Preprints of the 21st IFAC World Congress (Virtual)
Berlin, Germany, July 12-17, 2020

A timed model for discrete event system identification and fault detection [⋆]

Ryan P. C. de Souza [*] **Marcos V. Moreira** [*] **Jean-Jacques Lesage** [**]

[*] *COPPE-Electrical Engineering Program, Federal University of Rio de Janeiro, Cidade Universitária, Ilha do Fundão, Rio de Janeiro, 21.945-970, RJ, Brazil (e-mail: {ryanpitanga,moreira.mv}@poli.ufrj.br).*
[**] *LURPA, ENS Paris-Saclay, Univ. Paris-Sud, Université Paris-Saclay, 94235 Cachan, France (e-mail: jean-jacques.lesage@ens-paris-saclay.fr).*

Abstract: We present in this paper a timed discrete event model for system identification with the aim of fault detection, called Timed Automaton with Outputs and Conditional Transitions (TAOCT). The TAOCT is an extension of a recent untimed model proposed in the literature, called Deterministic Automaton with Outputs and Conditional Transitions (DAOCT). Differently from the DAOCT, where only the logical behavior of the discrete event system is considered, the TAOCT takes into account information about the time that the events are observed, and, for this reason, it can be used for the detection of faults that cannot be detected by using untimed models, such as faults that lead the fault detector to deadlocks. The TAOCT represents the fault-free system behavior, and a fault is detected when the observed behavior is different from the behavior predicted by the model, considering both logical and timing information. A practical example is used to illustrate the results of the paper.

Keywords: Discrete-event systems, System identification, Fault detection, Finite-state automata.

1. INTRODUCTION

In the past years, interest in the domain of fault diagnosis of discrete-event systems (DES) has increased. Since the introduction of the concept of diagnosability of discrete-event systems in Sampath et al. (1995), several methods have been proposed in the literature for fault diagnosis of DES (Debouk et al., 2000; Moreira et al., 2011; Zaytoon and Lafortune, 2013; Cabral et al., 2015). These methods provide a theoretical framework for the study of fault diagnosis of DES. However, their application to complex real systems, such as industrial plants, is a difficult task since these methods rely on an accurate model of the system, including its post-fault behavior.

In general, the modeling of a real-world system is very laborious and time consuming, since, depending on its size and complexity, it is very difficult to take into account all possible behaviors of the system in the model. This problem is amplified when we consider the post-fault behavior, since there may exist unpredicted consequences to a fault occurrence. In addition, the modeling process requires engineers that know the plant behavior, and are familiar with discrete-event modeling techniques. All these problems restrict the application of methods based on the complete system model to small systems, where the fault-free and faulty behaviors can be completely known.

In order to circumvent the aforementioned problems, fault detection techniques based on an identified fault-free model of the system have been proposed in the literature (Roth et al., 2011; Klein et al., 2005; Moreira and Lesage, 2019a). In these works, the two main ideas are: (i) to automate the process of obtaining the fault-free model of the system by using identification; and (ii) when a fault has been detected through a discrepancy between the system behavior and the model, to use a technique based on residuals for fault localization.

A model for the identification of closed-loop industrial DES, called Non-Deterministic Autonomous Automaton with Outputs (NDAAO), is presented in Klein et al. (2005). The identification procedure is based on the acquisition of binary signals exchanged between the programmable logic controller (PLC) and the plant. These signals correspond to sensor readings (inputs to the controller) and actuator commands (outputs of the controller), as shown in Fig. 1. Using the recorded data, an automaton which is capable of simulating the observed behavior is constructed. In Klein et al. (2005), with a view to obtaining a compact model, loops are introduced in the identified model, leading to the generation of sequences that have not been observed during the identification process. These sequences form the exceeding language generated by the identified model, and can be associated to non-detectable faults, since they are included in the fault-free model.

In order to reduce the exceeding language while keeping the compactness of the identified model, in Moreira and Lesage (2019a) a new automaton model, called Deterministic Automaton with Outputs and Conditional Transitions (DAOCT), is proposed. Conditions for the transposition of the transitions associated with the observed paths are added to the model by using a path estimation function. The use of the path estimation function reduces the exceeding language generated by the model in comparison with the NDAAO, and, consequently, it reduces the number of non-detectable faults. In Moreira and Lesage (2019b), an algorithm for fault diagnosis using the DAOCT model proposed in Moreira and Lesage (2019a) is presented.

[⋆] This study was financed in part by CNPq, FAPERJ, and the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

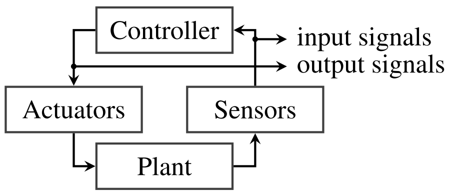

Fig. 1. Closed-loop system showing the signals exchanged between plant and controller.

It is important to remark that some faults cannot be detected by using untimed models. For instance, faults that prevent the system from generating events that are expected by the fault detector, lead it to a deadlock that makes the fault non-detectable. Thus, in some cases, it is necessary to add timing information in order to improve the capability of detecting faults. In Schneider et al. (2012), the NDAAO model is extended to consider timing information, and timing constraints in the form of guards are added to each transition. Each transition has a guard that is defined as a single timing interval, and the guard pre-condition is satisfied when the event occurs in this interval. This model, called Timed Autonomous Automaton with Outputs (TAAO), is capable of detecting faults if the system deadlocks.

In this paper we introduce a new timed model for DES identification with the aim of fault detection. The new model, called Timed Automaton with Outputs and Conditional Transitions (TAOCT), is an extension of the DAOCT model in which timing information is added to the transitions in the form of guards for the transposition of the transitions. Since the TAOCT is based on the DAOCT, it inherits the same advantages of the underlying DAOCT model in terms of exceeding language reduction in comparison with the TAAO model, which is based on the NDAAO. In addition, differently from the TAAO model, the procedure to define the guards of the transitions of the TAOCT is capable of distinguishing different timing intervals associated with the same transition. Thus, more accurate guards can be defined, increasing the number of detectable faults in comparison with the TAAO model that considers a single timing interval for each transition. In addition, the timing information can also be used to refine the path estimate carried out by the path estimation function of the underlying DAOCT, reducing the number of non-detectable faults. It is important to remark that, as in the TAAO, deadlocks caused by faults can also be detected by using the TAOCT.

This paper is structured as follows. In Section 2, we present some preliminary concepts. Then, in Section 3, we explain the identification procedure. The TAOCT is defined in Section 4. A practical example is presented and discussed in Section 5. Finally, conclusions are drawn in Section 6.

2. PRELIMINARIES

Let G be a Timed Automaton with Timing Intervals (TATI) defined as follows (Cassandras and Lafortune, 2008).

Definition 1. A timed automaton with timing intervals is a six-tuple:

$$G = (X, \Sigma, f, c_g, guard, x_0),$$

where X is the set of states, Σ is the finite set of events, $f : X \times \Sigma^* \to X$ is the transition function, c_g is the global clock with value $c_g(t) \in \mathbb{R}^+$, $t \in \mathbb{R}^+$, $guard : X \times \Sigma \to \mathscr{A}$ is the guard function, where $\mathscr{A}$ is the set of admissible timing intervals for the global clock c_g, and $x_0 \in X$ is the initial state. □

A timed automaton with timing intervals is an automaton to which a global clock c_g is added. Thus, timed sequences are generated by the TATI. Function $guard : X \times \Sigma \to \mathscr{A}$ specifies the timing conditions that need to be satisfied on the global clock for the transition to occur, and $\mathscr{A}$ are the admissible clock constraints defined as timing intervals. In the TATI, the global clock is reset to zero each time an event occurs. Let t' denote the time that a state $x \in X$ is reached in the timed automaton, and let τ denote the time that has elapsed after reaching state x. Then, transition (x, σ, x'), where $x' = f(x, \sigma)$, occurs in the TATI if event σ occurs at time $t' + \tau$, and $\tau \in guard(x, \sigma)$.

The TATI is a simplified version of the timed automaton with guards (Cassandras and Lafortune, 2008). In this paper, a similar formalism is used to add timing information to the proposed identification model.

A timed path of a timed automaton with timing intervals G is a sequence of states, events and time values that can be executed by G. A timed path p can be written in the following form:

$$p = (x_1, \sigma_1, \tau_1, x_2, \sigma_2, \tau_2, ..., x_{l-1}, \sigma_{l-1}, \tau_{l-1}, x_l), \qquad (1)$$

where $x_j \in X$, $j = 1, ..., l$, are the states visited in the path, $\sigma_j \in \Sigma$ are the events that trigger the transition from state x_j to x_{j+1}, $j = 1, ..., l-1$, and $\tau_j \in \mathbb{R}^+$ are the time durations where the system remains in state x_j up to the occurrence of event σ_j, $j = 1, ..., l-1$. The length of a path p, denoted by $|p|$, is equal to the number of vertices in the path, *i.e.*, $|p| = l$.

The timed sequence associated with timed path $p = (x_1, \sigma_1, \tau_1, x_2, \sigma_2, \tau_2, ..., x_{l-1}, \sigma_{l-1}, \tau_{l-1}, x_l)$ is the sequence formed of events σ and time durations τ, $s_t = (\sigma_1, \tau_1)(\sigma_2, \tau_2)\dots(\sigma_{l-1}, \tau_{l-1})$. It is important to remark that, in general, in the literature, timed sequences consider an absolute time that specifies when the event has occurred with respect to the beginning of the sequence execution, and not with respect to the sojourn time of each state of the path. However, the same information is contained in both types of timed sequences, since it is always possible to obtain a timed sequence with respect to the beginning of the process, from the timed sequence considering time durations s_t, and vice-versa.

The set of non-negative integers is denoted by $\mathbb{N}$, and the set formed only with 0 and 1 is denoted by $\mathbb{N}_1 = \{0, 1\}$. The symbol '!' denotes 'is defined'.

3. IDENTIFICATION PROCEDURE

Let us consider a closed-loop industrial DES whose scheme is shown in Fig. 1. Suppose the controller has n_i input signals, which correspond to sensor readings, and n_o output signals, which correspond to actuator commands. Let us define the I/O vector, whose elements are the current status of each one of the controller signals:

$$u(t) := [i_1(t) \;\; i_2(t) \;\; \dots \;\; i_{n_i}(t) \;\; o_1(t) \;\; o_2(t) \;\; \dots \;\; o_{n_o}(t)]^T,$$

where $i_m(t) \in \mathbb{N}_1$, $m \in \{1, ..., n_i\}$, are the values of the input signals at time $t \in \mathbb{R}^+$, and $o_m(t) \in \mathbb{N}_1$, $m \in \{1, ..., n_o\}$, are the values of the output signals at time $t \in \mathbb{R}^+$.

Suppose that at a given time instant t_j, $j \in \mathbb{N}$, the I/O vector becomes $u_j := u(t_j)$. Then, this vector remains unchanged until at least one of the controller signals changes its value at time t_{j+1}, leading to I/O vector $u_{j+1} = u(t_{j+1})$. The events of the identified model are defined as the observed changes in the controller signals. Thus, when the I/O vector changes from u_j to u_{j+1}, we consider that an event σ_j has occurred. The sojourn time τ_j is the time duration during which the I/O vector remained equal to u_j, *i.e.*, $\tau_j = t_{j+1} - t_j$. Thus, when the observed I/O vector changes from u_j to u_{j+1}, we say that transition $(u_j, \sigma_j, \tau_j, u_{j+1})$ has been observed. A finite sequence of such transitions constitutes an observed timed path executed by the system $p = (u_1, \sigma_1, \tau_1, u_2, \sigma_2, \tau_2, ..., u_{l-1}, \sigma_{l-1}, \tau_{l-1}, u_l)$. It is assumed in this paper that the system has a unique initial state, whose corresponding I/O vector is denoted as u_0, and that all observed paths start at the initial state of the system, with the same I/O vector u_0. The initial state may correspond, for example, to the beginning of a production cycle in the case of an industrial process.

The objective of system identification is to compute a model that simulates the observed behavior described by the paths generated by the system. When untimed models are used, the behavior of the system is described only by the sequences of events that it generates. However, in the case of timed models, the identified model must simulate the observed sequences of events and must also be coherent with the time these events occur, *i.e.*, it must be capable of simulating the timed sequence of events of the system. In both cases, it is possible that the identified model generates more sequences than those observed. In this case, there is an exceeding language that is associated with non-detectable faults if their sequences cannot be generated by the original fault-free system. In order to obtain an accurate model for identification, its exceeding language must be reduced.

It is important to remark that, in order to obtain a model by identification of a DES, capable of reproducing all possible behaviors of the system in fault-free operation, it is necessary to observe the paths executed by the system for an arbitrarily long time. However, in a finite time, only part of the paths generated by the system can be observed. This fact has a direct impact on the accuracy of the identified model and, therefore, on fault detection, since an observed behavior that the model is not capable of executing is interpreted as being faulty. If it is a fault-free behavior, then the fault detection system generates a false alarm. Thus, in order to reduce the number of false alarms, as in Moreira and Lesage (2019a), we assume in this paper that the system has been observed for a sufficiently long time such that all untimed paths of length up to a given number $n_0 \in \mathbb{N}$ have been observed. Moreover, we assume that each untimed path has been observed a number of times sufficient to capture the timing information regarding event occurrences.

The set formed of all observed events during the identification procedure is denoted by Σ. The set of all I/O vectors that have been observed is denoted by Ω. Thus, $\Omega \subseteq \mathbb{N}_1^{n_i+n_o}$.

3.1 Time-interval paths

Let us assume that N timed paths (not necessarily distinct) have been observed. Each path is denoted by $p_n = \big(u_{n,1}, \sigma_{n,1}, \tau_{n,1}, u_{n,2}, \sigma_{n,2}, \tau_{n,2}, ..., u_{n,l_n-1}, \sigma_{n,l_n-1}, \tau_{n,l_n-1}, u_{n,l_n}\big)$, where $n \in \{1, ..., N\}$, $u_{n,j} \in \Omega$, for $j = 1, ..., l_n$, and $\sigma_{n,j} \in \Sigma$ and $\tau_{j,n} \in \mathbb{R}^+$, for $j = 1, ..., l_n - 1$.

The set containing all of the N observed paths is denoted by P. Let us partition P as follows: $P = P_1 \dot\cup P_2 \dot\cup ... \dot\cup P_r$, $r \leq N$, where all paths that form P_i share the same logical sequence of states and events, given by a unique untimed path $p_{u_i} = (u_{i,1}, \sigma_{i,1}, u_{i,2}, \sigma_{i,2}, ..., u_{i,l_i-1}, \sigma_{i,l_i-1}, u_{i,l_i})$ for each set P_i, $i = 1, \ldots, r$. As in Moreira and Lesage (2019a), we assume here that none of the untimed paths p_{u_i} has an associated sequence of events $s_i = \sigma_{i,1}\sigma_{i,2}\ldots\sigma_{i,l-1}$ that is a prefix of the sequence of events of another path p_{u_j}, where $i \neq j$.

For each set of paths P_i we build a *time-interval path* p'_i given by:

$$p'_i = \big(u_{i,1}, \sigma_{i,1}, I_{i,1}, ..., u_{i,l_i-1}, \sigma_{i,l_i-1}, I_{i,l_i-1}, u_{i,l_i}\big),$$

where $u_{i,j} = u_{n,j}$, $j = 1, \ldots, l_i$, $\sigma_{i,j} = \sigma_{n,j}$, $j = 1, \ldots, l_i - 1$, for any $n \in \{1, \ldots, N\}$ such that $p_n \in P_i$, and $I_{i,j} \subset \mathbb{R}^+$, for $j = 1, \ldots, l_i - 1$. Note that, in the time-interval path p'_i, instead of timing values $\tau_{i,j}$, sets $I_{i,j} \subset \mathbb{R}^+$ are used. Each set $I_{i,j}$ is built in the following manner: (i) construct set $T_{i,j} = \{\tau_{n,j} : p_n \in P_i\}$; (ii) form a set of clusters $S = \{S_1, \ldots, S_K\}$ from $T_{i,j}$; (iii) form set $I_{i,j} = \bigcup_{h=1}^{K} [\max\{0, \min S_h - \delta\}, \max S_h + \delta]$. Note that an uncertainty $\delta \in \mathbb{R}^+$ is added to form each real interval in $I_{i,j}$ to represent possible errors in measuring the time occurrence of the events, caused, for example, by the scan cycle of the controller. The value of δ must be chosen such that the intervals do not overlap with each other, *i.e.*, the user-specified threshold for defining the separation of the clusters must be greater than $2 \times \delta$.

The cluster function, used in the process of forming the sets $I_{i,j}$, takes as input a finite set of real values, and gives as output a set of subsets S, each one containing the values which are more densely grouped compared with the values in the other subsets of the input set. We say that the values in each subset $S_h \in S$ form a cluster. Clustering methods have been widely discussed in the literature, and there are several methods that can be used to implement the cluster function in the process of obtaining the sets $I_{i,j}$, as it can be seen in Jain et al. (1999). In this paper, the NEAREST NEIGHBORS method (Lu and Fu, 1978) is used. In this method, an element is assigned to a cluster if the minimum distance between this element and the other members of the cluster is smaller than a user-specified threshold.

3.2 Parametric identification approach

In Moreira and Lesage (2019a), inspired by Klein et al. (2005), the parametric identification algorithm allows to obtain a model satisfying an important property called k-completeness that guarantees that, once the value of a free parameter k has been chosen for system identification, all and only all sequences of length 1 to $k+1$ that have been observed are generated by the identified DAOCT. By increasing the value of the free parameter k, the exceeding language generated by the DAOCT reduces, but the size of the model grows. Thus, there is a trade-off to be found between complexity and accuracy of the identified model.

Since the timed model proposed in this paper is based on the DAOCT model presented in Moreira and Lesage (2019a), we use the same strategy proposed in Moreira and Lesage (2019a) to obtain a parameterized model for identification. In order to do so, the same steps for the computation of the DAOCT model are considered here, where the first step is the computation of modified paths p'^{k}_i, according to the free parameter k, as follows:

$$p'^{k}_i = \big(y_{i,1}, \sigma_{i,1}, I_{i,1}, ..., y_{i,l_i-1}, \sigma_{i,l_i-1}, I_{i,l_i-1}, y_{i,l_i}\big),$$

where

$$y_{i,j} = \begin{cases} (u_{i,j-k+1}, ..., u_{i,j}), & \text{if } k \leq j \leq l_i \\ (u_{i,1}, ..., u_{i,j}), & \text{if } j < k \end{cases}.$$

Set Ω^k is defined as $\Omega^k := \bigcup_{i\in R}\{y_{i,j} : j = 1,\dots,l_i\}$. The last element of $y_{i,j}$ is denoted by $y^l_{i,j}$. Thus, $y^l_{i,j} = u_{i,j}$, $j = 1,\dots,l_i$, and $i \in R$.

4. TIMED AUTOMATON WITH OUTPUTS AND CONDITIONAL TRANSITIONS

We present in the sequel the timed model proposed in this paper for the identification of DES. The model is based on Definition 1 of Timed Automaton with Timing Intervals, and on the DAOCT model proposed in Moreira and Lesage (2019a).

Definition 2. The Timed Automaton with Outputs and Conditional Transitions is a ten-tuple:

$$\text{TAOCT} = (X,\Sigma,f,c_g,\Omega,\lambda,R,g,x_0,X_f),$$

where X is the finite set of states, Σ is the finite set of events, $f : X\times\Sigma^* \to X$ is the transition function, c_g is the global clock, with value $c_g(t) \in \mathbb{R}^+$, $t \in \mathbb{R}^+$, $\Omega \subseteq \mathbb{N}_1^{n_i+n_o}$ is the set of outputs, where n_i and n_o denote, respectively, the number of inputs and outputs of the controller, $\lambda : X \to \Omega$ is the state output function, $R = \{1,2,...,r\}$ is the set of path indexes, $g : X\times\Sigma\times R \to \mathscr{C}$ is the guard function, $x_0 \in X$ is the initial state, and $X_f \subseteq X$ is the set of final states. □

The set of admissible constraints $\mathscr{C}$ is formed of all sets $I \subset \mathbb{R}^+$. As in the TATI, presented in Definition 1, in the TAOCT a unique global clock c_g is used, and it is reset every time a transition occurs. Function $g(x,\sigma,i)$ specifies a subset of $\mathbb{R}^+$ to which the clock value $c_g(t)$ must belong so that transition (x,σ,x'), where $x' = f(x,\sigma)$, associated with observed path p'_i, can occur. The output function λ is the same presented in the DAOCT model, and associates an I/O vector with each state $x \in X$ of the model.

Differently from the DAOCT, where a path estimation function is defined in the model, in the TAOCT the path estimation function $\theta : X\times\Sigma\times\mathbb{R}^+ \to 2^R$ can be defined using the guard function g as follows:

$$\theta(x,\sigma,\tau) = \{i \in R : \tau \in g(x,\sigma,i)\}.$$

Function $\theta(x,\sigma,\tau)$ provides the path estimate when the current state is x, and event σ occurs at clock value $c_g(t) = \tau$. If $f(x,\sigma)$ is not defined for a given path j, then $g(x,\sigma,j)$ is not defined, and $j \notin \theta(x,\sigma,\tau)$ for any value of τ. Moreover, if $f(x,\sigma)$ is defined, but the observed time τ does not belong to $g(x,\sigma,i)$, then $i \notin \theta(x,\sigma,\tau)$.

Algorithm 1 presents the procedure for constructing the identified TAOCT model from the modified time-interval paths p'^k_i, for $i = 1,\dots,r$. In order to do so, function $\tilde{\lambda} : X \to \Omega^k$ is defined. The guards and its associated path estimation function are used in the TAOCT evolution rule to reduce the exceeding language generated by the model, reducing the number of non-detectable faults. We present in the sequel the timed language generated by the model. In order to do so, we first define the set of timed events Σ_t of the TAOCT as follows:

$$\Sigma_t := \left\{(\sigma,\tau) \in \Sigma\times\mathbb{R}^+ : (\exists x \in X)[\theta(x,\sigma,\tau) \neq \emptyset]\right\}.$$

Set Σ_t is formed of all pairs (σ,τ), where $\sigma \in \Sigma$ and $\tau \in \mathbb{R}^+$, such that there exist $x \in X$, where $f(x,\sigma)$ is defined, and at least one path p'_i, such that $\tau \in g(x,\sigma,i)$. Let us define Σ_t^* as the set of all possible sequences formed of elements $\sigma_t \in \Sigma_t$, and assume that the empty sequence ε belongs to Σ_t^*.

Function $\psi : \Sigma_t^* \to \Sigma^*$ removes the timing information from a timed sequence in Σ_t^*, obtaining its equivalent untimed sequence, *i.e.*, for any $s_t = (\sigma_1,\tau_1)...(\sigma_n,\tau_n) \in \Sigma_t^*$, then $\psi(s_t) = \sigma_1...\sigma_n \in \Sigma^*$. By convention, $\psi(\varepsilon) := \varepsilon$.

Algorithm 1: TAOCT identification

```
Input: Modified time-interval paths $p'^k_i$, for $i = 1,...,r$
Output: TAOCT = $(X,\Sigma,f,c_g,\Omega,\lambda,R,g,x_0,X_f)$
Create initial state $x_0$ and define $\lambda(x_0) = \tilde{\lambda}(x_0) = y_{1,1}$
$X \leftarrow \{x_0\}$, $\Sigma \leftarrow \emptyset$, $\Omega \leftarrow \{y_{1,1}\}$, $R \leftarrow \{1,...,r\}$, $X_f \leftarrow \emptyset$
Create global clock $c_g$
for $i = 1$ to $r$ do
    for $j = 1$ to $l_i - 1$ do
        Find state $x \in X$ such that $\tilde{\lambda}(x) = y_{i,j}$
        if $\tilde{\lambda}(x) \neq y_{i,j+1}\ \forall x \in X$ then
            Create state $x'$ and define $\tilde{\lambda}(x') = y_{i,j+1}$
            $X \leftarrow X \cup \{x'\}$
            $\Sigma \leftarrow \Sigma \cup \{\sigma_{i,j}\}$
            $\lambda(x') \leftarrow y^l_{i,j+1}$
            $\Omega \leftarrow \Omega \cup \{\lambda(x')\}$
        else
            Find $x' \in X$ such that $\tilde{\lambda}(x') = y_{i,j+1}$
        $f(x,\sigma_{i,j}) \leftarrow x'$
        if $g(x,\sigma_{i,j},i)!$ then
            $g(x,\sigma_{i,j},i) \leftarrow g(x,\sigma_{i,j},i) \cup I_{i,j}$
        else
            $g(x,\sigma_{i,j},i) \leftarrow I_{i,j}$
        if $j = l_i - 1$ then
            $X_f \leftarrow X_f \cup \{x'\}$
```

Let us define, now, recursively, the extended path estimation function $\theta_s : X\times\Sigma_t^* \to 2^R$ as follows: $\theta_s(x,\varepsilon) = R$, and for any sequence $s_t(\sigma,\tau) \in \Sigma_t^*$, where $s_t \in \Sigma_t^*$ and $(\sigma,\tau) \in \Sigma_t$, we have that:

$$\theta_s(x,s_t(\sigma,\tau)) = \begin{cases} \theta_s(x,s_t) \cap \theta(x',\sigma,\tau), \text{ where} \\ \quad x' = f(x,\psi(s_t)), \text{ if } f(x,\psi(s_t)\sigma)! \\ \text{undefined, otherwise.} \end{cases}$$

The timed language generated by the TAOCT model is given by:

$$L_t := \{s_t \in \Sigma_t^* : \theta_s(x_0,s_t) \neq \emptyset\}.$$

Language L_t simulates the timed sequences of the paths used in the identification procedure as stated in the sequel.

Theorem 1. Let $L_{t,Obs}$ denote the language formed of all timed sequences associated with the observed paths p_n, for $n = 1,2,\dots,N$. Then, $L_{t,Obs} \subseteq L_t$. □

Proof. The proof is straightforward from the construction of the TAOCT according to Algorithm 1, and will be omitted due to the lack of space. ■

Since timing information is added to the TAOCT model, the fault detection scheme based on the timed model can detect faults that are not possible to be detected by using the DAOCT model. There are three faulty scenarios for which the fault can be detected by the TAOCT and not by the DAOCT model: (i) faults that lead the system to a deadlock; (ii) faults that cause the occurrence of an event σ in a state x, such that $f(x,\sigma)$ is defined, at a time instant τ that does not belong to any set defined by the guard conditions $g(x,\sigma,i)$, for all $i \in R$; and (iii)

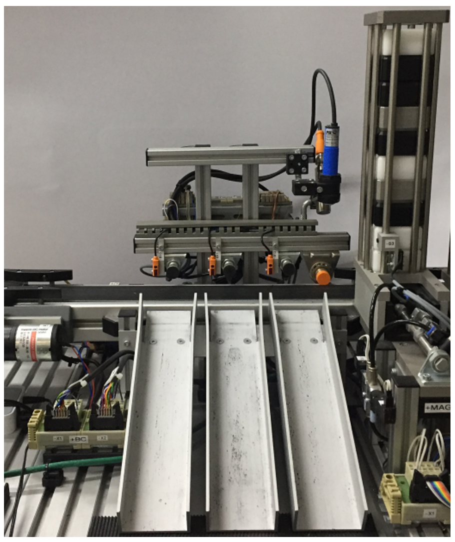

Fig. 2. Sorting unit system of the practical example.

faults that cause the occurrence of an event σ at a time instant τ that satisfies a guard condition, but leads the path estimation function to $\theta_s(x_0, s_t) = \emptyset$. The first type of fault can be detected by counting the time elapsed after reaching a state of the model. If the elapsed time is greater than all possible times defined in the guards, then the deadlock is detected. The second and third types of faults can be detected by verifying if the observed timed sequence can be generated by the model, since, in both cases, it can be seen that, although $f(x_0, \psi(s_t))$ is defined, we have that $\theta_s(x_0, s_t) = \emptyset$, *i.e.*, the observed timed sequence is not in L_t.

5. PRACTICAL EXAMPLE

The identification method proposed in this paper is illustrated using the sorting unit system presented in Figure 2. Three different types of pieces are sorted in the system: white plastic pieces (WP), black plastic pieces (BP), and metallic pieces (M). Each type of piece is pushed to one of the three slides shown on the bottom of Figure 2, such that pieces of type WP are pushed to the right slide, pieces of type M are pushed to the slide in the middle, and pieces of type BP are pushed to the left slide.

On the right of Figure 2, there is a stack magazine where the pieces are stored in any order. In the sorting process, the pieces at the bottom of the stack magazine are placed onto the conveyor belt by a pneumatic pusher. Then, the conveyor belt is turned on, and the piece is moved in the direction of two sensors in order to determine its type. An inductive sensor detects metallic pieces (type M), and an optical sensor detects metallic (type M) and white plastic pieces (type WP). If a black plastic piece (type BP) is on the conveyor, then none of the sensors is capable of detecting it. The optical sensor is located close to the inductive sensor, such that metallic pieces are detected by both sensors almost at the same time.

It is also important to remark that there is a photoelectric sensor next to each sorting pusher on the conveyor. When a piece is detected by the photoelectric sensor next to the pusher that should remove it from the conveyor, the conveyor is stopped and the pusher is extended. Then, the pusher is retracted and a new piece can be placed on the conveyor by the pusher of the stack magazine.

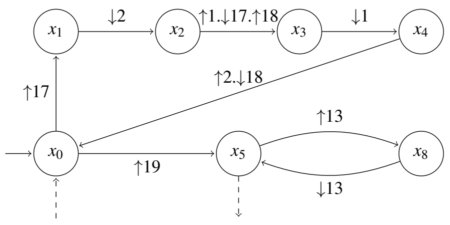

Fig. 3. Part of the TAOCT model obtained for the practical example.

The sorting unit system has 13 sensors and 6 actuator signals. Thus, the controller has 19 input and output signals. The initial state of all observed paths is defined as the I/O vector corresponding to the case where the conveyor belt is turned off, and all pushers are retracted.

During the identification process, 2294 timed paths were observed, which corresponds to two hours and fifty three minutes of observation of the controller signals. From all the observed timed paths, 12 logically different paths were obtained. As there are only three types of pieces, we could expect only three different logically distinguishable sequences. However, it has been observed that, since the inductive and optical sensors are very close to each other, the order of sensor readings (rising and falling edges) may change for different sorting cycles of metallic pieces, increasing the number of paths.

In this practical example, we have chosen the threshold for the clustering algorithm to be 100ms, since this value is coherent with the dynamics of the sorting system. In this case, the cluster function returned a single closed interval for each transition of the time interval paths p_i', for $i = 1, \dots, 12$.

After obtaining the time interval paths p_i', the TAOCT model is identified following the steps of Algorithm 1. In this case, for $k = 1$, the identified TAOCT has 26 states and 38 transitions. In Figure 3, we present only part of the identified TAOCT due to lack of space. Since each transition of p_i' is associated with a single time interval, then all guards $g(x, \sigma, i)$ of the identified TAOCT are single closed intervals. In Figure 3 we do not present the guards due to lack of space. The events of the TAOCT are rising and falling edges of the elements of the I/O controller vector, which are represented by $\uparrow z$ and $\downarrow z$, respectively, where z is the position of the signal in the I/O vector. It is important to remark that an event can be formed of more than one rising or falling edge of controller signals.

In the sequel, we illustrate the three faulty scenarios for which the fault can be detected thanks to the timing information added to the TAOCT model.

In the first scenario, let us consider that, after a piece is placed on the conveyor belt by the pusher of the stack magazine, the pusher stuck extended and cannot be retracted. The path associated with this behavior is $p = (x_0, \uparrow 17, x_1, \downarrow 2, x_2, \uparrow 1. \downarrow 17. \uparrow 18, x_3)$, where 17, 2, 1, and 18 are, respectively, the command to extend the pusher of the stack magazine, the sensor that indicates that the pusher of the stack magazine is retracted, the sensor that indicates that the pusher of the stack magazine is extended, and the command to retract the pusher. In this case, the system deadlocks, since the conveyor belt is turned on only after the retraction of the pusher is detected ($\uparrow 2$). This fault cannot be detected by using the DAOCT model, but can be detected by using the TAOCT model, since the falling edge of the sensor that indicates that the pusher is extended ($\downarrow 1$) must occur before the maximum time of the guard $g(x_3, \downarrow 1, 1) = [87, 161]$. Thus, when the elapsed time is greater than 161 milliseconds, the fault is detected.

To illustrate the second faulty scenario, let us consider a fault in the speed controller of the conveyor that makes it work faster than expected. Let us consider, for instance, that after a BP piece is placed on the conveyor belt, and the conveyor is turned on ($\uparrow 19$), which corresponds to state x_5 of the TAOCT, it reaches the photoelectric sensor next to the first sorting pusher ($\uparrow 13$) in a time smaller than the minimum time of the guard $g(x_5, \uparrow 13, 4) = [4058, 4146]$. Thus, the fault is detected. It is important to remark that, since $\uparrow 13$ is coherent with the logical behavior of the system, then the diagnosis system based on the DAOCT model would not be capable of detecting the fault.

The third faulty scenario can be illustrated by the following example. Consider all observed paths associated with BP or M pieces, and consider that the piece is in front of the photoelectric sensor next to the right sorting pusher, *i.e.*, the rising edge of the photoelectric sensor ($\uparrow 13$) has been observed, which corresponds to state x_8 of Figure 3. By analyzing the time elapsed between $\uparrow 13$ and $\downarrow 13$, two distinct sets of time values can be defined according to the type of piece, as shown in Figure 4. This occurs because metallic pieces are detected for a longer time than plastic pieces by the photoelectric sensor due to their brightness. Let us consider now a fault that causes both the optical and inductive sensors to fail at the same time. In this case, a piece of type M would lead to the same logical behavior than a BP piece, making such a fault non-detectable by the diagnosis system based on the DAOCT model. However, as shown in Figure 4, it is possible to distinguish the types of pieces by using the guards associated with the timed paths. While a BP piece would take some time in the interval $g(x_8, \downarrow 13, 4) = [915, 937]$, a metallic piece would take some value in the interval $[1035, 1052]$ milliseconds, which corresponds to the union of all guards defined in state x_8, for event $\downarrow 13$, and the timed paths associated with metallic pieces. Thus, if a metallic piece is on the conveyor belt, and the fault occurs, the time elapsed between the rising and falling edges of the photoelectric sensor will be compatible with the guard condition associated with metallic pieces, and non-compatible with the guard condition of black plastic pieces. Since the logical behavior is not coherent with the timing behavior, the fault is detected.

6. CONCLUSIONS

In this paper, a new timed model for the identification of DES with the aim of fault detection is proposed. In this model, called Timed Automaton with Outputs and Conditional Transitions (TAOCT), timing information regarding the occurrence of events are added as guards to the transitions. By doing so, a refinement of the path estimation can be carried out using the timing information, and faults that cannot be detected by using untimed models, can be detected by using the TAOCT. We are currently working on a fault detection and isolation algorithm based on the TAOCT.

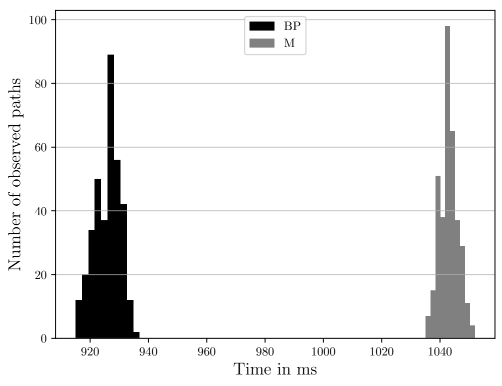

Fig. 4. Histogram showing the observed times elapsed between the rising edge and falling edge of the photoelectric sensor next to the first sorting pusher (BP in black and M in grey) in the observed timed paths.

REFERENCES

Cabral, F.G., Moreira, M.V., Diene, O., and Basilio, J.C. (2015). A Petri net diagnoser for discrete event systems modeled by finite state automata. *IEEE Transactions on Automatic Control*, 60(1), 59–71.

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer Publishing Company.

Debouk, R., Lafortune, S., and Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems*, 10(1-2), 33–86.

Jain, A.K., Murty, M.N., and Flynn, P.J. (1999). Data clustering: A review. *ACM Comput. Surv.*, 31, 264–323.

Klein, S., Litz, L., and Lesage, J.J. (2005). Fault detection of discrete event systems using an identification approach. *IFAC Proceedings Volumes*, 38(1), 92–97. 16th IFAC World Congress.

Lu, S. and Fu, K.S. (1978). A sentence-to-sentence clustering procedure for pattern analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 8(5), 381–389.

Moreira, M.V., Jesus, T.C., and Basilio, J.C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 56(7), 1679–1684.

Moreira, M.V. and Lesage, J.J. (2019a). Discrete event system identification with the aim of fault detection. *Discrete Event Dynamic Systems*, 29(2), 1–19.

Moreira, M.V. and Lesage, J.J. (2019b). Fault diagnosis based on identified discrete-event models. *Control Engineering Practice*, 91, 104101.

Roth, M., Lesage, J.J., and Litz, L. (2011). The concept of residuals for fault localization in discrete event systems. *Control Engineering Practice*, 19(9), 978–988.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995). Diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 40(9), 1555 – 1575.

Schneider, S., Litz, L., and Lesage, J.J. (2012). Determination of timed transitions in identified discrete-event models for fault detection. In *51st IEEE Annual Conference on Decision and Control (CDC'12)*, 5816–5821. Maui, HI, USA.

Zaytoon, J. and Lafortune, S. (2013). Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, 37(2), 308 – 320.