Journal name
000(00):1–13

DOI:doi number
http://mms.sagepub.com

A dynamic Bayesian network approach for modelling and optimizing predictive maintenance policies

J. Foulliaron*, **L. Bouillaut**
Universite Paris-Est, IFSTTAR, GRETTIA, F-93166 Noisy-le-Grand, France

P.Aknin
SNCF- Innovation & Research , 40 avenue des Terroirs de France, 75611 PARIS cedex 12, France

A.Barros
RAMS Group, Norwegian University of Science and Technology S. P. Andersens veg 5, Valgrinda 2.308A 7491 Trondheim Norway

Abstract

The maintenance optimization of complex systems is a key question. One important objective is to be able to anticipate future maintenance actions required to optimize the logistic and future investments. That is why, over the past few years, the predictive maintenance approaches have been an expanding area of research. They rely on the concept of prognostic. Many paper have shown how Dynamic Bayesian Networks (DBN) can be relevant to represent multicomponent complex systems and carry out reliability studies. The Diagnosis & maintenance group from IFSTTAR developed a model (VirMaLaB : Virtual Maintenance Laboratory)based on DBN in order to model a multicomponent system with its degradation dynamic and its diagnosis and maintenance processes. Its main purpose is to model a maintenance policy to be able to optimize the maintenance paremeters due to the use of DBN. A discrete state space system is considered, periodically observable through a diagnosis process. This paper presents a prognosis algorithm whose purpose is to compute the Remaining useful life (RUL) of the system and update this estimation each time a new diagnosis is available. Then, a representation of this algorithm is given as a DBN in order to be integrated into the VirMaLaB to integrate the set of predictive maintenance policies. Inference computation questions on the considered DBN will be discussed. Finally, an application on simulated data will be presented.

* Corresponding author; e-mail: josquin.foulliaron@ifsttar.fr

Keywords
Reliability, Prognosis, Dynamic Bayesian Network, degradation modelling, Discrete state space systems

1. Introduction

For the last fifty years, systems in most industrial fields have increased their complexity. A failure can lead at best, to a poorer performance in some parts of the system, and at worst to a complete shut-down that can cause an important security risk. When a failure is detected, it is necessary to find its cause, to get new components to replace and then to repair the damaged part. All these operations make the system unavailable for quite a long time which costs money. In the specific case of a transport system like a railway, a failure on a train can block all the train traffic. It has important consequences on the performance of the line. Moreover, the number of users is increasing and industrials have to do with specific security norms. It thus makes necessary both to increase the availability of the material and to guarantee a high level of security keeping down maintenance costs. For this reason, optimizing the maintenance of complex systems has become a key issue. Currently, most industrials use a compromise between systematic and corrective maintenance. Anyway, a maintenance action can be very costly specially if it has not been anticipated. Consequently, optimization of maintenance policies requires an anticipation of the future failure time which corresponds to a prognosis computation. The predictive maintenance is based on this principle and has been an expanding area of research since past few years **??**. The prognosis result could be subsequently used to optimize the maintenance and diagnosis schedule.
To describe the behaviour of a complex system and its diagnosis and maintenance process, we need a model that can represent a multicomponent system with potential dependency relationships between its components, the diagnosis devices, the maintenance actions and their influence on the degradation process of each component.

Probabilistic Graphical Models are mathematical objects that can model complex systems behaviour. Particularly, Dynamic Bayesian Networks (DBN) are powerful tools to represent complex systems evolving in time because of their simplicity and specially in their way of representing the causal relationships between system components **??**.
It is usually assumed that a DBN has the Markov property that makes possible to simplify its complexity and thereby the complexity of the computations. But the Markov property implies that the sojourn time in each state is exponentially distributed (or geometrically distributed in the discrete case). In practice, systems behaviours are usually different and it was proved that considering geometrical laws for sojourn times implies losing a lot of information about the degradation process. It thus leads to poor estimations concerning the future state of the system and consequently inadequate optimization of maintenance parameters.**?**. To address this problem, Donat et al. **?** developed the Graphical Duration Model (GDM) that is a modification of duration variables from **?**. It brings the possibility to use very kind of discrete distribution for sojourn times associated to each state.
In **?**, Foulliaron et al. proposed an extension of GDM in order to improve the degradation modeling. This new degradation model can consider some existing degradation modes and can adapt the degradation modelling to a mode change.
The Diagnosis &Maintenance group from IFSTTAR developed a graphical model called VirMaLab (Virtual Maintenance Laboratory) based on DBN and GDM to describe the degradation dynamic of a system and its Diagnosis/Maintenance process.**?**. It can be used to optimize the maintenance parameters according to multiple constraints, (reliability rate, costs, ..)
Some studies like **?** used a DBN to perform prognostic computation but the RUL was never represented as a specific node in a DBN. A discrete state space system that is periodically observable through a diagnosis process is considered. The aim of this paper is to present a prognosis algorithm and its representation in order to integrate it into an extended VirMaLaB model that could represent a multicomponent system and every kind of maintenance policy and particularly the set of predictive maintenance strategies based on prognosis computations. It is divided into two parts:

In the first part, we will present the framework from where we will start. The most commonly used maintenance policies will be first reviewed. Then, the context and the point of view used to model the degradation process will be explained. At last, the mathematical formalism (Dynamic Bayesian networks, Graphical Duration Model and its extension used to build the VirMaLab model will be presented.
The second part will focuses on the contribution of this paper. It will be divided into three sections. The first one will present a prognosis algorithm based on GDM and its extension whose purpose is to compute and update a RUL estimation each time a new diagnosis of the system is available. Some explanations and illustrations will be added to make the easier the understanding of how it works. In a second section, a representation of the prognosis algorithm as a DBN will be given. The third section will discuss the different inference methods that could be used on this DBN. Because of the complexity of the graph and the size of some nodes, a specific inference algorithm will be proposed to perform exact inference computation on the prognosis part of the DBN. The last section will present an application case, on simulated data, to illustrate how the methodology could be used in practice.
Finally, remaining difficulties concerning the complexity of computations and the main existing perspectives will be discussed.

List of acronyms

BN : Bayesian network
DBN : Dynamic Bayesian network
CPT : Conditional probability table
GDM : Graphical duration model
STD : Sojourn time distribution
CSTD : Conditional sojourn time distribution
$X^{(t)}$: Stochastic process representing the state of the considered system
$S^{(t)}$: Stochastic process representing the remaining sojourn time for the system to be on current state.
$\Psi_{x/y}$: Conditional sojourn time distribution on state x knowing state y
x_F : Failure state
T_S : Maximal possible sojourn time among all states
$T_{ECS}^{(t)}$: Time elapsed on the current state at time t
$T_{RCS}^{(t)}$: Estimation of the remaining time on the current state at time t
$R^{(t)}$: Sum of all sojourn times until the last state before the failure Δt : Constant time interval between diagnoses
$\Psi[a,b]$: Normalized restriction of the function Ψ on interval [a,b]
RUL : Remaining useful life.

2. Starting framework

2.1. To a predictive maintenance approach

This part gives an overview of the most used maintenance policies, and try to explain why the predictive maintenance can be very useful to optimize the maintenance schedule.

The set of usually used maintenance policies can be divided in two categories : corrective maintenance and preventive maintenance **?**. Principe of corrective maintenance is to repair a component when a failure occurs. In this case, a complete repair of the component(*curative maintenance*), or a minimal temporarily repair *(palliative maintenance)* can be performed.

These approaches can be used if broken components could be easily replaced and the failure state can be reached without causing risks of security or big costs. In other situations, preventive maintenance is used, consisting in preventing failures by acting before their apparitions. In this category, three approaches exist: *systematic maintenance, condition based maintenance (CBM) and predictive maintenance*.
Systematic maintenance is defined by a planning of replacement of components. In this approach, we don't try to know in what level of degradation we are, components are replaced at planned instant whatever state they are in. In this approach the question is to choose the best instant to guarantee a tolerable level of safety keeping reasonable costs and avoiding useless repairs on good health components. The advantage of this maintenance policy is to be able to handle the maintenance requirement.
Condition based maintenance is based on a monitoring of the system, providing a diagnosis of the operating state through some indicators on which threshold will be defined. This approach requires to get a diagnosis of the system state. In this case, the question is to optimize the inspection frequency and the chosen threshold **?**. The interest is to have "just in time" maintenance actions. However, it leads to unanticipated maintenance.
The predictive maintenance is attempting to integrate both advantage of systematic and condition-based maintenance. Its principle is to adapt maintenance actions according to a computed estimation of the future failure time of the system.
In this case, it is possible to optimize the logistics by anticipating the maintenance requirement. This kind of maintenance policy can be the best one, if it is correctly optimized.

The projection of the future health state of the system relies on the concept of prognosis. There are many ways in the literature to define what a prognosis computation is **?**.
In this paper, we will consider the classical definition: **The prognosis is the computation of a duration called the remaining useful life (RUL)**. It is defined as the remaining time before the system reaches a state that is considered to be intolerable. It can be the failure state or another state that is close to it. To perform a prognosis computation, modelling the dynamics of the considered system is required. Many different approaches are commonly used. They can be classified into three groups **?**.
- Models based approach : is interesting when an analytic expression (like differential equation, or dynamical systems) modelling the degradation process of the system is available. This kind of equations is frequently obtained from mechanical laws.
- Data based approach : consists in using monitoring data to predict the future behaviour using statistical methods (linear regression , time series models,..). In this approach no comprehension about why the system evolves as observed is needed. A global trend about the health state evolution is just extracted from collected observations and the methodology is to compute projections from this trend. - Reliability approach : consists in using feedback, expert opinion, return on operating experience, to estimate the probability laws of possible events. Data are collected to learn a degradation model using probabilistic graphical models (Bayesian network, Petri nets, neuronal networks..) or stochastic processes (Gamma process, Poisson process,..).
In the next section, the context of the study and the considered degradation modelling will be introduced.

2.2. Context and degradation model

In our framework, no mechanical degradation model and no continuous monitoring devices are available. The system is just periodically observable through a diagnosis process. So, ans hybrid approach based both on reliability and data will be developed. In this part, we are at the component scale. The state of a component is represented with a stochastic process with discrete space state and discrete time denoted $X^{(t)}$ where state 1 is the new state, and x_F is the failure state. So $t \in \mathbb{N}$ and $X^{(t)} \in \{1, .., x_F\}$. This state is considered to be hidden and can only be known at some instant through a diagnosis

device when this one is available. The diagnosis result could have some confusion rate.
It is assumed that $X^{(t)}$ is an increasing monotone process. From a state, it is only possible to stay on it or go to a more advanced degradation state. $\forall t,\ X^{(t+1)} \geq X^{(t)}$. It is assumed that there is no auto repair, and the failure state is absorbent. It can only be left with a corrective maintenance action. In this context, the RUL computed at time t for prognosis is a random variable formally defined as :

$$RUL^{(t)} = \inf_{T, T>t} \{X^{(T)} = x_F\} - t$$

A visual description of the RUL is presented on figure 1.

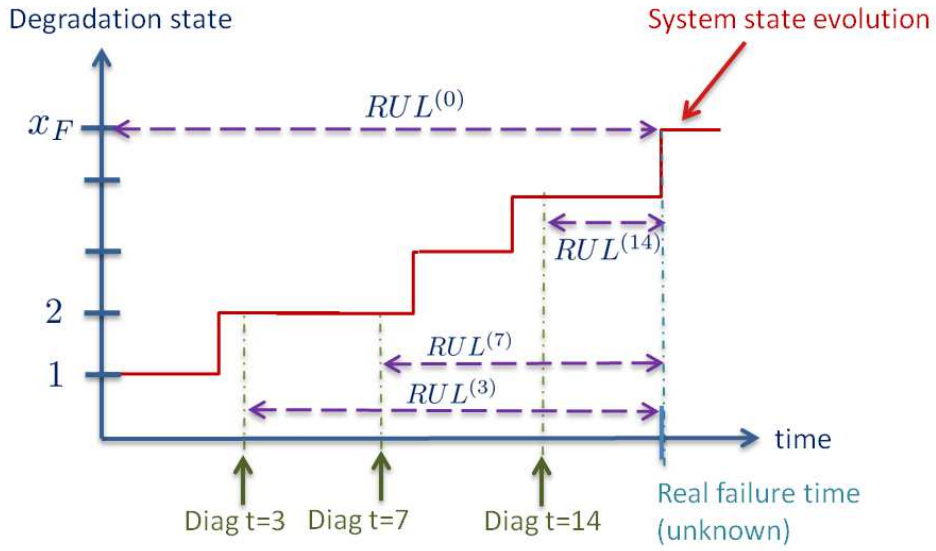


Fig. 1. RUL illustration in the presented context

Unlike most models in reliability, we don't try to get a function that describes the evolution of $X^{(t)}$ in function of t, but we consider the opposite point of view that is based on sojourn time distribution (STD) of each degradation state. The active sojourn time is a random variable denoted S. The active sojourn time on state x is denoted S_x. This approach is illustrated in Figure 2. T_S is denoted as the maximal possible sojourn time among all states.

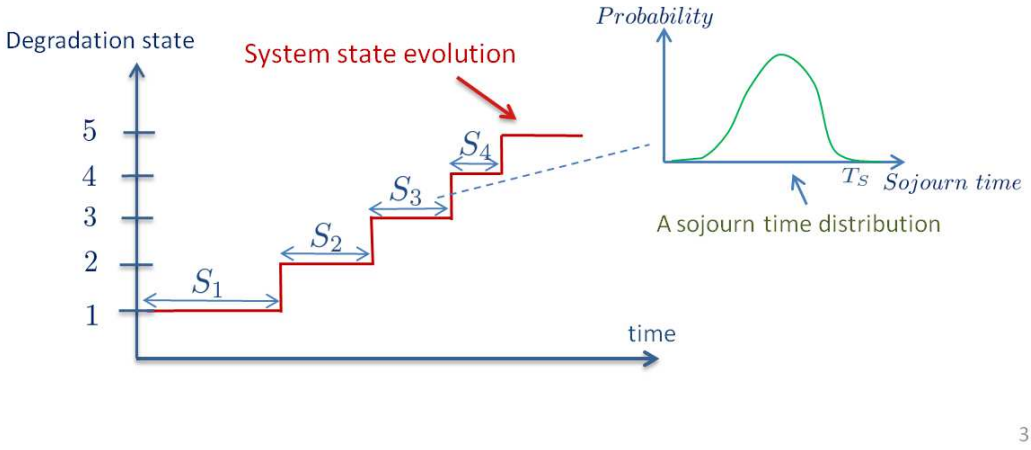


Fig. 2. Sojourn time based degradation model

Whereas classical models in reliability use stochastic processes (like Gamma process, Wiener process) with two or three parameters , this discrete "sojourn time" approach requires a large amount of parameters that are not always easy to

learn, but the main advantage is that no assumptions are required concerning the degradation dynamic modelling and its stochastic properties. Moreover, it perfectly match mixtures without needing to know if this mixture exists and to estimate its number of components.

The STD for each state is learnt with the maximum likelihood method. In the discrete case of our framework, by denoting Ψ_x the STD on state x , applying this method leads to the following formula :

$$\Psi_x(k) = p(S_x = k) = \frac{N_{x,k}}{|D|}$$

where $|D|$ is the number of observations in the database and $N_{x,k}$ the number of observations that contain a sojourn time equal to k on state x. So, the desired probabilities can be directly deduced from the frequencies histogram associated to the observation database. In the case of an incomplete database, an EM algorithm could be used to get those probabilities**?**. So the degradation model, can be seen as the set of discrete distribution $\{\Psi_1, .., \Psi_{x_F}\}$ where

$$\forall i, \sum_{x=1}^{T_S} \Psi_i(x) = 1$$

2.3. Extension with conditional sojourn times distributions

This section briefly presents an extended degradation model using the concept of conditional sojourn time distribution (CSTD). This extension is explained in detail in **?**. The main idea is to consider that STDs associated to each state are not independent. So, conditional sojourn time distributions are used. It is considered that the system can be in many different degradation modes (or dynamic). The STD in each state, is a mixture model. Each component of a mixture is linked to one degradation mode. We try to follow the effect of a degradation mode over all states by considering CSTD in a state knowing the system was on a degradation mode in the previous state. At each time, the most likely mode is estimated. STD in futur states can be changed when a mode change is detected by using adequate CJSD. This modified GDM using CSTD is denoted GDM-CSTD.

Next section presents the mathematical objects that have been used to represent the degradation model that have just been presented and its extension.

2.4. From Bayesian network to Dynamic Bayesian network

Bayesian networks (BNs) are directed acyclic graphs in which each node represents a random variable **?**. An arrow from a node X_i to anothoer one X_j represents a dependency relationship between the random variables X_i and X_j (an influence of X_a on X_j).
The parameter of each node is the conditional probability table (CPT) of its variable given the variables of the parent nodes. A CPT mathematically corresponds to a multidimensional array also called "potential".
An example of BN is shown in Figure 6. Blue arrows are the causality relationships, and dotted red lines represent the parameters associated with the nodes.

The conditional independence relationships between some random variables lead to the main property of the BN. The joint law of all random variables can be factorized as the product of local CPTs. If we denote $pa(X_i)$ as the set of parents nodes of node X_i. we have the general formula :

$$P(X_1, X_2, .., X_n) = \prod_{i=i}^{n} P(X_i|pa(X_i))$$

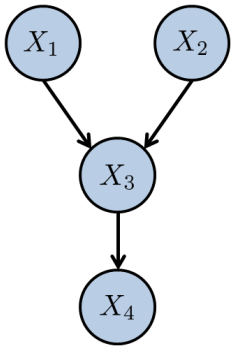


Fig. 3. Example of 4 nodes BN

Dynamic Bayesian Networks are an extension of BN adding the time dimension **?**. Random variables are not static but stochastic processes. They evolve over time. In our framework, we only use discrete random variables with discrete time. DBNs are like BNs with temporal dependency.

Usually, to simplify the model, the first order Markov property is assumed in the stochastic process. It means that the states of all random variables at time t only depend on states at the previous time. In other words, in the graph, the dependence arrows from a node in slice t only go to nodes in the same slice or to other nodes in slice t+1. It is assumed that the graph is homogeneous(conditional probabilities of $X^{(t)}|X^{(t-1)}$ don't depend on t).

From this assumption, it is just necessary to formally define a first order DBN like this : (denoted 2-DBN for 2 time slice DBN)

-A static BN_1 at initial time :

$$P(X_1^{(1)}, X_2^{(1)}, .., X_n^{(1)}) = \prod_{k=1}^{n} P(X_k^{(1)}|pa(X_k^{(1)}))$$

-A transition model $BN_{\rightarrow}$:

$$P(X_1^{(t)}, X_2^{(t)}, .., X_n^{(t)}|X_1^{(t-1)}, X_2^{(t-1)}, .., X_n^{(t-1)}) = \prod_{k=1}^{n} P(X_k^{(t)}|pa(X_k^{(t)}))$$

where $pa(X_k)$ is the set of parent nodes of X_k.

Figure 7 is a complete representation of a first order DBN. It contains the initial and the transition model. Dotted arrows correspond to time dependencies.

The main advantage of this representation is that both qualitative and quantitative nodes can be mixed in a same model

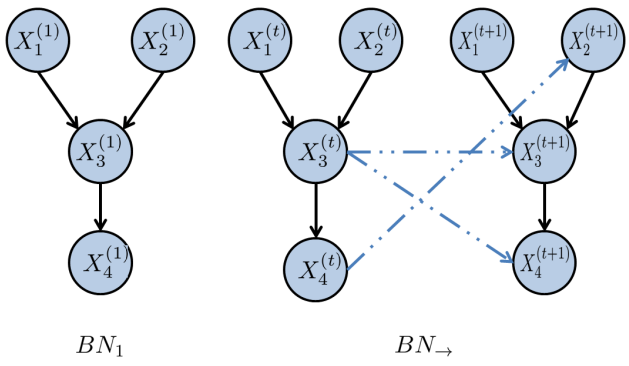


Fig. 4. Example of 1-DBN with 4 variables (X_1, X_2, X_3, X_4)

and represent many causality relationships. Most of the DBN used have nodes with discrete space states. There are thus many parameters for each nodes to learn in order to to build the model. Moreover, if nodes have large domains and have many parents, the size of potentials can be very large, and then computations could be very heavy.

In its "standard" approach, DBN induces a Markovian behaviour for each state, inducing geometrically distributed sojourn time. To overcome this limitation and keep the "sojourn time" point of view mentioned previously, a particular graphical model was proposed and will be reviewed in the next section.

2.5. Graphical Duration Models

Graphical Duration Model **?** is a specific Dynamic Bayesian network in which each considered variable X has its own duration variable denoted $S^{(t)}$ that controls its sojourn time . $S^{(t)}$ represents the remaining time at time t before the next transition of X. The probability of $X^{(t+1)}$ is conditioned by $X^{(t)}$ and $S^{(t)}$. At each time step, the value of $S^{(t)}$ decreases by one unit. When the value of $S^{(t)}$ is different from one, the state of $X^{(t)}$ is blocked, If $S^{(t)}$ is equal to 1, $X^{(t)}$ is free to go to the next state, and the new value for $S^{(t)}$ is chosen according to the sojourn time distribution associated with the new state of $X^{(t)}$.
The CPT of node $S^{(t)}$ is a n* T_S potential where n is the number of possible states for X. T_S is the maximal possible sojourn time among all states of X. The CPT of $X^{(t)}$ and $S^{(t)}$ are defined as below:

$$\begin{cases} S^{(t+1)} = S^{(t)} - 1 \text{ if } S^{(t)} \neq 1 \\ S^{(t+1)} \sim \Psi_{X^{(t)}} \text{ if } S^{(t)} = 1 \end{cases}$$

$$X^{(t+1)} = \begin{cases} X^{(t)} + 1 \text{ if } S^{(t)} = 1 \\ X^{(t)} \quad \text{otherwise} \end{cases}$$

where Ψ_x is the STD for state x. A representation of GDM is shown in Figure 5.

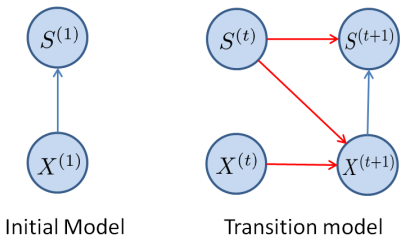


Fig. 5. DBN reprentation of GDM

If we want to use the conditional sojourn time distributions (CSTD) improvement mentioned in part 3.5, it is necessary to add one node denoted $C_1^{(t)}$ to the graphical duration model. $C_1^{(t)}$ represents the active mode at time t and is defined so that :

$$\begin{cases} C^{(t+1)} = C^{(t)} \text{ if } S^{(t)} \neq 1 \\ C^{(t+1)} = \arg\max_k (P(\text{mode} = k | S^{(t+1)})) \text{ if } S^{(t)} = 1 \end{cases}$$

Figure 6 illustrates the graphical duration model with its extension integrating the CSTD. Some texts describe the dependencies represented by arrows. More explanations can be found in **?**.To sum-up, this model gives the possibility to better modelize the degradation process without the drawback of a geometric law on sojourn time. $S^{(t)}$ can follow any discrete probability distribution. In this case, $X^{(t)}$ is said to be a semi-markovian process. It can consider many degradation nodes, due to node C_1 and can adapt the considered STDs to a mode change.

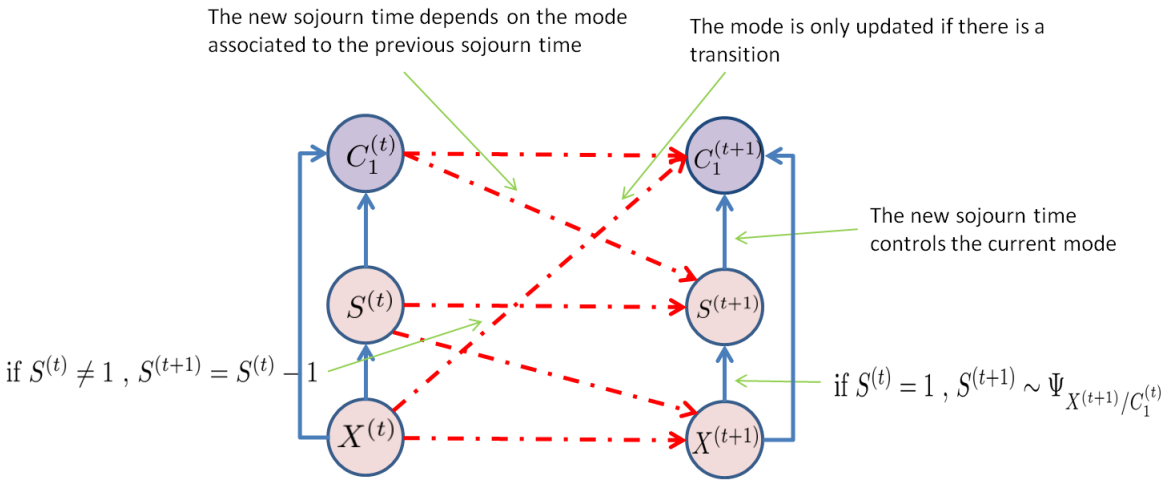


Fig. 6. GDM extended to conditional sojourn time distribution

2.6. The VirMaLab model

The VirMaLab (Virtual Maintenance Laboratory) approach aims to develop a support decision tool for optimization of maintenance parameters of complex systems **?**. It is a multicomponent scale development of the degradation model described before, including the modelling of diagnosis devices and maintenance actions.

The model is split into three connected modules. The degradation process, the maintenance modelling, and the evaluation module. In the degradation process module, the real state of the system is hidden. The diagnosis nodes in the maintenance module provide an estimation of the real hidden state of the system. A non identity matrix as the CPT of the diagnosis node makes possible to model the risk of error committed by the diagnosis device modelling its false alarm rates, non detection rates and mismatch rates. The diagnosis result will determine the next maintenance decision. It will modify the real hidden state of the system. Utility nodes can be used to associate costs to some maintenance or diagnosis actions to make possible to perform cost-based optimization of maintenance parameters. The interest of the VirMaLaB model is to be able to evaluate and compare some given maintenance policies, by providing probabilistic information and reliability indicators in function of parameters, in order to optimize the maintenance strategy.

Figure 7 below represents the VirMaLab model and its three main modules. For the moment, the possible maintenance policies that can be represented with VirMaLaB are corrective, systematic and condition based maintenance. To be able to model the predictive maintenance policies based on prognosis, a prognosis algorithm has to be established and then represented as a DBN in order to be integrated into the VirMalaB model. The next part will deal with this task.

3. The prognosis algorithm

3.1. Algorithm

a) Formal definitions :

The aim of this prognosis algorithm is to compute a first estimation of the RUL ($RUL^{(0)}$) and then to update and improve this estimation each time a new diagnosis is available. This version is based on a degradation process modelling by GDM. Before introducing the algorithm, some intermediate variables need to be detailed. Let us define :

- t_{co} : Time of current observation (last diagnosis time)
- t_{po} : Time of previous observation
- $x^{(t_{co})} = x_{co}$: State observed at time t_{co} ($x \in \{1, 2, ..., x_F\}$).

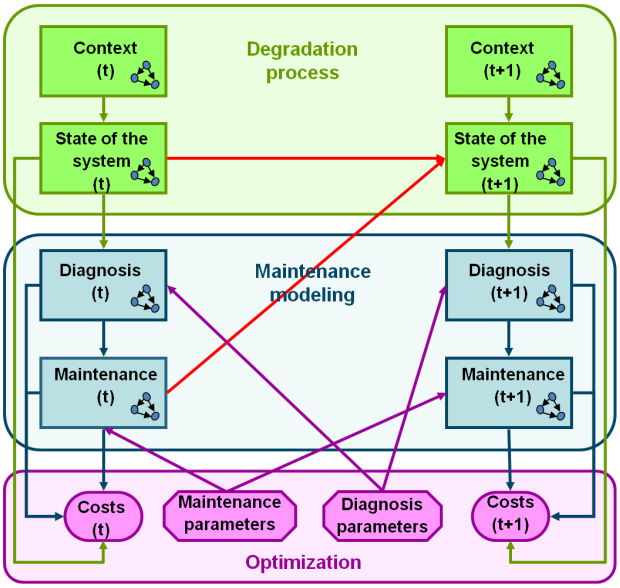


Fig. 7. VirMaLab model:

- $x^{(t_{po})}$: State observed at time t_{po}
- $T_{ECS}^{(t_{co})}$: Time elapsed at current state x_{co} until time t_{co}
- $T_{RCS}^{(t_{co})}$: Remaining time in the current state after t_{co}
- $S_x^{(t)}$: Estimation of the total sojourn time in state x computed at time t
- Δt : Time between the current observation and the last one $(t_{co} - t_{po})$
- $R^{(t)}$: Sum of all SJ in next states until the last state before failure.

Some of the previous variables are illustrated in Figure 8.

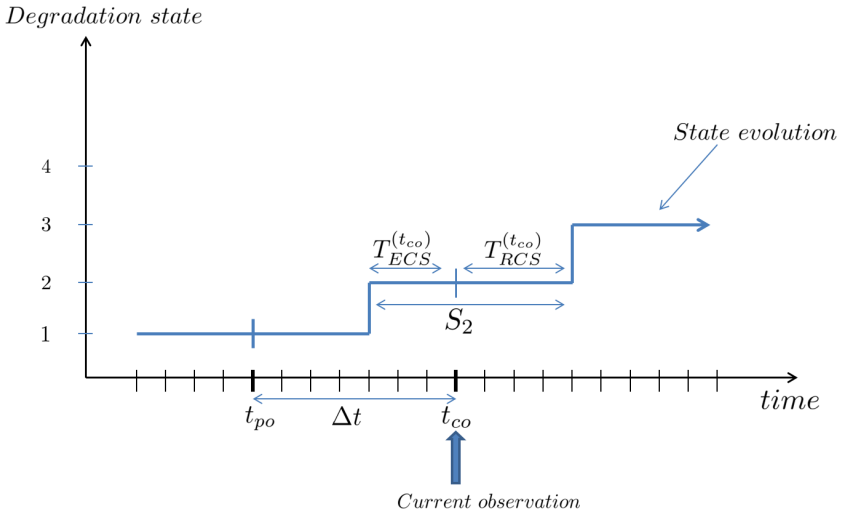


Fig. 8. Illustration of the main variables considered for the proposed prognosis algorithm

- Ψ_x is the STD on state x that represents the discrete conditional probability $p(S|X = x)$.

Notation $A \sim \Psi_x([a, b])$ means that the value for A is randomly drawn according to a new density denoted Ψ' that is a normalization of Ψ in interval $[a, b]$. So it is defined as : $\forall x_i \in [a, b], \Psi'(x_i) = \frac{\Psi(x_i)}{\sum_{x_i \in [a,b]} \Psi(x_i)}$

- α_j^{max} is the maximal possible value for sojourning in state j. In other words : $\alpha_j^{max} = sup\left\{k, P(S^{(t)} = k|X^{(t)} = j) > 0\right\}$

- $U[a, b]$ corresponds to the uniform distribution in [a,b] interval
- Time (0) is the algorithm starting instant.
- S_j compatible means compatible with the last current observation.

In this algorithm , elementary time unit was considered. It is assumed that, if the system is observed precisely at a transition time, the visible state is that after the transition and that minimal sojourn time on a state is one time unit. The abbreviation SJ is used for "sojourn time". All steps of the prognosis computation (initialization and update of the RUL) are detailed in algorithm 1. We can note that all random sampling in the algorithm can be replaced by choosing the average of the considered distribution. It has the advantage of reducing the risk of error on the prognosis.

b) Interpretations and discussion :

For each degradation state, during the initialization phase, a sojourn time is randomly drawn ($S_j^{(0)}$ is assigned for each state j). By adding these estimations , $R^{(0)}$ is computed. $T_{RCS}^{(0)}$ is chosen with a random sampling in a specific interval taking into account the information from $T_{ECS}^{(0)}$. Then, a first estimation of the RUL is computed. At each new available observation, the RUL is re-estimated by updating intermediate variables T_{ECS} and T_{RCS}. This update depends on the compatibility between the new observation and the last computed estimations. If a transition is observed, that isn't compatible with the previous estimations, the missed intermediate transition time is rebuilt using the STD associated to the previous state. If a lot of degradation states have been missed between two diagnosis, the real sojourn time of each intermediate state is very small so, the new T_{ECS} value is estimated with a uniform sampling. Then, the RUL is obtained by adding $T_{RCS}^{(t)}$ and $R^{(t)}$. Some of those update processes are illustrated in Figures 9 and 10.

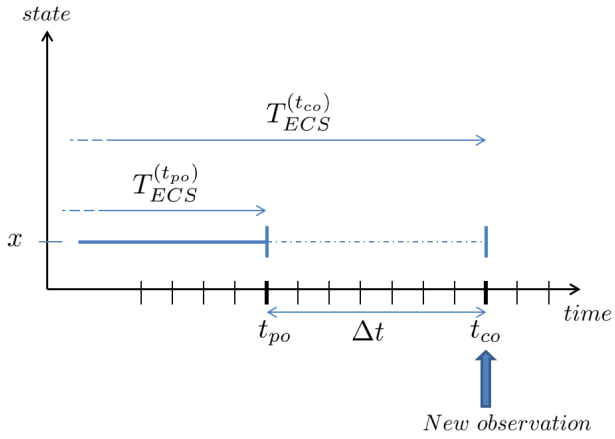


Fig. 9. T_{ECS} extending in the case where $x_{co} = x_{po}$

Denoting $S_j^{(t)} = T_{ECS}^{(t)} + T_{RCS}^{(t)}$, from the algorithm, the following formula is obtained :

$$RUL^{(t)} = \sum_{j=x^{(1)}}^{x_N - 1} S_j^{(t)} - n\Delta t$$

So, the error committed in the RUL computation depends on the error committed in each sojourn time estimation $S_j^{(t)}$. These estimations are updated and improved at each diagnosis time. If we denote $\epsilon(X) = X - \overline{X}$ where X is the estimated quantity X and $\overline{X}$ the real hidden one. The equation becomes :

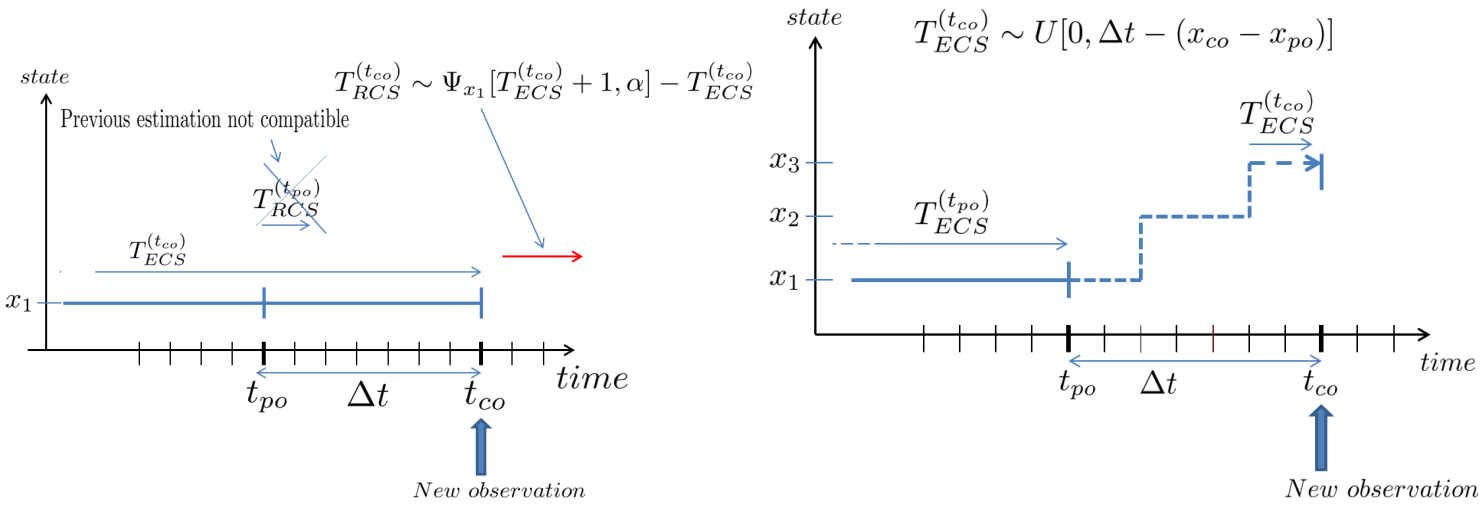
Algorithm 1 Prognosis algorithm

Initialization:

$\forall i = 1..x_F - 1\ ,\ \ S_i^{(0)} \sim \Psi_i$ (Initial random sampling of sojourn times)
$R^{(0)} = \sum_{j=x^{(0)}+1}^{x_{F-1}} S_j^{(0)}$
$T_{ECS}^{(0)}$ initialized by the user depending the past of the component
$T_{RCS}^{(0)} \sim \Psi_{x^{(0)}}([T_{ECS}^{(0)} + 1, \alpha^{MAX}] - T_{ECS}^{(0)}$
$RUL^{(0)} = T_{RCS}^{(0)} + R^{(0)}$

Update: Let $x^{(t_{co})}$ be a new available diagnosis result :

if $\mathbf{x^{(t_{co})}} = \mathbf{x^{(t_{po})}}$ (case where no transition is observed) **then**
  $T_{ECS}^{(t_{co})} = T_{ECS}^{(t_{po})} + \Delta t$
  if $(T_{RCS}^{(t_{po})} > \Delta t)$ (case where the previous estimation is compatible) **then**
    $T_{RCS}^{(t_{co})} = T_{RCS}^{(t_{po})} - \Delta t$
  else
    $T_{RCS}^{(t_{co})} = \Psi_{x^{(t_{co})}}[T_{ECS}^{(t_{co})} + 1, \alpha^{MAX}] - T_{ECS}^{(t_{co})}$
  end if
  $R^{(t_{co})} = R^{(t_{po})}$
else if $(\mathbf{x^{(t_{co})}} - \mathbf{x^{(t_{po})}} = \mathbf{1})$ (if a transition is observed) **then**
  $R^{(t_{co})} = R^{(t_{po})} - S_{x^{(t_{co})}}^{(0)}$
  if $(T_{RCS}^{(t_{po})} \leq \Delta t)$(if the previous estimation of T_{RCS} is compatible) **then**
    $T_{ECS}^{(t_{co})} = \Delta t - T_{RCS}^{(t_{po})}$
  else
    $T_{ECS}^{(t_{co})} = \Delta t - \Psi_{x^{(t_{po})}}[T_{ECS}^{(t_{po})} + 1, T_{ECS}^{(t_{po})} + \Delta t] + T_{ECS}^{(t_{po})}$
  end if
  if $R^{(t_{co})} - R^{(t_{po})} > T_{ECS}^{(t_{co})}$ (if the initial SJ sampling is compatible) **then**
    $T_{RCS}^{(t_{co})} = R^{(t_{co})} - R^{(t_{po})} - T_{ECS}^{(t_{co})}$
  else
    $T_{RCS}^{(t_{co})} = \Psi_{x^{(t_{co})}}[T_{ECS}^{(t_{co})} + 1, \alpha^{MAX}] - T_{ECS}^{(t_{co})}$
  end if
else if If $k = \mathbf{x^{(t_{co})}} - \mathbf{x^{(t_{po})}} > \mathbf{1}$ (if many transitions are observed) **then**
  $T_{ECS}^{(t_{co})} \sim U[0, \Delta t - k]$
  $T_{RCS}^{(t_{co})} = \Psi_{x^{(t_{co})}}[T_{ECS}^{(t_{co})} + 1, \alpha_{x^{(t_{co})}}^{MAX}] - T_{ECS}^{(t_{co})}$
  $R^{(t_{co})} = R^{(t_{po})} - \sum_{j=x^{(t_{po})}+1}^{x^{(t_{co})}} S_j$
end if
At last $RUL^{(t_{co})} = T_{RCS}^{(t_{co})} + R^{(t_{co})}$
$\forall t \in]t_{co}, t_{fo}[\quad RUL^{(t_{po})} = RUL^{(t_{co})} - (t - t_{co})$ where *fo* is the future observation.



(a) T_{RCS} updating in the case of incompatibility

(b) T_{ECS} updating in the case where $x_{co} - x_{po} > 1$

Fig. 10. Update process in some cases

$$\epsilon(RUL^{(t)}) = \sum_{j=x^{(t)}}^{x_N - 1} \epsilon(S_j^{(t)})$$

The quality of the RUL computations depends on the initial sojourn time sampling and the variance of the sojourn time distributions, because the probability $p(\epsilon(S_j^{(t)}) < k)$ is directly linked to the variance of the STD associated with state j. Consequently, one important question is to find a way to reduce the variance of the considered STDs. The purpose of the extended GDM is to limit this problem. The next section describes a modification of algorithm 1 to be adapted to the extended GDM-CSTD degradation model.

c) Extension to conditional sojourn time :

The use of the extended GDM using CSTD requires adapting the prognosis algorithm. In this case, the prognosis algorithm is modified as follows :

Initialization :

- According to the original random sampling for sojourn time in state 1, the most likely mode is determined using the conditional a posteriori probabilities (mode given the sojourn time). Then, future sojourn times are chosen according to the conditional distribution associated with this mode. So $R^{(0)}$ and $T_{RCS}^{(0)}$ computations are modified.

Update :

Each time T_{ECS} is updated, the most likely active mode is evaluated according to the new value of T_{ECS} as sojourn time. If the new mode is different, then the new value of T_{RCS} is chosen according to the conditional distribution associated with this new mode. $R^{(t)}$, that is the sum of all sojourn times in future states until failure state, is completely re-estimated, by performing a new drawing for all sojourn time, according to the new conditional sojourn time distributions associated with the new detected mode. All these modifications lead to algorithm 2. $m^{(co)}$ is denoted as the active mode at time co . Notation $\Psi(x)$ is the value of the STD at x. $\Psi(x)/m^{(co)}$ is the CSTD in state x given the active mode $m^{(co)}$.

Algorithm 2 Extended version of the prognosis algorithm using CSTD

Initialization:

$S_1^{(0)} \sim \Psi_1$
for i=2 ... x_F-1 **do**
  $m = \arg\max_k(\Psi_{i-1/k}(S_{i-1}^{(0)}))$
  $S_i^{(0)} \sim \Psi_{i/m}$
end for
$m^{(po)} = m$
$R^{(0)} = \sum_{j=x^{(0)}+1}^{x_{F-1}} S_j^{(0)}$
$T_{ECS}^{(0)}$ initialized by the user depending the past of the component
$T_{RCS}^{(0)} \sim \Psi_{x^{(0)}/m}([T_{ECS}^{(0)} + 1, \alpha^{MAX}] - T_{ECS}^{(0)}$
$RUL^{(0)} = T_{RCS}^{(0)} + R^{(0)}$

Update: Let $x^{(t_{co})}$ be a new available diagnosis result :

if $\mathbf{x^{(t_{co})}} = \mathbf{x^{(t_{po})}}$ (case where no transition is observed) **then**
  $T_{ECS}^{(t_{co})} = T_{ECS}^{(t_{po})} + \Delta t$
  $m^{(co)} = \arg\max_k(\Psi_{x^{(co)}/k}(T_{ECS}^{(co)}))$
  if $(T_{RCS}^{(t_{po})} > \Delta t)$ (case where the previous estimation is compatible) **then**
    $T_{RCS}^{(t_{co})} = T_{RCS}^{(t_{po})} - \Delta t$
  else
    $T_{RCS}^{(t_{co})} = \Psi_{x^{(t_{co})}/m^{(co)}}[T_{ECS}^{(t_{co})} + 1, \alpha^{MAX}] - T_{ECS}^{(t_{co})}$
  end if
  $R^{(t_{co})} = R^{(t_{po})}$
else if $(\mathbf{x^{(t_{co})}} - \mathbf{x^{(t_{po})}} = \mathbf{1})$ (if a transition is observed) **then**
  $R^{(t_{co})} = R^{(t_{po})} - S_{x^{(t_{co})}}^{(0)}$
  if $(T_{RCS}^{(t_{po})} \leq \Delta t)$ **then**
    $T_{ECS}^{(t_{co})} = \Delta t - T_{RCS}^{(t_{po})}$
  else
    $T_{ECS}^{(t_{co})} = \Delta t - \Psi_{x^{(t_{po})}/m^{(po)}}[T_{ECS}^{(t_{po})} + 1, T_{ECS}^{(t_{po})} + \Delta t] + T_{ECS}^{(t_{po})}$
  end if

  $m^{(co)} = \arg\max_k(\Psi_{x^{(po)}/k}(T_{ECS}^{(po)} + \Delta t - T_{ECS}^{(co)}))$
  if $R^{(t_{co})} - R^{(t_{po})} > T_{ECS}^{(t_{co})}$ **then**
    $T_{RCS}^{(t_{co})} = R^{(t_{co})} - R^{(t_{po})} - T_{ECS}^{(t_{co})}$
  else
    $T_{RCS}^{(t_{co})} = \Psi_{x^{(t_{co})}/m^{(co)}}[T_{ECS}^{(t_{co})} + 1, \alpha^{MAX}] - T_{ECS}^{(t_{co})}$
  end if
end if
if $m^{(co)} = m^{(po)}$ **then**
  $R^{(t_{co})} = R^{(t_{po})} - \sum_{j=x^{(t_{po})}+1}^{x^{(t_{co})}} S_j$
else
  $\forall j \in [x^{(co)} + 1, x_F - 1]$,$S_j \sim \Psi_{j/m^{(co)}}$
  $R^{(t_{co})} = \sum_{j=x^{(co)}+1}^{x_{F-1}} S_j$
end if
$m^{(po)} = m^{(co)}$
At last $RUL^{(t_{co})} = T_{RCS}^{(t_{co})} + R^{(t_{co})}$
$\forall t \in]t_{co}, t_{fo}[\quad RUL^{(t_{po})} = RUL^{(t_{co})} - (t - t_{co})$ where *fo* is the future observation.

All these modifications leads to algorithm 2. The main objective of CSTD is to get more precise modelling of the degradation process considering smaller variance distributions with. This prognosis algorithm needs now to be represented as a DBN.

3.2. Representation of the prognosis algorithm as a DBN

This part introduces graphical representation of the prognosis algorithm presented in section 3.1 as a DBN in order to introduce it into the global VirMaLab DBN as a prognosis module.
An intuitive way to proceed is to define a node for each intermediate variable defined in the prognosis algorithm.

So, variables T_{ECS} and T_{RCS} are represented and some other have to be added.

- $D^{(t)}$ is the result given by the diagnosis device when it is enabled.
- $Act^{(t)}$ is a binary node set on one if the diagnosis device is enabled
- Δt : In the case of one transition , it determines how many states have been skipped.
- $C_1^{(t)}$and $C_2^{(t)}$: represent respectively the current active mode in the degradation module and in the prognosis module.

We remind that the purpose of the nodes that are in the hidden degradation model.
- $X^{(t)}$ is the hidden state of the system
- $S^{(t)}$ is the sojourn time variable associated to X^t (from GDM)

Each relationship dependency is represented by an arrow in the graph. The purpose of node Δt is to represent the conditional structure "if then" in the algorithm. The presence of $C_2^{(t)}$ node is due to the use of conditional sojourn time distributions because the choice of the degradation mode influences the computations of three variables in the prognosis computation ($T_{RCS}^{(t)}$,$T_{SEC}^{(t)}$ and $R^{(t)}$).
The set of nodes is split into three groups:
- Green nodes are linked to the hidden degradation model.
- Blue nodes represent the variables used in the prognosis algorithm.
- Purple nodes are linked to the use of CSTD.

The prognosis part of the DBN is a graphical representation of the computing process of the prognosis algorithm described in part 3.1. The interest of this DBN is to be able to perform inference computations to get out new informations from the model. Figure 11 gives a representation of the prognosis algorithm as a specific DBN and its link with the extended degradation module from the VirMalaB model. The prognosis DBN for algorithm 1 is the same without purple nodes).

The properties of the prognosis DBN presented above are studied.
N_S,T_S and N_M : denote respectively the number of state of the modelled system, the maximal value for sojourn time in all states and the number of considered degradation modes.
The size of the range of node X is denoted $|X|$. Thus:

$|X^{(t)}| = |D^{[t)}| = |\Delta^{(t)}| = N_S$.
$|C_1^{(t)}| = |C_2^{(t)}| = N_M$.
$|S| = |T_{ECS}| = |T_{RCS}| = T_S$
$|RUL| = |R^{(t)}| = (N_S - 1)T_S$
The set of nodes can be divided into two groups. The nodes that have a small domain (X, $Act^{[t)}$,$D^{(t)}$,$\Delta^{(t)}$) and those which have a large domain (S, T_{ECS}, T_{RCS}, R,RUL).

Moreover, the transition laws in nodes are deterministic or semi-deterministic. By using an inference algorithm, this model can compute the future distributions of the RUL and provide a confidence interval associated to with RUL computations.
Given t : the time of the last diagnosis, α: the RUL computed at time t, x_F, the failure state and $\{d_1, d_2, ...d_n\}$: the set of

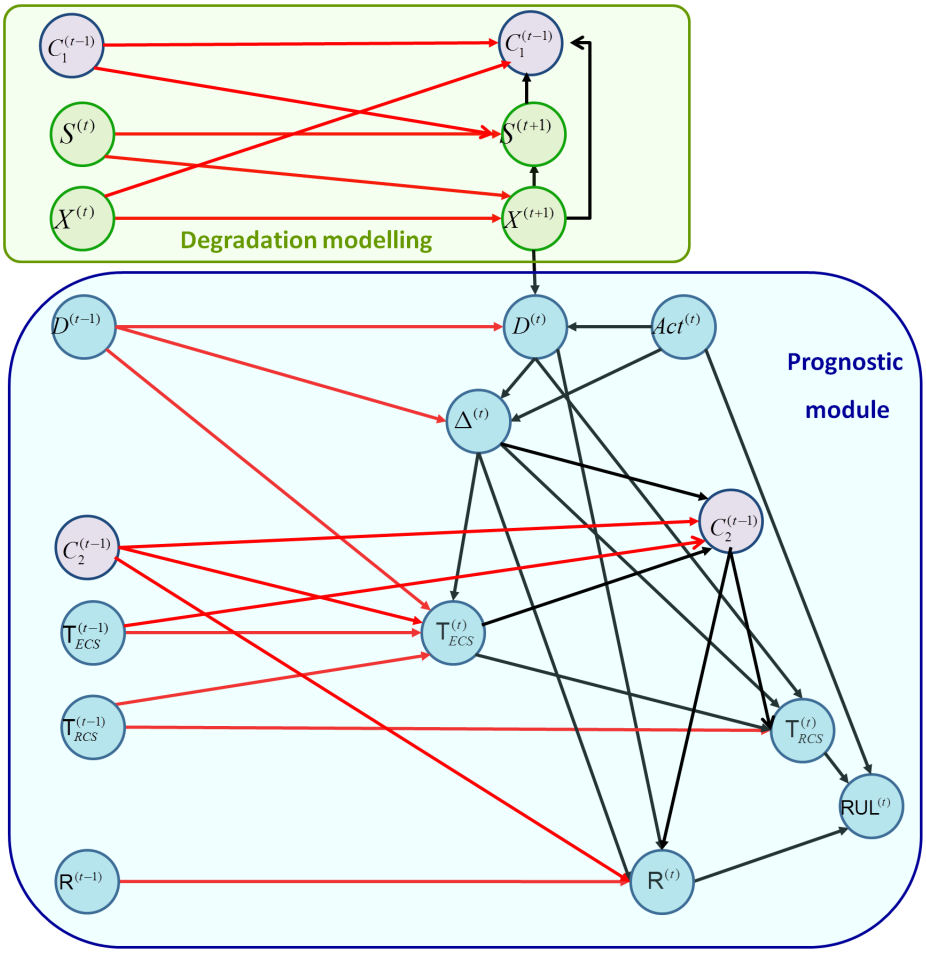


Fig. 11. The extended prognosis algorithm as a DBN

all previous diagnosis results: according to this model, the probability for the error associated with the RUL computation to be inferior to ϵ is :
$p(Err^{(t)} < \epsilon) = \sum_{|k|<\epsilon} p(X^{(t+\alpha+k)} = x_F, X^{(t+\alpha+k-1)} \neq x_F | RUL^{(t)} = \alpha, D^{(1)} = d_1, D^{(2)} = d_2, .., D^{(t)} = d_n)$

With the conditional probability relationship induced by the graph, $X^{(t+\alpha+k)}, X^{(t+\alpha+k-1)}$ is independent from RUL^t given $(D_1, ..D_t)$ so, this probability is equal to

$$p(Err_t < \epsilon) = \sum_{|k|<\epsilon} p(X^{(t+\alpha+k)} = x_F, X^{(t+\alpha+k-1)} \neq x_F | D^{(1)} = d_1, D^{(2)} = d_2, .., D^{(n)} = d_n)$$

We then face to a prediction problem. If we consider a set of random values following a given probability distribution, the mathematical expectancy has the property to minimize the total of the distances squared. In other words, it minimizes the risk of error. The use of the variant method in the prognosis algorithm, leads to a RUL estimation that minimizes the risk of error.
In section 3, the inference process that is used to compute this probability will be detailed.

3.3. Inference process and algorithm for computations

Two different approaches are commonly used to perform inference computations in a DBN **?** :
- Exact inference.
- Approximate inference

Approximate inference consists in using a Monte-Carlo simulation. In this case many trajectories are simulated with the DBN, and the frequency of trajectories corresponding to the inference request converge to the desired probability. The drawback is that many simulations are needed to hope to get accurate an estimation of the desired probability.
The exact methods lays on algebraic computations on CPTs of the nodes. In this paper we focus on the elimination variable algorithm**?** . Let remember that in a first order DBN, the left interface (I^-) is defined as the set of nodes that have at least one child in the next time slice. The interface algorithm from **?** is an improvement of the elimination variable algorithm using the Markov property in a DBN. It rely on the following proposition.
All of the nodes at time t+1 are independent from all the past given the left interface. So to get the joint law at slice t+1 is just needed the joint law of the left interface at time t. In our case, prediction computation are requested (that is to say of type : $p(X^{(t+\Delta t)}|X^{(t)})$ so, a forward process is used to compute the desired probability (ie nodes on slice t-1 are eliminated before nodes from slice t). But this algorithm requires to stock the left interface for each iteration slice update.
In the prognosis DBN presented in figure 11, the left interface contains 8 nodes.

$$I^- = (X^{(t)}, S^{(t)}, C_1^{(t)}, D^{(t)}, T_{ECS}^{(t)}, C_2^{(t)}, T_{RCS}^{(t)}, R^{(t)})$$

Because all of these nodes represent quantitative values, the size of each potential is large, and the order of magnitude of the potential product of the joint law of the left interface is

$$|I^-| = N_M^2 N_S^2 T_S^4$$

The classical update of the left interface requires computing $p(I^{-(t)}) \times p(I^{-(t+1)})$. This implies stocking temporarily in the RAM memory a potential with a size at an order of magnitude of $(N_M^2 N_S^2 T_S^4)^2 = N_M^4 N_S^6 T_S^8$.
So, we need to find a way to avoid stocking the complete left interface. We will propose a specific algorithm adapted to the prognosis DBN. It is a modification of the interface algorithm that aims to perform a slice update without stocking all the left interface.
The principle is to conduct the forward process by using conditional independence relationships and to compute some specific conditional probability to avoid the computation of a large product. The left interface is split into three groups:

- group 1 : $(X^{(t)}, S^{(t)}, C_1^{(t)})$
- group 2 : $(D^{(t)}, C_2^{(t)}, T_{ECS}^{(t)}, T_{RCS}^{(t)})$
- group 3 : $(R^{(t)}, C_2^{(t)})$

The general inference process is detailed in algorithm 4 and 5. Its goal is to compute the joint law of the three groups then to update them from slice t to slice t+1. It computes the distribution of $RUL^{(t)}$ too. The notation $n(X)$ means that the CPT (the potential) of node X is considered. The $\rightarrow$ symbol means that a marginalization operation or a division by a subset of variables has been used to determine the probability on the right side.
The largest potential now has an order of magnitude of : $T_S^3 N_S^3 N_S^2$

Error computation

As mentioned in the previous section, the error computation is a conditional probability with the form

$$p(X^{(t+h)}, X^{(t+h-1)}|D^{(1)} = d_1, D^{(2)} = d_2, .., D^{(n)} = d_n)$$

In the inference process detailed in algorithm 4 and 5, the conditional information can be directly introduced in the forward step. Because, in this version of the model, the RUL has not been connected yet to the maintenance nodes, the error computation only requires an inference on the set of group 1 nodes. So, algorithm 3 only shows the modified inference update process for group 1

Algorithm 3 Inference Modification on group 1 for introducing conditionning with D_t

We suppose that we have $P(X^{(t)}, S^{[t)}, C_1^{(t)}|D^{(1)} = d_1, ..D^{(t-1)} = d_{t-1})$ and we want to compute $P = p(X^{(t+1)}, S^{(t+1)}, C_1^{(t+1)}|D^{[1)} = d_1, .., D^{(t)} = d_t)$

1: $P(X^{(t)}, S^{[t)}, C_1^{(t)}|D^{(1)} = d_1, .., D^{(t-1)} = d_{t-1}) \times n(X^{(t+1)}) \times n(S^{[t+1)}) \times n(C_1^{(t+1)}) \rightarrow p(X^{(t)}, X^{(t+1)}, S^{(t+1)}, C_1^{(t+1)}|D^{[1)} = d_1, .., D^{(t-1)} = d_{t-1})$

2: $p(X^{(t)}, X^{(t+1)}, S^{[t+1)}, C_1^{(t+1)}|D^{[1)} = d_1, .., D^{(t-1)} = d_{t-1}) \times p(D^{(t)}|X^{(t)}|D^{(1)}, ..D^{[t-1)}) \rightarrow p(X^{(t+1)}, S^{(t+1)}, C_1^{(t+1)}, D^{(t)}|D^{(1)} = d_1, ..D^{(t-1)} = d_{t-1})$

Let be M(:,:,:,:) the potential for $p(X^{(t+1)}, S^{(t+1)}, C_1^{(t+1)}, D^{(t)}|D^{(1)} = d_1, .., D^{(t-1)} = d_{t-1})$

If the conditioning is $D^{[t)} = d$ then we consider $P = M(:, :, :, d)$ and $P \rightarrow \frac{P}{p(D^{(t)}=d_t)}$ then we have $P = p(X^{(t+1)}, S^{(t+1)}, C_1^{(t+1)}|D^{(1)} = d_1, .., D^{(t)} = d_t)$

Algorithm 4 Specific inference algorithm adapted to the prognosis DBN : part 1 : Initialization

Initial slice t=0:

1: $n(X^{(0)}) \times n(S^{(0)}) \times n(C_1^{(0)}) \rightarrow \mathbf{p(X^{(0)}, S^{(0)}, C_1^{(0)})}$ **(group)**

2: $n(X^{(0)}) \times n(Act^{(0)}) \times n(D^{(0)}) \times n(\Delta^{(0)}) \rightarrow p(X^{(0)}, D^{(0)}, \Delta^{(0)})$

3: $p(X^{(0)}, S^{(0)}, C_1^{(0)}, D^{(0)}, \Delta^{(0)}) \rightarrow \mathbf{p(D^{(0)}, \Delta^{(0)})}$

4: $p(X^{(0)}, S^{(0)}, C_1^{(0)}, D^{(0)}, \Delta^{(0)}) \rightarrow p(D^{(0)}, X^{(0)}) \rightarrow \mathbf{p(D^{(0)}|X^{(0)})}$

5: $\mathbf{p(D^{(0)}, \Delta^{(0)})} \times n(T_{ECS}^{(0)}) \times n(C_2^{(0)}) \times n(T_{RCS}^{[0)}) \rightarrow p(D^{(0)}, \Delta^{(0)}, T_{ECS}^{(0)}, C_2^{(0)}, T_{RCS}^{(0)}) \rightarrow \mathbf{p(D^{(0)}, T_{ECS}^{(0)}, C_2^{(0)}, T_{RCS}^{(0)})}$ **(group 2)**

6: $p(D^{(0)}, \Delta^{(0)}, T_{ECS}^{(0)}, C_2^{(0)}, T_{RCS}^{(0)}) \rightarrow p(D^{(0)}, \Delta^{(0)}, C_2^{(0)})$

7: $p(D^{(0)}, \Delta^{(0)}, T_{ECS}^{(0)}, C_2^{(0)}, T_{RCS}^{(0)}) \rightarrow p(T_{RCS}^{(0)}|D^{(0)}, \Delta^{(0)}, C_2^{(0)})$

8: $p(D^{(0)}, \Delta^{(0)}, C_2^{(0)}) \times n(R^{(0)}) \times p(T_{RCS}^{(0)}|D^{(0)}, \Delta^{(0)}, C_2^{(0)}) \rightarrow \mathbf{p(D^{[0)}, C_2^{(0)}, R^{(0)})}$ **(group 3)** and $p(T_{RCS}^{(0)}, R^{(0)})$

9: $p(T_{RCS}^{(0)}, R^{(0)}) \times n(RUL^{(0)}) \rightarrow \mathbf{p(RUL^{(0)})}$

Algorithms 3 and 4 use the conditional independence relationships induced by the graph to decrease the complexity of the update part of inference computation. But the main problem in terms of complexity concerns the update of node T_{ECS} and T_{RCS}, because of the size of their potentials and the high number of parents. It is the drawback of using a discrete approach. In the analytic approach, analytic expressions are hard to determine, and in most cases, the obtained integrals

Algorithm 5 Specific inference algorithm adapted to the prognosis DBN : part 2 : update from slice t to slice t+1

1: $p(X^{(t)}, S^{(t)}, C_1^{(t)}) \times n(X^{(t+1)}) \times n(S^{[t+1)}) \rightarrow p(X^{(t)}, C_1^{(t)}, X^{(t+1)}, S^{(t+1)})$
2: $p(X^{(t)}, C_1^{(t)}, X^{(t+1)}, S^{(t+1)}) \times n(C_1^{(t+1)}) \rightarrow \mathbf{p}(\mathbf{X^{(t+1)}}, \mathbf{S^{(t+1)}}, \mathbf{C_1^{(t+1)}})$ **(New group 1)** and $p(X^{(t)}, X^{(t+1)})$

3: $p(X^{(t)}, X^{(t+1)}) \times \mathbf{p}(\mathbf{D^{(t)}}|\mathbf{X^{(t)}}) \times n(Act^{(t+1)}) \times n(D^{(t+1)}) \rightarrow p(D^{[t+1)}|D^{[t)})$ and $p(D^{(t+1)}, X^{(t+1)})$
4: $X = \mathbf{p(bloc2)}$
5: $X \times p(D^{(t+1)}|D^{(t)}) \times n(\Delta^{[t+1)}) \rightarrow p(X, D^{(t+1)}, \Delta^{(t+1)})$
6: $p(X, D^{(t+1)}, \Delta^{(t+1)}) \times n(T_{ECS}^{(t+1)}) \times n(C_2^{(t+1)}) \rightarrow \mathbf{p}(\mathbf{X}, \mathbf{D^{(t+1)}}, \mathbf{\Delta^{(t+1)}}, \mathbf{T_{ECS}^{(t+1)}}, \mathbf{C_2^{(t+1)}})$
7: $p(X, D^{(t+1)}, \Delta^{(t+1)}, T_{ECS}^{(t+1)}, C_2^{(t+1)}) \rightarrow p(T_{RCS}^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, T_{ECS}^{(t+1)}, C_2^{(t+1)})$
8: $p(T_{RCS}^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, T_{ECS}^{(t+1)}, C_2^{(t+1)}) \times n(T_{RCS}^{(t+1)}) \rightarrow p(D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}, T_{RCS}^{(t+1)})$ and $\mathbf{p}(\mathbf{D^{(t+1)}}, \mathbf{T_{ECS}^{(t+1)}}, \mathbf{C_2^{(t+1)}}, \mathbf{T_{RCS}^{(t+1)}})$ **(New group 2)**

9: $p(D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}, T_{RCS}^{(t+1)}) \rightarrow p(T_{RCS}^{(t+1)}|D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)})$
10: $\mathbf{p}(\mathbf{X}, \mathbf{D^{(t+1)}}, \mathbf{\Delta^{(t+1)}}, \mathbf{T_{ECS}^{(t+1)}}, \mathbf{C_2^{(t+1)}}) \rightarrow p(D^{(t)}, C_2^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}) \rightarrow p(D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}|D^{(t)}, C_2^{(t)})$
11: $\mathbf{Y = n(group\ 3)}$
12: $Y \times p(D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}|D^{(t)}, C_2^{(t)}) \rightarrow p(C_2^{(t)}, R^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)})$
13: $p(C_2^{(t)}, R^{(t)}, D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}) \times n(R^{(t+1)}) \rightarrow p(D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}, R^{(t+1)})$
14: $p(D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}, R^{(t+1)}) \rightarrow \mathbf{p}(\mathbf{D^{(t+1)}}, \mathbf{C_2^{(t+1)}}, \mathbf{R^{(t+1)}})$ **(New group 3)**
15: $p(D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}, R^{(t+1)}) \times p(T_{RCS}^{(t+1)}|D^{(t+1)}, \Delta^{(t+1)}, C_2^{(t+1)}) \rightarrow p(T_{RCS}^{(t+1)}, R^{(t+1)})$
16: $p(T_{RCS}^{(t+1)}, R^{(t+1)}) \times n(RUL^{(t+1)} \rightarrow \mathbf{p}(\mathbf{RUL^{(t+1)}})$

are not directly computable and need to be approximated using numeric methods, whereas discrete expressions are easy to write and are a sequence of additions and products on potentials but the size of computation is very large because of the high number of numerical values we consider for each node. For this reason, with a computer with 8 Go RAM, T_S value is limited to 70 to avoid RAM overloading.

4. Application

Context

The aim of this section, is to use the prognosis algorithm and its representation to produce some prognostic results, and compute an associated confidence interval. In this section, we work on simulated databases. To show the interest of conditional sojourn time distributions, data are simulated with a mixture of two degradation modes. A four discrete states system is considered. It is supposed to be periodically observable. The sojourn time distributions for each state are a mixture of two Weibull distributions with the following parameters.

- state 1 :

mode 1 : $W(\alpha = 15, \beta = 2)$ mode 2 : $W(\alpha = 33, \beta = 6)$

- state 2 :

mode 1 : $W(\alpha = 10, \beta = 6)$ mode 2 : $W(\alpha = 25, \beta = 9)$

- state 3 :

mode 1 : $W(\alpha = 5, \beta = 6)$ mode 2 : $W(\alpha = 15, \beta = 15)$

Each degradation mode has the same weight(0.5,0.5). From the previous sojourn time distributions, a discrete database was simulated. The interval between two diagnosis was set as 5 time units. The generated database contained 10000 observation sequences with 500 in each mode.

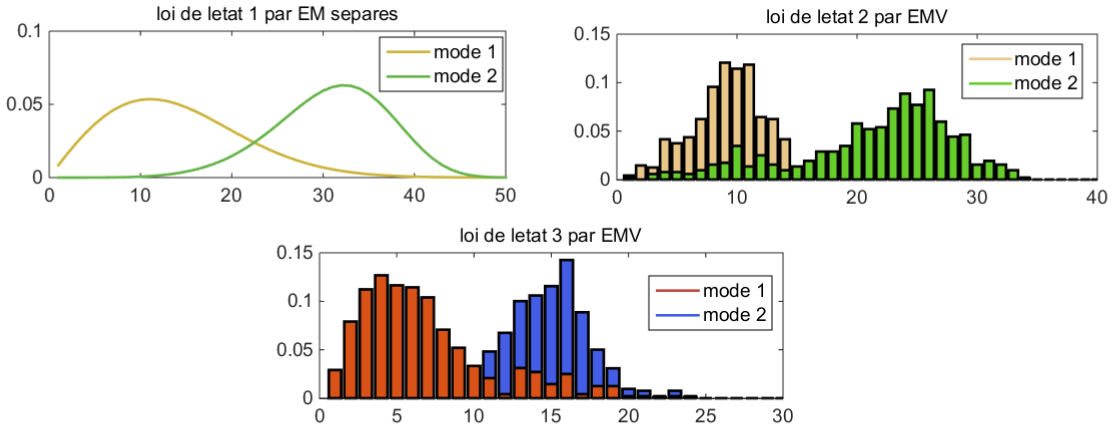


Fig. 12. STD learnt with EM algorithm and Max likelihood method:

Learning

About the first state, the parameters of the mixing model were learnt using the EM algorithm **?**. Then, the most likely mode was associated with each observation of the database using the a posteriori probabilities. And the CSTD for future states were learnt using the maximum likelihood method on subsets of observations associated with each mode. Because we are in a discrete case, the probability of a sojourn time is its frequency of appearance in the simulated database. The STD are shown in Figure 12.

Test of the prognosis algorithm

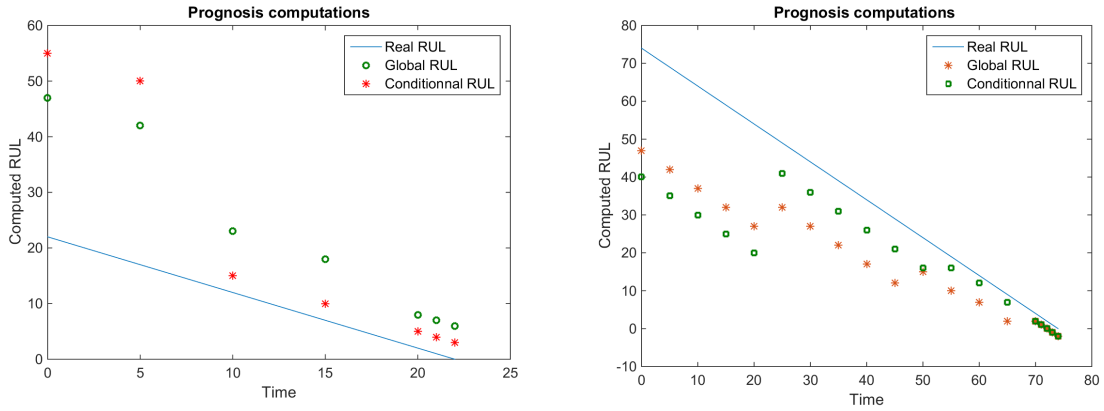
Two observation sequences were chosen from the simulated database. The first observation was generated with the first degradation mode (mode 1) and the second one with mode 2. The prognosis algorithm was launched from these observation sequence. It computed a RUL estimation at each available diagnosis. the variant method of the algorithm was used, so the sojourn time are drawn taking the average value of the STDs. The real RUL was deduced from the real failure time that was given because the real hidden sojourn times that had been used to build the observation sequences are known. The RUL estimated with the prognosis algorithm using the conditional STD (Conditional RUL) was computed and compared with the results of the RUL computed by applying the prognosis algorithm with the original GDM (with independent STD). Numerical results produced with the prognosis algorithm are shown in Figure 13,14. Then, the results are plotted for comparison on figure 15. As expected, the predictions were logically better using the conditional STD, than using the global independent STD. The real active mode was detected after the first transition, so the predictions with CSTD were better. The intermediate sojourn times for each state could then be updated, only when a transition was detected, or due to a sojourn time extension.

Time	0	5	10	15	20	25
Observed sequence	1	1	2	2	3	4
Real RUL	22	17	12	7	2	0
Global RUL	47	42	23	18	8	0
Conditional RUL	55	50	15	10	5	0
Active mode detected	1	1	1	1	1	1

Fig. 13. Prognosis result on Obs 1

Time	0	5	10	15	20	25	30	35	40	45	50	55	60	65	70	75
Observed sequence	1	1	1	1	1	1	1	2	2	2	2	2	3	3	3	4
Real RUL	74	69	64	59	54	49	44	39	34	29	24	19	14	9	4	0
Global RUL	47	42	37	32	27	32	27	22	17	12	15	10	7	2	2	0
Conditional RUL	40	35	30	25	20	41	36	31	26	21	16	16	12	7	2	0
Active mode detected	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2

Fig. 14. Prognosis result on Obs 2



(a) Prognosis on the first obs sequence

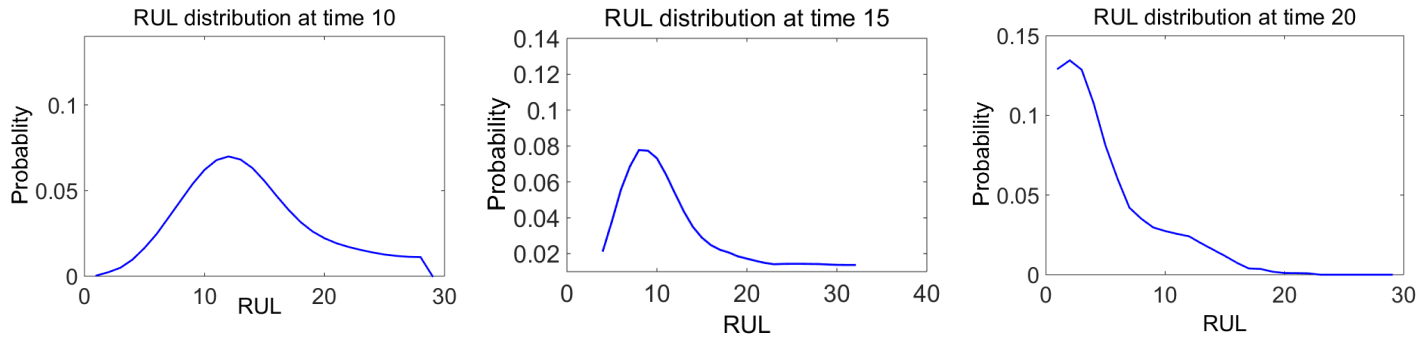
(b) Prognosis on the second obs sequence

Fig. 15. Prognosis results on two observation sequences

A confidence interval was computed on the computed RUL at times 10,15,20 from the first observation sequence using the inference algorithm. The results are shown in Figure 16. The transition at time 15 to state 2 provides new information on sojourn time and the risk of error logically decreases.

To have an more complete view of the behaviour of the prognosis algorithm, the computations are now performed on the 1000 observation sequences from the simulated database. Figure 17 compares the evolution of the committed error for trajectories generated with mode 1 and those generated with mode 2.

For mode 1 trajectories, i.e. faster degradation scenarios, we can observe the same global behaviour of the RUL estimation algorithm. Nevertheless, this case also illustrates the drawback of the proposed approach that might be improved in further works. Globally, 34 of the 500 trajectories in mode 1 have at least one sojourn time in the intersection range of two modes, inducing a wrong estimation of the most probable degradation mode, such a way that the adaptation ability of the algorithm, cannot process in these cases and the RUL estimation is absolutely unusable.



(a) Confidence interval at time 10 (b) Confidence interval at time 15 (c) Confidence interval at time 20

Fig. 16. Confidence interval on RUL estimations

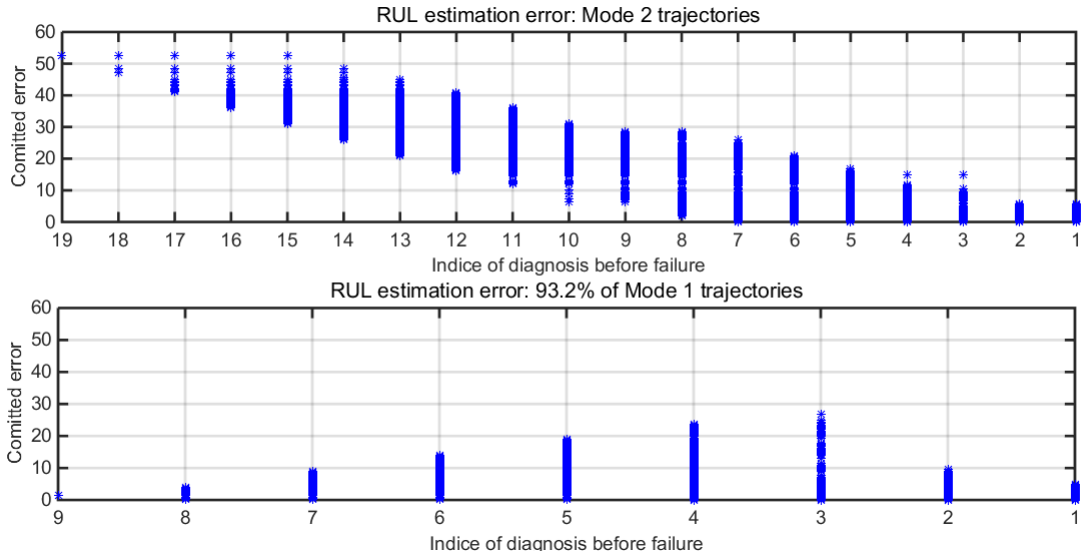


Fig. 17. Error comparison depending on the mode :

4.1. Conclusion and perspective

In this paper, the formalism of Probabilistic Graphical Models was investigated to perform prognosis computations for systems with discrete and finite states space. This approach is based on a semi-Markovian degradation model called "Graphical duration model" in order to be not dependent of the assumption of geometrical distributed STDs implied by the use of a classical markovian approach. A prognosis algorithm and its representation as a DBN has been presented. It can be used to perform prognosis computation based on periodic available diagnoses. The quality of the prognosis computation depends on the variance of STD . One possible way to reduce this variance is to use the CSTD. Nevertheless, some convergence problems remains, particularly for cases with very fast degradation speed and for case where real sojourn time are at the intersection of two CSTD from different modes. This algorithm can be represented as a DBN to compute a confidence interval around the prediction. The inference methods has been used to compute confidence interval.

The next step consists in integrating the prognosis module to the global VirmaLab model in order to represent global maintenance strategies based on prognosis. The main remaining problems concerns the space complexity of computations. The specific inference algorithm only make possible to use a STD with a size inferior to 70 time units. Moreover, the next step consisting in the integration of the prognostic module to the VirmalaB model, will again increase the size of the interface of the model and particularly the future connexion between the RUL and the maintenance action nodes. One

way we are working on is to use a sparse structure in the programming to take into account the fact that most of the potentials in the RBD are sparse or semi-sparse. The problem then is to find a good storage format that can speed up the computation time of the two elementary operations commonly used in the inference process : **potential product** and **potential marginalization**. The final VirmalaB model could be used to optimize every kind of maintenance policy and, particularly, the predictive maintenance strategies.

References

Bouillaut L., Donat R., Neji A., Aknin P. ; *Modelling the maintenance of multi-components system: Comparison of two Graphical Models approaches*, 13th IFAC Symposium on Information Control Problems in Manufacturing, Moscow, Russia, (2009)

Bouillaut L., Aknin P., Donat R., Bondeux S. ; *VirMaLab : A generic approach for optimizing maintenance policies for complex systems*, 9th World Congress on Railway Research (WCRR), (2011)

Byington C.S, Roermer M.J, Kacprzynski G.J ; *Prognostic Enhancements to Diagnostic Systems for Improved Condition-Based Maintenance*, IEEE Aerospace Conference, 2002

Dechter R., *Bucket elimination : A unifying framework for reasoning*, Artificial Intelligence 113(1-2), 41-85.

Dempster A.P., Laird N. M., Rubin D. B. , 1977. *Maximum likelihood from incomplete data via the EM algorithm*, 1977

Donat R., Leray P., Bouillaut L., Aknin P. ; *A dynamic Bayesian network to represent discrete duration models*, Neurocomputing, Volume 73, Issues 4âŁ“6, January, Pages 570-577 (2010)

Donat R. ; *Modelization of reliability and maintenance using graphical probabilistic models*, PhD thesis (2009)

Foulliaron J., Bouillaut L., Barros A. & Aknin P., *An extension of Graphical Duration Models integrating conditional sojourn time distributions*, Neurocomputing(2014-2015)

Horenbeek V., Scarf P.A, Cavalcante C.A.V. ; *On the use of predictive information in a joint maintenance and inventory policy*, Safety, Reliability and Risk Analysis: Beyond the Horizon âŁ“ Steenbergen et al. (Eds) Taylor & Francis Group (2014)

Jensen F.V ; *Bayesian Networks and Decision Graphs*, Springer, (2007)

Kothamasu R., Huang S. & Verduin W. ; *System health monitoring and prognostics: a review of current paradigms and practices*, International Journal of advanced manufacturing technology, 28(9-10), pp. 1012-1024, (2006)

Lebold M., Thurston M. ; *Open standard for conditions based maintenance and prognostic system*, In 5th Annual Maintenance and Reliability Conference (2001)

Murphy K.P ; *Dynamic Baysian network: Representation, inference and Learning*, PhD Thesis, (2002)

Nguyen K. ; Do Van P., Grall A. ; *A predictive maintenance strategy for multi-component systems using importance measure* , Safety, Reliability and Risk Analysis: Beyond the Horizon âŁ“ Steenbergen et al. (Eds) Taylor & Francis Group (2014)

J.Pearl ; *Bayesian Networks: A Model of Self-Activated Memory for Evidential Reasoning* , Proceedings of the 7th Conference of the Cognitive Science Society, p. 329âŁ“334,(1985)

Tobon-Mejia D.A. , Medjaher K., Zerhouni N., Tripot G. ; *Hidden Markov models for failure diagnostic and prognostic*, International Conference on Prognostics and Health Management - PHM, (2011)

AFNOR Norm X 60-000, 2002

Besnard F., Bertling L. ; *An Approach for Condition-Based Maintenance Optimization Applied to Wind Turbine Blades*, Sustainable Energy IEEE Transactions on (Volume 1 ,Issue 2), 2010