ADEPS: A Methodology for Designing Prognostic Applications

Jose Ignacio Aizpurua[1] and Victoria M. Catterson[2]

[1,2] *Institute for Energy and Environment, University of Strathclyde, Glasgow, United Kingdom*
jose.aizpurua@strath.ac.uk
v.m.catterson@strath.ac.uk

ABSTRACT

Prognostics applications predict the future evolution of an asset under study, by diagnosing the actual health state and modeling the future degradation. Due to rapidly growing interest in prognostics, different prediction techniques have been developed independently without a consistent and systematic design. In this paper we formalize the prognostics design process with a novel methodology entitled ADEPS (Assisted Design for Engineering Prognostic Systems). ADEPS combines prognostics concepts with model-based safety assessment, criticality analysis, knowledge engineering and formal verification approaches. The main activities of ADEPS include synthesis of the safety assessment model from the design model, prioritization of the system failure modes, systematic prognostics model selection and verification of the adequacy of the prognostics results with respect to design requirements. By linking system-level safety assessment models and prognostics results, design and safety models are updated with online information about different failure modes. This step enables system-level health assessment including prognostics predictions of different failure modes. The end-to-end application of the methodology for the design and evaluation of a power transformer demonstrates the benefits of the proposed approach including reduced design time and effort, complete consideration of prognostics algorithms and updated system-level health assessment.

1. INTRODUCTION

Prognostics is the ability to acquire knowledge about events before they actually occur (Vachtsevanos, Lewis, Roemer, Hess, & Wu, 2007). In engineering, failure prognostics is aimed at foretelling the Remaining Useful Life (RUL) of an asset taking into account the likely future evolution of its failure mode(s). Successful implementations of prognostic techniques provide benefits for maintenance planning and cost-effective operation of assets (Vachtsevanos et al., 2007).

Model-based systems engineering concepts provide mechanisms which can simplify the process of designing suitable prognostics systems for engineering applications (Ramos, Ferreira, & Barcelo, 2012). On the one hand, the systems engineering viewpoint integrates a holistic perspective of the problem, which takes into account asset interrelationships and lifecycle design requirements. On the other hand, models play an important role in the system design process because they are able to (Rumbaugh, Jacobson, & Booch, 1999):

- Capture and state requirements and domain knowledge.
- Organize, examine and edit information of large systems.
- Explore feasibility of alternative solutions.
- Master complex systems.

Many of the current industrial systems address multiple failure modes and their impact on the overall system performance may be very different (Espiritu, Coit, & Prakash, 2007). Accordingly, prognostics implementations of some failure modes will be more cost effective than others. Therefore the selection of an adequate prognostics technique depends on the failure mode justification according to the system design.

After the failure mode selection, it is necessary to choose an adequate prognostics model among the available techniques. So as to reduce the time and effort required to develop an accurate prognostics application we implement knowledge engineering concepts which aid in the systematic prognostics model selection process according to design requirements. In order to avoid undesirable consequences and for correct maintenance planning, prognostics results need to be verified against the prognostics design requirements.

Often the system-level failure is not caused by the isolated failure occurrence of a single failure mode, but due to the simultaneous occurrence of interacting failure modes (Daigle, Bregon, & Roychoudhury, 2014). Accordingly, from the system - level perspective, it is possible to integrate independent component-level prognostics applications in the overall system design process for system-level health assessment.

Integrating all these concepts in the design flow, in this paper

we present a novel methodology entitled ADEPS (Assisted Design for Engineering Prognostic Systems). The main goal of ADEPS is the systematic design of prognostics applications, by choosing *a priori* an adequate prognostics algorithm that meets the system requirements. We organize all the proposed activities around a system design model which acts as the core model for prognostics studies and system design.

In previous work we focused on the systematic prognostics model selection process (Aizpurua & Catterson, 2015b) and formal verification of prognostics results (Aizpurua & Catterson, 2015a). The main contribution of this paper is the conception of ADEPS for the end-to-end design of prognostics applications starting from component-level analysis up to the system-level health assessment. The application of the set of interconnected approaches within ADEPS enables the systematic design of prognostics applications, verification of design requirements with prognostics prediction results and evaluation of the impact of prognostics predictions at the system-level.

The remainder of the paper is organized as follows. Section 2 presents the state-of-the-art analysing existing prognostics methodologies. Section 3 defines the ADEPS methodology and the activities undertaken within the methodology. Section 4 presents the case study to design prognostics applications for power transformers. Finally, Section 5 draws conclusions.

2. RELATED WORK

In recent years a plethora of new techniques have been proposed for prognostics of engineering assets. In this context of independent and rapid evolution of prognostics techniques, there have been some attempts to organise prognostics design steps with a common design thread.

(Uckun, Goebel, & Lucas, 2008) identified the need for a universal methodology to design prognostics and health management systems and listed some of the key activities of ADEPS. Some of these steps have been formalized by others: transformation from high-level requirements to business case (Saxena et al., 2012); metric selection (Saxena et al., 2008); and validation and verification tests (Tang, Orchard, Goebel, & Vachtsevanos, 2011). A key step that the methodology must integrate is the quantification of metrics as a means to consistently compare alternative techniques.

(Cocheteux, Voisin, Levrat, & Iung, 2009) presented requirements for prognostics design including failure mode selection and prognostics model selection. To select failure modes the concept of FMAP (Failure Mode Analysis for Prognostics) is presented, which is inspired from the traditional FMECA (Failure Mode, Effects and Criticality Analysis) (US Department of Defense, 1980) including influential variables, observable indicators and properties. No explicit approach is proposed for prognostics model selection and for requirements validation. The approach is continued in (Cocheteux, Voisin, Levrat, & Iung, 2010) emphasizing system-level performance indicators for supporting proactive maintenance strategies.

Some prognostics methodologies limit their applicability to specific prognostics prediction models: (Kumar, Torres, Chan, & Pecht, 2008) focuses on hybrid prognostics models; and similarly, (Peysson et al., 2009) formalizes the system specification for multi component systems, but it lacks a prognostic model selection process. Instead of focusing on a specific prognostics algorithm which may work for some specific scenarios, a prognostics model selection process is needed for the general applicability of a prognostics methodology. There are other approaches which have considered the prognostics model selection process. For instance, (Lee, Liao, Lapira, Ni, & Li, 2009) presented a methodology for the design of e-manufacturing systems. It ranks prognostics algorithms based on process properties and implements the highest ranked technique. However, the prognostics techniques considered are a subset of data-driven techniques and they do not include model-based and hybrid prognostics techniques.

(Bousdekis, Magoutas, Apostolou, & Mentzas, 2015) does not present a methodology, but they address the model selection concept. They select a subset of prognostics techniques, characterize them in terms of available input and desirable outputs, knowledge of the degradation process, and domain knowledge embedded in a utility function. Subsequently they fed this information into a decision tree learning algorithm so as to obtain a prognostics model-selection tree.

Although the need to develop a generally applicable methodology has been recognized in the literature, most of the proposed prognostics methodologies do not consider a holistic system viewpoint. Some have confined the application of the methodology to specific prognostics algorithms, preventing the generalization of the approach. Some of the proposed approaches have used a particular solution technique (e.g., (Kumar et al., 2008)), while others have not considered the problem in the context of a methodology (e.g., (Bousdekis et al., 2015)).

In the area of maintenance modeling, there have been methodologies focused on the systems engineering viewpoint so as to implement lifecycle maintenance concepts (Takata et al., 2004). For instance, (Ruin, Levrat, Iung, & Despujols, 2014) presented an engineering centered methodology for the quantification of complex maintenance programs.

However, there is no generally applicable methodology which suggests a prognostic technique according to the user requirements, verifies that the obtained results are coherent with the design requirements and re-evaluates the impact of the results at the system-level. Therefore, the main originality of ADEPS arises from the systematic integration of these activities through model-based systems engineering, knowledge

engineering, safety engineering and formal verification approaches taking into account prognostics-specific constraints.

3. ADEPS METHODOLOGY: ASSISTED DESIGN FOR ENGINEERING PROGNOSTIC SYSTEMS

ADEPS focuses on model-based systems engineering concepts and integrates the following properties:

- Design of monitoring system architectures including different design options, e.g., number and type of sensors.
- Failure mode and prognostics model selection guidance.
- Formal verification of the resulting prognostics systems.
- Prognostics-updated system-level health assessment.

ADEPS links system-level design with the design of failure-mode specific prognostics applications through model-based safety assessment (Joshi, Heimdahl, Miller, & Whalen, 2006; Papadopoulos et al., 2011) and prognostics-specific activities (Aizpurua & Catterson, 2015b, 2015a). Besides, we add the capability to update the system-level perspective using prognostics information. Figure 1 shows the ADEPS methodology including different modeling and analysis activities.

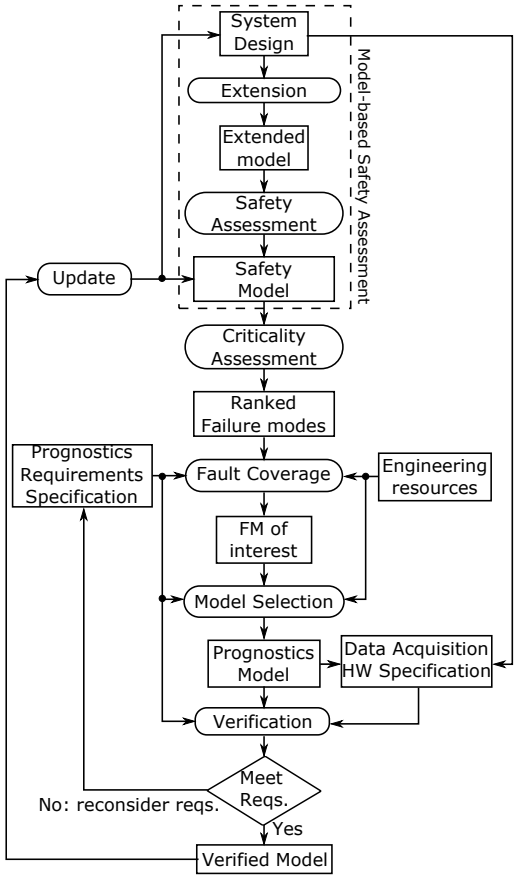

Figure 1. ADEPS methodology.

The methodology starts from the *system design* model specification. This model specifies the functional behavior of the system describing dependencies and nominal operation of system components. It is comprised of connected components via input and output ports. Simulink (MathWorks, 2016) and SysML (Weilkiens, 2011) are two examples of well-known model-based system design specification languages. In this paper we focus on Simulink for subsequent tool support for model-based safety assessment (see Subsection 3.1), but it is possible to repeat the same process using other approaches.

We *extend* the system design model with failure specifications, defining for the system design components all possible deviations from normal operation. The failure specification defines for each component its internal failure modes and the relation between input and internal failure modes, i.e., failure propagation logic. The failure propagation logic specifies the failure responses of a component to its input failure modes. Figure 2a shows two assets with failure propagation and failure transformation properties: Asset_1 *propagates* the input failure mode FM_A to the output port, whereas Asset_2 *transforms* FM_A into another failure mode FM_B.

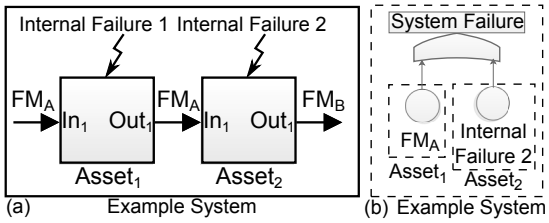

Figure 2. Example system: (a) failure propagation and transformation; (b) FTA synthesis.

Component output responses can be specified with relationships between input failure modes and internal failure events (Papadopoulos et al., 2011):

$$output\ FM{-}out_1 = \textbf{Logic}(internal\ fail,\ input\ FM{-}in_1) \quad (1)$$

where $input\ FM{-}in_1$ covers the input failure mode(s) (resp. output) of the component at port in_1, $internal\ fail$ denotes internal failure events and **Logic** links failure modes using the boolean and temporal logic functions displayed in Table 1.

Table 1. Logic gates.

Logic	Logic Function Behavior	Symbol
Y=**AND**(A,B)	If *A* occurs and *B* occurs, then *Y* occurs	
Y=**OR**(A,B)	If *A* occurs or *B* occurs, then *Y* occurs	
Y=**PAND**(A,B)	If *A* occurs before the occurrence of *B* or at the same time, then *Y* occurs	
Y=**DUR$_\text{d}$**(A)	If *A* occurs for longer or equal than *d* time units, then *Y* occurs	d

For instance, we can annotate the example system in Figure 2a with the failure specification shown in Table 2, where output deviations are defined according to Eq. (1).

Table 2. Failure specification of the components in Figure 2a.

Comp.	Internal FM	Output Deviations	
		Output FM	Logical Causes
Asset 1	Internal Failure 1	FM_A-Out1	FM_A
Asset 2	Internal Failure 2	FM_B-Out1	**OR**(FM_A-In1, Internal Failure 2)

After specifying all the system components with their corresponding failure behavior, the *extended model* is analysed by applying Model-Based Safety Assessment (MBSA) concepts. This process results in the automatic synthesis of *safety models* from the extended design model. The MBSA paradigm enables the automatic transformation of the design model into a safety assessment model in order to evaluate the influence of alternative design decisions on system failure probability (Joshi et al., 2006; Papadopoulos et al., 2011).

In this paper we focus on Fault Tree Analysis (FTA) models (Vesely, Dugan, Fragola, Minarick, & Railsback, 2002) for the system safety model specification although other tools may also be suitable. The FTA model defines the effect of failure modes on the system-level failure expressed with (temporal) combinatorial logic (cf. Table 1). The lowest level basic-events model captures all possible failure modes of the system under study and at the highest level the top-event models the system failure occurrence through the combination of basic events. Assuming that the system in Figure 2a is annotated with the failure specification shown in Table 2, Figure 2b shows the automatically synthesized FTA model.

Applying *criticality assessment* techniques on the FTA model (Van der Borst & Schoonakker, 2001), we sort asset failure modes according to their criticality. These *ranked failure modes* are then connected with the *fault coverage* step to select a failure mode for prognostics studies. The failure mode selection is performed according to the criticality of the failure mode, requirements specification and available engineering resources, i.e., run-to-failure data or knowledge of physics-of-failure model (see Subsection 3.2).

Subsequently we undertake the *prognostics model selection* according to the process presented in (Aizpurua & Catterson, 2015b). For the selected failure mode, we analyse prognostics requirements and available engineering resources (see Subsection 3.3). Once we select the prognostics model, this model is used to perform different predictions and estimate the remaining useful life of the asset under study.

In order to verify the adequacy of the model with respect to design requirements, we implement the *prognostics verification* approach based on formal verification concepts (Aizpurua & Catterson, 2015a). The verification step includes the effect of the possible failures arising from the data acquisition hardware architecture (see Subsection 3.4).

If the selected prognostics model does not *meet the prognostics requirements*, the designer may need to reconsider design decisions. Revision of design requirements may result in the reconsideration of failure mode selection, prognostics model-selection or verification activities. Otherwise, if the designed model meets requirements, it is possible to *update* the system design and failure models with up-to-date health assessment information via prognostics prediction results (see Subsection 3.5). The designer can repeat the process with other failure modes, establishing a set of prognostics prediction models for the different failure modes of the system under study.

3.1. Model-Based Safety Assessment

Model-Based Safety Assessment (MBSA) aids in the design process of safety-related systems (Papadopoulos et al., 2011; Joshi et al., 2006). Namely, it enables the (automatic) synthesis of safety assessment models from operational design models. This process has advantages such as alleviating the need for creating architecture-specific safety models manually for each design alternative. As a result, MBSA introduces a shift in the design process from being manual, tedious and failure-prone towards an automated and reusable approach.

The integration of MBSA concepts within ADEPS enables to frame from a system-level perspective the design of prognostics applications. This is achieved through safety models which define how the combination of different failure modes cause the system failure. In turn, for each of these failure modes, it is possible to design a prognostics model systematically according to ADEPS. MBSA plays a pivotal role in ADEPS by providing a centralized system design framework.

The design and safety models evolve dynamically to include design decisions adopted at different stages and prognostics results obtained at different prediction times. On the one hand, ADEPS makes use of the synthesized safety models to rank failure modes according to their criticality. This way, any architectural design decision will impact the underlying safety model, and it will affect the criticality analysis, failure mode ranking and prognostics model selection. On the other hand, the links between safety and prognostics models enable the continuous update of safety models with prognostics results. As a result, the designer obtains an up-to-date (system-level) health assessment including future degradation trends.

There are different MBSA approaches which extract different analysis models from design specifications (see (Aizpurua & Muxika, 2013) for an overview). In this paper we use the HiP-HOPS approach because it provides flexibility and support for specifying the design model and extracting (temporal) FTA models (Papadopoulos et al., 2011). As shown in Figure 3,

the specification of the design model in HiP-HOPS is done with hierarchical block diagrams which can include different design alternatives (e.g., alternative redundancy strategies).

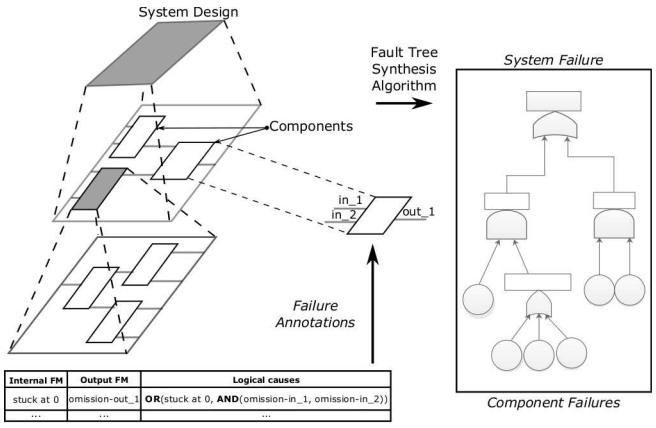

Figure 3. System design, failure annotations and Fault Tree synthesis step in HiP-HOPS.

For each component in the design model its failure behavior is specified including internal malfunctions and the logic that links internal failures with the incoming failures (see Eq. (1) and Figure 3). HiP-HOPS takes the design model with failure annotations and analyses the failure propagation logic from basic causes to the system-level failure occurrence. The component connections in the design model with the failure propagation logic enable the automated synthesis of FMECA and FTA models from the extended system design model.

3.2. Fault Coverage

Engineering systems are comprised of different assets which work in cooperation to perform a system-level function. Each of these assets has different failure modes which have a different impact on the system-level failure occurrence. Prognostics applications may prioritize a single fault type, aging behaviour or a number of important failure modes. The fault coverage activity focuses on failure mode selection to design a prognostics model.

In the proposed methodology the failure mode selection is driven by three parameters: criticality of the failure mode, available engineering resources for the failure mode under study and design requirements. Ideally all the necessary engineering resources (run-to-failure data and/or physics of failure models) for all the failure modes of the system will be available for the designer. Given the open choice to select a failure mode for prognostics studies, we focus on the extraction of indicators to assist in the failure mode selection.

The failure mode criticality has been considered as a useful design indicator for prognostics failure mode selection. FMECA is a valid approach for criticality assessment and failure mode selection (e.g., (Uckun et al., 2008), (Cocheteux et al., 2009)). However, FMECA is a qualitative cause-effect approach which requires a thorough understanding of the failure mechanisms. Even with a perfect understanding of the failure modes, sometimes it is difficult to determine the criticality of failure modes due to intermediate events.

FTA is an effect-cause approach which can integrate qualitative and quantitative assessments. If the link between basic events and top-event failure occurrence is identified, it has potential to automate the criticality analysis through importance measurements (Van der Borst & Schoonakker, 2001). Accordingly we can classify failure modes into two groups:

- Critical failure modes: system breakdown occurs when the component failure mode occurs.
- Non-critical failure modes: system breakdown does not occur when the component failure mode occurs.

According to this logic we use the FTA model so as to weight the contribution of each component to the system-level failure occurrence. Namely, we evaluate when the occurrence of a failure mode causes the system-level failure and extract the failure criticality index (Hilber, 2008): the ratio between the number of system failures caused by the failure mode to the total number of system failures.

After ranking all the failure modes with respect to their criticality, we select the most critical one and check if there are engineering resources available for this failure mode. If there are engineering resources, we proceed with the next activity of the methodology. However, if there are no resources, we take the next ordered failure mode until finding a failure mode with available engineering resources.

This fault coverage process assures the prognostics assessment of the most critical failure mode for which there are available engineering resources.

3.3. Prognostics Model Selection

Prognostics prediction models can be classified into the following high-level groups: data-driven, model-based and hybrid approaches (Aizpurua & Catterson, 2015b).

The selection of the group depends on the available engineering resources. Namely, when run-to-failure data or knowledge of the system's degradation equation is available, data-driven or model-based approaches are selected respectively. When both engineering resources are available, the selection of the high-level group incurs a trade-off decision between the availability of statistically significant run-to-failure data and complexity of the degradation equation. If the complexity is manageable and there is enough run-to-failure data hybrid prognostics techniques can be selected.

In (Aizpurua & Catterson, 2015b) we presented ordered design decision points to choose a prognostics model using the

failure mode under study, design requirements and available engineering resources. The decision points for each of the high-level groups are different (Aizpurua & Catterson, 2015b):

- Data-driven: RUL format, data monotonicity, prediction horizon, knowledge of degradation states, availability of expert knowledge, Markovian degradation process analysis, availability of multiple run-to-failure data...
- Model-based: availability of observations, linearity of the degradation trend, assumptions about Gaussian noise.
- Hybrid: availability of expert knowledge, complementary parameter estimation, combination of features.

Data-driven approaches include more decision points because there are simply more of these techniques to choose between. Model-based techniques are more specific to the field of study. Namely, physics of failure models are specifically designed to predict the degradation of a particular failure mode. Hybrid prognostics models include the systematic combination of data-driven and model-based prognostics techniques with complementary properties.

We organize these decision points strategically in different flowcharts so as to aid the designer in the prognostics model-selection process according to design requirements and available engineering resources—please see (Aizpurua & Catterson, 2015b) for the exhaustive list of design decision points.

3.4. Verification of Prognostics Requirements

The verification of prognostics applications is crucial for building trust in their predictions. Prognostics engineering literature suggests prognostics metrics for the evaluation and verification of the correctness, timeliness, and confidence of prognostics models (Saxena et al., 2008). The quantification of these metrics requires case-by-case implementation of their logic with each application. A requirements verification technique which is model independent would assist in doing this task semi-automatically for any prognostics model.

Furthermore, online prognostics applications depend on a data acquisition hardware architecture to generate correct prognostics predictions. Accordingly, when verifying prognostics requirements compliance, it is necessary to include the effect of hardware failures on prognostics predictions.

We use formal verification techniques for the integrated verification of prognostics applications including hardware and software components. Figure 4 shows the overall verification approach (Aizpurua & Catterson, 2015a). We define a *probabilistic model-checking pattern* which is used to synthesize prognostics prediction results, asset information and the data architecture specification. This pattern is formally expressed with prognostics requirements. The probabilistic model-checking engine performs an exhaustive verification to check if the requirements are satisfied by the prognostics pattern. If satisfied, the results can be used as argumentation of verified design requirements.

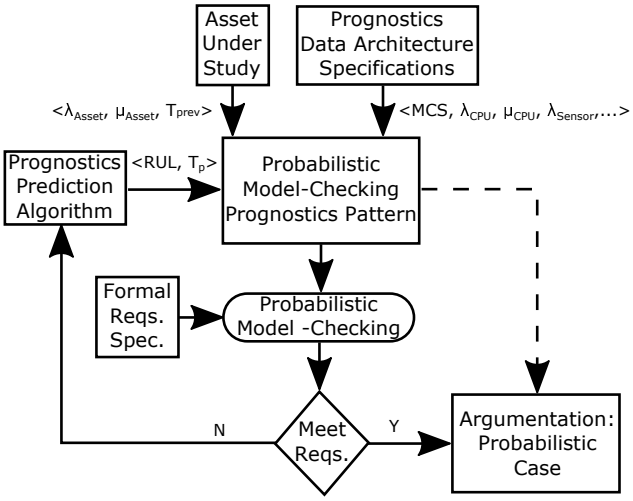

Figure 4. Verification activity of the methodology.

In this paper, we use the PRISM tool (Kwiatkowska, Norman, & Parker, 2011) for the implementation of probabilistic model-checking concepts, but other tools may also be applicable. PRISM enables the specification of state-based probabilistic models including continuous and discrete time Markov chains, Markov decision processes, probabilistic timed automata and Markov decision processes. Among these formalisms we use Continuous-Time Markov Chains (CTMC) for the specification of the prognostics pattern and we use Continuous Stochastic Logic (CSL) for the verification of system requirements—see (Aizpurua & Catterson, 2015a) for the rationale and limits of the selected approach.

Figure 5 shows the probabilistic model-checking prognostics pattern which is specified as a CTMC in PRISM (Aizpurua & Catterson, 2015a).

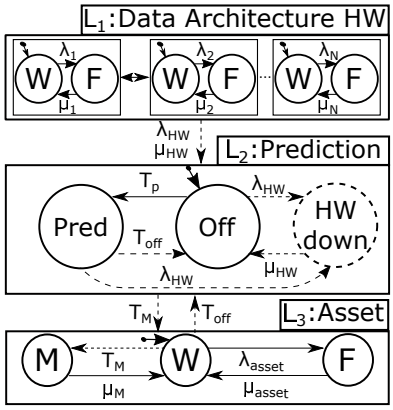

Figure 5. Probabilistic model-checking prognostics pattern.

The probabilistic model-checking pattern takes as input:

- Specification of the *data architecture hardware* including the combination of component failures that cause the system failure, i.e. Minimal Cut Set (MCS) (Vesely et al., 2002) and failure (λ) and repair rate (μ) values of system components.
- Prognostics *prediction* results including the prognostics prediction time (T_p) and RUL estimation.
- Information on the asset under study including ground truth data (λ_{Asset}), mean time to repair the asset (μ_{Asset}) and periodic preventive maintenance period (T_{prev}).

The MCS defines the performance of the data acquisition hardware (λ_{HW}, μ_{HW}). Under nominal conditions, the prediction module performs predictions at different instants T_p. The result of these predictions is the estimation of the RUL, which in turn, can be transformed into a maintenance interval T_M taking into account a Safety Factor (SF): T_M=RUL-SF. After each prediction, the prediction module goes back to *Off* state with time T_{off}. The asset module takes into account ground truth data and it is repaired after a constant time interval.

At every prognostics prediction instant we update pattern parameters with prognostics prediction results. Then we verify if requirements are met via probabilistic model-checking. Requirements are formally expressed using the CSL formalism as a function of the parameters in Figure 5. The outcome of the verification of prognostics results is the quantification of prognostics metrics.

CSL formulas are interpreted over the states of the CTMC to check if the stated formula is satisfied (Katoen, Kwiatkowska, Norman, & Parker, 2001). The main operators for property specification are: **P** for the specification of the *probability* that the observed execution of the model satisfies a given specification; **S** to compute *steady-state* probabilities; and **R** to express *reward-based* properties. The **P** operator is used in conjunction with temporal operators defined over a state or a path of the CTMC model.

The main operators for temporal state or path specifications are: **G** for properties that need to be satisfied *globally*, **F** for properties that become true *eventually*, **X** for properties that become true in the *next state* and **U** for properties that are not satisfied *until* another property is true.

The **S** operator is used to reason about the steady-state operation and it has no timed-variants. As for the **R** operator it is possible to combine it with **F** for *reachability* properties, **C** for *cumulative* properties, and **I** for *instantaneous* properties.

All these operators have time-bounded extensions. Table 3 displays some examples of formal properties expressed in CSL and their informal meaning. Note that *prop* denotes property, which is a condition defined over the CTMC model, e.g. in Figure 5: *prop*=(Prediction=Off $\wedge$ Asset=F).

Table 3. Examples of CSL properties.

#	CSL	Meaning
1	$\mathbf{P}_{=?}$[**F**<t *prop1*]	Prob. of *prop1* is true eventually before *t*
2	$\mathbf{P}_{=?}$[**G**[t_1,t_2] *prop1*]	Prob. globally *prop1* is true within the time instant [t_1, t_2]
3	$\mathbf{P}_{=?}$[*prop1* **U**<=t *prop2*]	Prob. of *prop2* not being true until *prop1* is not true in the interval [0,t]
4	$\mathbf{R}\{$"*oper*"$\}_{=?}$ [**C**<=t]	Expected cumulative *operational* time
5	$\mathbf{R}\{$"*time*"$\}_{=?}$ [**F** prop]	Accumulated *time* until prop is satisfied

We can define conditions (rewards) in PRISM to evaluate False Positive (FP) and False Negative (FN) metrics. Assuming that *asset=1* identifies failed state, *asset=2* identifies maintenance state, *pred=2* identifies hardware down state in the prediction module, and CI_x identifies the confidence interval of the event *X*, where $X=\{FP, FN\}$; we define the following conditions:

$$FN=(asset=1)\wedge(RUL+T_p>\lambda_{Asset}-CI_{FN})\vee(pred=2) \quad (2)$$

$$FP=(asset=2)\wedge(\lambda_{Asset}-(RUL+T_p))>(CI_{FP}) \quad (3)$$

When reward Equations (2) and (3) are satisfied by the prognostics pattern in Figure 5, they will be increased by a unit quantifying the occurrence of these events.

3.5. Update System Design with Prognostics Results

Traditionally MBSA is used early in the design phase, whereas prognostics predictions are performed after the asset is deployed for some time. However, it is possible to align both approaches by updating the extended design model and safety model parameters according to prognostics prediction results at different prediction time instants. The main benefit of this step is the up-to-date consideration of the system health state including prognostics prediction results.

Prognostics results (i.e., RUL estimations) can be seen as random variables which can be categorized into three groups:

- A deterministic value.
- A deterministic value $\pm$ a confidence interval.
- A probability density function.

These values can be used for failure specification or analysis of maintenance strategies. For failure specification, prognostics predictions constitute the basic failure unit of the failure mode or asset under study. It is possible to propagate these values for further reliability evaluations using high level approaches such as FTA. To this end, it is necessary to parameterize prognostics results with an equivalent Probability Density Function (PDF). The deterministic RUL can be approximated with the exponential distribution calculating the

bounds for the confidence interval (Banjevic & Jardine, 2006). As for RUL results specified as a PDF, they can be approximated using a parameterized PDF with regression techniques.

After the parameterization, it is possible to recompute the system level failure probability through the FTA model. To this end, the failure specification of the failure modes needs to be updated through conditional probabilities. There are analytic formulations that integrate conditional probability functions, e.g., Bayes theory (Gelman, Carlin, Stern, & Rubin, 2003). There are also simulation based approaches which can update distribution parameters during simulation, e.g., Stochastic Activity Networks (SAN) (Sanders & Meyer, 2001).

4. CASE STUDY

Power transformers are important assets in electrical power grids with a direct impact on the reliability of the grid. The main goal of power transformers is to transfer the electric energy from one voltage level to another under magnetic induction reaction. One of its main benefits is the reduction of power transmission cost by increasing the transmission voltage and reducing the required current for transmission.

The main components of the transformer are the tank, winding, core, tap changer and bushings. The tank is the assembly and physical protection for the active part of the transformer, i.e. winding and core. Winding is a conductor material which aims to satisfy the increase in power rating and voltage requirements. Windings are arranged as shells around the core, where each strand is wrapped with insulation paper. Core is a magnetic circuit which reduces core losses. Tap changer regulates the voltage level by transferring electrical power from one tap winding to the adjacent one and bushings are the electrical isolation between tank and windings. The failure of any of these components can cause the transformer failure.

Transformers are the most expensive assets in the power network with a costly and time-consuming repair process. As a result, the implementation of condition-based maintenance strategies through prognostics is a potential solution to extending their useful life. The tank is a cornerstone part of the power transformer design, but its degradation can be assessed easily with visual inspection. The winding is a critical subsystem of the transformer which initiates most of the transformer failure events and its health assessment requires investigation of all root causes (CIGRÉ, 2015). As a result, in this case study we will focus on the winding analysis.

Figure 6a shows high-level dependencies between the transformer and its data acquisition hardware system. The data acquisition hardware system monitors the generated current and temperature of the winding circuit (see Subsection 4.4). Figure 6b shows the block diagram of the active part of the transformer. The winding generates a magnetic flux which travels within the core. The core increases efficiency and it provides an effective magnetic flux. The winding provides as output effective output current at the designed level and circuit temperature for monitoring purposes.

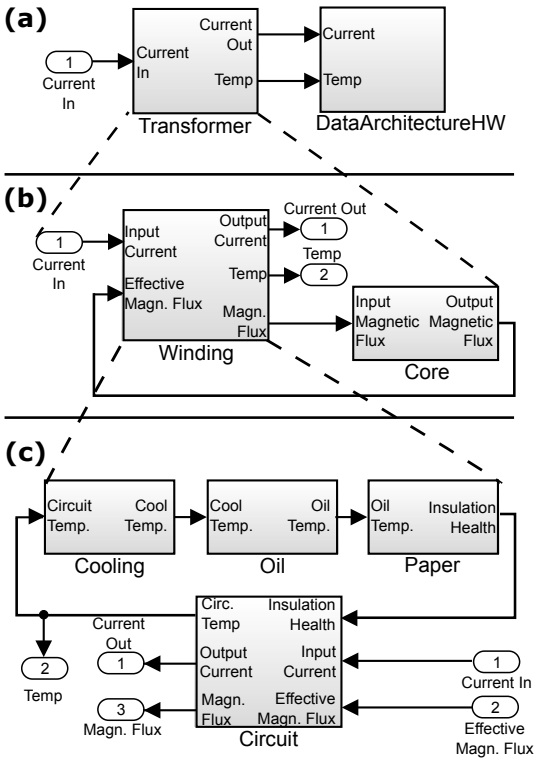

Figure 6. Transformer design: (a) high-level dependencies; (b) active part of the transformer; (c) winding block diagram.

Figure 6c shows the winding block diagram. The oil and paper act as insulators materials for the winding. The cooling system keeps the oil temperature at acceptable levels and the paper insulation degradation process depends on the oil temperature (CIGRÉ, 2015). The paper acts as an insulation material of the winding circuit which produces a magnetic flux to which travels through the core and an output current which is produced using the effective magnetic flux.

4.1. Model-based Safety Assessment (MBSA)

In order to apply the MBSA concepts, first we extend the design model in Figure 6c with failure annotations. Table 4 displays the functional failure modes and their deviations.

Figure 7 shows the propagated failure modes after annotating the winding design model in Figure 6c with failure deviations in Table 4.

After adding the failure specifications (which are subject to expert knowledge) to the design model, we synthesize automatically from Figure 7 the winding FTA model shown in Figure 8 via HiP-HOPS.

Winding failure (or equivalently Omission-Magn. Flux event)

Table 4. Failure specification of the winding subsystem — cf. Figure 6c.

Component	Internal FM	Output Deviations	
		Output FM	**Logical Causes**
Cooling	Pump Failure	Omission-Cool Temp.	Pump Failure
Oil	Low Oil Level, Moisture	PaperDegradation-Oil Temp.	**OR**(Omission-Cool Temp., Low Oil Level, Moisture)
Paper	Partial Discharge	ExcessivePaperDegradation-Insulation Health	$\mathbf{DUR_d}$(PaperDegradation-Oil Temp.)
		ElectricArc-Insulation Health	$\mathbf{DUR_d}$(Partial Discharge)
		PaperDegradation-Insulation Health	PaperDegradation-Oil Temp.
Circuit	Short Circuit	Deformation-Output Current	**PAND**(Short Circuit, PaperDegradation-Insulation Health)
		ExcessivePaperDegradation-Output Current	ExcessivePaperDegradation-Insulation Health
		ElectricArc-Output Current	ElectricArc-Insulation Health
Winding	-	Omission-Magn. Flux	**OR**(Deformation-Circuit.OutputCurrent, ElectricArc-Circuit.OutputCurrent, ExcessivePaperDegradation-Circuit.OutputCurrent)

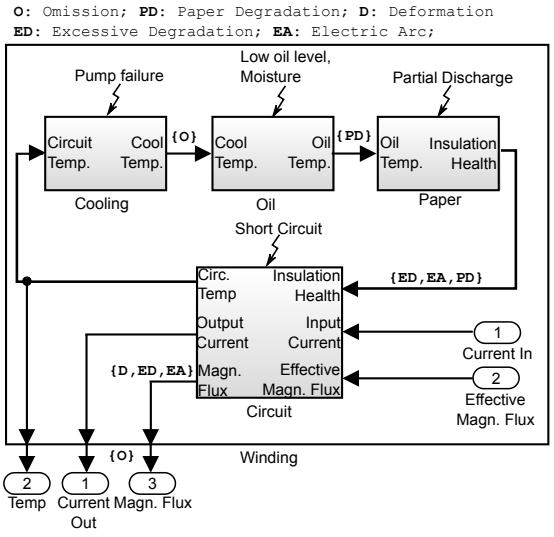

Figure 7. Extended model of the winding subsystem.

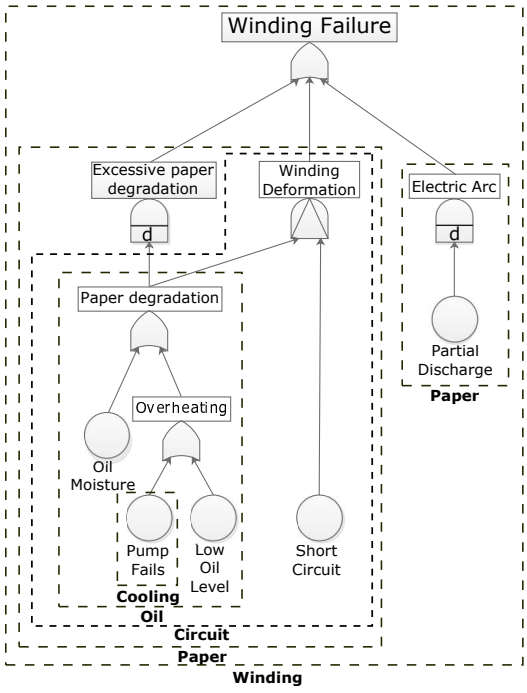

Figure 8. FTA model of the winding subsystem.

occurs either because of *winding deformation*; *electric arc*; or because the paper degradation lasts more than a predefined period of d time units and reaches an *excessive degradation level*. The root causes for excessive paper degradation are oil moisture, pump failure or low oil level. The winding deformation happens when first the paper degradation event occurs and then the short circuit failure happens. Finally, the electric arc occurs as a result of a partial discharge event which lasts more than a time period of d time units.

The quantification of the FTA model in Figure 8 requires implementing the logic of the gates in Table 1. Since there is no available solution in HiP-HOPS for the duration gate, we opt for implementing stochastic Monte Carlo simulations by extending previous work with Dynamic Fault Trees (Aizpurua, Muxika, Papadopoulos, Chiacchio, & Manno, 2016) including repairable basic events and the duration gate logic.

Table 5 displays the failure and repair rates of the basic events shown in Figure 8 extrapolated from (CIGRÉ, 2015) assuming exponential distributions for failure and repair events. The duration of the events causing excessive paper degradation and electric arc are assumed to be 5 and 10 years respectively.

4.2. Fault coverage

For the failure mode study, we take all the failure modes in the winding FTA model (Figure 8) and assess the criticality of these failure modes. Making use of the Monte Carlo simulations for the FTA solution, we have extended the failure criticality index presented in (Aizpurua et al., 2016) for re-

Table 5. Failure and repair rates of the analysed failure modes.

Failure Mode	$\lambda\,(years^{-1})$	$\mu\,(years^{-1})$
Oil moisture	0.00038	0.25
Pump failure	0.000272	0.25
Low Oil Level	0.000987	0.25
Short Circuit	0.000955	0.25
Partial Discharge	0.0104	0.25

pairable systems.

Table 6 displays ordered failure models with respect to their criticality and available engineering resources: run-to-failure data (D), or knowledge of physics of failure equations (K).

Table 6. Ranked failure modes.

Failure Mode	Failure Criticality Index	Available Resources
Exc. Paper Degradation	0.5408	K, D
Electric Arc	0.4331	K, D
Winding Deformation	0.0021	-
Short Circuit	0.0021	-
Partial Discharge	0.0012	K, D
Paper Degradation	1.019e-5	K, D
Low oil	6.052e-6	-
Oil moisture	2.39e-6	-
Pump fail	1.7557e-6	-

As we can see in Table 6 the most critical failure mode is the excessive paper degradation followed by the electric arc. Accordingly, we select excessive paper degradation for subsequent prognostics assessment.

4.3. Prognostics Model-selection

According to the prognostics model selection process (cf. Subsection 3.3), first we choose a high-level prognostics algorithm group. We focus on model-based approaches because observation data and knowledge of physics of failure equations are available (cf. Table 6). We proceed as follows in the model-selection process with the flowchart in Figure 9 (Aizpurua & Catterson, 2015b):

- The degradation equation and observation data are available. Besides, the process is Markovian and therefore, Bayesian tracking solutions are considered.
- The degradation of the transformer aging is not linear.
- There is no need to assume a Gaussian distribution for the state and noise.

Therefore, we choose the `Particle Filter` algorithm. Transformer aging involves deterioration of the paper insulation due to temperature. A model for paper aging is given in IEEE standard C57.91 (IEEE Power and Energy Society,

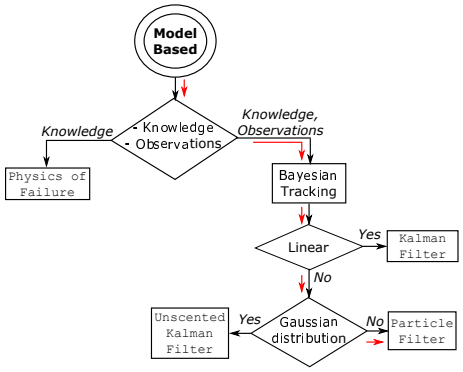

Figure 9. Model-based prognostics algorithm selection.

2011). The standard defines an aging acceleration factor based on the hotspot temperature. This equation can be rearranged to give a particle filter process model, by converting it into a recurrence relation for remaining paper life (Catterson, Melone, & Garcia, 2016):

$$L_t = L_{t-1} - e^{15000/383 - 15000/(273+\Theta_{H_t})} + u_t \qquad (4)$$

where t is the time index, L_t is the RUL at time t, Θ_{H_t} is hotspot temperature at time t, and u_t is the process noise.

Equation (4) is updated with measurement information that is the hotspot temperature measured by:

$$\Theta_{H_t} = \Theta_{to} + (80 - \Delta\Theta_{to/a,R}) \times K^{2m} \qquad (5)$$

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

At each simulation step we calculate the degradation state (Eq. (4)) and the weight of the likelihood of each particle—please see (Catterson et al., 2016) for more details. Figure 10 shows estimated RUL values at different prediction instants (T_p) based on the Particle Filter equations and available observation data (ambient and top oil temperature, load current).

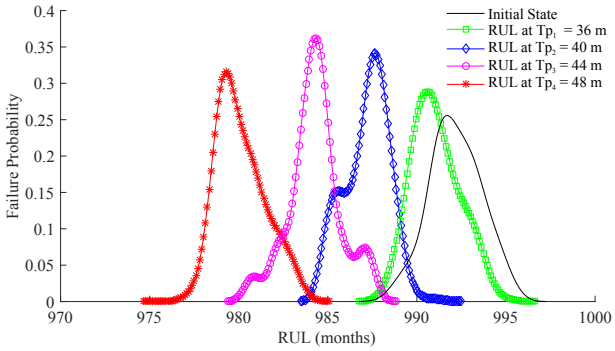

Figure 10. RUL predictions at different prediction times T_p.

Transformer RUL predictions results (in months) are as follows: T_{p_1} (36m) = 992.5 $\pm$ 2.89m; T_{p_2} (40m) = 987 $\pm$ 2.64m; T_{p_3} (44m) = 984.3 $\pm$ 2.76m; and T_{p_4} (48m) = 979.3 $\pm$ 3.05m.

4.4. Verification of Prognostics Results

In order to perform the verification of the prognostics application we need to take into account the hardware architecture which acquires the necessary data and calculates prognostics estimations (cf. Figure 6a, *DataArchitectureHW*).

For high criticality transformers, a typical data acquisition architecture will use temperature and current sensors. Data collection will employ a High-Frequency Network (HFN) where available (e.g., critical substations), supported by the lower frequency SCADA network. The SCADA platform includes a Remote Terminal Unit (RTU) in the substation reporting to the central Master Station (MS). Both SCADA and higher frequency data are then archived, using a system such as a PI Historian. Figure 11 shows the inner architecture of the data architecture hardware block shown in Figure 6a.

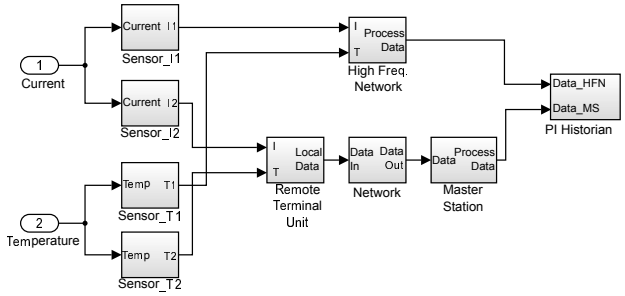

Figure 11. Transformer data architecture hardware.

According to the verification process defined in Subsection 3.4, we need to identify the failure condition of the data acquisition hardware architecture, i.e., Minimal Cut Set (MCS) function. To this end, we repeat the process with HiP-HOPS failure annotations for the data acquisition hardware architecture. Figure 12 shows the FTA model of the data acquisition hardware architecture.

The MCS equation of the FTA in Figure 12 is as follows:

$$MCS = PI \vee (T_1 \wedge T_2) \vee (I_1 \wedge I_2) \vee [HFN \wedge (MS \vee Net \vee RTU)] \quad (6)$$

where PI indicates the failure of the PI historian, T_i and I_i indicate the failure of the i-th temperature and current sensor respectively, Net indicates the failure of the network, and HFN, MS, and RTU indicate the failure of the identified components. For the analysis we have used hypothetical failure and repair rates displayed in Table 7.

Taking the prognostics pattern in Figure 5 as a reference, we use Equations (2) and (3) to evaluate false negative and positive metrics, respectively. For the transformer we have used the following reliability figures (in months): λ_{asset}=1/1038 m^{-1} (transformer failure rate); μ_m=0.1 m (maintenance time); μ_{asset}=1 m (repair time); CI_{FP}=10 m; CI_{FN}=4 m, and SF=4 m (safety factor).

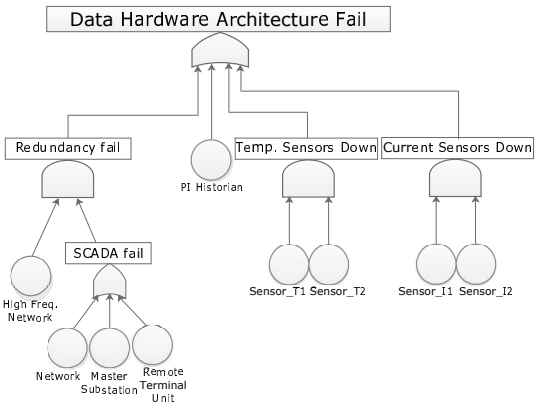

Figure 12. FTA model of the data architecture hardware.

Table 7. Failure and repair rates of the hardware components.

Component	$\lambda\ (years^{-1})$	$\mu\ (years^{-1})$
PI Historian	0.001	0.25
T_i, I_i, MS, RTU, HFN	0.01	0.25
Network	0.0001	0.25

After specifying FP and FN rewards in PRISM using the property #4 in Table 3, Figures 13 and 14 show the obtained results. If the designer has a threshold for an acceptable rate of false positive or false negative events, it can be identified whether these values are acceptable or not.

For FP events we have used different prediction results from Figure 10 including their deviation (see Figure 13). After the prediction at T_p=T_{p_2}*-deviation*, the prognostics predictions become accurate enough to avoid false positive occurrences.

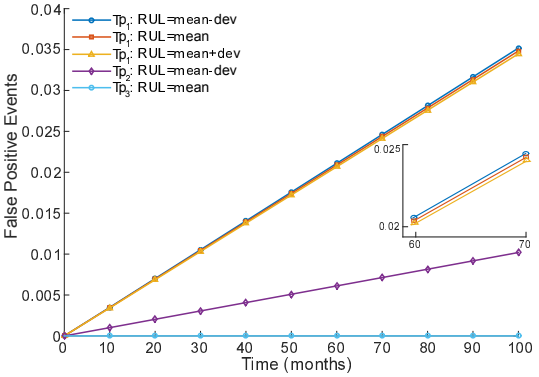

Figure 13. False positive event.

For the false negative event we have used the mean RUL prediction value at T_{p_1}. Figure 14 shows the difference between

the metric with and without the hardware omission failure effect. The incorporation of the hardware omission failure enables us to account for the uncertainties that may arise in the prognostics application environment.

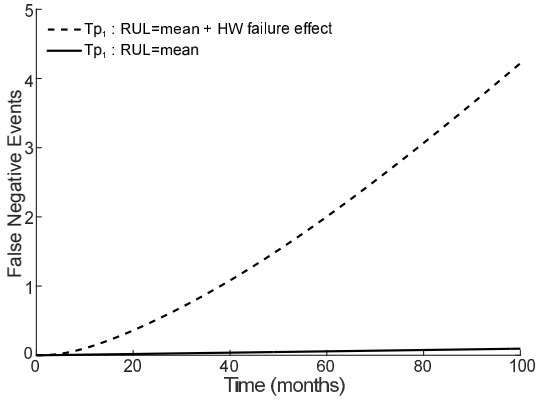

Figure 14. False negative event.

The failure of the data acquisition hardware architecture causes a failure to produce a prognostics prediction, which in turn leads to a non-updated maintenance schedule at the asset level. We have defined a penalty function using rewards which includes the effects of downtime (i.e., the asset in a failed state incurs a penalty of 1 while in maintenance it incurs 0.5), and false positive and negative events multiplied by the probability of failure of the asset under study.

Figure 15 shows the effect of different failure rates of both the transformer and data acquisition hardware failures. Apart from the uncertainty arising from the application context, the specification of the failure rate of the asset (or ground truth) has uncertainties too. The ground truth is estimated either under some specific conditions or it is an average failure behavior. Therefore when using it as a reference failure model, uncertainty estimations should be included. In this case study uncertainty in the ground truth value makes little difference.

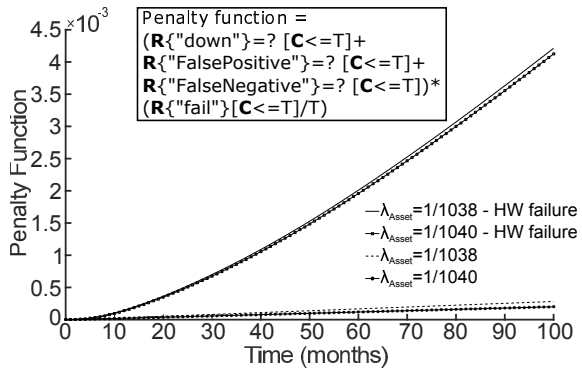

Figure 15. Penalty function.

4.5. Update Design Models

The failure specifications of the FTA model in Figure 8 can be updated dynamically with prognostics prediction results obtained for the paper degradation model (cf. Figure 10).

To this end, first we approximate the PDF values of the transformer degradation prediction with the corresponding distribution. Although the PDFs in Figure 10 can be approximated with Gaussian or Weibull distributions, the transformer's paper degradation process is governed by the exponential law (cf. Eq. (4)). Accordingly, in order to adhere to the real degradation process, we implement the exponential degradation law taking the mean and standard deviation values of the PDFs in Figure 10. Using the resampling mechanism of SAN we update initial failure rate distributions (cf. Table 5) at different prediction times during the simulation.

Figure 16 shows the obtained system-level health assessment results updated with prognostics predictions at T_p=3 years and T_p=4 years for the FTA model shown in Figure 8.

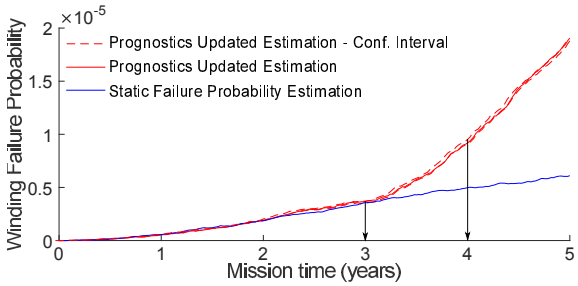

Figure 16. Winding failure probability.

As shown by Figure 16, prognostics predictions provide an up-to-date health assessment estimation of the asset under study including possible changes in the deterioration rate due to environmental influences. In this case the degradation rate increases due to the harsh environmental conditions affecting the paper degradation model in Eq. (4).

5. CONCLUSION

In this paper we have presented a novel methodology entitled ADEPS (Assisted Design for Engineering Prognostic Systems) for the implementation of a prognostics-centred lifecycle design process. ADEPS integrates a holistic system-level design process that includes systems engineering and reliability engineering concepts through model-based safety assessment techniques.

In ADEPS importance measurements are implemented to rank failure modes according to their criticality and aid in the failure mode selection for prognostics studies. Besides, integrated knowledge engineering and probabilistic model-checking techniques permit systematic prognostics model selection and exhaustive verification of requirements respectively.

ADEPS also includes connections between dependability and prognostics approaches so as to perform system-level health assessments updated with prognostics prediction results.

ADEPS provides benefits including a reduction of the design time, complete consideration of prognostics algorithms and dynamically updated system-level health assessment.

As for our possible future work, we may focus on the implementation of the following activities:

- Evaluate the performance of ADEPS though different case studies.
- Complete the verification of requirements integrating other prognostics metrics.
- Refine the prognostics model selection process analysing decision points and their possible dynamic organization.

ACKNOWLEDGMENT

This work was supported by the EPSRC through grant number EP/M008320/1.

REFERENCES

Aizpurua, J. I., & Catterson, V. (2015a). On the use of probabilistic model-checking for the verification of prognostics applications. In *IEEE Int. Conf. on Intelligent Computing and Information Systems, Symposium on Knowledge Engineering for Decision Support System.*

Aizpurua, J. I., & Catterson, V. (2015b). Towards a methodology for design of prognostics systems. In *Annual conference of the prognostics and health management society* (Vol. 6).

Aizpurua, J. I., & Muxika, E. (2013). Model-based design of dependable systems: Limitations and evolution of analysis and verification approaches. *International Journal on Advances in Security*, *6*(1, 2).

Aizpurua, J. I., Muxika, E., Papadopoulos, Y., Chiacchio, F., & Manno, G. (2016). Application of the D3H2 methodology for the cost-effective design of dependable systems. *Safety*, *2*(2), 9. doi: 10.3390/safety2020009

Banjevic, D., & Jardine, A. K. S. (2006). Calculation of reliability function and remaining useful life for a markov failure time process. *IMA Journal of Management Mathematics*, *17*(2), 115-130. doi: 10.1093/imaman/dpi029

Bousdekis, A., Magoutas, B., Apostolou, D., & Mentzas, G. (2015). Supporting the selection of prognostic-based decision support methods in manufacturing. In *Proc. of int. conf. on enterprise information systems* (p. 487-494). doi: 10.5220/0005372104870494

Catterson, V. M., Melone, J., & Garcia, M. S. (2016, January). Prognostics of transformer paper insulation using statistical particle filtering of on-line data. *IEEE Electrical Insulation Magazine*, *32*(1), 28-33. doi: 10.1109/MEI.2016.7361101

CIGRÉ. (2015). *Transformer Reliability Survey* (No. 642).

Cocheteux, P., Voisin, A., Levrat, E., & Iung, B. (2009). Prognostic design: requirements and tools. In *Proc. of MITIP 2009.* Bergame, Italy.

Cocheteux, P., Voisin, A., Levrat, E., & Iung, B. (2010). System performance prognostic: context, issues and requirements. In *Proc. of AMEST.* Lisbon: IFAC.

Daigle, M. J., Bregon, A., & Roychoudhury, I. (2014, June). Distributed prognostics based on structural model decomposition. *IEEE Transactions on Reliability*, *63*(2), 495-510. doi: 10.1109/TR.2014.2313791

Espiritu, J. F., Coit, D. W., & Prakash, U. (2007). Component criticality importance measures for the power industry. *Electric Power Systems Research*, *77*(56), 407 - 420. doi: http://dx.doi.org/10.1016/j.epsr.2006.04.003

Gelman, A., Carlin, J. B., Stern, H. S., & Rubin, D. B. (2003). *Bayesian data analysis*. Chapman and Hall/CRC.

Hilber, P. (2008). *Maintenance optimization for power distribution systems* (PhD Thesis). KTH.

IEEE Power and Energy Society. (2011). IEEE Guide for Loading Mineral-Oil-Immersed Transformers and Step-Voltage Regulators. *IEEE Std. C57.91*.

Joshi, A., Heimdahl, M., Miller, S., & Whalen, M. (2006). *Model-Based Safety Analysis* (Vol. NASA/CR-2006-213953; Tech. Rep. No. ID: 20060006673). NASA.

Katoen, J.-P., Kwiatkowska, M., Norman, G., & Parker, D. (2001). Process algebra and probabilistic methods. performance modelling and verification. In (pp. 23–38). Springer. doi: 10.1007/3-540-44804-7_2

Kumar, S., Torres, M., Chan, Y., & Pecht, M. (2008, June). A Hybrid Prognostics Methodology for Electronic Products. In *IEEE IJCNN 2008* (p. 3479-3485). doi: 10.1109/IJCNN.2008.4634294

Kwiatkowska, M., Norman, G., & Parker, D. (2011). PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. of CAV'11* (Vol. 6806, pp. 585–591). Springer.

Lee, J., Liao, L., Lapira, E., Ni, J., & Li, L. (2009). Informatics Platform for Designing and Deploying e-

Manufacturing Systems. In *Collaborative Design and Planning for Digital Manufacturing* (p. 1-35). Springer London. doi: 10.1007/978-1-84882-287-0_1

MathWorks. (2016). *Matlab/Simulink.* `http://www.mathworks.com`;.

Papadopoulos, Y., Walker, M., Parker, D., Rüde, E., Hamann, R., Uhlig, A., ... Lien, R. (2011). Engineering failure analysis and design optimisation with HiP-HOPS. *Engineering Failure Analysis*, *18*(2), 590-608.

Peysson, F., Ouladsine, M., Outbib, R., Leger, J.-B., Myx, O., & Allemand, C. (2009, June). A Generic Prognostic Methodology Using Damage Trajectory Models. *IEEE Transactions on Reliability*, *58*(2), 277-285. doi: 10.1109/TR.2009.2020123

Ramos, A., Ferreira, J., & Barcelo, J. (2012). Model-based systems engineering: An emerging approach for modern systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, *42*(1), 101-111. doi: 10.1109/TSMCC.2011.2106495

Ruin, T., Levrat, E., Iung, B., & Despujols, A. (2014). Complex maintenance programs quantification (CMPQ) to better control production systems. *Journal of Manufacturing Technology Management*, *25*(4), 491-509. doi: 10.1108/JMTM-04-2013-0042

Rumbaugh, J., Jacobson, I., & Booch, G. (1999). The unified modeling language reference manual [Computer software manual].

Sanders, W. H., & Meyer, J. F. (2001). Stochastic activity networks: Formal definitions and concepts. In *Lectures on formal methods and performance analysis* (Vol. 2090, p. 315-343). Springer Berlin Heidelberg. doi: 10.1007/3-540-44667-2_9

Saxena, A., Celaya, J., Balaban, E., Goebel, K., Saha, B., Saha, S., & Schwabacher, M. (2008). Metrics for Evaluating Performance of Prognostic Techniques. In *PHM 2008* (pp. 1–17).

Saxena, A., Roychoudhury, I., Celaya, J., Saha, B., Saha, S., & Goebel, K. (2012). Requirements Flowdown for Prognostics and Health Management. In *Infotech@Aerospace*. AIAA. doi: 10.2514/6.2012-2554

Takata, S., Kirnura, F., van Houten, F., Westkamper, E., Shpitalni, M., Ceglarek, D., & Lee, J. (2004). Maintenance: Changing role in life cycle management. *CIRP Annals - Manufacturing Technology*, *53*(2), 643 - 655. doi: http://dx.doi.org/10.1016/S0007-8506(07)60033-X

Tang, L., Orchard, M., Goebel, K., & Vachtsevanos, G. (2011). Novel Metrics and Methodologies for the Verification and Validation of Prognostic Algorithms. In *Aerospace Conference, 2011 IEEE* (p. 1-8). doi: 10.1109/AERO.2011.5747583

Uckun, S., Goebel, K., & Lucas, P. (2008, Oct). Standardizing research methods for prognostics. In *PHM 2008* (p. 1-10). doi: 10.1109/PHM.2008.4711437

US Department of Defense. (1980). *Procedures for Performing, a Failure Mode, Effects, and Criticality Analysis (MIL-STD-1629A)*. Washington, DC.

Vachtsevanos, G., Lewis, F., Roemer, M., Hess, A., & Wu, B. (2007). *Intelligent Fault Diagnosis and Prognosis for Engineering Systems*. John Wiley & Sons, Inc. doi: 10.1002/9780470117842

Van der Borst, M., & Schoonakker, H. (2001). An overview of psa importance measures. *Reliability Engineering & System Safety*, *72*(3), 241-245.

Vesely, W., Dugan, J., Fragola, J., Minarick, & Railsback, J. (2002). *Fault Tree Handbook with Aerospace Applications* (Handbook). NASA.

Weilkiens, T. (2011). *Systems engineering with SysML/UML: modeling, analysis, design*. Morgan Kaufmann.

BIOGRAPHIES

Jose Ignacio Aizpurua is a Research Assistant 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) during autumn 2014. His research interests include prognostics, RAMS analysis 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.