

Improved Dynamic Dependability Assessment through Integration with Prognostics

Jose Ignacio Aizpurua, Victoria M. Catterson, *Senior Member, IEEE*, Yiannis Papadopoulos, Ferdinando Chiacchio, and Gabriele Manno

Abstract—The use of average data for dependability assessments results in a outdated system-level dependability estimation which can lead to incorrect design decisions. With increasing availability of online data, there is room to improve traditional dependability assessment techniques. Namely, prognostics is an emerging field which provides asset-specific failure information which can be reused to improve the system level failure estimation. This paper presents a framework for prognostics-updated dynamic dependability assessment. The dynamic behaviour comes from runtime updated information, asset inter-dependencies, and time-dependent system behaviour. A case study from the power generation industry is analysed and results confirm the validity of the approach for improved near real-time unavailability estimations.

Index Terms—Prognostics, dynamic dependability, model to model transformation, risk monitor, remaining useful life, condition monitoring.

ACRONYMS AND ABBREVIATIONS

| | |
|-----------------|---------------------------------------|
| BDMP | Boolean logic Driven Markov Processes |
| CB | Circuit Breaker |
| CBFTA | Condition Based Fault Tree Analysis |
| CDF | Cumulative Distribution Function |
| DBN | Dynamic Bayesian Networks |
| ETA | Event Tree Analysis |
| FTA | Fault Tree Analysis |
| HAZOP | Hazard and Operability Study |
| IG | Input Gate |
| PDF | Probability Density Function |
| PHM | Prognostics and Health Management |
| PS | Power Supply |
| PSA | Probabilistic Safety Assessment |
| RUL | Remaining Useful Life |
| SAN | Stochastic Activity Networks |
| SD | Standard Deviation |
| SF ₆ | Sulphur Hexafluoride |
| OG | Output Gate |

NOTATION

L_i BDMP leaf i where $i=\{F, SF\}$

J. I. Aizpurua and V. M. Catterson are with the Institute of Energy and Environment, University of Strathclyde, Glasgow, UK (e-mail: jose.aizpurua@strath.ac.uk; v.m.catterson@strath.ac.uk).

Y. Papadopoulos is with the Department of Computer Science, University of Hull, Hull, UK (e-mail: Y.I.Papadopoulos@hull.ac.uk).

F. Chiacchio is with the Department of Industrial & Mechanical Engineering, University of Catania, Catania, Italy (e-mail: chiacchio@dmf.unict.it).

G. Manno is with the Strategic Research and Innovation Department, DNV GL, Høvik, Norway (e-mail: Gabriele.Manno@dnvgl.com).

| | |
|-------------------------|---|
| G_i | Fault tree gate i in the BDMP model |
| T_i | BDMP trigger i |
| N_i | BDMP node i |
| TE | Top-event |
| $Input(G_i)$ | Inputs of the gate G_i |
| $Orig(T_i)$ | Origin of the trigger i |
| $Dest(T_i)$ | Destination of the trigger i |
| λ | Constant failure rate: working to failed state |
| λ_s | Constant failure rate: standby to failed state |
| μ | Constant repair rate: failed to working or standby state |
| W | Working state |
| F | Failed state |
| R | Repair state |
| S | Standby state |
| P_i | Triggered Markov process of the leaf i |
| $Z_k^i(t)$ | Homogeneous Markov process of the leaf i , where $k = \{0, 1\}$ |
| A_k^i | State space of Z_k^i |
| $f_{a \rightarrow b}^i$ | Probability transfer function from process a to process b of the leaf i |
| act | SAN activation place |
| $deact$ | SAN deactivation place |
| $fault_{StandBy}$ | SAN timed activity from standby to failed state |
| β | Shape parameter of the Weibull distribution |
| η | Scale parameter of the Weibull distribution |
| T_{el} | Elapsed time of the conditional Weibull distribution |
| $F(t T_{el})$ | Conditional failure probability at time instant t given that T_{el} has already elapsed |
| T_p | Prediction time |
| T_m | Mission time |
| $m(x)$ | Marking of the SAN place x |
| F_x | Failure event of the component x |
| R_x | Repair event of the component x |
| F_{AA} | Insulation paper aging acceleration factor |
| Θ_{Ht} | Hotspot temperature at time instant t |
| L_t | RUL at time instant t |
| ξ_t | Process noise at time instant t |
| Θ_{to} | Top oil temperature |
| $\Delta\Theta_{to/a,R}$ | Temperature difference between top oil and ambient at rated current |
| K | Ratio of measured load to rated load |
| m | Cooling mode of the transformer |

I. INTRODUCTION

SYSTEM dependability is a term that encompasses a range of attributes which include safety, reliability, availability, maintainability, confidentiality, and integrity [1]. We will not consider confidentiality and integrity attributes because security aspects are outside of the scope of this paper. In the nuclear industry, a key tool for dependability assessment is a process called Probabilistic Safety Assessment (PSA). PSA employs a combination of Event Tree Analysis (ETA) and Fault Tree Analysis (FTA) to address reliability and safety. In their traditional forms, both techniques are inadequate and inaccurate when dealing with dynamic failure scenarios and system operation. Classical FTA is a combinatorial technique, i.e., focused on analysis of combinations as opposed to a sequence of failures and does not have effective means for representation of mode and state changes. Classical ETA explores successful or failed responses to a sequence of failures but, mathematically speaking, sequences are treated as combinations of events [2].

To enable a more accurate analysis of dynamic scenarios that include mode and state changes as well as sequencing of failures, several techniques for dynamic dependability analysis have emerged as alternatives to a classical PSA. Dynamic Fault Trees and Boolean Driven Markov Processes are examples of prominent emerging techniques [3]. Despite the advances made in this field, even recent dynamic techniques only provide a prediction of the dependability that is established *a priori*, i.e., before system deployment using average past operational data typically drawn from reliability databases. This prediction, however, leads to inaccurate estimates of system dependability attributes such as system safety and reliability that ignore the operational history and state of components used in the specific system.

In this paper, we argue that the increasing capabilities for condition monitoring and the availability of operational data in many engineering fields [4]-[6] create opportunities for changing this situation by forming a more accurate picture of the health of the system as it evolves during operation. Data from multiple sensors and monitoring systems can improve dependability prediction and inform maintenance planning.

Prognostics and Health Management (PHM) is a collection of activities focused on the system degradation management including the following main groups of activities [7], [8]:

- Anomaly detection: monitoring and detection of abnormal conditions in the system operation.
- Diagnosis: if an anomaly is detected, diagnose the cause of the fault.
- Prognostics: predict the likely future degradation of the component and estimate its remaining useful life.
- Operation and maintenance planning: mitigate the effects of failure and reduce unnecessary planned maintenance.

PHM techniques have emerged as promising solutions for cost-effective asset management. Traditionally a substantial focus of PHM has been on anomaly detection and diagnosis techniques applied in different fields such as nuclear [9], power transmission [10], or spacecraft [11]. During the last years there has been an increased interest in prognostics across different fields [12], [13]. The focus of this paper is on prognostics because it can be used within the dependability

assessment process integrating up-to-date health data and likely future degradation predictions. We will not focus on maintenance planning, but obtained results can be used for predictive and condition-based maintenance planning [14].

Prognostics is a relatively immature field, where few successful industrial implementations exist [12] while dependability analysis and PSA are well-known areas where many successful implementations exist across different industries [15]. Although the two share the goal of improving dependability attributes, there are differences between prognostics and PSA techniques. Table I summarizes these differences in four areas: scope of application, inputs, supported maintenance strategies, and outcomes.

TABLE I
PROGNOSTICS VS DEPENDABILITY ANALYSIS AND PSA

| Properties | Prognostics | Dependability Analysis and PSA |
|-----------------------------|---|---|
| <i>Scope of application</i> | Components | Components and systems |
| <i>Inputs</i> | Operational data, component degradation equations | Component failure data, system fault propagation models |
| <i>Maintenance</i> | Condition-based and predictive maintenance based on parameter monitoring and degradation forecast | Predetermined maintenance based on mathematical planning of maintenance schedules |
| <i>Outcomes</i> | Prolonged useful life, increased reliability and availability of components | Global prediction and management of reliability and safety of the system |

There are several asset-specific prognostics applications reported in the literature. For instance, transformer prognostics has the potential to improve maintenance planning and potentially extend the useful life of power transformers [16]. However, the effects of prognostics at the system-level cannot be easily or automatically established and quantified. There are many elements in a power network, so the improvement achieved in the performance of a specific asset via prognostics does not automatically lead to understanding the positive effects in the system context. This latter knowledge is important for gaining assurance about dependability and for designing system-level prognostic and maintenance strategies in a rational, evidenced way that verifiably achieves improved trade-offs between costs and system dependability.

The input data for prognostics comes from sensors and degradation equations that determine the asset-specific degradation behaviour, whereas for system dependability analysis estimates of component failure probability are established from generic databases with population-based component reliability values. From the viewpoint of system dependability engineering, refining the estimates with real-time or near real-time condition monitoring asset-specific information would be beneficial, as it would provide a more realistic quantification of the system failure probability, and accordingly a more accurate dependability assessment [17].

The planning of maintenance supported by prognostics and dependability analysis is different [18], [19]. Dependability approaches support predetermined maintenance strategies in accordance with established intervals of time, but without pre-

vious condition investigation. Prognostics integrates usefully with predictive and condition-based maintenance techniques, extending the useful life of assets by postponing the replacement instant where possible within acceptable thresholds and reducing maintenance costs. However, prognostics focuses on components and ignores the system effects of maintenance.

Finally, in terms of outcomes prognostics results are normally focused on the estimation and improvement of the remaining useful life of a component, whereas dependability analysis uses system models to establish global estimates of dependability attributes such as reliability, availability, maintainability, and safety.

This paper presents work which develops a state-of-the-art dynamic system dependability analysis technique to improve accuracy of prediction via component-level prognostics.

We use the Boolean Driven Markov Process (BDMP) formalism [20] for expressing a system dependability prediction model that can be developed by designers and analysts of the system. BDMP is a strong dynamic dependability technique sometimes considered as a generalization of Dynamic Fault Trees [20] but does not currently support the integration of prognostics concepts. To address this need, we connect this framework to the formalism of Stochastic Activity Networks (SAN) [21]. SAN is supported by a powerful computational tool that can quantitatively analyse models and establish dependability estimates using simulation techniques [22]. In this approach, prognostics results are regularly extracted during the system operation and are used to update component failure probabilities in the system dependability prediction model. Using new observations of a plant, the dependability estimates for the future trajectory of the system can be updated. This approach provides improved prediction of dependability. The approach has been applied to a case study from the nuclear power industry.

The main contribution of this paper is the integration of a state-of-the-art dynamic dependability analysis technique with prognostics. The second contribution of the paper is the technical model transformation algorithm from BDMP models into SAN models which enables computational operationalization of the proposed approach, but can also be useful in a broader context.

The rest of this article is organized as follows. Section II presents the approach for prognostics-updated dynamic dependability analysis. Section III includes a case study from the power generation industry. Section IV discusses this work in the context of other relevant work. Finally, Section VI draws conclusions and identifies future research challenges.

II. AN APPROACH TO DYNAMIC DEPENDABILITY ANALYSIS ENCOMPASSING PROGNOSTICS

Fig. 1 sketches the proposed prognostics-updated dynamic dependability analysis approach. The *system design* is the starting point of the dependability analysis process. It defines the functional operation of the system specifying how assets are arranged to perform the system function. Two parallel sequences of activities on system dependability modelling and prognostics follow the system design and at some point join to complete the approach.

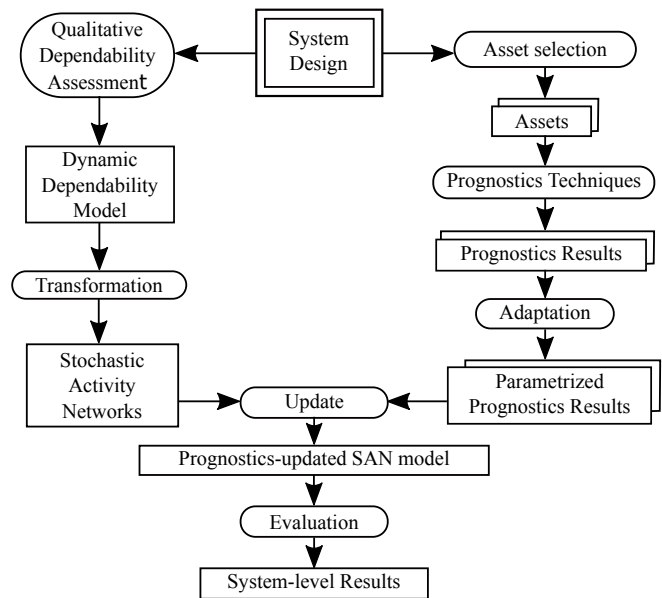


Fig. 1. Prognostics-updated dynamic dependability analysis approach.

The first step of the prognostic sequence is *asset selection*. For the different *assets* that constitute the system design, the designer may have different degradation specifications. To specify a prognostics model, degradation equations or run-to-failure data are compulsory [13]. Therefore, the *asset selection* activity for prognostics evaluation is driven by the availability of data or equations. According to the available resources different *prognostics techniques* can be considered to design a prognostics model for each asset.

At the system level, in order to perform the dynamic dependability analysis, the first step is the *qualitative dependability assessment*. In dynamic dependability analysis we are interested in finding the minimal (temporal) combination of failures of assets that causes the system failure, i.e., minimal cut-sequence sets [23]. A number of techniques have been proposed for calculating these sets, so in this paper we will focus on the prognostic-updated quantification of already established minimal cut-sequence sets.

Minimal cut-sequence sets are represented with a *dynamic dependability model* defining the system failure behaviour as a function of assets failure occurrences linked with temporal operators. In the dynamic dependability model the failure specification of each asset is initialized with average reliability values taken from reliability databases. BDMP is the formalism used in this paper for the representation and evaluation of the dependability of complex dynamic systems.

Moving back to the prognostic sequence, *prognostics results* depend on the nature of the prognostics estimation technique. Different approaches provide alternative representations of the remaining useful life which may include uncertainty representation mechanisms. In order to connect asset-specific results of any prognostics technique with system-level results an *adaptation* activity is essential. This activity *parametrizes prognostics results* so that they can be used directly to update dynamic dependability evaluation models.

To evaluate prognostics-updated system-level failure prob-

ability, the *transformation* from the dynamic dependability evaluation model into *Stochastic Activity Networks (SAN)* model is performed. SAN has a generic and flexible semantic framework that allows the encoding of a range of different dynamic dependability and performance evaluation formalisms [24]-[26]. Accordingly, we can define transformation rules for the transformation of these formalisms into SAN generalizing the application of the approach. This way, we avoid restricting our approach to the assumptions of different dynamic dependability evaluation techniques.

Taking the parametrized prognostics results, the dynamic dependability evaluation model in SAN is *updated* through advanced simulation practices. The updated SAN model includes mechanisms to resample dynamically the probability distribution parameters during the simulation time, enabling the integration of parametrized prognostics results at runtime.

Finally the system-level *evaluation* is performed to quantify the combination of assets that lead the system to failure. The system-level results are specified with the Cumulative Distribution Function (CDF) of the system failure probability. Using basic reliability theory, it is possible to extract the Probability Density Function (PDF) of the system, from which system reliability, availability, and where appropriate safety can be estimated.

A. Prognostics Techniques and Adaptation of Prognostics Results

The main output of prognostics techniques is the Remaining Useful Life (RUL), i.e., prediction of the time to failure of an asset which is deployed in some specific conditions at a specific time instant. Prognostics techniques require a prediction model of the asset degradation process and a failure threshold to determine the remaining time to failure of the asset from the current prediction instant. Prognostics techniques can be classified into data-driven and model-based approaches based on the available engineering resources, i.e. run-to-failure data and physics-of-failure equations respectively. Hybrid approaches result from the combination of data-driven and model-based prognostics techniques. See [12], [13] for an overview of prognostics techniques.

Depending on the nature of the prognostics technique the RUL is estimated in different formats. Depending on the format, different transformation steps would be needed to integrate prognostics results into the dynamic dependability analysis model. A review of formats and examples of techniques for the RUL specification are given in [27]; they include deterministic RUL value (e.g. calculated employing neural networks [12]), deterministic RUL value with confidence intervals (e.g. estimated with autoregressive-moving-average models [12]), or PDF of the RUL (e.g. derived using particle filters [16]).

When predicting the future behaviour of the system via prognostics, there are different sources of uncertainty that influence the RUL prediction. Therefore, uncertainty has an important role when estimating the system RUL. Accordingly, the PDF of the RUL is preferable to the fixed constant RUL estimation because the PDF includes information about

the possible variability of the system and the deterministic RUL specifies a single point estimation [28]. Parametrized distributions are necessary to model, integrate, and propagate prognostics results into dynamic dependability models.

The deterministic RUL estimation is used together with the exponential distribution assuming that the constant failure rate is the inverse of the RUL ($\lambda \approx 1/RUL$) [29]. In this case, it is assumed that a component with an estimated constant value of RUL, fails in time following the exponential distribution with a constant failure rate of $\lambda \approx 1/RUL$. When the deterministic RUL is obtained with confidence intervals, the same adaptation is applied, but this time calculating the minimum and maximum failure rates that correspond to the confidence intervals.

When the prognostics technique provides the PDF of the RUL, it can be adapted with a generic probability distribution to specify the failure behaviour of an asset. The Weibull PDF is considered general enough to integrate different distributions (e.g., exponential, Rayleigh) [30]. However, if the RUL is estimated from physics-of-failure equations (i.e., model-based prognostics approaches), the PDF should be chosen in agreement with the asset-specific prognostics degradation equation.

When adapting the RUL with the failure PDF, the variance of the PDF specifies the degradation behaviour of the asset. For instance, if the variance of the RUL is narrow (which is common for accurate RUL predictions) its corresponding CDF models an asset that degrades almost instantaneously (i.e., the time from failure-free operation to the total-failure operation is determined by this variance) and vice versa.

For a real-time risk monitor reaction times are crucial to take timely remedial actions. As the system assets deteriorate, early indicators are needed to prevent the occurrence of system failure events. If the PDF of the RUL has a narrow variance and this PDF is used as the failure specification of the asset, this specification may prevent the designer from establishing a safety failure threshold and acting accordingly.

Therefore, depending on the available RUL specifications design decisions need to be adopted between the direct transformation of the RUL via Weibull regression [31] and asset-specific physics-of-failure degradation equation according to the variance of the PDF of the RUL estimation.

B. Dynamic Dependability Models

Once the component failure data has been updated with prognostics, system-level dependability analysis can follow. The BDMP is chosen as a dynamic dependability model for this purpose. BDMP is a powerful development in the state-of-the-art which can be seen as a generalization of various types of recent proposed notations for representing Dynamic Fault Trees and therefore deemed appropriate for the purposes of this work.

1) *Preliminaries on BDMP*: A BDMP model [32] is defined by a 4-tuple $\langle L, G, T, TE \rangle$ where:

- $L = \{L_i\}$ is the set of leaves which can be of two types: F leaves (failure can occur only in active mode - L_F) and SF leaves (failure can occur only in active and standby mode - L_{SF}): $L = L_{SF} \cup L_F$.

- $G = \{G_i\}$ is the set of fault tree gates which can be of two types: OR and AND: $G = G_{OR} \cup G_{AND}$. Each gate G_i has at least two inputs which are defined by $Inputs(G_i) \subseteq \mathbb{N}$. The set of nodes $N = L \cup G$ in a BDMP model are defined as the union of leaves and gates.
- $T = \{T_i\}$ is the set of triggers of a BDMP model. A trigger T_i has an origin $Orig(T_i)$ and a destination $Dest(T_i)$. Both these elements are nodes from N . A trigger T_i is defined by a 2-tuple $(Orig(T_i), Dest(T_i)) \in \mathbb{N}^2$.
- TE is the top-event, i.e., failure of the modelled system.

Dynamic dependencies in BDMP come from the trigger mechanism which combines fault trees and Markov models with flexible mathematical properties [20]. The trigger signal is activated by a Boolean function determined by the failure occurrence of its origin elements (which can be comprised of a leaf or more generally origin elements can be subsystems including interconnected leaves, gates and triggers). The activation of the trigger affects the destination leaves by changing the Markovian process associated with each leaf.

Namely, each leaf L_i of the BDMP tree is represented with a Markov process P_i called a triggered Markov process [20]. P_i is comprised of the following set of elements: $\{Z_0^i(t), Z_1^i(t), f_{0 \rightarrow 1}^i, f_{1 \rightarrow 0}^i\}$.

$\{Z_0^i(t), Z_1^i(t)\}$ are two homogeneous Markov processes. For $k \in \{0, 1\}$, the state-space of $\{Z_0^i(t), Z_1^i(t)\}$ is A_k^i . For each A_k^i ($i \in L, k \in \{0, 1\}$) we will need to refer to a part F_k^i of the state space A_k^i . Generally, F_k^i will correspond to the failure states of the component or subsystem modelled by the process P_i .

$f_{0 \rightarrow 1}^i$ and $f_{1 \rightarrow 0}^i$ are two probability transfer functions defined as follows:

- for any $x \in A_0^i$, $f_{0 \rightarrow 1}^i(x)$ is a probability distribution on A_1^i , such that if $x \in F_0^i$, then $\Pr(f_{0 \rightarrow 1}^i(x) \in F_1^i) = 1$
- for any $x \in A_1^i$, $f_{1 \rightarrow 0}^i(x)$ is a probability distribution on A_0^i , such that if $x \in F_1^i$, then $\Pr(f_{1 \rightarrow 0}^i(x) \in F_0^i) = 1$

The triggered Markov process switches instantaneously from one of its modes to the other one, via the relevant transfer function, according to the state of some externally defined Boolean variable, i.e. process selectors or triggers.

Fig. 2 (a) shows the Markov processes for an SF leaf and an F leaf. Informally we can denote process 1 as a required mode and process 0 as a non-required mode. In the required mode both leaves transit from the working to the failed state with the corresponding failure rate λ (process 1 in Fig. 2 (a)). In the non-required mode, the SF leaf can also transit from the standby state to the failed state with λ_s . This is not possible in the case of the F leaf, as it does not have a standby state in the non-required mode and only the transition from failed state to working state is allowed in this mode (process 0 in Fig. 2 (a)). Both leaves pass from the failed state to working (process 1) or standby states (process 0) according to the repair rate (μ).

Fig. 2 (b) shows an example of a BDMP model with an actively operating leaf A (initially with process 1 in Fig. 2(a)) whose failure affects leaf B (F leaf process 0 in Fig. 2(a)) and leaf C (SF leaf process 0 in Fig. 2(a)) via the trigger mechanism (graphically depicted with a dashed arrow). Until the failure occurrence of leaf A, leaf B cannot fail and leaf C

is not operating actively. Once leaf A fails, leaf B and leaf C change their operation to the required mode (i.e., process 1 in Fig. 2 (a)) and they operate actively with their corresponding failure and repair rates. If leaf A is repaired before the failure of leaf B or leaf C, both leaf B and leaf C return back to operate according to the process 0. The top-event occurs when leaf A is in a failed state and simultaneously leaf B or leaf C fails. The model is formally defined as: $L = \{L_A, L_B, L_C\}$; $G = \{G_{OR}, G_{AND}\}$; $T = \{T_1\}$; where $T_1 = \{L_A, G_{OR}\}$.

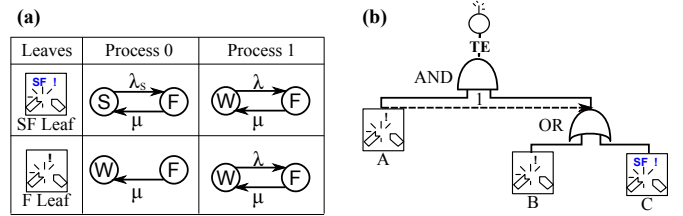


Fig. 2. (a) BDMP leaves and associated Markov processes; (b) BDMP example.

The numerical solution of BDMP models focus on the quantification of the underlying overall Markovian model. Owing to the mathematical properties of BDMP, the state-space is decreased by trimming irrelevant events and reducing the combinatorial explosion [20]. Using the KB3 tool the BDMP model is automatically transformed into the Figaro language which can be solved analytically or via Monte Carlo simulation depending on the properties of the BDMP model [33].

BDMP is a strong dynamic dependability analysis formalism with well-defined mathematical properties [20]. However, to the best of authors' knowledge there is no possibility to integrate directly prognostics predictions into BDMP. This limitation comes from the decision of designing a high-level dynamic dependability analysis formalism. BDMP encodes the dynamic logic with user-friendly modelling constructs (triggers and leaves) simplifying the design of dynamic dependability models.

C. Transformation into Stochastic Activity Networks

In order to integrate runtime information of prognostics models into the dynamic dependability evaluation model, it is necessary to regularly update component failure predictions according to prognostics. The Stochastic Activity Networks formalism provides a sufficient modelling and analysis mechanism to integrate prognostics results into the dynamic dependability evaluation model intuitively [21].

1) *Preliminaries on Stochastic Activity Networks:* Stochastic Activity Networks (SAN) was first introduced in the mid-1980s [34] and it has been used for many different applications. For the sake of readability and simplicity we will introduce the main SAN modelling constructs informally in this subsection. For a formal definition of the SAN main constructs refer to the Appendix.

SAN extends stochastic Petri Nets generalizing the stochastic relationships and adding mechanisms to construct hierarchical models. Fig. 3 shows SAN modelling primitives [21].









| | | | |
|---|---|---|--|
| Standard Place  | Extended Place  | Input Gate  | Output Gate  |
| Instantaneous Activity  | Timed Activity  | Join  | Atomic/ Composed model  |

Fig. 3. Notation of SAN elements.

Places represent the state of the modelled system. Each place contains a certain number of tokens defining the marking of the place. A *standard* place contains an integer number of tokens, whereas *extended* places contain data types other than integers (e.g., floats, array). We will denote the marking function of the place x as $m(x)$, e.g., $m(x)=1$ means that the place x has a marking equal to 1.

There are two types of *activities*: *instantaneous* which complete in negligible amount of time, and *timed* whose duration has an effect on the system performance and their completion time can be a constant or a random value. When it is a random value, it is ruled by a probability distribution function defining the time to fire the activity.

Activities fire based on the conditions defined over the marking of the net and their effect is to modify the marking of the places. The completion of an activity of any kind is enabled by a particular marking of a set of places. The presence of at least one token in each input place enables the firing of the activity removing the token from its input place(s) and placing them in the output place(s).

Each activity has a *reactivation function* that defines when the activity is aborted and a new activity time is immediately obtained from the activity time distribution. The reactivation function provides a mechanism for restarting activities that have been activated, either with the same or a different distribution. To this end it is necessary that both the reactivation predicate holds for the new marking and for the marking in which the activity was originally activated; and the activity remains enabled (see Figure 8 for an example).

Another way to enable a certain activity consists of *input gates* and *output gates*. Input and output gates make the SAN formalism general and powerful enough to model complex real situations. They determine the marking of the net based on user-defined C++ rules.

Input Gates (IG) control the enabling of activities and define the marking changes that will occur when an activity completes. A set of places is connected to the input gate and the input gate is connected to an activity. A Boolean condition (or guard) enables the activity connected to the gate and a function determines the effect of the activity completion on the marking of the places connected to the gate. *Output Gates (OG)* specify the effect of activity completion on the marking of the places connected to the output gate. An output function defines the marking changes that occur when the activity completes.

The performance measurements are carried out through *reward functions* defined over the designed model. Reward functions are evaluated as the expected value of the reward

function and they are defined based on:

- the marking of the net (*state reward function*), e.g. quantification of the probability for being in a specific place;
- completion of activities (*impulse reward function*), e.g. count the number of times an activity triggers within a time interval.

In order to alleviate substantially the state explosion problem SAN makes use of reduced base models [35]. This concept enables the implementation of join operators and hierarchical modelling of complex systems.

Fig. 4 shows a simple repairable system example (i.e. process 1 in Fig. 2 (a)). In this case the SAN places are initialized to working state $\langle m(W), m(F) \rangle = \langle 1, 0 \rangle$. The token will move from W to the F place according to the CDF determined by the `fault` timed activity. The time to failure will be calculated with the parameters of the `fault` activity and after the time to failure has elapsed the system will move to the failed state $\langle 0, 1 \rangle$. After moving to the failure state the time to repair will be calculated from the `repair` timed distribution and the token will move from F to the W place after the calculated time to repair has elapsed.

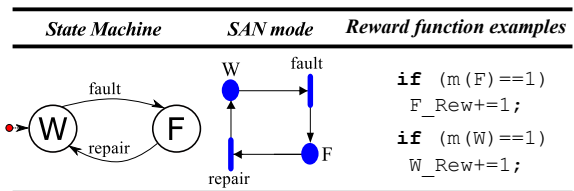


Fig. 4. Repairable system example in SAN.

In this paper we focus on Monte Carlo simulations for the quantification of different probabilities. If we want to evaluate the unavailability or availability we can use the reward functions indicated in Fig. 4 with `F_Rew` and `W_Rew` reward variables respectively. These statements are evaluated for a large number of Monte Carlo trials and the expected value of these random variables evaluated at different time instants will give the unavailability and availability indicators. Note that the required number of iterations will depend on the required confidence level for the reward variables.

The inverse transform sampling method [36] extracts the stochastic occurrence times of timed activities using Monte Carlo simulations. Let *CDF* be a cumulative distribution function, r be a random variable drawn from the uniform distribution $r \sim U([0, 1])$, and *TTF* the time to fire the activity. Then, the inverse sampling method applies the relation $F^{-1}(r) = TTF$ to draw the time to fire according to the CDF.

The SAN models which include the specified SAN elements are modelled in a SAN *atomic model* (see Fig. 5 *Reusable Block* column). The *join operator* links through a compositional tree structure different SAN models in a unique *composed model*. As shown in Fig. 6 places of different atomic models are joined to represent interactions between different SAN models. It is possible to link atomic models, composed models, or combinations thereof. In the tree structure, the composed and atomic SAN models are linked through join

operators using shared places between the composed and atomic SAN models. Thus, the analyst can focus on specific characteristics of the system behaviour through fit-for-purpose atomic and composed models and later join independently validated models to obtain a more complex composed system model.

The SAN model in Fig. 4 can be reused so as to link the places with other components. For instance, if we want to initiate another (timed or immediate) activity in another component when the failure of the model in Fig. 4 occurs, we can join the F place with the destination SAN model through the SAN join mechanism (see Fig. 6 for an example). For the composed system the quantification of system probabilities is exactly the same as for the simple system in Fig. 4.

2) *Transformation from BDMP into SAN*: Fig. 5 shows the BDMP to SAN dictionary including: state machine models, SAN models with input and output gate specifications, and the reusable building block which is later on used to synthesize BDMP models from SAN models. For clarity we do not have named instantaneous activities in the SAN models.

The only difference between F and SF leaves is the inclusion of the transition from Standby (S) to the Failed (F) state caused by the $\text{fault}_{\text{stdby}}$ timed activity (see dashed lines in Fig. 5). Initially leaves can be either in Working (W) or standby state. Accordingly, the markings of SAN places are initialized to $\langle m(W), m(S), m(F), m(\text{act}), m(\text{deact}) \rangle$: working $\langle 1, 0, 0, 0, 0 \rangle$; or standby $\langle 0, 1, 0, 0, 0 \rangle$ states. The IG and OG specification determines the marking changes that are done when the activity linked to the IG (or OG) fires. That is, the marking will be updated according to the Check Deact output gate when the repair activity fires. Besides, note that if $m(W)=1$ and $m(\text{deact})=1$ then $m(S)=1$ instantaneously.

If the F leaf is not the destination element of a trigger, then the SAN model reduces to the model shown in Fig. 4, i.e. always in required mode or process 1 in Fig. 2 (a). If the F leaf is the destination element of a trigger, as shown in Fig. 5 we use a standby state to model the situation in which the leaf is operating in non-required mode or process 0 in Fig. 2 (a).

The detailed behaviour of the Boolean (AND, OR) gates is described by the state machine, where a doubled circle indicates the failure state, and F_x and R_x indicate the failure and repair events of the component x respectively. For both gates the initial state is assumed to be healthy $\langle m(A), m(B), m(Y) \rangle = \langle 0, 0, 0 \rangle$. These gates can be easily generalized adding more places to the SAN model and changing the IG specification accordingly.

Finally, a simplified version of the BDMP's trigger construct is modelled. It is assumed that the trigger has an origin event F_T which activates and deactivates the trigger. In SAN an extra place is needed to control the status of the trigger, i.e., s_{act} , s_{deact} places for activation and deactivation respectively. Note that the act and deact places cannot be used because these are shared with the destination element that will be triggered.

Any BDMP model can be synthesized into an equivalent SAN model using the building blocks in Fig. 5 and based on the shared places and hierarchical modelling concepts of SAN. Fig. 6 shows an example of the composition of SAN models.

BDMP leaves $Tr1$ and $Tr2$ are modelled using the F leaf SAN building block. The trigger that goes from $Tr1$ to $Tr2$ is modelled using the $Trig$ construct, sharing act and deact places of the $Trig$ model (activated by the F place of the $Tr1$) with the act and deact places of $Tr2$ model. That is, when the failure of $Tr1$ occurs this activates $Tr2$ and when $Tr1$ is repaired the trigger deactivates $Tr2$.

Then, the reusable composed SAN model $Trig_Tr$ is connected with the AND gate to create the TE . That is, AND will be true (i.e. $m(Y)=1$ in Fig. 5 second row) when both $Tr1$ and $Tr2$ are in failed state simultaneously, i.e. $m(F)=1$ in Fig. 6. Generalizing this simple example, it is possible to parse systematically a BDMP model to find its equivalent SAN model.

As the BDMP model is acyclic, the failure logic is calculated at each level starting from the bottom leaf-level up to the top-event level. Algorithm 1 defines the synthesis process assuming that all the gates are ordered hierarchically bottom-up (for clarity we have not included shared states in the algorithm, but these are shared as depicted in Fig. 6) using a set of functions subsequently defined.

Algorithm 1 uses the following set of functions:

- $Out_1 = \text{PropagateTrigger}(In_1)$; in the BDMP model In_1 , for each trigger $T_i \in T$ with a gate as a destination element, it propagates the effect of the trigger down to the leaf-level and saves the new BDMP model in Out_1 . After propagating the trigger, the destination elements of every trigger will be leaves, e.g., in the BDMP model of Fig. 2 (b), the trigger that goes from leaf A to the OR gate is replaced with two triggers going from leaf A to leaf B and leaf C .
- $Out_1 = \text{Transform}(In_1, In_2)$; transforms each BDMP element in model In_1 ($L_i \in L$, $G_i \in G$, $T_i \in T$) into a vector Out_1 with SAN equivalent elements using the given transformation dictionary In_2 (cf. Fig. 5). With this function we get a set of independent BDMP models in SAN. Hierarchical information of the BDMP model is stored in the input and output of gates and triggers.
- $Out_1 = \text{Leaves}(In_1)$; stores in Out_1 all the leaves ($L_i \in L$) that are in the vector In_1 as SAN elements.
- $Out_1 = \text{Inputs}(In_1, In_2)$; stores in Out_1 the input nodes of the gate In_1 located in the vector In_2 as SAN elements.
- $Out_1 = \text{Join}(In_1, In_2, \dots, In_N)$; implements the join operation between the inputs $\{In_1, In_2, \dots, In_N\}$, and creates the reusable composed SAN model Out_1 (cf. Fig. 6).
- $\text{SetInitState}(In_1, \text{init})$; sets the initial state to init to the leaf In_1 .
- $Out_1 = \text{Orig}(In_1)$; finds the origin node of the trigger In_1 and stores in Out_1 .
- $Out_1 = \text{Dest}(In_1)$; finds the destination node of the trigger In_1 and stores in Out_1 .
- $Out_1 = \text{getSubtree}(In_1, In_2)$; extracts the subtree Out_1 from the given tree In_2 , starting from the given node In_1 downwards until the leaf-level.

Algorithm 1 proceeds as follows in the BDMP model shown in Fig. 6:

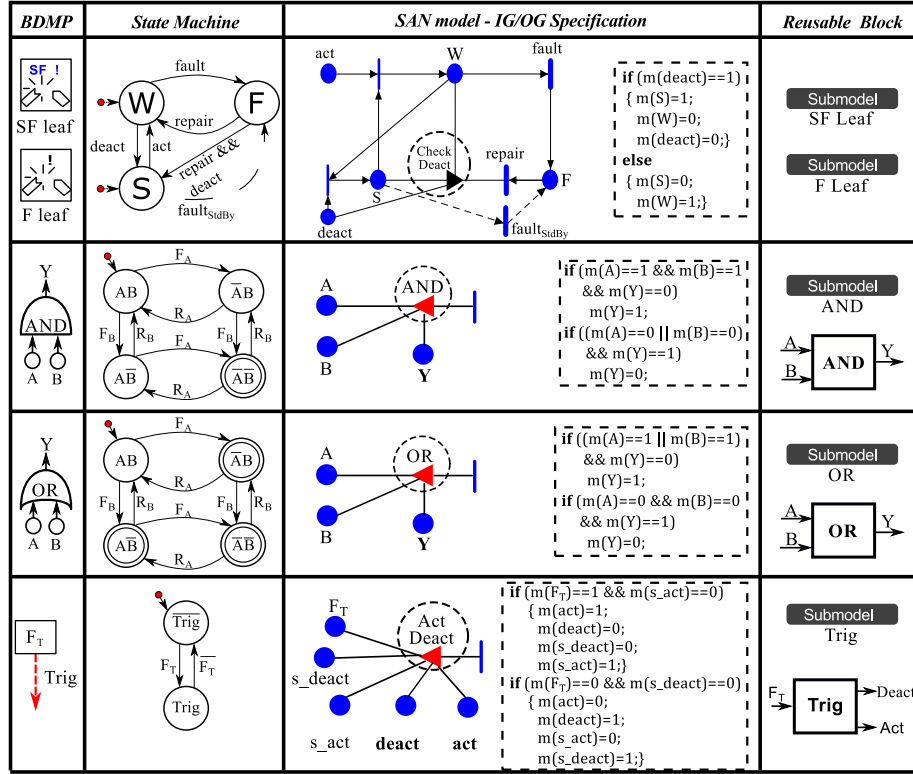


Fig. 5. BDMP to SAN dictionary: building blocks.

- Line 2: there is no need to further propagate the trigger effect.
- Line 3: the variable vec_BDMP_SAN is set according to the dictionary in Fig. 5; $vec_BDMP_SAN = [Tr1, Tr2, Trig, AND]$.
- Line 4: the variable $tree_SAN$ is directly assigned: $tree_SAN = \{[Tr1], [Tr2]\}$.
- Lines 5–9: the BDMP model consists of a single gate. Thus, the SAN tree variables are updated as: $input_nodes_1 = [Tr1, Tr2]$; $subtree_1 = [join, AND, Tr1, Tr2]$; and $tree_SAN = subtree_1$.
- Lines 10–14: the initial state of $Tr1$ is set to *Active* and $Tr2$ is set to *Standby*.
- Lines 15–17: the unique trigger in the BDMP model is created linking origin and destination variables: $subtree_o = [Tr1]$; $subtree_d = [Tr2]$; $trigger = [join, Trig, Tr1, Tr2]$.
- Line 18: the SAN trigger variable is added to the previously created SAN tree of the BDMP model; $tree_SAN = \{[join, Trig, Tr1, Tr2], [join, AND, Tr1, Tr2]\}$.

The synthesis Algorithm 1 along with the BDMP to SAN dictionary in Fig. 5 presents the correspondence between BDMP and SAN models. This transformation has benefits for both formalisms. SAN is a generic and powerful formalism that can be adapted to model different dynamic dependability models. However, this generality comes with modelling costs. The proposed transformation process enables the synthesis of SAN models that implement BDMP models, which alleviates the modelling costs and enables the analysis of complex systems in a user-friendly manner. BDMP models can make

use of SAN constructs to extend the formalism and analyse situations which cannot be handled with BDMP, e.g., dynamically updated dependability estimates.

D. Prognostics-updated Failure Specification of Assets

BDMP leaf nodes model the assets under study and these are defined by a triggered Markov process (see Subsection II-B1). Depending on the type of the leaf and if the leaf node is a destination element of a trigger mechanism or not, the state-space of the leaf is different.

In the most complex case (see Fig. 5, first row) the BDMP leaf includes working, failed and standby places, *fault*, *repair* and *fault_{standby}* timed activities, and also *act* and *deact* instantaneous activities. In the simplest case, i.e. a F leaf node without trigger influence, the leaf model in Fig. 5 is reduced to the model shown in Fig. 4.

In both cases, the transition from working (W) to failed (F) state is triggered by a *fault* timed activity. The *repair* timed activity can transit from failed (F) to standby (S) or working (W) state depending on if the leaf is the destination element of a trigger mechanism or not respectively. Both *fault* and *repair* timed activities are fully specified by the CDF of the failure and repair distribution.

Prognostics studies are performed regularly throughout the lifetime of the asset, and updated with online information. The link between prognostics and dynamic dependability assessment addressed in this paper focuses on updating the dependability model with asset-specific characteristics obtained from prognostics analyses. That is, the CDF parameters of the

Algorithm 1 Synthesis of SAN models from a BDMP model

```

1: function SYNTHESIZE_SAN(BDMP_model, dictionary)
2:   BDMP2=PropagateTrigger(BDMP_model);
3:   vec_BDMP_SAN=Transform(BDMP2, dictionary);
4:   tree_SAN=Leaves(vec_BDMP_SAN);
5:   for each  $G_i \in G$  do
6:     input_nodesi=Inputs( $G_i$ , tree_SAN);
7:     subtreei=Join(vec_BDMP_SAN( $G_i$ ), input_nodesi);
8:     tree_SAN=tree_SAN  $\cup$  subtreei;
9:   end for
10:  for each  $L_i \in L$  do
11:    SetInitState(tree_SAN( $L_i$ ), 'Active');
12:  end for
13:  for each  $T_i \in T$  do
14:    SetInitState(tree_SAN(Dest( $T_i$ )), 'StandBy');
15:    subtreeo=getSubtree(Orig( $T_i$ ), tree_SAN);
16:    subtreed=getSubtree(Dest( $T_i$ ), tree_SAN);
17:    trigger=Join(Trig, subtreeo, subtreed);
18:    tree_SAN=tree_SAN  $\cup$  trigger;
19:  end for
20:  return tree_SAN
21: end function

```

▷ propagate triggers' destination to the leaf level
 ▷ transform BDMP elements into SAN
 ▷ store all the leaves in the vector
 ▷ parse all the gates bottom-up
 ▷ take all input SAN elements of the gate G_i
 ▷ link gate G_i with inputs in SAN
 ▷ nest in the tree the generated subtree in SAN
 ▷ at this point the BDMP model is created without triggers and initial states
 ▷ set the initial state of every leaf initially to active
 ▷ now add the trigger effects and standby states
 ▷ set to standby destination leaves
 ▷ get SAN subtree of the origin of the trigger
 ▷ get trigger destinations, i.e., SAN leaves
 ▷ join via trigger origin and destination
 ▷ nest in the tree

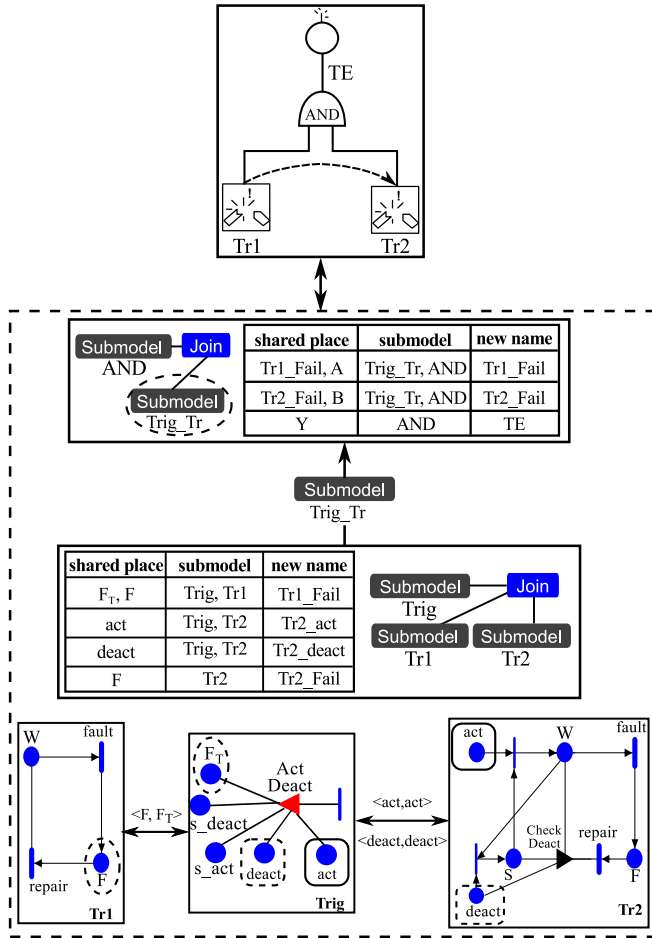


Fig. 6. BDMP synthesis example.

failure distribution can be updated with prognostics prediction results.

To this end, the following steps are implemented in SAN and repeated until reaching the mission time T_m (cf. Fig. 7):

- pause the simulation time at the prognostics prediction time (T_p);
- read and parametrize prognostics results;
- update the failure distribution parameters, resampling the failure CDF according to the prognostics estimations.

After each update a new prognostics prediction time is obtained from T_p' .

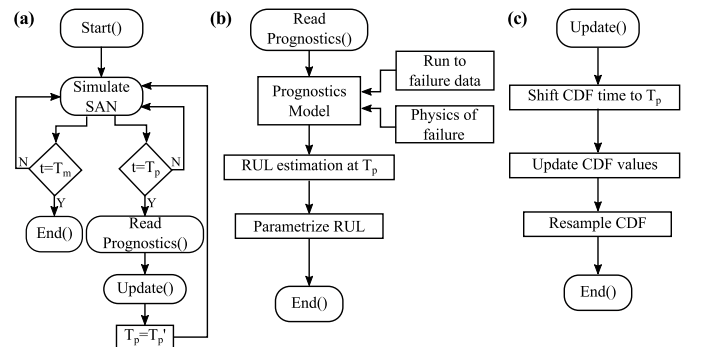


Fig. 7. Update process: (a) SAN engine; (b) prognostics reading; (c) resample distribution.

Conditional distributions are needed to integrate the results with the corresponding time shift because the updating process is performed at different time instants. In this case, we have updated the results according to the conditional Weibull distribution given by

$$F(t|T_{el}) = 1 - e^{-\left[\left(\frac{T_{el}+t}{\eta}\right)^\beta - \left(\frac{T_{el}}{\eta}\right)^\beta\right]} \quad (1)$$

where β and η are the shape and scale parameters of the Weibull distribution respectively, and T_{el} is the elapsed time of operation up to the start of the new mission at t [30].

The updating process is implemented in SAN with some advanced simulation practices. Based on the simulation time, prediction times, and prognostics results; reactivation predicates are determined so that the leaf takes new firing times (i.e., time to fire of the `fail` activity in Fig. 5) ruled by the updated distribution function at the prognostics-specified prediction times.

For instance, if we assume that a component degrades according to the exponential distribution, the failure distribution parameters of the `fail` activity are updated dynamically

$$\lambda(t) = \begin{cases} \lambda_0 & 0 < t < T_{p_i} \\ \lambda_i & t = T_{p_i} \end{cases} \quad (2)$$

where λ_0 is the initial failure rate estimate typically taken from a reliability database, T_{p_i} is the i -th prediction time instant, and λ_i is the failure rate at i -th prediction time instant. The number of prediction time instants depends on the specific prognostics application (and available data) varying from 1 up to P prediction instants $i = \{1, \dots, P\}$.

Fig. 8 shows a reactivation function example for a non-repairable asset with `OK` and `KO` places initialized to $\langle m(\text{OK})=1, m(\text{KO})=0 \rangle$ and a `fail` timed activity defined with an exponential distribution, with an initial failure rate λ_0 . Any timed activity in SAN has the possibility to reactivate and resample its distribution function, obtaining a new time to fire [21]. In Fig. 8 this happens when the reactivation predicate is true ($m(\text{Reactivate})=1$) and the marking conditions to enable the activity are satisfied ($m(\text{OK})=1$). The asset block is connected with the `monitor` block through `Lambda` and `Reactivate` places. The `monitor` block is executed deterministically every Δt time instant. The code embedded in the `Control` output gate evaluates if the simulation time (`time`) matches with prediction times (`PredTimes`) and accordingly (i) enables the resampling of the `Fail` activity through the `Reactivate` place and (ii) updates the marking of the failure rate through the `Lambda` place. `PredTimes` and `PredValues` store different prediction time instants and values in numeric vectors, respectively.

The time to fire (TTF) of the activity changes throughout the mission time: (a) at t_0 the initial TTF_0 is drawn from the exponential distribution with rate λ_0 ; (b) at t_{r_1} a new TTF_1 is obtained from the exponential distribution with rate λ_1 extending the time to fire until t_3 ; (c) at t_{r_2} a new TTF_2 is obtained from the exponential distribution with rate λ_2 extending the time to fire until t_5 ; (d) the overall TTF of the activity lasts $TTF = t_5 - t_0$. This update process results in updating the failure distribution at the stated reactivation time instants (e.g. see an example in Fig. 18).

Note that the example in Fig. 8 models the resampling process for the exponential distribution. Due to the memoryless property of this distribution, there is no need to consider conditional distributions. However, if other distributions are deemed more appropriate, the corresponding conditional failure distribution parameters need to be updated according to

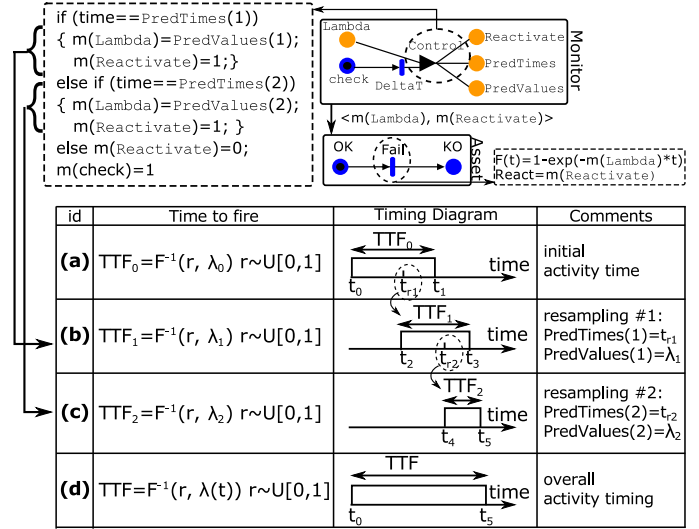


Fig. 8. Distribution resampling example.

prognostics prediction results, e.g. $\beta(t)$ and $\eta(t)$ parameters for the conditional Weibull distribution as defined in (1).

Using the SAN reactivation mechanism along with the prognostics prediction information for each leaf, we can update the underlying triggered Markov process to include new failure distribution parameters when prognostics predictions are performed.

III. CASE STUDY: POWER SUPPLY OF ELECTRIC DISTRIBUTION BOARD

The electric distribution board supplies power to the components which support the cooling and the control functions of a nuclear reactor core. Fig. 9 shows the configuration of the power supply of an electric distribution board [32].

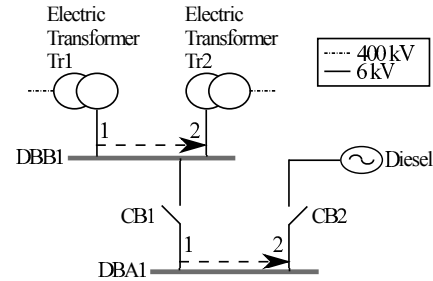


Fig. 9. Power supply of an electric distribution board [32].

A. Dynamic Dependability Evaluation Model

Failure to supply the power results in hazardous consequences for the nuclear power plant. Fig. 10 shows the correspondent BDMP model of the power supply divided into hierarchical levels [32].

Failure of the active transformer $Tr1$ activates the standby transformer $Tr2$. Anytime when the $Tr1$ is repaired, $Tr2$ returns to the standby (or dormant) state. When the transformers ($Tr1$ and $Tr2$), distribution board ($DBBI$), or circuit breaker ($CB1$) fail (see `Fail_PS_DBBI` gate in Fig. 10), the standby diesel

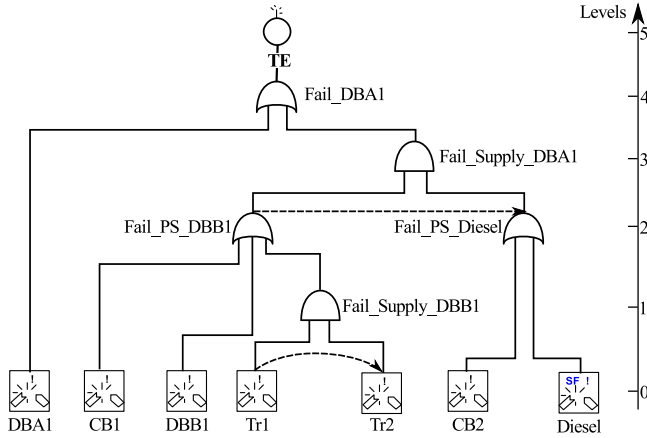


Fig. 10. BDMP model of the power supply in Fig. 9 [32].

supply is activated, comprised of a circuit breaker $CB2$ and a *Diesel* generator. When any of the primary supply elements are repaired (so that $Fail_PS_DBB1=0$); then $CB2$ and *Diesel* return to standby state. Note that the *Diesel* supply can fail in standby mode (SF leaf) while $Tr2$ cannot (F leaf).

Table II displays the failure rate figures used for the quantitative assessment assuming a mean time to repair of one day for each asset.

TABLE II
FAILURE RATE VALUES OF THE ASSETS IN FIG. 9.

| Asset | λ ($year^{-1}$) |
|---|---------------------------|
| Circuit breakers ($CB1$, $CB2$) | $6.2e-1 y^{-1}$ [17] |
| Transformers ($Tr1$, $Tr2$) | $1.2264e-2 y^{-1}$ [17] |
| Diesel | $4.6428 y^{-1}$ [37] |
| Distribution boards ($DBB1$, $DBA1$) | $8.76e-7 y^{-1}$ [38] |

Note that the life expectancy of power transformers is much longer than circuit breakers, diesel generators, and distribution boards with an average lifetime of over fifty years [39], [40].

B. Transformation into SAN

The BDMP model in Fig. 10 is solved using the KB3 workbench through Monte Carlo simulations [33]. We generate the equivalent Stochastic Activity Networks model applying the synthesis Algorithm 1 to the BDMP model in Fig. 10 as follows:

- Line 2: no need to propagate the trigger $Trig1$ from $Tr1$ to $Tr2$. Propagate the trigger effects in the BDMP model from $Fail_PS_DBB1$ to $CB2$ and *Diesel* leaves through dedicated triggers $Trig2$ and $Trig3$, respectively.
- Line 3: generate low-level equivalent SAN atomic models using the dictionary in Fig. 5. $vec_BDMP_SAN = [DBA1, CB1, DBB1, Tr1, Tr2, CB2, Diesel, Trig1, Trig2, Trig3, AND1, AND2, OR1, OR2, OR3]$; where:
 - F leaf: $DBA1, CB1, DBB1, Tr1, Tr2, CB2$;
 - SF leaf: *Diesel*;
 - Trigger: $Trig1, Trig2, Trig3$;
 - AND: $AND1, AND2$;
 - OR: $OR1, OR2, OR3$;

- Line 4: $tree_SAN = \{[DBA1], [CB1], [DBB1], [Tr1], [Tr2], [CB2], [Diesel]\}$.
- Lines 5–9: $tree_SAN = subtree_1$; $subtree_1 = \{join, OR1, DBA1, subtree_2\}$; $subtree_2 = \{join, AND1, subtree_3, subtree_4\}$; $subtree_3 = \{join, OR2, CB1, DBB1, subtree_5\}$; $subtree_4 = \{join, OR3, CB2, Diesel\}$; $subtree_5 = \{join, AND2, Tr1, Tr2\}$.
- Lines 10–14: initial state of $Tr1$ is set to active, initial states of $Tr2$, $CB2$, and *Diesel* are set to standby.
- Lines 15–17: $trigger1 = [join, Trig1, Tr1, Tr2]$; $trigger2 = [join, Trig2, subtree_3, CB2]$; $trigger3 = [join, Trig3, subtree_3, diesel]$.
- Line 18: $tree_SAN = \{[join, Trig1, Tr1, Tr2], [join, Trig2, subtree_3, CB2], [join, Trig3, subtree_3, diesel], subtree_1\}$.

Fig. 11 shows the resultant composed SAN model. For simplicity, in this figure we have omitted the atomic models of trigger and Boolean gates and we have only shown the atomic models of F and SF leaves created in Line 3 of the Algorithm 1 using the dictionary in Fig. 5. Note that both $trigger2$ and $trigger3$ models contain the $Fail_PS_DBB1$ model, which is shared when linking with $Fail_PS_Diesel$ via $AND1$ gate to create the $Fail_Supply_DBA1$ model. Accordingly, the $Fail_PS_DBB1$ model contains the $Fail_Supply_DBB1$ model showed in Fig. 6. For clarity, we have omitted the shared states, but these also need to be created as showed in Fig. 6.

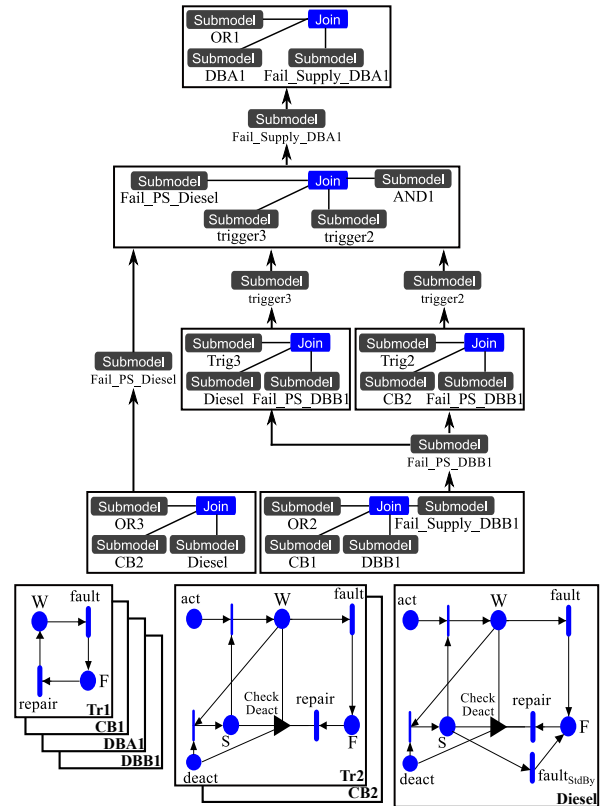


Fig. 11. SAN composed model of the power supply in Fig. 9.

In order to validate the proposed algorithm and models, we have quantified the BDMP model in Fig. 10 through the KB3

tool [33] and the equivalent SAN model in Fig. 11 using the Möbius tool through Monte Carlo simulations [22]. In SAN the system-level unavailability is calculated by evaluating the occurrence of the top-event (i.e., OR1 submodel in Fig. 11) through reward variables (see Subsection II-C1). Fig. 12 shows the system's unavailability obtained using BDMP and the equivalent SAN model.

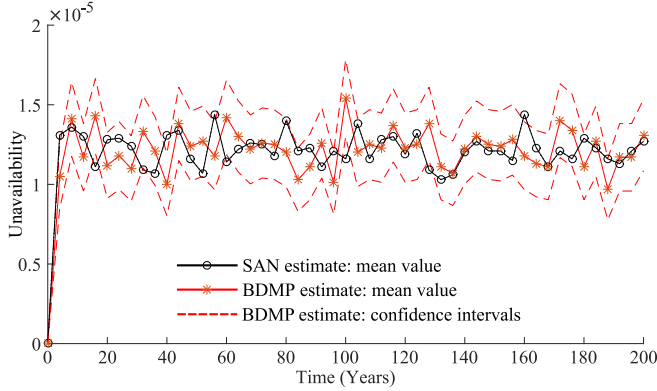


Fig. 12. Unavailability of the model in Fig. 10 solved via BDMP and SAN.

The unavailability of the SAN model falls within the bounds of the BDMP estimation. Therefore, the values obtained with BDMP and SAN are equivalent. Although evaluating a lifetime of 200 years in industrial practice is unrealistic, we have deliberately evaluated the performance of the model for the long term so as to validate the created model. Both models are evaluated with a confidence level of 0.99 and a confidence interval of $1e-6$. Table III displays the mean values shown in Fig. 12.

TABLE III
MEAN VALUES IN FIG. 12

| Time (Years) | SAN ($\times 10^{-5}$) | BDMP ($\times 10^{-5}$) |
|--------------|--------------------------|---------------------------|
| 5 | 1.49 | 1.47 |
| 10 | 1.51 | 1.59 |
| 20 | 1.59 | 1.42 |
| 30 | 1.49 | 1.4 |
| 40 | 1.53 | 1.69 |
| 50 | 1.49 | 1.36 |
| 100 | 1.51 | 1.53 |
| 150 | 1.57 | 1.43 |
| 200 | 1.53 | 1.46 |

With a repair rate of one day, transformer failure has little influence on the system unavailability. However, the true repair rate for this size of transformer can be up to one year [39], [40]. Therefore RUL predictions should be in a similar timescale for timely maintenance action planning.

C. Asset-level Prognostics Models and Parametrization

For this case study, it is assumed that condition monitoring is focused towards the end of expected life of the assets.

Accordingly, 59 years after the installation of the transformers, different data-gathering systems were installed. One year of data was used to diagnose the current health state and predict the remaining useful life of the transformers.

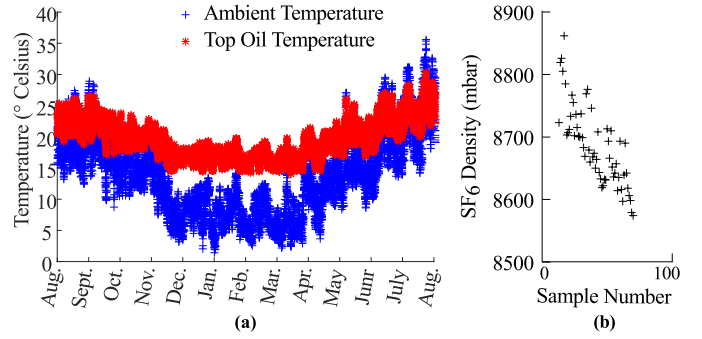


Fig. 13. (a) Transformer top oil and ambient temperature samples. (b) Circuit breaker SF₆ density data.

The monitored variables include the current loading, top-oil temperature and ambient temperature (see Fig. 13 (a)).

In parallel, 60 years after the installation of the power transformers, the previous circuit breakers were replaced with new ones which had monitoring sensors [41]. In this case, one year of SF₆ gas density data was calculated from measured gas pressure and ambient temperature. The density of SF₆ gas is used to interrupt current flow in a circuit breaker [42]. When this density drops to a predefined lockout level (i.e., 8192 mbar) it is no longer able to operate.

The first step was to analyse if a gas leak exists or not. Generally, in the case of a leak, SF₆ gas density decreases linearly [43]. In order find an appropriate prognostics model, we evaluate the monotonicity *mono* of the data as follows [44]:

$$mono = mean\left(\left|\frac{\#positive \frac{d}{dt}}{n} - \frac{\#negative \frac{d}{dt}}{n}\right|\right) \quad (3)$$

where n is the number of data windows in the dataset and t is the time scale.

With a window of $n = 10$ samples, the monotonicity of the available dataset is $mono = 0.81$ which confirms that the degradation trend is linear. Based on the prognostics technique selection approach in [13], we select linear regression as an appropriate technique for the problem under study.

Therefore, the initial SF₆ gas density data was divided into blocks of data points, and a linear regression model was applied to see the degradation over time [43]. This step requires fitting a generic linear function to the data with the form

$$\hat{y} = mx + n \pm sd \quad (4)$$

where \hat{y} is the predicted value of the dependent variable, m is the slope of the line, x is the independent variable, n is the y-intercept, and sd is the standard deviation of the density

$$sd = \sqrt{E[(x - \bar{x})^2]} \quad (5)$$

where x is the density data sample, \bar{x} is the mean value of the dataset and E denotes the expected value.

The linear regression involves finding the values of m and n so that the sum of the squared prediction error *SPE* is minimized

$$SPE = \sum_{i=1}^M (y_i - \hat{y}_i)^2 \quad (6)$$

where M is the dataset length, y_i is the historical value of the dependent variable, and \hat{y}_i is as defined immediately above.

The slope parameter m was used to characterize the SF₆ gas changes over time. If the slope was steeper than a certain threshold, it was considered a true indicator of SF₆ leak [43]. Fig. 13 (b) shows an example true indicator of an SF₆ leak and Fig. 14 shows the RUL estimation based on the linear regression with different data samples.

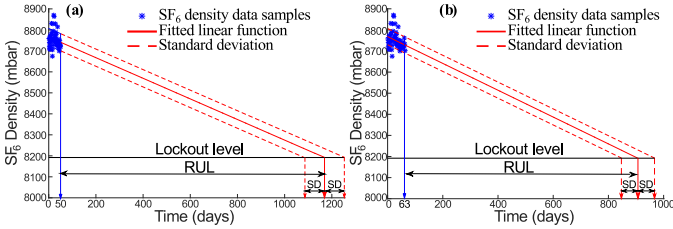


Fig. 14. Circuit breaker prognostics results with predictions at different prediction times. (a) $T_{p1} = 50$ days. (b) $T_{p2} = 63$ days.

Extending the predicted linear function up to the lockout level (cf. Fig. 14), we get the RUL values displayed in Table IV.

TABLE IV
PARAMETERIZATION OF RESULTS IN FIG. 14

| T_p (days) | RUL (days) | SD (days) |
|---------------|------------|-----------|
| $T_{p1} = 50$ | 1118 | 82 |
| $T_{p2} = 63$ | 844 | 60 |

In this specific case, it can be seen that with more available data the asset-degradation slope increases and the variance decreases. The slope increment is caused by an increased SF₆ gas leakage. However note that this is not a generalizable effect and the asset-degradation depends on the asset-specific operation. The reduction of the variance comes from the increased certainty level of the predicted data which reduces with more available data in this case.

Through the application of the linear regression model we obtain single point RUL estimates. Accordingly when updating the failure distributions of the circuit breakers we use the values in Table IV as failure rate parameters of the exponential distribution (see Subsection II-A). Namely, *CB1* and *CB2* are implemented in SAN as BDMP F leaf models (see Fig. 10) using Algorithm 1 and the dictionary in Fig. 5. As shown in Fig. 11, the SAN models of *CB1* and *CB2* are different because *CB1* is not affected by the trigger mechanism.

The prognostics model of the transformer is based on the aging of paper insulation which is the most critical failure for power transformers [40], [45]. With time the paper becomes more brittle, insulation integrity is lost, and the transformer ceases its operation. IEEE C57.91 defines a paper aging acceleration factor [46],

$$F_{AA} = e^{\left[\frac{15000}{383} - \frac{15000}{273 + \Theta_H} \right]} \quad (7)$$

where Θ_H is the hotspot temperature in °C.

The probabilistic degradation of the system was analysed based on the Bayesian particle filtering approach [16]. To this end, it is necessary to rewrite the physics-of-failure degradation equation (7) as a recurrence relation [16],

$$L_t = L_{t-1} - e^{\left[(15000 + \xi_t) \left(\frac{1}{383} - \frac{1}{1 + \Theta_{H_t}} \right) \right]} \quad (8)$$

where t is the time in service in hours, L_t is the RUL at time t , Θ_{H_t} is the hotspot temperature at time t , and ξ_t is the process noise.

The process noise models the variation in lifetime reduction for a given hotspot temperature. This is caused by the small differences in the activation energy required to break cellulose chains. Therefore the process noise must account for the uncertainties surrounding the activation energy process [16]. In this case a normal distribution with $\mathcal{N}(0, 20)$ was used.

The transformer hotspot temperature is not directly measurable and it is inferred from other parameters [46],

$$\Theta_H = \Theta_{to} + (80 - \Delta\Theta_{to/a,R}) \times K^{2m} \quad (9)$$

where Θ_{to} is the measured top oil temperature, $\Delta\Theta_{to/a,R}$ is the difference in temperature between top oil and ambient at rated current, K is the ratio of measured load to rated load, and m is related to the cooling mode of the transformer.

The particle filter model is based on many different simulations (i.e., particles) with different initial conditions and state transition probabilities. At each simulation time step, the new state of the system is evaluated given the previous state and the probability of each particle is weighted using the likelihood function so as to evaluate the true current state of the system. See [16] for more details about the transformer degradation and particle filtering implementation.

Through the repeated application of (8) and (9) with the yearly historical load current and ambient and oil temperature measurements (see Fig. 13 (a)), the Bayesian particle filter model predicts the effect of various possible future conditions on the transformer life.

In particular, we can assess the effect of overload conditions on the transformer RUL estimation. We have evaluated different hypothetical future operation conditions at different prediction times, including overload conditions caused by stochastic external events, which require additional cooling in the power station by increasing the load up to 120%.

- $T_{p1}=3y$: 90% loaded for 3 years.
- $T_{p2}=12y$: 90% loaded for 11 years, 120% for others.
- $T_{p3}=15y$: 90% loaded for 11 years, 120% for others.
- $T_{p4}=20y$: 90% loaded for 14 years, 120% for others.

Note that these scenarios summarize the overall conditions, but overload events can occur at different time instants with different durations, e.g. one year of total overload in T_{p2} can be originated from three overload events occurring at 4, 7, and 10 year time instants which last for four months each.

Fig. 15 shows the PDF of the transformers' RUL estimations at different prediction times (T_p).

According to the analysed overload scenarios, we can confirm that the increased overload condition has a direct

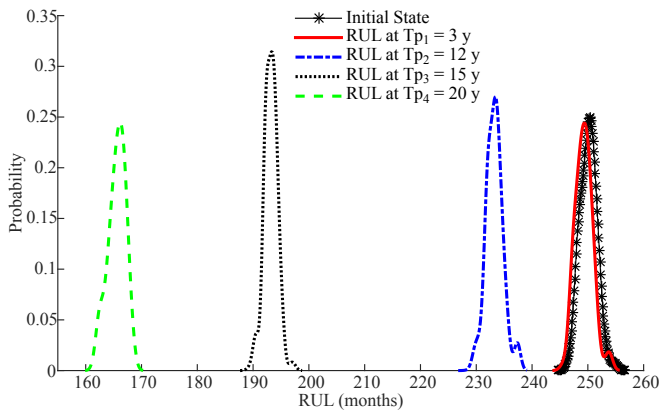


Fig. 15. Transformer RUL values at different prediction times.

impact on the transformer degradation and RUL prediction, in agreement with (8).

Based on the RUL estimations in Fig. 15 we update the unavailability of the transformers in the dynamic dependability model (Fig. 11) and check differences with the model without updates. We have evaluated two different probability distributions for the update of the transformer's failure distribution with prognostics values (Subsection II-A): the exponential distribution with maximum likelihood RUL and variance values and the Weibull distribution with its shape (β) and scale (η) parameters. Table V displays equivalent distribution parameters of PDFs in Fig. 15.

TABLE V
PARAMETERIZATION OF PDFS IN FIG. 15

| T_p (m) | Max. (m) | Dev. (m) | β | η |
|----------------|----------|----------|---------|--------|
| $T_{p1} = 36$ | 249.38 | 3.53 | 249.4 | 165.3 |
| $T_{p2} = 144$ | 233.47 | 3.6 | 233.4 | 170.3 |
| $T_{p3} = 180$ | 193.39 | 3.25 | 193.4 | 164.7 |
| $T_{p4} = 240$ | 166.21 | 3.06 | 166.2 | 110.8 |

Assuming non-repairable events, the cumulative distribution function of the Weibull and exponential distributions are expressed through (1) with $T_{el} = 0$, and with $T_{el} = 0$ and $\eta = 1$, respectively. Fig. 16 and Fig. 17 show the corresponding cumulative distribution functions with the parameters extracted from prognostics estimations.

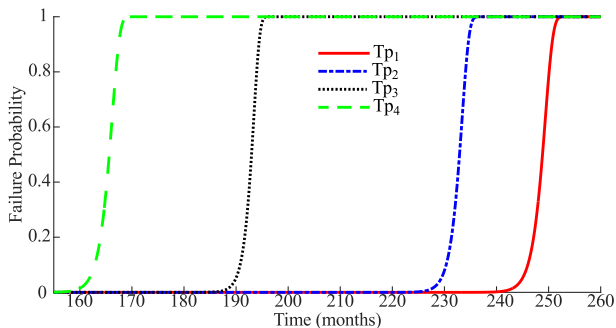


Fig. 16. Non-repairable transformer CDF using Weibull distribution.

In Fig. 16, we can see that the use of the Weibull distribution results in an asset degradation behaviour which changes

rapidly due to the low variance of the RUL estimation.

Given that the transformer's paper degradation process is governed by the exponential law in (8), we have also modelled the corresponding degradation behaviour in Fig. 17 using the maximum likelihood RUL value as the failure rate parameter.

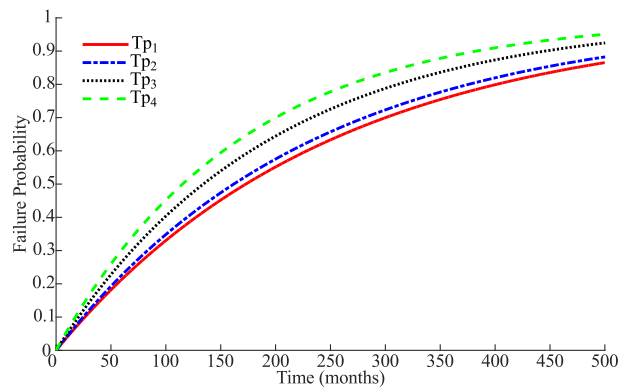


Fig. 17. Non-repairable transformer CDF using exponential distribution.

For this case study the exponential degradation is considered more representative than the Weibull distribution. Besides, note that if the Weibull RUL estimations are used with the BDMP failure logic (cf. Fig. 10), it is always the case that $Fail_Supply_DBB1 = 0$ because the $Tr2$ is in standby operation until the failure of $Tr1$. This scenario makes overlapping failures of $Tr1$ and $Tr2$ impossible at any point in time, i.e., $Fail_Supply_DBB1 = 0$.

Therefore we will use the exponential distribution with $\lambda(T_{p_i}) \approx 1/RUL$ for the failure rate update of circuit breakers and transformers including the different prognostics results for circuit breakers (Table IV) and transformers (Table V).

D. Prognostics-updated Unavailability Estimations

First we will focus on the prognostics-updated unavailability estimations at the asset level and then we will evaluate the system level unavailability according to the BDMP model in Fig. 10. In order to compare the results we will also calculate asset and system level unavailability without prognostics information directly with parameters taken from Table II.

Let us first focus on the unavailability of transformers. The failure behaviour of the transformers is defined in Fig. 10 as the $Fail_Supply_DBB1$ event and this subsystem was previously developed in Fig. 6. The SAN models of $Tr1$ and $Tr2$ are shown in Fig. 11. Initially $Tr1$ is in working state and $Tr2$ is in standby state. The synthesis example in Fig. 6 shows which places are linked through the SAN join operator. When $Tr1$ fails, $Tr2$ transits from standby to working state due to the activation signal sent by the trigger mechanism and when $Tr1$ is repaired, $Tr2$ returns back to standby state (see triggered Markov processes in Subsection II-B1).

Apart from the BDMP constructs validated in Fig. 12, we update dynamically the failure rate of the transformer as specified in Subsection II-D with prognostics prediction results. Namely, up to the first prediction instant, we use the average failure rate displayed in Table II. After predicting the RUL at the first prediction time instant, we update the initial

failure rate with the newly obtained RUL value. We apply the same update process for the subsequent prediction instants and values shown in Table V.

It is necessary to shift by 60 years the prediction instants in Fig. 15 because the transformer monitoring data was obtained at this time instant: $T_{p1}=60y+3y=756m$, $T_{p2}=60y+12y=864m$, $T_{p3}=60y+15y=900m$, $T_{p4}=60y+20y=960m$. Fig. 18 shows asset-level unavailability results for $Tr1$ with exponential update parameters. Vertical dashed lines indicate prediction instants and updates. Confidence intervals are obtained by using the maximum and minimum RUL deviation values (RUL \pm Dev) in Table V as the failure distribution parameters.

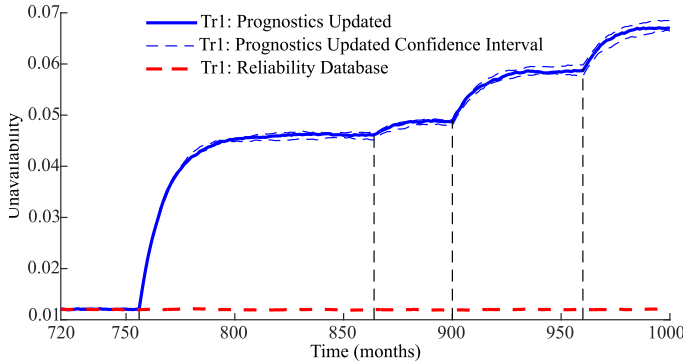


Fig. 18. Unavailability with exponential update: Transformer $Tr1$.

In Fig. 18 we can see that there is a noticeable difference between the reliability database estimate and the prognostics-updated failure estimation using particle filtering results (cf. Table V). The biggest differences can be identified at T_{p1} (756 m) and T_{p3} (900 m) due to the difference with the reliability database information and applied overload conditions, respectively. In this case the reliability database information is below the prognostics-updated unavailability estimation because this information does not take into account the specific overload conditions applied to the model in (8) and (9).

The asset-level unavailability estimation process for circuit breakers is similar to the transformers. The failure rate values are updated with the circuit breaker prognostics prediction parameters in Table IV. The failure behaviour of the circuit breakers is defined in Fig. 10 and the SAN models are shown in Fig. 11. $CB1$ operates actively, whereas the activation of $CB2$ is dependent on the trigger occurrence, i.e. the $Fail_PS_Diesel$ event occurrence. When this event occurs $CB2$ is activated until the $Fail_PS_Diesel$ event is repaired.

In the circuit breakers' case it is necessary to shift by 61 years the prognostics prediction instants because the data gathering architecture for circuit breakers was installed at 61 years: $T_{p1}=61y+50d=734m$, $T_{p2}=61y+63d=734.5m$. Fig. 19 shows asset-level unavailability results for the circuit breaker $CB1$ with exponential update parameters.

Fig. 19 also shows that there is a considerable difference between the unavailability of the circuit breaker updated with linear regression compared with the static reliability estimation using database values. As opposed to the previous case, the circuit breaker unavailability estimation using the database information is more conservative compared with the probability

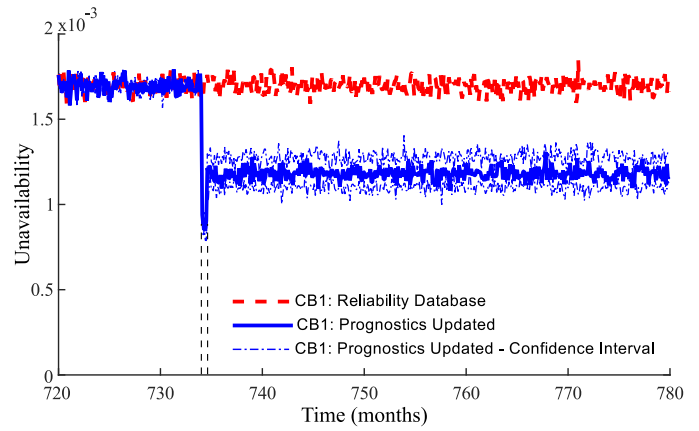


Fig. 19. Unavailability with exponential update: Circuit Breaker $CB1$.

estimate updated with real-time information. The SF_6 data shown in Fig. 13 (b) and Fig. 14 was captured during normal operation and does not include stress conditions such as the overload cases applied to the transformer.

In Figs. 18 and 19 we can see that the proposed approach is able to update the average failure rate estimations using prognostics information at the asset level. Depending on the specific usage of the asset the updated unavailability can be higher (transformer) or lower (circuit breaker) than the average failure rate estimations. The same update process applies to $CB2$ and $Tr2$.

The asset level models can be connected according to the dynamic system failure logic so as to define the system level failure occurrence and evaluate the system unavailability. Accordingly using the SAN model shown in Fig. 11 with the update process reported in Subsection II-D we have evaluated the system-level unavailability. For circuit breakers ($CB1$, $CB2$) and transformers ($Tr1$, $Tr2$) we update the initial failure rate in Table II using the prognostics results displayed in Table IV and V as shown in Figs. 18 and 19. For the rest of assets ($DBB1$, $DBA1$, Diesel) we have taken failure rates displayed in Table II. Fig. 20 shows the system-level unavailability.

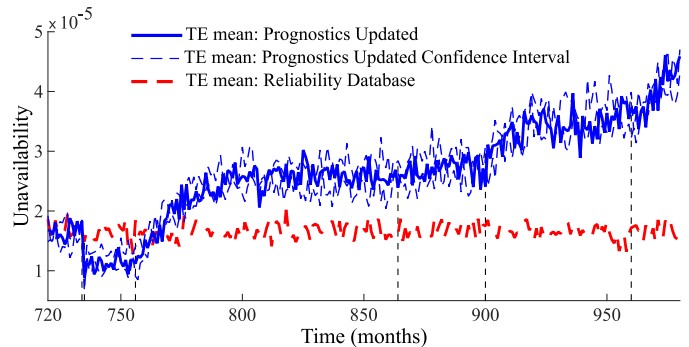


Fig. 20. Power supply of the distribution board: prognostics-updated system-level unavailability.

Fig. 20 confirms that the average estimate of the unavailability of the system changes when online prognostics information is integrated. After applying the system failure logic shown

in Fig. 10, the prognostics-updated system-level unavailability estimation changes dynamically with asset-specific prognostics prediction update values. When updating the unavailability values with the circuit breaker results ($t=734$ m, $t=734.5$ m) the prognostics-updated results are lower, whereas when updating with the transformer prediction results ($t=756$ m, $t=864$ m, $t=900$ m, $t=960$ m) the prognostics-updated unavailability is higher.

Table VI displays mean values shown in Fig. 20.

TABLE VI
MEAN VALUES IN FIG. 20 USING DATABASE AND
PROGNOSTICS-UPDATED ESTIMATIONS

| Time (m) | Database ($\times 10^{-9}$) | Prognostics-updated ($\times 10^{-9}$) |
|----------|-------------------------------|--|
| 730 | 1.61 | 1.64 |
| 740 | 1.51 | 1.02 |
| 750 | 1.71 | 1.15 |
| 760 | 1.69 | 1.52 |
| 800 | 1.52 | 2.58 |
| 850 | 1.7 | 2.73 |
| 875 | 1.63 | 2.49 |
| 910 | 1.66 | 3.37 |
| 950 | 1.85 | 3.64 |
| 960 | 1.72 | 3.73 |
| 980 | 1.72 | 4.59 |

The unavailability calculation of asset-level and system-level failure occurrences is performed by monitoring the marking of the failure place (see Subsection II-C1). That is, for *Tr1* and *CBI* we monitor the marking of the *F* place of the corresponding SAN model (see Fig. 11), and for the system-level failure occurrence we monitor the marking of the top event gate which links all the asset-level models and intermediate failure logic. In this case the top-event gate is an OR gate (see *Fail_DBA1* in Fig. 10, and OR1 in Fig. 11) and we monitor the marking of the place *Y* (Fig. 5 third row).

It is possible to calculate the PDF of the remaining useful life at the system-level by calculating the derivative of the unavailability shown in Fig. 20. However, the PDF of the RUL is not representative on this occasion because the CDF is almost stable at a fixed value after each update step (due to the assets' repair processes).

The prognostics-updated unavailability estimations showed in Figs. 18-20 impact directly on the different dependability attributes considered in this paper. The dynamic dependability model quantifies the probability of occurrence of hazardous events. With the consideration of prognostics information, from the point of view of safety, a better picture of the system health is obtained which can help to improve the definition of safety margin values and avoid hazardous consequences through up-to-date operational information. The dynamic dependability model also includes repair actions, and accordingly, the proposed approach provides indicators to track the evolution of reliability and availability. In this paper we have shown unavailability estimations, but assessing the probability of being in the working state (see Fig. 5 first row) would lead directly to the availability assessment. Finally, from the point of view of maintainability, the proposed approach quantifies the effect of asset-specific prognostics predictions on the system unavailability. On the one hand, it is possible to minimize the RUL waste by defining a threshold

before failure occurrence and adopting maintenance actions in a timely manner. On the other hand, it is possible to save maintenance costs by evaluating the effect of asset-level prognostics-updated unavailability estimations on the system-level unavailability, by implementing waiting options until a critical asset failure occurs.

IV. RELEVANT WORK

Dynamic dependability models analyse reliability, availability, maintainability and safety attributes (confidentiality and integrity are outside the scope of this work). There are dependability models that analyse these attributes from a combinatorial failure logic perspective such as Fault Trees, Reliability Block Diagrams, or Event Tree Analysis [47]. However, the failure of some systems (such as reconfigurable and fault-tolerant systems) is caused by time-ordered event sequences and conditional triggering events which combinatorial logic is not powerful enough to model [26], [48].

Dynamic dependability models enable the modelling and probabilistic analysis of dynamic failure logic systems. There are a range of dynamic dependability models that address stochastic and temporal dependencies: BDMPs [20], Dynamic Fault Trees (DFT) [49], [50], Dynamic Bayesian Networks [51], [52], Dynamic Reliability Block Diagrams [53], State-Event Fault Trees [54], Temporal Fault Trees [55], or hybrid DFT models [56] (see [3] for a more complete overview of dynamic dependability models).

For most of the dynamic dependability models the dynamic criteria come from temporal and stochastic dependencies and they assume *a priori* established dependability estimates. The proposed approach can be situated within a recent body of work aiming to integrate asset-specific operational data with dependability models.

The link between Fault Tree Analysis (FTA) and condition monitoring was introduced with the concept of condition-based FTA (CBFTA) [57]. Failure rate values of the exponential distribution were updated with predefined asset-specific equations. Although this approach addresses relevant properties for the integration of condition monitoring data, prognostics and dynamic dependability concepts are not considered.

A Dynamic Bayesian Networks (DBN) centred method for pre-warning of complex systems is introduced in [58]: Hazard and Operability Study (HAZOP) is used to identify the DBN structure; Markov chains are used to model the degradation based on condition monitoring data, and a DBN is used to evaluate conditional dependencies over different nodes using the condition monitoring data. The safety assessment is based on the inference of hidden states and prognostics focuses on future reliability assessment.

Authors in [59], [60] integrate prognostics concepts with system-level FTA for system level RUL estimation assuming that asset-specific prognostics results are represented with Gaussian probability density functions. The transformation process from prognostics results into FTA is not addressed and for the system-level reliability analysis classical non-repairable FTA is used.

Similarly, Kim *et al.* in [61] integrate prognostics results with static FTA and ETA models to implement a real-time risk monitor. The update process is governed by Bayes' law, which uses prognostics results as prior estimations. It provides a solid direction for the analytic computation of prognostic-updated parameters. Nonetheless, the static nature of the system failure model may be too limiting for time-dependent scenarios.

Likewise, a dynamic failure methodology for the chemical industry was presented in [62] comprising the following steps. Initially, ETA is used to identify failure scenarios and Bayesian prior estimations are initialized using generic data. Then the posterior estimation is computed based on the likelihood function. Finally, the process is updated with plant-specific information. Dynamics come from the plant-dependent data, but the failure specification logic is static and dependencies between failures of different events are not considered.

Pattison *et al.* in [63] use DBN to represent the failure behaviour of wind turbines and update the conditional failure probabilities of DBN nodes with condition monitoring data estimated through a Kalman filter. Although dynamic dependencies between assets are not modelled, asset-specific information is used to update system-level failure probability calculations.

It has been demonstrated that the Bayesian framework provides a solid theoretical framework to perform dependability analyses with time-updated data. To this end, mathematical expressions are required to specify the prior information and the likelihood function for the posterior estimation. However, when including dynamic asset dependencies, mathematical expressions become complicated for complex systems. See [64], [65] for the algebraic framework of non-repairable dynamic dependability.

Although prognostics techniques have been focused on the component-level [12], [13], the system-level RUL prediction problem is starting to attract the interest of researchers. Daigle *et al.* in [66] proposed a distributed RUL estimation method computed through the unscented Kalman filter and analytic equations. The system RUL is defined as a violation of pre-specified system behaviour constraints and this is projected into individual subsystems. A valve model is analysed through decomposing the system health into four health state estimators and then the estimators are combined into two prediction models. Finally, the system RUL is computed as the minimum of all the distributed subsystem RULs, which are computed in parallel through distributed stochastic simulations.

Recently Khorasgani *et al.* in [67] proposed a system-level RUL prediction approach based on physics-of-failure equations. The approach is applied to a rectifier case study comprised of a transformer, two diodes, three capacitors and an inductive load. Particle filtering is first applied to estimate the system health state and then an extended version of the first-order reliability method [68] is used to estimate the system RUL based on the ripple factor of the output current. The system degradation is modelled through the physics-of-failure equations of capacitors, but the degradation models of transformers and diodes are not considered.

Analytic equations are an elegant solution for the system-level prognostics problem, but it is not always feasible to

integrate the physics-of-failure models of all the components and their interactions. Besides, the system failure condition which determines the final RUL varies from system to system. Sometimes it can be defined as a performance indicator of the system [66], [67], but there are situations in which time-dependent failure occurrences of assets are needed to cause the system failure [64], [65] and this complicates the analytical treatment of the problem. In these situations, the applicability of analytic equations for system-level RUL is challenging.

In summary, although there is work in linking condition monitoring and prognostics to static dependability analysis model, very little has been done with dynamic models. Accordingly, we have focused on simulation techniques to address dynamic asset dependencies while including asset-specific prognostics results updated at runtime. We focused on Stochastic Activity Networks because they are able to integrate not only the Bayesian update process, but more complex scenarios [69]. Our goal has not been to adhere to a single prognostics technique, but instead establish a transformation layer to use the results of any prognostics approach. Therefore, the proposed framework is able to integrate independent prognostics results into the dynamic dependability model.

V. DISCUSSION

The study confirms the hypothesis that the proposed framework can be used to improve the dependability analysis in the context of time-dependent scenarios compared with estimations which use database reliability figures and static-logic dependability assessment techniques. As shown in Fig. 20 there is room to adopt fit-for-purpose operational and maintenance decisions accounting for real-time system operation conditions. However, the accuracy of the system-level unavailability estimation and the application of the framework for real-time risk monitoring tasks can be limited by the different issues discussed in this section.

A. Accuracy of the System-level Prognostics Assessment

System-level prognostics estimation is an open problem in the PHM field [6]. The system RUL prediction is more than a simple combination of individual component failures [27], [67] because there are stochastic and temporal dependencies between assets that need to be taken into account.

In this direction, analytical techniques overcome time-consuming simulation issues and they can integrate complex detailed dependencies between variables. For simple systems the use of analytic solutions may be feasible and provides a faster solution. However, as the complexity of the system increases, the implementation of analytical solutions is more challenging. Analytical solutions become too complex for systems which include for each asset or subsystem: time-dependent behaviour, inter-dependencies, and alternative repair strategies. If we include the complexity of the system itself which can be comprised of many different assets and subsystems (depending on the industry) it is not difficult to see the advantage (and necessity) of using simulation methods. Although detailed solutions to this problem have

been proposed recently [67], we think that the scalability of the approach can be an issue for complex systems.

The same situation happens with traditional dependability and PSA techniques (see Table I). Although some of the low level details may be lost, the analyst obtains a manageable system-level approach. Accordingly the proposed modelling process is inspired from well-defined dynamic dependability analysis techniques.

In the proposed framework we have combined dynamic dependability models with prognostics results so as to obtain an up-to-date dependability assessment while accounting for temporal and stochastic dependencies. To this end, we have used BDMP and its trigger mechanism to capture dependencies between assets, but it is possible to apply the same framework with other dynamic dependability formalisms too.

Accordingly, the proposed solution enables the systematic assessment of the system-level unavailability based on the following steps (Fig. 1):

- Asset prognostics predictions and parametrization.
- Basic-events failure rate update.
- System-level dynamic failure logic.
- Top-event failure occurrence quantification.

The asset prognostics prediction model depends on the specific asset under study. These results need to be parametrized according to the degradation behaviour so as to update failure distributions of assets (see Subsection II-D). The system-level dynamic failure logic defines the system degradation model including the interactions between assets. For system-level failure modelling and top-event failure occurrence quantification we use BDMP [20] and SAN [21] models because they are able to model dynamic and dependency properties with a well-defined underlying mathematical foundation.

We acknowledge that all the low-level interactions may not be taken into account with the proposed framework, but this is a conscious trade-off decision that we adopt to enable the systematic probabilistic assessment of prognostics-updated complex systems.

B. Simulation Time

The duration of the Monte Carlo simulations depends on the complexity of the system and the required level of accuracy of the results. For all the presented results, the confidence level is 0.99 and confidence interval is $1e-6$. When updating unavailability estimates with prognostics prediction results, the resampling mechanism also adds computational complexity. In the case study, the results start to converge after $5e6$ iterations, and simulation times can go up to 3 hours with a standard desktop Intel i7 with 8 cores and parallel computing.

For repairable Markov processes it is possible to improve the efficiency of the simulations based on the asymptotic unavailability of the model [70] and this can be applied to the BDMP models in Fig. 12. However, this cannot be applied for all the cases analysed in this work because the asymptotic behaviour depends on the prognostics prediction instants. Some of the models quickly reach the asymptotic behaviour (Fig. 19), but generally the proposed method will require updating the unavailability estimates with new prognostics

prediction results and this will incur a continuous change in the asymptotic behaviour (Fig. 20).

The applicability of the method for real-time risk monitoring and maintenance planning is determined by the relation between simulation time and the prognostics prediction horizon. With a long term prognostics prediction horizon the simulation time is not an issue, but as we reach the end of life of a system, the simulation time can become critical. However, note also that for real-time risk monitoring tasks the predictions will be focused on the short-term horizon and there is no need for long-term predictions (e.g. months ahead as in Fig. 20) which can reduce substantially the computational complexity.

VI. CONCLUSIONS

In this paper we have presented a framework which integrates traditional dynamic dependability assessment techniques with prognostics estimations. The main benefit of the proposed solution is the more realistic and accurate prediction of the system unavailability. We have focused on general properties so that it is possible to integrate other prognostics and dynamic dependability analysis formalism into the framework. To this end, it is necessary to adapt prognostics results, and define transformation rules from any dynamic dependability formalism into a formalism which is able to integrate prognostics and any time-dependent failure logic. Stochastic Activity Networks have shown to be valid for these goals.

Another important aspect arising from this work is the adaptation between prognostics results and dynamic dependability analysis techniques. For deterministic Remaining Useful Life (RUL) prognostics estimations it is reasonable to use the exponential distribution with the RUL estimation as a failure rate parameter. However, the use of the Probability Density Function (PDF) should be analysed on a case-by-case basis depending on the degradation of the asset and target application of the dynamic dependability model. The variance of the PDF of the RUL determines the warning time of the asset, i.e., with higher variance the warning time is longer and vice versa. For safety applications in which the designer needs a timely warning of the system degradation based on failure thresholds, the narrow variance of RUL may prevent the raising of a timely alarm. Therefore, it is necessary to identify the degradation pattern of the system and model the assets' failure behaviour accordingly.

As for the future research activities we have identified four key areas:

- Analysis of analytic techniques: study possible complementary techniques in order to find a feasible direction for online risk estimation.
- Add more flexibility to the framework: although the Weibull distribution is considered a general probability distribution, it is possible to add more flexibility to the framework through customized distributions.
- Analysis of the uncertainty propagation: evaluate other methods to propagate prognostics results, e.g. multiagent systems [6].
- Comparison of maintenance strategies with respect to dependability and cost.

APPENDIX

Before formally defining a Stochastic Activity Networks model, let us define concepts related with the marking of the net. Let P denote the set of places of the network. If S is a set of places ($S \subseteq P$), a marking of S is a mapping $\mu : S \rightarrow \mathbb{N}$. Similarly, the set of possible markings of S is the set of functions $M_S = \{\mu | \mu : S \rightarrow \mathbb{N}\}$. Formally a Stochastic Activity Networks model is defined as follows [21]:

A Stochastic Activity Network (SAN) model is a 5 tuple $SAN = \langle AN, \mu_0, C, F, G \rangle$ where,

- AN is the activity network, which is a 8 tuple $AN = \langle P, A, IG, OG, \gamma, \tau, i, o \rangle$, where
 - P is some finite set of places.
 - A is a finite set of activities.
 - IG is a finite set of input gates; each input gate $ig \in IG$ defined as a triple $ig = (g, ena, f)$ where $G \subseteq P$ is the set of places associated with the gate, $ena : M_G \rightarrow \{0, 1\}$ is the enabling predicate of the gate, and $f : M_G \rightarrow M_G$ is the input function of the gate.
 - OG is a finite set of output gates, each output gate $og \in OG$ defined as a pair $og = (G, f)$.
 - $\gamma : A \rightarrow \mathbb{N}^+$ specifies the number of cases for each activity.
 - $\tau : A \rightarrow \{timed, instantaneous\}$ specifies the type of each activity.
 - $i : IG \rightarrow A$ maps input gates to activities.
 - $o : OG \rightarrow \{(a, c) | a \in [1, \gamma(a)]\}$ maps output gates to cases of activities.
- $\mu_0 \in M_P$ is the initial stable marking.
- C is the case distribution assignment.
- F is the activity time distribution function assignment. An assignment of continuous functions to timed activities such that for any timed activity a , $F_a : M_P \times \mathbb{R} \rightarrow [0, 1]$. Furthermore, for any stable marking $\mu \in M_P$ and timed activity a that is enabled in μ , $F_a(\mu, \cdot)$ is a continuous probability distribution function called the activity time distribution function of a in μ ; $F_a(\mu, \tau) = 0$ if $\tau \leq 0$.
- G is the reactivation function assignment, an assignment of functions to timed activities such that for any timed activity a , $G_a : M_P \rightarrow \mathcal{P}(M_P)$, where $\mathcal{P}(M_P)$ denotes the power set of M_P . Furthermore, for any stable marking $\mu \in M_P$ and timed activity a that is enabled in μ , $G_a(\mu, \cdot)$ is a set of markings called the reactivation markings of a in μ .

The graphical representation and informal description of these constructs can be found in Subsection II-C1. Note that we have deliberately defined case distribution assignments C at a high level because in this work we have not used these constructs. Refer to [21] for more details about the behaviour of Stochastic Activity Networks, Theorems, Proofs, and computation algorithms. Refer to [22] for more details about the implementation framework.

ACKNOWLEDGMENT

This work was supported by the EPSRC through grant number EP/M008320/1. The authors would like to thank the

reviewers for their valuable comments that helped to improve the clarity and completeness of the paper.

REFERENCES

- [1] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr, "Basic concepts and taxonomy of dependable and secure computing," *IEEE Transactions on Dependable and Secure Computing*, vol. 1, no. 1, pp. 11–33, 2004.
- [2] P. Varde and M. G. Pecht, "Role of prognostics in support of integrated risk-based engineering in nuclear power plant safety," *International Journal of Prognostics and Health Management*, vol. 3, no. 8, p. 23, 2012.
- [3] J. I. Aizpurua and E. Muxika, "Model Based Design of Dependable Systems: Limitations and Evolution of Analysis and Verification Approaches," *International Journal on Advances in Security*, vol. 6, pp. 12–31, 2013.
- [4] A. McAfee and E. Brynjolfsson, "Big Data: The management revolution," *Harvard Business Review*, vol. 90, no. 10, pp. 60–68, 2012.
- [5] W. Q. Meeker and Y. Hong, "Reliability meets big data: Opportunities and challenges," *Quality Engineering*, vol. 26, no. 1, pp. 102–116, 2014.
- [6] E. Zio, "Some challenges and opportunities in reliability engineering," *IEEE Transactions on Reliability*, vol. PP, no. 99, pp. 1–14, 2016.
- [7] G. Vachtsevanos, F. Lewis, M. Roemer, A. Hess, and B. Wu, *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Inc., 2007.
- [8] M. Pecht, *Prognostics and Health Management of Electronics*. Wiley, 2008.
- [9] B. Stephen, G. M. West, S. Galloway, S. D. J. McArthur, J. R. McDonald, and D. Towle, "The use of hidden Markov models for anomaly detection in nuclear core condition monitoring," *IEEE Transactions on Nuclear Science*, vol. 56, no. 2, pp. 453–461, April 2009.
- [10] V. M. Catterson, S. D. J. McArthur, and G. Moss, "Online conditional anomaly detection in multivariate data for transformer monitoring," *IEEE Transactions on Power Delivery*, vol. 25, no. 4, pp. 2556–2564, Oct 2010.
- [11] M. J. Daigle, I. Roychoudhury, G. Biswas, X. D. Koutsoukos, A. Patterson-Hine, and S. Poll, "A comprehensive diagnosis methodology for complex hybrid systems: A case study on spacecraft power distribution systems," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 40, no. 5, pp. 917–931, Sept 2010.
- [12] A. K. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mechanical Systems and Signal Processing*, vol. 20, no. 7, pp. 1483 – 1510, 2006.
- [13] J. I. Aizpurua and V. Catterson, "Towards a methodology for design of prognostics systems," in *Annual Conference of the Prognostics and Health Management Society*, vol. 6, no. 22, 2015.
- [14] J. I. Aizpurua, V. M. Catterson, F. Chiacchio, and D. D'Urso, "A cost-benefit approach for the evaluation of prognostics-updated maintenance strategies in complex dynamic systems," in *Proceedings of the European Safety & Reliability Conference (ESREL'16)*, Glasgow, UK, 2016.
- [15] E. Zio, "Reliability engineering: Old problems and new challenges," *Reliability Engineering & System Safety*, vol. 94, no. 2, pp. 125 – 141, 2009.
- [16] V. M. Catterson, J. Melone, and M. S. Garcia, "Prognostics of transformer paper insulation using statistical particle filtering of on-line data," *IEEE Electrical Insulation Magazine*, vol. 32, no. 1, pp. 28–33, January 2016.
- [17] IAEA, "Component Reliability Data For Use In Probabilistic Safety Assessment, IAEA-TECDOC-478," Tech. Rep., 1988.
- [18] K. Kobbacy and D. Murthy, *Complex System Maintenance Handbook*, ser. Springer Series in Reliability Engineering. Springer, 2008.
- [19] European Committee for Standardization, *EN 13306: Maintenance terminology*. BSI, 2010.
- [20] M. Bouissou and J.-L. Bon, "A new formalism that combines advantages of fault-trees and markov models: Boolean logic driven markov processes," *Reliability Engineering & System Safety*, vol. 82, no. 2, pp. 149 – 163, 2003.
- [21] W. H. Sanders and J. F. Meyer, "Stochastic activity networks: Formal definitions and concepts," in *Lectures on Formal Methods and Performance Analysis*, ser. Lecture Notes in Computer Science, E. Brinksma, H. Hermanns, and J.-P. Katoen, Eds. Springer Berlin Heidelberg, 2001, vol. 2090, pp. 315–343.

- [22] D. D. Deavours, G. Clark, T. Courtney, D. Daly, S. Derisavi, J. M. Doyle, W. H. Sanders, and P. G. Webster, "The mobius framework and its implementation," *IEEE Transactions on Software Engineering*, vol. 28, no. 10, pp. 956–969, Oct 2002.
- [23] M. Walker and Y. Papadopoulos, "Qualitative temporal analysis: Towards a full implementation of the fault tree handbook," *Control Engineering Practice*, vol. 17, no. 10, pp. 1115 – 1125, 2009.
- [24] A. Bertolino, A. Calabr, F. Di Giandomenico, and N. Nostro, "Dependability and performance assessment of dynamic connected systems," in *Formal Methods for Eternal Networked Software Systems*, ser. Lecture Notes in Computer Science, M. Bernardo and V. Issarny, Eds. Springer Berlin Heidelberg, 2011, vol. 6659, pp. 350–392.
- [25] C. Di Martino, M. Cinque, and D. Cotroneo, "Automated generation of performance and dependability models for the assessment of wireless sensor networks," *Computers, IEEE Transactions on*, vol. 61, no. 6, pp. 870–884, June 2012.
- [26] J. I. Aizpurua, Y. Papadopoulos, E. Muxika, F. Chiacchio, and G. Manno, "On cost-effective reuse of components in the design of complex reconfigurable systems," *Quality and Reliability Engineering International*, 2017. [Online]. Available: <http://dx.doi.org/10.1002/qre.2112>
- [27] G. Manno, K. Knutsen, and B. Vartdal, "A risk based framework for system level condition monitoring," in *Condition Monitoring and Machinery Fault Prevention Technologies*, 2014.
- [28] S. Sankararaman, "Significance, interpretation, and quantification of uncertainty in prognostics and remaining useful life prediction," *Mechanical Systems and Signal Processing*, vol. 5253, pp. 228 – 247, 2015.
- [29] D. Banjevic and A. K. S. Jardine, "Calculation of reliability function and remaining useful life for a markov failure time process," *IMA Journal of Management Mathematics*, vol. 17, no. 2, pp. 115–130, 2006.
- [30] R. B. Abernethy, *The new Weibull handbook : reliability & statistical analysis for predicting life, safety, risk, support costs, failures, and forecasting warranty claims, substantiation and accelerated testing, using Weibull, Log normal, crow-AMSAA, probit, and Kaplan-Meier models*, 5th ed. North Palm Beach, Fla. : R.B. Abernethy, 2006.
- [31] F. Chiacchio, M. Cacioppo, D. D'Urso, G. Manno, N. Trapani, and L. Compagno, "A weibull-based compositional approach for hierarchical dynamic fault trees," *Reliability Engineering & System Safety*, vol. 109, pp. 45 – 52, 2013.
- [32] P.-Y. Chauv, J.-M. Roussel, J.-J. Lesage, G. Deleuze, and M. Bouissou, "Qualitative analysis of a BDMP by finite automata," in *20th European Safety & Reliability Conf. (ESREL'11)*, Troyes, France, 2011, pp. 2055–2057.
- [33] EDF, "KB3 Platform Tools," <http://researchers.edf.com/software/kb3-44337.html>; Accessed: 16/09/2015, 2015.
- [34] J. F. Meyer, A. Movaghar, and W. H. Sanders, "Stochastic activity networks: Structure, behavior, and application," in *International Workshop on Timed Petri Nets*. Washington, DC, USA: IEEE Computer Society, 1985, pp. 106–115.
- [35] W. H. Sanders and J. F. Meyer, "Reduced base model construction methods for stochastic activity networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 1, pp. 25–36, Jan 1991.
- [36] E. Zio, *The Monte Carlo Simulation Method for System Reliability and Risk Analysis*. Springer, 2013.
- [37] K. D. Rao, V. Gopika, V. S. Rao, H. Kushwaha, A. Verma, and A. Srividya, "Dynamic fault tree analysis using monte carlo simulation in probabilistic safety assessment," *Reliability Engineering & System Safety*, vol. 94, no. 4, pp. 872 – 883, 2009.
- [38] E. Breton, M. Bouissou, and J. Aupied, "A new tool for reliability studies of electrical networks with stand-by redundancies: OPAL," in *International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)*, June 2006, pp. 1–7.
- [39] National Grid, "Transformer replacement program for low-voltage dry-type transformers," 2014.
- [40] CIGRÉ, *Transformer Reliability Survey*, 2015, no. 642.
- [41] Grid Solutions, *CBWatch3 Modular circuit breaker monitoring manual*. [Online]. Available: <http://www.gegridsolutions.com/AlstomEnergy/grid/microsites/grid/>
- [42] S. Rudd, V. Catterson, S. McArthur, and C. Johnstone, "Circuit breaker prognostics using SF₆ data," in *Power and Energy Society General Meeting, 2011 IEEE*, July 2011, pp. 1–6.
- [43] V. M. Catterson, J. J. A. Costello, G. M. West, S. D. J. McArthur, and C. J. Wallace, "Increasing the adoption of prognostic systems for health management in the power industry," *Chemical Engineering Transactions*, vol. 33, 2013.
- [44] J. Coble, "Merging data sources to predict remaining useful life an automated method to identify prognostic parameters," PhD Thesis, University of Tennessee, 2010.
- [45] J. Aizpurua and V. Catterson, "ADEPS: a methodology for designing prognostic applications," in *Proceedings of the Third European Conference of the Prognostics and Health Management Society 2016*, I. Eballard and A. Bregon, Eds., 2016, pp. 86–100.
- [46] IEEE Power and Energy Society, "IEEE Guide for Loading Mineral-Oil-Immersed Transformers and Step-Voltage Regulators," *IEEE Std. C57.91*, 2011.
- [47] J. D. Andrews and S. J. Dunnett, "Event-tree analysis using binary decision diagrams," *IEEE Transactions on Reliability*, vol. 49, no. 2, pp. 230–238, Jun 2000.
- [48] J. I. Aizpurua, E. Muxika, Y. Papadopoulos, F. Chiacchio, and G. Manno, "Application of the d3h2 methodology for the cost-effective design of dependable systems," *Safety*, vol. 2, no. 2, 2016.
- [49] J. B. Dugan, S. J. Bavuso, and M. A. Boyd, "Dynamic fault-tree models for fault-tolerant computer systems," *IEEE Transactions on Reliability*, vol. 41, no. 3, pp. 363–377, Sep 1992.
- [50] G. Manno, F. Chiacchio, L. Compagno, D. D'Urso, and N. Trapani, "Conception of Repairable Dynamic Fault Trees and resolution by the use of RAATSS, a Matlab toolbox based on the ATS formalism," *Reliability Engineering & System Safety*, vol. 121, pp. 250 – 262, 2014.
- [51] H. Boudali and J. B. Dugan, "A continuous-time bayesian network reliability modeling, and analysis framework," *IEEE Transactions on Reliability*, vol. 55, no. 1, pp. 86–97, March 2006.
- [52] L. Portinale and D. C. Raiteri, *Modeling and Analysis of Dependable Systems: A Probabilistic Graphical Model Perspective*. River Edge, NJ, USA: World Scientific Publishing Co., Inc., 2015.
- [53] S. Distefano and A. Puliafito, "Dependability evaluation with dynamic reliability block diagrams and dynamic fault trees," *IEEE Transactions on Dependable and Secure Computing*, vol. 6, no. 1, pp. 4–17, Jan 2009.
- [54] B. Kaiser, C. Gramlich, and M. Förster, "State/event fault tree safety analysis model for software-controlled systems," *Reliability Engineering & System Safety*, vol. 92, no. 11, pp. 1521 – 1537, 2007.
- [55] Y. Papadopoulos, M. Walker, D. Parker, S. Sharvia, L. Bottaci, S. Kabir, L. Azevedo, and I. Sorokos, "A synthesis of logic and bio-inspired techniques in the design of dependable systems," *Annual Reviews in Control*, vol. 41, pp. 170 – 182, 2016.
- [56] F. Chiacchio, D. D'Urso, G. Manno, and L. Compagno, "Stochastic hybrid automaton model of a multi-state system with aging: Reliability assessment and design consequences," *Reliability Engineering & System Safety*, vol. 149, pp. 1 – 13, 2016.
- [57] D. M. Shalev and J. Tiran, "Condition-based fault tree analysis (CBFTA): A new method for improved fault tree analysis (FTA), reliability and safety calculations," *Reliability Engineering & System Safety*, vol. 92, no. 9, pp. 1231 – 1241, 2007, critical Infrastructures.
- [58] J. Hu, L. Zhang, L. Ma, and W. Liang, "An integrated method for safety pre-warning of complex system," *Safety Science*, vol. 48, no. 5, pp. 580 – 597, 2010.
- [59] L. Ramos Rodrigues, J. Gomes, C. de Oliveira Bizarria, R. Galvao, and T. Yoneyama, "Using prognostic system and decision analysis techniques in aircraft maintenance cost-benefit models," in *Aerospace Conference, 2010 IEEE*, March 2010, pp. 1–7.
- [60] J. Gomes, L. Ramos Rodrigues, R. Galvao, and T. Yoneyama, "System level RUL estimation for multiple-component systems," in *Annual Conference of the Prognostics and Health Management Society*, vol. 4, no. 15, 2013.
- [61] H. Kim, S.-H. Lee, J.-S. Park, H. Kim, Y.-S. Chang, and G. Heo, "Reliability data update using condition monitoring and prognostics in probabilistic safety assessment," *Nuclear Engineering and Technology*, vol. 47, no. 2, pp. 204 – 211, 2015.
- [62] A. Roy, P. Srivastava, and S. Sinha, "Dynamic failure assessment of an ammonia storage unit: A case study," *Process Safety and Environmental Protection*, vol. 94, pp. 385 – 401, 2015.
- [63] D. Pattison, M. Segovia Garcia, W. Xie, F. Quail, M. Revie, R. I. Whitfield, and I. Irvine, "Intelligent integrated maintenance for wind power generation," *Wind Energy*, vol. 19, no. 3, pp. 547–562, 2016.
- [64] G. Merle, J. M. Roussel, J. J. Lesage, and A. Bobbio, "Probabilistic algebraic analysis of fault trees with priority dynamic gates and repeated events," *IEEE Transactions on Reliability*, vol. 59, no. 1, pp. 250–261, March 2010.
- [65] G. Merle, J.-M. Roussel, and J.-J. Lesage, "Quantitative analysis of dynamic fault trees based on the structure function," *Quality and Reliability Engineering International*, vol. 30, no. 1, pp. 143–156, 2014.

- [66] M. J. Daigle, A. Bregon, and I. Roychoudhury, "Distributed prognostics based on structural model decomposition," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 495–510, June 2014.
- [67] H. Khorasgani, G. Biswas, and S. Sankararaman, "Methodologies for system-level remaining useful life prediction," *Reliability Engineering & System Safety*, vol. 154, pp. 8 – 18, 2016.
- [68] S. Sankararaman, M. J. Daigle, and K. Goebel, "Uncertainty quantification in remaining useful life prediction using first-order reliability methods," *IEEE Transactions on Reliability*, vol. 63, no. 2, pp. 603–619, June 2014.
- [69] D. Codetta-Raiteri and L. Portinale, "Approaching dynamic reliability with predictive and diagnostic purposes by exploiting dynamic bayesian networks," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 2014.
- [70] M. Bouissou and Y. Lefebvre, "A path-based algorithm to evaluate asymptotic unavailability for large markov models," in *Proceedings of RAMS 2012*, 2002, pp. 32–39.



Gabriele Manno is Senior Researcher at DNV GL in the Strategic Research and Innovation department. He received his Bachelor, M.Sc., and Ph.D. degrees from the University of Catania and a MSc in Business Administration from IISole24Ore Business School. His interests include dependability theory and advanced prognostics as well as digitalization, big data and industrial platforms with specific focus on the shipping industry.



Jose Ignacio Aizpurua is a Research Associate within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. He received his Eng., M.Sc., and Ph.D. degrees from Mondragon University (Spain) in 2010, 2012, and 2015 respectively. He was a visiting researcher in the Dependable Systems Research group at the University of Hull (UK) in 2014. His research interests include prognostics, dependability theory, condition monitoring, and model-based systems engineering.



Victoria M. Catterson is a Senior Lecturer within the Institute for Energy and Environment at the University of Strathclyde, Scotland, UK. She received her B.Eng. (Hons) and Ph.D. degrees from the University of Strathclyde in 2003 and 2007 respectively. Her research interests include condition monitoring, diagnostics, and prognostics for power engineering applications



Yiannis Papadopoulos is a professor and leader of the Dependable Systems research group at the University of Hull. He pioneered the HiP-HOPS MBSA method and contributed to the EAST-ADL automotive design language, working with Volvo, Honda, Continental, Honeywell, and DNV-GL, among others. He is actively involved in two technical committees of IFAC (TC 1.3 & 5.1).



Ferdinando Chiacchio is a Researcher in the Department of Electrical Electronic and Computer Engineering at the University of Catania. He received his laurea and Ph.D. degrees from University of Catania in 2005 and 2010 respectively. His research areas concern reliability, performability, communication protocols for home and industrial control and automation, HPC computing and immunomics.