Combining Dynamic Bayesian Networks and Continuous Time Bayesian Networks for Diagnostic and Prognostic Modeling

Jordan Schupbach
Dept. Mathematical Sciences
Montana State University
Bozeman, USA
jordan.schupbach@montana.edu

Elliott Pryor, Kyle Webster, John Sheppard
Gianforte School of Computing
Montana State University
Bozeman, USA
{first.last}@montana.edu

***Abstract*—The problem of performing general prognostics and health management, especially in electronic systems, continues to present significant challenges. The low availability of failure data, makes learning generalized models difficult, and constructing generalized models during the design phase often requires a level of understanding of the failure mechanism that elude the designers. In this paper, we present a new, generalized approach to PHM based on two commonly available probabilistic models, Bayesian Networks and Continuous-Time Bayesian Networks, and pose the PHM problem from the perspective of risk mitigation rather than failure prediction. We describe the tools and process for employing these tools in the hopes of motivating new ideas for investigating how best to advance PHM in the aerospace industry.**

I. Introduction

In previous work, we have discussed the development of tools for diagnostic modeling using Bayesian and dynamic Bayesian networks. The tool we developed – the Standards-based Analysis Platform for Predictive Health and Integrated Reasoning Environment (SAPPHIRE) [1] conforms to IEEE Std 1232-2010, Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE) [2].

More recently, we have also presented work utilizing the same type of information needed for creating diagnostic Bayesian networks to create more advanced models designed to prognostics. The models are based on Continuous-Time Bayesian Networks (CTBN) [3] and represent systems as factored continuous-time conditional Markov processes. As such, they are well-suited to addressing problems in Prognostics and Health Management (PHM). The work we discussed included methods for converting diagnostic dependency models (i.e., D-matrices) and reliability fault trees into CTBNs. We have also provided methods for incorporating performance functions and decision nodes to produce continuous-time decision networks (CTDN) [4] that have since been embodied in a new tool called the Continuous-time Hazard Analysis and Risk Mitigation (CHARM) system.

Although the two models are natural to combine, there has been little to no application of these models used in combination for conducting both diagnostics and prognostics under a single modeling framework. In this work, we discuss an approach to combining their use, illustrating that combination with simple and illustrative models to demonstrate the utility of this combined modeling strategy. Our intent is not to focus on the SAPPHIRE or CHARM tools specifically, but rather to discuss how such models (DBNs and CTDNs) can be used together in an integrated fashion to support PHM. Therefore, we use SAPPHIRE and CHARM for example purposes only.

Ultimately, this paper is about describing a new process for PHM that combines elements of diagnostics and health state information as a starting point from which predictive diagnostics (i.e., prognostics) can then be performed. The proposed process also incorporates the elements of probabilistic risk analysis as an alternative method for evaluating the effectiveness of the PHM process. Currently, this process is being formalized under contract with the US Navy in support of the F-35 Lightening II program to assist the F-35 Joint Program Office (JPO) in better meeting its requirements for PHM.

The rest of this paper is organized as follows. In Section II, we provide necessary background information by defining what we mean by PHM and covering some of the necessary material related to Bayesian networks and CTBNs. In Section III, we present the specific type of Bayesian network we use for diagnostics. We then explain the prognostic model based on CTBNs in Section IV. We tie the together by describing a PHM process in Section V and then wrap up our discussion in Section VI by considering the path forward.

II. Background

The purpose of this section is to provide background information necessary to follow the method presented in this paper. To accomplish these, we consider a means for defining what we mean by Prognostics and Health Management (PHM) relative to current views in the industry, followed by presenting the main tools we will employ in our approach.

A. Prognostics and Health Management

Simply put, there is little agreement in the field about the scope and relevant practice of PHM. Therefore, we begin by providing some perspectives on how we are using the term. We take a rather literal approach when considering the PHM discipline in that we believe PHM must include *both* aspects of state estimation (health management) and prediction (prognostics). This is contrary to many who believe the focus is on health management as a practice of diagnostics and condition-based maintenance, which is largely centered on the state estimation task alone.

In 2006, Vichare and Pecht noted that "The term 'diagnostics' pertains to the detection and isolation of faults or failures. 'Prognostics' is the process of predicting a future state (of reliability) based on current and historic conditions. Prognostics and health management (PHM) is a method that permits the reliability of a system to be evaluated in its actual life-cycle conditions, to determine the advent of failure, and mitigate the system risks [5]." While introducing the notion of prediction, we see both inspiration and limitation in this view. The inspiration is that we can use reliability information *during the design phase* as a means of creating initial predictive models and considering the risks associated with system failure. The limitation is that there is no tie between the diagnostics and prognostics in this view of PHM.

Also in 2006, Kalgren *et al.* provide a definition of PHM. Here they say PHM is "a health management approach utilizing measurements, models, and software to perform incipient fault detection, condition assessment, and failure progression prediction. The capability allows end users to improve fault isolation, better plan maintenance, reduce or eliminate inspections, and decrease time-based maintenance intervals with confidence. When coupled with Autonomic or Performance-based Logistics, PHM enables improved mission-critical system reliability and availability, reduced logistics delay time and tail, on-demand repair actions and sparing, as well as an overall decrease in life cycle costs [6]. Since their view includes the concept of *incipient fault detection* and *condition assessment*, we begin to see ties back to the current health state of the system. However, their views related to *failure progression prediction* largely depend upon physics-of-failure models, which are not generalizable or scalable in complex systems.

As a third and more recent example begins to pose PHM more literally, as we do. Specifically, Riu *et al.* say, "Prognostic and Health Management (PHM) systems support aircraft maintenance through the provision of diagnostic and prognostic capabilities, leveraging the increased availability of sensor data on modern aircraft. Diagnostics provide the functionalities of failure detection and isolation, whereas prognostics can predict the remaining useful life (RUL) of the system [7]." In this definition, however, the diagnostics is limited to on-board systems, and the prognostics is focused specifically on remaining useful life. We adapt this idea to consider off-board diagnostics and time-to-failure.

To conclude this section, we also consider the ideas expressed in the recently approved IEEE Standard 1856, which divides the definition of PHM into two parts [8]. First, the standard defines prognostics to be "the process of predicting an object system's RUL by predicting the progression of a fault given the current degree of degradation, the load history, and the anticipated future operational and environmental conditions to estimate the time at which the object system will no longer perform its intended function within the desired specifications." Once again, the focus is on remaining useful life and on failure progression, which would largely be from a PoF point of view. Second, the standard defines health management as "The process of decision-making and implementation of actions based on the estimate of the state of health derived from health monitoring and expected future use of the system." This is good in the sense that the dependence is on state of health, but the issue excludes the health assessment itself.

We suggest that PHM depends on understanding the current health state of the system, which is at the heart of diagnostics. Previously, we have asserted that all aspects of health assessment, including fault detection, localization, isolation, and even determining there are no faults are diagnostic processes [9]. Therefore, we assert that the PHM process begins with diagnosis. From there, the question arises as to when future failures might occur, and this is the realm of prognosis. We also like to refer to prognostics and predictive diagnostics in that we also want to know what faults are occurring when. This then sets up a pipeline process whereby PHM consists of a sequence of steps: 1) monitoring, 2) health state assessment (diagnosis), 3) prediction, 4) assessment, and 5) action. Thus it is an evidence-based decision making process that leads to the overall support of the system.

B. Risk-based PHM

As mentioned in Section II-A, we are motivated by the definition by Vichare and Pecht by drawing on reliability information in doing PHM. To that end, we employ a "risk-based" approach to PHM. By this, we mean that we seek to introduce a cohesive framework that includes both diagnostics and prognostics and incorporates effects or hazards using the same model semantics. By building hazards into the model itself, predictions can be made not only about likely faults, but also about the effects that may occur as a result of those faults. Furthermore, our approach incorporates user-specified performance functions that place value on various system states, which allows one to assess potential impact on mission outcomes should the hazards be realized vs averted. The framework also allows different conditions to be modeled such that strategies for risk mitigation can be employed directly into the decision making process. This supports a multi-objective view, whereby tradeoffs can be assessed when determining the best course of action in maintaining a system.

The basic approach employed involves combining two different types of models, one focused on diagnostics and the other on prognostics. We employ Bayesian networks as a way to address the diagnostics problem, which allows us to reason with uncertainty so as to best consider the state of health in the system. Once the health state is determined, we use that as a form of "virtual evidence" in a companion model based on a continuous time Bayesian network (CTBN) to reason through time. The CTBN uses estimates of the health state as the starting point for this temporal process but also propagates failures through different hazard scenarios to determine how best to mitigate the risks associated with future failure.

C. Bayesian Networks

A considerable amount of literature has been written on Bayesian networks, including by the authors of this paper. For our purposes here, we will present only a light introduction and refer the reader to the literature for more detail.

In short, a Bayesian network is a graph-based representation of a joint probability distribution. Given a set of random variables $\mathbf{X} = \{X_1, \ldots, X_n\}$, the Bayesian network provides a compact representation of the joint distribution $P(\mathbf{X}) = P(X_1, \ldots, X_n)$ by applying the product rule of probabilities and properties of conditional independence among the variables. As a result, a Bayesian network can be regarded as a "factored" representation of the joint distribution corresponding to

$$P(X_1, \ldots, X_n) = \prod_{X_i \in \mathbf{X}} P(X_i | \text{Pa}(X_i)).$$

Here, we represent a conditional probability, $P(X_i|X_j)$ in a directed acyclic graph where the vertex for X_j is connected by an outward directed edge to the vertex for X_i, in which case we say X_i is the parent of X_i. In other words, $\text{Pa}(X_i) = X_j$. The complete graph structure, combined with a parameterization of the local distributions associated with each random variable X_i corresponds to the specification of a Bayesian network. More information on Bayesian networks, including material on representation, inference, and learning, can be found in the book by Koller and Friedman [10].

Previously, we developed SAPPHIRE to incorporate basic prognostic abilities as well through the use of "Dyanamic" Bayesian networks (DBN). In the current discussion, we do not employ the predictive capabilities of the DBN in SAPPHIRE; however, work has been done using DBNs for diagnostics by incorporating temporal measurements from the past to obtain better estimates of current health state [11]. Therefore, we briefly describe the primary differences between a Bayesian network and a DBN.

In a basic DBN, we begin with a normal Bayesian network; however, we consider that network as a representation of a "prior" distribution over the state of the system being model. As a prior distribution, we modify the notation slightly to reflect this as an initial point in time:

$$P(X_1^0, \ldots, X_n^0) = \prod_{X_i^0 \in \mathbf{X}} P(X_i^0 | \text{Pa}(X_i^0)).$$

Here, the superscript "0" indicates the initial point in time. We also refer to this as a "template" network where we can then provide a temporal dependency as

$$P(\mathbf{X}^t | \mathbf{X}^{t-1})$$

for all $t > 0$. Thus this defines a DBN as a first-order multivariate Markov process.

D. Continuous Time Bayesian Networks

To support the predictive modeling, we propose using a relatively new model that combines concepts from Bayesian networks, dynamic Bayesian networks, and Markov processes. As with the discussion above, we refer the interested reader to fundamental literature on CTBNs and provide a brief, high-level (albeit mathematical) description here.

At the heart of a CTBN is what is referred to as a *continuous time Markov Process*. We use the definition given by Perreault. "Let $\mathcal{X}$ be a continuous time random process, consisting of a set of variables $\mathbf{X}$ that change as a function of continuous time. A CTMP is a model over $\mathcal{X}$ consisting of two parts: an initial distribution $P_{\mathcal{X}}(0)$ and a transition intensity matrix $\mathbf{Q}_{\mathcal{X}}$ defined over the states of $\mathcal{X}$. Each entry $q_{i,j}$ in row i, column j of the matrix $\mathbf{Q}_{\mathcal{X}}$ defines the non-negative intensity with which the process will transition from state x_i to state x_j as a function of time. The diagonal entry for some row i and column i is denoted $q_{i,i}$ or simply q_i, and is constrained to be the negative sum of the rest of the row. Formally, $q_i = -\sum_{j \neq i} q_{i,j}$ [4]." The distribution indicating if the process remains in state i is exponential with rate q_i:

$$f_{q_i} = -q_i \exp(q_i t).$$

If it is determined that a transition out of state i is occuring at some time t, then $\mathbf{X}$ transitions from state x_i to state x_j according to a multinomial distribution with probabilities

$$P(x_j | x_i, t) = -\frac{q_j}{q_i}.$$

A continuous time Bayesian network is a generalization of the continuous time Markov process. It is based on a *conditional* Markov process where the behavior of the Markov process depends upon the state of another Markov process. Similar to the above, we use the definition of a CTBN by Perreault. "A Continuous Time Bayesian Network $\mathcal{N}$ is a factored representation of a CTMP over a set of discrete random variables $\mathbf{X} = \{X_1, X_2, \ldots, X_n\}$. The model consists of two parts: a graph structure $\mathcal{G}$ and a set of parameters $\mathbf{P}$. Graph $\mathcal{G}$ is a directed, possibly cyclic graph with nodes corresponding to the variables in $\mathbf{X}$. Parameterization $\mathbf{P}$ is a set of Conditional Markov Processes, one for each $X_i \in \mathbf{X}$, conditioned on its parents in graph $\mathcal{G}$ [4]." Of note is that the intensity matrices that go with each CTMP is conditioned on parent states. We refer to these intensity matrices as "conditional intensity matrices" (CIM). As described in Section IV, we use this model to capture the failure and hazard dynamics of the system under test.

III. Diagnostic Bayesian Networks

In this section, we present the basic formulation for the diagnostic Bayesian network. Recall that a CTMP (and thereby a CTBN) requires a prior distribution to kick start the process. We use the diagnostic Bayesian network as the basis for that prior distribution. Furthermore, we define the diagnostic Bayesian network using a formulation that is relatively well known in the fault diagnostics field known as a D-matrix [12].

A. D-Matrices

A variety of diagnostic models are possible for purposes of establishing health state. These include models such as fault trees [9], first principle models [13], expert systems [14], and Bayesian networks [15]. For our purposes due to the fact they integrate well in our overall framework, we choose to use a Bayesian network that is derived from a diagnostics dependence matrix, also known as a D-matrix [12].

A D-matrix is a binary matrix $\mathbf{D}$ that maps faults to tests. More formally, let $\mathbf{F} = \{F_1, \ldots, F_d\}$ be a set of a faults or diagnostic conclusions to be drawn in a system. Assume each F_i is a Boolean variable that can either by true or false. Unless otherwise asserted, we assume that all faults are true unless otherwise indicated. Let $\mathbf{T} = \{T_1, \ldots, T_n\}$ be a set of tests or information sources designed to detect the presence of faults. Assume each test is also a Boolean variable where "true" indicates the test has failed and "false" indicates the test has passed. Finally, let $\mathbf{D}$ be a $d \times n$ binary matrix where

$$D_{i,j} = \begin{cases} 1 & F_i \text{ is detected by } T_j \\ 0 & \text{otherwise.} \end{cases}$$

Given this, a D-matrix can be represented as a Bayesian network, similar to the model described by Schwe *et al.* with their QMR-DT system [15]. Specifically, each F_i and T_j are defined as random variables (i.e., vertices) in the network, and conditional dependence relationships are defined where $D_{i,j} = 1$ indicates F_i is a parent of T_j. Then, to reduce computational complexity, the parent relationships of a test T_j are interpreted to correspond to a "noisy-OR" model, as defined by Judea Pearl [16]. Prior probabilities on each F_i can be based on reliability data, and conditional probabilities $P(T_j|F_i)$ can be defined based on properties of the underlying test system [17].

B. Virtual Evidence

One of the issues associated with probabilistic diagnostic systems involves accounting for the uncertainty in the evidence collected (i.e., uncertainty in the test results). Two different formalisms have been defined to address evidence uncertainty in Bayesian networks: soft evidence and virtual evidence [18].

Soft evidence corresponds to the process of replacing the conditional probability $P(T_j|F_i)$ at the time an observation is made (i.e., the test is performed) to capture, directly, the confidence in the test result. Inference is then applied using the new distribution. More formally, if $P(T_j)$ reflects the probability of a test result we would derive that from the model by computing $P(T_j) = \sum_{\mathbf{T}\setminus T_j} P(\mathbf{D})$, meaning we

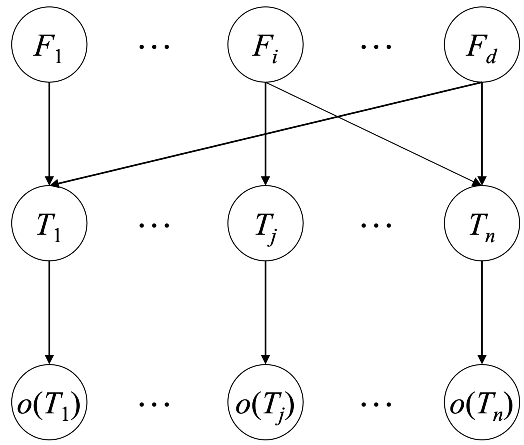

Fig. 1. Diagnostic Bayesian network with virtual evidence

marginalize out the rest of the network. With soft evidence, we replace $P(T_j)$ with a revised estimate $P'(T_j)$ and update using Jeffrey's rule:

$$P'(\mathbf{D}) = \sum_{T'_j} P(\mathbf{D}|T'_j)P(T'_j).$$

Virtual evidence, on the other hand, actually inserts an additional vertex into the model to reflect the confidence of the evidence, $P(obs(T_j)|T_j)$. This is reflected graphically as shown in Figure 1. In this case, we pre-set the test confidences through the definition of the observation distributions and apply the evidence to those vertices. We then infer the corresponding state of the fault vertices using the usual inference methods.

IV. Prognostic CTBNs

In the previous section, we spent time setting up tools for performing probabilistic fault diagnosis. This approach is beneficial since it allows us to take observation uncertainty, dependency uncertainty, and failure uncertainty into account in a unified way. As we will see, it also provide a way to specify the prior distribution $P_{\mathcal{X}}(0)$ for the CTBN that we will be using for prognosis. In this section, we discuss the way prognostic CTBNs are constructed.

A. Fault Trees

Within the automatic test systems community, many will have encountered the concept of a fault tree. The question facing us, however, is what kind of fault tree? In test program sets (TPS), a fault tree corresponds to the decision process of specifying a test, observing an outcome, and branching to the next step until a diagnosis or call out can be returned. However, an alternative form of fault tree arising from a process known as "Fault Tree Analysis" (FTA) [19].

A fault tree arising from FTA corresponds to a directed acyclic graph that satisfies the properties of a tree where edge directions all proceed upward, from leaf to root. The leaves of the tree correspond to faults in the system, which we obtained from $\mathbf{F}$. Interior vertices of the graph then correspond to

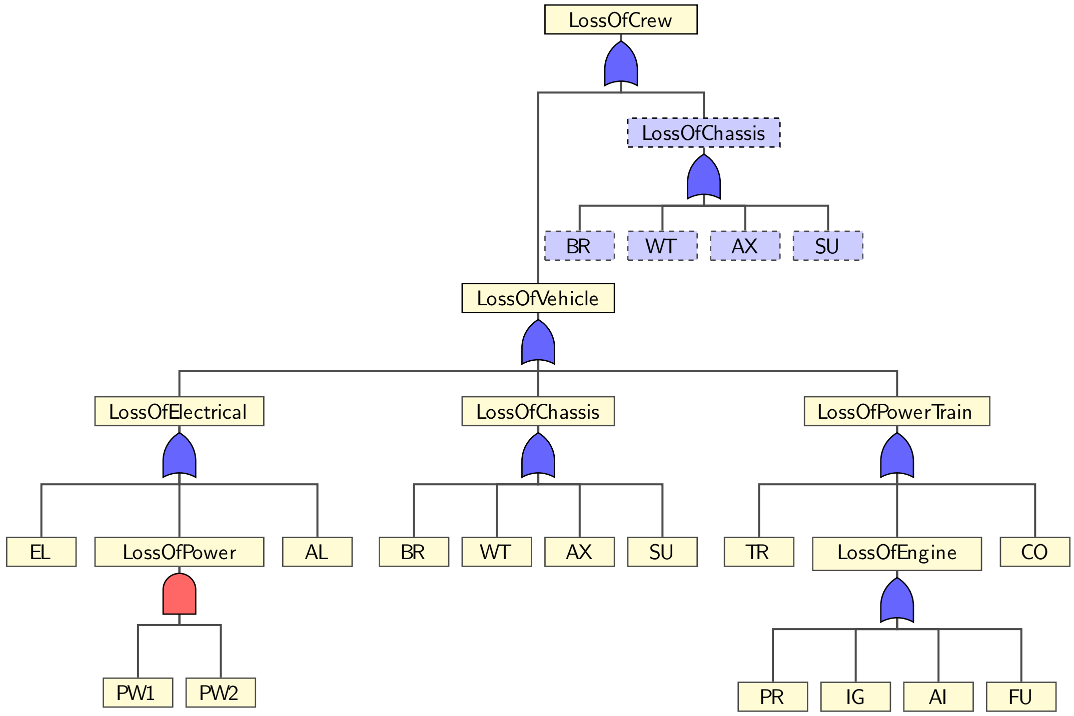

Fig. 2. Sample Fault Tree [4]

failures, effects, or hazards that might emerge as a result of a fault occurring. Thus the leaf states propagate upward through the tree. The interior vertices are then represented using logic gates (e.g., AND, OR, XOR) where the truth value indicates whether the corresponding effect is expected to occur as a consequence of fault(s) at the leaves of the tree. An example fault tree taken from [4] is shown in Figure 2.

Previously, Perrault *et al.* showed how to encode a fault tree as a CTBN [20]. To summarize, to parameterize fault nodes, we use an intensity matrix for fault F corresponding to

$$Q_F = \begin{array}{c} \\ f^0 \\ f^1 \end{array} \begin{array}{c} \begin{array}{cc} f^0 & f^1 \end{array} \\ \begin{pmatrix} -\lambda_f & \lambda_f \\ \mu_f & -\mu_f \end{pmatrix} \end{array}$$

where f^x indicates the logical state of the fault, λ_f is the failure rate of the fault, and μ_f is the repair rate of the fault. Without loss of generality, if we assume the interior nodes all have only two children, each of them requires two conditional intensity matrices. For the AND nodes, the intensity matrices correspond to

$$Q_{X|\mathrm{Pa}(X)} = \begin{array}{c} \\ x^0 \\ x^1 \end{array} \begin{array}{c} \begin{array}{cc} x^0 & x^1 \end{array} \\ \begin{pmatrix} -\lambda_X & \lambda_X \\ 0 & 0 \end{pmatrix} \end{array},$$

when $F_X(\mathrm{Pa}(X)) = 1$ (all ones) and

$$Q_{X|\mathrm{Pa}(X)} = \begin{array}{c} \\ x^0 \\ x^1 \end{array} \begin{array}{c} \begin{array}{cc} x^0 & x^1 \end{array} \\ \begin{pmatrix} 0 & 0 \\ \mu_{X|\mathrm{Pa}(X)} & -\mu_{X|\mathrm{Pa}(X)} \end{pmatrix} \end{array},$$

when $F_X(\mathrm{Pa}(X)) = 0$ (not all ones). On the other hand, for OR nodes, the intensity matrices correspond to

$$Q_{X|Pa(X)} = \begin{array}{c} \\ x^0 \\ x^1 \end{array} \begin{array}{c} \begin{array}{cc} x^0 & x^1 \end{array} \\ \begin{pmatrix} 0 & 0 \\ \mu_X & -\mu_X \end{pmatrix} \end{array},$$

when $F_X(\mathrm{Pa}(X)) = 0$ (all zeroes) and

$$Q_{X|\mathrm{Pa}(X)} = \begin{array}{c} \\ x^0 \\ x^1 \end{array} \begin{array}{c} \begin{array}{cc} x^0 & x^1 \end{array} \\ \begin{pmatrix} -\lambda_{X|\mathrm{Pa}(X)} & \lambda_{X|\mathrm{Pa}(X)} \\ 0 & 0 \end{pmatrix} \end{array},$$

when $F_X(\mathrm{Pa}(X)) = 1$ (not all zeroes).

B. Mitigation Strategies

When considering the risk-based approach to PHM, the intent is to be proactive in mitigating risks. This is generally captured by implementing condition-based maintenance strategies that perform system support *prior* to system failure, thereby mitigating the potential effects of failure occurring

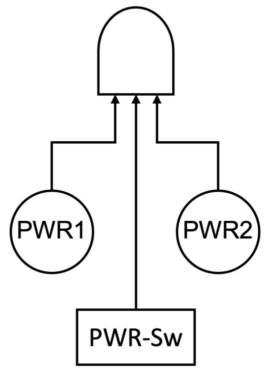

Fig. 3. Simple mitigation of power loss

during a mission. Within the context of a CTBN, mitigation strategies can be added in as model components.

To incorporate mitigation strategies, we first extend the basic CTBN model to incorporate decision nodes. Perreault referred to the resulting model as a Continuous Time *Decision* Network (CTDN) [4]. Specifically, the CTBN is augmented with two additional types of vertices—decision vertices and utility vertices. We will discuss utility vertices in the next subsection. For now, we focus on the decision vertices since those encode decisions or actions that serve as mitigation strategies.

Formally, Perreault defines a decision vertex as follows. "A continuous time decision node X is a special type of CTBN node that has no parents, and where the state of the process is known at all times $t = [0.0, \infty)$, thereby defining a local trajectory over the variable $\sigma[X]$. The states in the trajectory $\sigma[X]$ must conform to a possibly empty set of constraints $\mathbf{C}$, defining valid decisions for the system. Each constraint $c \in \mathbf{C}$ consists of a set of tuples $(t_s, t_e, \mathbf{Y})$ defining the set of possible states $\mathbf{Y}$ that may be assigned over the time interval $[t_s, t_e)$ [4]." Simply put, a decision vertex is a CTBN vertex where the state is predefined over the given time interval, thus forcing a particular child CIM to be activated.

An example mitigation strategy for Figure 2 is shown in Figure 3. In this example, two different power sources are available to to power the vehicle. The "PWR-Sw" decision node is used to switch between PWR1 and PWR2 based on the health of these two power sources by defining the CIM for the AND node to be conditioned on the state of the decision node and the power nodes.

C. Performance Functions

As mentioned in the previous section, another important component of the CTDN is the inclusion of utility nodes. In the context of the CTBN literature, these are defined via *performance functions* [21]. Utility nodes can be used to provide a numerical assessment of the quality of the mitigation strategy in order to trade options against each other. A performance function is represented with yet another vertex in the network; however, this vertex does not have a CIM associated with it. Rather, the vertex depends upon one or more CTMP vertices and defines a function based on trajectories defined over those CTMPs. More formally, let $\sigma[\mathbf{Y}]$ be a trajectory defined over a set of variables $\mathbf{Y} \subseteq \mathbf{X}$. Let $\langle t_s, t_e, \mathbf{Y}_t \rangle$ be a set of observations over these CTMPs. Then the performance function for $\mathbf{Y}$ can be defined as

$$f(\sigma) = \sum_{\langle t_s, t_e, \mathbf{Y}_t \rangle} f_{\mathbf{Y}}(t_s, t_e, \mathbf{Y}_t).$$

Note that this idea can be extended to include "factored" utility functions, more detail of which can be found in [22].

V. The PHM Process

Now that we have described the constituent elements, we are in a position to outline the process for performing risk-based PHM. For this discussion, we will use the diagram in Figure 4. As we describe the proposed process, we emphasize that this process is not employing on-board health monitoring but is depending upon test information collected from a TPS on an automatic test system (ATS). The intent is to collect health data, not only to perform fault isolation, but also to establish the state of health for the rest of the unit under test (UUT). Based on state of health, risk assessments can be made based on failure progression and mitigation/maintenance strategies assessed while the UUT is still under maintenance.

At the start of the rPHM process is the UUT. At this point, the UUT has already been pulled from the system and sent to be tested. The UUT is tested on an ATS, such as the US Navy's eCASS system and fault isolated. Once fault isolation is complete, the UUT is repaired and run through re-test to determine if it is able to be returned to service.

Following return to service testing, the test results are captured, perhaps in standard form [23] and provided to a separate diagnostic engine based on a Bayesian network derived from a D-matrix. The test results are furnished as virtual evidence to the Bayesian network to provide a means to better quantify the uncertainty of the health state. Note that "fault" conditions would need to be included in the Bayesian network to account for degraded states since simply relying on faulty/fault-free states will not provide sufficient granularity of health. For example, constructs based on fuzzy random variables could be used to capture degradation information in the model [24].

Once the health state is determined, the resulting information can then be provided to that CTDN that assesses potential hazards and mitigation strategies. A default mode where no mitigation is performed can be used to assess baseline performance using the utility nodes on the CTDN. If the utility is deemed to be too low, alternative mitigation strategies can be tested to assess changes from of utility. If it is determined that additional maintenance is warranted (for example), then information can be provided to technicians to take action, re-test, and re-assess health and failure progression.

VI. Next Steps

In this paper, we have described a process where by offboard diagnostics can be paired with a risk-based, probabilistic assessment of failure progression. The intent is to take advantage

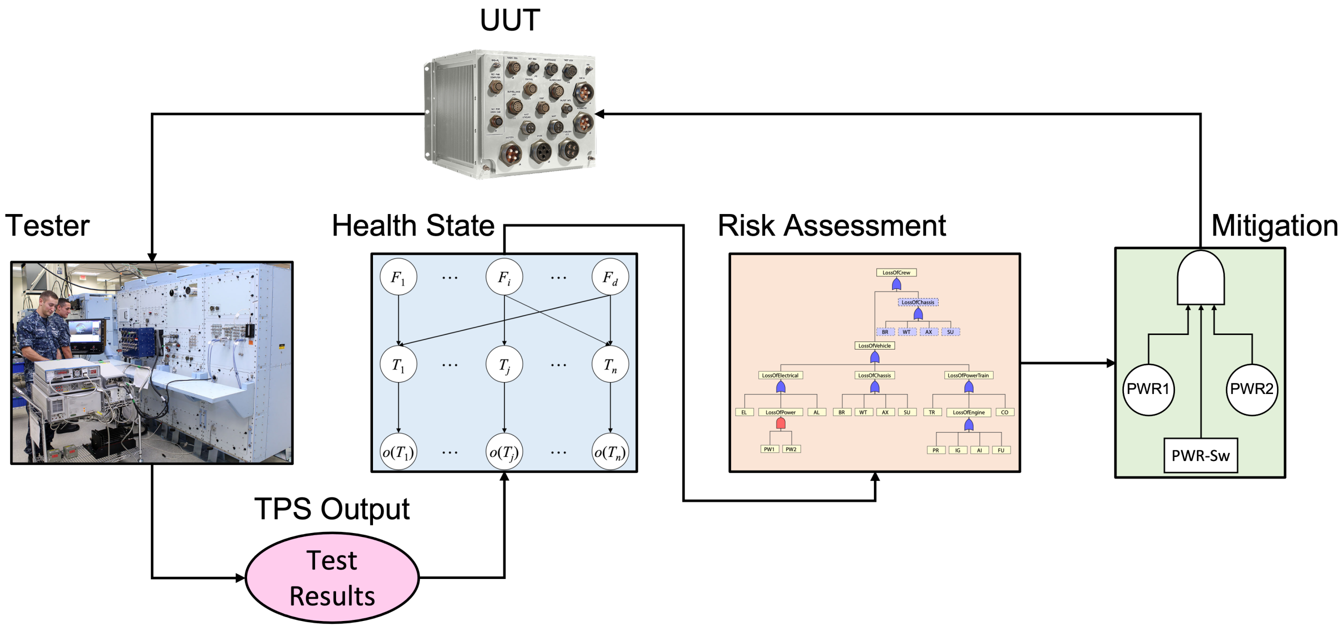

Fig. 4. A probabilistic risk-based PHM process

of existing maintenance practices to improve the overall supportability of a system through condition-based maintenance, accurate health assessment, probabilistic risk assessment, and risk-informed decision making.

To date, tools have been created for processing diagnostic Bayesian networks and prognostic Continuous Time Decision Networks. The Bayesian network has been implemented in the SAPPHIRE system, developed under support of the US Navy. Of particular note is that SAPPHIRE is designed to conform to the AI-ESTATE standard [2]; however, AI-ESTATE is overdue for update and re-approval. As a result, the first area of continued work is to update the standard to accomplish two things. First the data interchange format needs to be updated to use a more modern format such as JSON. Second, a model for CTBNs was proposed previously [25] but was never able to be considered for inclusion the standard. The update process would enable this model to be considered for standardization.

In addition to SAPPHIRE, the CHARM system was developed under support from NASA as a way to address risk-informed decision making for deep space missions. While a prototype system has been developed, it was written in a way that is not compatible with the current SAPPHIRE system, so it is in the process of being redeveloped and refactored to create an integrated system. In the process, inference algorithms are being benchmarked and optimized to improve computational performance [26].

Of particular importance in this effort is ensuring we are making maximum use of available standards. We are already employing the AI-ESTATE standard; however, the current implementations of SAPPHIRE and CHARM do not support either the SIMICA Test Results standard [23] or the SIMICA Maintenance Action Information Standard [27]. To support the process fully, to include things like model maturation [28], these standards need to be implement to capture the historical information necessary for identifying deficiencies in the model.

Finally, this work is being supported under a contract with the US Navy with close coordination with the F-35 Joint Program Office. This will enable the tools to be validated with actual systems currently being tested on Lockheed LM-STAR testers. Initial models are in the process of being constructed based on data collected from fault insertion on three F-35 UUTs.

Acknowledgments

This work is being supported under a Small Business Technology Transfer contract in collaboration with GSS LLC. We would like to thank several people affiliated with the US Navy and F-35 JPO include Michael Malesich, Jennifer Fernandi, and Anthony Conway. We would also like to thank the members of the Numerical Intelligent Systems Laboratory at Montana State University for their support, encouragement, and feedback during different points in the development of this work.

References

[1] L. Sturlaugson, N. Fortier, P. Donnelly, and J. W. Sheppard, "Implementing AI-ESTATE with prognostic extensions in java," in *IEEE AUTOTESTCON Conference Record*, 2013.

[2] "IEEE Standard for Artificial Intelligence Exchange and Service Tie to All Test Environments (AI-ESTATE)," *IEEE Std 1232-2010*.

[3] U. Nodelman, "Continous time Bayesian networks," Ph.D. dissertation, Department of Computer Science, Stanford University, 2007.

[4] L. Perreault, "On the usability of continuous time Bayesian networks: Improving scalability and expressiveness," Ph.D. dissertation, Gianforte School of Computing, Montana State University, 2017.

[5] N. M. Vichare and M. G. Pecht, "Prognostics and health management of electronics," *IEEE Transactions on Components and Packaging Technologies*, vol. 29, no. 3, pp. 222–229, March 2006.

[6] P. W. Kalgren, C. S. Byington, and M. J. Roemer, "Defining PHM, a lexical evolution of maintenance and logistics," in *IEEE AUTOTESTCON Conference Record*, 2006, pp. 353–358.

[7] R. Li, W. J. Verhagen, and R. Curran, "A systematic methodology for prognostic and health management system architecture definition," *Reliability Engineering System Safety*, vol. 193, p. 106598, 2020.

[8] "IEEE Standard Framework for Prognostics and Health Management of Electronic Systems," *IEEE Std 1856-2017*.

[9] W. R. Simpson and J. W. Sheppard, *System Test and Diagnosis*. Springer, 1994.

[10] D. Koller and N. Friedman, *Probabilistic Graphical Models: Principles and Techniques*. The MIT Press, 2009.

[11] S. Singh, K. Choi, A. Kodali, K. Pattipati, J. Sheppard, S. M. Namburu, S. Chigusa, D. V. Prokhorov, and L. Qiao, "Dynamic multiple fault diagnosis: Mathematical formulations and solution techniques," in *Proceedings of the 18th International Workshop on Principles of Diagnosis (DX-07)*, Nashville, TN, May 2007, pp. 383–390.

[12] J. W. Sheppard and S. G. W. Butcher, "A formal analysis of fault diagnosis with d-matrices," *Journal of Electronic Testing: Theory and Applications*, vol. 23, no. 4, pp. 309–322, 2007.

[13] R. Reiter, "A theory of diagnosis from first principles," *Artificial Intelligence*, vol. 32, pp. 57–95, 1987.

[14] E. Shortliffe, "Mycin: A knowledge-based computer program applied to infectious diseases," in *Proceedings of the Annual Symposium on Computer Applications in Medical Care*, October 1977, pp. 66–69.

[15] M. A. Shwe, B. Middleton, D. E. Heckerman, M. Henrion, E. J. Horvitz, H. P. Lehmann, and G. F. Cooper, "Probabilistic diagnosis using a reformulation of the INTERNIST-1/QMR knowledge base: I. the probabilistic model and inference algorithms," *Methods of Information in Medicine*, vol. 30, pp. 241–255, 1991.

[16] J. Pearl, *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann Publishers, 1988.

[17] J. W. Sheppard and M. A. Kaufman, "A bayesian approach to diagnosis and prognosis using built-in test," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 3, pp. 1003–1018, 2005.

[18] J. Bilmes, "On virtual evidence and soft evidence in bayesian networks," May 2004, University of Washington Electrical Engineering Technical Report UWEETR-2004-0016.

[19] L. Xing and S. V. Amari, "Fault tree analysis," in *Handbook of Performability Engineering*, K. B. Misra, Ed. London: Springer London, 2008, pp. 595–620.

[20] L. Perreault, M. Thornton, and J. W. Sheppard, "Deriving prognostic continuous time Bayesian networks from fault trees," in *Proceedings of the Annual Conference of the Prognostics and Health Management Society*, 2016.

[21] L. Sturlaugson, "Extensions to modeling and inference in continuous time Bayesian networks," Ph.D. dissertation, Department of Computer Science, Montana State University, 2014.

[22] L. Sturlaugson and J. Sheppard, "Factored performance functions with structural representation in continuous time bayesian networks," in *Proceedings of the International Florida Artificial Intelligent Research Society Conference*, 2014, pp. 512–517.

[23] "IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Test Results and Session Information via the eXtensible Markup Language (XML)," *IEEE Std 1636.1-2018*.

[24] N. Ryhajlo, L. Sturlaugson, and J. W. Sheppard, "Diagnostic bayesian networks with fuzzy evidence," in *IEEE AUTOTESTCON Conference Record*, 2013.

[25] "Using continuous-time Bayesian networks for standards-based diagnostics and prognostics," in *IEEE AUTOTESTCON Conference Record*, 2014.

[26] J. Schupbach, E. Pryor, K. Webster, and J. Sheppard, "Benchmarking continuous time bayesian networks for prognostic modeling," in *IEEE AUTOTESTCON Conference Record*, 2022.

[27] "IEEE Standard for Software Interface for Maintenance Information Collection and Analysis (SIMICA): Exchanging Maintenance Action Information via the Extensible Markup Language (XML)," *IEEE Std 1636.1-2018*.

[28] S. Strasser and J. Sheppard, "Diagnostic model maturation," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, no. 1, pp. 34–43, 2013.