

Fault Diagnosis of Discrete-Time Linear Systems Using Continuous-Time Delay Petri Nets

Sobhi Baniardalani[†]

Department of Electrical Engineering, Faculty of Energy, Kermanshah University of Technology, Kermanshah, Iran

A | This paper deals with the fault diagnosis of a linear continuous variable dynamical system represented by a discrete
B | state-space model. The proposed fault diagnoser was based on a special Petri net (PN) called continuous-time delay Petri
S | net (CTDPN). According to the theorem presented in this paper, an exact correspondence between discrete state-space
T | equations and fundamental equations of the CTDPN was established. Based on this theorem, a systematic method was
R | proposed for the realization of classical parity equations by a CTDPN that played the role of the fault diagnoser. By
A | integrating the concept of state-space models and PNs in this paper, new and effective methods can be proposed for the
C | analysis and fault diagnosis of hybrid systems. Finally, the performance of the proposed method was investigated for the
fault diagnosis of a DC motor. The results demonstrated that, with the help of the proposed PN, fault diagnosis can be
performed well and the traditional diagnoser can be replaced with this novel network.

Article Info

Keywords:

Continuous-Time Delay Petri Net, Fault Diagnosis, Parity Equations, State-Space Model

Article History:

Received 2019-01-11

Accepted 2019-09-29

I. INTRODUCTION

Fault diagnosis and condition monitoring methods in dynamical systems have increasingly been developed in recent years, with numerous methods proposed for a wide range of applications [1, 2]. Some of these methods utilize Petri nets (PNs) [3]. PNs are powerful modeling tools with intuitive graphical representation for the simulation of systems. They also provide a formalism for setting up state equations and algebraic equations governing the behavior of the system [4]. Considering the PN properties, the present paper focused on the fault diagnosis of continuous variable dynamical systems (CVDSs) using a special PN known as the continuous-time delay Petri net (CTDPN). Most studies cited here related to fault diagnosis by PN focused on discrete event systems (DESSs) [5-8]. From a formal perspective, a DES can be thought of as a dynamic system whose

state-space is a discrete set and the state transition mechanism is event-driven [9].

However, in the real world, we are faced with CVDSs represented by differential equations or difference equations. These systems cannot be exactly modeled and analyzed by conventional classic PNs. To cope with this limitation, alternative classes of PNs such as continuous, timed continuous, and hybrid PNs have been introduced [4, 10]. In [11, 12] a special type of this class, CTDPN, has been introduced for the modeling and controller design of continuous systems. CTDPN is a continuous PN in which the marking of places can be non-zero real numbers and transitions can be fired if each of their input places has a non-zero marking. In addition, the speed of transitions is infinite. In [12] this PN has been employed to implement the discrete state-space of a dynamic system, and in [11] an approach has been presented to design a state controller based on this PN for a continuous dynamic system.

According to the outstanding features of CTDPN for the simulation of a discrete-time state-space model of a dynamic system, it can be an effective tool for the fault diagnosis of

[†]Corresponding Author :S.baniardalani@kut.ac.ir
Tel: +98-83-38305000,Fax: +98-83-38305006, Kermanshah University of Technology

discrete-time CVDSs as shown in the present paper. In this paper, a novel approach is presented for the fault diagnosis of a CVDS represented by a discrete-time state-space model. In fact, by the proposed method of this paper, a conventional diagnoser is designed and is then realized by a CTDPN. From this point of view, the proposed method can be regarded as the generalization and development of the algebraic fault diagnosis approach presented for DESs [13-15]. Due to the compatibility of CTDPN with the continuous systems, we can utilize the well-known fault diagnosis methods to design the CTDPN fault diagnoser. In this paper, the discrete-time parity approach has been adopted to generate structured residuals indicating the occurred faults [16]. These residuals are represented by marking of places in the realized CTDPN.

In the past decade, the application of powerful programmable logic controllers (PLCs) has become widespread in various industries [17, 18]. As a result, the idea of implementing an online diagnoser and a feedback controller, both on the same PLC, has received further attention. Some advantages of this implementation scheme are the reduction of hardware needed for diagnosis, no need for the sensor installation to observe command events, no need for communication buses, and a higher fault diagnosis speed [19-21].

A challenging issue in this regard is that the on-line diagnoser system should be easily implemented by a standard PLC programming language. As reported by the few works addressing the problem of implementing online diagnosis on a PLC, PN-based diagnosis has superior compatibility with PLC programming languages such as the ladder diagram (LD) or sequential function chart (SFC) [19-22]. All the studies cited before were on DESs and, to the best of the author's knowledge, no research has dealt with the problem of implementing online PN-based diagnosers for CVDSs on a PLC. Thus, the main motivations for the present work are as follows:

- The lack of a general framework for implementing the well-known fault diagnosis methods for CVDS on PLCs (This paper proposes an appropriate PN-based diagnosis for these systems as the first and most important step.)
- Facilitating access to unified formalism based on PNs for fault diagnosis in hybrid systems exhibiting both continuous and discrete dynamic behaviors.

Related literature

The use of PNs for fault diagnosis systems has been reviewed in a number of papers along with other models such as automata [5-8, 23-26]. In [5], three main methods, including the algebraic method for fault diagnosis, have been investigated. In [13-15], faults in both PN transitions and places have been considered, and a redundant PN has been constructed and attached to the main PN. In the diagnosis scheme in [13], the nonzero marking of redundant places has

been decoded to indicate the occurred faults. Similar to [13], in this paper, we constructed a redundant CTDPN in which the markings of places were equal to residuals indicating the occurred faults. In [27], the faults have been modeled by unobservable transitions and an algorithm has been presented for the separation of the faulty part. In this method, there is no need for a redundant network for fault diagnosis and this is the main advantage of this method over our paper. However, this method is applicable only for pure DESs. In [7, 8] a diagnoser based on the reachability graph has been proposed for bounded PNs (a PN with limited marking value) and analyzed. Since the reachability graph cannot be accessed for continuous PNs, this method is not applicable for CVDSs.

As mentioned before, alternative PNs such as continuous Petri nets (CPNs), timed continuous Petri nets (TCPN), hybrid Petri nets (HPN), and CTDPNs have been introduced to model and analyze continuous dynamic systems [4, 11, 28]. In [28] a survey of using HPNs for logistic purposes has been presented, and in [29] CPNs have been applied for modeling and analyzing biological systems. The major drawback of PNs introduced in [28-30] is that their markings are positive real numbers, and thus these PNs cannot be utilized for modeling CVDSs whose state variables have negative values, while this limitation does not exist in the network employed in the present study.

A few studies have addressed the fault diagnosis in TCPN [31-33]. In [31], a fault modeling framework based on hybrid automata has been presented and an online monitoring approach based on a timed PN model that abstracts away the continuous dynamics has also been described. Contrary to [31], the continuous dynamics was directly simulated by CTDPN in the present paper. Moreover, [32] has implemented an adaptive fault diagnoser for a system modeled by a TCPN under infinite server semantics. This work has proposed a single diagnoser model whose structure was known and some of its parameters were updated depending on the fault occurrence. In this model, identification algorithms, based on heuristic optimization methods, have been used to identify unknown fault parameters.

In comparison with the present study, the method proposed in [32] is more complex and cannot be implemented online by PLCs. In [33], the fault isolation problem has been investigated for the systems modeled by TCPNs under infinite server semantics. In order to isolate faults, a diagnoser based on a set of robust reduced-order finite-time observers has been proposed in this work.

In comparison with the cited works, the main contributions of the present paper are:

- Introducing a novel approach for the fault diagnosis of a CVDS by a CTDPN
- Presenting a direct path to implement conventional quantitative diagnosers by PNs
- Developing the previous algebraic methods for fault diagnosis in DESs by PNs to CVDSs by CTDPNs

The remainder of this paper is organized as follows: In Section 2, we present preliminaries and basic definitions. Section 3 illustrates how to convert a discrete state-space model into CTDPN. Section 4 describes the design procedure of diagnosis systems, and Section 5 presents the result of using the proposed diagnoser for a DC motor.

II. PRELIMINARIES

In this section, some basic definitions required in other sections are reviewed.

A. Discrete PN

Definition 1 [34]: A PN is a structure $N = (P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{M}_0)$ where $P = \{P_1, P_2, \dots, P_m\}$ is the set of m places, $T = \{t_1, t_2, \dots, t_n\}$ is the set of n transitions, $\mathbf{Pre}: P \times T \rightarrow N$ is the pre-incidence function specifying the number of arcs directed from places to transitions (called “pre” arcs) and represented by an $m \times n$ matrix, $\mathbf{Post}: P \times T \rightarrow N$ is the post-incidence function specifying the number of arcs directed from transitions to places (called “post” arcs) and represented by an $m \times n$ matrix, and \mathbf{M}_0 is the initial marking. The incidence matrix \mathbf{W} is equal to $\mathbf{W} = \mathbf{Post} - \mathbf{Pre}$. For a transition $t \in T$, its sets of input and output places are respectively defined as:

$\bullet t = \{p \in P \mid \text{Pre}(p, t) > 0\}$, and $t \bullet = \{p \in P \mid \text{Post}(p, t) > 0\}$, while given a place $p \in P$, its set of input and output transitions are defined as:

$\bullet p = \{t \in T \mid \text{Post}(p, t) > 0\}$, $p \bullet = \{t \in T \mid \text{Pre}(p, t) > 0\}$.

Definition 2 [34]: A marking is a function $\mathbf{M}: P \rightarrow N$ that assigns to each place a non-negative integer number of tokens. The marking of a PN defines its state.

Definition 3 [34]: A transition t_i is enabled at a marking \mathbf{M} , if $\mathbf{M} \geq \mathbf{Pre}(\cdot, t_i)$, i.e., if each place $p \in P$ contains a number of tokens greater than or equal to $\text{Pre}(p, t_i)$.

Definition 4 [34]: A transition t_i enabled at a marking \mathbf{M} can fire. The firing of t_i removes $\text{Pre}(p, t_i)$ tokens from each place $p \in P$ and adds $\text{Post}(p, t_i)$ tokens in each place $p \in P$, yielding a new marking:

$$\mathbf{M}' = \mathbf{M} - \mathbf{Pre}(\cdot, t_i) + \mathbf{Post}(\cdot, t_i) = \mathbf{M} + \mathbf{W}(\cdot, t_i) \quad (1)$$

To denote that the firing of t_i from \mathbf{M} leads to \mathbf{M}' , we write $\mathbf{M} [t_i > \mathbf{M}']$. Let S be a firing sequence which can be performed from a marking \mathbf{M}_0 , and suppose that \mathbf{M}_k is reachable from \mathbf{M}_0 by applying S [4, 11]. The characteristic vector of sequence S , written as \mathbf{s} , is the m -component vector whose component number i corresponds to the number of firings of transition t_i in sequence S , i.e.

$$\mathbf{s} = \mathbf{s}_1 + \mathbf{s}_2 + \dots + \mathbf{s}_{k-1}, \quad (2)$$

where \mathbf{s}_j represents the firing transition of t_j . In this case,

\mathbf{M}_k is obtained by fundamental equation:

$$\mathbf{M}_k = \mathbf{M}_0 + \mathbf{W}\mathbf{s}. \quad (3)$$

B. Continuous PN (CPN)

Definition 5 [4]: A marked autonomous continuous PN is a five-tuple $R = (P, T, \mathbf{Pre}, \mathbf{Post}, \mathbf{M}_0)$ such that P and T are the same as Definition 1, $\mathbf{Pre}: P \times T \rightarrow R_+$ and $\mathbf{Post}: P \times T \rightarrow R_+$ are the input and output incidence applications, respectively, and $\mathbf{M}_0: P \rightarrow R_+$ is the initial marking. $\mathbf{Pre}(P_i, t_j)$ denotes the weight of the arc $P_i \rightarrow t_j$ and is a positive real number if the arc exists and 0 otherwise. Similarly, $\mathbf{Post}(t_j, P_i)$ is the weight of the arc $t_j \rightarrow P_i$. In addition, a place marking must be a real number since it may change continuously. In a continuous PN, places and transitions are represented by a double line.

Definition 6 [4]: In a continuous PN, the enabling degree of transition t_j for marking \mathbf{M} , denoted by q or $q(t_j, \mathbf{M})$, is the real number q such that

$$q = \min_{i: P_i \in \bullet t_j} \left(\frac{\mathbf{M}(P_i)}{\mathbf{Pre}(P_i, t_j)} \right). \quad (4)$$

If $q > 0$, transition t_j is enabled, and it is said to be q -enabled. Definition 6 is applied to a generalized PN. In the particular case of a PN in which the weight is 1 for all input arcs to the transitions, Equ. (5) can be simplified as:

$$q = \min_{i: P_i \in \bullet t_j} (\mathbf{M}(P_i)). \quad (5)$$

Definition 7 [4]: A timed continuous PN (TCPN) is a pair (R, Spe) such that R is a marked autonomous continuous PN (c. f. Definition 5), and Spe indicates a function from the set T of transitions to $R_+ \cup \{\infty\}$. For each t_j , $\text{Spe}(t_j) = V_j$ is the maximal speed associated with transition t_j . The instantaneous firing speed $v_j(t_j)$ satisfies the following condition [4]:

$$v_j(t_j) \leq V_j(t_j)$$

For a timed continuous PN, between times τ and $\tau + d\tau$, the quantity of firing of t_j is $v_j(\tau) \cdot d\tau$; then, \mathbf{s} corresponds to the vector $\mathbf{v}(\tau) \cdot d\tau$. It follows that:

$$\mathbf{M}(\tau + d\tau) = \mathbf{M}(\tau) + \mathbf{W} \cdot \mathbf{v}(\tau), \quad (6)$$

then

$$\mathbf{M}(\tau_2) = \mathbf{M}(\tau_1) + \mathbf{W} \int_{\tau_1}^{\tau_2} \mathbf{v}(\tau) d\tau, \quad (7)$$

is the fundamental equation for a timed continuous PN. It works for any $0 \leq \tau_1 \leq \tau_2$ [4].

C. Continuous-Time Delay Petri Net

Definition 8 [11]: A continuous-time delay Petri net is a timed continuous PN in which each transition firing plays the role of a unit time delay. In fact, in this TCPN, when a transition is enabled, it is fired after a time delay (e.g. T_s). Moreover, in this PN, the following assumptions are held [11]:

Assumption 1. Place tokens and weights of the arcs in CTDPN can be negative or non-negative real numbers at any time.

Assumption 2. A transition is enabled if $\mathbf{M}(p_i) > 0$ or $\mathbf{M}(p_i) < 0$.

Assumption 3. The speed of the transitions is infinity.

Assumption 4. When transitions are fired, values of tokens of input places become zero.

III. DYNAMICS OF THE CTDPN

This paper is based on the application of CTDPN for realizing a well-known fault diagnosis system (represented by a set of difference equations). Therefore, in the first step, the CTDPN dynamics should be determined and then the relation between this dynamic and the difference equations must be established. To do this, the following theorem illustrates how the marking vector of a CTDPN changes with time and the enabling degree of the transitions.

Theorem 1: For a given CTDPN, if $\mathbf{M}(T_s)$ is the marking vector at time kT_s , then $\mathbf{M}((k+1)T_s) = \mathbf{M}(kT_s) + \mathbf{W}\mathbf{q}(kT_s)$, in which $\mathbf{M}((k+1)T_s)$ indicates the marking vector at the time $(k+1)T_s$, \mathbf{W} denotes the incidence matrix, and $\mathbf{q}(kT_s) = [q_1(kT_s) \ q_2(kT_s) \ \dots \ q_m(kT_s)]^T$ represents the enabling degree vector of all transitions at time kT_s .

Proof: As previously mentioned (c.f. Definition 7), $v_j(\tau)$ denotes the mark flow rate of t_j . Therefore, the mark transported by t_j (firing quantity of t_j), between times τ and $\tau + \Delta\tau$, is

$$\Delta m_j = v_j(\tau) \cdot \Delta\tau. \quad (8)$$

Since there is no limit to the firing quantity of t_j , Δm_j is only limited to the enabling degree of t_j ; it means that $\Delta m_j = q_j(\tau)$. Here, $q_j(\tau)$ is the enabling degree of t_j and is defined by Equ. (4). As a result, Δm_j is finite and, because of the infinity speed of transitions, (c.f. Assumption 3 in section 2.3), it results from Equ. (8) that $\Delta\tau \rightarrow 0$. Thus Equ. (8) can be written as:

$$dm_j = v_j(\tau) \cdot d\tau. \quad (9)$$

It means that when t_j is fired at time instance T_s , the entire marking is transported instantaneously at this time such that:

$$\Delta m_j = \int_{T_s^-}^{T_s^+} dm_j = q_j(T_s^-) = \int_{T_s^-}^{T_s^+} v_j(\tau) \cdot d\tau. \quad (10)$$

Here, $T_s^- = T_s - \varepsilon$ and $T_s^+ = T_s + \varepsilon$ where $\varepsilon \rightarrow 0$.

For a CTDPN with initial enabling degree vector $\mathbf{q}(0) = [q_1(0) \ q_2(0) \ \dots \ q_m(0)]^T$, all transitions can be fired only at kT_s instances, where $k=1,2,\dots$. Thus Equ. (10) can be rewritten for all transitions as follows:

$$\mathbf{q}(kT_s^-) = \int_{kT_s^-}^{kT_s^+} \mathbf{v}(\tau) \cdot d\tau, \text{ for } k=1,2,\dots \quad (11)$$

Here, $\mathbf{v}(\tau)$ denotes the transition speed vector. Now, if Equ. (7) is written for $t_2 = kT_s^+$ and $t_1 = kT_s^-$, and also using Equ. (11), then

$$\begin{aligned} \mathbf{M}(kT_s^+) &= \mathbf{M}(kT_s^-) + \mathbf{W} \int_{kT_s^-}^{kT_s^+} \mathbf{v}(\tau) \cdot d\tau, \\ \mathbf{M}(kT_s^+) &= \mathbf{M}(kT_s^-) + \mathbf{W}\mathbf{q}(kT_s^-). \end{aligned} \quad (12)$$

By replacing $kT_s^+ = kT_s$ and $kT_s^- = (k-1)T_s$ for $k=1,2,\dots$, we have for $k=1,2,\dots$

$$\mathbf{M}(kT_s) = \mathbf{M}((k-1)T_s) + \mathbf{W}\mathbf{q}((k-1)T_s), \quad (13)$$

Finally, for $k=0,1,\dots$, (13) can be written as

$$\mathbf{M}((k+1)T_s) = \mathbf{M}(kT_s) + \mathbf{W}\mathbf{q}(kT_s) \quad (14)$$

□

For simplicity, kT_s is abbreviated to k henceforth. In addition, to avoid conflict, it is assumed that for each place, its output transitions (e.g. t_j) have more priority over its input transitions. For further illustration, consider a place that is an input place for t_j and an output place for t_k . At time instance T_s , both transitions are fired simultaneously. According to this assumption, the marking of this place is transported via t_j at first, then it accepts the transported marking from t_k . In the following, a corollary of the presented lemma is given.

Corollary 1: Consider a particular CTDPN in which the weights are 1 for all input arcs to the transitions (e.g. $Pre(P_i, t_j) = 1$ for all P_i and t_j), and each transition has one and only one input place (e.g. $g \bullet t_j = P_j$ for all t_j). From Equ. (5), it is concluded that:

$q(t_j) = \min_{i: P_i \in \bullet t_j} (m(P_i)) = m(P_j)$. Thus Equ. (14) in the theorem is simplified as:

$$\mathbf{M}(k+1) = \mathbf{M}(k) + \mathbf{W}\mathbf{M}(k). \quad (15)$$

IV. FAULT DIAGNOSER DESIGN

In the previous sections, CTDPN was introduced and its dynamics were obtained according to Equ.s (14) and (15). In this section, a fault diagnoser is constructed by the particular CTDPN illustrated by Corollary 1 and its dynamic Equ. (15). In the first step, the diagnoser was designed. Here, we used the well-known method of parity equations (briefly reviewed in what follows) to design a fault diagnoser for the continuous system. The results of this stage were the set of parity equations determining residual vector $\mathbf{r}(k)$ (c.f. Equ. (19)). In the second step, a CTDPN was constructed to realize the parity equations and obtaining the residual vector. In this step, the state-space equations and parity equations were combined to obtain dynamic equations of the CTDPN (c.f. Equ.s (24) and (25)).

A. Fault diagnoser design with parity equations [35]:

A straightforward method for detecting process faults is to compare the process behavior with a process model describing the nominal, i.e. non-faulty behavior. The difference of signals between the process and the model is expressed by residuals. The residuals can be made in state-space formulation and are used to identify the faults occurring in the process. For a given linear system with p inputs, r outputs, and n state variables, the discrete-time state-space model is described by:

$$\mathbf{x}(k+1) = \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k) + \mathbf{L}\mathbf{f}(k) \quad (16)$$

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k) + \mathbf{M}\mathbf{f}(k), \quad (17)$$

in which $\mathbf{A}, \mathbf{B}, \mathbf{C}$, and \mathbf{D} denote the state-space matrices, $\mathbf{f}(k)$ are additive faults, and \mathbf{L} and \mathbf{M} are gain matrices of faults, respectively. $\mathbf{f}(k)$ may be composed of additive faults $f_L(k)$ on the input, and $\mathbf{f}_m(k)$ are additive faults on the output $\mathbf{f}^T(k) = [f_L^T(k) \mathbf{f}_m^T(k)]$.

For simplicity, it is assumed that $\mathbf{D} = 0$ which is true in many real cases and does not basically change the equations. Introducing Equ. (6) to (17) yields

$$\mathbf{y}(k+1) = \mathbf{CAx}(k) + \mathbf{CBu}(k) + \mathbf{CLf}(k) + \mathbf{Mf}(k) \quad (18)$$

For the next sampled output, it holds:

$$\begin{aligned} \mathbf{y}(k+2) &= \mathbf{CAx}(k+1) + \mathbf{CBu}(k+1) + \mathbf{CLf}(k+1) + \mathbf{Mf}(k+1) \\ &= \mathbf{CA}^2\mathbf{x}(k) + \mathbf{CABu}(k) + \mathbf{CBu}(k+1) + \mathbf{CALf}(k) + \\ &\quad \mathbf{CLf}(k+1) + \mathbf{Mf}(k+2) \end{aligned}$$

and for the q^{th} sample ($q \leq m$):

$$\mathbf{y}(k+q) = \mathbf{CA}^q\mathbf{x}(k) + \mathbf{CA}^{q-1}\mathbf{Bu}(k) + \dots + \mathbf{CBu}(k+q-1) + \mathbf{CA}^{q-1}\mathbf{Lf}(k) + \mathbf{CA}^{q-2}\mathbf{Lf}(k+1) + \dots + \mathbf{CLf}(k+q-1) + \mathbf{Mf}(k+q). \quad (19)$$

Equ. (19) gives the redundant equations for different time instants. The equations for a time window of length $q+1$ ($q \leq m$) are summarized, leading to:

$$\mathbf{Y}(k+q) = \mathbf{Tx}(k) + \mathbf{QU}(k+q) + \mathbf{Q}_F\mathbf{F}(k), \quad (20)$$

when time is shifted by q backwards

$$\mathbf{Y}(k) = \mathbf{Tx}(k-q) + \mathbf{QU}(k) + \mathbf{Q}_F\mathbf{F}(k-q), \quad (21)$$

with the vectors

$$\mathbf{Y}(k) = \begin{bmatrix} \mathbf{y}(k-q) \\ \mathbf{y}(k-q+1) \\ \vdots \\ \mathbf{y}(k) \end{bmatrix}, \quad \mathbf{U}(k) = \begin{bmatrix} \mathbf{u}(k-q) \\ \mathbf{u}(k-q+1) \\ \vdots \\ \mathbf{u}(k) \end{bmatrix}, \quad (22)$$

$$\mathbf{F}(k-q) = \begin{bmatrix} \mathbf{f}(k-q) \\ \mathbf{f}(k-q+1) \\ \vdots \\ \mathbf{f}(k) \end{bmatrix}$$

and the matrices

$$\mathbf{T} = \begin{bmatrix} \mathbf{C} \\ \mathbf{CA} \\ \vdots \\ \mathbf{CA}^q \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CB} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CAB} & \mathbf{CB} & & \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{q-1}\mathbf{B} & \mathbf{CA}^{q-2}\mathbf{B} & \dots & \mathbf{CB} & \mathbf{0} \end{bmatrix} \text{ and,} \quad (23)$$

$$\mathbf{Q}_F = \begin{bmatrix} \mathbf{M} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{CL} & \mathbf{M} & \dots & \mathbf{0} \\ \mathbf{CAL} & \mathbf{CL} & \mathbf{M} & \mathbf{0} \\ \vdots & \vdots & \vdots & \vdots \\ \mathbf{CA}^{q-1}\mathbf{L} & \mathbf{CA}^{q-2}\mathbf{L} & \dots & \mathbf{CL} & \mathbf{M} \end{bmatrix}$$

As the state vector $\mathbf{x}(k-q)$ is unknown, Equ. (21) is multiplied by a weighting matrix \mathbf{V} ,

$$\mathbf{VY}(k) = \mathbf{VTx}(k-q) + \mathbf{VQU}(k) + \mathbf{VQ}_F\mathbf{F}(k-q). \quad (24)$$

By selecting $\mathbf{V}_{h \times (q+1)r}$ such that $\mathbf{VT} = \mathbf{0}$, an input-output relation results. Consequently, residuals are defined as:

$$\mathbf{r}(k) = \mathbf{VY}(k) - \mathbf{VQU}(k) = \mathbf{VQ}_F\mathbf{F}(k). \quad (25)$$

In Equ. (25), $\mathbf{r}(k)$ includes h residuals. When $\mathbf{F}(k) = \mathbf{0}$, then $\mathbf{r}(k)$ becomes zero, meaning that no fault has occurred. If $\mathbf{F}(k) \neq \mathbf{0}$, then computing $\mathbf{r}(k)$ from (25) leads to a nonzero vector. The weighting matrix \mathbf{V} can be chosen such that the special properties of the residuals, e.g. structured residuals, are obtained. Here, structured residual vectors are generated such that each residual is at least independent from one of the faults. As a result, it is easier to detect the faults by these residuals.

B. Realization parity equations by CTDPN

This subsection describes the procedure of the realization of parity Equ. (25) by a CTDPN that plays the role of the fault diagnoser.

1) For each signal variable in Equ. (25), a place is assigned in the PN. Therefore, the CTDPN marking vector $\mathbf{M}(k)$ contains input and output vectors and their q backward is shifted (*i.e.* $\mathbf{U}(k)$ and $\mathbf{Y}(k)$ in Equ. (22)), and also $r_{h \times 1}$ are the residuals containing h residuals. Thus, the total number of places is $t = (p+r)(q+1) + h$.

$$\mathbf{M}(k) = [\mathbf{U}^T(k) \mathbf{Y}^T(k) \mathbf{r}^T(k)]^T = [m_1(k) m_2(k) \dots m_t(k)]^T. \quad (26)$$

2) The relationship between $\mathbf{M}(k)$ and $\mathbf{M}(k-1)$ is determined using Equ.s (22)-(25). Given the structure of $\mathbf{M}(k)$, it can be observed that for input vector $\mathbf{U}(k)$, and for $i = 0, 1, \dots, q$, $m(l) = u_j(k-q+i)$ (27)

where $l = j + ip$ and $j = 1, \dots, p$. For output vector $\mathbf{Y}(k)$, $m(l) = y_j(k-q+i)$ (28)

where $l = (q+1)p + j + ir$ and $j = 1, \dots, r$.

For residual vector $\mathbf{r}(k)$,

$$m(l) = r_j(k)$$

for $l = (q+1)p + (q+1)r + j$, and $j = 1, \dots, h$. (29)

3) It can be verified that the relationships described by Equ.s (27)-(29) can be written in the form of a matrix equation such that:

$\mathbf{M}(k) = \mathbf{HM}(k-1)$, where

$$\mathbf{H} = \begin{bmatrix} \mathbf{0}_{qp \times p} & \mathbf{I}_{qp \times qp} & \mathbf{0}_{qp \times r} & \mathbf{0}_{qp \times qr} & \mathbf{0}_{qp \times h} \\ \mathbf{0}_{p \times p} & \mathbf{0}_{p \times qp} & \mathbf{0}_{p \times r} & \mathbf{0}_{p \times qr} & \mathbf{0}_{p \times h} \\ \mathbf{0}_{qr \times p} & \mathbf{0}_{qr \times qp} & \mathbf{I}_{qr \times r} & \mathbf{0}_{qr \times qr} & \mathbf{0}_{qr \times h} \\ \mathbf{0}_{r \times p} & \mathbf{0}_{r \times qp} & \mathbf{0}_{r \times r} & \mathbf{0}_{r \times qr} & \mathbf{0}_{r \times h} \\ & & & -\mathbf{VQ} & \mathbf{V} & \mathbf{0}_{h \times h} \end{bmatrix}. \quad (30)$$

Note that $\mathbf{u}(k)$ and $\mathbf{y}(k)$ are independent of the other signals $\mathbf{u}(k-q)$ and $\mathbf{y}(k-q)$. Therefore, Rows 2 and 4 are in matrix \mathbf{H} are equal to zero.

4) By using Corollary 1 for this CTDPN,

$$\mathbf{post} = \mathbf{H}, \text{ and } \mathbf{Pre} = \mathbf{I}_{t \times t} \quad (31)$$

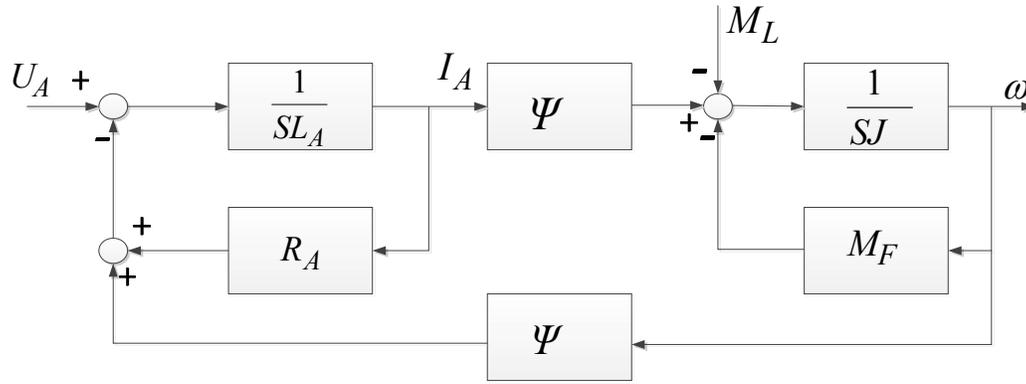


Fig. 1. Signal flow graph of the DC motor

For a better explanation of the method presented in this paper, an illustrative example is given in the next section.

V. AN ILLUSTRATIVE EXAMPLE

This section presents how to apply the proposed method to implement a fault diagnoser for a DC motor. Here, a permanently excited DC motor with a rated power of $P=550\text{W}$ at rated speed $n=2500\text{ rpm}$ is considered [35]. Fig. 1 depicts the signal flow graph of the motor and its data are given in Table 1.

The linear state-space model of this system becomes

$$\dot{\mathbf{x}} = \begin{bmatrix} \dot{I}_A \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{R_A}{L_A} & -\frac{\psi}{L_A} \\ \frac{\psi}{J} & -\frac{M_F}{J} \end{bmatrix} \begin{bmatrix} I_A \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_A} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} U_A \\ M_L \end{bmatrix}, \quad (32)$$

$$\mathbf{y} = \begin{bmatrix} I_A \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}.$$

Thus, the linear discrete-time state-space representation of the DC motor with a sampling time of 0.001 seconds is as follows:

$$\begin{aligned} \mathbf{x}(k+1) &= \begin{bmatrix} I_A(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} 0.7966 & -0.0433 \\ 0.1538 & 0.9959 \end{bmatrix} \begin{bmatrix} I_A(k) \\ \omega(k) \end{bmatrix} + \\ &\begin{bmatrix} 0.1313 & 0.0117 \\ 0.0117 & -0.5201 \end{bmatrix} \begin{bmatrix} U_A(k) \\ M_L(k) \end{bmatrix}. \\ \mathbf{y}(k) &= \begin{bmatrix} I_A \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \mathbf{x}(k). \end{aligned} \quad (33)$$

Here, I_A , U_A , and ω are the measured armature current, armature voltage, and speed of the motor, respectively, and M_L is the load torque.

From Equ. (27), discrete state-space parameters are obtained as follows:

TABLE I
DATA FOR THE DC MOTOR [35]

Armature resistance	$R_A = 1.52 \Omega$
Armature inductance	$L_A = 6.82 \cdot 10^{-3}$
Magnetic flux	$\psi = 0.33 \text{ V s}$
Voltage drop factor	$K_B = 2.21 \cdot 10^{-3} \text{ V s/A}$
Inertia constant	$J = 1.92 \cdot 10^{-3} \text{ kg m}^2$
Viscous friction	$M_F = 0.36 \cdot 10^{-3} \text{ Nm s}$

$$\mathbf{A} = \begin{bmatrix} 0.7966 & -0.0433 \\ 0.1538 & 0.9959 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.1313 & 0.0117 \\ 0.0117 & -0.5201 \end{bmatrix} \quad \text{and} \\ \mathbf{C} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}.$$

A. Fault diagnoser design

In this subsection, a fault diagnoser based on Section 4 is designed for the previously mentioned DC motor. The diagnoser is based on parity equations presented in Subsection 4.B. Two faults are considered here, a bias in the armature current sensor (f_1) and a bias in the speed sensor (f_2). Therefore, two structured residuals (r_1 and r_2) should be made such that when a fault occurs, only its corresponding residual remains unchanged and the other residual changes. In [35], a fault diagnoser has been designed for this case, and the results of that work are used here to realize a CTDPN-based fault diagnoser. According to [35], $q=2$ and therefore, from Equ. (17), we have:

$$\mathbf{Y}(k) = [I_A(k-2) \ \omega(k-2) \ I_A(k-1) \ \omega(k-1) \ I_A(k) \ \omega(k)]^T, \quad (34)$$

$$\mathbf{U}(k) = [U_A(k-2) \ M_L(k-2) \ U_A(k-1) \ M_L(k-1) \ U_A(k) \ M_L(k)]^T \quad (35)$$

$$\mathbf{r}(k) = [r_1(k) \ r_2(k)]^T. \quad (36)$$

Here, $r_1(k)$ and $r_2(k)$ are the residuals corresponding to f_1 and f_2 , respectively.

$\mathbf{M}(k)$ is obtained by substituting Equ.s (34)-(36) in (26) and the matrix \mathbf{Q} is computed using Equ. (23) for the matrices \mathbf{A} , \mathbf{B} , and \mathbf{C} of Equ. (32). For generating structured residuals, the weighting matrix \mathbf{V} is adapted from [35] as follows:

$$\mathbf{V} = \begin{bmatrix} \alpha & 0 & \beta & 0 & JL_A & 0 \\ 0 & \alpha & 0 & \beta & 0 & JL_A \end{bmatrix}, \quad (37)$$

where $\alpha = \psi^2 + R_A M_F$ and $\beta = L_A M_F + J R_A$. Selecting this weighting matrix causes the residuals to be independent of one another. Finally, the matrices $\mathbf{post} = \mathbf{H}$ and $\mathbf{Pre} = \mathbf{I}_{14 \times 14}$ are obtained from Equ. (31). Here, only the

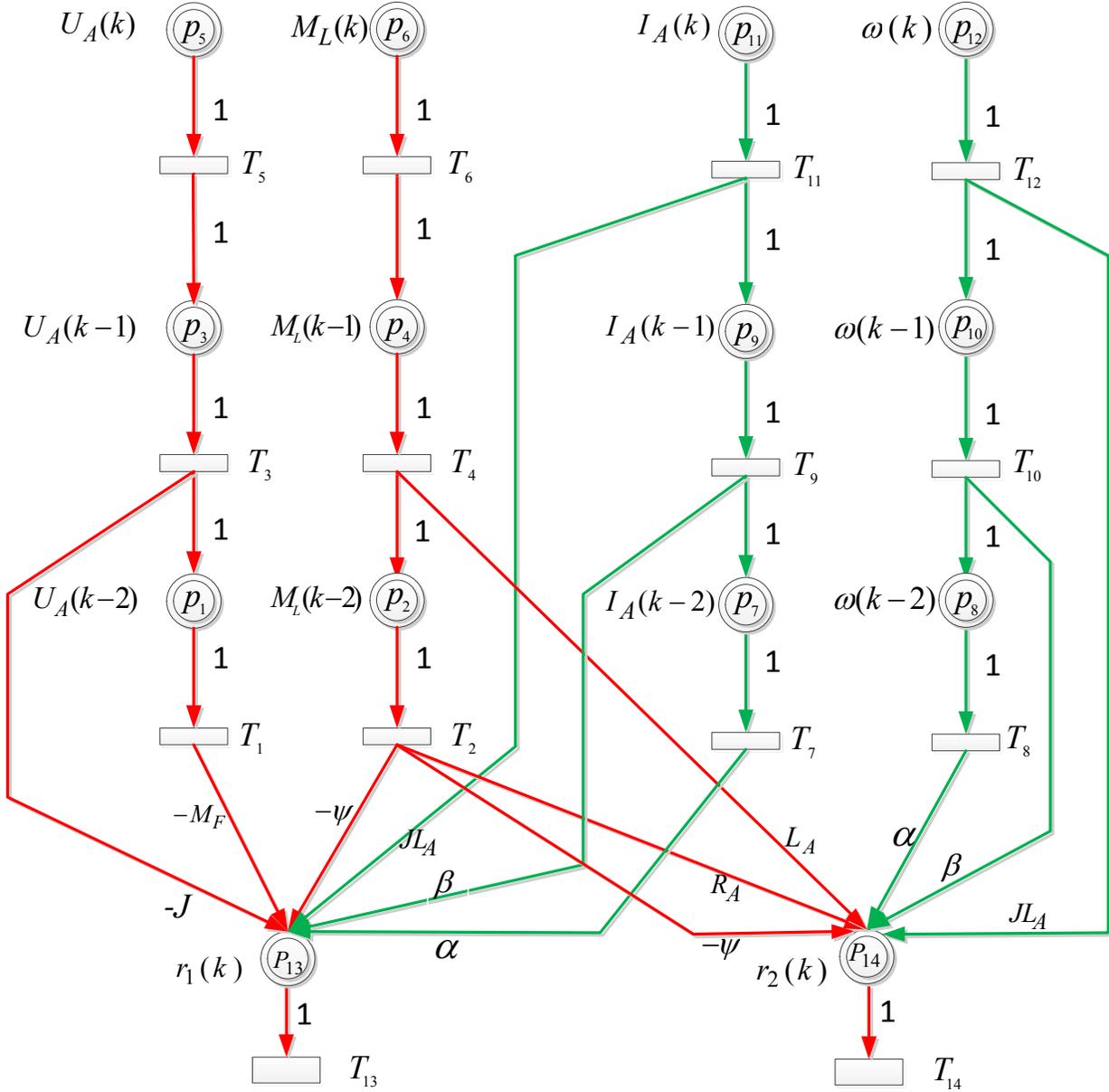


Fig. 2. CTDPN fault diagnoser for the DC motor

last two lines of matrix \mathbf{H} , which describe the parity equations, are presented.

$$\mathbf{W}_h = [-\mathbf{V}\mathbf{Q} \quad \mathbf{V}] = \begin{bmatrix} -M_F & -\psi & -J & 0 & 0 & 0 & \alpha & 0 & \beta & 0 & JL_A & 0 \\ -\psi & R_A & 0 & L_A & 0 & 0 & 0 & \alpha & 0 & \beta & 0 & JL_A \end{bmatrix} \quad (38)$$

According to Equ. (25), the corresponding CTDPN that plays the role of a fault diagnoser for the main system is obtained and depicted in Fig. 2. In this CTDPN, the places p_{13} and p_{14} are redundant places that compute the residuals indicating the faults. The remainder of the network provides the delay timed signals from the inputs and the outputs, required for residual calculations.

The performance of the proposed diagnoser is investigated here in three cases. In all cases, $U_A = 24 \text{ v}$ and $M_L = 0.3 \text{ Nm}$ are applied to the DC motor.

B. Case 1: Faultless system

No fault occurs in this case. Fig. 3-a shows the residuals generated by the CTDPN fault diagnoser (i.e. $r_1(k)$ and $r_2(k)$, the markings of places p_{13} and p_{14} in Fig. 2). Evidently, the two residual values converge to zero after a transient time (about 0.2 s), indicating no fault occurrence.

C. Case 2: The armature current sensor fault

In this case, a 10% positive bias is added to the measured

current armature due to a fault of the current sensor. It is assumed that the fault affected the sensor at $t=0.3$ s. Fig. 3-b shows that $r_1(k)$ remains unchanged, while $r_2(k)$ is changed abruptly at $t=0.3$ s, indicating that the fault f_1 has occurred at $t=0.3$ s. For more clarity, the residuals are normalized with respect to their thresholds and depicted for steady-state time ($t \geq 0.2$ s).

D. Case 3: The speed sensor fault

This case is similar to Case 2, differing in that only the speed sensor is affected by the same fault of that case. The changes in the residuals are depicted in Fig. 3-c. In this figure, $r_2(k)$ remains unchanged, whereas $r_1(k)$ is changed, indicating that f_2 occurred at $t=0.3$ s.

VI. COMPARISON WITH SIMILAR STUDIES

A. Classical methods:

The equations used in this paper are the same as the classical equations in fault diagnosis methods such as parity equations [16], but the main feature of this paper is that it provided an intuitive model for implementing classical methods. The use of PNs also provided the capability of developing the proposed method for hybrid systems.

B. Redundant PN diagnoser:

Similar to [13-15], in this paper, we constructed a redundant PN as a diagnoser in which the markings of places were equal to residuals indicating the occurred faults. The main difference between our paper and [13-15] lies in the type of network used and its equations. In this paper, a continuous type of PN was employed with difference equations, while the PNs used in [13-15] were ordinary PNs with algebraic equations for DESs

Differential PNs:

In the literature review, the difference between CTDPN and regular CPNs and HPNs was stated. In [36] a special CPN has been introduced and known as differential Petri nets (DPNs). In a DPN, similar to a CTDPN, markings and arc values can be negative real numbers and are, therefore, suitable for modeling dynamic systems. The main difference between this network and CTDPN is that DPN is applicable for modeling continuous-time dynamical systems and differential equations, whereas CTDPN is suitable for discrete-time systems and difference equations. In [37], DPN is incorporated for modeling a hybrid system, and in [38], it is applied as a state observer for a continuous-time linear switched system.

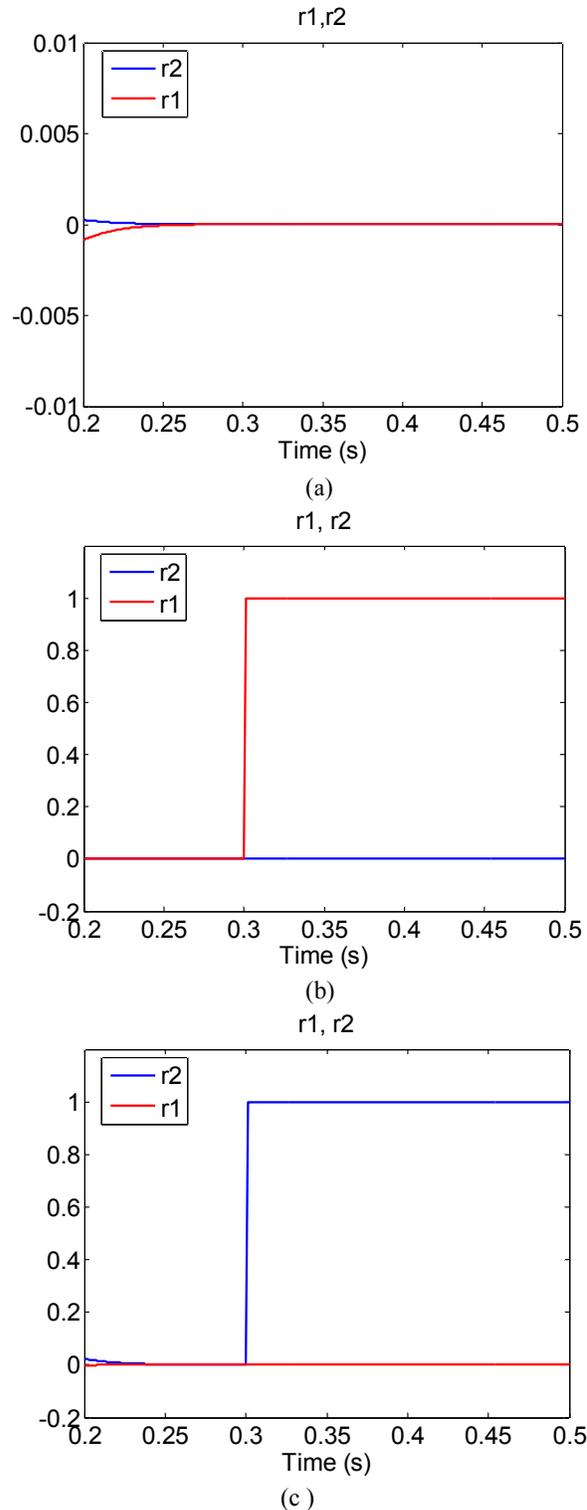


Fig. 3. Residuals generated by the CTDPN fault diagnoser for a) the faultless system, b) the fault f_1 , c) the fault f_2

VII. CONCLUSIONS

In this paper, a new method for detecting the faults of a continuous linear system by a special continuous PN (CTDPN) was presented. Based on this method, one can design a classic fault diagnoser (e. g. parity equation fault diagnoser) and then realize it by a CTDPN. An advantage of this diagnosis system is the ability for implementation on the same PLC that controls the system. The realization of the CTDPN diagnosis system by PLCs, as a relevant future research direction, can lead to a reduction in costs and an increase in the speed of the fault diagnosis of industrial systems.

Another issue related to this paper is the transformation of the difference equations describing the main system to a CTDPN, as will appear in our future studies. From this point of view, the fault diagnoser CTDPN is regarded as a redundant CTDPN, similar to the idea presented in [13] for monitoring faults in DESs. Integrating these two lines of research will result in a unique and general framework for the fault diagnosis of hybrid systems.

ACKNOWLEDGMENT

Author would like to acknowledge the financial support of Kermanshah University of Technology for this research under grant number S/P/T/1194.

REFERENCES

- [1] A. Karimabadi, M. E. Hajiabadi, E. Kamyab, A. A. Shojaei, "The effect of condition monitoring of circuit breaker on the reliability and maintenance cost of substation", *IECCO*, Vol. 2, No. 3, pp. 167-176, July 2019.
- [2] Z. Gao, C. Cecati, S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: fault diagnosis with model-based and signal-based approaches", *IEEE Trans. on industrial electronics*, Vol. 62, NO. 6, pp. 3757-3767, June 2015.
- [3] A. Giua, M. Silva, "Petri Nets and automatic control: A historical perspective", *Annual Reviews in Control*, Vol. 45, pp. 223-239, 2018.
- [4] R. David and H. Alla, *Discrete, continuous, and hybrid Petri Nets*, Springer-Verlag, Berlin Heidelberg, 2010.
- [5] Dimitri Lefebvre, *Diagnosis of discrete event systems with Petri Nets*, chapter 16 of *Petri Net, Theory and Applications*, Book edited by Vedran Kordic, Vienna, Austria, pp. 534, Feb. 2008.
- [6] M. Iordache and P. Antsaklis, "A survey on the supervision of Petri Nets," *DES Workshop PN 2005*, Miami, FL, June 21, 2005.
- [7] N. Ran, H. Su; A. Giua; and C. Seatzu, "Codiagnosability analysis of bounded Petri Nets," *IEEE Trans. on Automatic Control*, Vol. 63, Issue 4, pp. 1192 – 1199, 2018.
- [8] N. Ran, S. Wang, H. Su, and C. Wang, "Fault diagnosis for discrete event systems modeled by bounded Petri Nets," *Asian Journal of Control*, Vol. 19, No. 6, pp. 1–10, Nov. 2017.
- [9] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, Kluwer, 2nd Edition, 2008.
- [10] C. Mahulea, C. Seatzu, M. P. Cabasino, and M. Silva, "Fault diagnosis of discrete event systems using continuous Petri Nets," *IEEE Trans. on Systems, Man, and Cybernetics—PART A: Systems and Humans*, Vol. 42, No. 4, pp. 970-984, July 2012.
- [11] A. A. Farahani and, A. Dideban, "Continuous-Time Delay-Petri Nets as a new tool to design state-space controller", *Information Technology and control*, T. 45, No. 4, pp. 401-411, 2016.
- [12] A. Dideban, A. A. Farahani and M. Razavi, "Modeling of continuous systems using modified Petri Net model", *Journal of Modeling and Simulation of Electrical and Electronics Engineering (MSEEE)*, Vol. 1, No. 2, pp. 19-23, May 2015.
- [13] Y. Wu and C. N. Hadjicostis, "Algebraic approaches for fault identification in Discrete-Event Systems," *IEEE Transaction on Automatic Control*, Vol. 50, No. 12, pp. 2048-2053, Dec. 2005.
- [14] V. Calderaro, C. N. Hadjicostis, A. Piccolo, and P. Siano, "Failure Identification in smart grids based on Petri Net modeling," *IEEE Trans. on Industrial Electronics*, Vol. 58, No. 10, pp. 4613-4623, Oct. 2011.
- [15] V. Calderaro, V. Gladi, A. Piccolo, and P. Siano, "Protection system monitoring in electric networks with embedded generation using Petri Nets," *International Journal of Emerging Electric Power Systems*, Vol. 9, Issue 6, Article 3, 2008.
- [16] R. Isermann, *Fault-Diagnosis Applications*, Springer-Verlag, Berlin Heidelberg 2011.
- [17] J. Zaytoon and B. Riera, "Synthesis and implementation of logic controllers—A review," *Annual Reviews in Control*, Vol. 43, pp. 152-168, 2017.
- [18] W. Bolton, *Programmable Logic Controllers*, sixth edition, Elsevier, 2015.
- [19] F. G. Cabral, M. V. Moreira, O. Diene and J. C. Basilio, "A Petri Net diagnoser for discrete event systems modeled by Finite State Automata," *IEEE Transactions on Automatic Control* Vol. 60, Issue: 1, pp. 59 - 71, Jan. 2015.
- [20] A. D. Vieira, E. A. P. Santos, M. H. de Queiroz, A. B. Leal, A. D. de Paula Neto; J. E. R. Cury, "A Method for PLC implementation of supervisory control of discrete event systems", *IEEE Trans. on Control Systems Technology*, Vol. 25, No. 1, pp. 175-191, JAN, 2017.
- [21] J. C. Quezada, J. Medina, E. Flores, J. C. Seck Tuoh, A. E. Solis, V. Quezada, "Simulation and validation of diagram ladder—Petri Nets", *The International Journal of Advanced Manufacturing Technology*, Vol. 88, Issue 5–8, pp. 1393–1405, February 2017.
- [22] P. Nazemzadeh, A. Dideban and, M. ZareieeA., "Fault modeling in discrete event systems using Petri Nets", *ACM Transactions on Embedded Computing Systems (TECS) - Special Issue on Modeling and Verification of Discrete Event*, Vol. 12, Issue 1, Article No. 12, Jan. 2013.
- [23] X. Wang, C. Mahulea and M. Silva, "Diagnosis of Time Petri Nets using fault diagnosis graph", *IEEE Transaction on Automatic Control*, Vol. 60, No., pp. 92321- 2335, September 2015.
- [24] X. Zhang; S. Yue; and X. Zha, "Method of power grid fault diagnosis using intuitionistic fuzzy Petri Nets," *IET*

- Generation, Transmission & Distribution, Vol: 12, Issue: 2, pp. 295–302, 2018
- [25] J. Zaytoon, S. Lafortune, “Overview of fault diagnosis methods for Discrete Event Systems, Annual Reviews in Control Vol. 37, pp. 308–320, 2013.
- [26] Baniardalani S. and Askari, J., “Fault diagnosis of timed discrete event systems using Dioid Algebra”, IJCAS, Vol. 11, No. 6, pp. 1095-1105, Dec. 2013.
- [27] G. Zhu, Z. Li, N. Wu, and A. Al-Ahmari, “Fault identification of discrete event systems modeled by Petri Nets with unobservable transitions,” IEEE Trans on systems, man, and cybernetics: Systems 2, pp., 333-345, Feb 2019.
- [28] G. Cavone, M. Dotoli, and C. Seatzu, “A survey on Petri Net models for freight logistics and transportation systems,” IEEE Trans on intelligent transportation systems, Vol. 19, Issue: 6, pp. 1795-1813, June 2018.
- [29] M. A. Drighiciu, “Hybrid Petri Nets a framework for hybrid systems modeling,” 2017 International Conference on Electromechanical and Power Systems, 11-13 Oct. 2017, Iasi, Romania.
- [30] G. Russo, M. Pennisi, R. Boscarino and F. Pappalardo, “Continuous Petri Nets and microRNA analysis in melanoma,” IEEE/ACM Transactions on Computational Biology and Bioinformatics, Vol. 15, Issue 5, pp. 1492-1499, 2018.
- [31] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich and P. Cheung, “Monitoring and fault diagnosis of hybrid systems,” IEEE Trans. on Systems, Man, and Cybernetics—PART A: Systems and Humans, Vol. 35, NO. 6, pp. 1225-1239, Dec. 2005.
- [32] R. Casas-Carrillo, O. Begovich, J. Ruiz-Le´on, and S. C`elikovsky, “Adaptive fault diagnoser based on PSO algorithm for a class of Timed Continuous Petri Nets,” IEEE 21st International Conference on Emerging Technologies and Factory Automation, Germany, 6-9 Sept., 2016.
- [33] J. A. Fraustro, J. Ruiz-Le´on, C.R. V´azquez, A. R. Trevino, “Structural fault diagnosis in timed continuous Petri Nets” Proceedings of the 13th International Workshop on Discrete Event Systems, Xi’an, China, May 30 - June 1, 2016.
- [34] C. Seatzu, M. Silva, and J. H. van Schuppen, Control of Discrete-Event Systems, Springer-Verlag London 2013.
- [35] R. Isermann, Fault-Diagnosis Systems, Springer-Verlag Berlin Heidelberg 2006.
- [36] I. Demongodin and N. T. Koussoulas, “Differential Petri Nets: Representing Continuous Systems in a Discrete-Event World,” IEEE Trans on automatic control, Vol. 43, No. 4, pp. 573-579, April 1998.
- [37] G. Davrazos, and N. T. Koussoulas, “Modeling and stability analysis of state-switched hybrid systems via Differential Petri Nets,” Simulation Modeling Practice and Theory 15 pp. 879–893. (2007)
- [38] F. Hamdi, N. Messai, and N. Manamanni, “State Estimation for switched systems described by differential Petri Nets Models,” Proceedings of the 3rd International Conference on Systems and Control, Algiers, Algeria, October 29-31, 2013.



Sobhi Baniardalani was born in Kermanshah, Iran. He received his B.Sc. degree in Electronics Engineering from Isfahan University of Technology, Isfahan, Iran, in 1989, and his M.S. degrees in Electronics Engineering from Amirkabir University of Technology, in 1994, and the Ph.D. degree from Isfahan

University of Technology, Isfahan, Iran, in 2010, in Control engineering. In 2014, he joined Kermanshah University of Technology as an Assistant Professor in Department of Electrical Engineering, Faculty of Energy. His current research interests include Fault Diagnosis Systems, Discrete Event Systems and Hybrid Systems.