Analysis and control of timed event graphs in (max,+) algebra for the active localization of time failures

Ibis Velasquez, Yannick Pencolé, Euriell Le Corronc

Analysis and control of timed event graphs in (max,+) algebra for the active localization of time failures

Ibis Velasquez, Yannick Pencolé, Euriell Le Corronc

LAAS-CNRS, Université de Toulouse, CNRS, INSA, UPS. Toulouse, France

16 January 2024

Abstract

This paper addresses the problem of active diagnosis in Timed Event Graphs for the localization of time failures. Active diagnosis is the process of controlling the system in order to refine a previous diagnosis. A first algorithm is proposed which sets up a multi-input control policy that ensures that the system's observable response is informative enough to identify the source of the delay more precisely, with an analysis of the propagation paths through the TEG. A second algorithm extends the first one to improve the performance of the localization by adding a specific method to analyze the effect of circuits when a time failure propagates.

Keywords : Model-based diagnosis, Active diagnosis, Time failure localization, Timed event graph, (Max,+) algebra, Control.

1 Introduction

This paper introduces the active diagnosis problem in Timed Event Graphs. Timed Event Graphs (TEGs) are a subclass of timed Petri nets which can be represented in a (max,+)-algebraic linear system ([1]). They are characterized by the fact that each place has precisely one upstream and one downstream transition and all arcs have weight 1. This paper specifically addresses non-autonomous TEGs that contain environmental inputs and outputs. Such TEGs are well suited to model timed discrete event systems with synchronization and delay phenomena for manufacturing, logistics and transportation systems, digital twins, communication networks, embedded microcontrollers, etc. This formalism has its own theory of control ([17, 8, 22]) and more recently some contributions to failure diagnosis have also been developed ([20, 14, 19, 13]). In particular, in [20] and [13], the first step of a diagnosis task consists in detecting time failures (unexpected delays associated with some places of the TEG) by

measuring the inputs and the outputs of the system and analyzing through the use of *indicators*. Based on this detection, the second task is to localize the place that actually holds the unexpected delay ([14, 13]). As only the inputs and outputs of the TEG can be measured, the localization of the place within the TEG can be very uncertain and the effective result of the diagnosis can be rather poor.

The motivation of this paper is to improve the precision of the diagnosis by applying an active diagnosis method on the system, refining previous, very uncertain diagnosis results. The idea consists in combining the previous control and diagnosis theories to provide a better localization of the time failure. Active diagnosis is indeed the problem of setting up and applying a control policy on the system that ensures that the system's observable response is informative enough to better identify the source of any previously detected malfunctions. Once a failure has been detected at operating time, an active diagnosis session is opened on the system ([5]) to ensure the algorithm has full control to actively perform the diagnosis (offline method) and identify the source of the failure within the system.

The paper is organized as follows. Section 2 details the related work. Section 3 then recalls the theoretical background of timed event graphs and their representation as (max,+)-linear systems. Section 4 introduces the active diagnosis problem investigated in the paper. A first algorithm for the localization of a time failure in TEGs is then fully detailed in Section 5. It relies on the algorithm CAMI (Control Algorithm for Multiple Inputs), which synthesizes a set of controlled inputs that aim at improving the localization of the time failure by suspecting a subset of paths along the TEG. The proposed method is proved to be sound and is generally more precise than the passive localization of [14, 13] by providing a smaller set of candidate places that could hold the time failure. Then, Section 6 proposes an extension of the previous algorithm whose objective is to focus on the discrimination of candidate places involved in the circuits of the TEG, and which improves the localization precision of the first method. By combining a specific analysis of circuits with the previous algorithm, an extended and more optimal version of CAMI is proposed, called CAMIC (Control Algorithm for Multiple Inputs and Circuits). Finally, the global localization algorithm, called ATFLAT (Active Time Failure Localization Algorithm for TEGs) uses the CAMIC control strategy to solve the active localization problem. ATFLAT is proved to be sound and is generally more accurate than the localization algorithm proposed in Section 5. Section 7 finally concludes and gives some perspectives.

2 Related work

Active diagnosis is the process of applying a set of actions, such as tests and controls, on a system in order to monitor its observable responses and then get or refine a fault diagnosis of the system [24].

In the context of discrete event systems, most of the contributions investi-

gate the active diagnosis problem for untimed autonomous discrete event systems. In [21], the system is partially observable and some events are assumed to be controllable so they can be avoided. In this context, the active diagnosis problem consists in designing a controller over the system to ensure that if the system becomes faulty, its behavior provides enough observable information to decide with certainty that the system is faulty. The work of [4] extends this framework of active diagnosis by introducing modalities for actions and states and a new capability for the controller, namely observing that the system is quiescent. The paper [11] then introduces *parameterized active diagnosis* and proposes as solutions, a set of optimal controllers with respect to a given delay for responses. In [25], another framework is proposed where the available set of possible actions over the system is restricted to the activation and deactivation, at operating time, of sensors that record some events of the system. The active diagnosis problem then consists in selecting the best strategy to perform an active acquisition of information by choosing which sensors to activate based on the previous readings of the system. The work of [12] introduces the problem of *pervasive diagnosis* on systems that run production plans. Pervasive diagnosis aims at designing production plans that, once applied on the system, maximize the observable information required for the monitoring of the health of the system's components. In [5], an *active diagnoser* is defined which monitors the behavior of the system online. Based on this diagnoser and its current health estimation, an active diagnosis session can be opened. In such a session, a planning problem can be drawn as finding a conditional plan of admissible actions whose goal is to lead the system into having a diagnosable behavior. For the active diagnoser to always be effective, the underlying system must be *actively diagnosable*. In [3] the active diagnosis problem introduced in [21] is extended to stochastic DES. In this context the objective is to define a policy that leads the system to a set of trajectories where the probability of fault ambiguity is null.

Active diagnosis has also been investigated in continuous and hybrid systems. For instance, the work of [23] designs an active detector over discrete-time stochastic systems, following a closed loop information processing strategy. Its design is formulated as an optimization problem, similar to the optimal stochastic control problem. More recently, [10] proposes an active diagnosis method for incipient faults in such systems. Active diagnosis has also been investigated in switched systems ([26]) based on an event-based diagnoser and a testing procedure. The work of [15] considers the design of an input signal for minimizing the time and energy required to detect and isolate faults in the outputs of a system. Here, the faults are represented by discrete switches between affine models with bounded disturbances and bounded measurement errors. Finally, the active diagnosis problem has also been addressed in hybrid systems. For instance in [2], the active diagnoser that is set up results from the discretization of the underlying system and its observable measurements, and from the computation of an active diagnoser similar to the one proposed in [5]. More recently, such a technique has been applied for the active diagnosis of on-board control procedures in satellites [6].

This paper addresses the active diagnosis problem for the localization of time shift failures in non-autonomous timed event graphs that has been introduced in our seminal work [27]. As opposed to previous work on active diagnosis, here the set of applied actions or policies has to take into account the notion of time and delays in order to properly generate discriminative timed observable responses. The proposed solution relies on the (max,+) framework proposed in [20, 14], which solves the localization problem in a passive way.

3 Timed event graph models of (max,+)-linear systems

3.1 Timed event graphs

The set of Timed Event Graphs (TEGs for short) is a subclass of timed Petri nets in which each place has exactly one upstream and one downstream transition and for which arcs have weight one (see Fig. 1).

Definition 1 (Timed event graph)**.** *A* timed event graph $\mathcal{G}$ *is a 5-tuple*

$$\mathcal{G} = \langle \mathcal{P}, \mathcal{T}, \mathcal{A}, M_0, \mathcal{HT} \rangle$$

such that

- $\mathcal{P}$ *is a finite set of nodes called* places*;*
- $\mathcal{T}$ *is a finite set of nodes called* transitions*;*
- $\mathcal{P} \cap \mathcal{T} = \varnothing$*;*
- $\mathcal{A} \subseteq (\mathcal{P} \times \mathcal{T}) \cup (\mathcal{T} \times \mathcal{P})$ *is the set of arcs so that* $\forall \mathrm{p} \in \mathcal{P}, |\{(\mathrm{p},\mathrm{t}), \mathrm{t} \in \mathcal{T}\} \cap \mathcal{A}| = 1$ *and* $|\{(\mathrm{t},\mathrm{p}), \mathrm{t} \in \mathcal{T}\} \cap \mathcal{A}| = 1$*;*
- $M_0 : \mathcal{P} \to \mathbb{N}$ *is the initial marking;*
- $\mathcal{HT} : \mathcal{P} \to \mathbb{N}$ *is the holding time for a token in each place.*

As for any type of Petri nets, the preset of a node n, denoted pre(n), is the set of nodes

$$\mathrm{pre(n)} = \{\mathrm{n}' \in \mathcal{P} \cup \mathcal{T}, (\mathrm{n}', \mathrm{n}) \in \mathcal{A}\}, \tag{1}$$

and the postset of a node n, denoted post(n), is the set of nodes

$$\mathrm{post(n)} = \{\mathrm{n}' \in \mathcal{P} \cup \mathcal{T}, (\mathrm{n}, \mathrm{n}') \in \mathcal{A}\}. \tag{2}$$

By Definition of a TEG, it follows that $\forall \mathrm{p} \in \mathcal{P}, |\mathrm{pre(p)}| = 1$ and $|\mathrm{post(p)}| = 1$. An *input transition* $\mathrm{t} \in \mathcal{T}$ is a transition such that $\mathrm{pre(t)} = \varnothing$, an *output transition* $\mathrm{t} \in \mathcal{T}$ is a transition such that $\mathrm{post(t)} = \varnothing$. The set of input transitions of TEG $\mathcal{G}$ is denoted $\mathcal{U}$, its set of output transitions is denoted $\mathcal{Y}$ and the set of internal transitions is denoted $\mathcal{X} = \mathcal{T} \setminus (\mathcal{Y} \cup \mathcal{U})$. An input transition is denoted u_i, $i \in \{1, \ldots, |\mathcal{U}|\}$. An output transition is denoted y_i, $i \in \{1, \ldots, |\mathcal{Y}|\}$. An internal transition is denoted x_i, $i \in \{1, \ldots, |\mathcal{X}|\}$. A *synchronization* is a transition t that is directly downstream of two or more different places, i.e. $|\mathrm{pre(t)}| \geq 2$.

Assumption 1 (Non-autonomous TEG)**.** *Through this paper, timed event graphs are not autonomous, i.e.:*

$$\mathcal{U} \neq \varnothing \text{ and } \mathcal{Y} \neq \varnothing.$$

Example 1. *Figure 1 presents such a non-autonomous TEG. It is composed of 17 places* $\mathcal{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_{17}\}$*. The holding time of each place is printed on the top of the place unless the holding time is 0 (i.e.* $\mathcal{HT}(\mathrm{p}_{15}) = 1$ *and* $\mathcal{HT}(\mathrm{p}_1) = 0$*). It is composed of four input transitions* $\mathcal{U} = \{\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3, \mathrm{u}_4\}$*, three output transitions* $\mathcal{Y} = \{\mathrm{y}_1, \mathrm{y}_2, \mathrm{y}_3\}$ *and six internal transitions* $\mathcal{X} = \{\mathrm{x}_1, \ldots, \mathrm{x}_6\}$*. In this example, every internal transition except* x_3 *is a synchronization.*

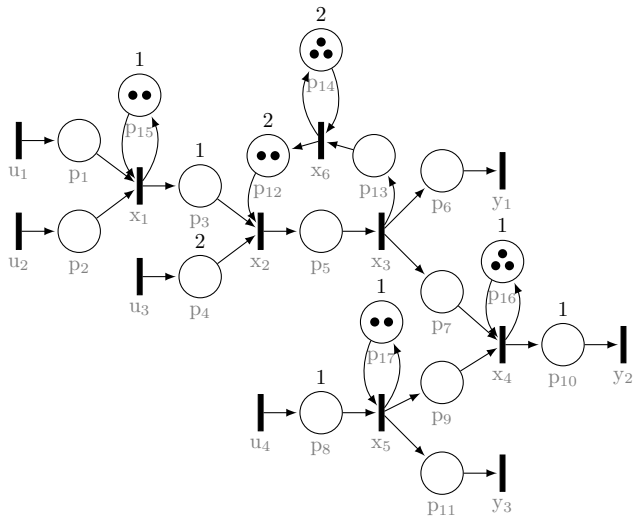

Figure 1: Non-autonomous timed event graph with four input transitions and three output transitions.

A marking M is a function $M : \mathcal{P} \to \mathbb{N}$ that maps each place p to the number of tokens $M(\mathrm{p})$ present in the place. The initial marking M_0 is the marking of the TEG at the initial time.

Example 2. *In Fig. 1, the initial marking* M_0 *is such that* $\forall p \in \{\mathrm{p}_{12}, \mathrm{p}_{15}, \mathrm{p}_{17}\}, M_0(\mathrm{p}) = 2$*,* $\forall p \in \{\mathrm{p}_{14}, \mathrm{p}_{16}\}, M_0(\mathrm{p}) = 3$*, and* $M_0(\mathrm{p}) = 0$ *otherwise.*

A transition $\mathrm{t} \in \mathcal{T}$ is said to be *enabled* in a marking M if and only if $\forall \mathrm{p} \in \mathrm{pre(t)}, M(\mathrm{p}) \geq 1$. Input transitions are always *de facto* enabled ($\mathrm{pre(u)} = \varnothing$). At the time a transition t is fired the current marking M of the TEG is modified to become marking M' defined as follows:

- if $\mathrm{p} \in \mathrm{pre(t)} \setminus \mathrm{post(t)}$ then $M'(\mathrm{p}) = M(\mathrm{p}) - 1$,
- if $\mathrm{p} \in \mathrm{post(t)} \setminus \mathrm{pre(t)}$ then $M'(\mathrm{p}) = M(\mathrm{p}) + 1$,

- otherwise $M'(\mathrm{p}) = M(\mathrm{p})$.

The firing dates of a transition in a TEG rely on the fact that each place $\mathrm{p} \in \mathcal{P}$ is associated with a holding time $\mathcal{HT}(\mathrm{p}) \in \mathbb{N}$ that is the minimal time duration for a token to stay in place p. Two cases hold:

1. if a token $\bullet$ is already in place p at initial time, its associated holding time in place p is $ht(\mathrm{p}, \bullet) = 0$;
2. otherwise the holding time of a token $\bullet$ in place p is $ht(\mathrm{p}, \bullet) = \mathcal{HT}(\mathrm{p})$.

Let t be either an internal or an output transition, the transition is fired at the *earliest date* d_f after it is enabled and the holding time of the tokens in each place of the preset has passed. Formally, suppose $\mathrm{pre(t)} = \{\mathrm{p}_i, i \in \{1, \ldots, |\mathrm{pre(t)}|\}\}$, let $\bullet_i^m$ be the m^{th} token that comes in place p_i, let d_i^m be the date token $\bullet_i^m$ comes in the place p_i ($d_i^m = 0$ if $\bullet_i^m$ is present at the initial time), the m^{th} fire of transition t is therefore at date d_f:

$$d_f = max_{\mathrm{p}_i \in \mathrm{pre}(t)}(\{d_i^m + ht(\mathrm{p}_i, \bullet_i^m)\}). \tag{3}$$

Regarding the input transitions, they represent the occurrence of events from the environment, so they can be fired at any time.

Example 3. *In Fig. 1, only input transitions are enabled in marking M_0. As an example, if transition u_4 fires at time 0 then the new marking M_1 is such that $M_1(\mathrm{p}_8) = 1$ and $M_1(\mathrm{p}_{17}) = 2$ which enables x_5. This latter transition then necessarily fires at time 1 as well as transition y_3. Note that, as $M_0(\mathrm{p}_{17}) = 2$, the holding time of place p_{17} has an influence only after the second fire of x_5.*

Path and Circuits

Definition 2 (Path/Elementary Path)**.** *Let $\mathrm{n}, \mathrm{n}' \in \mathcal{P} \cup \mathcal{T}$ be two nodes of a TEG $\mathcal{G}$, a* path *from node* n *to node* n' *is a sequence of nodes $\pi = \mathrm{n}_1 \ldots \mathrm{n}_k$ such that:*

1. $\mathrm{n} = \mathrm{n}_1$, $\mathrm{n}' = \mathrm{n}_k$, $k \geq 2$;
2. $\forall i \in \{1, \ldots, k-1\}, (\mathrm{n}_i, \mathrm{n}_{i+1}) \in \mathcal{A}$.

Moreover, such a path is elementary *if the following condition also holds*

$$\forall i, j \in \{1, \ldots, k\}, i \neq j \Rightarrow \mathrm{n}_i \neq \mathrm{n}_j.$$

A node n belonging to a path or an elementary path π is denoted by $\mathrm{n} \in \pi$. The set of paths from n to n' is denoted $\Pi(\mathrm{n}, \mathrm{n}')$ and the set of elementary paths from n to n' is denoted $\Pi_{\mathcal{E}}(\mathrm{n}, \mathrm{n}')$. Symbol $\hookrightarrow$ (resp. $\rightsquigarrow$) denotes a binary relation over $(\mathcal{P} \cup \mathcal{T})^2$ such that $\mathrm{n} \hookrightarrow \mathrm{n}'$ (resp. $\mathrm{n} \rightsquigarrow \mathrm{n}'$) means that there is an elementary path (resp. path) from n to n' in $\mathcal{G}$: i.e. $\mathrm{n} \hookrightarrow \mathrm{n}' \Leftrightarrow \Pi_{\mathcal{E}}(\mathrm{n}, \mathrm{n}') \neq \varnothing$ and $\mathrm{n} \rightsquigarrow \mathrm{n}' \Leftrightarrow \Pi(\mathrm{n}, \mathrm{n}') \neq \varnothing$. $\mathcal{P}_{\mathcal{E}}(\pi)$ will denote the set of places involved in the elementary path π:

$$\mathcal{P}_{\mathcal{E}}(\pi) = \{\mathrm{n} \in \pi\} \cap \mathcal{P}. \tag{4}$$

By extension, $\mathrm{n} \hookrightarrow \mathrm{n}'$ denotes any elementary path from $\Pi_{\mathcal{E}}(\mathrm{n}, \mathrm{n}')$ and $\mathrm{n} \rightsquigarrow \mathrm{n}'$ denotes any path from $\Pi(\mathrm{n}, \mathrm{n}')$.

Example 4. *In the TEG of Fig. 1, the sequence* $\mathrm{u_1p_1x_1p_{15}x_1p_3}$ *is a path (*$\mathrm{u_1} \rightsquigarrow \mathrm{p_3}$ *holds) but it is not elementary. An elementary path between* $\mathrm{u_1}$ *and* $\mathrm{p_3}$ *is* $\mathrm{u_1p_1x_1p_3}$ *(*$\mathrm{u_1} \hookrightarrow \mathrm{p_3}$ *holds,* $\mathcal{P}_{\mathcal{E}}(\mathrm{u_1p_1x_1p_3}) = \{\mathrm{p_1}, \mathrm{p_3}\}$*). There is no elementary path (so no path) between places* $\mathrm{p_1}$ *and* $\mathrm{p_2}$ *(*$\mathrm{p_1} \not\rightsquigarrow \mathrm{p_2} \Rightarrow \mathrm{p_1} \not\hookrightarrow \mathrm{p_2}$*).*

A *circuit* σ is a path from a node n to itself ($\sigma \in \Pi(\mathrm{n}, \mathrm{n})$). An *elementary circuit* $\sigma_{\mathcal{E}}$ is a sequence of nodes $\mathrm{n_1} \ldots \mathrm{n}_k$ composed of an elementary path $\mathrm{n_1} \ldots \mathrm{n}_{k-1}$ such that $n_1 = n_k$ and $\mathrm{n}_{k-1} \in \mathrm{pre}(\mathrm{n}_k)$. The set of elementary circuits is therefore:

$$\mathcal{C}_{\mathcal{E}} = \{\sigma\mathrm{n}, \sigma \in \Pi_{\mathcal{E}}(\mathrm{n}, \mathrm{n}'), (\mathrm{n}, \mathrm{n}') \in (\mathcal{P} \cup \mathcal{T})^2 \text{ and } \mathrm{n}' \in \mathrm{pre}(\mathrm{n})\}. \tag{5}$$

Notation $\mathcal{P}_{\mathcal{E}}$ is extended to elementary circuits as follows: $\forall \sigma \in \mathcal{C}_{\mathcal{E}}, \mathcal{P}_{\mathcal{E}}(\sigma) = \{\mathrm{n} \in \sigma\} \cap \mathcal{P}$.

Example 5. *For instance, in Fig. 1, the sequence* $\mathrm{x_2p_5x_3p_{13}x_6p_{14}x_6p_{12}x_2}$ *characterizes a circuit, but it is not elementary. However, circuit* $\mathrm{x_2p_5x_3p_{13}x_6p_{12}x_2}$ *is elementary and* $\mathcal{P}_{\mathcal{E}}(\mathrm{x_2p_5x_3p_{13}x_6p_{12}x_2}) = \{\mathrm{p_5}, \mathrm{p_{13}}, \mathrm{p_{12}}\}$.

Definition 3 (Structural observability [13, 16])**.** *A TEG is* structurally observable *if, from every internal transition* x*, there exists a path to at least one output transition* y*:*

$$\forall \mathrm{x} \in \mathcal{X}, \exists \mathrm{y} \in \mathcal{Y}, \mathrm{x} \rightsquigarrow \mathrm{y}. \tag{6}$$

Assumption 2 (Structural observability)**.** *All along this paper, timed event graphs are structurally observable.*

Structural observability ensures that for every internal transition x, there is at least one output transition y whose sequence of fires may be impacted by x. From a diagnosis viewpoint, if only output transitions are measurable (see Section 4.1), the localization of a time failure in a place in any path from transition x would be irrelevant if structural observability does not hold for x. The TEG represented in Fig. 1 is visibly structurally observable.

3.2 Dioids and residuation theories

3.2.1 Dioids

The dioid theory is the mathematical framework for modeling timed event graphs as (max,+)-linear systems. This section briefly recalls some results, more details can be found in [1].

Definition 4 (Dioid)**.** *A* dioid *is a set* $\mathcal{D}$ *equipped with two binary operations denoted* $\oplus$ *(addition) and* $\otimes$ *(multiplication) such that:*

1. $\oplus$ *is associative and commutative;*
2. $\otimes$ *is associative;*
3. $\otimes$ *is left/right distributive with respect to* $\oplus$ *(*$c \otimes (a \oplus b) = c \otimes a \oplus c \otimes b$ *and* $(a \oplus b) \otimes c = a \otimes c \oplus b \otimes c$*);*
4. $\mathcal{D}$ *has a* zero element ε *(*$\forall a \in \mathcal{D} : a \oplus \varepsilon = a$*);*
5. ε *is an absorbing element for* $\otimes$ *(*$\forall a \in \mathcal{D} : a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon$*);*
6. $\mathcal{D}$ *has an* identity element e *(*$\forall a \in \mathcal{D} : a \otimes e = e \otimes a = a$*);*
7. $\oplus$ *is idempotent (*$\forall a \in \mathcal{D} : a \oplus a = a$*).*

Moreover, if $\otimes$ is commutative ($a \otimes b = b \otimes a$) then the dioid is commutative. A dioid $(\mathcal{D}, \oplus, \otimes)$ is *complete* if it is closed for infinite sums ($\bigoplus_{i=0}^{+\infty} a_i \in \mathcal{D}$ with $a_i \in \mathcal{D}, i \geq 0$) and left/right distributivity of $\otimes$ with respect to $\oplus$ holds for infinite sums ($b \otimes \bigoplus_{i=0}^{+\infty} a_i = \bigoplus_{i=0}^{+\infty} (b \otimes a_i)$, $(\bigoplus_{i=0}^{+\infty} a_i) \otimes b = \bigoplus_{i=0}^{+\infty} (a_i \otimes b)$).

Example 6. *As an example, consider the set* $\overline{\mathbb{Z}}_{max} = \mathbb{Z} \cup \{-\infty, +\infty\}$*, then* $(\overline{\mathbb{Z}}_{max}, \max, +)$ *is a commutative dioid with* $\oplus$ *being the* max *operator and* $\otimes$ *being the classical addition* $+$*. Moreover,* $\varepsilon_{\overline{\mathbb{Z}}_{max}} = -\infty$ *and* $e_{\overline{\mathbb{Z}}_{max}} = 0$*. Another example is the commutative dioid* $(\mathbb{B}, \text{or}, \text{and}) = (\mathbb{B}, \oplus_{\mathbb{B}}, \otimes_{\mathbb{B}})$*, that is the Boolean set* $\mathbb{B} = \{0, 1\}$ *equipped with the logical* or *as the addition* $\oplus$ *and the logical* and *as the multiplication* $\otimes$ *with the zero element* $\varepsilon_{\mathbb{B}} = 0$ *and the identity element* $e_{\mathbb{B}} = 1$*. Both dioids are complete.*

The definition of $\oplus$ induces a partial order $\succeq$ in the dioid $(\mathcal{D}, \oplus, \otimes)$:

$$\forall a, b \in \mathcal{D}, a \succeq b \Leftrightarrow a = a \oplus b. \tag{7}$$

A function $f : \mathcal{D} \to \mathcal{D}$ is *isotone* if $\forall a, b \in \mathcal{D}, a \succeq b \Rightarrow f(a) \succeq f(b)$. In the following, for the sake of simplicity, products like $a \otimes b$ are simply denoted ab. One important result in the dioid theory proposes a way to solve the equation $ax \oplus b = x$. Let $a \in \mathcal{D}$, we denote $a^0 = e$ and $\forall i \in \mathbb{N} \setminus \{0\}, a^i = aa^{i-1}$. The isotone *Kleene star* operator a^* is then defined as $a^* = \bigoplus_{i \geq 0} a^i$.

Theorem 1 (Solution of $ax \oplus b = x$ [1])**.** *Let* $(\mathcal{D}, \oplus, \otimes)$ *be a complete dioid, the least solution of* $ax \oplus b = x$ *is* $x = a^*b$*.*

Finally, note that addition $\oplus$ and multiplication $\otimes$ can be extended to matrices with entries in a dioid $(\mathcal{D}, \oplus, \otimes)$. By extension, for a matrix $M \in \mathcal{D}^{n \times n}$, matrix $M^* = \bigoplus_{i \geq 0} M^i$.

3.2.2 Residuation theory

The mapping $R_a : x \mapsto x \otimes a$ induced by the operator $\otimes$ on a dioid is not invertible. However, a *pseudo-inverse* operator can be defined based on the *residuation theory*, briefly described here below.

Let $f : \mathcal{D} \to \mathcal{C}$ be an isotone mapping between two complete dioids $\mathcal{D}$ and $\mathcal{C}$.

Definition 5 (Residuated mapping)**.** *Mapping f is* residuated *if for every $b \in \mathcal{C}$, there exists a greatest solution for equation $f(x) \preceq b$. This greatest solution is denoted*

$$f^\sharp(b) = \bigoplus_{x \in \mathcal{D}, f(x) \preceq b} x.$$

If mapping f is residuated, then mapping $f^\sharp : \mathcal{C} \to \mathcal{D}$ is called the *residual* of f. Moreover, $f^\sharp$ is the unique isotone mapping such that $f \circ f^\sharp \preceq Id_{\mathcal{C}}$ and $f^\sharp \circ f \succeq Id_{\mathcal{D}}$ where $Id_{\mathcal{C}}$ and $Id_{\mathcal{D}}$ are respectively the identity mappings on $\mathcal{C}$ and $\mathcal{D}$.

Mapping $R_a : \mathcal{D} \to \mathcal{D}$ over a dioid $\mathcal{D}$ is such a residuated mapping and its residual is

$$R_a^\sharp(b) = \bigoplus_{x \in \mathcal{D}, xa \preceq b} x = b \not/ a. \tag{8}$$

Operator $\not/$ means that $b \not/ a$ is the greatest solution to inequality $x \otimes a \preceq b$.

3.3 Dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$

Timed event graphs can be modeled as $(max, +)$-linear systems using the dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. This dioid is defined as a quotient of the set $\mathbb{B}[\![\gamma, \delta]\!]$ of formal power series s in two commutative variables γ and δ with exponents in $\mathbb{Z}$ (series of monomials of type $\gamma^n \delta^t$) and coefficients in $\mathbb{B} = \{\varepsilon_{\mathbb{B}}, e_{\mathbb{B}}\}$ denoted as follows:

$$s = \bigoplus_{(n,t) \in \mathbb{Z}^2} c(n,t) \gamma^n \delta^t \text{ with } c : \mathbb{Z}^2 \to \mathbb{B}. \tag{9}$$

Let $s_i = \bigoplus_{(n,t) \in \mathbb{Z}^2} c_i(n,t) \gamma^n \delta^t \in \mathbb{B}[\![\gamma, \delta]\!]$, $i \in \{1, 2\}$ denote any couple of series, addition $\oplus$ is then defined as:

$$s_1 \oplus s_2 = \bigoplus_{(n,t) \in \mathbb{Z}^2} (c_1(n,t) \oplus_{\mathbb{B}} c_2(n,t)) \gamma^n \delta^t$$

and multiplication $\otimes$ is defined as:

$$s_1 \otimes s_2 = \bigoplus_{(n,t) \in \mathbb{Z}^2} c(n,t) \gamma^n \delta^t \text{ with } c(n,t) = \bigoplus_{\substack{\mathbb{B} \\ n = n_1 + n_2 \\ t = t_1 + t_2}} c_1(n_1, t_1) \otimes_{\mathbb{B}} c_2(n_2, t_2).$$

Endowed with addition $\oplus$ and multiplication $\otimes$, $\mathbb{B}[\![\gamma, \delta]\!]$ is a complete commutative dioid. The zero element of $\mathbb{B}[\![\gamma, \delta]\!]$ is $\varepsilon = \bigoplus_{(n,t) \in \mathbb{Z}^2} \varepsilon_{\mathbb{B}} \gamma^n \delta^t$ and its identity element is $e = e_{\mathbb{B}} \gamma^0 \delta^0 \oplus \bigoplus_{(n,t) \in \mathbb{Z}^2 \setminus \{0,0\})} \varepsilon_{\mathbb{B}} \gamma^n \delta^t$. As $\varepsilon_{\mathbb{B}}$ stands for the absence of a monomial in the series and $e_{\mathbb{B}}$ stands for its presence, series of $\mathbb{B}[\![\gamma, \delta]\!]$ can be simply denoted:

$$s = \bigoplus_{(n,t) \in \mathbb{Z}^2, c(n,t) = e_{\mathbb{B}}} \gamma^n \delta^t \text{ with } c : \mathbb{Z}^2 \to \mathbb{B}. \tag{10}$$

Example 7. *For instance, the identity element of* $\mathbb{B}[\![\gamma,\delta]\!]$ *is denoted* $e=\gamma^0\delta^0$. *Graphically, a series of* $\mathbb{B}[\![\gamma,\delta]\!]$ *characterizes a collection of points of coordinates* $(n,t)\in\mathbb{Z}^2$ *with* γ *as horizontal axis and* δ *as vertical axis. For instance, series* $s_1=e\oplus\gamma^0\delta^1\oplus\gamma^2\delta^3\oplus\gamma^3\delta^2\oplus\gamma^4\delta^5$ *is graphically represented in Fig. 2 as a collection of five points (represented as 3 black squares and 2 black circles).*

Definition 6 (Dioid $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!],\oplus,\otimes$)**.** *Dioid* $(\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!],\oplus,\otimes)$ *is the quotient of dioid* $(\mathbb{B}[\![\gamma,\delta]\!],\oplus,\otimes)$ *induced by the following congruence relation* $\equiv$*:*

$$\forall s_1,s_2\in\mathbb{B}[\![\gamma,\delta]\!],s_1\equiv s_2\Leftrightarrow\gamma^*(\delta^{-1})^*s_1=\gamma^*(\delta^{-1})^*s_2. \tag{11}$$

An element of $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is an equivalent class $[s]_{\gamma^*(\delta^{-1})^*}$ gathering all the series of $\mathbb{B}[\![\gamma,\delta]\!]$ that are equivalent modulo $\gamma^*(\delta^{-1})^*$. The zero element in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is the class $[\varepsilon]_{\gamma^*(\delta^{-1})^*}$ and the identity element in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is the class $[e]_{\gamma^*(\delta^{-1})^*}$. Any series s of $\mathbb{B}[\![\gamma,\delta]\!]$ is a *representation* of class $[s]_{\gamma^*(\delta^{-1})^*}$.

Example 8. *Consider the series* s_1 *from the previous example, it represents the class* $[s_1]_{\gamma^*(\delta^{-1})^*}$. *This class gathers an infinite set of points, some of which are represented by black squares, black circles and gray circles in Fig. 2. This class can also be represented by a set of lines that delimit the set of points that belong to the class of* s_1 *in* $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$.

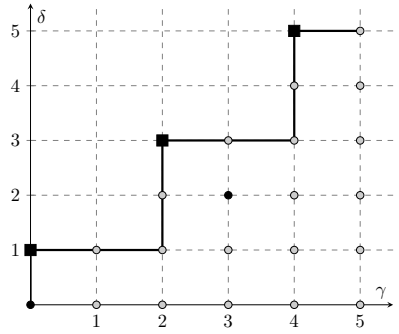

Figure 2: Graphical representation of the class $[s_1]_{\gamma^*\delta^{-1}}\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ whose minimal representation is $\gamma^0\delta^1\oplus\gamma^2\delta^3\oplus\gamma^4\delta^5$.

For the sake of simplicity through the rest of the paper, the expression $s\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ will denote without ambiguity the class in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ which is represented by series s from $\mathbb{B}[\![\gamma,\delta]\!]$. It follows for instance that ε and e represent the zero element and the identity element in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ respectively ($\varepsilon\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!],e\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$). For $s,s'\in\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, the expression $s=s'$ denotes that the class of s is equal to the class of s' in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ (which means $s\equiv s'$ in $\mathbb{B}[\![\gamma,\delta]\!]$). Then, by definition of $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, the following computation

rules hold:

$$\gamma^n\delta^t \oplus \gamma^{n'}\delta^t = \gamma^{min(n,n')}\delta^t \tag{12}$$

$$\gamma^n\delta^t \oplus \gamma^n\delta^{t'} = \gamma^n\delta^{max(t,t')} \tag{13}$$

$$\gamma^n\delta^t \otimes \gamma^{n'}\delta^{t'} = \gamma^{n+n'}\delta^{t+t'} \tag{14}$$

Among the set of representations of a class of $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, there are two particular representations, namely the minimal and the maximal representation.

Definition 7 (Minimal/Maximal representation)**.** *The* minimal representation *of a class of* $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ *is a series:*

$$s_{min} = \bigoplus_{(n,t)\in J_{s_{min}}\subset\mathbb{Z}^2} \gamma^n\delta^t$$

where $J_{s_{min}}$ *is a set such that* $\forall(n,t) \in J_{s_{min}}, \nexists(n',t) \in J_{s_{min}}, n' < n$ *and* $\nexists(n,t') \in J_{s_{min}}, t' > t$*. The corresponding* maximal representation *is:*

$$s_{max} = \bigoplus_{(n,t)\in J_{s_{max}}\subset\mathbb{Z}^2} \gamma^n\delta^t$$

such that $J_{s_{max}} = \{(n,t) \in \mathbb{Z}^2, \exists(n',t),(n,t') \in J_{s_{min}}, n \geq n'$ *and* $t \leq t'\}$.

Example 9. *Back to Fig. 2, the minimal representation is* $s_{1,min} = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5$ *which corresponds to the collection of black squares. The maximal representation then explicitly enumerates the complete collection of points associated with the represented class:* $s_{1,max} = \bigoplus_{(n,t),n\geq 0,t\leq 1} \gamma^n\delta^t \oplus \bigoplus_{(n,t),n\geq 2,t\leq 3} \gamma^n\delta^t \oplus \bigoplus_{(n,t),n\geq 4,t\leq 5} \gamma^n\delta^t$.

Finally, in the following, $\gamma^n\delta^{+\infty}$ denotes the element of $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ whose maximal representation is defined as:

$$\gamma^n\delta^{+\infty} = \bigoplus_{(n',t)\in\mathbb{Z}^2, n'\geq n, t\in\mathbb{Z}} \gamma^n\delta^t \tag{15}$$

3.4 Time shift function

One other interesting representation of a series $s \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ is its dater function $\mathcal{D}_s$.

Definition 8 (Dater function)**.** *Let* $s \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$ *and* s_{max} *be its maximal representation, the dater function of* s *is the non-decreasing function denoted* $\mathcal{D}_s : \mathbb{Z} \to \overline{\mathbb{Z}}$ *such that*

$$\mathcal{D}_s(n) = max(\{t,(n,t) \in J_{s_{max}}\}) \text{ with } s_{max} = \bigoplus_{(n,t)\in J_{s_{max}}\subset\mathbb{Z}^2} \gamma^n\delta^t.$$

Example 10. *Figure 3 graphically represents the dater function* $\mathcal{D}_{s1}$ *of the series* $s_1 = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5$ *(see Fig. 2).*

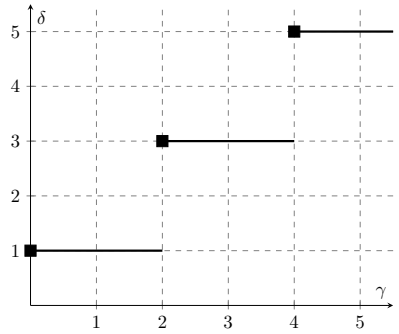

Figure 3: Graphical representation of the dater function of $s_1 = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5$: $\mathcal{D}_{s_1}(0) = 1, \mathcal{D}_{s_1}(1) = 1, \mathcal{D}_{s_1}(2) = 3, ...$

The dater function can be used as a way to compare series.

Definition 9 (Time shift function)**.** *Let $s, s' \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ and let $\mathcal{D}_s$ and $\mathcal{D}_{s'}$ be their respective dater functions. The* time shift function *between s and s' is:*

$$\mathcal{T}_{s,s'} : \mathbb{Z} \to \overline{\mathbb{Z}}$$
$$\mathcal{T}_{s,s'}(n) = \mathcal{D}_{s'}(n) - \mathcal{D}_s(n).$$

$\underline{\mathcal{T}}_{s,s'}$ (resp. $\overline{\mathcal{T}}_{s,s'}$) denotes the lower bound (resp. upper bound) of $\mathcal{T}_{s,s'}$ (Fig. 4):

$$\underline{\mathcal{T}}_{s,s'} = min_{n\in\mathbb{Z}}(\mathcal{T}_{s,s'}(n)), \overline{\mathcal{T}}_{s,s'} = max_{n\in\mathbb{Z}}(\mathcal{T}_{s,s'}(n))$$

As formally detailed in the following theorem, the lower bound and the upper bound of the time shift function $\mathcal{T}_{s,s'}$ can be obtained by computing the value of a dater function on the couple of residuals $s'\oslash s$ and $s\oslash s'$.

Theorem 2 ([16])**.** *Let $s, s' \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$, the time shift function $\mathcal{T}_{s,s'}$ is bounded as follows:*

$$\forall n \in \mathbb{Z}, \underline{\mathcal{T}}_{s,s'} = \mathcal{D}_{s'\oslash s}(0) \leq \mathcal{T}_{s,s'}(n) \leq -\mathcal{D}_{s\oslash s'}(0) = \overline{\mathcal{T}}_{s,s'}. \tag{16}$$

3.5 Timed event graphs as (max,+)-linear systems

The complete dioid $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ aims at modeling timed sequences of events. Indeed, series in $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ are non-decreasing and can be used to represent the accumulation of event occurrences over time. It follows that series from $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ can represent a firing sequence of a given transition in a timed event graph, also called an *event flow* or an *event trajectory*. A monomial $\gamma^n\delta^t$ in such a series is interpreted as the $(n+1)^{th}$ event (i.e. transition fire) that occurs at the earliest time t. A monomial $\gamma^n\delta^{+\infty}$ in the series then represents the fact that the trajectory describes a finite number n of events, as the $(n+1)^{th}$ event never happens.

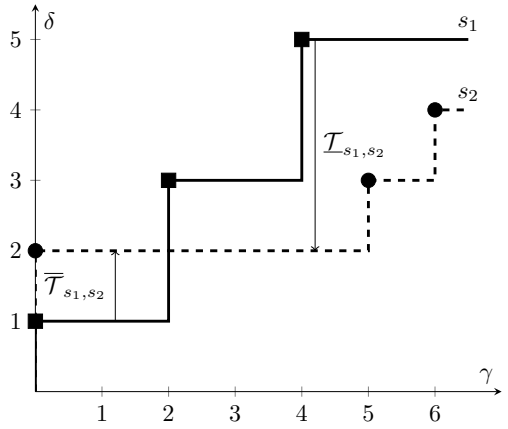

Figure 4: Time shift function between $s_1 = \gamma^0\delta^1 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^5)$ and $s_2 = \gamma^0\delta^2 \oplus \gamma^5\delta^3 \oplus \gamma^6\delta^4$: $\underline{\mathcal{T}}_{s_1,s_2} = \mathcal{D}_{s_2}(4) - \mathcal{D}_{s_1}(4) = 2 - 5 = -3$, $\overline{\mathcal{T}}_{s_1,s_2} = \mathcal{D}_{s_2}(1) - \mathcal{D}_{s_1}(1) = 2 - 1 = 1$.

Let $\mathrm{u}_i \in \mathcal{U}$ be an input transition of a TEG $\mathcal{G}$, the event flow associated with this transition is denoted $u_i \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$. Then $U = [u_1, \ldots, u_{|\mathcal{U}|}]^T \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{|\mathcal{U}|}$ denotes the input vector of $\mathcal{G}$. Similarly, the event flow of an internal transition $\mathrm{x}_i \in \mathcal{X}$ (resp. an output transition $\mathrm{y}_i \in \mathcal{Y}$) is denoted as a series $x_i \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ (resp. $y_i \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$). Vector $X = [x_1, \ldots, x_{|\mathcal{X}|}]^T$ is the corresponding *state vector* of $\mathcal{G}$ and vector $Y = [y_1, \ldots, y_{|\mathcal{Y}|}]^T$ is the corresponding *output vector* of $\mathcal{G}$. X and Y are defined by the following state representation system:

$$\begin{cases} X = AX \oplus BU \\ Y = CX \end{cases} \tag{17}$$

where matrices A, B, C are such that $A \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{|\mathcal{X}|^2}$, $B \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{|\mathcal{X}| \times |\mathcal{U}|}$, and $C \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{|\mathcal{Y}| \times |\mathcal{X}|}$.

By applying Theorem 1, a direct relation between Y and U can be obtained:

$$Y = HU \text{ with } H = CA^*B. \tag{18}$$

Matrix $H \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{|\mathcal{Y}| \times |\mathcal{U}|}$ is called the *transfer matrix*.

Example 11. *Back to the TEG of Fig. 1, its state representation is therefore characterized by matrices* $A \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{6\times 6}, B \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{6\times 4}, C \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{3\times 6}$. *For instance the element* $A_{2,1}$ *of matrix* A *is* $A_{2,1} = \gamma^0\delta^1$ *which means that the* n^{th} *fire of* x_2 *is at least 1 time unit after the* n^{th} *fire of* x_1 *while the element* $A_{2,6}$ *of matrix* A *is* $A_{2,6} = \gamma^2\delta^2$ *which means that the* n^{th} *fire of* x_2 *is at least 2 time units after the* $n^{th} - 2$ *fire of* x_6. *Finally, any element like* $A_{4,2}$ *is assigned with* ε *as* $\mathrm{pre}(\mathrm{x}_4) \cap \mathrm{post}(\mathrm{x}_2) = \varnothing$. *Matrix* B *involves input transitions,*

so for instance $B_{5,4} = \gamma^0\delta^1$ *(between input* u_4 *and transition* x_5*) and* $B_{6,4} = \varepsilon$*. Similarly, Matrix* C *involves output transitions (*$C_{1,3} = \gamma^0\delta^0$*,* $C_{3,2} = \varepsilon, \ldots$*). The transfer matrix then is a matrix* $H \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{3\times 4}$*. For instance, element* $H_{1,1} = \gamma^0\delta^1(\gamma^2\delta^2)^*$ *rules the fires of output* y_1 *with respect to input* u_1*; the full transfer matrix is detailed in Section 6.3 by Equation* (32)*.*

Finally, the next property trivially follows from Equation (18):

Proposition 3. *Let* $\mathrm{u}_j \in \mathcal{U}, \mathrm{y}_i \in \mathcal{Y}$*, a path* $\mathrm{u}_j \rightsquigarrow \mathrm{y}_i$ *exists if and only if the transfer function* H *is such that* $H_{ij} \neq \varepsilon$*.*

4 Active diagnosis of a TEG: tools and assumptions

Active diagnosis is the generic problem of setting up and applying an action plan or a control policy on the system in order to monitor its observable response and refine a previous diagnosis. The success of an active diagnosis process relies on the synthesis of action plans or control policies that, once applied, ensure that the observable response is discriminative enough and leads to the pruning of diagnostic candidates that were determined by a previous diagnosis stage. In this paper, we address the problem of active diagnosis of time failures in systems modeled as timed event graphs. The objective is to synthesize a sequence of controls to finally better localize the origin of the time failure within the system. The effective application of this sequence of controls and the analysis of their respective observable responses with the help of health indicators is called an *active diagnosis session* [5]. To design such an active diagnosis session, two subtasks are necessary: the control task in charge of synthesizing and applying a control sequence and the diagnosis task that is in charge of monitoring the observable response and determining the new set of potential time failure sources. Subsection 4.1 formally describes the diagnosis task that is used by the proposed active diagnosis method. This task is based on a set of time failure indicators defined in [20, 14] over a (max,+)-linear system that are designed for the detection and localization of such failures. Note that it is nonetheless possible to use other time failure indicators such as the ones of [18] or any other time failure indicators as long as they are sound. Then, Subsection 4.2 briefly recalls the control theory in (max,+)-linear systems that is used all along this paper for the implementation of the control task. Subsection 4.3 introduces the active diagnosis problem investigated in this paper and details the global assumptions required by the proposed method.

4.1 Diagnosis of time failures in TEGs

All along this paper, we consider that the behavior of the supervised underlying system is modeled as a timed event graph $\mathcal{G} = \langle \mathcal{P}, \mathcal{T}, \mathcal{A}, M_0, \mathcal{HT} \rangle$ which has at least one input transition ($\mathcal{U} \neq \varnothing$) and one output transition ($\mathcal{Y} \neq \varnothing$). A transition $\mathrm{t} \in \mathcal{T}$ is said to be measurable if the date of every fire of the transition t is

known by the supervisor in charge of performing an active diagnosis session. In this paper, we consider that the system is partially measurable. More precisely, only input and output transitions are measurable: i.e. the timed sequence of fires of an input transition $\mathrm{u_i} \in \mathcal{U}$ (resp. an output transition $\mathrm{y_i} \in \mathcal{Y}$) is known by the supervisor at any operating time; in other words, its corresponding series u_i (resp. y_i) is known. This is not the case of internal transitions ($\mathcal{X}$), whose information on events is considered to be unavailable to the supervisor.

4.1.1 Time failures

The purpose that is addressed is to improve the way of localizing the origin of a time failure within a partially observable system $\mathcal{G}$. A time failure is a phenomenon that occurs on a system processing resources and which produces a *delay*. For instance, in an automated assembly line, a machine tool may work in a degraded mode and operate with unexpected delays. A resource can also be a transport resource impacted by delays from the environment to go from a site A to a site B, etc. In the context of systems modeled as TEGs, a resource is modeled by a place $\mathrm{p} \in \mathcal{P}$ and its processing duration is modeled by the duration $d = \mathcal{HT}(\mathrm{p})$. In this setting, a time failure is formally defined as follows.

Definition 10 (Time failure)**.** *A* time failure *held by a place* $\mathrm{p} \in \mathcal{P}$ *whose normal duration is* $d = \mathcal{HT}(\mathrm{p})$*, is a relative delay* $\theta \in \mathbb{N} \setminus \{0\}$ *so that the real duration associated with* p *is* $d + \theta \in \mathbb{N}^+$*.*

Assumption 3 (Permanent time failure)**.** *All along this paper, only permanent time failures are considered, i.e.* $\mathcal{HT}(\mathrm{p}) + \theta$ *holds at any time within the system* $\mathcal{G}$*.*

Assumption 3 states that the effect of a time failure is permanent on the holding time in the involved place p and has an effect on any token reaching and then waiting in that place p.

Proposition 4. *Let* H *be the normal transfer function of a system with normal output* $\tilde{Y} = HU$*, if the system holds a set of permanent time shift failures then there exists a transfer function* H' *such that*

1. $Y = H'U$ *is the real output of the system;*
2. $H' \succeq H$*.*

Proof. Firstly, let us prove the result for the presence of a single time failure held on place $\mathrm{p}_f \in \mathcal{P}$ with a duration $\theta > 0$. Let $\mathcal{G} = (\mathcal{P}, \mathcal{T}, \mathcal{A}, M_0, \mathcal{HT})$ be the TEG of the system and $H = CA^*B$ with A, B, C as defined in Equation (17). As it is permanent, it is possible to design the TEG $\mathcal{G}' = (\mathcal{P}, \mathcal{T}, \mathcal{A}, M_0, \mathcal{HT}')$ of the failing system by simply replacing the holding time p_f: $\mathcal{HT}'(\mathrm{p}_f) = \mathcal{HT}(\mathrm{p}_f) + \theta$ and $\forall \mathrm{p} \in \mathcal{P} \setminus \{\mathrm{p}_f\}, \mathcal{HT}'(\mathrm{p}) = \mathcal{HT}(\mathrm{p})$. Let H' be the transfer function of $\mathcal{G}'$, it follows that $Y = H'U$.

1. Suppose first that $\mathrm{post}(\mathrm{p}_f) = \{\mathrm{y}_f\} \subseteq \mathcal{Y}$. As $H = CA^*B$, it follows that there exists C' such that $H' = C'A^*B$ with A, B, C' being the matrices in the state representation of TEG $\mathcal{G}'$. It follows $y_f = \bigoplus_{i=1}^{|\mathcal{X}|} C'_{fi}x_i$. By construction of $\mathcal{G}'$ and $\mathcal{G}$, if $C_{fi} = \varepsilon$, then $C'_{fi} = \varepsilon$, otherwise, $C_{fi} = \gamma^{n_{fi}}\delta^{t_{fi}}$ and $C'_{fi} = \gamma^{n_{fi}}\delta^{t_{fi}+\theta}$. Now remark that $\gamma^{n_{fi}}\delta^{t_{fi}} \oplus \gamma^{n_{fi}}\delta^{t_{fi}+\theta} = \gamma^{n_{fi}}\delta^{max(t_{fi},t_{fi}+\theta)} = \gamma^{n_{fi}}\delta^{t_{fi}+\theta}$, it follows that $C'_{fi} \succeq C_{fi}$ and $C' \succeq C$. Finally, $H' \succeq H$.

2. Now suppose $\mathrm{pre}(\mathrm{p}_f) = \{\mathrm{u_f}\} \subseteq \mathcal{U}$. Let $\{\mathrm{x_f}\} = \mathrm{post}(\mathrm{p}_f) \subseteq \mathcal{X}$. It follows that $x_f = \bigoplus_{i=1}^{|\mathcal{X}|} A_{fi}x_i \oplus \bigoplus_{i=1}^{|\mathcal{U}|} B'_{fi}u_i$. By construction of $\mathcal{G}'$ and $\mathcal{G}$, if $B_{fi} = \varepsilon$, then $B'_{fi} = \varepsilon$, otherwise, $B_{fi} = \gamma^{n_{if}}\delta^{t_{if}}$ and $B'_{fi} = \gamma^{n_{if}}\delta^{t_{if}+\theta}$. Similarly, as in Case 1, $B' \succeq B$, so $H' \succeq H$.

3. The last case is when $\mathrm{pre}(\mathrm{p}_f) = \{\mathrm{x}_\ell\} \subseteq \mathcal{X}$ and $\mathrm{post}(\mathrm{p}_f) = \{\mathrm{x}_m\} \subseteq \mathcal{X}$. It follows that $x_m = \bigoplus_{i=1}^{|\mathcal{X}|} A'_{mi}x_i \oplus \bigoplus_{i=1}^{|\mathcal{U}|} B_{mi}u_i$. By construction, the only difference between A and A' is their respective elements $A_{\ell m}$ and $A'_{\ell m}$ with $A_{\ell m} = \gamma^{n_{\ell m}}\delta^{t_{\ell m}}$ and $A'_{\ell m} = \gamma^{n_{\ell m}}\delta^{t_{\ell m}+\theta}$. It follows that $A' \succeq A$. As the Kleene operator is isotone, it follows that $(A')^* \succeq (A)^*$. Hence $H' \succeq H$.

Finally, the proof of the result for any finite set of time failures in $\mathcal{G}$ is by induction on the set of failures. Indeed, considering the normal system now as the system modeled by $\mathcal{G}'$ and considering a second time failure, the same previous reasoning applies. □

Among the possible set of time failures, there are two subclasses. The first class gathers the time failures that do indeed have an observable effect by delaying at least one measurable output y_i. In this case the time failure is detectable. The other class contains the time failures that have no observable effect. This is usually due to a downstream synchronization that compensates the effect of the time failure by waiting for tokens from healthy places.

Definition 11 (Detectable time failure)**.** *A time failure is said to be* detectable *if it leads to the production of a real output Y that is different from the normal output $\tilde{Y}$.*

Corollary 5. *If a permanent time failure is detectable then $H' \succ H$.*

Proof. By Proposition 4, we know that $H' \succeq H$. As the time failure is detectable, $\tilde{Y} \neq Y$, therefore $H' \neq H$. □

Assumption 4 (Detectability of time failure)**.** *All along this paper, only detectable time failures are considered.*

Assumption 4 asserts that the aim of the diagnosis task is to localize the source of detectable time failures only. The case when a time failure is indeed present in $\mathcal{G}$ but not detectable is not addressed. For the sake of simplicity in the following, the notion of time failure will stand for permanent detectable time failure.

4.1.2 Indicators

The diagnosis task involved in the proposed active diagnosis method is composed of two subtasks: detection and localization. The detection task consists in measuring the inputs and the outputs of the real system and comparing them to the expected inputs and outputs. A difference between the reality and the expectation means that the modeled behavior does not represent the real behavior, hence the occurrence of a time failure. The localization task then consists in identifying, among the set of places, which ones hold the time failure.

Regarding the detection, as input U is measurable, the real input is always the expected input, however the real output Y might be different from the expected output $\tilde{Y} = HU$ (the expectation is that the output $\tilde{Y}$ results from the model defined in Equation (17) based on the real input U). In [20], a method that exploits this is proposed, relying on the set of indicators explained in the following.

Definition 12 (Indicator of time failures)**.** *Let U be the measurable input of $\mathcal{G}$ such that $HU \neq \varepsilon$ and let $Y = [y_1, \ldots, y_{|\mathcal{Y}|}]^T \neq \varepsilon$ be its measurable output. Indicator $I(U, y_i)$ for single output $y_i \neq \varepsilon$ is the Boolean function:*

$$I(U, y_i) = \begin{cases} \mathit{false} & \text{if } \Delta(y_i, \tilde{y}_i) = [0;0] \text{ for } \tilde{y}_i \text{ such that } [\tilde{y}_1, \ldots, \tilde{y}_{|\mathcal{Y}|}]^T = HU, \\ \mathit{true} & \text{otherwise} \end{cases}$$

with

$$\Delta(y_i, \tilde{y}_i) = \ [\underline{\mathcal{T}}_{\tilde{y}_i, y_i} \ ; \ \overline{\mathcal{T}}_{\tilde{y}_i, y_i}], \tag{19}$$

the time interval *of y_i, and the bounds of $\mathcal{T}_{\tilde{y}_i, y_i}$ as defined in Theorem 2.*

As detailed in [13], the following result holds.

Theorem 6 (Sound indicators)**.** *Let U be the measurable input of $\mathcal{G}$ such that $HU \neq \varepsilon$ and let $Y = [y_1, \ldots, y_{|\mathcal{Y}|}]^T \neq \varepsilon$ be its measurable output, at least one detectable time failure is present in the system if there is at least one output y_i such that $I(U, y_i) = \mathit{true}$.*

In the following, any output y_i that raises the indicator I will be called a *delayed output*, formally:

Definition 13 (Delayed output)**.** *Let U be an input for system $\mathcal{G}$, and $Y = [y_1, \ldots, y_{|\mathcal{Y}|}]^T$ be the real output of $\mathcal{G}$ when operating U, y_i is a* delayed output *if*

$$I(U, y_i) = \mathit{true}.$$

The set of delayed output transitions of $\mathcal{G}$ is therefore denoted:

$$\mathcal{Y}_{del,U} = \{\mathrm{y_i} \in \mathcal{Y} | I(U, y_i) = \mathit{true}\}.$$

4.1.3 Localization of time failures

Once the detection task concludes that at least one time failure is present in the system, the localization task can start in order to identify the set of places p that could hold the time failure. Based on the parsimony principle in model-based diagnosis, we will suppose that the detection results from the presence of one time failure only.

Assumption 5 (Single time failure)**.** *The localization task assumes that only one time failure is present in the system.*

The objective of the localization task is therefore to exploit the results from the set of indicators in order to determine a set of places that could potentially hold the time failure and ultimately the unique place p that holds the time failure.

Proposition 7. *The place* p_f *holding the time failure is upstream of all the delayed outputs, i.e.*

$$\forall \mathrm{y} \in \mathcal{Y}_{del,u}, \mathrm{p}_f \rightsquigarrow \mathrm{y}. \tag{20}$$

Proof. This property is a direct consequence of Assumption 2 on the structural observability of the TEG. □

In [14], a method for localizing such a time failure exploiting the structure of $\mathcal{G}$ has been proposed.

Definition 14 (Structure-based localization)**.** *Let U be the measurable input of $\mathcal{G}$ such that $HU \neq \varepsilon$ and let $Y \neq \varepsilon$ be its measurable output such that $\exists \mathcal{Y}_{true} \subseteq \mathcal{Y}, \mathcal{Y}_{true} \neq \varnothing$ and $\forall \mathrm{y_i} \in \mathcal{Y}_{true}, I(U, y_i) = true$, the* Structure-based localization *of the time failure in $\mathcal{G}$ is the set:*

$$Loc(\mathcal{G}, U, Y) = \{\mathrm{p} \in \mathcal{P}, \forall \mathrm{y_i} \in \mathcal{Y}, \mathrm{y_i} \in \mathcal{Y}_{true} \Rightarrow \mathrm{p} \rightsquigarrow \mathrm{y_i}\}.$$

Intuitively speaking, a place p is a candidate for holding the time failure if the place is in the upstream of *every* output transition for which the corresponding indicator is true. As proved in [14], the next result follows.

Theorem 8. *Under the assumption of a single time failure, place* p *holds the time failure only if* $\mathrm{p} \in Loc(\mathcal{G}, U, Y)$.

4.2 Control in (max,+)-linear systems

The principle of active diagnosis is to set up control policies over the supervised system $\mathcal{G}$ in order to analyze the response of $\mathcal{G}$ and refine the localization of time failures. In this paper, the proposed method for controlling the system $\mathcal{G}$ relies on the control theory of (max,+)-linear systems. Generally speaking, the control aims at designing a specific input U that ensures that the system $\mathcal{G}$ achieves a behavior X among a set of pre-specified target behaviors.

Definition 15 (Control Problem)**.** *Let $TB \subseteq \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{|\mathcal{X}|}$ be a set of pre-specified target behaviors of the TEG $\mathcal{G}$, the* control problem *over $\mathcal{G}$ is the synthesis of an input U to ensure that, at any time, state X (with $X = AX \oplus BU$) is such a target behavior ($X \in TB$).*

The result of a control problem is a synthesized input U also called *control* U. In (max,+)-linear systems, control U usually aims at delaying specific events of the input flows to finally get a target behavior of the system.

Example 12. *For example, among existing controllers, the so-called* optimal control *for TEG ([7]) is an open-loop strategy that consists in computing the greatest input flow U_{opt} to ensure that state X_{opt} is such that $X_{opt} = AX_{opt} \oplus BU_{opt}, Y_{opt} = CX_{opt}, Y_{opt} \preceq Y_r$ where Y_r is a known reference output [17].*

With series of $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$, the "greatest series" is the most delayed series and a lesser series is faster.

4.3 Active diagnosis: objectives and assumptions

This paper details an algorithm to be used in an active diagnosis session over a system $\mathcal{G}$. An active diagnosis session can start as soon as a time shift failure has been detected in the system at operating time. Let U_{op} be the vector of inputs that has been applied at *operating time* and Y_{op} the corresponding measured output. A time shift failure has been detected in the system as soon as there exists a delayed output in Y_{op} (see Definition 13), in other words, as soon as:

$$\mathcal{Y}_{del,U_{op}} \neq \varnothing. \tag{21}$$

Instead of directly performing a structure-based localization based on U_{op}, Y_{op} only, the objective of the active diagnosis is to design a sequence of controls $U_1, \ldots, U_k, \ldots$ to get more information and a more precise diagnosis (see Fig. 5). At each step k of the active diagnosis session, the first stage is to synthesize a new input U_k that is then applied to the real system. The system then produces the output Y_k as an observable response. The failure detection module then computes the expected output $\tilde{Y}_k = HU_k$ and the results of the indicator $I(U_k, Y_k) = \bigvee_{1 \leq i \leq |\mathcal{Y}|} I(U_k, y_i)$. Based on these new results, a new localization analysis is performed to provide a new set of candidate places $\mathcal{P}_{cand}$. Then, based on the active diagnosis strategy, either a new step $k+1$ is performed or the active diagnosis session ends and provides $\mathcal{P}_{cand}$ as the result.

Throughout this paper, the stages consisting in applying control U_k on the real system, recording Y_k and computing $\tilde{Y}_k$ and $I(U_k, Y_k)$ will be simply represented in the proposed algorithms by a call to the function:

$$(Y_k, \tilde{Y}_k, I(U_k, Y_k)) \leftarrow \mathit{ApplyControl}(U_k).$$

The key to success when starting an active diagnosis session is to perfectly know the initial marking of the system $\mathcal{G}$ before applying any control. The proposed method thus requires a set of assumptions on the timed event graph $\mathcal{G}$.

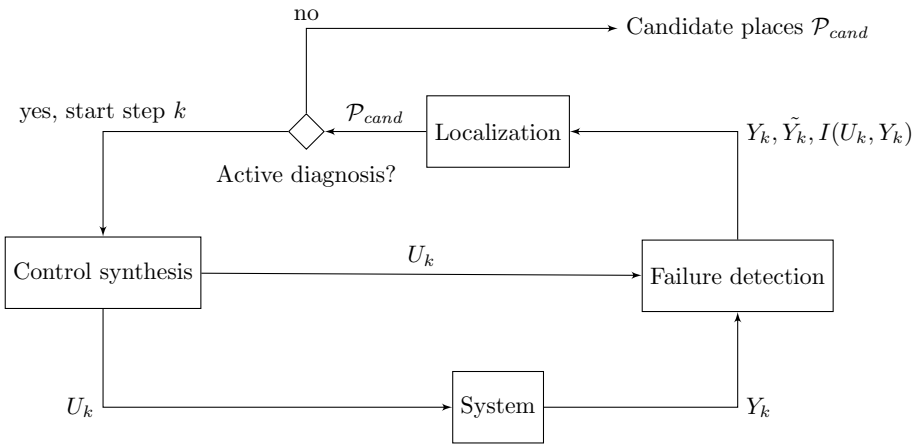

Figure 5: Active diagnosis architecture

Definition 16 (Output in phase with input)**.** *Let H be the transfer function of $\mathcal{G}$ based on its initial marking M_0 ($Y = HU$), an output $\mathrm{y_i} \in \mathcal{Y}$ of $\mathcal{G}$ is said* to be in phase *with an input $\mathrm{u_j} \in \mathcal{U}$ if*

$$(\mathrm{u_j} \rightsquigarrow \mathrm{y_i}) \Rightarrow (H_{ij} \succeq \gamma^0\delta^0). \tag{22}$$

Intuitively speaking, if $\mathrm{y_i}$ is in phase with $\mathrm{u_j}$, the first fire of $\mathrm{y_i}$ requires the first fire of $\mathrm{u_j}$. Note that the definition of H_{ij} depends on the initial marking M_0.

Definition 17 (In phase)**.** *A TEG $\mathcal{G}$ is said to be* in phase *if every output transition $\mathrm{y_i}$ of $\mathcal{G}$ is in phase with the set of input transitions $\mathrm{u_j}$ upstream of it, formally:*

$$\forall \mathrm{y_i} \in \mathcal{Y}, \forall \mathrm{u_j} \in \mathcal{U}, (\mathrm{u_j} \rightsquigarrow \mathrm{y_i}) \Rightarrow (H_{ij} \succeq \gamma^0\delta^0). \tag{23}$$

Intuitively speaking, when a TEG is in phase, it means that, starting at time $t = 0$ from the initial marking M_0, the first fire of an output transition $\mathrm{y_i}$ is due to the presence of tokens in the preset of $\mathrm{y_i}$ that are all resulting from the first fire of every input transition $\mathrm{u_j}$ that leads to transition $\mathrm{y_i}$. Transition $\mathrm{y_i}$ cannot be fired as long as every input transition $\mathrm{u_j}$ has been fired once and the resulting produced tokens are not present in the preset of $\mathrm{y_i}$. The proposed active diagnosis method requires this property.

Example 13. *A single-input single-ouput TEG $\mathcal{G}_1$ with the transfer function $h_1 = \gamma^0\delta^1 \oplus \gamma^1\delta^2$ would be in phase. A TEG $\mathcal{G}_2$ with the transfer function $h_2 = \gamma^1\delta^1$ would not be in phase ($h_2 \not\succeq \gamma^0\delta^0$): the first event of its output takes places before the first event of its input.*

Assumption 6 ($\mathcal{G}$ is in phase)**.** *TEG $\mathcal{G}$ is assumed to be in phase: condition (23) holds in $\mathcal{G}$.*

This first assumption is structural and also depends on the initial marking M_0, but it does not depend on the behavior of the TEG at a given time $t > 0$. The next assumption states some properties regarding the marking that are required to properly start an active diagnosis session at $t > 0$.

Definition 18 (Stuck TEG)**.** *The TEG $\mathcal{G}$ is* stuck *at time t if none of the internal and output transitions of $\mathcal{G}$ are enabled at time t.*

In other words, in a stuck TEG at time t, internal or output transitions cannot fire as long as no input transitions are fired. The first transition that can be fired in $\mathcal{G}$ after time t is an input transition.

Definition 19 (Empty TEG)**.** *A TEG $\mathcal{G}$ is empty at time t if*

1. *$\mathcal{G}$ is* stuck *at time t;*
2. *all the tokens in $\mathcal{G}$ are contained in places that belong to circuits;*
3. *for every elementary circuit, the current marking M is such that $M(\mathrm{p}) \geq 1$ for one and only one place* p *in the elementary circuit.*

When $\mathcal{G}$ is empty at time t, it firstly means that the underlying system does not operate anymore at time t and waits for new environmental inputs. The second and third conditions also mean that the processes involved in the system are properly reinitialized, and there are no blocked resources waiting only for a synchronization.

As Assumption 6 must hold and depends on M_0, the second assumption that is required to start a diagnostic session follows.

Assumption 7 ($\mathcal{G}$ is empty)**.** *$\mathcal{G}$ is empty in the initial marking M_0. Any active diagnosis session starts at a time t only when the current marking is the initial marking M_0.*

It is very likely that when a failure has been detected based on the inputs U_{op}, the current resulting marking M does not fulfill the previous assumptions. In this case, to start the diagnostic session, the system must be reset, either by synthesizing a supplementary input U to empty the system as required by Assumption 7 or by performing a hard reset on the system represented by a new marking compatible with Assumption 7. Moreover, the clock is reset ($t = 0$) when the active diagnosis session starts and at any step k.

Finally, for the sake of generality, note that an active diagnosis session may start on a marking M_1 different from the initial marking by redefining the TEG as $\langle \mathcal{P}, \mathcal{T}, \mathcal{A}, M_1, \mathcal{HT} \rangle$ if it is in phase and empty for the marking M_1 (which would now be considered the initial marking). This would lead to a new transfer matrix H, but the structure of the TEG remains unchanged.

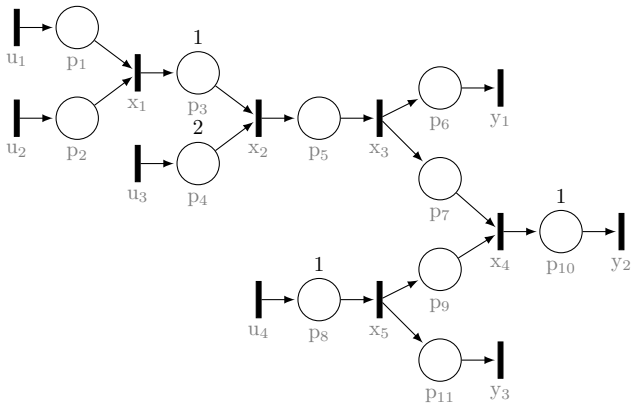

Figure 6: TEG with four inputs and three outputs.

5 Active localization for Multiple-Input TEGs

This section introduces a method for the localization of time failures in TEGs which constitutes an active diagnosis algorithm as introduced in Section 4.3.

The section is divided in four parts: Section 5.1 introduces the theoretical elements related to the localization of time failures, Section 5.2 describes an active localization algorithm for multiple-input TEGs, Section 5.3 contains the properties of this algorithm, and finally, Section 5.4 details the general localization method.

5.1 Localization of failures in multiple-input TEGs

This section introduces the properties of the inputs and paths of TEGs that contain time failures. Specifically, these properties concern the place containing the failure and allow for a more precise localization of this place if the TEG contains multiple inputs.

Definition 20 ($\mathcal{U}_{del,U}$)**.** *The set of input transitions leading to all the delayed outputs for a given control U is:*

$$\mathcal{U}_{del,U} = \{\mathrm{u}_k | \forall \mathrm{y}_i \in \mathcal{Y}_{del,U}, \mathrm{u}_k \rightsquigarrow \mathrm{y}_i\}. \tag{24}$$

This set contains the inputs that could potentially be upstream of the place containing the time failure p_f. $\mathcal{U}_{del,U}$ can be obtained by determining which series in the transfer matrix H are different from ε because $\mathrm{u}_k \rightsquigarrow \mathrm{y}_i$ is true if and only if $H_{ik} \neq \varepsilon$ (see Proposition 3).

Example 14. *Let us consider the TEG in Fig. 6. It is a MIMO TEG with four inputs and three outputs. Its transfer matrix is:*

$$H = \begin{pmatrix} \gamma^0\delta^1 & \gamma^0\delta^1 & \gamma^0\delta^2 & \varepsilon \\ \gamma^0\delta^2 & \gamma^0\delta^2 & \gamma^0\delta^3 & \gamma^0\delta^2 \\ \varepsilon & \varepsilon & \varepsilon & \gamma^0\delta^1 \end{pmatrix}. \tag{25}$$

Let us assume that there is a time failure of 1 *time unit on place* p_5 *(the value and the place of the failure are unknown). An input* U *results in the detection of a time shift on outputs* y_1 *and* y_2*. The set of delayed outputs is* $\mathcal{Y}_{del,U} = \{\mathrm{y}_1, \mathrm{y}_2\}$ *and the set of inputs leading to all the delayed outputs is* $\mathcal{U}_{del,U} = \{\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3\}$*.* u_4 *is not included in the latter because it is not upstream of* y_1*, as shown by the transfer matrix (*$H_{14} = \varepsilon$*).*

Proposition 9. *If an input* u_k *is upstream of the place containing the failure* p_f*, then it belongs to* $\mathcal{U}_{del,U}$*:* $\mathrm{u}_k \rightsquigarrow \mathrm{p}_f \Rightarrow \mathrm{u}_k \in \mathcal{U}_{del,U}$*.*

Proof. The relation $\rightsquigarrow$ (see Definition 2) is transitive. Therefore,

$$\begin{aligned} &\mathrm{u}_k \rightsquigarrow \mathrm{p}_f \text{ and } \forall \mathrm{y}_i \in \mathcal{Y}_{del,U},\ \mathrm{p}_f \rightsquigarrow \mathrm{y}_i && \text{(By Proposition 7)} \\ &\Rightarrow \forall \mathrm{y}_i \in \mathcal{Y}_{del,U}, \mathrm{u}_k \rightsquigarrow \mathrm{y}_i \Leftrightarrow \mathrm{u}_k \in \mathcal{U}_{del,U}. \end{aligned}$$

□

Remark 1. $\mathcal{U}_{del,U}$ *cannot be empty:* $\mathcal{U}_{del,U} \neq \varnothing$*. This is a direct consequence of Proposition 9.*

Proposition 10. *If* $\mathcal{U}_{del,U}$ *contains only one input* u_k*, then* u_k *is the only input upstream of the place containing the failure:* $|\mathcal{U}_{del,U}| = 1 \Rightarrow \exists! \, \mathrm{u}_k \,|\, \mathrm{u}_k \rightsquigarrow \mathrm{p}_f$*.*

Proof. The existence of at least one $\mathrm{u}_k \in \mathcal{U}_{del,U}$ upstream of the place containing the failure was proven in the previous propositions. At present let us prove that if an input u_j is different from u_k (which belongs to the singleton $\mathcal{U}_{del,U}$) then it is not upstream of the place containing the failure p_f:

$$\begin{aligned} &|\mathcal{U}_{del,U}| = 1 \Leftrightarrow \exists! \, \mathrm{u}_k \,|\, \mathrm{u}_k \in \mathcal{U}_{del,U} \\ &\Rightarrow \forall \mathrm{u}_j \neq \mathrm{u}_k, \mathrm{u}_j \notin \mathcal{U}_{del,U} \\ &\Rightarrow \forall \mathrm{u}_j \neq \mathrm{u}_k, \mathrm{u}_j \not\rightsquigarrow \mathrm{p}_f. && \text{(by Proposition 9)} \end{aligned}$$

□

Example 15. *Let us consider the TEG in Fig. 6 and its transfer matrix (Equation (25)), but this time, a time failure occurs on place* p_{11}*. An input* U *which leads to the detection of a time failure on* y_3 *would lead us to the set of delayed outputs* $\mathcal{Y}_{del,U} = \{\mathrm{y}_3\}$ *and the set of inputs leading to all the delayed outputs* $\mathcal{U}_{del,U} = \{\mathrm{u}_4\}$*.*

Since u_1*,* u_2 *and* u_3 *are not upstream of the delayed output* y_3*, they cannot be upstream of the failure. The only remaining input is* u_4*, so it is necessarily upstream of the time failure (see Proposition 10).*

The control algorithm for multiple inputs described in the following section helps determine which inputs are actually upstream of the place containing the time failure. After the normal operation of the system has concluded, let us consider U_{op} as defined in Section 4.3. If $\mathcal{U}_{del,U_{op}}$ contains a single input, then this input is upstream of the place containing the failure (Proposition 10) and

the control algorithm cannot provide any more information (see Example 15). However, if $\mathcal{U}_{del,U_{op}}$ contains at least two different inputs (as in Example 14), this set can be used to improve the localization of the time failure.

5.2 Control Algorithm for Multiple Inputs (CAMI)

This section describes an active localization algorithm called Control Algorithm for Multiple Inputs (CAMI). Throughout this section, the sets $\mathcal{U}_{del,U_{op}}$ and $\mathcal{Y}_{del,U_{op}}$ introduced in the previous section will simply be denoted $\mathcal{U}_{del}$ and $\mathcal{Y}_{del}$ respectively.

CAMI is designed to determine which inputs among those in $\mathcal{U}_{del}$ are actually upstream of the place containing the time failure. This algorithm uses an *ad hoc* control sequence containing $|\mathcal{U}_{del}|$ independent control steps. Each control step k helps determine whether input $\mathrm{u}_k \in \mathcal{U}_{del}$ is upstream of the place containing the failure.

In empty TEGs, one of the setbacks in the detection of time failures is the fact that the delays on places containing tokens (which are all contained in circuits, see Assumption 7) only intervene after these initial tokens have been used. Therefore, the choice of the number of events for each input U_k in an active diagnosis session is important, as one of the targets is to allow the durations of all places in all circuits to be expressed in order to properly observe the behavior of the system.

Let us begin by defining the necessary concepts for the synthesis of a relevant control sequence for the active localization of failures in TEGs.

Let Ω be defined as follows:

$$\Omega = 1 + \sum_{i=1}^{|P|} M_0(\mathrm{p}_i). \tag{26}$$

An input containing Ω events guarantees that the transitions on the elementary paths from inputs to outputs are fired enough times to express the durations (and therefore the delays) of all places outside these elementary paths.

Ω is designed to be conservative and a lesser number of events may be sufficient to achieve its goal for certain TEGs, which could be determined on a case-by-case basis.

Example 16. *For the TEG in Fig. 1,* $\Omega = 1 + 12 = 13$. *This number of events ensures that, for instance, transition* x_2 *is fired enough times to effectively observe the durations of places* p_{12}, p_{13} *and* p_{14}, *so even though these places are outside the elementary paths from the inputs to the outputs, any delay could still be detected.*

Ω will therefore be used as the number of events for the inputs in the proposed active diagnosis algorithm. The following two definitions will be used when establishing the dates of the events of said inputs.

Definition 21 (Traversal time t_{ij})**.** *t_{ij} is the traversal time for Ω tokens to travel from u_j to y_i:*

$$t_{ij} = \mathcal{D}_{H_{ij}}(\Omega - 1).$$

This traversal time is equal to the dater of event $\Omega - 1$ of the transfer function from input u_j to output y_i.

For TEGs that do not contain any circuits, t_{ij} is simply the sum of the holding times of the places on the slowest path going from u_j to y_i.

Example 17. *For the TEG in Fig. 6, $\Omega = 1$. The traversal times from each input to each output are:*

- *for input u_1: $t_{11} = 1$, $t_{21} = 2$, $t_{31} = -\infty$,*
- *for input u_2: $t_{12} = 1$, $t_{22} = 2$, $t_{32} = -\infty$,*
- *for input u_3: $t_{13} = 2$, $t_{23} = 3$, $t_{33} = -\infty$,*
- *for input u_4: $t_{14} = -\infty$, $t_{24} = 2$, $t_{34} = 1$.*

Definition 22. *The time $t_{max,k}$, where k is the number of an input transition of the TEG, is defined as:*

$$t_{max,k} = \max_{i,j}(t_{ij}) + 1 \text{ with } 1 \leq i \leq |\mathcal{Y}| \text{ and } \mathrm{u}_j \in \mathcal{U} \setminus \mathrm{u}_k.$$

The traversal times used to compute $t_{max,k}$ are those from the inputs in $\mathcal{U}$ except u_k to all outputs y_i. $t_{max,k}$ is therefore the greatest traversal time for Ω tokens to travel along all paths $\mathrm{u}_{j\neq k} \rightsquigarrow \mathrm{y}_i$ plus one.

$t_{max,k}$ can be computed as follows: let S_k be the sum in $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ of the series in matrix H corresponding to the inputs in $\mathcal{U} \setminus \{\mathrm{u}_k\}$:

$$S_k = \bigoplus_{i\geq 1}^{|\mathcal{Y}|} \bigoplus_{j\geq 1, j\neq k}^{|\mathcal{U}|} H_{ij}.$$

The time $t_{max,k}$ used for control sequence k can be computed as:

$$t_{max,k} = \mathcal{D}_{S_k}(\Omega - 1) + 1. \tag{27}$$

$t_{max,k}$ is one plus the dater of $\Omega - 1$ of the sum of the series in H on all columns except column k.

Example 18. *In Example 14 (Fig. 6), the set $\mathcal{U}_{del,U} = \{\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3\}$ was obtained for $\mathcal{Y}_{del,U} = \{\mathrm{y}_1, \mathrm{y}_2\}$.*

The following traversal times are considered in the definition of $t_{max,1}$:

$$t_{max,1} = \max(t_{12}, t_{13}, t_{14}, t_{22}, t_{23}, t_{24}t_{32}, t_{33}, t_{34}) + 1 = \max(-\infty, 1, 2, 3) + 1.$$

$t_{max,2}$ and $t_{max,3}$ are defined in a similar fashion. The resulting times for $k = 1, 2, 3$ are:

$$t_{max,1} = 4, \quad t_{max,2} = 4, \quad t_{max,3} = 3.$$

Definition 23. *Let $U_{MI,k} \in \mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]^{|\mathcal{U}|}$ be the input vector used in control step k: $U_{MI,k} = (r_1 \ ... \ r_{|\mathcal{U}|})^T$. Each row r_j of $U_{MI,k}$ is defined as:*

$$r_j = \begin{cases} \gamma^0\delta^0 \oplus \gamma^\Omega\delta^{+\infty} & \text{if } j \neq k, \\ \gamma^0\delta^{t_{max,k}} \oplus \gamma^\Omega\delta^{+\infty} & \text{if } j = k. \end{cases} \tag{28}$$

Both trajectories r_k and $r_{j\neq k}$ introduce the same amount of tokens (Ω), but the former does it at $t = 0$, making it faster than the latter, which introduces them at $t = t_{max,k}$.

The trajectory on input u_k (for step k) is therefore slower than the trajectories of all other inputs. The tokens introduced on the other inputs at $t = 0$ will be stopped at any synchronization merging paths from u_k and will not reach the outputs downstream of it at least until tokens are injected on u_k at $t = t_{max,k}$. This choice is aimed at maximizing the delay caused by the failure if the failure is on a place downstream of u_k.

In the following, the computation of $U_{MI,k}$ in the context of an algorithm will be denoted $ComputeU_{MI,k}(H, \Omega, k)$.

Example 19. *Following Example 18, the generated inputs are:*

$$U_{MI,1} = \begin{pmatrix} \gamma^0\delta^4 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \end{pmatrix}, \quad U_{MI,2} = \begin{pmatrix} \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^4 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \end{pmatrix},$$

$$U_{MI,3} = \begin{pmatrix} \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^3 \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^1\delta^{+\infty} \end{pmatrix}.$$

After controlling the system with input $U_{MI,k}$ on step k of the active diagnosis session, the analysis of the effect of this input can be done by computing the set of delayed outputs $\mathcal{Y}_{del,U_{MI,k}}$.

Another relevant concept is the amount of time by which these outputs are delayed. Using the indicators introduced in Definition 12, for an output y_i, this is equal to $\overline{\mathcal{T}}_{\tilde{y}_i,y_i} = \max_{0\leq n<\Omega}(\mathcal{D}_{y_i}(n) - \mathcal{D}_{\tilde{y}_i}(n))$. However, any indicator that provides this information can be used.

Definition 24. *Let $D_{i,k}$ be the greatest time shift found on output y_i for step k:*

$$D_{i,k} = \max_{0\leq n<\Omega}(\mathcal{D}_{y_i}(n) - \mathcal{D}_{\tilde{y}_i}(n)).$$

It follows that $\mathrm{y}_i \in \mathcal{Y}_{del,U_{MI,k}} \Leftrightarrow D_{i,k} > 0$.

After all steps of an active diagnosis session are concluded, the following elements provide a summary of the information needed to make a decision concerning the place containing the failure.

Definition 25 (Greatest delay)**.** *The greatest delay for any input on any output, obtained after all the control steps of an active diagnosis algorithm are completed, is:*

$$D_{max} = \max_{i,k}(D_{i,k}).$$

Definition 26. *Let $\mathcal{U}_{MI}$ be the set of input transitions such that the associated control step generates the greatest delay, it is defined by:*

$$\mathcal{U}_{MI} = \{\mathrm{u}_j | \exists i, D_{i,j} = D_{max}\}.$$

Definition 27. *Let $\mathcal{Y}_{MI}$ be the set of delayed outputs obtained after all steps:*

$$\mathcal{Y}_{MI} = \mathcal{Y}_{del} \cup \bigcup_{k|\mathrm{u}_k \in \mathcal{U}_{del}} \mathcal{Y}_{del,U_{MI,k}},$$

where $\mathcal{Y}_{del}$ is the set of delayed outputs for original control U_{op}.

This set constitutes a new "global" set of delayed outputs that factor both the initial information and the information provided by the applied control sequences.

Example 20. *Continuing on the results of Examples 14 and 18, step 1 (input $U_{MI,1}$) results in the following expected output and measured output (the differences in dates are in bold):*

$$\tilde{Y} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{pmatrix} = \begin{pmatrix} \gamma^0\delta^{\mathbf{5}} \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^{\mathbf{6}} \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^{1} \oplus \gamma^1\delta^{+\infty} \end{pmatrix}, \qquad Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \gamma^0\delta^{\mathbf{6}} \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^{\mathbf{7}} \oplus \gamma^1\delta^{+\infty} \\ \gamma^0\delta^{1} \oplus \gamma^1\delta^{+\infty} \end{pmatrix}.$$

This leads to $D_{1,1} = 1$, $D_{2,1} = 1$, and $D_{3,1} = 0$.
The delays for the following steps are:

- *step 2:* $D_{1,2} = 1, \quad D_{2,2} = 1, \quad D_{3,1} = 0,$
- *step 3:* $D_{1,3} = 0, \quad D_{2,3} = 0, \quad D_{3,1} = 0,$

which results in $D = 1$.
The inputs that generate the delay $D_{max} = 1$ form the set $\mathcal{U}_{MI}$:

$$\mathcal{U}_{MI} = \{\mathrm{u}_1, \mathrm{u}_2\}.$$

In this example, the set of delayed outputs remains unchanged: $\mathcal{Y}_{MI} = \mathcal{Y}_{del,U_{op}} = \{\mathrm{y}_1, \mathrm{y}_2\}$.

Finally, since the purpose of the active diagnosis session is to localize the time failure, a set of candidate places must be defined. For CAMI, this set will be noted $\mathcal{P}_{CAMI}$.

Definition 28 (Set of candidate places found with CAMI)**.**

$$\mathcal{P}_{CAMI} = \{\mathrm{p}_m | \forall \mathrm{u}_j \in \mathcal{U}_{MI}, \forall \mathrm{y}_i \in \mathcal{Y}_{MI}, \mathrm{u}_j \rightsquigarrow \mathrm{p}_m \rightsquigarrow \mathrm{y}_i, \}.$$

$\mathcal{P}_{CAMI}$ is composed of the places upstream of all the delayed outputs and downstream of all the inputs in $\mathcal{U}_{MI}$.

Step-by-step description of CAMI

CAMI (Algorithm 1) takes an empty TEG $\mathcal{G}$ and the sets $\mathcal{Y}_{del}$ and $\mathcal{U}_{del}$ as inputs. It returns a set of candidate places.

CAMI runs as follows: after initializing Ω (line 1), the algorithm iterates on each step k (loop starting in line 2) to compute input vector $U_{MI,k}$ (line 3). It then applies this control to the system; the output Y is measured and the expected output $\tilde{Y}$ is computed (line 4). For each individual output in these vectors, the greatest time shift is computed (line 6). This will be used in the definition of D_{max} in line 9.

After all inputs in $\mathcal{U}_{del}$ have been studied, the sets $\mathcal{U}_{MI}$ and $\mathcal{Y}_{MI}$ are computed (lines 10 and 11 respectively).

CAMI returns a set of candidate places as defined in Definition 28.

Algorithm 1 Control Algorithm for Multiple Inputs (CAMI)

Input: $\mathcal{G}, \mathcal{Y}_{del}, \mathcal{U}_{del}$
Output: $\mathcal{P}_{CAMI}$
1: $\Omega \leftarrow 1 + \sum_{i=1}^{|P|} M_0(\mathrm{p}_i)$ ▷ Eq. 26
2: **for** $\mathrm{u}_k \in \mathcal{U}_{del}$ **do** ▷ Loop on each control step k
3: $U_{MI,k} \leftarrow ComputeU_{MI,k}(H, \Omega, k)$ ▷ Def. 23
4: $(Y, \tilde{Y}, I(U_{MI,k}, Y)) \leftarrow ApplyControl(U_{MI,k})$
5: **for** $\mathrm{y}_i \in \mathcal{Y}$ **do** ▷ Loop on each output
6: $D_{i,k} \leftarrow \max_{0 \leq n < \Omega}(\mathcal{D}_{y_i}(n) - \mathcal{D}_{\tilde{y}_i}(n))$ ▷ Def. 24
7: **end for**
8: **end for**
9: $D_{max} \leftarrow \max_{i,k}(D_{i,k})$ ▷ Def. 25
10: $\mathcal{U}_{MI} \leftarrow \{\mathrm{u}_j | \exists i, D_{i,j} = D_{max}\}$ ▷ Def. 26
11: $\mathcal{Y}_{MI} \leftarrow \mathcal{Y}_{del} \cup \bigcup_{k|\mathrm{u}_k \in \mathcal{U}_{del}} \mathcal{Y}_{del,U_{MI,k}}$ ▷ Def. 27
12: $\mathcal{P}_{CAMI} \leftarrow \{\mathrm{p}_m | \forall \mathrm{u}_j \in \mathcal{U}_{MI}, \forall \mathrm{y}_i \in \mathcal{Y}_{MI}, \mathrm{u}_j \rightsquigarrow \mathrm{p}_m \rightsquigarrow \mathrm{y}_i\}$ ▷ Def. 28
13: **return** $\mathcal{P}_{CAMI}$

Example 21. *Following Example 20 for the TEG in Fig. 6, the set of candidate places resulting from this algorithm is composed of the places downstream of* $\mathcal{U}_{MI} = \{\mathrm{u}_1, \mathrm{u}_2\}$ *and upstream of* $\mathcal{Y}_{MI} = \{\mathrm{y}_1, \mathrm{y}_2\}$*:*

$$\mathcal{P}_{CAMI} = \{\mathrm{p}_3, \mathrm{p}_5\}.$$

5.3 Results and properties of CAMI

Proposition 11. *If a time failure on a multiple-input TEG is detectable, there is a step* k *for which input vector* $U_{MI,k}$ *generates a delayed output:*

$$\exists k | (Y, \tilde{Y}, I(U_{MI,k}, Y) = True) \leftarrow ApplyControl(U_{MI,k}).$$

Proof. Let us begin by defining $\tilde{Y}$ and Y for an input $U_{MI,k}$. Let $U_{MI,k} \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{|\mathcal{U}|}$ be the input vector for step k of CAMI:

$$U_{MI,k} = \begin{pmatrix} r_1 \\ r_2 \\ \vdots \\ r_{|\mathcal{U}|} \end{pmatrix},$$

with r_j as defined in Definition 23.

For simplicity, let us consider an empty MISO system $\mathcal{G}$ and its expected transfer matrix $H \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{1\times|\mathcal{U}|}, |\mathcal{U}| > 1$. All the results that follow hold for MIMO systems (this proof can be repeated for each output). We have:

$$H = \begin{pmatrix} H_1 & \dots & H_{|\mathcal{U}|} \end{pmatrix}.$$

The presence of a detectable time failure in $\mathcal{G}$ results in a change of its dynamics and the system therefore behaves according to a new, unknown transfer matrix denoted $H' \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]^{1\times|\mathcal{U}|}$ different from H (see Corollary 5). For each step k, the expected output $\tilde{y} \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ and the measured output $y \in \mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ are therefore:

$$\tilde{y} = HU_{MI,k} = \begin{pmatrix} H_1 & H_2 & \dots & H_{|\mathcal{U}|} \end{pmatrix} \begin{pmatrix} r_1 \\ \vdots \\ r_{|\mathcal{U}|} \end{pmatrix} = H_1 r_1 \oplus H_2 r_2 \oplus ... \oplus H_{|\mathcal{U}|} r_{|\mathcal{U}|}.$$

$$y = H'U_{MI,k} = \begin{pmatrix} H'_1 & H'_2 & \dots & H'_{|\mathcal{U}|} \end{pmatrix} \begin{pmatrix} r_1 \\ \vdots \\ r_{|\mathcal{U}|} \end{pmatrix} = H'_1 r_1 \oplus H'_2 r_2 \oplus ... \oplus H'_{|\mathcal{U}|} r_{|\mathcal{U}|}.$$

Since $H' \neq H$, there exists at least one $H'_l \neq H_l$.

Let us develop $\tilde{y}$ for step k:

$$\tilde{y} = H_1 r_1 \oplus H_2 r_2 \oplus ... \oplus H_{|\mathcal{U}|} r_{|\mathcal{U}|}$$

$$\tilde{y} = H_k r_k \oplus \bigoplus_{j=1, j\neq k}^{|\mathcal{U}|} H_j r_j \qquad \text{(by grouping all } r_{j\neq k} \text{ in a sum)}$$

$$\tilde{y} = H_k(\gamma^0\delta^{t_{max,k}} \oplus \gamma^\Omega\delta^{+\infty}) \oplus \bigoplus_{j=1, j\neq k}^{|\mathcal{U}|} H_j(\gamma^0\delta^0 \oplus \gamma^\Omega\delta^{+\infty})$$

(by replacing r_j and r_k)

$$\tilde{y} = H_k\gamma^0\delta^{t_{max,k}} \oplus \bigoplus_{j=1, j\neq k}^{|\mathcal{U}|} H_j \oplus \bigoplus_{j=1}^{|\mathcal{U}|} H_j\gamma^\Omega\delta^{+\infty}.$$

(by factorizing by $\gamma^\Omega\delta^{+\infty}$, and $\gamma^0\delta^0 = e$)

By Assumption 6, the system is in phase, so it follows that $\forall i\, H_i \succeq e$, and $H_k \succeq e$ in particular. Due to the congruence that defines $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ (see

Definition 6), H_k can then be rewritten as an infinite sum of monomials starting on $\gamma^0\delta^{t_0}$:

$$H_k = \bigoplus_{n=0}^{\infty} \gamma^n \delta^{t_n}$$

$$\Rightarrow H_k \gamma^0 \delta^{t_{max,k}} = \bigoplus_{n=0}^{\infty} \gamma^n \delta^{t_n} \gamma^0 \delta^{t_{max,k}} = \bigoplus_{n=0}^{\infty} \gamma^n \delta^{t_n + t_{max,k}}.$$

Let us proceed similarly with any $H_j, j \neq k$:

$$H_j = \bigoplus_{m=0}^{\infty} \gamma^m \delta^{d_{j,m}}$$

$$\Rightarrow \bigoplus_{j=1, j\neq k}^{|\mathcal{U}|} H_j = \bigoplus_{j=1, j\neq k}^{|\mathcal{U}|} \bigoplus_{m=0}^{\infty} \gamma^m \delta^{d_{j,m}}.$$

The factor $\gamma^\Omega \delta^{+\infty}$ in the last term of $\tilde{y}$ absorbs all other factors H_j in the sum:

$$\bigoplus_{j=1}^{|\mathcal{U}|} H_j \gamma^\Omega \delta^{+\infty} = \bigoplus_{j=1}^{|\mathcal{U}|} \bigoplus_{m=0}^{\infty} \gamma^m \delta^{d_{j,m}} \gamma^\Omega \delta^{+\infty}$$

$$\Leftrightarrow \bigoplus_{j=1}^{|\mathcal{U}|} H_j \gamma^\Omega \delta^{+\infty} = \bigoplus_{j=1}^{|\mathcal{U}|} \bigoplus_{m=0}^{\infty} \gamma^{m+\Omega} \delta^{d_{j,m}+\infty}$$

$$\Leftrightarrow \bigoplus_{j=1}^{|\mathcal{U}|} H_j \gamma^\Omega \delta^{+\infty} = \bigoplus_{j=1}^{|\mathcal{U}|} \bigoplus_{m=0}^{\infty} \gamma^{m+\Omega} \delta^{+\infty}$$

$$\Leftrightarrow \bigoplus_{j=1}^{|\mathcal{U}|} H_j \gamma^\Omega \delta^{+\infty} = \bigoplus_{m=0}^{\infty} \gamma^{m+\Omega} \delta^{+\infty}$$

(the addition in $\mathcal{M}_{in}^{ax}[\![\gamma, \delta]\!]$ is idempotent and the terms don't depend on j)

$$\Leftrightarrow \bigoplus_{j=1}^{|\mathcal{U}|} H_j \gamma^\Omega \delta^{+\infty} = \gamma^{\min_{m\geq 0}(m+\Omega)} \delta^{+\infty}$$

(adition of monomials with the same date)

$$\Leftrightarrow \bigoplus_{j=1}^{|\mathcal{U}|} H_j \gamma^\Omega \delta^{+\infty} = \gamma^\Omega \delta^{+\infty}.$$

Therefore, we can rewrite $\tilde{y}$ as:

$$\tilde{y} = \bigoplus_{n=0}^{\infty} \gamma^n \delta^{t_n + t_{max,k}} \oplus \bigoplus_{j=1, j\neq k}^{|\mathcal{U}|} \bigoplus_{m=0}^{\infty} \gamma^m \delta^{d_{j,m}} \oplus \gamma^\Omega \delta^{+\infty}.$$

All terms for $n \geq \Omega$ and $m \geq \Omega$ are absorbed by $\gamma^{\Omega}\delta^{+\infty}$:

$$\tilde{y} = \bigoplus_{n=0}^{\Omega-1} \gamma^{n}\delta^{t_n+t_{max,k}} \oplus \bigoplus_{j=1,j\neq k}^{|\mathcal{U}|} \bigoplus_{m=0}^{\Omega-1} \gamma^{m}\delta^{d_{j,m}} \oplus \gamma^{\Omega}\delta^{+\infty}.$$

Additionally, we have $t_{max,k} > d_{j,m}\, \forall m < \Omega$ (see Definition 22), so the second term in $\tilde{y}$ is absorbed by the first term and:

$$\tilde{y} = \bigoplus_{n=0}^{\Omega-1} \gamma^{n}\delta^{t_n+t_{max,k}} \oplus \gamma^{\Omega}\delta^{+\infty}. \tag{29}$$

Therefore, the value of the expected output $\tilde{y}$ for step k does not depend on any $d_{j,n}$, which are the expected traversal times for $n+1$ tokens from input $\mathrm{u}_{j\neq k}$ to the output, or, equivalently, the daters of series $H_{j\neq k}$ for event n.

The development of $y = H'U_{MI,k}$ is similar except for the last passage because whether $t_{max,k}$ is greater than the daters of H'_j is unknown. With $H'_k = \bigoplus_{n=0}^{\infty} \gamma^{n}\delta^{t'_n}$ and $H'_j = \bigoplus_{m=0}^{\infty} \gamma^{m}\delta^{d'_{j,m}}$, we have:

$$y = \bigoplus_{n=0}^{\Omega-1} \gamma^{n}\delta^{t'_n+t_{max,k}} \oplus \bigoplus_{j=1,j\neq k}^{|\mathcal{U}|} \bigoplus_{m=0}^{\Omega-1} \gamma^{m}\delta^{d'_{j,m}} \oplus \gamma^{\Omega}\delta^{+\infty}.$$

As previously stated, a detectable time failure being present in the system means that at least one transfer function H'_l is different from its expected counterpart H_l. The number of events Ω as defined in Equation (26) ensures that the duration of all places intervene in the dates of the output series when possible, as discussed in Section 5.2. This number of events is therefore sufficient to detect any time failures, so there exists $n < \Omega$ such that $t'_n > t_n$ or $d'_n > d_n$ or both (since the time failure causes a delay).

Let us name $d'_{M,m} = \max(d'_{j\neq k,m})$ for a given m. This is by definition the greatest dater among those of all series $H'_{j\neq k}$ for event m, or, equivalently, the greatest traversal time for $m+1$ tokens from the inputs $\mathrm{u}_{j\neq k}$ to the output. We have:

$$y = \bigoplus_{n=0}^{\Omega-1} \gamma^{n}\delta^{\max(d'_{M,n},t'_n+t_{max,k})} \oplus \gamma^{\Omega}\delta^{+\infty}, \tag{30}$$

by grouping the sums on n and m according to the rules for the addition of monomials in $\mathcal{M}_{in}^{ax}[\![\gamma,\delta]\!]$. The value of the measured output y for step k therefore depends on both $d'_{M,n}$ and t'_n (the latter being the dater of series H_k for event n and representing the traversal time for $n+1$ tokens from u_k to the output).

Regarding the date $\max(d'_{M,n}, t'_n + t_{max,k})$ in Equation (30), if it is equal to $d'_{M,n}$, then there exists another a step during which t'_n will receive that value and the max will be equal to $t'_n + t_{max,k}$.

Then, since there exists a step k and an event number n such that $t'_n > t_n$, y will be different from $\tilde{y}$ and the time failure will be detected.

□

Proposition 12. *All inputs in $\mathcal{U}_{MI}$ are upstream of the place containing the failure p_f:* $\mathrm{u}_j \in \mathcal{U}_{MI} \Rightarrow \mathrm{u}_j \rightsquigarrow \mathrm{p}_f$.

Proof. The proof can be done in two steps:

- Determining the requirements for an input transition to belong to $\mathcal{U}_{MI}$.
- Establishing the implication that such an input transition is upstream of the place containing the failure p_f.

The expected output $\tilde{y}$ and the output y for the input $U_{MI,k}$ were defined in Equations 29 and 30 respectively. Now let us determine under which circumstances an input transition belongs to $\mathcal{U}_{MI}$.

We are interested in the greatest time shift between $\tilde{y}$ and y, D_k. This is equal to the difference between the dates of y and $\tilde{y}$ for the same event, as shown in Definition 24. For a given step k:

$$D_k = \max_{0 \leq n < \Omega} (\mathcal{D}_y(n) - \mathcal{D}_{\tilde{y}}(n))$$

$$\Leftrightarrow D_k = \max_{0 \leq n < \Omega} (\max(d'_{M,n}, t'_n + t_{max,k}) - (t_n + t_{max,k})),$$

where n is an event number from series $\tilde{y}$ and y.

By definition, input u_k belongs to $\mathcal{U}_{MI}$ if and only if D_k is equal to the greatest upper bound found with CAMI, D_{max}.

We can separate the results in two cases depending on the value of $\max(d'_{M,n}, t'_n + t_{max,k})$:

- $d'_{M,n} > t'_n + t_{max,k}$: this means that there is an input $\mathrm{u}_{j \neq k}$ for which a subsequent step j will generate $t'_n > d'_{M,n}$, which will then fall under the next case. Regarding step k, we will obtain $D_k = \max_{0 \leq n < \Omega}(d'_{M,n} - t_n - t_{max,k})$. This will be equal to the greatest delay D_{max} for all steps if:

  $$\max_{0 \leq n < \Omega} (d'_{M,n} - t_n - t_{max,k}) \geq \max_{0 \leq n < \Omega,\, 1 \leq j \leq |\mathcal{Y}|} (d'_{j,n} - d_{j,n}),$$

  since the term on the right side of the inequality will be the value of D_j at a subsequent step j (see next case).

  For a given j, $d'_{j,n} = d'_{M,n}$, so

  $$\max_{0 \leq n < \Omega} (d'_{M,n} - t_n - t_{max,k}) \geq \max_{0 \leq n < \Omega} (d'_{M,n} - d_{M,n})$$

  $$\Leftrightarrow \max_{0 \leq n < \Omega} (-t_n - t_{max,k}) \geq \max_{0 \leq n < \Omega} (-d_{M,n}). \tag{31}$$

  Yet, by definition, $t_n \geq 0$ and $t_{max,k} > d_{M,n}\ \forall n$, so we have

  $$t_n + t_{max,k} > d_{M,n} \quad \forall n$$

$$\Leftrightarrow -t_n - t_{max,k} < -d_{M,n} \quad \forall n,$$

and Inequality 31 does not hold. Therefore, in this case, u_k cannot belong to $\mathcal{U}_{MI}$.

- $t'_n + t_{max,k} \geq d'_{M,n}$: this results in $D_k = \max_{0 \leq n < \Omega}(t'_n - t_n)$. Since there is at least one step for which $t'_n > t_n$ (because a time failure has been detected), for this expression to be the greatest delay D_{max} for all steps (and obtain $\mathrm{u}_k \in \mathcal{U}_{MI}$), t'_n must be greater than t_n.

Therefore, for u_k to belong to $\mathcal{U}_{MI}$, there needs to exist n such that $t'_n > t_n$. This means that the unknown transfer function H'_k is different from the expected transfer function H_k, so there is a failure on a place downstream of u_k: $\mathrm{u}_k \rightsquigarrow \mathrm{p}_f$. □

Note: there may be inputs upstream of the place containing the failure that do not belong to $\mathcal{U}_{MI}$.

Since the place containing the time failure is downstream of all inputs in $\mathcal{U}_{MI}$, it can be localized more precisely via a structural analysis by calculating the intersection of the sets of places downstream of the inputs in this set. This is what is done in the definition of $\mathcal{P}_{CAMI}$ (Definition 28).

Proposition 13. *CAMI is sound: the place containing the time failure* p_f *is necessarily contained in the resulting set:* $\mathrm{p}_f \in \mathcal{P}_{CAMI}$.

Proof. As stated in Proposition 7, the place containing the failure p_f is upstream of all delayed outputs for a given input U: $\mathrm{y}_i \in \mathcal{Y}_{del,U} \Rightarrow \mathrm{p}_f \rightsquigarrow \mathrm{y}_i$. Since the time failure is permanent, this also holds for the set of delayed outputs for the different control steps of CAMI, $\mathcal{Y}_{MI}$: $\mathrm{y}_i \in \mathcal{Y}_{MI} \Rightarrow \mathrm{p}_f \rightsquigarrow \mathrm{y}_i$. Proposition 12 states that the place containing the failure is downstream of all the inputs in $\mathcal{U}_{MI}$: $\mathrm{u}_j \in \mathcal{U}_{MI} \Rightarrow \mathrm{u}_j \rightsquigarrow \mathrm{p}_f$. Since $\mathcal{P}_{CAMI}$ is combination of both of these results, the place where the time failure occurs is necessarily contained in this set: $\mathrm{p}_f \in \mathcal{P}_{CAMI}$. □

Proposition 14. *The set of candidates obtained with CAMI is a subset of the set obtained with the structure-based localization for original control* U*:* $\mathcal{P}_{CAMI} \subseteq \mathit{Loc}(\mathcal{G}, U, Y)$.

Proof. Let us divide the definition of $\mathcal{P}_{CAMI}$ in two parts:

$$\mathcal{P}_{CAMI} = \{\mathrm{p}_m | \forall \mathrm{y}_i \in \mathcal{Y}_{MI}, \mathrm{p}_m \rightsquigarrow \mathrm{y}_i, \} \cap \{\mathrm{p}_m | \forall \mathrm{u}_j \in \mathcal{U}_{MI}, \mathrm{u}_j \rightsquigarrow \mathrm{p}_m\}.$$

Let us name the set on the left $\mathcal{P}_{\mathcal{Y}}$: $\mathcal{P}_{\mathcal{Y}} = \{\mathrm{p}_m | \forall \mathrm{y}_i \in \mathcal{Y}_{MI}, \mathrm{p}_m \rightsquigarrow \mathrm{y}_i, \}$. Since $\mathcal{Y}_{MI} = \mathcal{Y}_{del} \cup \bigcup_{k | \mathrm{u}_k \in \mathcal{U}_{del}} \mathcal{Y}_{del,U_{MI,k}}$, $\mathcal{P}_{\mathcal{Y}}$ can also be defined as:

$$\mathcal{P}_{\mathcal{Y}} = \{\mathrm{p}_m | \forall \mathrm{y}_i \in \mathcal{Y}_{del}, \mathrm{p}_m \rightsquigarrow \mathrm{y}_i, \} \cap \{\mathrm{p}_m | \forall \mathrm{y}_i \in \mathcal{Y}_{del,U_{MI,k}}, \mathrm{p}_m \rightsquigarrow \mathrm{y}_i, \}$$

$$\Leftrightarrow \mathcal{P}_{\mathcal{Y}} = Loc(\mathcal{G}, U, Y) \cap \bigcap_{k} Loc(\mathcal{G}, U_{MI,k}, Y). \qquad \text{(by Definition 14)}$$

$\mathcal{P}_{CAMI}$ is an intersection involving $\mathcal{P}_{\mathcal{Y}}$, and $\mathcal{P}_{\mathcal{Y}}$ is an intersection involving $Loc(\mathcal{G}, U, Y)$, so:

$$\mathcal{P}_{CAMI} \subseteq \mathcal{P}_{\mathcal{Y}} \text{ and } \mathcal{P}_{\mathcal{Y}} \subseteq Loc(\mathcal{G}, U, Y) \Rightarrow \mathcal{P}_{CAMI} \subseteq Loc(\mathcal{G}, U, Y).$$

Hence the result. □

5.4 Localization method

The localization method is shown in Algorithm 2. It is a combination of an analysis that uses the definitions and propositions in Section 5.1 and the control algorithm in Section 5.2 (CAMI). Its inputs are an empty TEG denoted $\mathcal{G}$, control U_{op} and its associated measured output Y.

The algorithm can be briefly described as follows: let us consider an empty TEG $\mathcal{G}$ in which a time failure has been detected for control U_{op}. If $\mathcal{G}$ is a single-input TEG, the resulting set of candidates is $Loc(\mathcal{G}, U_{op}, Y)$ (Definition 14). Otherwise, a study of the input transitions of $\mathcal{G}$ using the properties in Section 5.1 and potentially the control algorithm in Section 5.2 will be executed.

More precisely, the algorithm begins by defining the resulting set of outputs as $Loc(\mathcal{G}, U_{op}, Y)$ if the TEG only has one input (line 2). Otherwise, the set $\mathcal{U}_{del,U_{op}}$ is defined (line 5) using $\mathcal{Y}_{del,U_{op}}$ (defined in line 4). This is equivalent to determining whether there are multiple inputs upstream of all the delayed outputs obtained for control U_{op}.

If the set $\mathcal{U}_{del,U_{op}}$ is a singleton, then the set of candidates is once again $Loc(\mathcal{G}, U_{op}, Y)$ (line 7); however, if $\mathcal{U}_{del,U_{op}}$ contains more than once input, CAMI is executed and provides the set of candidates (line 9).

Proposition 15. *The localization method for multiple-input TEGs is sound.*

Proof. The resulting set of the active localization $\mathcal{P}_{cand}$ is defined as either $Loc(\mathcal{G}, U_{op}, Y)$ or $\mathcal{P}_{CAMI}$; since both of these sets result from sound algorithms, the active localization for multiple-input TEG is also sound.

6 Extension for circuits

6.1 Overview

This section introduces an extension of the localization method described in the previous section. The extension is called Active Time Failure Localization Algorithm for TEG (ATFLAT) and it is aimed at reducing the set of candidate places for TEG that contain circuits, whether they contain multiple inputs or not.

Algorithm 2 Time Failure Localization Method

Input: $\mathcal{G}, U_{op}, Y$
Output: $\mathcal{P}_{cand}$

if $|\mathcal{U}| = 1$ **then**
  $\mathcal{P}_{cand} \leftarrow Loc(\mathcal{G}, U_{op}, Y)$ ▷ Def. 14
else
  $\mathcal{Y}_{del,U_{op}} \leftarrow \{\mathrm{y_i} \in \mathcal{Y} | I(U_{op}, y_i) = true\}$ ▷ Def. 13
  $\mathcal{U}_{del,U_{op}} \leftarrow \{\mathrm{u}_k | \forall \mathrm{y}_i \in \mathcal{Y}_{del,U_{op}}, \mathrm{u}_k \rightsquigarrow \mathrm{y}_i\}$ ▷ Def. 20
  if $|\mathcal{U}_{del,U_{op}}| = 1$ **then**
    $\mathcal{P}_{cand} \leftarrow Loc(\mathcal{G}, U_{op}, Y)$
  else
    $\mathcal{P}_{cand} \leftarrow \mathrm{CAMI}(\mathcal{G}, \mathcal{Y}_{del,U_{op}}, \mathcal{U}_{del,U_{op}})$ ▷ Algorithm 1
  end if
end if
return $\mathcal{P}_{cand}$

This section is divided in four parts. Section 6.2 introduces properties related to the presence of time failures in elementary paths and circuits. Section 6.3 describes a localization algorithm which exploits these analytical properties. Section 6.4 introduces an extension of CAMI that improves the localization by exploiting an analysis of circuits. Finally, Section 6.5 explains the global algorithm ATFLAT.

6.2 Localization of failures in TEGs that contain circuits

The objective of the proposed extension is to reduce the set of candidate places that is proposed by the localization method of Section 5, either by differentiating places that are contained in circuits from those that are not, or by distinguishing between circuits holding different amounts of tokens. ATFLAT relies on the following properties of a TEG. The next property offers a way to differentiate failing places that are part of elementary paths between inputs and outputs from those that aren't.

Proposition 16. *For a given control U, a time failure can be detected as soon as a first event (event number zero) occurs in an output $\mathrm{y}_i \in \mathcal{Y}$ if and only if the time failure is on a place contained in the elementary path from an input transition u to the output y_i:*

$$\exists i | \mathcal{D}_{y_i}(0) \neq \mathcal{D}_{\tilde{y}_i}(0) \Leftrightarrow \exists \mathrm{u} | \mathrm{p}_f \in \bigcup_{\pi \in \Pi_{\mathcal{E}}(\mathrm{u}, \mathrm{y}_i)} \mathcal{P}_{\mathcal{E}}(\pi).$$

Proof. ($\Rightarrow$) As $\mathcal{D}_{y_i}(0) \neq \mathcal{D}_{\tilde{y}_i}(0)$ and TEG $\mathcal{G}$ is in phase and empty, the first event occurrence of y_i depends on the first event occurrence of any input u such that $\mathrm{u} \hookrightarrow \mathrm{y}_i$. Suppose that p_f does not belong to at least one elementary path from such an input u to the output y_i, it means that it

is in a circuit and by the firing rules of a TEG, the holding time of p_f is not involved in the propagation of the first event of u. Contradiction.

($\Leftarrow$) If p_f is in an elementary path between input u and output y, as TEG $\mathcal{G}$ is in phase and empty, the result is straightforward.

□

The second proposition provides a way to distinguish between a failing place that is in an elementary circuit holding n tokens and a failing place that is in an elementary circuit holding $m < n$ tokens. Let $\mathcal{C}_{\mathcal{E}}(n)$ be the set of elementary circuits of $\mathcal{C}_{\mathcal{E}}$ such that they contain a place p with $M_0(\mathrm{p}) = n$.

Proposition 17. *Suppose* p_f *is in an elementary circuit of* $\mathcal{C}_{\mathcal{E}}(n)$ *and not part of an elementary path between an input and an output of the TEG. Suppose also that* p_f *is not part of any elementary circuit of* $\mathcal{C}_{\mathcal{E}}(m), m < n$*. The time failure of* p_f *cannot be detected before the* $(n+1)^{th}$ *event (event number* n*) on any output of the TEG.*

Proof. Let $C \in \mathcal{C}_{\mathcal{E}}(n)$ be one of the elementary circuits containing p_f. Either Circuit C contains at least a synchronization transition $\mathrm{t} \in \mathcal{T}$ involved in a path $\pi \in \Pi_{\mathcal{E}}(\mathrm{u}, \mathrm{y}), \mathrm{u} \in \mathcal{U}, \mathrm{y} \in \mathcal{Y}$ or not. If t exists, by the firing rules of a TEG, the time failure of p_f has an effect only on the $(n+1)^{th}$ fire of transition t so then potentially on the $(n+1)^{th}$ fire of transition y. If C does not contain such a transition t, it contains at least a synchronization transition t' involved in another circuit. For the same reason as above, the time failure of p_f has an effect only on the $(n+1)^{th}$ fire of transition t' and still no effect on transitions involved in elementary paths from an input to an output. □

Corollary 18. *If a time failure is detected on event* $m > 0$ *of an output, then it cannot be on a place which is only contained in elementary circuits with* $n > m$ *tokens.*

Proof. Direct consequence of Proposition 17. □

For a failure detected on the $(n+1)^{th}$ event, Proposition 17 and Corollary 18 prune out places that are *only* contained in elementary circuits with more than n tokens. p_f must be in *at least* one elementary circuit with n or fewer tokens, but may also belong to other circuits with more tokens.

6.3 Analysis Algorithm for Circuits

Algorithm 3 (denoted CircuitAnalysis) details the Analysis Algorithm for Circuits. This algorithm is part of the global localization algorithm ATFLAT. Its first objective is to return a set of candidate places $\mathcal{P}_{circ}$ by exploiting Proposition 16 and Corollary 18 on the current measured output Y and the expected output $\tilde{Y}$ at a given step k. It also returns two boolean variables: TestCircuits and TestContinue.

TestCircuits is True as long as it is necessary to analyze circuits for the localization of candidate places. It is set to False as soon as the analysis concludes that:

1. the candidate place must be in an elementary path (by Proposition 16); or
2. $\mathcal{P}_{circ}$ only contains one place; or
3. $\mathcal{P}_{circ}$ contains only one circuit of $\mathcal{C}_{\mathcal{E}}(m), m \leq n$ while the failure is detected on a output y_i with event number n (by Corollary 18).

The boolean variable TestContinue is true as long as the analysis does not provide any definitive conclusions (that are Cases 2 and 3 here above): if TestContinue is set to False, so is TestCircuits.

The analysis starts with a loop that iterates on each output (line 6). It then defines the elementary paths from any input to the current output (line 7) and exploits Proposition 16 to define the set of candidates if the first event of the output (event zero) presents a time shift (line 9). In this case, TestCircuits is set to False (line 10). Otherwise, if the place containing the failure may be in a circuit, it iterates on the events of the series (n) (loop starting in line 13) in order to reduce the set of candidate places to those in circuits containing fewer tokens than the number of the event presenting a time shift (lines 17 and 18), which exploits Corollary 18.

TestContinue is set to False if the resulting set contains only one place or if there is only one circuit with fewer tokens than the number of the first event where the time shift is detected (line 20). This is done because all the places in a circuit are downstream of the same input transitions and upstream of the same output transitions, no additional information can come from the available data (which is the input series and the output series). TestContinue being set to False is an exit condition for the loop iterating on the outputs. Otherwise, the algorithm continues to iterate to attempt to further reduce the set of candidates $\mathcal{P}_{circ}$.

Example 22. *To illustrate the proposed circuit analysis, let us again consider the TEG in Fig. 1. This multiple-input multiple-output TEG is similar to the TEG in Fig. 6 (see the examples in Section 5), but six places contained in circuits have been added, namely:* p_{12}, p_{13}, p_{14}, p_{15}, p_{16} *and* p_{17}. *Note that* p_5 *is now contained in a circuit as well.*

There are five elementary circuits in this TEG. Three of them contain two tokens (on transitions x_1, x_2, x_3, x_6 *and* x_5*) and two of them contain three tokens (on transitions* x_6 *and* x_4*). For this TEG,* $\Omega = 13$ *(see Equation (26)).*

Its transfer matrix is:

$$H = \begin{pmatrix} \gamma^0\delta^1(\gamma^2\delta^2)^* & \gamma^0\delta^1(\gamma^2\delta^2)^* & \gamma^0\delta^2(\gamma^2\delta^2)^* & \varepsilon \\ \gamma^0\delta^2(\gamma^2\delta^2)^* & \gamma^0\delta^2(\gamma^2\delta^2)^* & \gamma^0\delta^3(\gamma^2\delta^2)^* & \gamma^0\delta^2(\gamma^2\delta^1)^* \\ \varepsilon & \varepsilon & \varepsilon & \gamma^0\delta^1(\gamma^2\delta^1)^* \end{pmatrix}. \tag{32}$$

Algorithm 3 Analysis Algorithm for Circuits (CircuitAnalysis)

Input: $\mathcal{G}, Y, \tilde{Y}$
Output: $\mathcal{P}_{circ}$, TestCircuits, TestContinue

$\mathcal{P}_{circ} \leftarrow \mathcal{P}$
$\Omega \leftarrow 1 + \sum_{i=1}^{|P|} M_0(\mathrm{p}_i)$ ▷ Eq. 26
TestCircuits $\leftarrow True$
TestContinue $\leftarrow True$
$i \leftarrow 1$
while $(i \leq |\mathcal{Y}|)$ and (TestContinue) **do** ▷ Loop on each output
  $\mathcal{P}_{\mathcal{E}} \leftarrow \bigcup_{\pi \in \Pi_{\mathcal{E}}(\mathrm{u}_j, \mathrm{y}_i)} \mathcal{P}_{\mathcal{E}}(\pi) \forall \mathrm{u}_j$
  if $\mathcal{D}_{y_i}(0) \neq \mathcal{D}_{\tilde{y}_i}(0)$ **then**
    $\mathcal{P}_{circ} \leftarrow \mathcal{P}_{circ} \cap \mathcal{P}_{\mathcal{E}}$ ▷ Exploits Prop. 16
    TestCircuits $\leftarrow False$
  else if TestCircuits **then**
    $n \leftarrow 1$
    while $(n < \Omega)$ and $(\mathcal{D}_{y_i}(n) = \mathcal{D}_{\tilde{y}_i}(n))$ **do** ▷ Loop on the events
      $n \leftarrow n + 1$
    end while
    if $\mathcal{D}_{y_i}(n) \neq \mathcal{D}_{\tilde{y}_i}(n)$ **then**
      $\mathcal{P}_n \leftarrow \bigcup_{m=1}^{n-1} \mathcal{C}_{\mathcal{E}}(m) \cap \{p | p \rightsquigarrow \mathrm{y}_i\}$
      $\mathcal{P}_{circ} \leftarrow \mathcal{P}_{circ} \cap (\mathcal{P}_n \setminus \mathcal{P}_{\mathcal{E}})$ ▷ Exploits Cor. 18
      if $(|\mathcal{P}_{circ}| = 1)$ or $(\exists! \mathrm{p} \in \mathcal{P}_{circ} | \, 0 < M_0(\mathrm{p}) \leq n)$ **then**
        TestContinue $\leftarrow False$
        TestCircuits $\leftarrow False$
      end if
    end if
  end if
  $i \leftarrow i + 1$
  if $|\mathcal{P}_{circ}| = 1$ **then** ▷ If there is only one candidate place
    TestContinue $\leftarrow False$
    TestCircuits $\leftarrow False$
  end if
end while
return ($\mathcal{P}_{circ}$, TestCircuits, TestContinue)

Let us consider a time failure of 3 *time units on place* p_{17} *(this information is unknown). The following input and output are measured:*

$$U = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{pmatrix} = \begin{pmatrix} \gamma^0\delta^1 \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^1 \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^1 \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^1 \oplus \gamma^6\delta^{+\infty} \end{pmatrix}, \; Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \gamma^0\delta^3 \oplus \gamma^2\delta^5 \oplus \gamma^4\delta^7 \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^4 \oplus \gamma^2\delta^{\mathbf{7}} \oplus \gamma^4\delta^{\mathbf{11}} \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^2\delta^{\mathbf{6}} \oplus \gamma^4\delta^{\mathbf{10}} \oplus \gamma^6\delta^{+\infty} \end{pmatrix}.$$

The expected output for this input is:

$$\tilde{Y} = \begin{pmatrix} \tilde{y}_1 \\ \tilde{y}_2 \\ \tilde{y}_3 \end{pmatrix} = \begin{pmatrix} \gamma^0\delta^3 \oplus \gamma^2\delta^5 \oplus \gamma^4\delta^7 \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^4 \oplus \gamma^2\delta^{\mathbf{6}} \oplus \gamma^4\delta^{\mathbf{8}} \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^2\delta^{\mathbf{3}} \oplus \gamma^4\delta^{\mathbf{4}} \oplus \gamma^6\delta^{+\infty} \end{pmatrix}.$$

The differences in dates between the expected output and the measured output are in bold. In the analysis for circuits, the algorithm begins with $i = 1$ *by computing* $\mathcal{P}_{\mathcal{E}} = \{\mathrm{p}_1, ..., \mathrm{p}_6\}$*. It then starts comparing* y_1 *and* $\tilde{y}_1$ *event by event and no time shift is found, so* $\mathcal{P}_{circ}$ *remains unchanged as the whole set of places* $\mathcal{P}$*.*

For $i = 2$*,* $\mathcal{P}_{\mathcal{E}} = \{\mathrm{p}_1, ..., \mathrm{p}_5, \mathrm{p}_7, ..., \mathrm{p}_{10}\}$*. There is no difference between the dates for the first event; the first difference is found for the third event (denoted 2). This results in the set* $\mathcal{P}_2 = \{\mathrm{p}_5, \mathrm{p}_{12}, \mathrm{p}_{13}, \mathrm{p}_{15}, \mathrm{p}_{17}\}$*. The set of candidates is therefore* $\mathcal{P}_{circ} = \{\mathrm{p}_{12}, \mathrm{p}_{13}, \mathrm{p}_{15}, \mathrm{p}_{17}\}$*. Since there are three places containing 2 tokens in this set (*$\mathrm{p}_{12}, \mathrm{p}_{15}$ *and* p_{17}*) (which means there are three elementary circuits with two tokens upstream of* y_2*), the algorithm continues.*

It proceeds similarly for $i = 3$ *up until line 18. In this case,* $\mathcal{P}_{circ}$ *becomes a singleton at that point (additionally, there is only one circuit containing two tokens upstream of* y_3*), so the algorithm stops at* $n = 2$*. The resulting set of candidates is* $\mathcal{P}_{circ} = \{\mathrm{p}_{17}\}$*. TestContinue and TestCircuits are both returned set to False.*

Proposition 19. *The analysis algorithm for circuits is sound:* $\mathrm{p}_f \in \mathcal{P}_{circ}$*.*

Proof. This algorithm exploits the results regarding p_f and the sets $\mathcal{P}_{\mathcal{E}}$ and $\mathcal{C}_{\mathcal{E}}$ proven in Proposition 16 and Corollary 18. The algorithm keeps removing from $\mathcal{P}_{circ}$ places that cannot be p_f. □

Proposition 20. *The set of candidate places returned by CircuitAnalysis is a subset of* $Loc(\mathcal{G}, U, Y)$*:* $\mathcal{P}_{circ} \subseteq Loc(\mathcal{G}, U, Y)$*.*

Proof. $Loc(\mathcal{G}, U, Y)$ is defined as the set of places upstream of all the delayed outputs. The analysis algorithm for circuits initializes the set of candidates as the entire set of places, and for each output y_i, it intersects it with another set of places if a time shift is found, which means the output is delayed. The sets that are used in the intersections are all subsets of $\{\mathrm{p} | \mathrm{p} \rightsquigarrow \mathrm{y}_i\}$ (the set of places leading to y_i), so the resulting set is an intersection of the sets of places upstream of each of the delayed outputs, which means it is a subset of $Loc(\mathcal{G}, U, Y)$. □

6.4 Control Algorithm for Multiple Inputs and Circuits (CAMIC)

The algorithm CircuitAnalysis performs an analysis in the context of an active diagnosis session as described in Section 4. Therefore, it is suitable to be integrated in a control algorithm in the case where the investigated TEG has multiple inputs.

The Control Algorithm for Multiple Inputs and Circuits (CAMIC) proposed in this section is a modified version of CAMI that takes into account the fact that there may be elementary circuits in the diagnosed TEG. The steps of CAMIC are very similar to those in CAMI, except that at each step k, as an analysis for circuits is performed and may be conclusive (with TestContinue set to false), CAMIC may end before performing all the steps that CAMI would have carried out.

CAMIC is detailed in Algorithm 4. The main novelties when compared with CAMI are the circuit analysis in line 12, which may modify the TestContinue variable (to stop all further testing), the TestCircuits variable (to stop the testing on circuits), and the set of candidates (to reduce it according the analysis of the previous section). If TestContinue is kept as True until the end of the loop on the inputs, it means that CircuitAnalysis is not conclusive but has filtered out some places that are certainly not holding the time failure. In this case, all the control steps have been performed and CAMIC has also computed the set of candidate places as CAMI would have done. So, the final result is the intersection of the candidate places in $\mathcal{P}_{CAMI}$ and the ones from CircuitAnalysis (lines 25 and 26).

Example 23. *Let us consider the TEG in Fig. 1 and its transfer function H (32). Let us now consider a time failure of* 1 *time unit on place* p_{13}*. For the input in Example 22, the following output is measured (date differences are in bold):*

$$Y = \begin{pmatrix} y_1 \\ y_2 \\ y_3 \end{pmatrix} = \begin{pmatrix} \gamma^0\delta^3 \oplus \gamma^2\delta^{\mathbf{6}} \oplus \gamma^4\delta^{\mathbf{9}} \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^4 \oplus \gamma^2\delta^{\mathbf{7}} \oplus \gamma^4\delta^{\mathbf{10}} \oplus \gamma^6\delta^{+\infty} \\ \gamma^0\delta^2 \oplus \gamma^2\delta^3 \oplus \gamma^4\delta^4 \oplus \gamma^6\delta^{+\infty} \end{pmatrix}.$$

This results in the set of delayed outputs $\mathcal{Y}_{del} = \{\mathrm{y}_1, \mathrm{y}_2\}$ *and the associated set of inputs* $\mathcal{U}_{del} = \{\mathrm{u}_1, \mathrm{u}_2, \mathrm{u}_3\}$.

The inputs computed by CAMIC are:

$$U_{MI,1} = \begin{pmatrix} \gamma^0\delta^{16} \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \end{pmatrix}, \quad U_{MI,2} = \begin{pmatrix} \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^{16} \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \end{pmatrix},$$

$$U_{MI,3} = \begin{pmatrix} \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^{15} \oplus \gamma^{13}\delta^{+\infty} \\ \gamma^0\delta^0 \oplus \gamma^{13}\delta^{+\infty} \end{pmatrix}.$$

Algorithm 4 Control Algorithm for Multiple Inputs and Circuits (CAMIC)

Input: $\mathcal{G}, \mathcal{Y}_{del}, \mathcal{U}_{del}$
Output: $\mathcal{P}_{CAMIC}$

$\Omega \leftarrow 1 + \sum_{i=1}^{|P|} M_0(\mathrm{p}_i)$ ▷ Eq. 26
TestCircuits $\leftarrow True$
TestContinue $\leftarrow True$
$\mathcal{P}_{CAMIC} \leftarrow \mathcal{P}$
$\mathcal{U}_{left} \leftarrow \mathcal{U}_{del}$
while ($\mathcal{U}_{left} \neq \varnothing$) and (TestContinue) **do** ▷ Loop on each control step k
    Let $\mathrm{u}_k \in \mathcal{U}_{left}$
    $\mathcal{U}_{left} \leftarrow \mathcal{U}_{del} \setminus \{\mathrm{u}_k\}$
    $U_{MI,k} \leftarrow ComputeU_{MI,k}(H, \Omega, k)$ ▷ Def. 23
    $(Y, \tilde{Y}, I(U_{MI,k}, Y)) \leftarrow ApplyControl(U_{MI,k})$
    if TestCircuits **then**
        ($\mathcal{P}_{circ}$, TestCircuits, TestContinue) $\leftarrow CircuitAnalysis(\mathcal{G}, Y, \tilde{Y})$
        $\mathcal{P}_{CAMIC} \leftarrow \mathcal{P}_{CAMIC} \cap \mathcal{P}_{circ}$
    end if
    if TestContinue **then**
        for $\mathrm{y}_i \in \mathcal{Y}$ **do** ▷ Loop on each output
            $D_{i,k} \leftarrow \max_{0 \leq n < \Omega}(\mathcal{D}_{y_i}(n) - \mathcal{D}_{\tilde{y}_i}(n))$ ▷ Def. 24
        end for
    end if
end while
if TestContinue **then**
    $D_{max} \leftarrow \max_{i,k}(D_{i,k})$ ▷ Def. 25
    $\mathcal{U}_{MI} \leftarrow \{\mathrm{u}_j | \exists i, D_{i,j} = D_{max}\}$ ▷ Def. 26
    $\mathcal{Y}_{MI} \leftarrow \mathcal{Y}_{del} \cup \bigcup_{k|\mathrm{u}_k \in \mathcal{U}_{del}} \mathcal{Y}_{del,U_{MI,k}}$ ▷ Def. 27
    $\mathcal{P}_{CAMI} \leftarrow \{\mathrm{p}_m | \forall \mathrm{u}_j \in \mathcal{U}_{MI}, \forall \mathrm{y}_i \in \mathcal{Y}_{MI}, \mathrm{u}_j \rightsquigarrow \mathrm{p}_m \rightsquigarrow \mathrm{y}_i\}$ ▷ Def. 28
    $\mathcal{P}_{CAMIC} \leftarrow \mathcal{P}_{CAMIC} \cap \mathcal{P}_{CAMI}$
end if
return $\mathcal{P}_{CAMIC}$

In this case, variables TestContinue and TestCircuits remain True, so the control sequence is executed for the three inputs in $\mathcal{U}_{del}$ and the set $\mathcal{P}_{CAMIC}$ at the end of the loop is $\mathcal{P}_{CAMIC} = \{\mathrm{p}_{12}, \mathrm{p}_{13}, \mathrm{p}_{15}\}$. After the loop, $\mathcal{P}_{CAMI} = \{\mathrm{p}_5, \mathrm{p}_{12}, \mathrm{p}_{13}, \mathrm{p}_{14}\}$ and $\mathcal{P}_{CAMIC}$ is finally defined as the intersection of both of these sets: $\mathcal{P}_{CAMIC} = \{\mathrm{p}_{12}, \mathrm{p}_{13}\}$. CAMIC is more precise than CAMI as the circuit analysis has pruned places p_5 and p_{14} from the list of candidate places. Place p_5 has been pruned as it is involved in an elementary path from an input to an output but there is no failure detection based on event zero in any output (see Propostion 16). Place p_{14} has been pruned as it is part of only one elementary circuit that holds 3 tokens so the potential detection of a time failure in p_{14} should have occurred on an event $n \geq 3$ (at least the 4^{th} event) in an output y, however, here $n = 2$.

Proposition 21. *CAMIC is sound.*

Proof. The set of candidate places computed with CAMIC results from the analysis of circuits and CAMI. Since both of these methods have been proven to be sound, so is CAMIC. Place p_f is always part of the candidates returned by CAMIC. □

6.5 Active Time Failure Localization Algorithm for TEGs (ATFLAT)

The Active Time Failure Localization Algorithm for TEGs (ATFLAT) is a global algorithm designed to localize time failures in TEGs regardless of the number of inputs or circuits. ATFLAT takes advantage of the results described in the previous sections to reduce the set of candidates by controlling the system with new inputs only when this reveals additional information.

In contrast with the previous localization method introduced in Section 5, ATFLAT tests properties of the TEG that relate to the presence of circuits and may apply new inputs to the system in situations where the previous algorithm would only return the worst-case scenario set of candidates, that is, $Loc(\mathcal{G}, U_{op}, Y)$.

Algorithm 5 introduces ATFLAT. It begins by determining whether the system contains circuits (line 1). As stated in Section 4.3, for an empty TEG, this is equivalent to whether the system contains places with an initial marking greater than zero. This information is first used on lines 3 through 10. As stated in the following Proposition 22, an input as defined in line 5 for a single-input TEG will reveal any detectable time failures in the system. The measured output for this input is then used in a circuit analysis as defined in Section 6.3.

Proposition 22. *An input series that can be written as $\gamma^0\delta^{t_0} \oplus \gamma^{\Omega}\delta^{+\infty}$, $t_0 \geq 0$ on a single-input TEG containing a detectable time failure results in an output y different from $\tilde{y}$.*

Proof. The reasoning is the same as the proof for Proposition 11. More precisely, we can consider $u = U_{MI,1}$ with $t_0 \geq 0$ replacing $t_{max,1}$ and chosen arbitrarily

Algorithm 5 Active Time Failure Localization Algorithm for TEGs (ATFLAT)

Input: $\mathcal{G}, U_{op}, Y$
Output: $\mathcal{P}_{cand}$

TestCircuits $\leftarrow \exists \mathrm{p} \in \mathcal{P} | M_0(\mathrm{p}) > 0$ ▷ True if there is at least one circuit
if $|\mathcal{U}| = 1$ **then**
  if TestCircuits **then**
    $\Omega \leftarrow 1 + \sum_{i=1}^{|P|} M_0(\mathrm{p}_i)$ ▷ Eq. 26
    $U_0 = e \oplus \gamma^{\Omega} \delta^{+\infty}$
    $(Y, \tilde{Y}, I(U_0, Y)) \leftarrow$ *ApplyControl*(U_0)
    ($\mathcal{P}_{cand}$, TestCircuits, TestContinue) $\leftarrow$ *CircuitAnalysis*$(\mathcal{G}, Y, \tilde{Y})$
  else
    $\mathcal{P}_{cand} \leftarrow$ *Loc*$(\mathcal{G}, U_{op}, Y)$ ▷ Def. 14
  end if
else
  $\mathcal{Y}_{del,U_{op}} \leftarrow \{\mathrm{y_i} \in \mathcal{Y} | I(U_{op}, y_i) = true\}$ ▷ Def. 13
  $\mathcal{U}_{del,U_{op}} \leftarrow \{\mathrm{u}_k | \forall \mathrm{y}_i \in \mathcal{Y}_{del,U_{op}}, \mathrm{u}_k \rightsquigarrow \mathrm{y}_i\}$ ▷ Def. 20
  if $|\mathcal{U}_{del,U_{op}}| = 1$ **then**
    if TestCircuits **then**
      Let $\mathcal{U}_{del,U_{op}} = \{\mathrm{u}_k\}$
      $\mathcal{P}_M \leftarrow \{\mathrm{p} | \forall \mathrm{y}_i \in \mathcal{Y}_{del,U_{op}}, \mathrm{u}_k \in \mathcal{U}_{del,U_{op}}, \mathrm{u}_k \rightsquigarrow \mathrm{p} \rightsquigarrow \mathrm{y}_i$ and $M_0(\mathrm{p}) > 0\}$
      if $\mathcal{P}_M \neq \varnothing$ **then**
        $\Omega \leftarrow 1 + \sum_{i=1}^{|P|} M_0(\mathrm{p}_i)$ ▷ Eq. 26
        $U_{MI,k} \leftarrow$ *Compute*$U_{MI,k}(H, \Omega, k)$
        $(Y, \tilde{Y}, I(U_{MI,k}, Y)) \leftarrow$ *ApplyControl*$(U_{MI,k})$
        ($\mathcal{P}_{cand}$, TestCircuits, TestContinue) $\leftarrow$ *CircuitAnalysis*$(\mathcal{G}, Y, \tilde{Y})$
      else
        $\mathcal{P}_{cand} \leftarrow$ *Loc*$(\mathcal{G}, U_{op}, Y)$
      end if
    else
      $\mathcal{P}_{cand} \leftarrow$ *Loc*$(\mathcal{G}, U_{op}, Y)$
    end if
  else
    if TestCircuits **then**
      $\mathcal{P}_{cand} \leftarrow$ CAMIC$(\mathcal{G}, \mathcal{Y}_{del,U_{op}}, \mathcal{U}_{del,U_{op}})$ ▷ Algorithm 1
    else
      $\mathcal{P}_{cand} \leftarrow$ CAMI$(\mathcal{G}, \mathcal{Y}_{del,U_{op}}, \mathcal{U}_{del,U_{op}})$ ▷ Algorithm 4
    end if
  end if
end if
return $\mathcal{P}_{cand}$

because there is only one input. The assumptions regarding H and H' are the same and the resulting outputs Y and $\tilde{Y}$ are similar. □

The same logic is used in lines 15 through 28; however, in this case the analysis is only executed if there is a circuit in a relevant path, that is, between the input that is upstream of the place containing the failure (u_k) and all the delayed outputs ($\mathcal{Y}_{del,U}$).

Finally, if there may be multiple inputs upstream of the place containing the time failure, CAMIC is executed if circuits are present in the TEG, CAMI is executed otherwise (lines 30 through 34).

Proposition 23. *ATFLAT is sound.*

Proof. The set of candidate places computed with ATFLAT results from either the structure-based localization method, the analysis of circuits, CAMIC or CAMI. Since all of these methods are sound, so is ATFLAT. □

Proposition 24. *The set of candidate places $\mathcal{P}_{cand}$ returned by ATFLAT is such that $\mathcal{P}_{cand} \subseteq Loc(\mathcal{G}, Y, \tilde{Y})$.*

Proof. In the worst case $\mathcal{P}_{cand} = Loc(\mathcal{G}, Y, \tilde{Y})$. $\mathcal{P}_{cand}$ may also result from one call to CircuitAnalysis so $\mathcal{P}_{cand} \subseteq Loc(\mathcal{G}, Y, \tilde{Y})$ by Proposition 20. Otherwise, $\mathcal{P}_{cand}$ results from a call to CAMIC or CAMI, therefore $\mathcal{P}_{cand} \subseteq Loc(\mathcal{G}, Y, \tilde{Y})$. □

In practice, ATFLAT is more precise than $Loc(\mathcal{G}, Y, \tilde{Y})$ for several reasons. As soon as the suspected paths from the inputs to the outputs are better identified by the use of CAMIC, $\mathcal{P}_{cand} \subset Loc(\mathcal{G}, Y, \tilde{Y})$. ATFLAT is also more precise as soon as CircuitAnalysis prunes out some places from the candidate set. ATFLAT may also be more conclusive by suspecting either a single place or a set of places in one elementary circuit.

6.6 Complexity Analysis and Implementation

The main objective of the ATFLAT algorithm is to first design a set of control sequences to be applied on the system and then call the function $ApplyControl(U_{MI,k})$ to exploit the results of the indicators for the localization. To analyze the overall complexity of ATFLAT, let firstly analyze the worst-case time complexity of *ApplyControl*. This complexity is actually the complexity of the $|Y|$ available time failure indicators $I(U, y_i)$. As detailed in [13], the complexity of $I(U, y_i)$ is in $O(\Omega^2 log(\Omega))$ so the complexity of *ApplyControl* is in $O(|Y| \times \Omega^2 log(\Omega))$. Algorithm 1 iterates over a subset of the inputs U and successively calls $ComputeU_{MI,k}$ and *ApplyControl*. Complexity of $ComputeU_{MI,k}$ is in $O(|U|)$ (see Definition 23), it follows that the worst-case complexity of CAMI is in $O(|U|^2 \times |Y|\Omega^2 log(\Omega))$ (which is by the way also the complexity of Algorithm 2). Now consider CAMIC (see Algorithm 4) and notice that the complexity of *CircuitAnalysis* is in $O(|Y| \times \Omega)$ so it is negligible with respect to the complexity of *ApplyControl*, so CAMIC is also in $O(|U|^2 \times |Y|\Omega^2 log(\Omega))$.

Finally, in the worst-case ATFLAT calls either CAMI or CAMIC so ATFLAT has the same worst-case complexity as CAMIC and CAMI.

All the algorithms presented in this paper, namely CAMI, CircuitAnalysis, CAMIC and ATFLAT have been included in the C++ library MaxPlusDiag ([13]) which relies on MinMaxGD ([9]). The examples in this paper were computed using this library.

7 Conclusion

This paper describes methods for the synthesis of control sequences and the analysis of the measurable elements of timed event graphs modeled as (max,+)-linear systems in the context of the active diagnosis of time failures.

Time failures are defined as changes in the normal delays imposed by the system. The objective of an active diagnosis session is to determine the origin of such failures, namely the place of the timed event graph that has had its duration altered, by applying carefully selected inputs to the system offline. This process is called active localization.

The definition of sets of transitions and places based on the analysis of the timed event graph is proposed in order to target the parts of the system that may contain the failure. The analysis is based on existing methods for detecting time failures online. These sets of input transitions, output transitions and places of the timed event graph are then used in a decision-making process to establish if the control of the system may indeed provide new, relevant information. In the first proposed method of failure localization, this is done by taking advantage of the presence of multiple inputs in the system if that is the case. The second method does the same by considering the presence of circuits.

In the control algorithms introduced in this paper, the analytical properties of the system are used to define the necessary elements for designing inputs for the system, namely the number and dates of events of input series for the timed event graph. Then, the inputs are applied, the outputs of the system are measured, and a new analysis and decision-making process is conducted. These steps are repeated until the set of places that may contain the time failure is considered to be irreducible. This is the case of both the Control Algorithm for Multiple Inputs (CAMI) and its extension that considers circuits (CAMIC). The outcome of both of these algorithms has been proven to be a subset of the set obtained with the pre-existing localization method.

The combination of the analysis algorithms and the control algorithms constitutes a global Active Time Failure Localization Algorithm for TEGs (ATFLAT). This algorithm works by using the specialized methods in the cases in which they are relevant, by determining for instance if there are multiple inputs, if there are circuits, if different inputs could potentially lead to the place containing the failure... Since all the different parts of this global algorithm provide a smaller set than a simple structure-based localization, the choice of executing ATFLAT on a system containing a time failure provides a real advantage.

The design of this active localization algorithm leads to several perspectives.

First, the problem of studying timed event graphs containing multiple time failures can be addressed; although this case was not considered in this paper, an adaptation of the algorithm and the assumptions on the system could make its consideration possible. Furthermore, this paper addresses time failures that slow the system down, but accelerations of the normal the durations of the system can also be taken into account. Other types of failures, such as the event-shift failures defined in [13], could also be considered. Finally, another step of the diagnosis of the system could be combined with the detection and localization, specifically the estimation of the value of the time failure.

Conflict of interest

The authors have no conflict of interest to declare that are relevant to this article.

References

[1] F. Baccelli, G. Cohen, G. Olsder, and J.-P. Quadrat. *Synchronization and linearity: an algebra for discrete event systems*. UK: Wiley and sons, 1992.

[2] M. Bayoudh and L. Travé-Massuyés. "An Algorithm for Active Diagnosis of Hybrid Systems Casted in the DES Framework". en. In: *IFAC Proceedings Volumes*. 2nd IFAC Workshop on Dependable Control of Discrete Systems 42.5 (June 2009), pp. 287–292. ISSN: 1474-6670. DOI: 10.3182/20090610-3-IT-4004.00054. URL: https://www.sciencedirect.com/science/article/pii/S1474667015356354.

[3] N. Bertrand, É. Fabre, S. Haar, S. Haddad, and L. Hélouët. "Active Diagnosis for Probabilistic Systems". en. In: *Foundations of Software Science and Computation Structures*. Ed. by A. Muscholl. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2014, pp. 29–42. ISBN: 978-3-642-54830-7. DOI: 10.1007/978-3-642-54830-7_2.

[4] S. Böhm, S. Haar, S. Haddad, P. Hofman, and S. Schwoon. "Active diagnosis with observable quiescence". In: *2015 54th IEEE Conference on Decision and Control (CDC)*. Dec. 2015, pp. 1663–1668. DOI: 10.1109/CDC.2015.7402449.

[5] E. Chanthery and Y. Pencolé. "Monitoring and Active Diagnosis for Discrete-Event Systems". In: *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes*. 2009.

[6] E. Chanthery, L. Travé-Massuyès, Y. Pencolé, R. De Ferluc, and B. Dellandréa. "Applying Active Diagnosis to Space Systems by On-Board Control Procedures". In: *IEEE Transactions on Aerospace and Electronic Systems* 55.5 (2019), pp. 2568–2580.

[7] G. Cohen, P. Moller, J.-P. Quadrat, and M. Viot. "Algebraic tools for the performance evaluation of discrete event systems". In: *Proceedings of the IEEE* 77.1 (1989), pp. 39–85. DOI: 10.1109/5.21069.

[8] B. Cottenceau, L. Hardouin, J.-L. Boimond, and J.-L. Ferrier. "Model Reference Control for Timed Event Graphs in Dioids". In: *Automatica* 37 (2001). Ed. by Elsevier, pp. 1451–1458.

[9] B. Cottenceau, M. Lhommeau, L. Hardouin, and J.-L. Boimond. "Data processing tool for calculation in dioid". In: *5th International Workshop on Discrete Event Systems*. http://www.istia.univ-angers.fr/~hardouin/outils.html. 2000.

[10] Y. Guo and X. He. "Active Diagnosis of Incipient Actuator Faults for Stochastic Systems". In: *IEEE Transactions on Industrial Electronics* (2023). Conference Name: IEEE Transactions on Industrial Electronics, pp. 1–9. ISSN: 1557-9948. DOI: 10.1109/TIE.2023.3247778.

[11] S. Haar, S. Haddad, T. Melliti, and S. Schwoon. "Optimal constructions for active diagnosis". en. In: *Journal of Computer and System Sciences* 83.1 (Feb. 2017), pp. 101–120. ISSN: 0022-0000. DOI: 10.1016/j.jcss.2016.04.007. URL: https://www.sciencedirect.com/science/article/pii/S0022000016300198.

[12] L. Kuhn, B. Price, J. De Kleer, M. B. Do, and R. Zhou. "Pervasive Diagnosis: The Integration of Diagnostic Goals into Production Plans". In: *23rd AAAI Conference on Artificial Intelligence*. 2008.

[13] E. Le Corronc, Y. Pencolé, A. Sahuguède, and C. Paya. "Failure detection and localization for timed event graphs in (max,+)-algebra". In: *Journal of Discrete Event Dynamic Systems* 31 (2021). Ed. by Springer, pp. 513–552.

[14] E. Le Corronc, A. Sahuguède, Y. Pencolé, and C. Paya. "Localization of time shift failures in (max,+)-linear systems". In: *14th Workshop on Discrete Event Systems*. 2018.

[15] G. R. Marseglia and D. M. Raimondo. "Active fault diagnosis: A multiparametric approach". en. In: *Automatica* 79 (May 2017), pp. 223–230. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2017.01.021. URL: https://www.sciencedirect.com/science/article/pii/S0005109817300316.

[16] MaxPlus. "Second order theory of min-linear systems and its application to discrete event systems". In: *30th IEEE Conference on Decision and Control*. 1991.

[17] E. Menguy, J.-L. Boimond, L. Hardouin, and J.-L. Ferrier. "Just-in-time control of timed event graphs: update of reference input, presence of uncontrollable input". In: *IEEE Transactions on Automatic Control* 45.11 (2000), pp. 2155–2159.

[18] C. Paya, E. L. Corronc, Y. Pencolé, and P. Vialletelle. "Observer-Based Detection and Localization of Time Shift Failures in (max, +)-Linear Systems". In: *17th IEEE International Conference on Automation Science and Engineering*. Lyon France, Aug. 2021.

[19] G. Provan. “An Algebraic Approach for Diagnosing Discrete-Time Hybrid Systems”. In: *28th International Workshop on Principles of Diagnosis*. 2018.

[20] A. Sahuguède, E. Le Corronc, and Y. Pencolé. “Design of indicators for the detection of time shift failures in (max, +)-linear systems”. In: *20th IFAC World Congress*. 2017.

[21] M. Sampath, S. Lafortune, and D. Teneketzis. “Active diagnosis of discrete-event systems”. In: *IEEE Transactions on Automatic Control* 43.7 (1998), pp. 908–929.

[22] G. Schafaschek, L. Hardouin, and J. Raisch. “Optimal Control of Timed Event Graphs with Resource Sharing and Output-Reference Update”. In: *Automatisierungstechnik* 68.7 (2020). Ed. by O. Verlag, pp. 512–528.

[23] M. Šimandl and I. Punčochář. “Active fault detection and control: Unified formulation and optimal design”. en. In: *Automatica* 45.9 (Sept. 2009), pp. 2052–2059. ISSN: 0005-1098. DOI: 10.1016/j.automatica.2009.04.028. URL: https://www.sciencedirect.com/science/article/pii/S0005109809002210.

[24] P. Struss. “Model Abstraction for Testing of Physical Systems”. In: *Eigth International Workshop on Qualitative Reasoning, QR-94*. Jan. 1994.

[25] D. Thorsley and D. Teneketzis. “Active Acquisition of Information for Diagnosis and Supervisory Control of Discrete Event Systems”. en. In: *Discrete Event Dynamic Systems* 17.4 (Dec. 2007), pp. 531–583. ISSN: 1573-7594. DOI: 10.1007/s10626-007-0027-y. URL: https://doi.org/10.1007/s10626-007-0027-y.

[26] J. Van Gorp, A. Giua, M. Defoort, and M. Djemai. “Active diagnosis for a class of switched systems”. In: *52nd IEEE Conference on Decision and Control*. 2013.

[27] I. Velasquez, E. Le Corronc, and Y. Pencolé. “Active Diagnosis Algorithm for the Localization of Time Failures in (Max,+)-Linear Systems”. en. In: *IFAC-PapersOnLine*. 16th IFAC Workshop on Discrete Event Systems WODES 2022 55.28 (Jan. 2022), pp. 276–283. ISSN: 2405-8963. DOI: 10.1016/j.ifacol.2022.10.354. URL: https://www.sciencedirect.com/science/article/pii/S2405896322023916.