IJE TRANSACTIONS A: Basics Vol. 36 No. 10, (October 2023) 1868-1879

International Journal of Engineering

Journal Homepage: www.ije.ir

Controller Design of Hybrid-Time Delay-Petri Nets Based on Lyapunov Theory by Adding Control Places

A. Ahangarani Farahani, A. Dideban*

Department of Electrical and Computer Engineering Semnan University, Semnan, Iran

PAPER INFO

Paper history:
Received 13 May 2023
Received in revised form 02 July 2023
Accepted 09 July 2023

Keywords:
Control Place
Hybrid-Time Delay-Petri Nets
Stability
Lyapunov Theory

ABSTRACT

The aim of this paper is to propose a new method for controller design using control places in special hybrid Petri Nets called Hybrid-Time Delay-Petri Nets (HTDPN). Most control approaches use the control place of the supervisory control for discrete Petri Nets. However, the new approach uses the place to control the linear dynamical systems which are modeled by the HTDPN tool. This controller consists of control places, transitions, arcs connected to the control place, and weights of the arcs, which are added to the HTDPN model of the system. In this paper, there are three main steps for the controller design. In the first step, the plant is modeled using the HTDPN tool, and in the second step, a controller is designed using the novel method presented. Finally, the weights of arcs connected to the control place are computed using the Lyapunov function theory, which guarantees closed-loop stability. The main advantage of this method is the possibility of using continuous and discrete places simultaneously in nonlinear systems. Unlike most previous approaches, in the proposed method, an expert designer can create a favorite controller in the graphical environment, and then apply changes to the mathematical environment of the HTDPN model. The performance of the proposed controller is evaluated by a comparative study. The comparison criteria in this article are: error criteria (IEA), energy consumption, rise time, settling time and simulation run time. The simulation results showed that the proposed method was 45% and 600% better conditions than the Model Predictive Conrol (MPC) and optimal control methods, respectively.

doi: *10.5829/ije.2023.36.10a.12*

1. INTRODUCTION

In new systems such as traffic systems, biological systems, etc., they are described by differential equations, therefore, they are motivated to develop new methods for analysis, modeling, evaluation and control of systems [1-3]. One of the most successful modeling approaches is Petri Nets [4].

The control of dynamic systems which are modeled by the HTDPN tool has been a matter of great interest. In the last decade, several researchers have been working on the control based on discrete Petri Nets [5, 6]. Supervisory control is one of the essential methods for controller design in Discrete Event Systems (DES) using Petri Nets tool [7]. In supervisory control, the behavior of the system is controlled by adding places and transitions [8]. Demongodin and Koussoulas [9, 10] modeled a controller which was designed based on supervisory control for the industrial system by differential Petri Nets. In the articles mentioned, differential equation modeling was carried out by Petri Nets, which requires the use of new definitions such as discrete implicit differential transition. Saleh et al. [11], a hybrid adaptive Petri Nets is introduced, in which transition commutes between discrete and continuous behavior depending on a threshold. Ruan and Li [12], for the control of traffic, first, a macroscopic model based on continuous Petri Nets is proposed, and then predictive control laws that improve the behavior of traffic systems are designed. Taleb et al. [13] designed a model predictive control for timed continuous Petri Nets systems. In the methods mentioned, controllers are designed based on the system variables that are generally flow. Continuous systems theory is often described by continuous-time differential or discrete-time differential equations. Therefore, this tool could not be practical to apply to all dynamic systems [14, 15].

*Corresponding Author Email: *adideban@semnan.ac.ir* (A. Dideban)

Please cite this article as: A. Ahangarani Farahani, A. Dideban, Controller Design of Hybrid-Time Delay-Petri Nets Based on Lyapunov Theory in by Adding Control Places, ***International Journal of Engineering, Transactions A: Basics***, Vol. 36, No. 10, (2023), 1868-1879

Previous attempts have been made to model and control continuous linear dynamic systems, which are modeled by differential equations using Petri Nets. Dideban. and Ahangarani Farahani [16] also designed an output feedback controller based on modified Petri Nets. Modified Petri Nets was introduced by Dideban et al. [17], where a discrete transfer function is modeled by Petri Nets. A PID controller based Petri Nets was also proposed by Dideban et al. [18] and Ahangarani Farahani [19]. In these works, some new concepts were added to the conventional Petri Nets, making them rather difficult to be analyzed. The state feedback controller by the Continuous-Time Delay-Petri Nets tool without adding new elements to Petri Nets has been presented by Ahangarani Farahani and Dideban [20]. In the following, the Hybrid Time Delay Petri Net is introduced as a tool for modeling systems with the current sample time signals, including various subsystems, and multi-mode systems [21]. Then a PID controller is designed in which the gains are tuned by the intelligent method. In all the introduced tools, none of the methods provide a controller design based on the graphical and mathematical capabilities of the continuous Petri Nets tool for dynamical systems.

The control place approach is an important method in the control of discrete event systems modeled by Petri Nets. Ma et al. [22] developed an algorithm an optimal control sequence in Petri Nets for designing, which drives a plant net from a source marking to a set of target markings without passing any pregiven forbidden markings. A shunt active power filter (SAPF) based on a three-phase serial flying capacitor multilevel inverter (FCMI) controlled using a Petri net tool is presented by Othman et al. [23]. This controller design is based on the structure of the investigated system and according to the capabilities of the Petri net for the control of discrete event systems. Bashir et al. [24] attempted to prevent deadlock in a manufacturing system, the design of supervisory control which was done based on the Petri Net tool and using the combination of place and transition control. Here, the combination of place and transition has been given flexibility to the designer. Chenand Hu [25] used the developed place-invariant control in automated manufacturing systems based on the Petri Nets tool. In this article, the extended place-invariant control principle is initially proposed. Second, three types of place-invariant, from the special to general, are developed. Finally, the use of this principle is presented to simplify the design of supervisory control. In these articles, all methods and controllers based on Petri Nets are designed for discrete event systems and to prevent the system from entering unsafe conditions. Therefore, these methods cannot be used for dynamic systems that are described by differential equations and discrete events, such as HTDPN.

The principal contribution of this article is to use the idea of control place to design controllers in linear dynamic systems which are modeled by the HTDPN tool. In the proposed approach, the use of control places technique, which is used for the supervisory control of conventional Petri Nets, is extended to HTDPN. Unlike classic control methods for dynamic systems, an important feature of this novel controller is the use of Petri Nets graphics capabilities. In this method, the user can design the controller by adding control places to the HTDPN model in the graphical environment. The designed controller is easily applied to the mathematical part by the incidence matrix. Therefore, controller design is done in the graphical environment instead of the mathematical environment. In other words, here, the controller design methods for discrete event systems are used to design the controller of the dynamic system modeled with a HTDPN. In this paper, the relationship between the control places and other components are determined using the GA (Genetic Algorithm) method. Another innovation in this article is to present the use of Lyapunov's theory to prove stability in Petri Nets based on the incidence matrix. Here, by applying the Lyapunov stability theory on the incidence matrix, the weights of the arcs connected to control place or the control coefficients are obtained. The ability to use continuous and discrete places simultaneously enables us to design a suitable controller in some nonlinear systems. Another innovation of this paper is the use of Lyapunov stability concepts for the mathematical part of the Petri Nets and its use for controller design. The simulation results show that the implementation of this control method using Petri Nets capabilities has better accuracy and less energy consumption than optimal control and MPC Method.

The paper is structured as follows. In section 2, the main concepts, definitions, and mathematics of the Continuous and Hybrid-Time Delay-Petri Nets are proposed. Controller design and stability proof based on Lyapunov theory in the HTDPN is presented in section 3. The dynamic model of the capsubot robot and the implementation of the control method on the system are presented in section 4. Section 5 is dedicated to simulation results, and finally, the conclusion is given in section 6.

2. CONTINUOUS AND HYBRID-TIME DELAY-PETRI NETS

In this section, the CTDPN and HTDPN tool, definitions, and properties are provided. In addition, the mathematical equation has been developed for the HTDPN tool.

A CTDPN is a mathematical and graphical modeling tool for dynamical systems, which are described by difference equations. The CTDPN tool is defined as follows [20]:

Definition 1: A Continuous -Time Delay-Petri Nets (CTDPN) is a 6-tuple $\mathrm{PN_C} = \{\mathrm{P}, \mathrm{T}, \mathrm{W^-(Pre)}, \mathrm{W^+(Post)}, \mathrm{M_0},\ \mathrm{T_s}\}$ such that:

$P = \{p_1, p_2, \cdots, p_n\}$ and $T = \{t_1, t_2, \cdots, t_m\}$ are finite sets of continuous places and transitions, respectively.

Pre and $Post$ are the incidence functions that specify the multiplicity of arcs between places and transitions. $M_0 \epsilon \mathbb{R}$ is the initial marking vector, and T_s is the time interval between each run cycle. To model the continuous dynamic system using the CTDPN, the following assumptions and rules should be considered:

Continuous transitions are corresponding to time delays.

In the CTDPN, $M \in \mathbb{R}^n$.

The enabling degree of a transition t_j at a marking $M(p_i)$ is defined as:

$$q(t_j, m) = \min_{i: p_i \in {}^{\circ}t_j} \left(\frac{M(p_i)}{Pre(p_i, t_j)}\right) \tag{1}$$

A continuous transition $t_j \in T$ is enabled, i.e., it can fire, if

$$|M(p_i)| > 0\ \forall p_i \in^{\circ} t_j$$

where ${}^{0}t_j = \{p_i \in P \,|\, Pre(p_i, t_j) > 0\}$ is the input place.

In the CTDPN tool the weights of the arcs can be negative or non-negative real numbers.

Property 1: The continuous transitions speed in the CTDPN used in the linear system are determined by the input place tokens ($M(p_i)$) divided by the sampling time (T_s).

$$v_j = \frac{M(p_i)}{T_s} \tag{2}$$

Proof: Proof is given in appendix.

Property 2: The fundamental state equation of the CTDPN can be written as follows:

$$m(n) = m(n-1) + Wm(n-1) \tag{3}$$

Proof: Proof is given in appendix.

Property 3: The eigenvalues of the dynamical system are equal to the eigenvalues of the W^+matrix by removing the value of zero.

Proof: Proof is given by Ahangarani Farahani and Dideban [20].

A HTDPN is a modeling tool to model dynamic systems such as systems with current sample time signals, system including various subsystems and multi-mode systems.

Definition 2: A HTDPN is defined as $\mathrm{PN_H} = \{\mathrm{P}, \mathrm{T}, \mathrm{W^-(Pre)}, \mathrm{W^+(Post)}, \mathrm{M_0},\ \mathrm{h}, \mathrm{T_s}\}$, where

$P = \{p_1, p_2, \cdots, p_n\}$ and $T = \{t_1, t_2, \cdots, t_m\}$ are a finite, not empty, set of continuous and discrete places and transitions, respectively. Pre and $Post$ are the backward and forward incidence mappings. M_0 and T_s were introduced in definition 1. $h: P \cup T \to \{C, D\}$ is a hybrid function which indicates whether each node is a discrete node or a continuous node. In HTDPN, discrete transitions are initially executed [20].

To illustrate this tool, consider the net in Figure 1. All concepts that can be modeled to HTDPN are shown in this figure. Places p_1 and p_2 and transitions $t_1$and t_2 are continuous places and transitions, respectively. Places p_3 and p_4 and transitions t_3 and t_4 are discrete places and transitions, respectively.

Here, transition t_1 is enabled only if there is at least one token in p_3 and $M(p_1) > 0$. Therefore the speed of transitions t_1 and $t_2$can be written as follows:

$$v_1 = \frac{M(p_1)}{T_s}, \quad v_2 = \frac{M(p_2)}{T_s} \tag{4}$$

However, the following assumptions and rules should be considered:

Continuous transitions are corresponding to time delays.

In the HTDPN, continuous places contain real values, while discrete places contain non-negative integer values.

A continuous transition $t_j \in T$ is enabled, if each of the continuous and discrete input places to transition t_j have the following condition at the same time:

$$\begin{array}{lll} |M(p_l)| > \mathrm{Pre}(p_l, t_j)\ \& & \text{If } p_l \text{ D-Place} & \multirow{2}{*}{$\forall p_{i\&l} \in^{\circ} t_j$} \\ |M(p_i)| > 0 & \text{If } p_i \text{ C-Place} & \end{array}$$

The firing speed of the continuous $t_j \in T$ is:

$$v_j = \frac{M(p_\mathrm{i})}{T_s} \qquad \text{If } p_i \text{ C-Place} \tag{5}$$

A discrete transition t_l is enabled at discrete $M(p_i)$, if $M(p_i) \geq Pre(p_i, t_l)$

In the first step, the discrete transitions must be evaluated and fired (if enabled) before continuous transitions. Therefore, the fundamental equation for the discrete part of the HTDPN is:

$$\begin{bmatrix} \acute{m}_c(n) \\ \acute{m}_d(n) \end{bmatrix} = \begin{bmatrix} m_c(n-1) \\ m_d(n-1) \end{bmatrix} + W \begin{bmatrix} 0 \\ X(n) \end{bmatrix} \tag{6}$$

where $X(n)$ is the firing vector of discrete transitions.

The fundamental equation for the continuous part can be written as follows:

$$\begin{bmatrix} m_c(n) \\ m_d(n) \end{bmatrix} = \begin{bmatrix} m_c(n-1) \\ m_d(n-1) \end{bmatrix} + W \begin{bmatrix} VT_s \\ 0 \end{bmatrix} \tag{7}$$

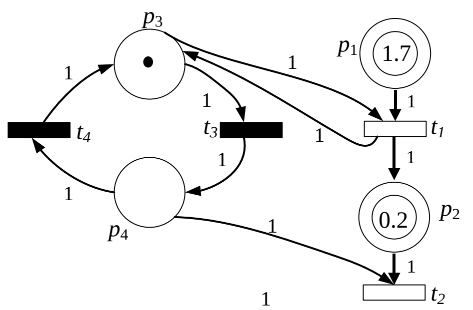

Figure 1. A hybrid Petri Nets model

where the incidence matrix is written as:

$$W=\begin{bmatrix}-1 & 0 & 0 & 0\\ 1 & -1 & 0 & 0\\ 0 & 0 & -1 & 1\\ 0 & 0 & 1 & -1\end{bmatrix}=\begin{bmatrix}W^{C} & W^{DC}\\ W^{CD} & W^{D}\end{bmatrix} \quad (15)$$

where W^C and W^D correspond to arcs among continuous and discrete nodes, respectively. W^{DC} corresponds to arcs among discrete transitions and continuous places. Arcs among continuous transitions and discrete places (W^{CD}) are zero and their effects are given in the V vector.

In this paper, the discrete transitions between continuous places were not used.

$$V(n)=\begin{bmatrix}\frac{M(p_1(n-1))}{T_s}\\ \frac{M(p_2(n-1))}{T_s}\\ 0\\ 0\end{bmatrix} \quad (8)$$

For the $X(n)$ vector, if $M(p_i)\geq Pre(p_i,t_l)$, the transition can be fired. In this example, t_3 can be fired and then $X_{t_3}(n)=1$ and for the transition t_4, $X_{t_4}(n)=0$. The simple hybrid system mentioned above has two modes. Therefore, this system can be converted into two continuous systems, in the first mode, the place p_3 has a token and in the second, p_4 has a token. To examine the stability property, it is necessary to construct the matrix J^{c+} . This matrix is defined in definition 3.

Definition 3: In each mode, the augmented continuous incidence matrix J^{c+} is extracted from the system incidence matrix W^+ as follows:

$$J^{c+}=[W^{C+} \;\vdots\; W^{DC+}.m_d(k)] \quad (9)$$

Where m_d can be obtained as:

$$m_d(k)=m_d(k-1)+W^D.X \quad (10)$$

And X is the firing vector of discrete transitions.

3. CONTROLLER DESIGN BASED ON CONTROL PLACE IN THE HTDPN TOOL

A detailed description of the controller design algorithm based on the HTDPN tool is presented in five steps as follows:

Algorithm 1:

Step 1. Calculate the open-loop poles of the system in each discrete mode using the augmented continuous incidence matrix J^{c+} based on property 1 [26].

$$det(zI-J^{c+})=0 \quad (11)$$

If the system in each mode is stable, there is no need to design a controller; otherwise, go to step 2.

Step 2. Add a control place to the HTDPN model of the system.

Step 3. Construct the J^{c+}_{new} in each mode. The dimensions of J^{c+}_{new} are $(n+2)\times(n+2)$.

Step 4. Obtain the fundamental equation of the system in each mode.

$$m_c(n)=m_c(n-1)+J^c vT_s \quad (12)$$

where

$$v=\frac{m(n-1)}{T_s} \quad (13)$$

So equation Equation (12) can be rearranged to Equation (14):

$$m_c(n)=J^{c+}m_c(n-1) \quad (14)$$

Step 5. Calculate k_i as the system is stabilized. Here, the Lyapunov method can be used.

A Lyapunov function can be exploited for the synthesis of nonlinear control systems. First, a Lyapunov function V must be found for the closed-loop system and then a control law is designed, which makes the ΔV negative for the required region of attraction [27]. For this purpose, the following Lyapunov's function candidate is defined:

$$V(n)=m_c^T P m_c>0 \quad (15)$$

and

$$\Delta V=V(n)-V(n-1)<0 \quad (16)$$

where

$$\Delta V=m_c^T(n-1)J^{c+T}PJ^{c+}m_c(n-1)-m_c^T(n-1)Pm_c(n-1)<0 \Rightarrow \Delta V=m_c^T(n-1)(J^{c+T}PJ^{c+}-P)m_c(n-1)<0$$
$$\Rightarrow \Delta V=m_c^T(n-1)(-Q)m_c(n-1)$$

where

$$Q=-(J^{c+T}PJ^{c+}-P) \quad (17)$$

In Lyapunov's method for stability, Q must be a positive definite constant matrix.

The flowchart of this algorithm is shown in Figure 2.

4. CONTROLLER DESIGN FOR CAPSUBOT ROBOT

4. 1. Capsubot Dynamic Model The capsubot is selected as the system, that is to be controlled by adding a control place. The simplified schematic model of the legless piezo capsule robot is depicted in Figure 3.

A mathematical model of the capsubot system is derived below [28].

$$\begin{cases}\dot{x}_1(t)=x_2(t)\\ \dot{x}_2(t)=-\frac{\mu_{1k}}{M}(M+m)g.sign(x_2)+\frac{\mu_{2k}}{M}mg.sign(x_4-x_2)+\frac{1}{M}u\\ \dot{x}_3(t)=x_4(t)\\ \dot{x}_4(t)=-\mu_{2k}g.sign(x_4-x_2)-\frac{1}{m}u\end{cases}\Rightarrow \dot{X}(t)=A_cX(t)+B_cu(t)+f_c(t) \quad (18)$$

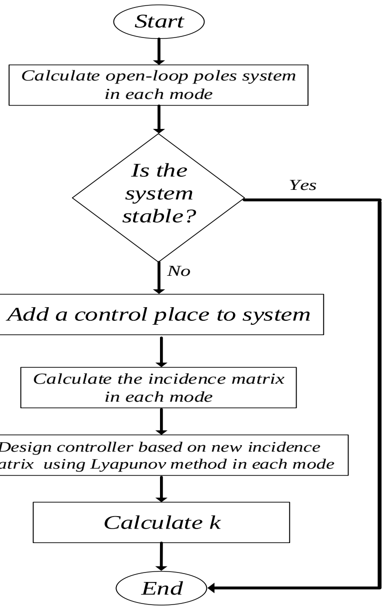

Figure 2. Flowchart algorithm to design a control place based on the HTDPN model

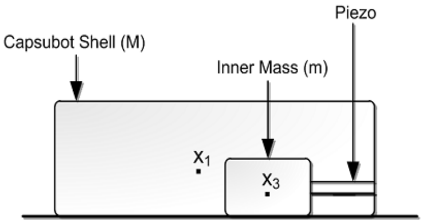

Figure 3. The schematic of the legless piezo capsule robot

where
$X(t) = [x_1(t) \quad x_2(t) \quad x_3(t) \quad x_4(t)]^T$
and

$$A_c = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{m} \end{bmatrix}$$

$$f_c(t) = \begin{bmatrix} 0 \\ -\frac{\mu_{1k}}{M}(M+m)g.sign(x_2) + \frac{\mu_{2k}}{M}mg.sign(x_4 - x_2) \\ 0 \\ -\mu_{2k}g.sign(x_4 - x_2) \end{bmatrix}$$

The parameters of the capsubot robot used are given in Table 1.

A mathematical model of the capsubot microrobot is described as follows:

Finally, the HTDPN with step input for this model is demonstrated in Figure 4.

Here 0^+ is the smallest measurable value in a digital system. The incidence matrix is depicted in Figure 5.

The HTDPN tool models multi-mode systems very well and provides a clear graphical model for analyzing and designing the controller. This system operates in four modes.

The augmented continuous incidence matrix for each mode is given in Table 2.

4. 2. Control Design Based on Place The dynamic model of the aforementioned capsule robot is a combined nonlinear model that consists of a discrete event part and a linear dynamic part. Therefore, the proposed technique is very difficult to control. In the following, the control method is presented in 5-steps.

Step 1. The open-loop poles of the system in each mode are calculated as:

TABLE 1. Parameters of the capsubot

$M_1(kg)$	$m_2(kg)$	$\mu_{1k}(N/M/s)$	$\mu_{2k}(N/M/s)$	$g\ (m/s^2)$
0.9	0.6	0.083	0.008	9.81

$$\begin{cases} \dot{x}_1(t) = x_2(t) \\ \dot{x}_2(t) = -1.3571.sign(x_2) + 0.0523.sign(x_4 - x_2) + 1.111u \\ \dot{x}_3(t) = x_4(t) \\ \dot{x}_4(t) = -0.0785.sign(x_4 - x_2) - 1.6667u \end{cases} \quad (19)$$

After converting the dynamic system from continuous-time system to discrete-time system with the sample time $T_s = 0.01(s)$, the resulting state space is:

$$\begin{aligned} x_1(k) &= x_1(k-1) + 0.01x_2(k-1) - 1.3571 \times 10^{-4}.sign(x_2(k-1)) \\ x_2(k) &= x_2(k-1) - 1.3571 \times 10^{-2}.sign(x_2(k-1)) + 0.0111u(k-1) \\ x_3(k) &= x_3(k-1) + 0.01x_4(k-1) \\ x_4(k) &= x_4(k-1) - 0.0785 \times 10^{-2}.sign(x_4(k-1) - x_2(k-1)) \\ &\quad -0.0167u(k-1) \end{aligned} \quad (20)$$

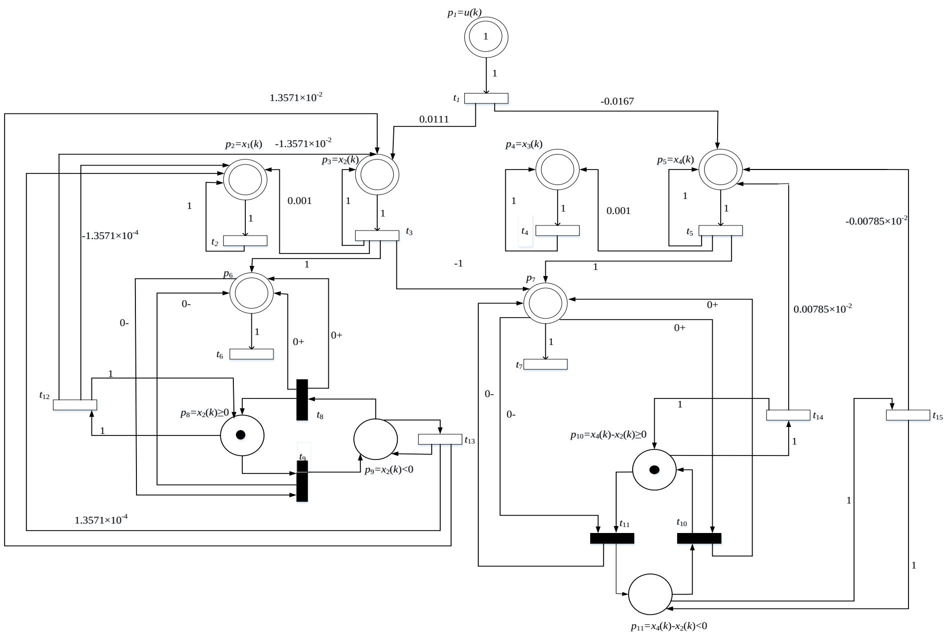

Figure 4. The HTDPN model of the capsubot

$$
W^{+}=\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.3571\times10^{-4} & 1.3571\times10^{-4} & 0 & 0\\
0.0111 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1.3571\times10^{-2} & 1.3571\times10^{-2} & 0 & 0\\
0 & 0 & 0 & 1 & 0.001 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0.0167 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.00785\times10^{-2} & -0.00785\times10^{-2}\\
0 & 0 & 1 & 0 & 0 & 0 & 0 & 0^{+} & 0^{-} & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0^{-} & 0^{+} & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1
\end{bmatrix}
$$

Figure 5. Matrix incidence presentations

TABLE 2. The incidence matrix for capsule robot in four modes

Mode	Condition	The augmented continuous incidence matrix
1	$x_2(k-1)>0$ & $x_4(k-1)-x_2(k-1)>0$	$J^{c+}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0.01 & 0 & 0 & -1.3571\times10^{-4}\\ 0.0111 & 0 & 1 & 0 & 0 & -1.3571\times10^{-2}\\ 0 & 0 & 0 & 1 & 0.01 & 0\\ -0.0167 & 0 & 0 & 0 & 1 & -0.0785\times10^{-2}\\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
2	$x_2(k-1)>0$ & $x_4(k-1)-x_2(k-1)<0$	$J^{c+}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0.01 & 0 & 0 & -1.3571\times10^{-4}\\ 0.0111 & 0 & 1 & 0 & 0 & -1.3571\times10^{-2}\\ 0 & 0 & 0 & 1 & 0.01 & 0\\ -0.0167 & 0 & 0 & 0 & 1 & 0.0785\times10^{-2}\\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
3	$x_2(k-1)<0$ & $x_4(k-1)-x_2(k-1)>0$	$J^{c+}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0.01 & 0 & 0 & 1.3571\times10^{-4}\\ 0.0111 & 0 & 1 & 0 & 0 & 1.3571\times10^{-2}\\ 0 & 0 & 0 & 1 & 0.01 & 0\\ -0.0167 & 0 & 0 & 0 & 1 & -0.0785\times10^{-2}\\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$
4	$x_2(k-1)<0$ & $x_4(k-1)-x_2(k-1)<0$	$J^{c+}=\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0.01 & 0 & 0 & 1.3571\times10^{-4}\\ 0.0111 & 0 & 1 & 0 & 0 & 1.3571\times10^{-2}\\ 0 & 0 & 0 & 1 & 0.01 & 0\\ -0.0167 & 0 & 0 & 0 & 1 & 0.0785\times10^{-2}\\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$

$$\begin{cases} z_1 = 1 \\ z_2 = 1 \\ z_3 = 1 \\ z_4 = 1 \\ z_5 = -0.0001 + 0.0105i \\ z_6 = -0.0001 - 0.0105i \end{cases}$$

The system is unstable, so it requires a controller.

Step 2. Add a control place to the HTDPN model of the system. The HTDPN model of the system is shown in Figure 4.

The HTDPN model of Equation (19) with the controller is depicted in Figure 6.

In Figure 6, transitions t_{16}, t_{17}, t_{18} and t_{19} act as a switch, and by placing the robot in any mode, these transitions determine which control coefficient to use.

Step 3. The augmented continuous incidence matrix J_{new}^{c+} is structured in Table 3.

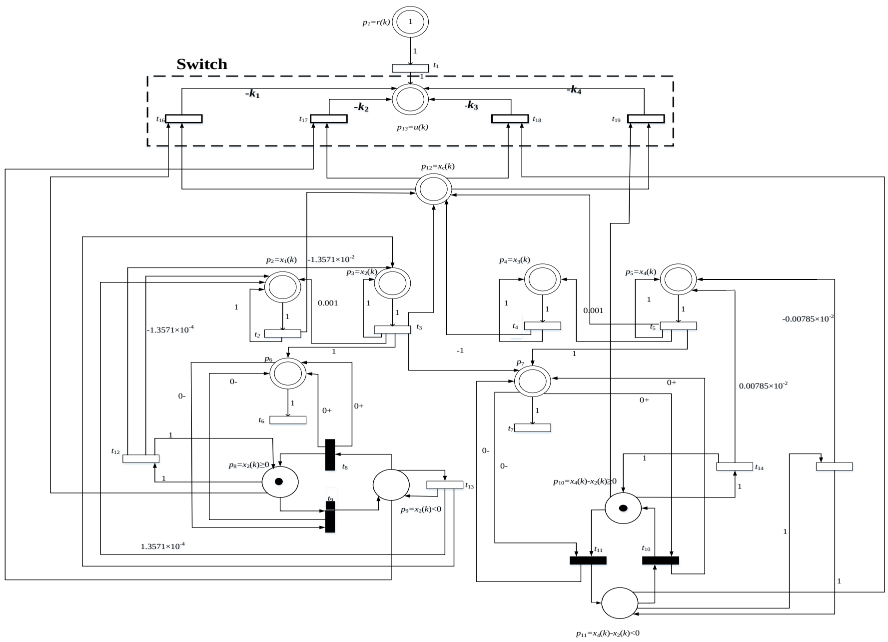

Figure 6. The HTDPN model of the system with Equation (27)

TABLE 3. The incidence matrix for capsule robot with control place in four modes

Mode	Condition	The augmented continuous incidence matrix
1	$x_2(k-1) > 0$ & $x_4(k-1) - x_2(k-1) > 0$	$J_{new}^{c+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.01 & 0 & 0 & -1.3571\times10^{-4} & 0 \\ 0 & 0 & 1 & 0 & 0 & -1.3571\times10^{-2} & 0.0111K_1 \\ 0 & 0 & 0 & 1 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -0.0785\times10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.0167K_1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
2	$x_2(k-1) > 0$ & $x_4(k-1) - x_2(k-1) < 0$	$J_{new}^{c+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.01 & 0 & 0 & -1.3571\times10^{-4} & 0 \\ 0 & 0 & 1 & 0 & 0 & -1.3571\times10^{-2} & 0.0111K_2 \\ 0 & 0 & 0 & 1 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.0785\times10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.0167K_2 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$
3	$x_2(k-1) < 0$ & $x_4(k-1) - x_2(k-1) > 0$	$J_{new}^{c+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.01 & 0 & 0 & 1.3571\times10^{-4} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1.3571\times10^{-2} & 0.0111K_3 \\ 0 & 0 & 0 & 1 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -0.0785\times10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.0167K_3 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

4	$x_2(k-1) < 0$ & $x_4(k-1) - x_2(k-1) < 0$	$J_{new}^{c+} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0.01 & 0 & 0 & 1.3571\times10^{-4} & 0 \\ 0 & 0 & 1 & 0 & 0 & 1.3571\times10^{-2} & 0.0111K_4 \\ 0 & 0 & 0 & 1 & 0.01 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0.0785\times10^{-2} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -0.0167K_4 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$

Step 4. Obtain the fundamental equation of the system.
$m_c(n) = m_c(n-1) + J^c v$
Step 5. Calculate K_i as the system is stabilized with the Lyapunov method.
$V = m_c^T(n) P m_c(n) > 0$
$$\Delta V = m_c^T(n-1)(J^{c+T} P J^{c+} - P) m_c(n-1) < 0 \Rightarrow \Delta V = m_c^T(n-1)(-Q) m_c(n-1)$$
Here, the genetic algorithm method is used to calculate the control coefficients. The genetic algorithm must satisfy Equation (3) and also minimize the following fitness function:

$$F_{obj} = \left(x_4(t) - x_{4d}(t)\right)^2 \quad (21)$$

The parameters of the genetic algorithm for the system are shown in Table 4.

The convergence trends in the GA for the controller are shown in Figure 7.

Consequently, the K_i for each of the modes are given in Table 5.

5. SIMULATION RESULTS

In this section, the performance of the controller design algorithm will be presented using a control place based on the HTDPN model. In this paper, to investigate the performance of the introduced method, this method is compared with the MPC. Figure 8 shows the capsubot step response in the proposed approach and MPC.

A comparison of the results in Figure 8 shows that the HTDPN response is stable and the proposed control method converges faster than MPC.

Figure 9 and Figure 10 depict capsubot velocity and inner mass velocity, respectively.

These figures show that the inner mass velocity and robot velocity of the proposed method has less oscillation and is smoother than the predictive control method; therefore, the result can be easily implemented.

The input signal in the proposed approach and MPC method is shown in Figure 10.

Figure 11 shows that the input signal in the proposed approach is smoother than the predictive control method. This is while the input signal peak is higher in the presented method. Energy consumption can also be calculated as follows:

$$W = \sum_{i=1}^{n} U(i).\Delta\theta(i) \quad (22)$$

TABLE 4. The Parameters of the GA.

Generations	Elite Count	Crossover Fraction
50	3	0.7

TABLE 5. Control coefficients of the capsule using Lyapunov theorem.

Mode	Condition	Gain
1	$x_2(k-1) > 0$ & $x_4(k-1) - x_2(k-1) > 0$	$K_1 = -8.78$
2	$x_2(k-1) > 0$ & $x_4(k-1) - x_2(k-1) < 0$	$K_2 = -17.436$
3	$x_2(k-1) < 0$ & $x_4(k-1) - x_2(k-1) > 0$	$K_3 = -16.121$
4	$x_2(k-1) < 0$ & $x_4(k-1) - x_2(k-1) < 0$	$K_4 = -16.508$

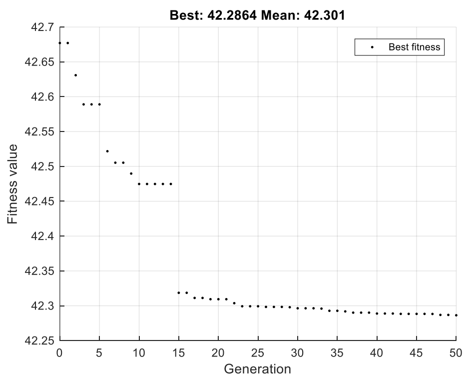

Figure 7. The GA convergence trend in the controller

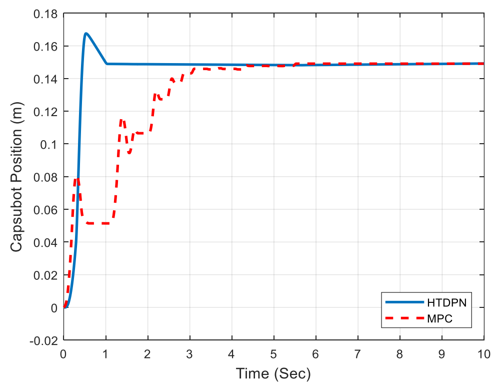

Figure 8. The output of the proposed approach and MPC

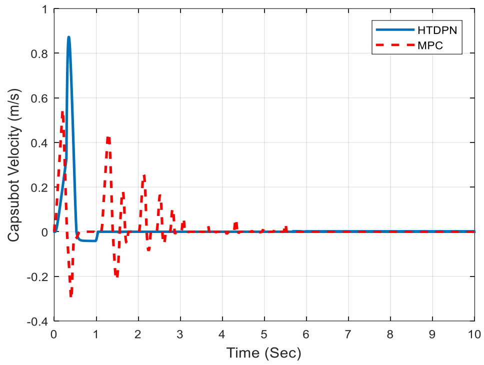

Figure 9. The capsubot velocity

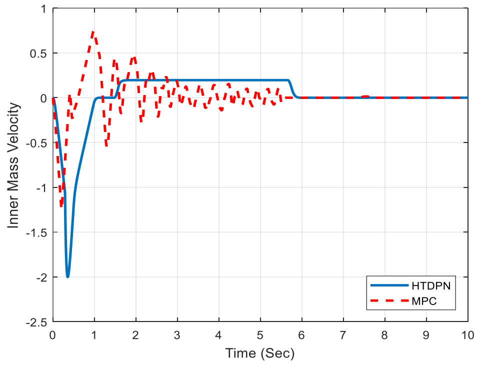

Figure 10. The inner mass velocity

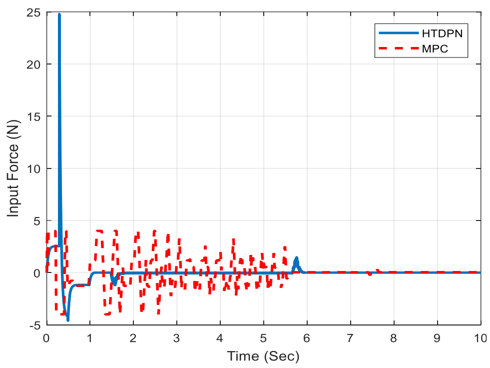

Figure 11. The input signal of the proposed approach and MPC method

The integral absolute error (IAE) is commonly used in the design and evaluation of practical control systems' performance. The IAE is calculated as follows:

$$IAE = \sum_{i=1}^{n} |e(i)| \tag{23}$$

Table 6 shows a comparison of different criteria for the four simulated methods.

Here, for a more accurate comparison of these methods in Table 5, these parameters are normalized as below:

$$S = \frac{Design\ parameter\ value}{Desired\ value} \tag{24}$$

The total of numbers is a criterion for comparing these methods. Table 7 illustrates these normalized criteria.

As Table 7 clearly shows, according to the design criteria, the control place scheme for dynamic systems has proper performance in comparison to the MPC. This controller is designed according to the capacity of the HTDPN tool. Therefore, the designer can perform the desired controller in the graphical environment of the HTDPN tool. Finally, the design result is applied in the incidence matrix for use in the simulation. In addition, the simulation of the system mentioned above by our novel algorithm and the conventional one via the same hardware configuration relays a significant advantage of the new method, which is time efficiency. For future works, it is suggested that other control design algorithms such as fuzzy and optimization methods be implemented

TABLE 6. Comparison of the controller design criteria for the different methods.

Method	Proposed method	MPC	Optimal Control	CLC
Energy Consumption (J)	0.1617	0.352	0.1372	0.3932
IAE (m)	5.572	19.075	581.8096	56.087
Rise Time (S)	0.2119	2.406	6.8857	6.8
Settling Time (S)	0.4651	4.319	8.4218	8.24
Force Peak (N)	25	4	1.518	4.4158
Run Time (S)	0.7336	0.951	0.9689	0.8974

TABLE 7. Comparing the normalized parameters for the different methods

Method / Normalized Values	Proposed method	MPC	Optimal Control	CLC
Energy Consumption	1.1794	2.5675	1	2.865
IAE	1	3.4234	104.4167	10.0658
Rise Time	1	20.366	32.4675	32.091
Settling Time	1	9.2851	18.1159	17.7166
Force Peak	16.4745	2.635	1	2.91
Run Time	1	1.2963	1.3208	1.223
Sum of NV	21.6539	39.5733	158.3209	66.871

with the HTDPN tool and the results be compared with the proposed method.

6. CONCLUSION

In this paper, a novel method for controller design was presented in the environment of the HTDPN tool. In this approach, the desired controller was designed by adding control places in the graphical environment of the HTDPN model system. Using the properties of the HTDPN tool, the controller designed in the graphical environment was transferred to the mathematical environment. Here, the control place technique used in the design of supervisory control in conventional Petri Nets was extended to a HTDPN. Then, by applying Lyapunov's theory to the incidence matrix, the coefficients of the controller were extracted. This controller guarantees that the system was stable. In addition, the control place inputs were determined by the GA method. To prove the performance of the controller, this method was implemented in the capsubot model. It was obvious that this method simplified system analysis and controller design. In addition, due to the use of environment matrices for changes of system states and algebraic operations instead of solving equations, the proposed approach provides a faster mathematical algorithm that can reduce simulation time and complexity for complex systems. Additionally, the results clearly showed that this approach could improve the performance of the controller design.

For future work, it is suggested that the analysis of some system properties, such as controllability and visibility, which are extracted using the state space, should also be investigated and analyzed in modeling using the HTDPN tool.

7. REFERENCES

1. Yin, L., Fang, H. and Shao, H., "Design and implementation of petri net for brain-computer interface system", in 2019 Chinese Automation Congress (CAC), IEEE. (2019), 5810-5814.
2. Du, N., Hu, H. and Zhou, M., "Robust deadlock avoidance and control of automated manufacturing systems with assembly operations using petri nets", ***IEEE Transactions on Automation Science and Engineering***, Vol. 17, No. 4, (2020), 1961-1975. doi: 10.1109/TASE.2020.2983672.
3. YadollahzadehTabari, M. and Mohammadizad, P., "Modeling and performance evaluation of energy consumption in s-mac protocol using generalized stochastic petri nets", ***International Journal of Engineering, Transactions C: Aspects***, Vol. 33, No. 6, (2020), 1114-1121. doi: 10.5829/IJE.2020.33.06C.08.
4. Liu, F., Chen, S., Heiner, M. and Song, H., "Modeling biological systems with uncertain kinetic data using fuzzy continuous petri nets", ***BMC Systems Biology***, Vol. 12, (2018), 63-74. doi: 10.1186/s12918.018.0568.8.
5. Feng, Y., Xing, K., Zhou, M., Wang, X. and Liu, H., "Robust deadlock prevention for automated manufacturing systems with unreliable resources by using general petri nets", ***IEEE Transactions on Systems, Man, and Cybernetics: Systems***, Vol. 50, No. 10, (2018), 3515-3527. doi: 10.1109/TSMC.2018.2884316.
6. Li, J., Yu, X. and Zhou, M., "Analysis of unbounded petri net with lean reachability trees", ***IEEE Transactions on Systems, Man, and Cybernetics: Systems***, Vol. 50, No. 6, (2018), 2007-2016. doi: 10.1109/TSMC.2018.2791527.
7. Gao, X. and Hu, X., "A petri net neural network robust control for new paste backfill process model", ***IEEE Access***, Vol. 8, No., (2020), 18420-18425. doi: 10.1109/ACCESS.2020.2968510.
8. Demongodin, I. and Koussoulas, N.T., "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, (2006), 543-553. doi: 10.1109/TSMCC.2005.848154.
9. Demongodin, I. and Koussoulas, N.T., "Differential petri nets: Representing continuous systems in a discrete-event world", ***IEEE transactions on Automatic Control***, Vol. 43, No. 4, (1998), 573-579. doi: 10.1109/9.665073.
10. Demongodin, I. and Koussoulas, N.T., "Modeling of mixed continuous/discrete-event systems via differential petri nets", in Proceedings of Third International Conference on Electronics, Circuits, and Systems, IEEE. Vol. 1, (1996), 475-478.
11. Saleh, A., Chiachío, M., Salas, J.F. and Kolios, A., "Self-adaptive optimized maintenance of offshore wind turbines by intelligent petri nets", ***Reliability Engineering & System Safety***, Vol. 231, (2023), 109013. doi: 10.1016/j.ress.2022.109013.
12. Ruan, K. and Li, L., "Traffic network modeling and volume control using labeled petri nets", in 2021 IEEE International Intelligent Transportation Systems Conference (ITSC), IEEE. (2021), 3578-3583.
13. Taleb, M., Leclercq, E. and Lefebvre, D., "Model predictive control for discrete and continuous timed petri nets", ***International Journal of Automation and Computing***, Vol. 15, (2018), 25-38. doi: 10.1007/s11633-016-1046-7.
14. Caligola, S., Carlucci, T., Fummi, F., Laudanna, C., Constantin, G., Bombieri, N. and Giugno, R., "Efficient simulation and parametrization of stochastic petri nets in systemc: A case study from systems biology", in 2019 Forum for Specification and Design Languages (FDL), IEEE. (2019), 1-7.
15. Ding, Z., Xiao, L. and Hu, J., "Performance analysis of service composition using ordinary differential equations", in 2008 12th IEEE International Workshop on Future Trends of Distributed Computing Systems, IEEE. (2008), 30-36.
16. Dideban, A. and Ahangarani Farahani, A., "A new adaptive controller for the three axis satellite simulator based on continuous time delay petri nets tool", ***Space Science and Technology***, (2022). doi: 10.22034/jsst.2022.314634.1376.
17. Dideban, A., Farahani, A.A. and Razavi, M., "Modeling continuous systems using modified petri nets model", ***Modeling and Simulation in Electrical and Electronics Engineering (MSEEE), Semnan, Iran***, Vol. 1, No. 2, (2015), 75-79. doi: 10.22075/MSEEE.2015.247.
18. Dideban, A. and Sabouri Rad, M., "Electrical circuit modelling with petri net by using of control arcs", ***Journal of Modeling in Engineering***, Vol. 11, No. 35, (2014), 39-47. doi: 10.22075/JME.2017.1657.
19. Farahani, A.A., Dideban, A. and Najafgholi, E., "Modeling continuous systems by petri nets using speed control arcs", in 2016 4th International Conference on Control, Instrumentation, and Automation (ICCIA), IEEE. (2016), 75-80.

20. Ahangarani, A. and Dideban, A., "Continuous-time delay-petri nets as a new tool to design state space controller", ***Information Technology and Control***, Vol. 45, No. 4, (2016), 401-411. doi: 10.5755/j01.itc.45.4.13665.
21. Ahangarani Farahani, A. and Dideban, A., "Hybrid time delay petri nets as a mathematical novel tool to model dynamic system with current sample time", ***Control and Optimization in Applied Mathematics***, Vol. 3, No. 1, (2018), 45-64. doi: 10.30473/coam.2019.41925.1090.
22. Ma, Z., Zou, M., Zhang, J. and Li, Z., "Design of optimal control sequences in petri nets using basis marking analysis", ***IEEE transactions on Automatic Control***, Vol. 67, No. 7, (2021), 3685-3692. doi: 10.1109/TAC.2021.3106883.
23. Othman, S., Alali, M.A., Sbita, L., Barbot, J.-P. and Ghanes, M., "Modeling and control design based on petri nets tool for a serial three-phase five-level multicellular inverter used as a shunt active power filter", ***Energies***, Vol. 14, No. 17, (2021), 5335. doi: 10.3390/en14175335.
24. Bashir, M., Zhou, J. and Muhammad, B.B., "Optimal supervisory control for flexible manufacturing systems model with petri nets: A place-transition control", ***IEEE Access***, Vol. 9, (2021), 58566-58578. doi: 10.1109/ACCESS.2021.3072892.
25. Chen, C. and Hu, H., "Extended place-invariant control in automated manufacturing systems using petri nets", ***IEEE Transactions on Systems, Man, and Cybernetics: Systems***, Vol. 52, No. 3, (2020), 1807-1822. doi: 10.1109/TSMC.2020.3035668.
26. Baniardalani, S., "Fault diagnosis of discrete-time linear systems using continuous time delay petri nets", ***International Journal of Industrial Electronics Control and Optimization***, Vol. 3, No. 1, (2020), 81-90. doi: 10.22111/ieco.2019.28395.1130.
27. Qiu, Z., Duan, C., Yao, W., Zeng, P. and Jiang, L., "Adaptive lyapunov function method for power system transient stability analysis", ***IEEE Transactions on Power Systems***, (2022). doi: 10.1109/TPWRS.2022.3199448.
28. Ashjaee, M. and Tavazoei, M. S., "Optimal tuning of implementable fractional order PI controllers based on ISE performance index", ***Sharif Journal of Mechanical Engineering***, Vol. 35, No. 1, (2019), 133-141. doi: 10.24200/J40.2018.10795.1431.

8. APPENDIX

The proof of the property 1 is as follows:
Proof: Due to the difference equations, the coefficient of the first side of the equation is equal to 1, therefore, in the CTDPN model, this relation always holds; $Pre(p_i, t_j) = 1$, therefore:

$$q(t_j, m) = \min_{i:p_i \in {}^{\circ}t_j} \left(\frac{M(p_i)}{Pre(p_i,t_j)}\right) = \min_{i:p_i \in {}^{\circ}t_j} \left(M(p_i)\right) \tag{25}$$

Since in the CTDPN, the maximum speed of a transition is assumed infinity; therefore, we can suppose that all of the tokens in the places before a transition tj are discharged at time T_s and then the transitions speed is a function of enabling degree for this transition. Therefore:

$$\begin{gathered} \int_{t_1}^{t_1+T_s} v_j(t)dt = \min_{i:p_i \in {}^{\circ}t_j} \left(M(p_i)\right) \\ \int_{t_1}^{t_1+T_s} v_j(t)dt = v_j(t_1)(t_1 + T_s - t_1) = \\ v_j(t_1)(T_s) \Rightarrow v_j(t_1) = \frac{M(p_i)}{T_s} \end{gathered} \tag{26}$$

The proof of the property 2 is as follows:
Proof: The fundamental equation for timed Continuous Petri Nets between times t_1 and t_2 is as follows:

$$m(t_2) = m(t_1) + \int_{t_1}^{t_2} Wv(t)dt \tag{27}$$

If $t_2 = nT_s$ and $t_1 = t_2 - dt = (n-1)T_s$ the following can be written:

$$m(nT_s) = m\left((n-1)T_s\right) + W \int_{(n-1)T_s}^{nT_s} v(t)dt \tag{28}$$

Here, T_s is sample time.
By property 1, the following holds true [26]:

$$\int_{(n-1)T_s}^{nT_s} v(t)dt = m\left((n-1)T_s\right) \tag{29}$$

Substituting Equation (29) into Equation (28) and rewriting it gives:

$$m(nT_s) = m\left((n-1)T_s\right) + Wm\left((n-1)T_s\right) \tag{30}$$

Therefore, the fundamental equation of HTDPN can be obtained as:

$$m(n) = m(n-1) + Wm(n-1) \tag{31}$$

Persian Abstract

چکیده

هدف از این مقاله معرفی یک روش جدید برای طراحی کنترل‌کننده با استفاده از کنترل مکان در سیستم مدل شده با ابزار شبکه پتری زمان تأخیری ترکیبی است. اغلب در سیستم‌های مدل شده با شبکه پتری گسسته، از کنترل مکان برای طراحی کنترل نظارتی سیستم استفاده می‌شود. در حالیکه، رویکرد جدید معرفی شده در این مقاله از کنترل مکان برای کنترل سیستم‌های دینامیکی مدل شده با ابزار شبکه پتری زمان تأخیری ترکیبی استفاده می‌کند. این کنترل‌کننده شامل، مکان‌های کنترلی، گذرگاه‌ها، کمان‌های متصل شده به مکان‌های کنترلی است که به مدل سیستم اضافه شده و سیستم را کنترل می‌کند. در این مقاله، برای طراحی کنترل‌کننده سه گام باید انجام گیرد. در گام اول سیستم با ابزار شبکه پتری زمان تأخیری ترکیبی مدل می‌شود، و در مرحله دوم با استفاده از روش جدید ارائه شده یک کنترل‌کننده طراحی می‌شود. در نهایت، وزن کمان‌های متصل به مکان‌های کنترلی با استفاده از تئوری تابع لیاپانوف محاسبه می‌شود که پایداری حلقه بسته را تضمین می‌کند. مزیت اصلی این روش امکان استفاده همزمان از مکان‌های کنترلی پیوسته و گسسته برای کنترل سیستم‌های غیرخطی است. برخلاف اکثر رویکردهای قبلی، در روش پیشنهادی، یک طراح خبره می‌تواند یک کنترل‌کننده مطلوب را در محیط گرافیکی ایجاد کند و سپس تغییراتی را در محیط ریاضی مدل شبکه پتری مشاهده نماید. برای بررسی کارایی کنترل‌کننده پیشنهادی، این روش با استفاده از معیارهای خطا، مصرف انرژی، زمان صعود، زمان نشست و زمان شبیه‌سازی با مقایسه ارزیابی شده‌است. نتایج شبیه‌سازی نشان می‌دهد که روش پیشنهادی به ترتیب ۴۵ و ۶۰۰ درصد شرایط بهتری نسبت به روش کنترل پیش‌بین و کنترل بهینه دارد.