JJEE
Jordan Journal of Electrical Engineering

Volume 7, Number 4, December 2021
Pages 324-343
ISSN (Print): 2409-9600, ISSN (Online): 2409-9619

Observer Based Fault Diagnosis Systems Using Continuous-Time Delay Petri Nets

Sobhi Baniardalani*

Department of Electrical Engineering, Kermanshah University of Technology, Kermanshah, Iran
E-mail: s.baniardalani@kut.ac.ir

Received: March 17, 2021 *Revised: May 09, 2021* *Accepted: May 13, 2021*

Abstract— This paper addresses fault diagnosis in dynamic systems represented by discrete state-space models. The main idea of the paper is to propose a systematic way to implement a Petri net-based fault diagnosis system. This procedure consists of three main steps. In the first step, a fault diagnosis system is built based on the Luenberger observer. In the second step, the obtained fault diagnoser equations are transformed to a suitable format. Finally, in the third step, the obtained equations are implemented by a Petri net called continuous-time delay Petri net (CTDPN) that can realize difference equations. Based on this method, a systematic approach is proposed for realizing a classical fault diagnoser by CTDPN. By integrating the concept of state-space observers and PNs in this paper, new and effective methods are developed for the analysis and fault diagnosis of systems - known as hybrid systems - that have both continuous and discrete variables. The performance of the proposed method is thoroughly investigated, and the obtained results show that the proposed CTDPN can precisely detect the occurred faults, their types and their occurrence time instances.

Keywords— Continuous-time delay Petri net; Fault diagnosis; Luenberger state observer; State-space model.

1. INTRODUCTION

As industrial processes become more complex, the need for more reliability and availability of these systems has increased. For this reason, the problem of fault diagnosis and fault tolerability in the system has been a major part of research in recent years [1-5]. Among the various methods for fault diagnosis in dynamic systems, model-based fault diagnosis methods are of great importance and have many applications in a wide range of fields. These methods, which have been common since the 1970s, use a model to predict system behavior. Due to the increasing capability and efficiency of computers and the development of control theories in recent decades, attention to these methods has also increased. Despite the breadth and variety of these methods, they can be divided into two general categories of quantitative and qualitative methods based on the type of model and the nature of the signals measured [6-8]. Quantitative model-based methods, which are the subject of this paper fall into these approaches and include three main categories: parameter estimation methods, observer-based methods and methods based on parity equations [9].

Applying the aforementioned methods to fault diagnosis for real plants faces critical challenges. One of the main challenges is that many of these systems have both continuous and discrete variables. These systems, known as hybrid systems, have many complexities and their modeling is a challenge [10, 11]. The most common frameworks for modeling these systems are hybrid automata and hybrid Petri nets (PNs). PNs are powerful modeling tools with intuitive graphical representation for the simulation of systems. Since their introduction by Carl Adam Petri, a wide variety of these networks have been developed and used for many applications. They also provide a formalism for setting up state equations and algebraic

* Corresponding author

Article's DOI: 10.5455/jjee.204-1616016470

equations governing the behavior of the system [12]. Although classic PNs were only used to model discrete event systems (DESs) in the early years, certain types of continuous PNs have been introduced – recently - to model continuous dynamic processes. This led to improved analysis and modeling of hybrid systems [13, 14].

In recent years, a lot of work has been done in the field of modeling hybrid systems by continuous PNs, but only a handful of these papers are devoted to the subject of fault diagnosis of such systems, while in light of the features of continuous PNs, more effective fault diagnosis systems can be built for hybrid systems [15, 16]. In [17, 18], a special type of continuous PN called continuous time delay Petri net (CTDPN) - which can model discrete-time continuous linear systems - was introduced. In [19], a systematic method was presented for implementing state equations by CTDPN. In [20], with the help of these PNs, a fault diagnosis system based on parity equations has been proposed.

1.1. Related Literature

Using PNs for fault diagnosis systems has been reviewed in some papers simultaneously with other models such as automata [21-26]. In [22], three main methods - including the algebraic method for fault diagnosis - have been investigated. In [27-29], faults have been considered for both transitions and places in PNs. Then, a redundant diagnoser PN has been constructed and attached to the main PN. In the diagnosis scheme presented in [27], nonzero marking values of redundant places are decoded to indicate the faults. Similar to [27], in this paper, we constructed a redundant CTDPN in which the marking values are identical to the residuals indicating the faults. In [30], faults were modeled by unobservable transitions and an algorithm was presented to separate the faulty part. In that 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 [24, 25], a diagnoser based on the reachability graph has been proposed and analyzed for bounded PNs (a PN with limited marking value). Since this graph cannot be constructed for continuous PNs, that method is not applicable for continuous variable dynamic systems (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 [17, 21, 31]. Authors of [31] presented a survey of using HPNs for logistic purposes. In [32], CPNs have been applied for modeling and analyzing biological systems. The marking values of PNs introduced in [31-33] are positive real numbers. Thus, those PNs cannot be utilized for modeling CVDSs whose state variables have negative values, while this limitation does not exist in the network employed in our paper.

A few studies have addressed the fault diagnosis in TCPN [34-36]. In [34], a fault modeling framework based on hybrid automata has been presented. In that paper, an online monitoring approach based on a timed PN model abstracted the continuous dynamic system into a DES. Contrary to [34], a CVDS is modeled directly by a CTDPN in the present paper. Moreover, authors of [35] have implemented an adaptive fault diagnoser for a system modeled by a TCPN under infinite server semantics. That work had proposed a single diagnoser model whose structure was known, and its parameters were updated depending on the fault occurrence. In that model, identification algorithms based on heuristic optimization methods have been used to identify unknown fault parameters.

Compared with the present study, the method proposed in [35] is more complex and cannot be implemented online by programmable logic controllers (PLCs).

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.

1.2. Motivation

Over the past decade, the use of powerful PLCs has been widespread in various industries [37, 38]. As a result, the idea of implementing the online diagnoser and the feedback controller on the same PLC, has been given more attention. This implementation reduces additional equipment for fault diagnosis purposes, such as sensors and communication buses. It also increases the fault detection speed [16, 39, 40].

In this way, an essential and challenging issue is that the online diagnoser system should be easily implemented by one of the standard PLC programming languages. As shown in the few works that addressed the problem of implementing online diagnosis on a PLC, PN-based diagnosis has better compatibility with PLC programming languages such as ladder diagram (LD) or sequential function chart (SFC) [16, 39-41]. All those works are about DESs, and to the best of the author's knowledge, no research has considered 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.
- Providing a unified formalism based on PNs for fault diagnosis in hybrid systems that exhibit both continuous and discrete dynamic behaviors.
- Access to more convenient and efficient online fault diagnosers implemented on PLCs.

1.3. Novelty

The main innovation of this paper is proposing an algorithm to construct a diagnostic system by a CTDPN. In [20], we have presented an approach for the fault diagnosis method based on parity equations. In this paper, this issue is developed for the fault diagnosis method based on an observer. To the best of the author's knowledge, this has not been done yet and could be an essential step in implementing well-known fault diagnosis methods for hybrid systems.

In this paper, based on the CTDPN introduced in [20], an observer-based fault diagnosis system is realized for a discrete-time continuous linear system. Initially, a fault diagnoser is designed using the conventional Luenberger method. The inputs of this diagnoser are observed values of the system's input and output signals. The output signals of the diagnoser known as residuals indicate the faults occurrence and their type.

This diagnosis system consists of a set of difference equations, which are then implemented by a Petri net. The network used is a CTDPN. The main innovation of this paper is to present a systematic method for building this network.

The CTDPN proposed in this paper is an integrated network that both simulates the system and generates residues. This paper presents a systematic method for implementing an observer-based diagnostic system with the help of PNs. In the resulting PN, places - whose marking values determine the residues - are considered. The most important novelty of the paper is that it provides a simple and intuitive way to diagnose faults in dynamic systems, and can be easily generalized to hybrid systems.

This paper is organized as follows: in section 2, an overview of the CTDPN is presented. In section 3, the observer-based fault diagnosis method for discrete-time systems is briefly reviewed. Section 4 illustrates the implementation steps of the proposed diagnoser by the proposed PN, and – finally - section 5 examines the performance of the proposed system using an example.

2. PRELIMINARIES

In this section, some of the main definitions concerned with continuous PN and CTDPN are introduced. For further reading, it is recommended to refer to [17, 26].

Definition 1 [26]: A PN is a structure $N = (P, T, Pre, Post, M_0)$ 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, $Pre: P \times T \rightarrow N$ is the pre-incidence function specifying the number of arcs directed from places to transitions (called *"pre"* arcs) and represented by an m×n matrix, $Post: P \times T \rightarrow N$ is the post-incidence function specifying the number of arcs directed from transitions to places (called *"post"* arcs) and represented by an m×n matrix, and M_0 is the initial marking. The incidence matrix W is equal to $W = Post - Pre$. For a transition $\in T$, its sets of input and output places are respectively defined as $\bullet t = \{p \in P \mid Pre\ (p, t) > 0\}$, and $t \bullet = \{p \in P \mid Post\ (p, t) > 0\}$, while given a place $p \in P$, its set of input and output transitions are defined as $\bullet p = \{t \in T \mid Post\ (p, t) > 0\}$, $p \bullet = \{t \in T \mid Pre\ (p, t) > 0\}$.

Definition 2 [26]: A marking is a function $M: P \rightarrow N$ that assigns to each place a non-negative integer number of tokens. The marking of a PN defines its state.

Definition 3 [26]: A transition t_i is enabled at a marking M, if $M \geq Pre\ (\cdot, t_i)$, i.e., if each place $p \in P$ contains a number of tokens greater than or equal to $Pre\ (p, t_i)$.

Definition 4 [26]: A transition t_i enabled at a marking M can fire. The firing of t_i removes $Pre\ (p, t_i)$ tokens from each place $p \in P$ and adds $Post\ (p, t_i)$ tokens in each place $p \in P$, yielding a new marking:

$$M' = M - Pre(.\,, t_i) + Post(.\,, t_i) = M + W(.\,, t_i). \tag{1}$$

To denote that the firing of t_i from M leads to M', we write $M\ [t_i \rangle\ M'$. Let S be a firing sequence which can be performed from a marking M_0, and suppose that M_k is reachable from M_0 by applying S [4, 11]. The characteristic vector of sequence S, written as 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.

$$s = s_1 + s_2 + \dots + s_{k-1}, \tag{2}$$

where s_j represents the firing transition of t_j. In this case, M_k is obtained by fundamental equation:

$$M_k = M_o + Ws. \tag{3}$$

Definition 5 [21]: A marked autonomous CPN is a five-tuple $R = (P, T, Pre, Post, M_0)$ such that P and T are the same as Definition 1, $Pre: P \times T \to R_+$ and $Post: P \times T \to R_+$ are the input and output incidence applications, respectively and $M_0: P \to R_+$ is the initial marking. $Pre(P_i, t_j)$ denotes the weight of the arc $P_i \to t_j$ and it is a positive real number if the arc exists, and 0 otherwise. Similarly, $Post(t_j, P_i)$ is the weight of the arc $t_j \to P_i$. 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 [21]: In a continuous PN, the enabling degree of transition t_j for marking M, denoted by q or q (t_j, M), is the real number q such that:

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

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, Eq. (4) can be simplified as:

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

Definition 7 [21]: A 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, Spe (t_j)=V_j is the maximal speed associated with transition t_j. The instantaneous firing speed $v_j(t_j)$ satisfies the condition $v_j(t_j) \leq V_j(t_j)$ [21]. For a TCPN, between times τ and $\tau + d\tau$, the quantity of firing of t_j is $v_j(\tau).d\tau$; then, s in Eq. (3) corresponds to the vector $v(\tau).d\tau$. It follows that:

$$dM(\tau) = W \cdot v(\tau), \tag{6}$$

then

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

is the fundamental equation for a TCPN. It works for any $0 \leq \tau_1 \leq \tau_2$ [21].

Definition 8 [17]: A CTDPN is a TCPN 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 [17]:

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 $M(p_i) > 0$ or $M(p_i) < 0$.

Assumption 3. The speed of the transitions is infinity.

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

According to the theorem presented in [20], it can be seen that when t_j is fired at time instance kT_s, the entire marking is transported instantaneously at this time such that:

$$\Delta m_j = \int_{kT_s^-}^{kT_s^+} dm_j = q_j(kT_s^-) = \int_{kT_s^-}^{kT_s^+} v_j(\tau) \,.\, d\tau. \tag{8}$$

Here, $kT_s^- = kT_s - \varepsilon = (k-1)T_s$ and $kT_s^+ = kT_s + \varepsilon = kT_s$, where $\varepsilon \to 0$. For simplicity, from now on, kT_s is abbreviated to k. From Corollary 1 of [15], it is also concluded that:
$q(t_j) = \underset{i:\, P_i \in \bullet t_j}{min}\big(m(P_i)\big) = m(P_j))$.

By considering all marking places, it can be seen that $\int_{\tau_1}^{\tau_2} v(\tau)\,.\,d\tau = M(k-1)$, where $\tau_1 = (k-1)T_s$ and $\tau_2 = \ kT_s$. Finally, Eq. (7) for this network can be written as follows:

$$M(k) = M(k-1) + WM(k-1) = (I+W)M(k-1) \qquad (9)$$

3. OBSERVER BASED FAULT DIAGNOSER DESIGN

This section briefly reviews how to design fault diagnoser system based on Leuenberger's observer [20].

For a given linear system with p inputs, r outputs and n state variables, the discrete-time state-space model is described by:

$$x(k+1) = Ax(k) + Bu(k) + Lf_l(k)) \qquad (10)$$

$$y(k) = Cx(k) + Du(k) + Mf_m(k), \qquad (11)$$

in which A, B, C and D denote the state-space matrices, $f_l(k)$ are additive faults on the states, $f_m(k)$ are additive faults on the outputs and L and M are gain matrices of faults. The number of additive faults with states is equal to l, and the number of additive faults with outputs is equal to m.

Based on Luenberger's design method, an observer is realized in the form of Eqs. (12) and (13).

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + H(y(k) - C\hat{x}(k) \qquad (12)$$

$$\hat{y}(k) = C\hat{x}(k). \qquad (13)$$

where $\hat{x}(k)$ is the estimated state vector, $\hat{y}(k)$ is the estimated output vector, and H is the observer gain matrix. Note that in the observer output equations, only the estimated states are considered so that the fault can be detected by comparing the output produced by the observer and the actual output [9]. Without losing the generality, and for simplicity, the matrix D is assumed zero according to most real examples in this paper.
Also, $\tilde{x}(k) = x(k) - \hat{x}(k)$ denotes state estimation error and

$$r(k) = e(k) = y(k) - \hat{y}(k) = c\tilde{x}(k) + Mf_m(k) \qquad (14)$$

is a vector that shows residuals used for fault diagnosis.

By calculating residuals at k, k+1, and k+2 time instances, the following results are obtained:

$$\hat{x}(k{+}1){=}A\hat{x}(k){+}Bu(k){+}Hy(k){-}HC\,\hat{x}(k){=}\big(A{-}HC\big)\hat{x}(k){+}Bu(k){+}Hy(k).$$

$$\hat{x}(k{+}1){=}\big(A{-}HC\big)\hat{x}(k){+}Bu(k){+}HCx(k){+}HMf_m(k).$$

$$\hat{y}(k{+}1){=}C\big(A{-}HC\big)\hat{x}(k){+}CBu(k){+}CHCx(k){+}CHMf_m(k)$$

$$y(k{+}1){=}Cx(k{+}1){+}Mf_m(k{+}1){=}CAx(k){+}CBu(k){+}CL\,f_L(k){+}\ Mf_m(k{+}1)$$

$$e(k+1) = y(k+1) - \hat{y}(k+1) = CAx(k) + CBu(k) + CLf_l(k) - C(A-HC)\hat{x}(k) - CBu(k) - CHCx(k) - CHMf_m(k) + Mf_m(k+1) = C(A-HC)x(k) - C(A-HC)\hat{x}(k) + CLf_l(k) - CHMf_m(k) + Mf_m(k+1) \qquad (15)$$

$$
\begin{aligned}
x(k+2) &= Ax(k+1) + Bu(k+1) + L\,f_L(k+1) \\
&= CA^2x(k)+CABu(k)+CALf_l(k)+CBu(k+1)+CLf_l(k+1)+Mf_m(k+2)
\end{aligned}
$$

$$y(k+2)=Cx(k+2)+M\,f_m(k+2)= CAx(k+1)+CBu(k+1)+CL\,f_l(k+1)+ M\,f_l(k+2)$$

$$y(k+2)=CA^2x(k)+CABu(k)+CALf_l(k)+CBu(k+1)+CLf_l(k+1)+Mf_m(k+2)$$

$$
\begin{aligned}
\hat{x}(k+2)=&(A\text{-}HC)^2\,\hat{x}(k)+(A\text{-}HC)Bu(k)+(A\text{-}HC)HCx(k)+(A\text{-}HC)HMf_m(k)+HCAx(k)+Bu(k+1)+ \\
&HCBu(k)+HCL\,f_l(k)+HMf_m(k+1)
\end{aligned}
$$

$$
\begin{aligned}
\widehat{y}(k+2) = &C(A-HC)^2\,\hat{x}(k) + C(A-HC)Bu(k) + CHCBu(k) + [C(A-HC)HC + CHCA]x(k) \\
&+C(A-HC)HMf_m(k) + CHMf_m(k+1) + CBu(k+1) + CHCLf_l(k)
\end{aligned}
$$

$$
\begin{aligned}
e(k+2)=&y(k+2)\text{-}\hat{y}(k+2)=e(k+2)=y(k+2)-\hat{y}(k+2) \\
&=CA^2\,x(k)+CABu(k)+CALf_l(k)+CBu(k+1)+CLf_l(k+1)+M\,f_m(k+2)\text{-}C\,(A\text{-}HC)^2\,\hat{x}(k)- \\
&C(A\text{-}HC)Bu(k)\text{-}CHCBu(k)-[C(A\text{-}HC)HC+CHCA]x(k)\text{-}C(A\text{-}HC)HMf_m(k) \\
&\text{-}CHMf_m(k+1)\text{-}CBu(k+1)\text{-}CHCf_l(k) \\
&= [CA^2\text{-}CAHC+C\,(HC)^2\text{-}CHCA]x(k)+CALf_l(k)+CLf_l(k+1)+Mf_m(k+2)\text{-} \\
&C(A\text{-}HC)^2\;\hat{x}(k)\text{-}C(A\text{-}HC)HMf_m(k)\text{-}CHMf_m(k+1)
\end{aligned}
$$

$$
\begin{aligned}
e\,(k+2)=&C[A\text{-}HC]^2\,(x(k)\text{-}\hat{x}(k))+Mf_m(k+2)\text{-}CHMf_m(k+1)\text{-}C(A\text{-}HC)HMf_m(k) \\
&+CLf_l(k+1) + CALf_l(k)
\end{aligned}
\tag{16}
$$

From Eqs. (14-16), results:

$$
\begin{aligned}
\begin{bmatrix} e(k) \\ e(k+1) \\ e(k+2) \end{bmatrix} = &\begin{bmatrix} C \\ C(A-HC) \\ C(A-HC)^2 \end{bmatrix} \tilde{x}(k) + \begin{bmatrix} M & 0 & 0 \\ \text{-}CHM & M & 0 \\ \text{-}C(A\text{-}HC)HM & \text{-}CHM & 0 \end{bmatrix} \begin{bmatrix} f_m(k) \\ f_m(k+1) \\ f_m(k+2) \end{bmatrix} \\
&+\begin{bmatrix} 0 & 0 & 0 \\ CL & 0 & 0 \\ C(A-HC)L & CL & 0 \end{bmatrix} \begin{bmatrix} f_l(k) \\ fl(k+1) \\ fl(k+2) \end{bmatrix}.
\end{aligned}
\tag{17}
$$

Of course, these equations can generally be continued for a window of length q (i.e., up to e (k +q)), which was written here briefly up to q=2.

Remark: A prerequisite for designing the observer is that the system, defined by Eqs. (10) and (11) should be completely observable. As a result, the system observability matrix, defined as shown in the following equation, must be a full rank matrix.

$$
\emptyset_o = \begin{bmatrix} C \\ CA \\ CA^2 \\ . \\ . \\ . \\ CA^{n-1} \end{bmatrix}
\tag{18}
$$

The matrix H must be chosen so that the eigenvalues of the matrix $\widehat{\mathrm{A}} = \mathrm{A} - \mathrm{HC}$ are all stable. Thus, it can be ensured that the estimation error $(\tilde{\mathrm{x}}(\mathrm{k}))$ converges to zero with increasing time (k). Also, if this matrix is selected optimally, the convergence time of the observer may be adjusted to the shortest time. We investigate the effect of the observer pole's location on the convergence time of the observer in the examples presented in section 5.

As a result, the vector $r(k) = [e(k)^T \; e(k+1)^T \; e(k+2)^T]^T$ only includes the effect of system faults. If there is no fault in the system, this vector is zero. By defining the appropriate filter $V = [v_0; \; v_1; \dots v_g]$, the structured residual vector can be calculated as follows

$$rs(k) = [r(k) \;\; r(k-1) \;\; r(k-2)].V \tag{19}$$

Here, g indicates the order of the filter. In this paper, g is equal to q=2. Values of each element of *rs* (*k*) indicate the occurrence of one of the faults in the system.

4. REALIZATION OF FAULT DIAGNOSER BY CTDPN

This section describes the steps for realizing an observer-based diagnostic system - described in the previous section - using a CTDPN.

a) Initially, the observer equations are rewritten in backward form as follows:

$$\hat{x}(k) = (A - HC)\hat{x}(k-1) + Bu(k-1) + Hy(k-1) \tag{20}$$

$$\hat{y}(k) = C\hat{x}(k) = C(A - HC)\hat{x}(k-1) + (CB + D)u(k-1) + CHy(k-1) \tag{21}$$

Also, the residual values for a window of length q=2 are defined as follows:

$$r(k) = y(k-1) - \hat{y}(k-1) \tag{22}$$

$$r_1(k) = r(k-1) \tag{23}$$

$$r_2(k) = \; r_1(k-1) \tag{24}$$

$$rs(k) = \big[\big(y(k-1) - \hat{y}(k-1)\big) \, r(k-1) \, r_1(k-1)\big].V \tag{25}$$

b) For each signal variable in Eqs. (20-25), a place is assigned in the CTDPN. Therefore, the CTDPN marking vector $M(k)$ contains actual and estimated input and output vectors and residual vectors. Thus,

$$M(k) = [u^T(k) \, y^T(k) \, \hat{x}^T(k) \, \hat{y}^T(k) \;\; r^T(k) \;\; r_1^T(k) \, r_2^T(k) \, rs^T(k)]^T. \tag{26}$$

c) For each place, there is only one arc directed to transitions whose weight equals one. In fact, for every i and j, we have $Pre(P_i, T_j) = 1$ and, therefore, for the whole network: $Pre = I_{t\times t}$, where t is the total number of places (transitions).

d) To specify the network, the *Post* matrix must also be determined. According to Eq. (9), the fundamental equation of the CTDPN can be expressed as follows:

$$M(k) = PM(k-1) \tag{27}$$

where

$$P = I + W = I + Post - Pre \tag{28}$$

since $Pre = I$, then $P = Post$. (29)

Finally, Eqs. (20-25) are rewritten according to the definition for M and based on Eq. (29) which identifies the P and, $Post$ matrices, consequently. Accordingly, the *Post* matrix for the system is obtained as follows:

$$P = \begin{bmatrix} 0_{p\times p} & 0_{p\times r} & 0_{p\times n} & 0_{p\times r} & 0_{p\times r} & 0_{p\times r} & 0_{p\times r} & 0_{p\times r} \\ 0_{r\times r} & 0_{r\times r} & 0_{r\times n} & 0_{r\times r} & 0_{r\times r} & 0_{r\times r} & 0_{r\times r} & 0_{r\times r} \\ B & H & \hat{A} & 0_{n\times r} & 0_{n\times r} & 0_{n\times r} & 0_{n\times r} & 0_{n\times r} \\ CB + D & CH & C\hat{A} & 0_{p\times r} & 0_{p\times r} & 0_{p\times r} & 0_{p\times r} & 0_{p\times r} \\ 0_{r\times p} & I_{r\times r} & 0_{r\times n} & -I_{r\times r} & 0_{r\times r} & 0_{r\times r} & 0_{r\times r} & 0_{r\times r} \\ 0_{r\times p} & 0_{r\times r} & 0_{r\times n} & 0_{r\times r} & I_{r\times r} & 0_{r\times r} & 0_{r\times r} & 0_{r\times r} \\ 0_{r\times p} & 0_{r\times r} & 0_{r\times n} & 0_{r\times r} & 0_{r\times r} & I_{r\times r} & 0_{r\times r} & 0_{r\times r} \\ 0_{r\times p} & 0_{r\times r} & 0_{r\times n} & 0_{r\times r} & \alpha I_{r\times r} & \beta I_{r\times r} & \gamma I_{r\times r} & 0_{r\times r} \end{bmatrix} \tag{30}$$

Here, the coefficients a, β, and γ are the filter coefficients V, which has been considered a second-order filter.

5. EXAMPLES

By the examples given in this section, implementing the CTDPN for an observer-based fault diagnoser is illustrated, and its performance is investigated.

5.1. Example 1

In this example a system with discrete-time state equations is considered. The state-space matrices are as follows:

$$A=\begin{bmatrix} 1 & 0 & 0 & -0.0098 \\ -0.0001 & 0.9995 & 0.001 & 0 \\ 0.0001 & 0.001 & 0.999 & 0 \\ 0 & 0 & 0.001 & 0 \end{bmatrix}, B=[0\ \ 0\ -0.0126\ \ 0]^T, C=[1\,0\,0\,0],$$

$$\mathrm{D}=0.$$

The system has one input (p=1), one output (r=1), and four state variables (n=4). Also, two additive state faults $f_1=[0.1\ 0.2\ 0\ 0.5]^T$ and $f_2=[0.5\ \ 0\ \ 0.3\ \ 0.1]^T$ and one additive output fault $f_3=0.5$ are considered.

5.1.1. Observer Designing

At First, the observability matrix of the system (OB) is calculated. This matrix - shown in the following equation - is full column rank, and the system is observable.

$$OB=\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0.9999921 & .19e-05 & -4.89e-06 & -0.00980 \\ 0.9999852 & .39e-05 & -1.95e-05 & -0.01961 \\ 0.9999783 & .59e-05 & -4.40e-05 & -0.02942 \end{bmatrix} \tag{31}$$

Thus, a full order state observer can be designed to generate the residuals. The critical step is to choose the location of the poles. If the selected poles are close to the unit circle, then the convergence time of the observer is longer but more stable. By locating the poles of the observer closer than to the origin, the convergence speed increases; but the convergence error fluctuations also increases. For further investigation, we consider two cases for the location of the observer poles: P_1 = [0.99 0.998 0.997 0.96] is the observer pole vector near the unit circle and P_2 =[0.6 0.991 0.7 0.85] is the observer pole vector near the origin. Fig. 1 shows the output estimation error changes for the two observer pole selection modes.

In this paper, structured residuals are used for fault diagnosis. These residues are obtained by filter V from the estimation error. This filter removes slow changes and the average error value is sensitive to large error fluctuations. As Fig. 2 shows, the structured residual values for the P_1 case are small. In Fig. 3, these residuals are compared for both P_1 and P_2. It can be seen that the residuals for the P_2 mode started at larger values and reach almost zero after about 15 ms.

The structured residual signal should be sensitive to the estimation error induced by faults, and - in the faultless case - it must converge to zero immediately. Therefore, by selecting the observer poles close to the unit circle, these conditions are provided.

By locating the poles of the observer according to P_1, the matrix H is obtained as follows:

$$H = [\,0.0535 \quad -0.1984 \quad -0.1472 \quad -0.0588]^T.$$

The V filter coefficients are also selected as V = [α; β; γ] = [1; -1.945; 0.945], which eliminates the average error values.

In the faultless case, the estimation error converges in less than 200 ms to zero and, the structured residual values are very small. The structured residuals start at low values (less than 0.001) and reach almost zero in less than 200 ms.

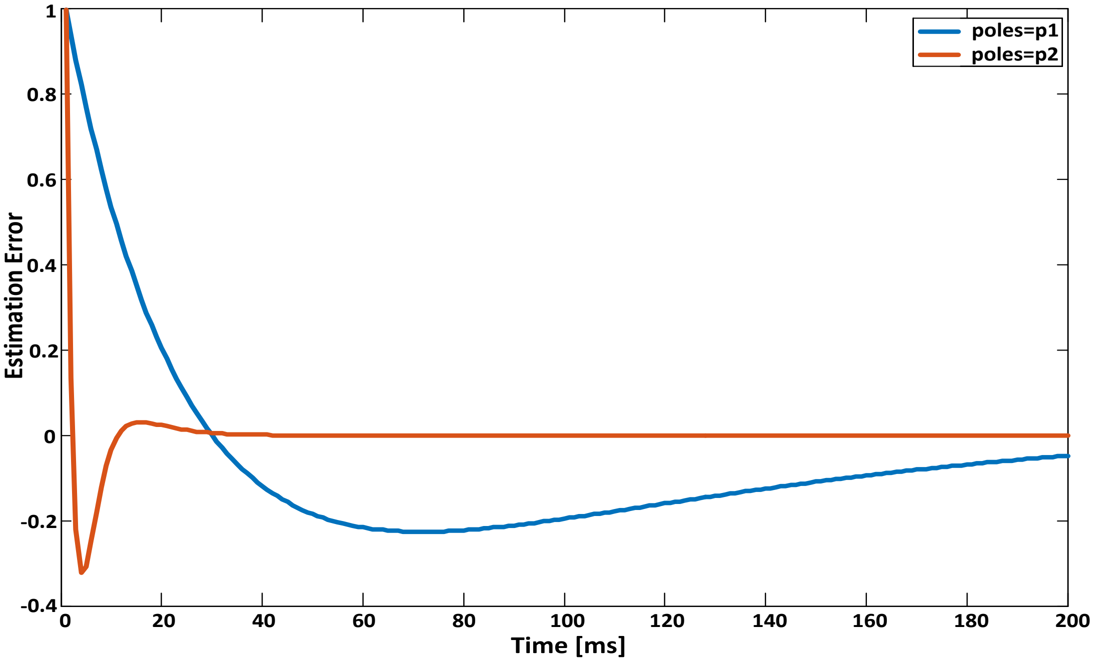

Fig. 1. Output estimation error for the two observer pole selection modes of example 1.

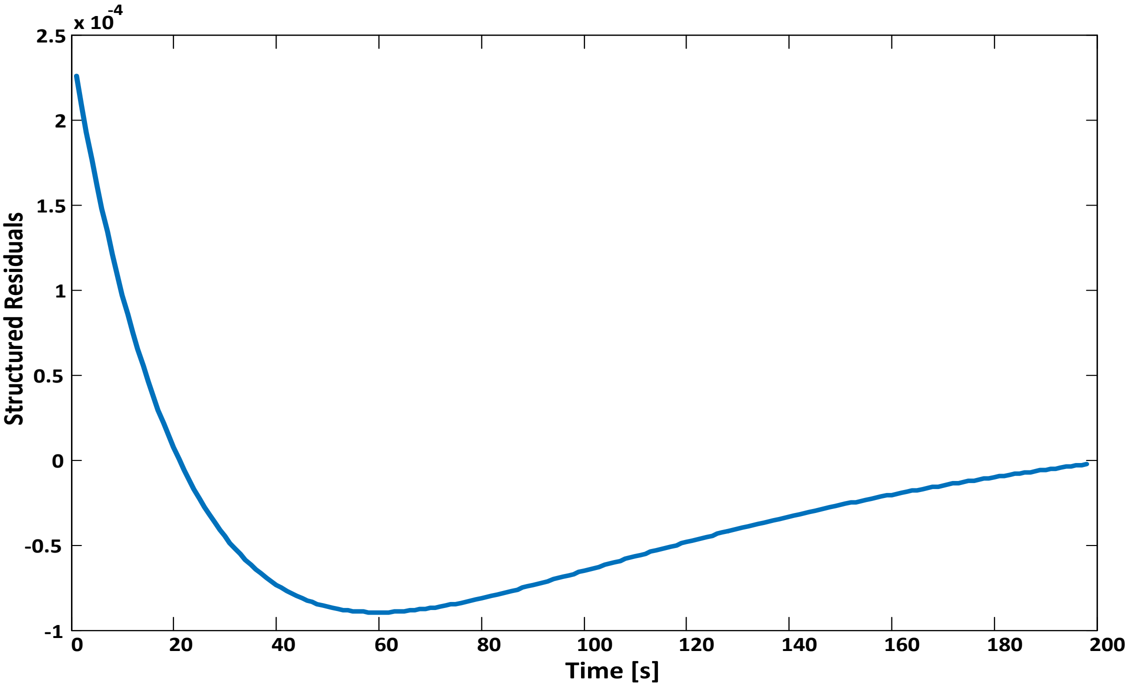

Fig. 2. Structured residual values for the P_1 case of example 1.

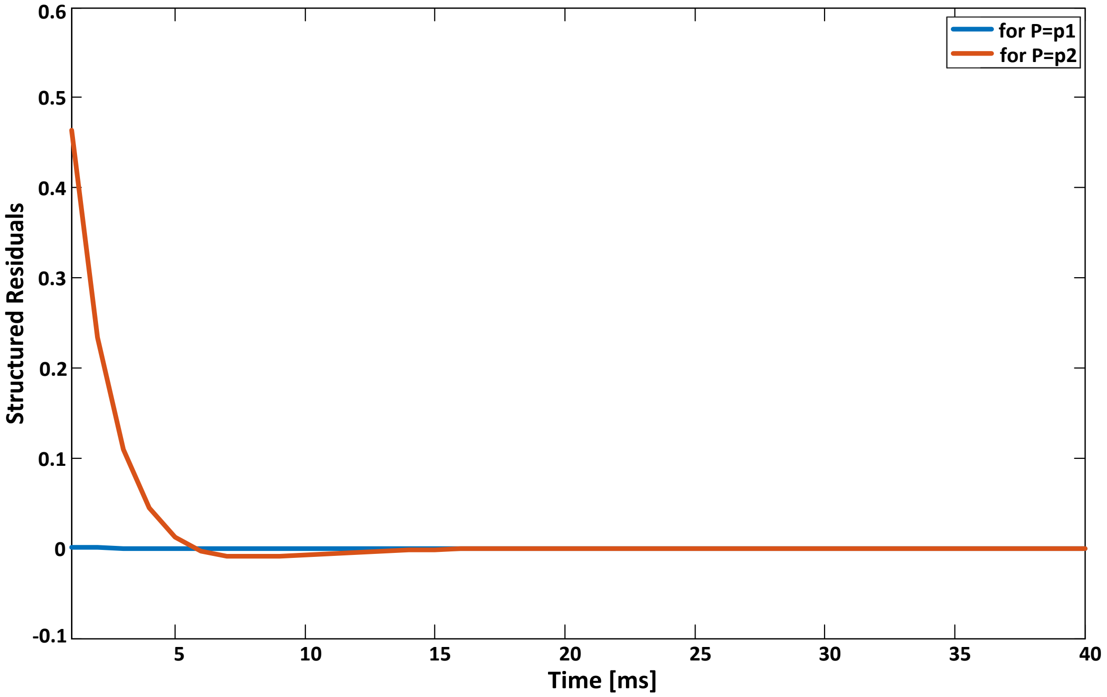

Fig. 3. Structured residuals for the two observer pole selection modes of example 1.

5.1.2. Implementation the Fault Diagnoser by CTDPN

According to Eq. (26), the marking vector of the realizing CTDPN was formed. It contains all needed signal variables to construct the fault diagnoser. In Fig. 4, the desired network is presented for the realization of the fault diagnosis system. Here, one place is provided for each element of the marking vector of Eq. (26). In the first layer of the CTDPN, actual inputs and outputs are given to the network (places p_1 and p_2). As a result, the total number of places is 11. Estimated states are generated by the second layer places (p_3 to p_6) and, the outputs are created by the third layer places (p_7). Places p_8 to p_{10} are used to generate the residue r, and its shift values. The p_{11} place was also used to create the structured residual *rs (k)*. $Pre = I_{11\times 11}$ and $H = Post$ matrix is determined by Eq. (30). In this figure, to clarify the image, only the input arcs to $\widehat{x_1}(k)$ were plotted, and the following abbreviations are also used:

$$s_{1\times 1} = CB + D, e_{1\times 1} = CH, K_{1\times 4} = C\hat{A} = [k_1\ k_2\ k_3\ k_4]$$

In order to evaluate the system performance, the value of the signal *rs (k)* - which represents the structured residual signal and is equal to the mark value of place P_{11} in Fig. 4 - is investigated in the following cases:

5.1.3. Case 1: Faultless System

In this case, the system is faultless. Fig. 2 shows the structured residuals generated by the place P_{11} of the CTDPN fault diagnoser (i.e. $rs(k)$) for the first 200 ms. It is seen that the value of this signal is very little (less than 0.001). Fig. 5 depicts the changes in signal values on a larger scale in the range of 1000 ms. The signal values are nearly zero and indicate that the actual and estimated outputs by the CTDPN are very close. This condition means that there is no fault in the system.

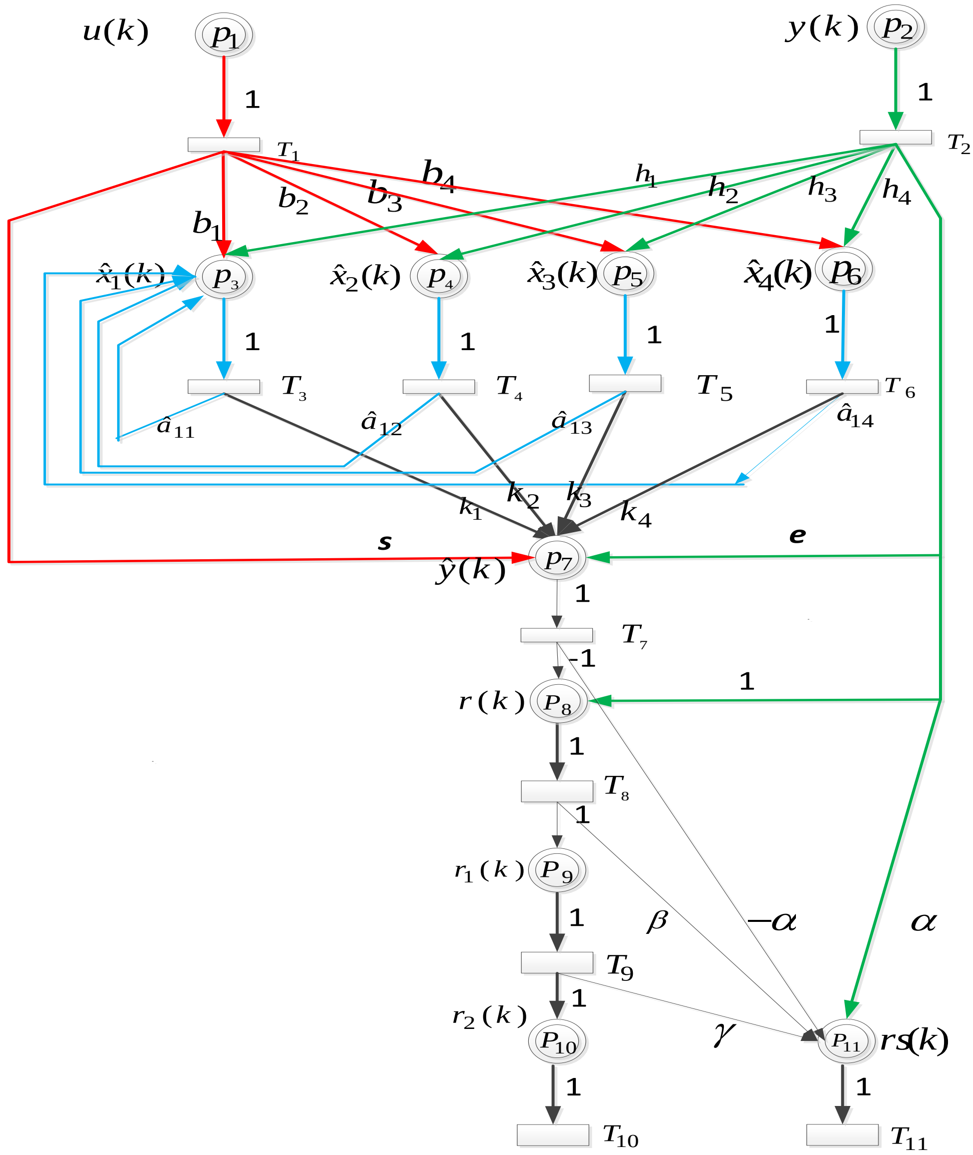

Fig. 4. CTDPN realization of the proposed fault diagnoser of example 1.

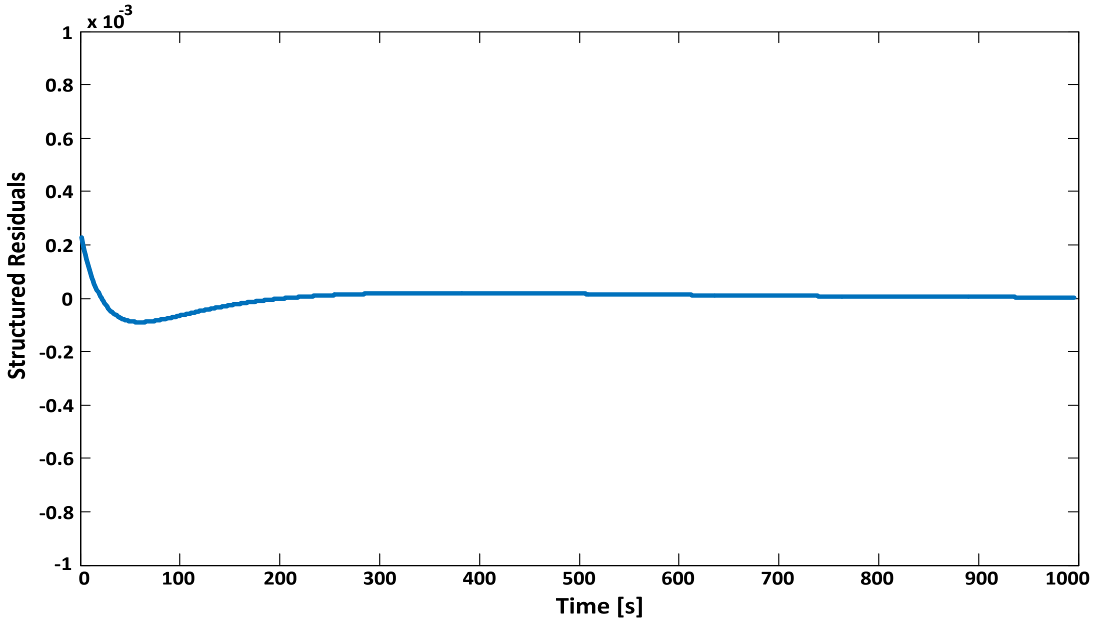

Fig. 5. The residual signal generated by the CTDPN fault diagnoser for the faultless case of example 1.

5.1.4. Case 2: Occurrence of Fault $f_1 = [0.1\ 0.2\ 0\ 0.5]^T$

This case occurs due to the fault $f_1 = [0.1\ 0.2\ 0\ 0.5]^T$, which is an additive fault on the system state variables. It is assumed that the fault occurres at t=400 ms. Fig. 6 shows how the signal changes from the moment that the fault occurs. This signal appears as a fast positive at the moment of fault occurrence. As will be seen in the following cases, the signal generated for each type of fault has a different shape. In addition, the maximum peak of the pulse is proportional to the fault size. Therefore, based on the structured residual signal generated by CTDPN, the type and size of a fault and its occurrence time can be identified.

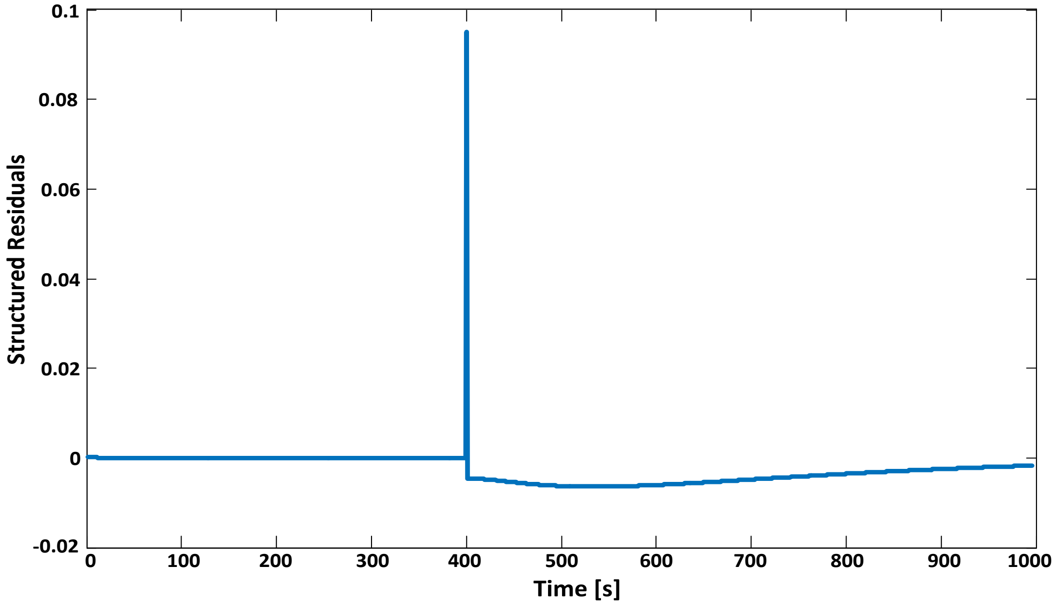

Fig. 6. Residual signal generated by the CTDPN fault diagnoser for case 2 of example 1.

5.1.5. Case 3: Occurrence of Fault $f_2 = [0.5\ \ 0\ \ 0.3\ \ 0.1]^T$

In this case, another additive state fault $f_2 = [0.1\ 0.2\ 0\ 0.5]^T$ is considered to occur at t=200 s. Fig. 7 shows the residual signal generated by the diagnoser.

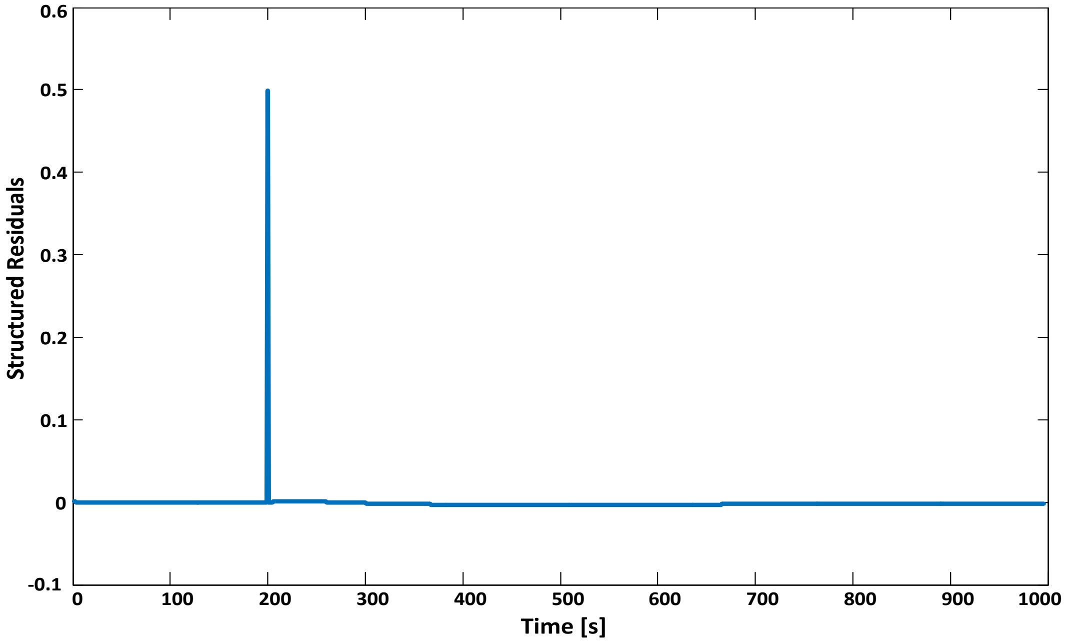

Fig. 7. Residual signal generated by the CTDPN fault diagnoser for case 3 of example 1.

A Comparison between Figs. 6 and 7 shows the differences in shape and size of the generated residuals very well. From this distinction, type of the faults can be determined.

5.1.6. Case 4: Occurrence of Additive Output Fault

In this case, a fault $f_m(k) = 0.5$ is added to the output at the moment t = 300s. This fault can be the result of a constant bias on an output meter. Fig. 8 shows how the generated residual signal differs from the ones in the previous cases. This signal determines the type of fault while specifying the time of the fault occurrence.

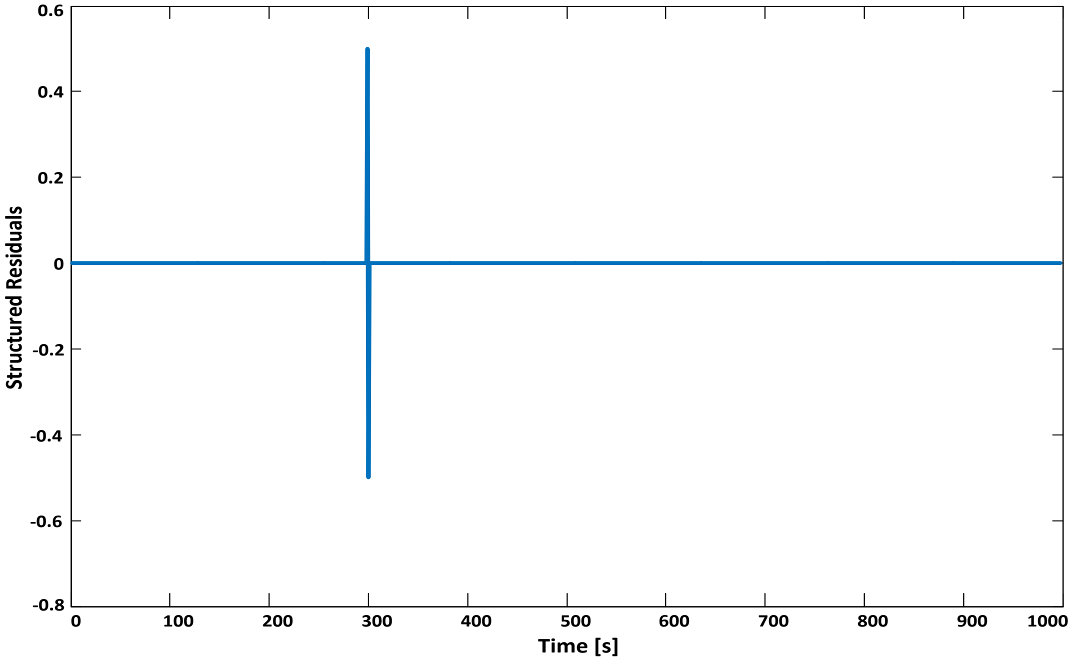

Fig. 8. Residual signal generated by the CTDPN fault diagnoser for case 4 of example 1.

5.2. Example 2

Here, a permanently excited DC motor with a rated power of *P*= 550 W at a rated speed *n*= 2500 rpm is considered [42]. Fig. 9 depicts the signal flow graph of the motor, and its specifications are given in Table 1.

The linear state-space model of this system becomes:

$$\dot{x} = \begin{bmatrix} \dot{I_A} \\ \dot{\omega} \end{bmatrix} = \begin{bmatrix} -\frac{R_A}{L_A} & -\frac{\psi}{L_A} \\ \frac{\psi}{J} & -\frac{M_F}{J} \end{bmatrix} \begin{bmatrix} I_A \\ \omega \end{bmatrix} + \begin{bmatrix} \frac{1}{L_A} & 0 \\ 0 & -\frac{1}{J} \end{bmatrix} \begin{bmatrix} U_A \\ M_L \end{bmatrix}, \tag{32}$$

$$y = \begin{bmatrix} I_A \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x \,. \tag{33}$$

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

$$x(k+1) = \begin{bmatrix} I_A(k+1) \\ \omega(k+1) \end{bmatrix} = \begin{bmatrix} 0.7966 & -0.0433 \\ 0.1538 & 0.9959 \end{bmatrix} \begin{bmatrix} I_A(k) \\ \omega(k) \end{bmatrix} + \begin{bmatrix} 0.1313 & 0.0117 \\ 0.0117 & -0.5201 \end{bmatrix} \begin{bmatrix} U_A(k) \\ M_L(k) \end{bmatrix}. \tag{34}$$

$$y(k) = \begin{bmatrix} I_A \\ \omega \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} x(k). \tag{35}$$

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 Eqs. (33) and (35), discrete state-space parameters are obtained as follows:

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

By calculating the eigenvalues of matrix A, we can see that $\lambda_1 = 0.8391$ and $\lambda_2 = 0.9534$ and the system is stable.

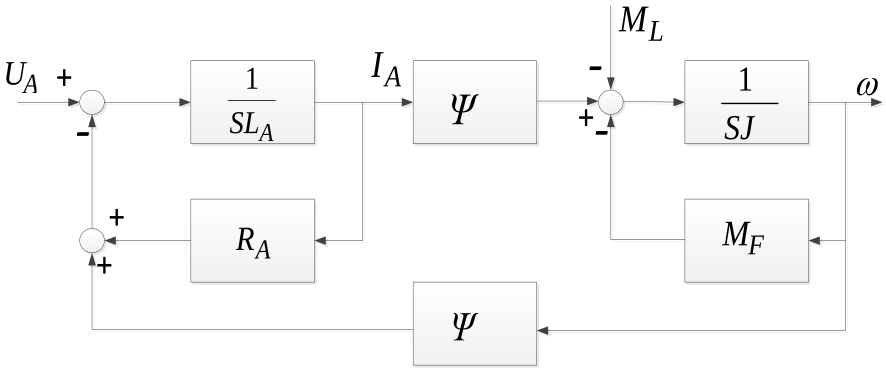

Fig. 9. Signal flow graph of the DC motor.

Table 1. Specifications of the DC motor [35].

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

By calculating $\emptyset_o = \begin{bmatrix}C\\CA\end{bmatrix}$, we can see that the system is completely observable. Thus, a full order state observer is designed to generate the residuals. Due to the system stability, pole placement of the observer is less sensitive than in the previous example. Here, the location of the observer poles is selected as $p_1 = 0.9$ and $p_2 = 0.8$ so that the convergence speed and the amount of oscillation of the estimation error are appropriate. Two faults are considered here: a bias in the armature current sensor (f_1) and a bias in the speed sensor (f_2). In this example, a fault diagnoser based on section 4 is designed for this DC motor. Therefore, two structured residuals (r_1 and r_2) should be made such that when a fault occurs, only its corresponding residual changes and the other residual remains unchanged. Here, $q = 2$ and therefore, we have:

$$Y(k) = [I_A(k-2)\ \omega(k-2)\ I_A(k-1)\ \omega(k-1)\ I_A(k)\ \omega(k)]^T, \tag{36}$$

$$U(k) = [U_A(k-2)\ M_L(k-2)\ U_A(k-1)\ M_L(k-1)\ \ U_A(k)\ M_L(k)]^T \tag{37}$$

$$rs(k) = [rs_1(k)\ rs_2(k)]^T\ . \tag{38}$$

Here, $rs_1(k)$ and $rs_2(k)$ are the structured residuals corresponding to f_1 and f_2, respectively. $M(k)$ is obtained by substituting Eqs. (36-38) in Eq. (26). For generating structured residuals, the weighting matrix V is selected to be the same as the matrix of the previous example. Selecting this weighting matrix causes the residuals to be independent of one another. Finally, the matrices $Pre = I_{16\times16}$ and $post = P$ are obtained from Eq. (30).

Based on Eqs. (26) and (30), this CTDPN is constructed as shown in Fig. 10, in which the places P_1 to P_{16} represent the signals, and the transitions T_1 to T_{16} represent the time delays.

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

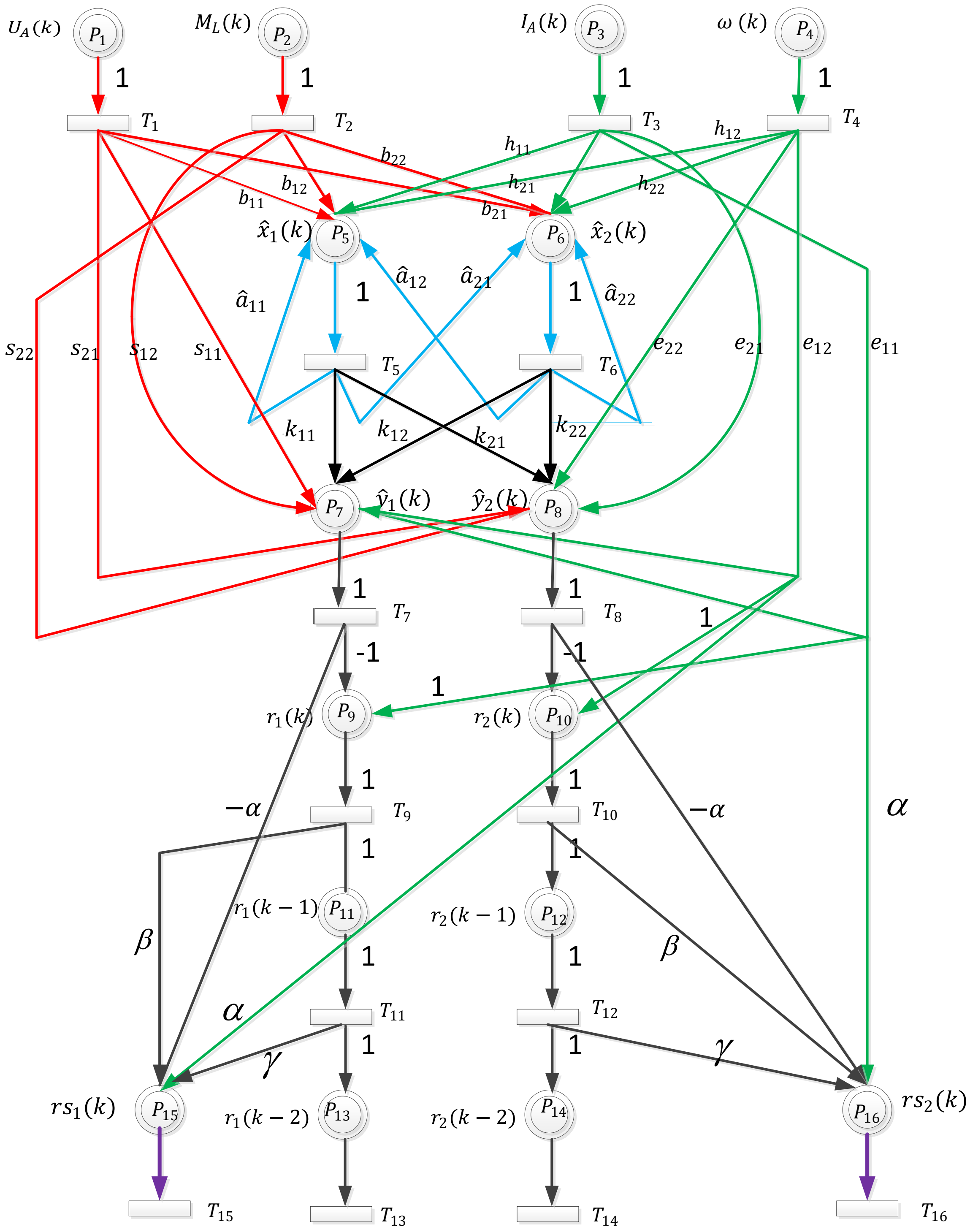

Fig. 10. CTDPN realization of the proposed fault diagnoser of example 2.

5.2.1. *Case 1: Faultless System*

No fault occurs in this case. Fig. 11 shows the residuals generated by the CTDPN fault diagnoser, i.e., $r_1(k)$ and $r_2(k)$. Evidently, the two residual values converge to zero after a transient time (about 0.02 s), indicating no fault occurrence.

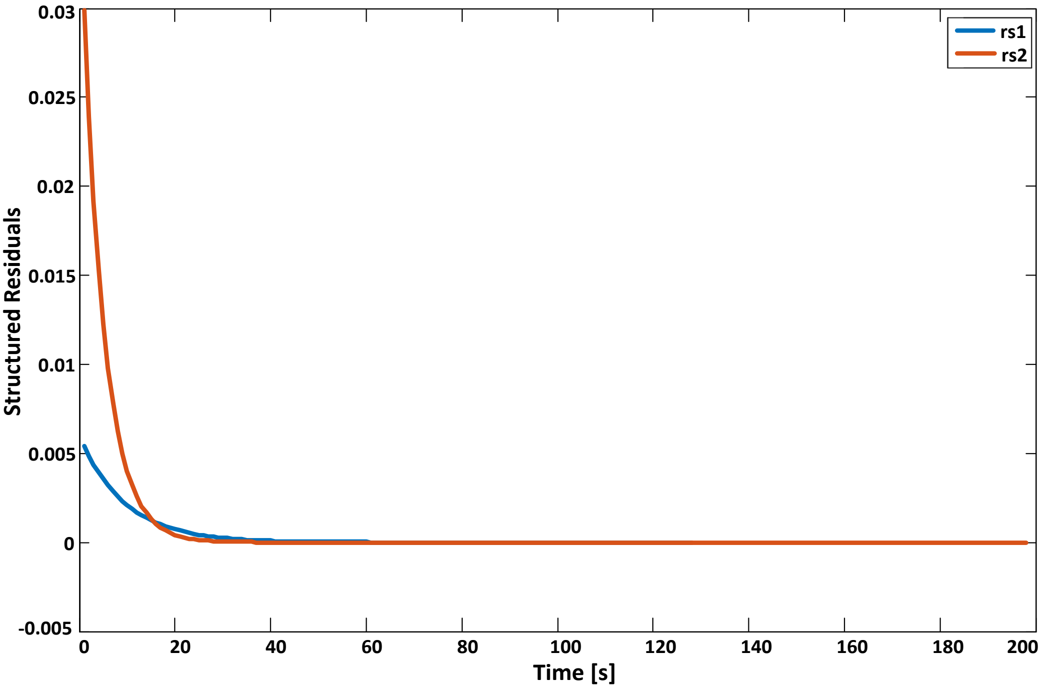

Fig. 11. Residual signals generated by the CTDPN fault diagnoser for the faultless case of example 2.

5.2.2. 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 affects the sensor at t=0.3 s. Fig. 12 shows that $r_1(k)$ is changed at t=0.3 s while $r_2(k)$remains unchanged. Changes in $r_1(k)$ - which reveal the current sensor fault - appear as a rapid positive and negative pulse, while changes in $r_2(k)$ are negligible. Changes in these structured residues indicate that the fault f_1 has occurred at t=0.3 s.

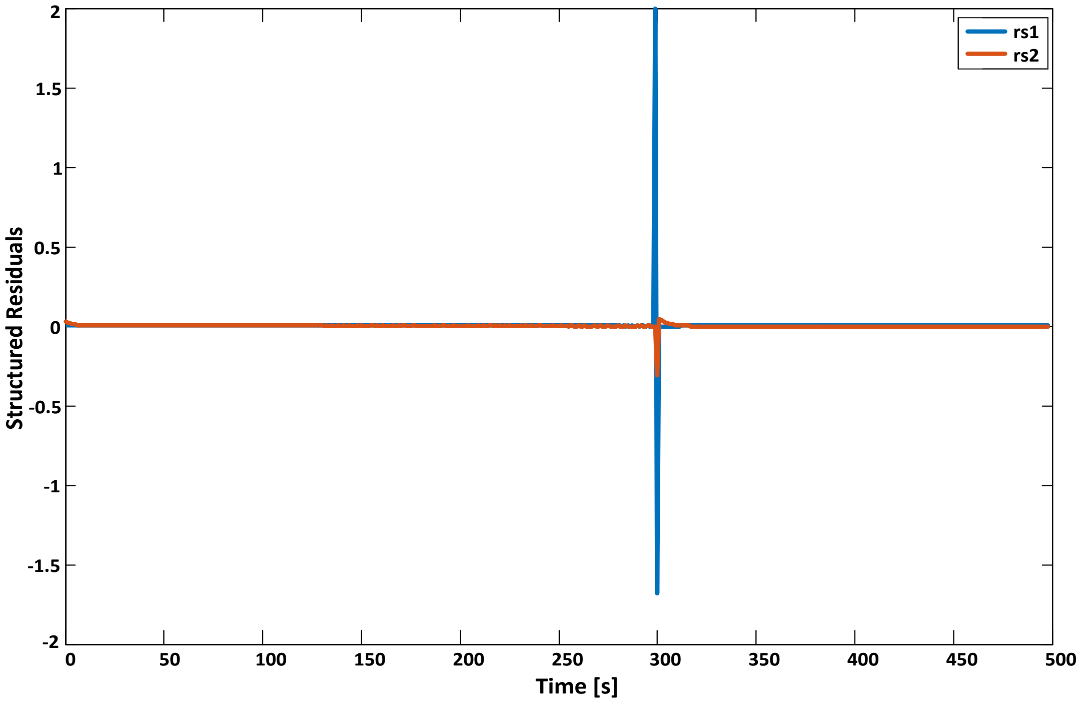

Fig. 12. Residual signals generated by the CTDPN fault diagnoser for case 2 of example 2.

5.2.3. 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. 13. In this figure, $r_1(k)$ remains unchanged, whereas $r_2(k)$ is changed, indicating that f_2 occurred at t=0.2 s.

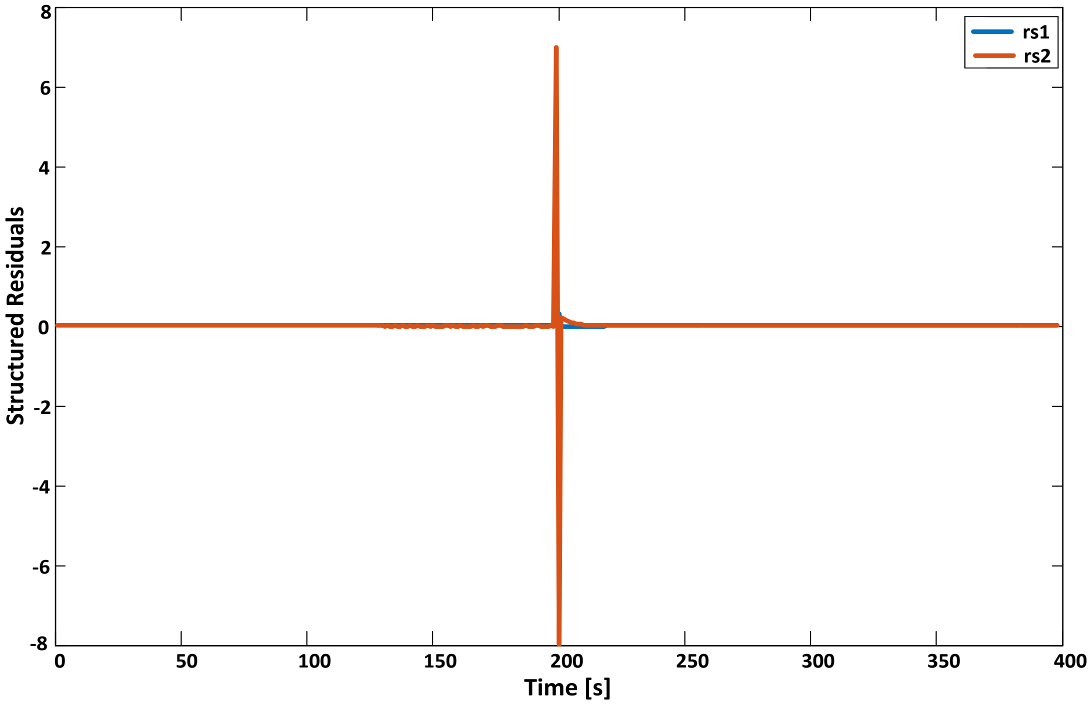

Fig. 13. Residual signals generated by the CTDPN fault diagnoser for case 3 of example2.

6. CONCLUSIONS

This paper presented a new method for realizing a fault diagnosis system for discrete-time linear dynamic systems based on CTDPN. The fault detection method was based on the state observer. As shown in the example, this network can detect and distinguish faults. One of the main advantages of Petri nets is the introduction of an intuitive model of the system. Based on the method proposed in this paper, this advantage becomes available for a classic fault diagnosis system.

In light of the CTDPN realization of the fault diagnosis system, the deadlock analysis for this system may be possible that, in its turn, improves the reliability of the fault detection system. Another related field for future researches is developing this method for hybrid systems; with developing the presented approach in this paper, an integrated fault diagnosis system can be introduced for them.

REFERENCES

[1] Z. Gao, C. Cecati, S. Ding, “A survey of fault diagnosis and fault-tolerant techniques part I: fault diagnosis with model-based and signal-based approaches,” *IEEE Transactions on Industrial Electronics*, vol. 62, no. 6, pp. 3757-3767, 2015.

[2] Z. Gao, C. Cecati, S. Ding, “A survey of fault diagnosis and fault-tolerant techniques part II: fault diagnosis with knowledge-based and hybrid/active approaches,” *IEEE Transactions on Industrial Electronics,* vol. 62, no. 6, pp. 3768-3774, 2015.

[3] S. Yin, B. Xiao, S. Ding, D. Zhou, "A review on recent development of spacecraft attitude fault tolerant control system," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 5, pp. 3311-3320, 2016.

[4] G. Kapoor, "A mathematical morphology-based fault detection and faulty phase categorization scheme for the protection of six-phase transmission line," *Jordan Journal of Electrical Engineering*, vol. 6, no. 1, pp. 35-48, 2020.

[5] S. Gupta, G. Kapoor, J. Rahul, "A DMWT-based relaying scheme for the recognition of shunt faults in a two-terminal series capacitor compensated transmission line," *Jordan Journal of Electrical Engineering*, vol. 6, no. 2, pp. 110-127, 2020.

[6] J. Chen, R. Patton, *Robust Model-Based Fault Diagnosis for Dynamic Systems*, Boston, MA, USA: Kluwer Academic, 1999.

[7] M. Blanke, M. Kinnaert, J. Lunze, M. Staroswiecki, *Diagnosis and Fault Tolerant Control*, Berlin, Germany: Springer, 2006.

[8] S. Baniardalani, J. Askari, "Fault diagnosis of timed discrete event systems using dioid algebra," *International Journal of Control, Automation and Systems*, vol. 11, no. 6, pp. 1095-1105, 2013.

[9] R. Isermann, *Fault-Diagnosis Applications*, Springer-Verlag, Berlin Heidelberg, 2011.

[10] M. Mouchaweh, *Fault Diagnosis of Hybrid Dynamic and Complex Systems*, Springer International Publishing AG Part of Springer Nature, 2018.

[11] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Transactions on Systems, Man, and Cybernetics—PART A: Systems and Humans*, vol. 35, no. 6, pp. 1225-1239, 2005.

[12] C. Petri, *Kommunikation mit Automaten (Communication with Automata),* PhD Thesis, Bonn: Institut Für Instrumentelle Mathematik Schriften des II MNr. 2, 1962.

[13] A. Giua, M. Silva, "Petri nets and automatic control: a historical perspective," *Annual Reviews in Control*, vol. 45, pp. 223-239, 2018.

[14] E. Villani, P. Miyagi, R. Valette, *Modeling and Analysis of Hybrid Supervisory Systems: A Petri Net Approach*, Springer Verlag, Advances in Industrial Control Series, 2007.

[15] X. Zhang, S. Yue, X. Zha, "Method of power grid fault diagnosis using intuitionistic fuzzy Petri nets," *IET Generation, Transmission and Distribution*, vol. 12, no. 2, pp. 295-302, 2018.

[16] A. Vieira, E. Santos, M. De Queiroz, A. Leal, A. De Paula, J. Cury, "A method for PLC implementation of supervisory control of discrete event systems," *IEEE Transactions on Control System Technologies*, vol. 25, no. 1, pp. 175-191, 2017.

[17] A. Farahani, A. Dideban, "Continuous-time delay-Petri nets as a new tool to design state space controller," *Information Technology and Control/Informacinės technologijos ir valdymas*, vol. 45, no. 4, pp. 401-411, 2016.

[18] A. Dideban, A. Farahani, M. Razavi, "Modeling of continuous systems using modified Petri net model," *Modeling and Simulation in Electrical and Electronics*, vol. 1, no. 2, pp. 19-23, 2015.

[19] S. Baniardalani, "Modeling of discrete-time systems using Petri nets," *27th Iranian Conference on Electrical Engineering*, pp. 1188-1192, 2019.

[20] S. Baniardalani, "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, pp. 81-90, 2020.

[21] R. David, H. Alla, *Discrete, Continuous, and Hybrid Petri Nets*, Springer-Verlag, Berlin Heidelberg, 2010.

[22] D. Lefebvre, *Diagnosis of Discrete Event Systems with Petri Nets*: *Petri Net, Theory and Applications*, Vienna:I-Tech Education, Australia, 2008.

[23] M. Iordache, P. Antsaklis, "A survey on the supervision of Petri nets," *DES Workshop PN 2005*, Miami, FL, 2005.

[24] N. Ran, H. Su, A. Giua, C. Seatzu, "Codiagnosability analysis of bounded Petri nets," *IEEE Transaction on Automatic Control*, vol. 63, no. 4, pp. 1192 – 1199, 2018.

[25] N. Ran, S. Wang, H. Su, C. Wang, "Fault diagnosis for discrete event systems modeled by bounded Petri nets," *Asian Journal of Control*, vol. 19, no. 6, pp. 1–10, 2017.

[26] C. Seatzu, M. Silva, J. Van Schuppen, *Control of Discrete-Event Systems*, Springer-Verlag, London, 2013.

[27] Y. Wu, C. Hadjicostis, "Algebraic approaches for fault identification in discrete-event systems," *IEEE Transaction on Automatic Control*, vol. 50, no. 12, pp. 2048-2053, 2005.

[28] V. Calderaro, C. Hadjicostis, A. Piccolo, P. Siano, "Failure identification in smart grids based on Petri net modeling," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 10, pp. 4613-4623, 2011.

[29] V. Calderaro, V. Gladi, A. Piccolo, P. Siano, "Protection system monitoring in electric networks with embedded generation using Petri nets," *International Journal of Emerging Electric Power Systems,* vol. 9, no. 6, 2008.

[30] G. Zhu, Z. Li, N. Wu, A. Al-Ahmari, "Fault identification of discrete event systems modeled by Petri nets with unobservable transitions," *IEEE Tansactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 2, pp. 333-345, 2017.

[31] G. Cavone, M. Dotoli, C. Seatzu, "A survey on Petri net models for freight logistics and transportation systems," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 6, pp. 1795-1813, 2018.

[32] M. Drighiciu, "Hybrid Petri nets a framework for hybrid systems modeling," *2017 International Conference on Electromechanical and Power Systems*, Lasi, Romania, pp. 20-25, 2017.

[33] G. Russo, M. Pennisi, R. Boscarino, F. Pappalardo, "Continuous Petri nets and microRNA analysis in melanoma," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 15, no. 5, pp. 1492-1499, 2018.

[34] F. Zhao, X. Koutsoukos, H. Haussecker, J. Reich, P. Cheung, "Monitoring and fault diagnosis of hybrid systems," *IEEE Tansactions on Systems, Man, and Cybernetics —PART A: Systems and Humans*, vol. 35, no. 6, pp. 1225-1239, 2005.

[35] R. Casas-Carrillo, O. Begovich, J. Ruiz-Le´on, S. Cˇ 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, pp. 1-7, 2016.

[36] J. Fraustro, J. Ruiz-Le´on, C. V´azquez, A. Trevino, "Structural fault diagnosis in timed continuous Petri nets," *Proceedings of the 13th International Workshop on Discrete Event Systems*, Xi'an, China, pp. 159-164, 2016.

[37] J. Zaytoon, B. Riera, "Synthesis and implementation of logic controllers – a review," *Annual Reviews in Control*, vol. 43, pp. 152-168, 2017.

[38] W. Bolton, *Programmable Logic Controllers*, Sixth Edition, Elsevier, 2015.

[39] F. Cabral, M. Moreira, O. Diene, J. Basilio, "A Petri net diagnoser for discrete event systems modeled by finite state automata," *IEEE Transactions on Automatic Control*, vol. 60, no. 1, pp. 59 - 71, 2015.

[40] J. Quezada, J. Medina, E. Flores, J. Tuoh, A. Solís, V. Quezada, "Simulation and validation of diagram ladder—Petri nets," *The International Journal of Advanced Manufacturing Technology*, vol. 88, no. 5–8, pp. 1393–1405, 2017.

[41] P. Nazemzadeh, A. Dideban, M. Zareiee, "Fault modeling in discrete event systems using Petri nets," *ACM Transactions on Embedded Computing Systems*,vol. 12, no. 1, pp. 1-19, 2013.

[42] R. Isermann, *Fault-Diagnosis Systems*, Springer-Verlag, Berlin Heidelberg, 2006.