# An overview on diagnosability and prognosability for system monitoring

Amaury Vignolles, Elodie Chanthery, Pauline Ribot

# An overview on diagnosability and prognosability for system monitoring

Amaury Vignolles[1,2], Elodie Chanthery [1,2] and Pauline Ribot [1,3]

[1] *CNRS, LAAS, 7 avenue du colonel Roche, F-31400 Toulouse, France*
*amaury.vignolles@laas.fr elodie.chanthery@laas.fr pauline.ribot@laas.fr*

[2] *Univ de Toulouse, INSA, LAAS, F-31400 Toulouse, France*

[3] *Univ de Toulouse, UPS, LAAS, F-31400 Toulouse, France*

## Abstract

The complexification of systems has brought the emergence of a new field of study: system health monitoring. This field is deemed necessary because it improves system availability and it avoids unnecessary maintenance costs. System health monitoring is performed through diagnosis and prognosis methods. Diagnosis consists in detecting and identifying faults that may lead to system failures. Prognosis is related to the prediction of the system Remaining Useful Life (RUL) that corresponds to the remaining time until the system failure. This paper aims at giving an overview on the properties related to diagnosis and prognosis on different types of systems. We will focus on the diagnosability and prognosability properties. This paper will first briefly present the different types of systems of interest for the system health monitoring community. We will consider Discrete Event Systems (DES), Continuous Systems (CS), Hybrid Systems (HS) or Heterogeneous Systems (HtS). The rest of this paper will present the definitions given in the literature for the concepts of diagnosability and prognosability. The similarities and differences in these definitions for the different types of systems will be highlighted. Some metrics associated with the prognosability property will also be discussed.

## 1. Introduction

Nowadays, systems are more and more complex and a more complex system is subject to more faults and failures reasons. To detect and identify these faults, a *diagnosis* reasoning is required. Diagnosis algorithms (Isermann, 1997) were developed to monitor systems and know if a fault occurred (detection) and which fault (or faults) occurred (isolation). Recently, industrials want not only to be able to know if a fault, and which one, occurred, but also to be able to predict the remaining time during which the system will be able to fulfill its purpose, or in other words, the time until the system encounters a failure, also known as the Remaining Useful Life (RUL) of a system. *Prognosis* is often related to this temporal prediction of the RUL. To understand well the difference between faults and failures, the definitions of faults (Isermann, 1997) and failures (Villemeur, 1988) must be reminded.

**Definition 1 (Failure)**
*A failure state is a state in which the system is not able to fulfill its purpose anymore.*

**Definition 2 (Faults)**
*Faults represent a non-acceptable deviation from a characteristic property or a parameter of the system.*

Faults might lead to failures, but not necessarily, as in some cases, the system might still be able to fulfill its purpose.
In this paper, we will survey the notion of *diagnosability* and *prognosability* in different types of systems. Diagnosability is a notion that has been well studied for a long time, while prognosability is a more recent notion. Intuitively, we can define diagnosability and prognosability as follows, based on the etymology of the words themselves.

**Definition 3 (Diagnosability-intuition)**
*Diagnosability represents the ability for a system to be diagnosed. In other terms, diagnosability is the ability for a system and its monitoring capacities to exhibit different symptoms for each fault situation.*

**Definition 4 (Prognosability-intuition)**
*Prognosability represents the ability for a system to be prognosed. In other terms, it represents the ability to predict a failure.*

In this paper, proper definitions given in the literature are to be found for different kinds of systems.
Four different kinds of systems will be considered: discrete-event systems (DES), continuous systems (CS), hybrid systems (HS) and finally heterogeneous systems (HtS).
A DES is defined as a system that will only encounter, and produce, discrete data. If a DES encounters continuous data, it will be abstracted to generate discrete events. Among the most widespread formalisms for DES are automata, state machines or Petri nets, interested readers can find more details in (Cassandras & Lafortune, 2009).
A CS will have continuous dynamics. In general, a CS is described by a dynamic equation $C$ of the form:

$$C = \begin{cases} x_{k+1} & = & \mathbf{f}(x_k, u_k) & + & \mathbf{v}(x_k, u_k) \\ y_k & = & \mathbf{h}(x_k, u_k) & + & \mathbf{w}(x_k, u_k) \end{cases}, \quad (1)$$

where $x_k \in \mathbb{R}^{n_x}$ is the continuous state vector of $n$ state variables at time $k$, $u_k \in \mathbb{R}^{n_u}$ is the vector of $n_u$ continuous input variables at time $k$, $\mathbf{f}$ is the noise-free continuous evolution function, $\mathbf{v}$ is a noise function, $y_k \in \mathbb{R}^{n_y}$ is the vector of $n_y$ continuous output variables at time $k$, $\mathbf{h}$ is the noise-free output function and $\mathbf{w}$ is the noise function associated with observation.
A HS will always be subject to continuous and discrete data. The most known formalisms for hybrid systems are hybrid automata (Cassandras & Lafortune, 2009), hybrid Petri nets (Alla & David, 1998) or hybrid bond graphs (Mosterman & Biswas, 1998).
Finally, we consider that a HtS is a system that could be described as different sub-systems, which are either DES, CS or HS. It means that a HtS will sometimes be affected by discrete data only, sometimes by continuous data only, and sometimes both.
For each of these type of systems, $\Sigma_f$ will denote the set of faults that may occur on the system.
For each one of these formalisms, one specific definition of diagnosability or prognosability can be proposed. The goal here is to see where these definitions converge or diverge and if it is possible to merge some interesting concepts for the specific HtS.

This paper is organized as follows. Section 2 will survey the diagnosability notion as defined by the literature for each of the previously presented category of systems. Based on Section 2 layout, Section 3 will survey the prognosability notion and introduce prognosability's metrics. Finally, Section 4 will conclude, criticize and highlight some interesting points on which advancements could be made.

## 2. DIAGNOSABILITY

This section exposes how diagnosability is defined in the literature. Table 1 shows the reviewed articles and the studied types of systems. The following subsections give more details about the definitions given by the articles.

### 2.1. Diagnosability in DES

Diagnosability in DES is quite well defined in the literature. One of the first definition was given in (Sampath et al., 1995) for systems modeled by automata.

**Definition 5 (Diagnosability by Sampath et al., 1995)**
*Diagnosability is the ability for a DES to provide a particular set of observations after the occurrence of a fault $f \in \Sigma_f$ belonging to a fault's partition.*

A fault's partition is a set $\Omega_f \subseteq \Sigma_f$ containing one or several faults, represented by unobservable discrete events. For example, the fault's partition $\Omega_1 = \{f_1; f_2\}$ contains the faults $f_1$ and $f_2$, whereas $\Omega_2 = \{f_3\}$ contains only the fault $f_3$. So, in this case, the two faults $f_1$ and $f_2$ are not distinguished as they are represented in the same fault partition. The set of observations must allow the monitor (also called the diagnoser) to detect the occurrence of a particular fault's partition. However, this definition of diagnosability does not imply that the fault which occurred will be precisely known. Moreover, the previous definition is quite strict and might not be applicable to some real systems. For example, if you consider the case of a door, it is impossible to know if the door is broken and stuck closed (event $f_{sc}$) without trying to open the door. Based on this idea, a relaxed definition has been proposed by the same authors in (Sampath et al., 1995), named the I-diagnosability.

**Definition 6 (I-diagnosability)**
*I-diagnosability is the ability for a DES to provide a particular set of observations, allowing the monitor to detect the occurrence of a fault $f \in \Sigma_f$ and to identify the fault's partition not after the occurrence of the said fault, but after the occurrence of an indicator event related to the fault.*

Considering the previous example of the door, the event $open_{door}$ is considered as an indicator event related to the fault $f_{sc}$.
Definition 5 is used as a base to define most of the notions related to diagnosability and is sometimes used as it is like in (Yoo & Lafortune, 2002), along with Definition 6. It was also applied to Petri Nets in (Basile et al., 2012) and (Madalinski et al., 2010). A new definition, k-diagnosability, is given in (Basile et al., 2012).

**Definition 7 (k-diagnosability)**
*k-diagnosability is the ability to be k-diagnosable. To be k-diagnosable, a fault event $f \in \Sigma_f$ must be diagnosable after the firing of at most $k$ transitions after its occurrence. If all the faults $f \in \Sigma_f$ are k-diagnosable, the system is k-diagnosable.*

In (Zanella, 2017), $||^{//l}$-diagnosability is presented to take into account temporal uncertainty into the diagnosability def-

| System Type | Articles |
|---|---|
| DES | (Sampath, Sengupta, & Lafortune, 1995) and (Ye, Dague, & He, 2019) [Automata]<br>(Zanella, 2017) [Twin plant : finite automata]<br>(Basile, Chiacchio, & De Tommasi, 2012) [Labelled Petri Nets]<br>(Yoo & Lafortune, 2002) [Finite-state automaton]<br>(Madalinski, Nouioua, & Dague, 2010) [Labelled Petri Nets] |
| CS | (Staroswiecki & Comtet-Varga, 2001) [algebraic dynamic systems]<br>(Zuniga, Sotomayor-Moriano, Chanthery, Travé-Massuyès, & Soto, 2019)<br>(Cordier, Travé-Massuyes, Pucel, et al., 2006) |
| HS | (Zaatiti, Ye, Dague, & Gallois, 2017) [Partially observed hybrid automata]<br>(Bayoudh & Travé-Massuyès, 2014)<br>(Fourlas, Kyriakopoulos, & Krikelis, 2002) [Hybrid I/O Automata] |
| HtS | (Hsieh & Chen, 2008) [distributed]<br>(Hua, Li, Ren, & Liu, 2016) |

Table 1. Reviewed articles on diagnosability classified by system types

inition for systems modeled by automata. This temporal uncertainty is up to a certain level, denoted $l$. For example, let us consider four events $a, b, c, d$, such that events $a$ and $d$ are temporally unrelated as illustrated by Figure 1. In this case, when $a$ and $d$ both occur, it is not possible to know if the sequence is $a.d$ or $d.a$. To represent this temporal uncertainty, the authors propose to write $a//d$. Thus, an example of observations for the system represented in Figure 1 is $b.a//d.c$. In this example, as there are two temporally unrelated events, we consider $l = 2$.

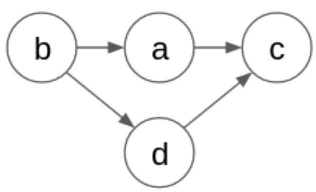


Figure 1. Temporaly unrelated events, from (Zanella, 2017)

**Definition 8 ($||^{//l}$-diagnosability)**
*$||^{//l}$-diagnosability represents the ability for a system to be $||^{//l}$-diagnosable. For a faulty behavior $f \in \Sigma_f$ to be $||^{//l}$-diagnosable, it should produce a set of observations that cannot be mistaken, even with temporal uncertainty on the events up to a level l, with a set of observations produced by a nominal behavior or a different fault. If all the faults $f \in \Sigma_f$ are $||^{//l}$-diagnosable, the system is $||^{//l}$-diagnosable.*

In (Ye et al., 2019), a weaker definition is introduced, and called *manifestability*. This notion is based upon the idea that diagnosability might be expensive in terms of sensors requirement and is defined for systems modeled by automata.

**Definition 9 (Manifestability)**
*The notion of manifestability represents the possibility for a fault occurrence to be manifestable, i.e. to be recognized through observations. A manifestable fault $f \in \Sigma_f$ will have effects on the observation given by the system. If all the faults $f \in \Sigma_f$ are manifestable, the system is said manifestable.*

Based on this notion, if a fault is not manifestable, then, whatever the event following its occurrence, it is not possible to know for sure whether it occurred or not. A non-manifestable fault is therefore not diagnosable.

### 2.2. Diagnosability in CS

Diagnosability in CS is usually composed of detectability and isolability and is based upon what is called Analytical Redundancy Relations (ARRs) (Staroswiecki & Comtet-Varga, 2001), (Pérez, Chanthery, Travé-Massuyès, & Sotomayor, 2017), which are analytical relations based upon the set *obs* of measured variables.

**Definition 10 (ARRs)**
*A dynamic constraint $C_{obs}(\bar{u}_k, \bar{y}_k, \bar{\epsilon}_k) = 0$ is an ARR for Equation 1 if for all ($u_k$,$y_k$) consistent with Equation 1, the dynamic constraint is statistically satisfied, with $u_k$ being the vector of input variables, $y_k$ the vector of output variables and $\epsilon_k$ the vector of noise variables, and $\bar{u}_k$ (resp. $\bar{y}_k$; $\bar{\epsilon}_k$ ) stands for the vector $u_k$ (resp. $y_k$; $\epsilon_k$) and its derivatives up to some unspecified order.*

Each ARR gives rise to a consistency indicator, called a residual.

**Definition 11 (Residual)**
*The residual $r$ is time-dependent and is given by a boolean value by the following application:*

$$r(t) = \begin{cases} 0 \text{ if } C_{obs} \text{ is satisfied by } (u(s), y(s)), \ s \leq t \\ 1 \text{ otherwise} \end{cases} \quad (2)$$

When the residual is equal to 0 it means that the observations are consistent with the model, so no fault has occurred. If the residual is 1, it means that a fault is detected.

We can split the residuals in two parts: a deterministic (depending on $\bar{u}(t)$ and $\bar{y}(t)$) and a stochastic part (depending on $\bar{\epsilon}(t)$).
The two following concepts related to diagnosability are defined in (Staroswiecki & Comtet-Varga, 2001).

**Definition 12 (Detectability)**
*Detectability is the ability for a fault to be detectable. A fault $f \in \Sigma_f$ is said to be detectable if its residual evaluation differs from zero. If all the faults $f \in \Sigma_f$ are detectable, the system is detectable.*

**Definition 13 (D-detectability)**
*D-detectability is the ability to be D-detectable. A fault $f \in \Sigma_f$ is said to be D-detectable if the deterministic part of its residual evaluation form differs from zero. If all the faults $f \in \Sigma_f$ are D-detectable, the system is D-detectable.*

D-detectability is a weaker concept than detectability as a non-D-detectable fault might influence the stochastic part of their residuals and, thus, could be detected.
After detecting a fault, it has to be isolated, for this purpose, a fault signature has to be defined.

**Definition 14 (Fault signature)**
*A fault signature is a function Sig associating a set of observables (residuals) to each fault.*

The following definition of isolability is given in (Zuniga et al., 2019).

**Definition 15 (Isolability)**
*Isolability is the ability to be isolable. Two faults $f_i$ and $f_j \in \Sigma_f$ are isolable iff their fault signatures differ. If all pairs of faults are isolable, the system is said isolable.*

The following definition of diagnosability for CS is given in (Cordier et al., 2006).

**Definition 16 (Diagnosability by Cordier et al., 2006)**
*Diagnosability is the ability to be diagnosable. Two faults $f_i$ and $f_j \in \Sigma_f$ are diagnosable iff the junction of their signatures is empty. Formally, it can be written as:*
$Sig(f_i) \cap Sig(f_j) = \emptyset$.
*If all pairs of faults are diagnosable, the system is said diagnosable.*

## 2.3. Diagnosability in HS

In (Fourlas et al., 2002), a first definition of a diagnosable fault is given for HS, on Hybrid I/O Automata. Here, a fault is modeled as an event whose occurrence impacts either the continuous variables, the discrete state or both.

**Definition 17 (Diagnosability by Fourlas et al., 2002)**
*Diagnosability is the ability to be diagnosable. A fault $f \in \Sigma_f$ is said diagnosable in HS if its occurrence can be detected after $t \in \mathbb{N}$ changes of trajectories (changes of continuous variables values) and actions (events) occurred on the system. If all the faults $f \in \Sigma_f$ of the system are diagnosable, so is the system.*

The following notions around the concept of diagnosability, for multimode systems, are defined in (Bayoudh & Travé-Massuyès, 2014). The first one is *mode diagnosability* and the second one is *fault diagnosability*, the former being stronger than the latter. From the ARRs of the systems, we can compute mode signatures, from which fault signatures are determined as the set of the systems modes in which the fault is present.

**Definition 18 (Mode)**
*A mode $q_i$ corresponds to a working state of the system and is associated with a set of continuous dynamics $C_i$ (see Equation 1) representing the evolution of the continuous variables.*

**Definition 19 (Mode signature)**
*The signature of a mode $q_i$ is the vector obtained by the concatenation of the vectors of boolean residuals associated to all the different modes $q_j$ of the system considered with the observations consistent with the considered mode $q_i$. It characterizes the behaviour of this particular mode with regard to all the other modes.*

**Definition 20 (Mode diagnosability in HS)**
*Two modes $q_i$ and $q_j$ are said diagnosable if their signatures are different ($Sig(q_i) \neq Sig(q_j)$). If all pairs of modes are diagnosable, the system is said to be mode diagnosable.*

**Definition 21 (Fault signature by Bayoudh et al., 2014)**
*The signature of a fault $f \in \Sigma_f$ is given by the set of signatures of the system modes in which the fault is present. Assuming that the model does not account for actions that repair the faults, the signature of a fault $f_i$ is equal to the set of signatures of the modes that are reachable from a transition labeled with the fault event $f_i$.*

**Definition 22 (Fault diagnosability)**
*A fault $f \in \Sigma_f$ is considered diagnosable if its occurrence can be detected after a given time window thanks to the observation of the discrete events and the measurements of the continuous variables available on the system. Two faults $f_i$ and $f_j \in \Sigma_f$ are diagnosable from the continuous observations if $Sig(f_i) \cap Sig(f_j) = \emptyset$.*

As the notion of mode diagnosability is stronger than the notion of fault diagnosability, a mode diagnosable system is fault diagnosable.
The following property is introduced and proved in (Bayoudh & Travé-Massuyès, 2014).

**Property 1 (System diagnosability)**
*For a hybrid system to be diagnosable, either its underlying DES or multimode system must be diagnosable.*

The diagnosability in Partially Observable Hybrid Automaton (POHA), in which some variables are not observed, is defined in (Zaatiti et al., 2017) as follows.

**Definition 23 ($\Delta$-diagnosability)**
*$\Delta$-diagnosability represents the ability to be $\Delta$-diagnosable. A fault event $f \in \Sigma_f$ is said $\Delta$-diagnosable in a POHA iff it is possible to detect its occurrence at least $\Delta$ time units after it occurred. If all faults $f \in \Sigma_f$ are $\Delta$-diagnosable, then the system is $\Delta$-diagnosable.*

Here, $\Delta$ represents the minimum required time (real time) before detecting and identifying the fault.

### 2.4. Diagnosability in HtS

Only a few articles study HtS, and even fewer the specific point of diagnosability in HtS. The first of the two following articles we found in the literature, (Hsieh & Chen, 2008), is based on a multiprocessors system. It is comprised of multiple communicating processors called nodes. These nodes can be of different natures: DES, CS or HS and can be faulty. The authors have defined what they called t-diagnosability and strong t-diagnosability.

**Definition 24 (t-diagnosability)**
*t-diagnosability consists in the fact that, as long as the number of faulty components does not exceed $t$, they can be detected without a replacement (a faulty node is replaced by a non-faulty one).*

**Definition 25 (Strong t-diagnosability)**
*Strong t-diagnosability means that the system is t-diagnosable and can achieve $(t + 1)$-diagnosability.*

A data driven method for quantitative fault diagnosability evaluation is proposed in (Hua et al., 2016). As this method is data-based, we consider that it can be applied on complex systems such as HtS. The definition of distinguability is introduced.

**Definition 26 (Distinguability)**
*Distinguability represents the difference between two modes (both faulty or one faulty and one normal) and is calculated with fuzzy numbers. It is represented by a number between 0 and 1.*

The closer distinguability is from 0, the harder it is to distinguish the two modes. The faults leading to the said modes are not isolable (therefore, not diagnosable).

### 2.5. Synthesis on diagnosability

Diagnosability is a quite-well defined notion for the different types of systems (DES, CS, HS). Following the needs and the specifications of systems, advanced notions of diagnosability have emerged, to consider uncertainty or to give more precise outputs.

Table 2 classifies the studied articles regarding the type of information provided by the diagnosability analysis. While some of the outputs are just boolean (diagnosable or not), some tend to be more precise and parameter dependent, giving for example a number of transitions representing the maximum delay until the detection of the fault, or a time window representing the minimum time from which the fault's occurrence can be detected. Another precision on the result of the diagnosability analysis is to represent the uncertainty on the diagnosability analysis. This uncertainty could be up to how much uncertainty on the time occurrence of the observations the diagnosability of the fault is sure. Another one could be up to represent the upper limit on the number of faulty components in our distributed system for it to be diagnosable. Finally, another type of result is a real number, between 0 and 1, representing how 'difficult' it is to distinguish the fault from another.
Different ways to consider uncertainty of the system are represented in Table 3.

- The temporal uncertainty, representing the idea that when events are temporally unrelated, it is not possible to know which event occurred first,
- The noise appearing on the sensors, potentially not giving the exact output. This has to be somehow taken into account to have a precise analysis,
- The reliability of the testing nodes in distributed systems, represented with uncertainty on the output. If a testing node is faulty, its output is totally unreliable.

Table 4 shows how the time is represented and taken into account in the diagnosability evaluation. As a matter of fact, some represent the time through a number of transitions or events (usually the case for DES), while some others represent it with continuous time. Some others represent the time with a number of actions or changes occurring on the system. Finally, the time could also not be taken into account and the process could be done online.

## 3. PROGNOSABILITY

In contrast to diagnosability, prognosability is not so well defined in the literature, as we will notice in this section. Prognosability is a property relative to the prediction of the future state of the system, related to the prognosis idea. For the PHM community, prognosis aims at predicting the system future states and its RUL or End of Life (EoL) from diagnosis and future inputs available from a mission scenario for example.

### 3.1. Prognosability in DES

In DES, prognosability has been studied for a very long time, even before the idea of diagnosability appeared. In discrete event models, the idea of time does noes appear as only sequence of events are considered. Prognosability in DES was

| Results of the diagnosability analysis | Articles |
|---|---|
| Diagnosable or not | (Sampath et al., 1995)<br>(Ye et al., 2019)<br>(Bayoudh & Travé-Massuyès, 2014)<br>(Staroswiecki & Comtet-Varga, 2001)<br>(Yoo & Lafortune, 2002)<br>(Madalinski et al., 2010)<br>(Fourlas et al., 2002) |
| Parameter Dependent | (Zanella, 2017) [Dependent on temporal uncertainty window]<br>(Basile et al., 2012) [Dependent on the number of transition]<br>(Zaatiti et al., 2017) [Dependent on a given time]<br>(Hsieh & Chen, 2008) [Dependent on number of detectable faulty components] |
| Real number | (Hua et al., 2016) |

Table 2. Reviewed articles on diagnosability classified by the type of result given by the diagnosability analysis

| Uncertainty | Articles |
|---|---|
| Temporal uncertainty | (Zanella, 2017) |
| Noise | (Bayoudh & Travé-Massuyès, 2014)<br>(Staroswiecki & Comtet-Varga, 2001) |
| Node reliability | (Hsieh & Chen, 2008) |

Table 3. Reviewed articles on diagnosability classified by the type of appearing uncertainty

indeed studied under different names such as predictability or trajectory prediction.
The very first work on predictability is (Cao, 1989), who works on automata, and introduces one of the first definition of predictability, allowing the monitor to predict the occurrence of a system event or set of events. It is based on the projection operation, defined as follows.

**Definition 27 (Projection)**
*The projection $P_L$ of an event $e$ on a language $L$ is defined as $P_L(\varepsilon) = \varepsilon$, with $\varepsilon$ being the empty word and*

$$P_L(e) = \begin{cases} e \text{ if } e \in L \\ \varepsilon \text{ if } e \notin L \end{cases} \quad (3)$$

**Definition 28 (Predictability by Cao, 1989)**
*Predictability is the ability to be predictable. Considering two automatons $H$ and $G$, it is said that a finite sequence of events $t$, defined as a string, belonging to the language of $H$ denoted $L(H)$ is predictable from a string $s$ in $L(G)$ if it matches the projection of $s$ on $H$. This definition is extended to a language, which is said predictable if all the strings contained in the said language are predictable.*

A string is defined as a finite sequence of events, but an event by itself could be considered as a string of length 1.
The following additional definition for systems modeled by automata is introduced in (Genc & Lafortune, 2009).

**Definition 29 (Predictability by Genc & Lafortune, 2009)**

*Predictability is the ability to be predictable. Let us consider any string $s$ ending with the event $e$ to be predicted. Such an event is said to be predictable if we can find, for all existing $s$, a prefix $t$ of $s$, not containing $e$ at first but which will contain the desired event after a sufficiently long continuation. If all events of a system are predictable, the system is said predictable.*

In this work, predictability is also proved to be a stronger property than diagnosability (see Definition 5), as both require to identify and isolate the desired event, but predictability must do it preemptively.
An algorithm to evaluate the predictability property applicable on distributed systems is proposed in (Ye, Dague, & Nouioua, 2015), based on automata as well.

Definition 29 is extended to define k-predictability on timed automata in (Cassez & Grastien, 2013) as follows.

**Definition 30 (k-predictability)**
*k-predictability represents the ability to be k-predictable. Let $f \in \Sigma_f$ be a fault. $f$ is considered as k-predictable if it is predictable $k$ event occurrences before its occurrence.*
*A system is k-predictable if all the faults $f \in \Sigma_f$ are k-predictable.*

In the case $k = 0$, the definition of 0-predictability matches Definition 29 .

Stochastic DES are DES in which each event $e$ has a probability $p(e)$ to occur. The following definition is given in (Chen & Kumar, 2014):

**Definition 31 ($S_m$-prognosability)**
*$S_m$-Prognosability means that for any threshold value $\rho > 0 \in \mathbb{R}$ and error bound $\tau > 0 \in \mathbb{R}$, there exists a reaction bound $k \geq m$, such that if a fault $f \in \Sigma_f$ cannot be pre-*

| Way to express time | Articles |
|---|---|
| Number of transitions | (Sampath et al., 1995)<br>(Basile et al., 2012)<br>(Yoo & Lafortune, 2002)<br>(Madalinski et al., 2010) |
| Number of trajectories modifications and actions | (Fourlas et al., 2002) |
| Number of events | (Zanella, 2017)<br>(Ye et al., 2019)<br>(Bayoudh & Travé-Massuyès, 2014) [For the HS underlying DES] |
| Time | (Zaatiti et al., 2017)<br>(Bayoudh & Travé-Massuyès, 2014)<br>(Staroswiecki & Comtet-Varga, 2001) |
| Can be done online | (Hua et al., 2016) |

Table 4. Reviewed articles on diagnosability classified by the way to express time during the diagnosability analysis

*dicted $k$ steps (i.e. $k$ events) in advance with confidence level $\rho$ (we cannot be $\rho\%$ sure that the fault is going to happen), the probability of its occurrence is smaller than $\tau$ .*

Definition 29 is also used in (Nouioua, Dague, & Ye, 2016) and applied on probabilistic DES. Based upon the probabilities associated to the occurrence of each event, they refine the definition and propose a *probabilistic predictability* for the system. As a matter of fact, they compute, for each fault $f \in \Sigma_f$, a probability $p_{ko}(f)$ to be in a trace in which the fault could not be predicted but occurred, a probability $p_{ok}(f)$ to be in a trace where the fault could be predicted and occurred and a probability $p_{nf}(f)$ to be in a trace in which the fault will not occur.

Definition 29 is extended to labelled Petri Nets in (Yin, 2017).

The notion of predictability given in Definition 29 is extended to decentralized systems in (Kumar & Takai, 2009) and (Wu, Yin, & Li, 2018), specifically Petri Nets for the latter. The notion of *coprognosability* is also introduced. The decentralized systems are monitored by local prognosers, each equipped with its own event observation sensors, observing a behavior (sequence of events) executed by a system and issuing their local prognostic decisions. These local prognostic decisions are gathered at a central decision unit and used to issue a global prognostic decision.

**Definition 32 (Coprognosability)**
*Coprognosability is the ability to be coprognosable. If a fault $f \in \Sigma_f$ is prognosable by a local prognoser, it is said coprognosable. If all faults $f \in \Sigma_f$ are coprognosable, the system is coprognosable.*

This article is one of the first to use the word "prognosability" on DES to refer to the predictability property.

### 3.2. Prognosability in CS

The community working on CS is quite active and consequent. However, prognosability in CS is not that studied and only a few works give a formal definition of this notion.

(M. J. Daigle & Sankararaman, 2013) work on prognostic on CS. They define the system with the set of equations:

$$\begin{cases} x_{k+1} = f(k, x_k, \theta_k, u_k, v_k) \\ y_k = h(k, x_k, \theta_k, u_k, n_k) \end{cases} \tag{4}$$

where $k$ is the discrete time variable, $x_k \in \mathbb{R}^{n_x}$ is the state vector at time $k$, $\theta_k \in \mathbb{R}^{n_\theta}$ is the unknown parameter vector at time $k$, $u_k \in \mathbb{R}^{n_u}$ is the input vector at time $k$, $v_k \in \mathbb{R}^{n_v}$ is the process noise vector at time $k$, $n_k \in \mathbb{R}^{n_n}$ is the measurement noise vector at time $k$ and $h$ is the output equation. $\theta_k$ is used to capture the model parameters whose values are unknown and time-varying stochastically.
In this work, their purpose is not to compute the date of occurrence of a fault but to compute the RUL of the system. They use an *estimation* process, computing the joint state-parameter estimate based on the observations up to the considered time and a *prediction* process, which computes a probability distribution of the EoL for each time. This probability distribution is computed with the joint state-parameter distribution and the system model, along with the distributions of possible parameters, inputs and process noise trajectories.
No formal definition of prognosability is given in the article, but some metrics, defined later in this paper, are used to check the performances of the result.
Many works on CS are actually based on these two estimation and prediction processes, among them are (Sankararaman, Daigle, Saxena, & Goebel, 2013) or (Acuña, Orchard, Reyes, & Zhang, 2019), which differs through the use of Lebesgue sampling, compared to the more resource consuming Rieman sampling, and thus use probability distribution for the occurrence of faults or particular events.

### 3.3. Prognosability in HS

On HS, the work of (M. Daigle, Roychoudhury, & Bregon, 2015) can be considered. Although a definition of prognosability is not given, we can deduce one from the article, as follows. A HS is considered as a set of components, each described by a set of discrete modes, with a set of constraints describing the continuous dynamics of the component in each mode.

**Definition 33 (Prognosability based on Daigle et al., 2015)**

*Prognosability is the ability to be prognosable. An event $e$ is said prognosable on a hybrid system if, given the system model, the initial state $s_0$, a time horizon $k$, the future parameters trajectory, the future input trajectory and the future process noise trajectory, it is possible to find one particular trace of events and variables evolution leading to the event $e$. If all the events $e$ of the system are prognosable, the system is prognosable.*

### 3.4. Prognosability in HtS

We have only found a few articles on HtS, as this type of system is not really studied. No definition of prognosability is given in (Bartram & Mahadevan, 2010) but the article is interesting as it apply a prognostic algorithm. The occurrence of a fault is represented with Bayes Networks and probabilities, and is updated through the occurrence of events on the system's components.
Even if most works do not directly consider prognosability, a set of metrics have to be defined to evaluate the prognostic results.

### 3.5. Metrics for Prognosability

When evaluating prognostic performance, commons metrics are used:

- The *precision*, representing the spread of the results,
- The *accuracy*, representing the bias,
- The *Mean Squarred Error* (MSE),
- The *Mean Absolute Percentage Error* (MAPE).

Some more advanced metrics are defined in (Saxena, Celaya, Saha, Saha, & Goebel, 2010). These metrics are computed one after the other as illustrated in Figure 2.

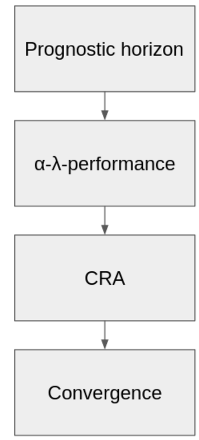


Figure 2. Sequence of Saxena's four metrics

The *prognostic horizon* represents the ability for the pronostic algorithm $\Phi$ to predict within a specified error margin (represented by a parameter $\alpha \in \mathbb{R}$) the EoL and how much time before it happens.
Then, the second metric, based on the accuracy, is the *Cumulative Relative Accuracy* (CRA), representing the ability for $\Phi$ to improve its estimation of the EoL as more information become available.
Finally, the third metric introduced is *convergence*, showing how fast $\Phi$ converges.
The $\alpha - \lambda$*-performance* property, representing the ability for $\Phi$ to predict within a specified error margin ($\alpha$) the EoL at any time $\lambda$.

Three metrics to choose a feature interesting to monitor to prognose a fault $f \in \Sigma_f$ are introduced in (Coble & Hines, 2009): the *monotonicity*, the *prognosability* and the *trendability*. Each of these metrics ranges from 0 to 1, and are based on data obtained through Run-to-Failures (RtF) of the system. A feature is a variable of the system, and is sampled during the RtF.
The first metric, monotonicity, characterizes the trend of the feature. It is given by the average absolute difference of the fraction of the derivatives of the data. When monotonicity is close to 1 it represents a constant trend of the feature all along the life of the system (it only decreases or increases).
Prognosability gives a measure of the variance in the critical failure value of a feature of the system. The prognosability metric is obtained by computing the deviation of the feature at the time where the fault occurs. The closer the prognosability metric is to 1, the better the chosen feature is to prognose the studied fault.
Finally, the last metric, trendability, shows if chosen features have the same underlying shape and can be described by the same functional form (i.e. the evolution of these different parameters are similar through the life of the system).

### 3.6. Synthesis on prognosability

As we have seen, prognosability is a notion which is not as well defined as diagnosability in the literature. The definitions of prognosability differ, be it on the object of prediction (what is prognosed) or the type of results given by the analysis.
For diagnosability property, the occurrence of a particular event (usually a fault) is what is looked for. For the prognosability property, works look to predict the occurrence of an event or of a set of events. The prognosis result is sometimes the prediction of the occurrence of a fault event or the temporal prediction of the RUL. These different types of prediction are gathered in Table 5. The results of the analysis also differs in most of the definitions, as can be seen in Table

6. Some definitions of the prognosability property give a prediction on how long, in terms of number of events, of steps, or time units before the occurrence of the fault we can know it is going to occurs. For some definitions, the idea of time is not present, as the results of the prognosability analysis just give the knowledge that the fault is prognosable or not. Finally, some works give a timed probability information: at time $k$, a fault $f \in \Sigma_f$ has $p(f)$ probability to occur. Some other works give a non-timed probability information: there is probability $p_1(f)$ for a fault $f \in \Sigma_f$ to occur and be predicted, $p_2(f)$ for the fault to occur without being predicted or $p_3(f)$ not to occur.

## 4. CONCLUSION

Through this review, it was shown that diagnosability is a notion quite well defined. This is not the case for prognosability, for which the definition and the ways to use it are quite different depending on the studied system. It can be explained by the fact that the PHM community is quite recent and first works were guided by applications. For discrete event systems, that will only produce and encounter discrete data, both definition of diagnosability and prognosability are well defined.

For continuous systems, described by a set of continuous dynamic equations, diagnosability is well defined, whereas prognosability is not.

For hybrid systems, who will always encounter both continuous and discrete data, diagnosability is well defined whereas prognosability is not, although the emergence of works on this type of system is really recent.

Finally, for heterogeneous systems, that we consider as a system that will sometimes be affected by discrete data only, sometimes by continuous data only and sometimes by both, neither diagnosability nor prognosability are well defined, and only a few works consider this type of system.

For our future works, we would like to consider an heterogeneous system under uncertainty and have some ideas about which definitions of diagnosability and prognosability will be used as a base for this kind of system. In the case where data are available for this kind of system, the prognosability metrics proposed by (Coble & Hines, 2009) are interesting to choose which of the feature are best to prognose a fault $f$. The others metrics could be useful to evaluate the performance of an health monitoring algorithm implemented on this system.

## REFERENCES


Acuña, D. E., Orchard, M. E., Reyes, C., & Zhang, B. (2019). Theoretical advances in lebesgue-sampling-based prognostic algorithms. In *2019 prognostics and system health management conference (phm-paris)* (pp. 7–12).

Alla, H., & David, R. (1998). Continuous and hybrid Petri nets. *Journal of Circuits, Systems, and Computers*, *8*(01), 159–188.

Bartram, G., & Mahadevan, S. (2010). Bayesian reliability prognosis for systems with heterogeneous information. *Annual Conference of the Prognostics and Health Management Society*.

Basile, F., Chiacchio, P., & De Tommasi, G. (2012). Decentralized k-diagnosability of Petri nets. *IFAC Proceedings Volumes*, *45*(29), 214–220.

Bayoudh, M., & Travé-Massuyès, L. (2014). Diagnosability analysis of hybrid systems cast in a discrete-event framework. *Discrete Event Dynamic Systems*, *24*(3), 309–338.

Cao, X.-R. (1989). The predictability of discrete event systems. *IEEE transactions on automatic control*, *34*(11), 1168–1171.

Cassandras, C. G., & Lafortune, S. (2009). *Introduction to discrete event systems*. Springer Science & Business Media.

Cassez, F., & Grastien, A. (2013). Predictability of event occurrences in timed systems. In *International conference on formal modeling and analysis of timed systems* (pp. 62–76).

Chen, J., & Kumar, R. (2014). Stochastic failure prognosability of discrete event systems. *IEEE Transactions on Automatic Control*, *60*(6), 1570–1581.

Coble, J., & Hines, J. W. (2009). Identifying optimal prognostic parameters from data: a genetic algorithms approach. In *Annual conference of the prognostics and health management society* (Vol. 27).

Cordier, M.-O., Travé-Massuyes, L., Pucel, X., et al. (2006). Comparing diagnosability in continuous and discrete-event systems. In *Proceedings of the 17th international workshop on principles of diagnosis (DX-06)* (pp. 55–60).

Daigle, M., Roychoudhury, I., & Bregon, A. (2015). Model-based prognostics of hybrid systems.

Daigle, M. J., & Sankararaman, S. (2013). Advanced methods for determining prediction uncertainty in model-based prognostics with application to planetary rovers.

Fourlas, G., Kyriakopoulos, K. J., & Krikelis, N. (2002). Diagnosability of hybrid systems. In *Proceedings of the 10th IEEE mediterranean conference on control and automation.*

Genc, S., & Lafortune, S. (2009). Predictability of event occurrences in partially-observed discrete-event systems. *Automatica*, *45*(2), 301–311.

Hsieh, S.-Y., & Chen, Y.-S. (2008). Strongly diagnosable product networks under the comparison diagnosis model. *IEEE Transactions on Computers*, *57*(6), 721–732.

Hua, Y., Li, Q., Ren, Z., & Liu, C. (2016). A data driven method for quantitative fault diagnosability evaluation. In *2016 chinese control and decision conference*

| Object of prediction | Articles |
|---|---|
| Fault | (Bartram & Mahadevan, 2010)<br>(Cassez & Grastien, 2013)<br>(Wu et al., 2018)<br>(Kumar & Takai, 2009)<br>(Chen & Kumar, 2014)<br>(Ye et al., 2015)<br>(Yin, 2017) |
| Any system event | (M. Daigle et al., 2015)<br>(Genc & Lafortune, 2009) |
| Any system event or set of event | (Cao, 1989) |
| Reach of a particular state | (Nouioua et al., 2016) |
| Remaining Useful Life | (M. J. Daigle & Sankararaman, 2013) |

Table 5. Reviewed articles on prognosability classified by the object of prediction

| Result of the analysis | Articles |
|---|---|
| Prognosable or not<br>without time information | (Cao, 1989)<br>(Genc & Lafortune, 2009)<br>(Ye et al., 2015) |
| Prognosable or not<br>with time information | (Kumar & Takai, 2009)<br>(Wu et al., 2018)<br>(Yin, 2017)<br>(Cassez & Grastien, 2013) |
| Prognosable or not<br>Parameter dependent | (Chen & Kumar, 2014) [threshold value, error bound and reaction bound] |
| Probability of prognosability<br>Without time information | (Nouioua et al., 2016) |
| Probability of fault occurence<br>Without time information | (Bartram & Mahadevan, 2010) |
| Probability of fault occurence at given time | (M. J. Daigle & Sankararaman, 2013) |
| Time of occurrence of an event | (M. Daigle et al., 2015) |

Table 6. Reviewed articles on prognosability classified by the type of result given by the analysis

*(CCDC)* (pp. 1890–1894).

Isermann, R. (1997). Supervision, fault-detection and fault-diagnosis methods—an introduction. *Control engineering practice*, *5*(5), 639–652.

Kumar, R., & Takai, S. (2009). Decentralized prognosis of failures in discrete event systems. *IEEE Transactions on Automatic Control*, *55*(1), 48–59.

Madalinski, A., Nouioua, F., & Dague, P. (2010). Diagnosability verification with Petri net unfoldings. *International Journal of Knowledge-Based and Intelligent Engineering Systems*, *14*(2), 49–55.

Mosterman, P. J., & Biswas, G. (1998). A theory of discontinuities in physical system models. *Journal of the Franklin Institute*, *335*(3), 401–439.

Nouioua, F., Dague, P., & Ye, L. (2016). Predictability in probabilistic discrete event systems. In *International conference on soft methods in probability and statistics* (pp. 381–389).

Pérez, C. G., Chanthery, E., Travé-Massuyès, L., & Sotomayor, J. (2017). Fault-driven structural diagnosis approach in a distributed context..

Sampath, M., Sengupta, R., & Lafortune, S. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on Automatic Control*, *40*(9), 1555-1575.

Sankararaman, S., Daigle, M., Saxena, A., & Goebel, K. (2013). Analytical algorithms to quantify the uncertainty in remaining useful life prediction. In *2013 IEEE aerospace conference* (pp. 1–11).

Saxena, A., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2010). Metrics for offline evaluation of prognostic performance. *International Journal of Prognostics and health management*, *1*(1), 4–23.

Staroswiecki, M., & Comtet-Varga, G. (2001). Analytical redundancy relations for fault detection and isolation in algebraic dynamic systems. *Automatica*, *37*(5), 687–699.

Villemeur, A. (1988). Dependability of industrial systems. *Book in French, Eyrolles, Paris, ISSN*, 0339–4198.

Wu, W., Yin, X., & Li, S. (2018). Verification of coprognosability in decentralized fault prognosis of labeled Petri nets. In *2018 IEEE conference on decision and control (CDC)* (pp. 4845–4850).

Ye, L., Dague, P., & He, L. (2019). Manifestability verification of discrete event systems. In *30th international* *workshop on principles of diagnosis DX'19,*.

Ye, L., Dague, P., & Nouioua, F. (2015). A predictability algorithm for distributed discrete event systems. In *International conference on formal engineering methods* (pp. 201–216).

Yin, X. (2017). Verification of prognosability for labeled Petri nets. *IEEE Transactions on Automatic Control*, *63*(6), 1828–1834.

Yoo, T.-S., & Lafortune, S. (2002). Polynomial-time verification of diagnosability of partially observed discrete-event systems. *IEEE Transactions on automatic control*, *47*(9), 1491–1495.

Zaatiti, H., Ye, L., Dague, P., & Gallois, J.-P. (2017). Counterexample-guided abstraction-refinement for hybrid systems diagnosability analysis. In *DX* (pp. 124–143).

Zanella, M. (2017). Asymmetric diagnosability analysis of discrete-event systems. In *DX* (pp. 78–93).

Zuniga, C. G. P., Sotomayor-Moriano, J., Chanthery, E., Travé-Massuyès, L., & Soto, M. (2019). Flotation process fault diagnosis via structural analysis. In *18th IFAC symposium on control, optimization and automation in mining, mineral and metal processing (MMM 2019).*