac{1}{s(s+12)}$               | $9.2 \times 10^{-4}(\text{sec})$ | $15 \times 10^{-4}(\text{sec})$ |
| Laser eye surgery System                             | $G(s) = \frac{2}{s(s+1)(s+4)}$           | $16 \times 10^{-4}(\text{sec})$  | $26 \times 10^{-4}(\text{sec})$ |
| Head dynamics                                        | $G(s) = \frac{1}{s(s+2)(s+3)}$           | $15 \times 10^{-4}(\text{sec})$  | $25 \times 10^{-4}(\text{sec})$ |
| Aircraft dynamics                                    | $G(s) = \frac{(s+20)}{s(s+10)^2}$        | $15 \times 10^{-4}(\text{sec})$  | $25 \times 10^{-4}(\text{sec})$ |
| Flexible arm                                         | $G(s) = \frac{1}{s(s^2+2s+4)}$           | $15 \times 10^{-4}(\text{sec})$  | $27 \times 10^{-4}(\text{sec})$ |
| The open-loop transfer function of direct driver arm | $G(s) = \frac{s+2}{s(s+5)(s^2+2s+5)}$    | $19 \times 10^{-4}(\text{sec})$  | $33 \times 10^{-4}(\text{sec})$ |
| Elevator Driver                                      | $G(s) = \frac{s+10}{s(s+1)(s+20)(s+50)}$ | $18 \times 10^{-4}(\text{sec})$  | $33 \times 10^{-4}(\text{sec})$ |

**6. Conclusion**

A continuous system described by differential equations cannot be modelled by ordinary Continuous Petri Nets. To simplify modelling, controller design and analysis, this paper introduces Continuous -Time Delay- Petri Nets (CTDPN). CTDPN allows to model all continuous systems using simple rules. The principal equation of Petri Nets plays the same role as discrete state space equation. The properties of the discrete state space are reserved in incidence matrix. A simple algorithm for controller design by CTDPN based on state feedback control law is presented and

forementioned system by our novel algorithm and the conventional one via the same hardware configuration relays a significant advantage of the new method which is the time efficiency. (The simulation time for the presented method was 0.0023 seconds compared to the 0.0063 seconds of the old one which means considerable amount of %64.27 decreases in run time). For further investigation, different types of common systems are considered and the time it takes to calculate the gains and simulate the model is compared [25]. Table 3 shows the simulation time result for the presented method compared to the conventional approach.

It can be observed that the advantage of the introduced method lies in reduced computation effort leading to time efficiency. Modelling with the Petri Nets provides powerful tools for the user. Petri Nets modelling gives graphical tools that provide an effective communication medium between the user and the system which enables us to model the dynamics of the systems visually.

simulation results show this approach is effective and this method has much less complexities. By using CTDPN, a visual and systematic method for modelling the dynamics of a continuous system is presented which has considerably less run time compared to conventional approach. Besides, this method can be utilized to model hybrid systems with variable parameters or multiple subsystems which suggested for the further works on the basis of this approach.

## References

- [1] **Z. Tao, L. Xie, D. Liang.** Controller Design of DES Petri Nets with Mixed Constraint. *Chinese Journal of Aeronautics*, 2005, Vol. 18, No.3, 283- 288.
- [2] **M. Kloetzer, C. Mahulea, C. Belta, M. Silva.** An Automated Framework Formal Verification of Timed Continuous Petri Nets. *IEEE Transactions on Industrial Informatics*, 2010, Vol. 6, pp. 460-471.
- [3] **K. Sacha.** Fault Analysis Using Petri Nets. *IEEE Real-time Embedded Systems Workshop*, 2001, pp. 130-133.
- [4] **M. Durmus, S. Takai.** Modeling Moving-Block Railway Systems: A Generalized Batches Petri Net Approach. *SICE Journal of Control, Measurement and System Integration*, 2013, Vol. 6, No. 6, 403–410.
- [5] **E. Fesina, S. Savdur.** Modeling of Sewage Bioremediation as a Modified Petri Net. *World Applied Sciences Journal*, 2014, Vol. 31, No. 6, 1191-1197.
- [6] **T. Jie, M. Ameen.** A Survey of Petri Net Tools. *ARPJ Journal of Engineering and Applied Sciences*, 2014, Vol. 9, No. 8, 1209-1214.
- [7] **A. Dideban, M. Zareiee, H. Alla.** Controller Synthesis with Very Simplified Linear Constraints in PN Model. *2<sup>nd</sup> IFAC Workshop on Dependable Control of Discrete Systems, Bari, Italy*, 2009, pp. 265-270.
- [8] **H. Alla, R. David.** A Modeling and Analysis Tool for Discrete Events Systems: Continuous Petri Net. *Performance Evaluation*, 1998, Vol. 33, No. 3, 175-199.
- [9] **H. Apaydin-Ozkan, J. Julvez, C. Mahulea, M. Silva.** Approaching Minimum Time Control of Timed Continuous Petri Nets. *Nonlinear Analysis: Hybrid Systems*, 2011, Vol. 5, No. 2, 136-148.
- [10] **C. Vázquez, A. Ramírez-Treviño, M. Silva.** Controllability of Timed Continuous Petri Nets with Uncontrollable Transitions. *International Journal of Control*, 2014, Vol. 87, No. 3, 537-552.
- [11] **Y. Wang, C. Chang.** A Hierarchical Approach to Construct Petri Nets for Modeling the Fault Propagation Mechanisms in Sequential Operation. *Computer & Chemical Engineering*, 2003, Vol. 27, No. 2, 259-280.
- [12] **L. Recalde, E. Teruel, M. Silva.** Autonomous Continuous P/T Systems. *Lecture Notes in Computer Science: Application and Theory of Petri Nets*, 20th International Conference, USA, 1999, pp. 107–126.
- [13] **R. Champagnat, R. Valette, J. Hochon, H. Pingaud.** Modelling, Simulation and Analysis of Batch Production Systems. *Discrete Event Dynamic Systems: Theory and Applications*, 2001, Vol. 11, No. 1, 119-136.
- [14] **C. Vázquez, L. Recalde, M. Silva.** Stochastic Continuous-State Approximation of Markovian Petri Net Systems. *47<sup>th</sup> IEEE Conference on Decision and Control, Cancun, Mexico*, 2008, pp. 901-906.
- [15] **D. Rowell.** Analysis and Design of Feedback Control Systems: State-Space Representation of LTI Systems, 2002.
- [16] **A. Dideban, A. Ahangarani Farahani, M. Razavi.** Modeling Continuous Systems Using Modified Petri Nets Model. *Modeling and Simulation in Electrical and Electronics Engineering (MSEEE), Semnan, Iran*, 2015, Vol. 1, No. 2, pp. 75-79.
- [17] **A. Hartmann, F. Schreiber.** Integrative Analysis of Metabolic Models—from Structure to Dynamics. *Front Bioeng Biotechnol*, 2015, Vol. 2, 1-10.
- [18] **Y. Q. Song, B. Li, K. Chen, T. Yang, H. Li.** Reliability Research of the Traffic Signal System Based on Extended Petri Nets. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 2015, Vol. 8, 321-332.
- [19] **A. Dideban, M. Kiani, H. Alla.** Implementation of Petri Nets Based Controller Using SFC. *Journal of Control Engineering and Applied Informatics*, 2011, Vol. 13, No. 4, 82-92.
- [20] **R. David, H. Alla.** Discrete, Continuous, and Hybrid Petri Nets. *Springer-Verlag Berlin Heidelberg*, 2005.

- [21] **M. Alcaraz-Mejia, R. Campos-Rodriguez, E. Lopez-Mellado, A. Ramirez-Trevino.** Partial Reconfiguration of Control Systems Using Petri Nets Structural Redundancy. *Information Technology and Control*, 2015, Vol. 44, No. 3, 287-301.
- [22] **E. Fraca, S. Haddad.** Complexity Analysis of Continuous Petri Nets. *Fundamenta Informaticae*, 2015, Vol. 137, 1-28.
- [23] **M. Kloetzer, C. Mahulea, C. Belta, L. Recalde, M. Silva.** Formal Analysis of Timed Continuous Petri Nets. In: 47<sup>th</sup> IEEE Conference on Decision and Control, Cancun, Mexico, 2008, pp. 245-250.
- [24] **M. Sam Fadali.** Digital Control Engineering: Analysis and Design. *Elsevier*, USA, 2009.
- [25] **R. Dorf, R Bishop.** Modern Control System. *Prentice Hall*, 9<sup>th</sup> Edition, 2011.

Received November 2015.