Payame Noor University

Control and Optimization in Applied Mathematics (COAM)
Vol. 3, No. 1, Spring-Summer 2018(45-64), ©2016 Payame Noor University, Iran

Hybrid Time Delay Petri Nets as a Mathematical Novel Tool to Model Dynamic System with Current Sample Time

A. Ahangarani Farahani [1], A. Dideban[2,*]

[1,2]Electrical and Computer Engineering Department, Semnan University, P.O. Box. 35131-19111, Semnan, Iran

Received: January 02, 2019; **Accepted:** September 23, 2019.

Abstract. The existing modeling methods using Petri Nets, have been successfully applied to model and analyze dynamic systems. However, these methods are not capable of modeling all dynamic systems such as systems with the current sample time signals, systems including various subsystems and multi-mode systems. This paper proposes Hybrid Time Delay Petri Nets (HTDPN) to solve the problem. In this approach, discrete and continuous Petri Nets are combined so that the continuous PNs part and the discrete PNs are responsible for past time samples and current sample time, respectively. To evaluate the performance of the proposed tool, it is employed to model a legless piezoelectric capsubot robot as a multi modes system and a PID controller, in which the gains tuned by the Genetic Algorithm are designed for the resulting model by HTDPN. Results show that the proposed method is faster in terms of mathematical calculations which can reduce the simulation time and complexity of complicated systems. It would be observed that the proposed approach makes the PID controller design simpler as well. In addition, a comparative study of capsubot has been performed. Simulation results show that the presented method is encouraging compared to the predictive control, which is used in the literature.

Keywords. Hybrid Petri Nets, Current sample time signals, Capsubot robot, Genetic algorithm.

MSC. 90C34; 90C40.

* Corresponding author
a.ahangarani@semnan.ac.ir, adideban@semnan.ac.ir
http://mathco.journals.pnu.ac.ir

1 Introduction

The complex nature of modern systems described by differential equations has motivated designers to develop methods and tools for the modeling, analysis, performance evaluation and control of such systems. One of the most successful modeling approaches has been Petri Nets. Petri Nets are useful for scheduling and supervisory control problems in the discrete event system [1]- [3]. The concept of Petri Nets was introduced by Dr Carl Adam Petri in his Ph.D. thesis in 1962 [4]. Utilizing the important features of Petri Nests, i.e. graphical and distributed system representation, has caused its extensive usage in modeling, analysis and synthesis of dynamic systems, particularly discrete event systems. Furthermore, it enhances computational efficiency [5].
On the other hand, in situations where the number of tokens increases more and more, continuous Petri Nets is a handy tool for modeling the discrete event system. Continuous Petri Nets has been defined by David and Alla in [6], and it has been proposed that this method has been successfully applied for the modeling and analyses of metabolic networks and traffic flow [7]-[8]. Moreover, a flow may suddenly be interrupted which can then be modeled by hybrid Petri Nets [9]. It is obvious that not all system variables generally flow in nature and can be described by differential equations and those tools that are mentioned above are not effective to be applied to all dynamic systems [10].
Previous attempts have been made to model continuous linear dynamic systems using Petri Nets [4], [11]-[13]. In [4], [13], to model a continuous dynamic system, speed arc control is defined which is not a principle of Petri Nets. In this method, the speed of each transition is tuned by its previous place and applied by a new element "speed arc control" to the transition. Therefore, new elements and definitions are added to the conventional Petri Nets. On the other hand, these approaches are limited to constraints that make them rather difficult to be analyzed. In [14], it is shown how hybrid Petri Nets can be used to describe a dynamic system with a unified mathematical description. The presented tool in [14] is defined as state jump, which can be used to jump in the state space and switches in the dynamic system. This approach is not able to provide a straightforward relation between the graphical model, which is created from system dynamics and mathematics governing hybrid Petri Nets. A tool for model and controller design based on the Petri Net is introduced in [11]. The proposed approach in [11] has some limitations, which consequently has no generality and cannot be used in the modelling and controller designing procedure for all dynamic systems. The main limitation of this approach is that it cannot model systems such as multi-mode systems, those that include switchable subsystems and those with current input sample signals. This paper develops a novel tool that is a new kind of Petri Net called Hybrid Time Delay Petri Nets (HTDPN) to address the mentioned problems. Here, a combination of continuous and discrete Petri Nets are used. The continuous Petri Nets models differential equations while discrete Petri Nets model the current sample time, switching between subsystems and the changing system modes. Based on this new tool, run time is reduced and helps to simplify the controller designing procedure. Simplifying controller design for multi-mode systems can be considered as one of the main advantages of this approach.

Consequently, a close-loop system including the capsubot robot which is an underactuated nonlinear dynamics system and a PID controller, modeled by HTDPN, employs the GA algorithm to optimally obtain and tune three terms of the PID controller, minimizing the defined fitness function. Then a comparison is made between the proposed approach and four control schemes. This paper is organized as follows: Section 2 introduces the main concepts of Hybrid Petri Nets and PID controller. The basic definition of the HTDPN is proposed in Section 3. Section 4 presents the PID and capsubot modeling by HTDPN. Simulation results are presented in Section 5 and finally the conclusion is given in Section 6.

2 Preliminary Definition

2.1 Hybrid Petri Nets

A marked Hybrid Petri Nets is a six-tuple $HPN = \{P, T, W^-(Pre), W^+(Post), M_0, h\}$ such that [15]-[16]
$P = \{p_1, p_2, \cdots, p_n\}$ is a nonempty and finite set of places.
$T = \{t_1, t_2, \cdots, t_f\}$ is a nonempty and finite set of transitions and $P \bigcap T = \varnothing$, i.e. sets P and T are disjointed.
$W^-(Pre) : (P \times T) \to R \geq 0$ or N^+ and $W^+(Post) : (T \times P) \to R \geq 0$ or N^+ are the weight of arcs from place to transition and transition to place, respectively [17].
M_0 is the initial condition of marking and $h : P \cup T \to \{C, D\}$ is a hybrid function which indicates whether each node is a discrete node or a continuous node.
A discrete transition is enabled when each place $p_i \in^o t_j$ meets the condition [18]:

$$|M(p_i)| > Pri(p_i, t_j)$$

A continuous transition is enabled if each place $p_i \in^o t_j$ meets the condition:
1-For discrete place p_i is:

$$m(p_i) > Pri(p_i, t_j)$$

2-And for continuous place p_i is:

$$m(p_i) > 0$$

2.2 PID Controller

A PID is a controller with three-terms, which is the sum of three types of control actions. The combination of the proportional, integral and derivative term actions yields:

$$\frac{U_s}{E(s)} = K_P + \frac{K_I}{T_i s} + K_D T_d s \tag{1}$$

Figure 1 shows the block diagram of the PID controller and a plant.

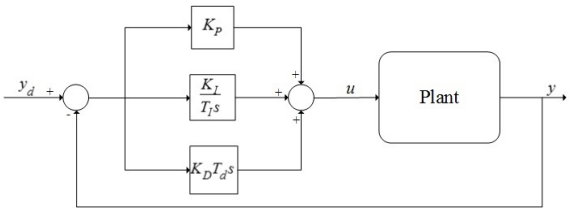

Figure 1: Block diagram of the PID controller and a plant.

2.3 Discrete PID Controller

To implement the PID controller using a digital computer we have to convert (1) from a continuous to a discrete representation. The transfer function of a discrete PID controller is given by [19]:

$$\begin{aligned} D(z) = \frac{U_z}{E(z)} = K_P + K_I \frac{T_s}{2}\frac{z+1}{z-1} + K_D \frac{z-1}{T_s z} = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 - z^{-1}} \\ \Rightarrow u_{PID}(k) = u_{PID}(k-1) + a_0 e(k) + a_1 e(k-1) + a_2 e(k-2) \end{aligned} \tag{2}$$

Where

$$\begin{cases} a_0 = K_P + \frac{K_I T_s}{2} + \frac{K_D}{T_s} \\ a_1 = -K_P + \frac{K_I T_s}{2} + \frac{2K_D}{T_s} \\ a_2 = \frac{K_D}{T_s} \end{cases}$$

3 Hybrid-Time Delay-Petri Nets (HTDPN)

Many systems in nature have dynamic equations. For studying the systems' features, the user must be able to model dynamic systems and analyze dynamic characteristics. Petri Nets is not useful to model continuous dynamic systems. Moreover, controller implementation in Petri Nets requires current sample time such as error signal, which is not calculable by Petri Nets. To overcome these problems, HTDPN is presented in this paper.

Definition 1. A Hybrid Time Delay Petri Net (HTDPN) is a 7-tuple as:

$$PNH = \{P, T, Pred, Postd, Prec, Postc, M_0\}$$

such that:
$P = \{p_1, p_2, \cdots, p_n\}$ represent a finite set of continuous places.
$T = \{t_1, t_2, \cdots, t_m\}$ represent a finite set of transitions, here, $T_c = \{t_1, t_2, \cdots, t_{m'}\}$ denotes the set of the $m^{'}$ continuous transitions which are represented by boxes.
$T^D = T - T^C$ also denotes the sets of discrete transitions which are depicted by bars.

Pre_d, $Post_d$ are the incidence functions that specify the multiplicity of arcs between places and discrete transitions.

Moreover, Pre_c, $Post_c$ are the incidence functions which specify the multiplicity of arcs between places and continuous transitions.
M_0 is the negative or non-negative real-valued initial marking vector. To model a continuous dynamic system using HTDPN, the following rules should be considered:
Rule 1: Here, discrete transitions T^D are initially executed before continuous transitions T^C.
Rule 2: Time delays correspond to continuous transitions T^C; however, the discrete transitions are executed immediately.
Rule 3: While places portray states of the system, in the HTDPN $M \in R$.
Rule 4: Firing of discrete transitions follows the two principles described below:

1- A discrete transition $t_j \in T^D$ at marking M is enabled, i.e., it can fire, iff $\forall p_i \in^o t_j$.

$$|M(p_i)| > 0$$

2- A discrete transition t_j starts firing as soon as it is enabled.

Rule 5: Firing of continuous transitions satisfies the following two principles:

1- A continuous transition $t_j \in T^C$ is enabled, i.e., it can fire, iff

$$|M(p_i)| > 0, \forall p_i \in^o t_j$$

2- The firing of a continuous transition t_j is not instantaneous.

Rule 6: After firing of a transition, the tokens of input places reach zero.
Rule 7: Since in the HTDPN the maximum possible speed of transition is assumed as infinity, the transition speed in continuous transitions is determined by the input place connected to the transition.
Rule 8: Unlike ordinary Petri Nets, the arcs are a passageway of signals and can thus be negative or positive.
In this approach, the fundamental equation consists of two parts: the first part is a discrete one and the second part is a continuous one and is executed in two steps. In the first step, the fundamental equation is executed for the discrete Petri Nets model shown in (3):

$$M^{'} = M(k-1) + W_d S \tag{3}$$

$M^{'}$ is the vector of marking calculated after firing discrete transitions and before continuous transitions. The discrete incidence matrix and firing vector are shown by $W_d = Post_d - Pre_d$ and S, respectively.
In the second step, the equation is executed for the CPN model explained in (4).

$$M(k) = M^{'} + W_c v \tag{4}$$

where $W_c = Post_c - Pre_c$ is the continuous incidence matrix and v is firing speed.
To implement this approach, first, continuous differential equations are converted to difference

equations and then by using HTDPN, the system is modeled. In discrete state space, a recursive function including delays is used instead of the derivative functions, which makes the modeling process simpler.
In this method, places play the role of input and output variables and states for systems.
To illustrate this method, consider a first-order state space with past time samples and current sample time as (5).

$$x(k) = \alpha x(k-1) + \beta u(k-1) + \gamma u(k) \tag{5}$$

where, $u(k)$ is the current sample time signal. The HTDPN model of (5) is shown in Figure 2.

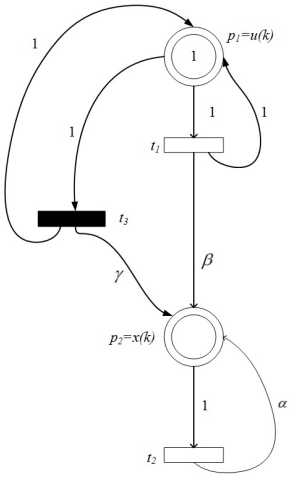

Figure 2: The HTDPN model of (5).

In Figure 2, place p_1 is the input variable and place p_2 depicts the output variable.The discrete fundamental equation for this model is:

$$M^{'} = M(k-1) + W_d S \tag{6}$$

where

$$W_d = \begin{bmatrix} 0 \\ \gamma \end{bmatrix} S = \begin{bmatrix} M(p_1(k-1)) \end{bmatrix}$$

And the continuous fundamental equation is explained in (7).

$$M(k) = M^{'} + W_c v \tag{7}$$

where

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

and

$$v = \begin{bmatrix} M(p_1) \\ M(p_2) \end{bmatrix}$$

It could be perceived that the graphical and mathematical description of the dynamic systems reduces the complexity of analysis.

Definition 2. The synchronization is defined as follows: assume m_1 is a token in place p_1, m_2 is a token in place p_2 and m_n is a token in place p_n and all of them are before a transition t_j, then the instantaneous firing speed of a transition is calculated as below:

$$v_j = min(|m_1|, |m_2|, \cdots, |m_n|) \tag{8}$$

In Figure 3 the two input places of t_1 are synchronized for the firing of transition t_1.

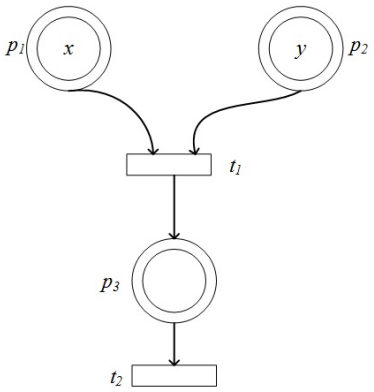

Figure 3: Synchronization in the HTDPN.

Here, after firing place p_3 is equal to a minimum of two places p_1 and p_2. The equation in a HTDPN is:

$$M(k) = M(k-1) + Wv \tag{9}$$

Where

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

and

$$v = \begin{bmatrix} min(|m_1|, |m_2|) \\ m_3 \end{bmatrix}$$

Definition 3. Concurrency is characterized by the existence of a forking transition that deposits tokens simultaneously in two or more output places. In Figure 4, transitions t_1, t_2 and t_3 are concurrent. In Figure 4, t_0 is the forking transition.

The state equation is:

$$M(k) = M(k-1) + Wv \tag{10}$$

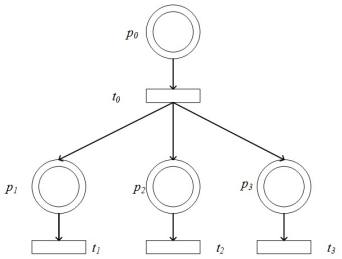

Figure 4: Concurrency in the HTDPN.

Where

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

and

$$v = \begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ m_4 \end{bmatrix}$$

Figure 5 presents the reachability graph of concurrency in the HTDPN.

$$m(0) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \xrightarrow{T_0} m(1) = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix} \xrightarrow{T_1,T_2,T_3} m(2) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Figure 5: Reachability graph of Concurrency in the HTDPN.

4 Modeling of PID Controller by HTDPN

For modeling the PID controller, its terms can be modeled by HTDPN and then integrated together.

4.1 Proportional Term

The transfer function of the proportional term is:

$$G_P(z) = K_P \Rightarrow \frac{U_P(z)}{E(z)} = K_P \Rightarrow U_P = K_P E(z) \tag{11}$$

The input-output properties of (11) can be described by:

$$u_P = K_P e(k) \tag{12}$$

The P controller that is modeled by HTDPN is shown in Figure 6.

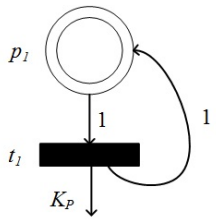

Figure 6: The HTDPN model of P controller.

4.2 Integral Term

The integral term has the transfer function [20]:

$$\begin{aligned} &G_I(z) = K_I \tfrac{1}{z-1} \Rightarrow \\ &G_I(z) = \tfrac{U_I(z)}{E(z)} = K_I \tfrac{z^{-1}}{1-z^{-1}} \Rightarrow \\ &\tfrac{U_I}{E(z)} = K_I \tfrac{z^{-1}}{1-z^{-1}} \Rightarrow \\ &U_I(z) = z^{-1}U_I(z) + K_I^{-1}E(z) \end{aligned} \tag{13}$$

The input-output equation can be given by:

$$u_I(k) = u_I(k-1) + K_I e(k-1) \tag{14}$$

Figure 7 shows the HCTDPN model of the integral term.

4.3 Derivative Term

The transfer function of D is [20]:

$$\tfrac{U_D(z)}{E(z)} = K_D(1 - z^{-1}) \Rightarrow U_D(z) = K_D E(z) - K_D z^{-1} E(z) \tag{15}$$

From the given transfer function, the input-output property for D is as (16):

$$u_D(k) = K_D e(k) - K_D e(k-1) \tag{16}$$

Using (16), the HCTDPN model is shown in Figure 8.

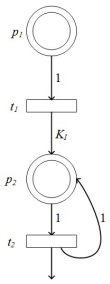

Figure 7: The HTDPN model of I controller.

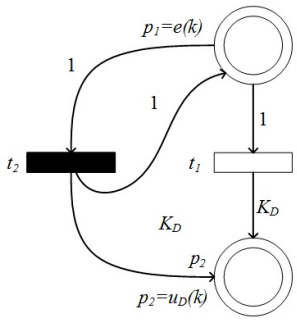

Figure 8: The HTDPN model of D controller.

4.4 Integration of P, I and D

The difference equation of PID can be written as:

$$u_{PID}(k) = u_{PID}(k-1) + a_0 e(k) + a_1 e(k-1) + a_2 e(k-2) \tag{17}$$

where, $a_0 e(k)$ is the current sample time signal. Figure 9 shows the PID model using HTDPN. It consists of proportional, integral and derivative terms.

4.5 Capsubot dynamic model

The simplified schematic model of the legless piezo capsule robot is depicted in Figure 10 [21].

The robot movement is driven by an internal impact force and friction; therefore, this has a complex dynamic model. This robot consists of three main parts: the capsubot shell, the inner mass and piezoelectric element which generate force. The relation of the capsubot system with the second law of Newton is as follow:

$$\begin{aligned} F_M &= M\ddot{x_1} \Rightarrow u - \mu_{1k}(M+m).g.sign(\dot{x_1}) + \mu_{2k} m.g.sign(\dot{x_1} - \dot{x_2}) = M\ddot{x_1} \\ F_m &= m\ddot{x_2} \Rightarrow m\ddot{x_2} + \mu_{2k} m.g.sign(\dot{x_1} - \dot{x_2}) = -u \end{aligned} \tag{18}$$

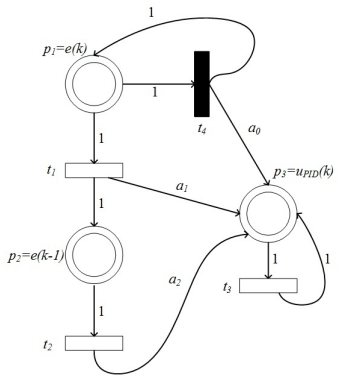

Figure 9: The HTDPN model of PID controller.

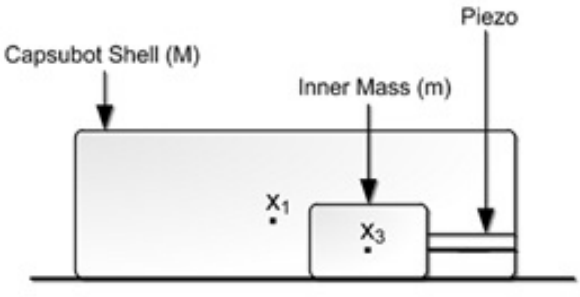

Figure 10: The schematic of the legless piezo capsule robot.

where x_1 is position of the inner mass (m) and x_2 is position of the capsubot shell with mass M. The kinetic friction coefficient between the capsubot shell and the ground and capsubot shell and the inner mass are μ_{1k} and μ_{2k}, respectively and g is gravitational acceleration. Therefore, the state space equations can be written as follow [22]:

$$\dot{X}(t) = A_c X(t) + B_c u(t) + f_c(t) \tag{19}$$

Where

$$X(t) = \begin{bmatrix} x_1(t) & x_2(t) & x_3(t) & x_4(t) \end{bmatrix}^T$$

and

$$X(t) = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, B_c = \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{m} \end{bmatrix}$$

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

Table 1: Parameters of the capsubot robot.

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

The parameters of the capsubot robot used are given in Table 1 [23].

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

$$\begin{cases} \dot{x_1}(t) = x_2(t) \\ \dot{x_2}(t) = -1.3571 sign(x_2) + 0.0523 sign(x_4 - x_2) + 1.111u \\ \dot{x_3}(t) = x_4(t) \\ \dot{x_4}(t) = -0.0785 sign(x_4 - x_2) - 1.6667u \end{cases} \tag{20}$$

After converting the dynamic system from continuous-time system to discrete-time system with the sample time $T_s = 0.01(sec)$, the resulting state space is [24]:

$$\begin{cases} x_1(k) = x_1(k-1) + 0.01x_2 - 1.3571 \times 10^{-4} sign(x_2(k-1)) \\ x_2(k) = x_2(k-1) - 1.3571 \times 10^{-2} sign(x_2(k-1)) + 0.0111u(k-1) \\ x_3(k) = x_3(k-1) + 0.01x_4(k-1) \\ x_4(k) = x_4(k-1) - 0.0785 \times 10^{-2} sign(x_4(k-1) - x_2(k-1)) - 0.0167u(k-1) \end{cases} \tag{21}$$

Finally, the HTDPN with step input for this model has been demonstrated in Figure 11.

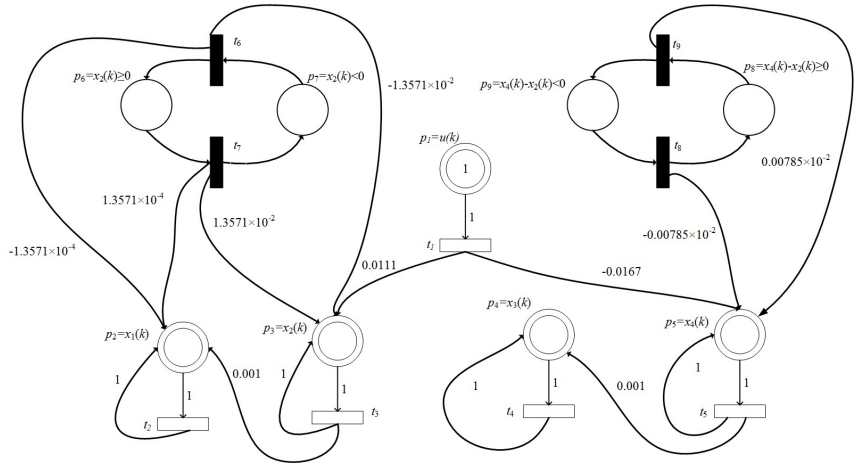

Figure 11: The HTDPN model of the capsubot.

In Figure 11, place p_1 indicates input variable and p_2, p_3, p_4 and p_5 show states of the capsubot. The incidence matrix for the capsule robot can be written as follows:

$$W = \begin{bmatrix} W_C & W_{CD} \\ 0 & W_D \end{bmatrix} =$$

$$\begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1.3571 \times 10^{-4} & 1.3571 \times 10^{-4} & 0 & 0 \\ 0.0111 & 0.01 & 0 & 0 & 0 & -1.3571 \times 10^{-4} & 1.3571 \times 10^{-4} & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -0.0167 & 0 & 0 & 0.01 & 0 & 0 & 0 & -0.0785 \times 10^{-2} & 0.0785 \times 10^{-2} \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix}$$

4.6 A brief introduction to genetic algorithm

Genetic Algorithm (GA) is a stochastic algorithm based on principle of natural section and genetics. A GA is one such direct search optimization technique. GA was proposed and developed in the 1960s by John Holland, his students, and his colleagues at the University of Michigan. GA consists of four steps, in the first step, GA starts with an initial population containing a number of chromosomes. In the second step, in order to obtain the optimal solution, the population of the GA must be evaluated. This evaluation consists of calculation the fitness value of each individual by minimizing an objective function. In the third step some individuals are selected based on their fitness value and sent to the crossover operator to produce two new individuals. This operation is then until the size of the population of new individuals reaches the main population size. In the last step, the mutation operator changes some gens of the new individuals. Figure 12 shows flowchart of GA for PID controller [25].

Using Figure 9, Figure 11 and Figure 12, it is easy to show that the closed-loop system can be shown as in Figure 13.

5 Simulation Results

In this section, to prove the validity of the proposed method, a closed-loop system described by HTDPN is simulated. In this approach, the genetic algorithm is used for gains tuning. The legless piezo capsubot microrobot with a nonlinear complex dynamic system is selected as the system that is controlled by a PID controller. The performance of the HTDPN tool and the employed control approach is evaluated and compared with the predictive control method [24]. Gains of the controller are carried out by HTDPN based on GA as shown in Figure 14.

Since the robot has a nonlinear dynamic system the gains of the PID controller could be tuned in any time. The GA convergence criterion is as (22):

$$|e(k)| \leq 0.01 \tag{22}$$

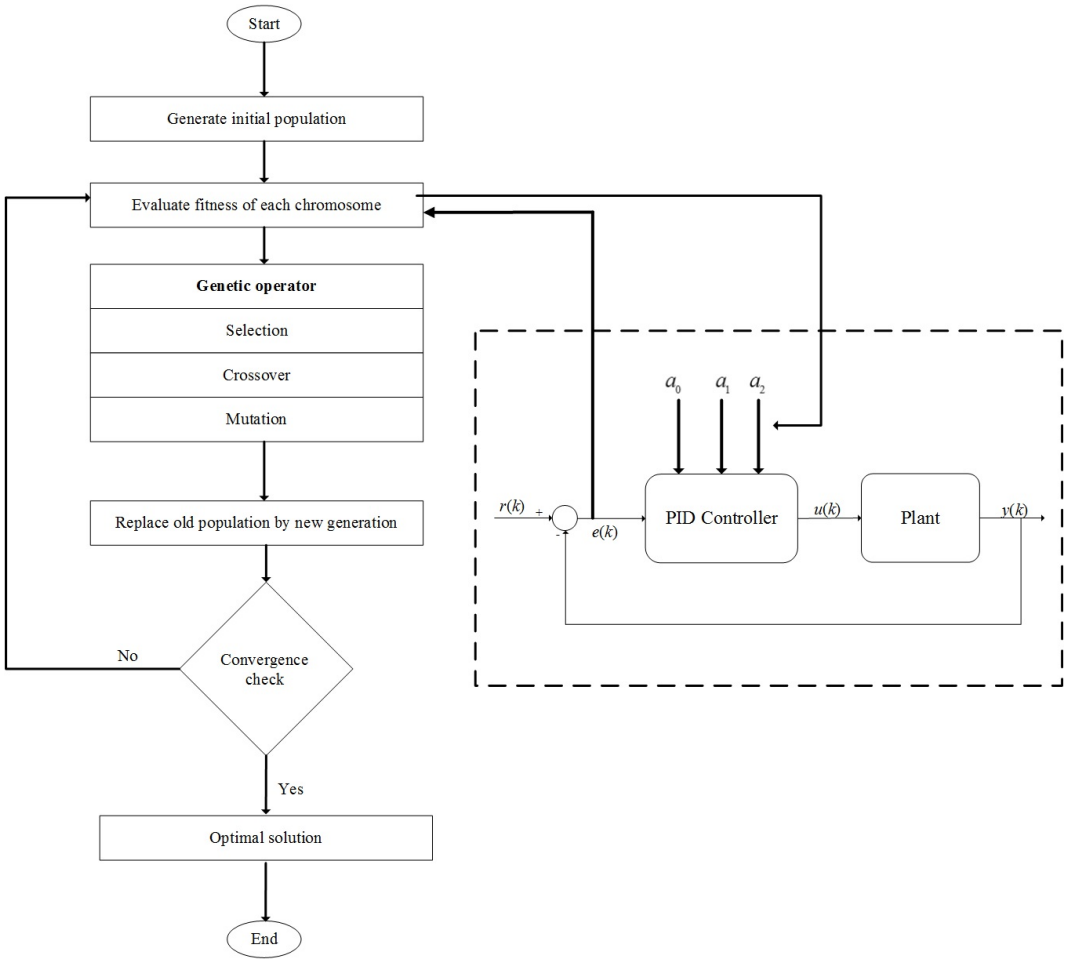

Figure 12: Flowchart of genetic algorithm for PID controller.

To compare the proposed approach and predictive control, the same displacement must be considered for them. Figure 15 illustrates the tracking of the reference signal for the two methods.

Figure 15 shows that the capsubot in the proposed approach is faster and tracks the reference signal with less oscillation. Figure 16 and Figure 17 depict the robot velocity and inner mass velocity, respectively.

Figure 16 and Figure 17 depict that the velocity trajectory of the proposed method has less oscillation and is smoother than the predictive control method; therefore, the result can be easily implemented. The output error in the two approaches is shown in Figure 18.

Figure 18 reveals that using the HTDPN tool, the error is rapidly decreased to an acceptable range. The performance of a tracking system could be measured by the summation of absolute errors. The summation of absolute errors can be defined as follows:

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

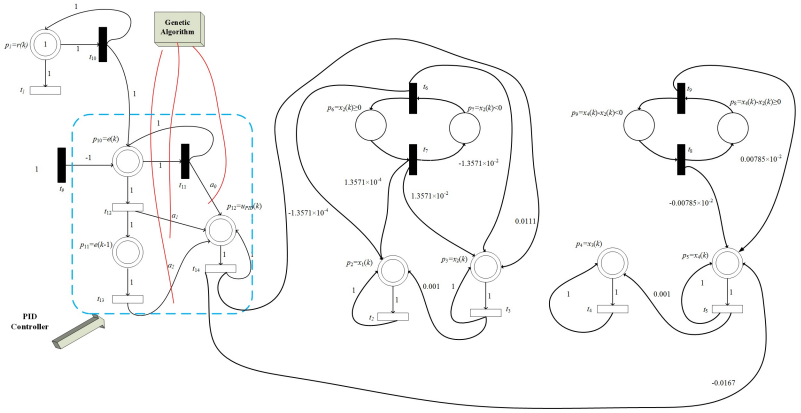

Figure 13: Plant and PID controller integrated together.

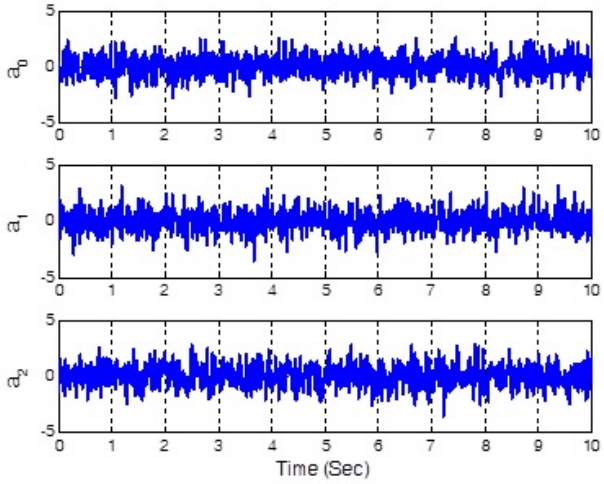

Figure 14: The gains of the PID controller based on the GA.

The summation of absolute errors in the HTDPN model and predictive control are 1.308 and 1.1239, respectively.

Figure 19 depicts the control input signal for the proposed method and predictive control.

The force in the proposed approach is smoother than predictive control. In addition, the absolute peak in the predictive control scheme is more than the latter one. The main criteria to compare controllers are energy consumption which can be considered as below:

$$W = \sum_{i=1}^{n} U(i)x(i) \tag{24}$$

One can see that energy consumption for the same condition in the HTDPN approach and predictive control are $0.2251j$ and $0.4890j$, respectively. These results lead to a cheaper

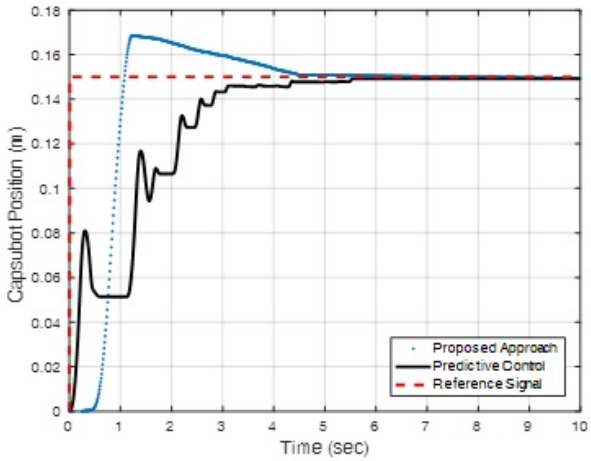

Figure 15: Trajectory of the capsul robot based on the proposed approach and predictive control.

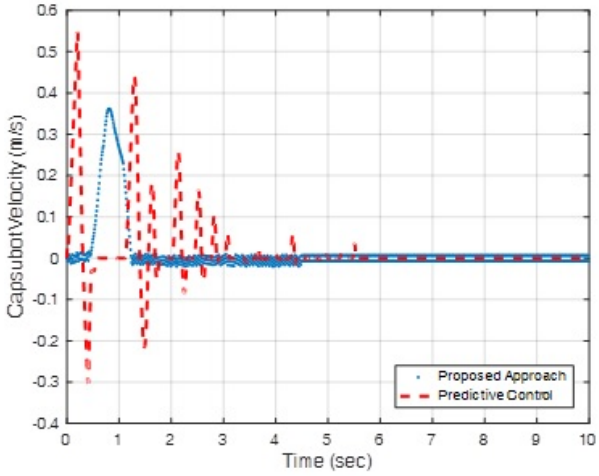

Figure 16: Capsubot velocity using proposed approach and predictive control.

piezoelectric element selection. Table 2 compares the performance of the HTDPN with four existing control approaches of the capsubot.

Table 2: Parameters of the capsubot robot.

Parameters	Proposed Approach	Opt. Control	OLC	CLC	Pre. Control
Absolut Error	1.308	7.0715	2.65	2.598	1.1239
Energy (j)	0.2251	0.1363	0.3968	0.3932	0.4890
Max Force (N)	2.7893	1.5188	4.35	4.4158	4

As it is observed in Table 2, the proposed method is compared to four other methods of open loop control (OLC), close loop control (CLC), optimal control (Opt. Control) and predictive

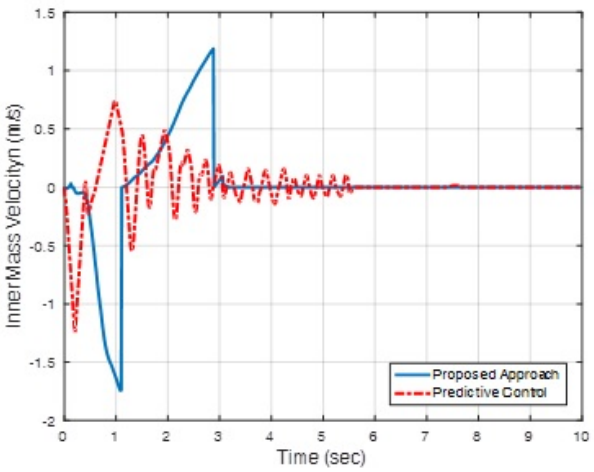

Figure 17: Inner mass velocity using proposed approach and predictive control.

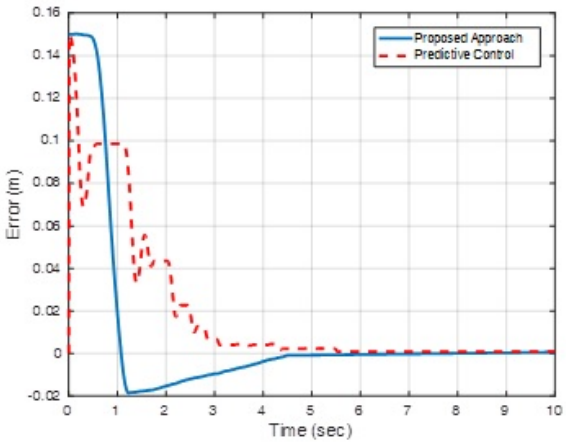

Figure 18: Error of two approaches.

control (Pre. Control) On the other hand, in comparison to the optimal control method, in some cases, such as tracking error and the time it takes to track the reference signal, performance is more efficient. The simulation time for the proposed method is 0.003 (Sec) compared to 0.0099 (Sec) of the conventional one, which indicates a considerable amount of decrease in the run time, i.e. 156.87%. To introduce the simplicity of the presented method of the HTDPN, a series of simulation tests, i.e. comparison between HTDPN and the state space model [26] are carried out, the results of which are depicted in Table 3.

As it is shown above, the issue of time efficiency caused by reduced computational effort using an incident matrix is obvious. In addition, the simplicity of the system analysis is another factor of attraction in utilizing this method.

Extending automated PID tuning methods and simplifying controller design for multi-mode systems could be evaluated as the two advantages of the method. It should be noted that the

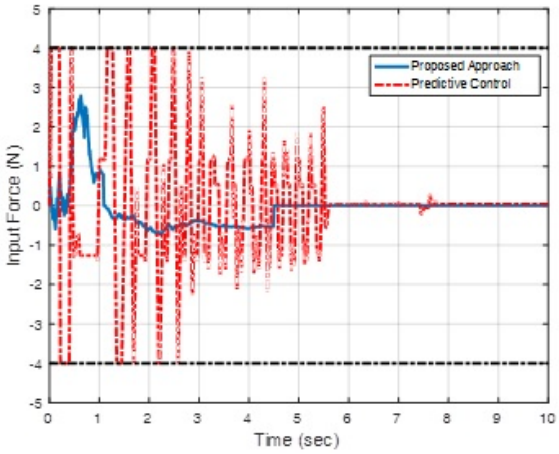

Figure 19: Control input of the HTDPN and predictive control.

Table 3: Parameters of the capsubot robot.

System	Transfer function	HTDPN	Conventional
Two link tennis player robot	$G(s) = \frac{1}{s(s+1)}$	$0.0038(Sec)$	$0.009(Sec)$
Robot fruit picker	$G(s) = \frac{1}{(s+1)^2}$	$0.004(Sec)$	$0.0086(Sec)$
Magnetic disc Driver	$G(s) = \frac{1}{s(0.001s+1)}$	$0.0038(Sec)$	$0.0093(Sec)$
Gap dynamics	$G(s) = \frac{1}{s(s+10)}$	$0.0043(Sec)$	$0.0097(Sec)$

GA technique used in this approach is only for data collection to obtain the auto-tuning gains controller function, i.e. auto tuning based on HTDPN.

6 Conclusion

The present paper theorizes the HTDPN to model all variants of continuous dynamic systems such as systems with current sample time signals, systems including various subsystems, multi-mode systems or systems with variable parameters. The HTDPN description of the legless piezo capsubot and *PID* controller is simulated and yields results which verify that HTDPN description are much more efficient in terms of analysis simplicity. The performance of our proposed controller is compared with four control approaches. Results show that the presented method is promising as compared to other approaches that are used in the literature.

References

[1] Wu N. Q., Zhou M. C., Li Z. W. (2015) "Short-Term scheduling of crude-oil operations: Petri Net-based control-theoretic approach", IEEE Robotics and Automation Magazine, 22, 64-76.

[2] Hazza M. H. F. A., Taha A. H., Adesta E. Y. T., Albakri A. (2015) "Modelling and analysing deadlock in flexible manufacturing system using untimed Petri Net", 4th International Conference on Advanced Computer Science Applications and Technologies (ACSAT), Kuala Lumpur, 261-265.

[3] Lennartson B., Basile F., Miremadi S., Fei Z., Hosseini M. N. (2014) "Supervisory control for state-vector transition models-A unified approach", IEEE Transaction on Automation Science and Engineering, 11, 33–47.

[4] Ahanarani Farahani A., Dideban A., Najafgholi E. (2016) "Modeling continuous systems by Petri Nets using speed control arcs", The 4th International Conference on Control, Instrumentation, and Automation, Iran, 75-80.

[5] Taleb M., Leclercq E., Lefebvre D. (2014) "Limitation of flow variation of timed continuous Petri Nets via model predictive control", American Control Conference (ACC) Portland Oregon USA, 4919-4924.

[6] Demongodin I., Giua A. (2014) "Dynamics and steady state analysis of controlled generalized batches Petri Nets", Nonlinear Analysis: Hybrid Systems, 12, 33–49.

[7] Milinkovic S., Markovic M., Veskovic S., Ivic M., Pavlovic N. (2013) "A fuzzy Petri Net model to estimate train delays", Simulation Modelling Practice and Theory, 33, 144–157.

[8] Wang Y., Ding D. (2015) "Revealing the structure and function of P. pastoris metabolic network using Petri Nets", 14th International Symposium on Distributed Computing and Applications for Business Engineering and Science, 360-363.

[9] Chen C., Yang Y., Zhang X. (2017) "Application study on knowledge representation of emergency scenario using hybrid Petri Net", 3rd International Conference on Information Management, 332-336.

[10] Ahangarani Farahani A., Dideban A. (2015) "Output feedback controller design and modeling of dynamics systems using Petri Nets", 3rd Nation and First International Conference in applied research on Electrical, Mechanical and Mechatronics Engineering Iran, 1-6.

[11] Ahangarani Farahani A., Dideban A. (2016) "Continuous-time delay-Petri Nets as a new tool to design state space controller", Information Technology and Control, 45, 401-412.

[12] Dideban A., Ahangarani Farahani A., Razavi M. (2015) "Modeling continuous systems using modified Petri Nets model", Modeling and Simulation in Electrical and Electronics Engineering (MSEEE), 1, 75-79.

[13] Dideban A., Sabouri Rad M. (2014) "Electrical circuit modelling with Petri Nets by using of control arcs", Journal of Modeling in Engineering, 11, 39-47. .

[14] Pettersson S., Lennarston B. (1995) “Hybrid modelling focused on hybrid Petri Nets”, 2nd European Workshop on Real-Time and Hybrid Systems, 303-309.

[15] Bibi Z., Ahmad J., Paracha R. Z., Siddiqa A. (2017) “Modeling and analysis of the signaling crosstalk of PI3K, AMPK and MAPK with timed hybrid Petri Nets approach”, 17th International Conference on Computational Science & Its Applications (ICCSA), 1-7.

[16] Lu X., Zhou M. C., Ammari A. C., Ji J. (2016) “Hybrid Petri Nets for modeling and analysis of microgrid systems”, IEEE/CAA Journal of Automatica Sinica, 3, 349-356.

[17] Nivznansk’a M., Kuvcera E., Haffner O., Kozákov’a A. (2016) “Modeling of hybrid systems by hybrid Petri Nets using open modelica and PNlib”, Proceedings of the 28th International Conference Cybernetics and Informatics Slovakia, 1-6.

[18] Fraca E., Júlvez J., Silva M. (2016) “Hybrid and hybrid adaptive Petri Nets: on the computation of a reachability graph”, Nonlinear Analysis: Hybrid Systems, 16, 24–39.

[19] Ogata K. H., (1987) ”Discrete-time control systems”, Prentice Hall, 2nd Edition.

[20] Ibrahim D. (2006) “Microcontroller based applied digital control”, John Wiley & Sons Inc., English.

[21] Mahmoudzadeh S., Mojallali H. (2013) “An optimized PID for legless capsubots using modified imperialist competitive algorithm”, First RSI/ISM International Conference on Robotics and Mechatronics (ICRoM), Iran, 194-199.

[22] Farahani A. A., Suratgar A. A., Talebi H. A. (2012) “Optimal controller design of Legless piezo capsubot movement”, International Journal of Advanced Robotic Systems, 9, 1-7.

[23] Liu Y., Yu H., Yang T. C. (2008) “Analysis and control of a capsubot”, Proceedings of the Word Congress Korea, 756-761.

[24] Behmardi Kalantari S., Farahani A. A., Doustmohammadi A., Menhaj M. B., Suratgar A. A., Talebi H. A. (2011) “Hybrid model predictive control of legless piezo capsubot”, 2nd International Conference on Control, Instrumentation and Automation (ICCIA), Iran, 941-945.

[25] Pareek S., Kishani M., Gupta R. (2014) “Optimal tuning of PID controller using genetic algorithm and swarm techniques”, International Journal of Electronic and Electrical Engineering, 7, 189-194.

[26] Dorf R. C., Bishop R. H. (2011) “Modern control system” Prentice Hall, 9th Edition.

شبکه پتری زمان تأخیری هیبریدی به عنوان یک ابزار ریاضی جدید برای مدل‌سازی سیستم‌های دینامیکی با سیگنال زمان فعلی

آهنگرانی فراهانی، ع.
دانشجوی دکتری، گروه برق و کامپیوتر،
ایران، سمنان، دانشکده مهندسی برق و کامپیوتر
a.ahangarani@semnan.ac.ir

دیدبان، ع.
دانشیار، گروه برق و کامپیوتر- نویسنده مسئول،
ایران، سمنان، دانشکده مهندسی برق و کامپیوتر
adideban@semnan.ac.ir

تاریخ دریافت: ۱۲ دی ۱۳۹۷ تاریخ پذیرش: ۱ مهر ۱۳۹۸

چکیده

ابزار شبکه پتری یکی از روش‌هایی است که به خوبی جهت مدل‌سازی و آنالیز سیستم‌های دینامیکی استفاده می‌شود. هر چند، این ابزارهای رایج قادر به مدل‌سازی همه سیستم‌های دینامیکی همچون سیستم‌های با سیگنال زمان فعلی، سیستم‌های دارای چند مود و سیستم‌های دارای چند زیر سیستم نمی‌باشند. این مقاله ابزار ریاضی جدیدی به نام شبکه پتری زمان تأخیری هیبریدی برای رفع این مساله را ارائه می‌کند. در این مقاله برای حل این مشکل، از اصول شبکه‌های پتری پیوسته و گسسته و ترکیب آن‌ها بهره برده می‌شود، به‌طوریکه از بخش پتری پیوسته برای مدل‌سازی زمان‌های گذشته و بخش پتری پیوسته جهت مدل‌سازی استفاده می‌شود. برای ارزیابی ابزار معرفی شده، از آن جهت مدل‌سازی یک ربات کپسولی که از دسته سیستم‌های چند مودی است و یک کنترلر PID همراه آن که بهره‌های کنترلی آن با روش ژنتیک الگوریتم تنظیم می‌شود، استفاده می‌شود. نتایج شبیه‌سازی نشان می‌دهد که روش ارائه شده به واسطه روابط ریاضی حاکم بر آن، دارای محاسبات سریع‌تری نسبت به روش‌های فضای حالت و متداول است. همچنین به‌راحتی قابل مشاهده است که طراحی و آنالیز کنترل‌کننده PID در این روش ساده‌تر شده‌است بعلاوه، یک مقایسه‌ای بین روش‌های استفاده شده برای ربات کپسولی صورت پذیرفته‌است. نتایج شبیه‌سازی نشان می‌دهد که روش ارائه شده در مقایسه با روش کنترل پیش‌بین نتایج بهتری دارند.

کلمات کلیدی

شبکه پتری هیبریدی، سیگنال‌های زمان فعلی، ربات کپسولی، الگوریتم ژنتیک.