Timed Residuals for Fault Detection and Isolation in Discrete Event Systems
Stefan Schneider, Lothar Litz, Mickaël Danancher

To cite this version:
Stefan Schneider, Lothar Litz, Mickaël Danancher. Timed Residuals for Fault Detection and Isolation in Discrete Event Systems. 3rd International Workshop on Dependable Control of Discrete Systems - DCDS 2011, Jun 2011, Saarbrücken, Germany. pp.35–40, 2011, <10.1109/DCDS.2011.5970315>. <hal-00595107>

HAL Id: hal-00595107
https://hal.archives-ouvertes.fr/hal-00595107
Submitted on 23 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers. L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Abstract—In this paper a new attempt for fault detection and isolation in discrete event systems is proposed. An identified model constitutes a timed observer of the fault-free system behavior. Non-acceptable plant operation is detected by comparing the behavior of the model with the observed system output. For fault isolation, timed residuals and generic fault symptoms – early and late events – are introduced. Time bounds are composed using Boolean conditions and statistical analysis. In case of a fault, timed and untimed residuals are concluded in order to refine a set of potential faulty candidates. The method is applied to the given benchmark system of a virtual production plant with an external controller.

Keywords—Discrete Event System; Timed Automata; Timed Residuals

I. INTRODUCTION

Fault Detection and Isolation (FDI) in industrial systems focuses on the reduction of production downtimes to increase availability. A particular challenge in this field is the development of diagnosis tools for large complex discrete event systems (DES). Several signal and model based approaches for different diagnosis applications have been introduced. Model based concepts perform a comparison of the modeled and the observed system behavior. In case of a deviation a fault is detected and isolated. The applied models can be characterized by two properties. First, models including faulty behavior and models which represent the fault-free behavior can be distinguished. The second property indicates whether the model includes time information or not. One example in literature is the diagnoser structure that models fault-free behavior as well as the behavior for given faults without considering time constraints. This class of models is studied in detail in [1] and an extension to dense-time automata is given in [2]. A Boolean decentralized structure with timed diagnosers is presented in [3] and an approach to timed FDI using fault-free models and template language is proposed in [4].

In this work timed residuals using timed fault-free models are introduced. Previous works provided an identification algorithm [5] to identify a monolithic automaton based on measured system data collected during fault-free system evolutions. An observer structure is used for fault detection purpose. Further developments presented a distributed approach [6] of untimed automata for DES. In order to reproduce concurrent system behavior the global system is divided into subsystems. Partial automata are identified to build a network with additional scalable restrictions on the behavior. A set of untimed residuals was introduced in [7] to perform fault isolation of the fault symptoms unexpected and missed behavior for a monolithic model. This enables a precise isolation of logic faults that occurred in the system. An extension of the fault-free model approach is presented in this paper considering new time based aspects. The fault symptoms early and late events are covered by timed residuals. Both timed and untimed residuals are treated as a compound.

The paper is structured as follows. In section 2 the timed model of a DES is introduced. A formal definition of the timed automaton model is given. The timed identification and composition of partial automata is explained. Section 3 deals with timed FDI. Time related faults are treated in detail including an illustrative example calculation. A case study of the benchmark system is given in section 4.

II. TIMED MODEL OF A DES

A. Problem classification

The observed system is considered as a closed-loop DES with information exchange between plant and controller. Since no knowledge about the control algorithm or plant structure is used the system is treated as a black box. Binary sensor signals are interpreted as controller inputs $I$ and binary actuator signals as controller outputs $O$. Fig. 1 gives a schematic of the input/output (I/O) relation. In the following the controller is assumed fault-free, i.e. the controller software behaves deterministically. The physical system is non-deterministic because of temporal process variations. A DES state represents the combined state of controller and plant. FDI efforts are restricted to sensor and actuator faults of the plant.

![Closed-loop DES](image.png)
B. Timed model definition

To discuss timed fault detection and isolation an appropriate formal model must be defined. According to the timed automaton with guards described in [8] an eight-tuple is introduced which is able to produce the same external behavior as the closed-loop DES with respect to time constraints. It is denoted as Timed Autonomous Automaton with Output (TAAO).

\[ \text{TAAO} = (X, x_0, \Omega, \lambda, C, g, Tra, TG) \]  

where:

- \( X \) : Finite set of states
- \( x_0 \) : Initial state
- \( \Omega \) : Output alphabet
- \( \lambda \) : Output function
- \( C \) : Set of clocks
- \( g \) : Clock mapping function
- \( Tra \) : Set of timed transitions
- \( TG \) : Set of time guards

\( X \) with cardinality \(|X| = n\). Two successive states are denoted as \( x \) and \( x' \), \( x, x' \in X \). \( x, x' \in X \).

\( \Omega \) with cardinality \(|\Omega| = p\). \( \Omega \) will be used to interpret the external behavior of the DES. The explanation is given in a section timed model identification.

\( C \) with cardinality \(|C| = |X|\) contains as many clocks as states.

\[ g \text{ assigns a clock } c \in C \text{ to a state } x \]

where \( g(x) \) addresses the clock \( c \) of state \( x \). The mapping is bijective. A clock interpretation \( f \) is defined as

\[ f : C \to \mathbb{R}^+ \]

where \( f(c) \) represents the time value of clock \( c \).

\( Tra \) is denoted as

\[ Tra \subseteq X \times TG \times C \times X. \]  

An element of the set is interpreted as

\[ (x, tguard(x, x'), g(x'), x') \in Tra. \]

The element \( g(x') \) represents the clock to be reset with this transition. Always the clock of the succeeding state \( x' \) is reset to zero, hence the transition labels are simplified and consist only of the time constraints \( tguard(x, x') \).

\( TG \) contains Boolean conditions expressed as functions of clocks. A time guard \( tguard \in TG \) is denoted as

\[ tguard(x, x') = \left( f \left( g \left( x \right) \right) \geq g(x') \right) \land \left( f \left( g \left( x \right) \right) \leq g(x') \right). \]  

\( t_{\text{min}}, t_{\text{max}} \in \mathbb{R}^+ \) constitute the time bounds of transition from \( x \) to \( x' \). \( |TG| \leq n(n - 1) \).

When \( x \) is activated \( \lambda(x) \) ascertains the output of the TAAO.

Remark:

A distinction is drawn between logical and temporal non-determinism. A TAAO is logical deterministic if all guards are mutually exclusive out of a given state. A TAAO is always temporal non-deterministic since a transition may occur at any time within the defined time bounds.

C. Timed model identification

DES with large number of I/Os are basically able to exhibit a lot of different behavior patterns. Building a model by hand which is able to reproduce all system states is impracticable and usually even impossible. To resolve this difficulty an identification approach is chosen.

For FDI purposes the TAAO has to be identified. It is essential that the DES performs similar repetitive production cycles to obtain an appropriate data base. The data set contains the observed controller input and controller output sequences of the closed-loop DES in Fig. 1. They are called DES output sequences in the following. In this work we assume that the observed DES behavior is fault-free. Initially, the eight-tuple except \( TG \) is identified. The appropriate algorithm is available in [9]. \( \Omega \) is used to accumulate the observed fault-free DES outputs. They are arranged in I/O vectors

\[ u^{\text{DES}}(j) = (I_O(j), \ldots, I_O(j)) \]

with \( j \)-th event step and \( m \) number of controller I/Os, \(|\Omega| \leq 2^m\).

The I/O enumeration convention is declared as follows. \( I_O \) is defined as \( I_O = I, \forall 1 \leq i \leq r \) with controller inputs \( I_1, \ldots, I_r \), and \( I_O^s = O, \forall 1 \leq i \leq s \) with controller outputs \( O_1, \ldots, O_s \) and \( m = r + s \) as defined in [9].

The concept for identification of \( TG \) is presented in the following. Based on all observed I/O vector sequences of the DES the corresponding time sequences are determined. With each new generated I/O vector a time span between two DES states is observed. This time span is called state sojourn time. With the identified TAAO so far and the determined state sojourn times a density interval distribution can be assigned to each transition. It shows how many times a modeled transition is observed within a defined time interval. Fig. 2 illustrates an example distribution of the benchmark system. Each transition in the automata model is related to one distribution. It may be noted that the obtained statistical data can be roughly approximated by a normal distribution. A statistical analysis leads to the determination of lower and upper time bounds for each transition in a generic way, e.g. \( \mu - 3\sigma \) and \( \mu + 3\sigma \) where \( \mu \) denotes the mean value and \( \sigma \) the standard deviation. \( TG \) is based on the identified time bounds with \( t_{\text{min}} \) minimum and
In case the DES generates a new event the model TAAO can reproduce the observed timed behavior of the closed-loop DES.

### Distributed subsystems

For DES with a high degree of concurrency it is advantageous to identify subsystems based on sub-vectors of the I/O vector. A global model is composed of subsystems by grouping related sensor and actuator components. This partitioning is performed either using expert a-priori knowledge or automatically by means of optimization algorithms as treated in [6]. Partitioning is not a subject of this paper. The explanation of the timed residuals is restricted to the monolithic model approach. The extension to distributed models is straightforward.

### III. FAULT DETECTION AND ISOLATION

#### A. Fault detection

FDI is performed based on the identified TAAO. Fig. 3 shows the basic online monitoring concept. The evaluator structure to observe the behavior of the DES is based on the timed system model. It is assumed that the current model state \( x \) is known. In case the DES generates a new event the model tries to reproduce the observed behavior. If the model contains a corresponding solution no deviation between the modeled and observed is concluded and the observed is interpreted as acceptable. If the evaluator is not able to reproduce the observed behavior no succeeding state \( x' \) can be determined based on \( x \) and identified time bounds. Hence, a fault is declared.

The output of a DES is a sequence of timed events. Each event is described by a new I/O vector \( u^{DES} (j) \), where \( u^{DES} (j) \neq u^{DES} (j-1) \). To generate an event at least one I/O must change its value. The evolution of a single I/O can be denoted as single event \( e_{oi} \). Evolution Set \( ES \) contains all single events \( e_{oi} \) between two I/O vectors \( u(j) \) and \( u(k) \)

\[
ES(u(j), u(k)) = \begin{cases} 
\epsilon_{oi} & \text{if } IO_j = 0 \wedge IO_{i} = 1 \\
\epsilon_{oi} & \text{if } IO_{j} = 1 \wedge IO_{i} = 0 \\
\end{cases} \quad \forall 1 \leq i \leq m \text{ for the } i\text{-th vector element.}
\]

IO can change its value either from 1 to 0 denoted with \( \epsilon_{oi} = 0 \) or from 0 to 1 denoted with \( \epsilon_{oi} = 1 \). Since more than one I/O can change its value during an event step, the evolution set can be interpreted as the set of all occurred single events between two I/O vectors.

To accept an observed DES behavior, an active state \( x \) and a timed transition \( \langle x, tguard\rangle \) of \( g(x'), x' \), see (5), must exist which satisfy the logic condition

\[
ES(\lambda(x), \lambda(x')) = ES(\lambda(x), u^{DES}(j))
\]

and the temporal condition

\[
tguard(x, x') = true.
\]

It is assumed that no fault was detected in the \((j-1)\)-th step \( \lambda(x) = u^{DES}(j-1) \). Fault detection is performed using the TAAO as fault-free system model. \( \lambda(x) \) and \( \lambda(x') \) determine the outputs of the model of \( x \) and \( x' \) according to definition (7). Based on the known \( x \) and all possible succeeding model states the logic condition checks whether the resulting ES is equal to the ES of the current model output \( \lambda(x) \) and the new observed DES I/O vector \( u^{DES}(j) \). If this holds for any succeeding state \( x' \) the TAAO can reproduce the observed DES behavior. In the next step the temporal condition has to be checked for all potential state candidates. At the time \( u^{DES}(j) \) is observed the clock value \( f(g(x)) \) of \( x \) must be within the identified time bounds \( t_{MIN}^{x'} \) and \( t_{MAX}^{x'} \) according to the corresponding \( tguard(x, x') \) between \( x \) and \( x' \). If the condition is fulfilled the time guard returns true, otherwise false. A DES behavior is declared as acceptable, if both the logic and the temporal condition are met and an unambiguous succeeding state \( x' \) is determined.

A deadlock is concluded if no new I/O vector \( u^{DES}(j) \) is observed based on \( x \) before the expiration of the maximum possible time bound of all potential transitions from \( x \)

\[
\left( \lambda^{x'}, \lambda^{x'} \right) = \left( t_{MAX}^{x'}, t_{MAX}^{x'} \right).
\]

In this section it is distinguished between logical and temporal misbehavior of the system and the fault detection is explained. If any of the described faults is detected the following fault isolation strategies are applied.
B. Residual isolation

When a fault has been detected the goal is to isolate the fault by determining sensors and actuators which may cause the system to behave in a non-acceptable way. Since these hardware components are directly connected with the controller it is possible to determine faulty candidates by analyzing exclusively the controller I/Os. The residual approach of Roth introduced in [7] is an appropriate way to obtain a small number of I/Os which could be related to an observed logic fault. This work presents an attempt to formalize the deadlock symptom and the extension to timed residuals in order to handle the generic fault symptoms early and late events.

C. Deadlock isolation

The isolation strategy of a deadlock fault is based on the known residual \( Res^4 \) defined in [7]. It is again assumed that the current model state \( x \) is known and no new I/O vector \( u^{DES} (j) \) is observed within the identified maximum possible time. \( Res^4' \) is determined as

\[
Res^4' (x) = \bigcup \{ ES (\lambda (x), \lambda (x')) \}
\]

The set \( ES (\lambda (x), \lambda (x')) \) contains all single events \( e_{\lambda j} \) which would have led to a valid succeeding state \( x' \). The union is applied to cover all possible states \( x' \) with an existing transition from \( x \) to \( x' \). Since no behavior is observed any of the identified model transitions could be missed and hence all related single events could be the reason for the missing observation. \( Res^4' \) denotes a special case of the missed behavior residual with \( ES (\lambda (x), u^{DES} (j)) = \{ \} \). It contains each missing single event which is possibly related to a deadlock fault.

D. Early and late behavior isolation

Faulty components can be isolated by determining behavior which is observed by unexpected or by missed events in a given context. In addition to these logical fault symptoms timed residuals \( TRes^1 \) and \( TRes^2 \) are introduced to deal with early and late events. A behavior which is observed out of time may be related to a faulty component.

The Time Guarded Evolution Set \( TGES \) contains all future and past single events between \( x \) and a succeeding state \( x' \). It represents the modeled behavior which is expected to occur in the future or past with respect to the determined state sojourn time. The \( TGES \) is denoted as

\[
TGES (x, x') = \{ ES (\lambda (x), \lambda (x')) \}
\]

\[
\{ f (g (x)) < \tau_{MIN} \} \cup \{ f (g (x)) > \tau_{MAX} \}
\]

with \( f (g (x)) \) the state sojourn time of state \( x \) when \( u^{DES} (j) \) is observed. Depending on \( f (g (x)) \) a time attribute early or late is assigned to each \( ES \): \( f (g (x)) < \tau_{MIN} \) \( \rightarrow \) early \( f (g (x)) > \tau_{MAX} \) \( \rightarrow \) late

If \( f (g (x)) < \tau_{MIN} \) holds, the actual observation \( u^{DES} (j) \) occurred before the transition from \( x \) to \( x' \) may be taken due to the time bounds. In this case \( TGES \) contains the single events of the modeled fault-free system behavior marked by the label early.

If \( f (g (x)) > \tau_{MAX} \) holds, the actual observation \( u^{DES} (j) \) occurred after the transition from \( x \) to \( x' \) may be taken due to the time bounds. In this case \( TGES \) contains the single events of the modeled fault-free system behavior marked by the label late.

The timed residual specification \( TRes^1 \) represents a set of expected single events \( e_{\lambda j} \) which occurred early and late based on the current active model state \( x \).

\[
TRes^1 (x, u^{DES} (j)) = ES (\lambda (x), u^{DES} (j)) \cap \left( \bigcup_{\{ y' | (x,y') \in TGES (x,x') \}} TGES (x,x') \right)
\]

\( TRes^1 \) is the intersection of the observed DES evolution \( ES (\lambda (x), u^{DES} (j)) \) and all single events which are expected to occur in future or past no matter which following state is taken \( \bigcup_{\{ y' | (x,y') \in TGES (x,x') \}} TGES (x,x') \). The system evolution contains the single events between the output state \( x \) and the observed DES I/O vector \( u^{DES} (j) \) which lead to fault detection. The residual compares the behavior of the model with the actual observed system output. If the current observation of single events is equal to a behavior which should already have occurred (late behavior) or which has not yet been expected (early behavior) the according events are given by the residuals. It is also possible to give a less strict formulation using the union operation. This leads to the notation of the timed residual \( TRes^2 \).

\[
TRes^2 (x, u^{DES} (j)) = ES (\lambda (x), u^{DES} (j)) \cap \left( \bigcup_{\{ y' | (x,y') \in TGES (x,x') \}} TGES (x,x') \right)
\]

with \( TRes^1 \subseteq TRes^2 \). \( TRes^2 \) is usually less restrictive than \( TRes^1 \) since it contains more elements. The results of both timed residuals are two small sets with possible faulty system components. In case of a fault the system operator should check the candidates of \( TRes^1 \). If the fault cannot be found at these components the resulting elements of \( TRes^2 \) should be considered to cover a wider field of potential candidates. In the following the attempt of timed residuals is applied to an illustrative example.

E. Illustrative example

A calculation is shown for the example TAAO in Fig. 4. It is assumed that \( x \) is the current active state of the model and the new observed DES I/O vector is \( u^{DES} (j) = (1,1) \) at time \( f (g (x)) = 5 \). Since \( f (g (x)) \) is smaller than the maximum upper time bound no deadlock occurred. The first step is to check whether the observed behavior can be reproduced by the model or not. Therefore the logical and temporal conditions for both successive states are applied.
The logic condition does not hold for \( x_1 \). The list of potential successive states is thus reduced to \( x_2 \). The temporal condition yields

\[
\text{tguard} (x_1, x_2) = \text{false}
\]

(21)

since \( f(g(x_1)) = 5 \) is not within the time bounds of the considered transition between \( x_1 \) and \( x_2 \), a fault is detected. For fault isolation

\[
ES(\lambda(x), u_{\text{DES}} (j)) = \{e_{\text{IO}} - 1\}
\]

(19)

The timed residuals investigating the expected behavior which occurs early or late represent the timed dimension.

IV. CASE STUDY

The proposed benchmark system (Fig. 6) is the virtual pick and place station of the ITS PLC simulation environment for industrial systems. Running on a PC it is connected with a real programmable logic controller (PLC) via a data acquisition box to build a virtual automated manufacturing system. Multiple virtual sensors S and actuators A are available to control the system. Each component is labeled based on its type and an individual number.

In the following the specification of the system structure and production process for this work is outlined. Conveyor (S2, A0) provides parts and conveyor (S3, A1) empty boxes to the corresponding pick and place station. The gripper is moved horizontally between the two conveyors by two double acting pneumatic cylinders (S4, S5, A2, A3, A4, A5) and vertically using the single acting pneumatic cylinder (S6, S7, A6). Parcels are placed inside a box using the magnetic gripper (S8, A7). The detailed description of the benchmark system refers to [10]. A production cycle is specified as the filling of one box with nine parts. Initially, one box and a sequence of parts are transported. Parts are sorted consecutively into the box without consideration about the type of each part. The fully loaded box is then delivered to the exit conveyor belt.

![Example automaton](image)

Figure 4. Example automaton

\[
x_1 : ES(\lambda(x), \lambda(x)) = ES(\lambda(x), u_{\text{DES}} (j))
\]

(19)

\[
x_2 : ES(\lambda(x), \lambda(x)) \neq ES(\lambda(x), u_{\text{DES}} (j))
\]

(20)

The proposed benchmark system (Fig. 6) is the virtual pick and place station of the ITS PLC simulation environment for industrial systems. Running on a PC it is connected with a real programmable logic controller (PLC) via a data acquisition box to build a virtual automated manufacturing system. Multiple virtual sensors S and actuators A are available to control the system. Each component is labeled based on its type and an individual number.

In the following the specification of the system structure and production process for this work is outlined. Conveyor (S2, A0) provides parts and conveyor (S3, A1) empty boxes to the corresponding pick and place station. The gripper is moved horizontally between the two conveyors by two double acting pneumatic cylinders (S4, S5, A2, A3, A4, A5) and vertically using the single acting pneumatic cylinder (S6, S7, A6). Parcels are placed inside a box using the magnetic gripper (S8, A7). The detailed description of the benchmark system refers to [10]. A production cycle is specified as the filling of one box with nine parts. Initially, one box and a sequence of parts are transported. Parts are sorted consecutively into the box without consideration about the type of each part. The fully loaded box is then delivered to the exit conveyor belt.

![Fault isolation dimensions](image)

Figure 5. Fault isolation dimensions
The simulated system enables introducing a variety of faults into the production system. In the following the investigated faults are related to the vertical operating cylinder exclusively. Three different faults are simulated:

**Fault #1:** After sorting the first parcel into the box, the extended vertical cylinder pulls back. Its arrival in the upper position is indicated by sensor S6. The sensor value is supposed to switch from zero to one as soon as the cylinder is completely contracted. It is assumed that sensor S6 is faulty. It is forced to switch early before the cylinder reaches its initial position.

**Fault #2:** The situation is the same as described with fault #1. S6 is assumed to be faulty again. In this case the sensor is forced to switch late from zero to one.

**Fault #3:** A stuck open fault of actuator A7 after the first parcel has been sorted represents to the third fault case. The event of the cylinder extension is not observed within the maximum possible time bound. Therefore Res4' is calculated to determine the single events which may be related to the fault. One can recognize that the faulty component A7 is represented in the residual set by the $e_{a7}$. By applying the method of timed residuals all faulty components are isolated accurately.

V. CONCLUSIONS

Timed residuals and timed fault detection is presented as a new attempt for FDI in DES. A timed automaton model denoted as TAAO is introduced. Fault detection is performed based on identified logic and timed conditions. The generic fault symptoms deadlock as well as early and late behavior are treated to isolate a small set of potential faulty candidates. The ability of the method is demonstrated by means of the given benchmark system.

### REFERENCES


