

# Continuous Time Bayesian Networks in Prognosis and Health Management of Centrifugal Pumps

Tyler Forrester<sup>1</sup>, Mark Harris<sup>2</sup>, Jacob Senecal<sup>3</sup>, and John Sheppard<sup>4</sup>

<sup>1,2,3,4</sup> *Montana State University, Bozeman, Montana, 59717, United States*

*tylerforrester@montana.edu*

*markharris@montana.edu*

*jacobsenecal@montana.edu*

*john.sheppard@montana.edu*

## ABSTRACT

This paper presents a novel method for performing risk-based prognosis and health management (rPHM) on centrifugal pumps. We present the rPHM framework and apply common modeling tools used in reliability and testability analysis—dependency (D) matrices and fault tree analysis—as a basis for constructing an underlying predictive model. We then introduce the mathematics of the Continuous Time Bayesian Network (CTBN), which is a probabilistic graphical model based on a factored Markov process that is designed to capture system evolution through time, and we explain how to apply a CTBN derived from D-matrices and fault trees to consider the impact of a set of faults common to centrifugal pumps on emerging hazards in the pump system. We demonstrate the utility of using CTBNs for rPHM analysis with two experiments showing the descriptive power of our modeling approach.

## 1. INTRODUCTION

Centrifugal pumps have many applications including operations in municipal water supplies and waste management, irrigation for farming, hydrant systems and many other industrial uses. Many authors have studied the reliability of centrifugal pumps but have not looked, specifically, at prognosis and health management (PHM) for pump systems. Our contribution is in proposing a novel method to model this pump failure progression for purposes of supporting PHM on such systems. The method combines commonly used fault tree analysis (FTA) and the diagnostic dependency matrix (D-matrix) through the use of a stochastic model called a Continuous Time Bayesian Network (CTBN). Similar to Dynamic Bayesian Networks (DBN), the CTBN models state transitions through time. Unlike DBNs that require representing

time evolution in discrete time steps, the CTBN performs inference over state progressions that occur in continuous time.

The CTBN is a type of Continuous Time Markov Process (CTMP), where the Markov process is factored into underlying components or subsystems. The factoring process provides a compact representation for capturing failure semantics from FTA and from the diagnostic D-matrix. The D-matrix is a valuable addition to the model, allowing the application of uncertain evidence from test observations to resolve fault ambiguity groups. It also provides a means to incorporate and testing error rates (false alarms and non-detects) directly into the model.

Our paper explains the theory and methodology behind applying a CTBN to PHM with a focus on centrifugal pumps. We introduce a novel fault tree and D-matrix for a centrifugal pump in the context of a CTBN. We perform inference on the resulting CTBN using the common inference approximation technique, importance sampling. Importance sampling is a variation of a commonly used forward sampling procedure that enables the incorporation of evidence into the model of different types and durations. The applied evidence is observed from performing maintenance inspections or test protocols to the pump to determine its current health state, so by applying evidence to the system, planners can model the risks associated with potentially emerging failures or different maintenance plans.

In general, exact inference with CTBNs is intractable, thus generally requiring approximate inference (U. D. Nodelman, 2007). In fact, it has been proven that both exact inference and approximate inference are NP-hard with CTBNs (Sturlaugson & Sheppard, 2014). For this reason, we apply importance sampling and discuss how the CTBN models risk and how the model can be used by planners to improve both uptime and risk mitigation for the centrifugal pumps.

---

Tyler Forrester et al. This is an open-access article distributed under the terms of the Creative Commons Attribution 3.0 United States License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

## 2. BACKGROUND

Centrifugal pumps are an integral part of modern day infrastructure, providing basic functions in municipal water supplies, sewage treatment, irrigation, heating and cooling systems, along with many industrial applications (Behrends, Houke, Bailey, Jansen, & Brown, 2001; Gülich, 2008). Identification and prediction of failures is of major concern to practitioners who strive to make systems both more efficient and cost effective. Toward this end, the framework of risk-based prognostics and health management (rPHM) has been proposed to manage the risk of these failures effectively (J. Sheppard, Perrault, & Forrester, 2019). We model the effects of an example centrifugal pump system using a CTBN, which manages the combinatorial explosion of relationships between faults and effects by assuming conditional independencies throughout the system. Our model differs from a Bayesian network (Pearl, 1988) in that conditional dependencies are now modeled through time. Our model also differs from a Dynamic Bayesian Network (DBN) (Murphy, 2002) because the model uses a Continuous Time Markov Process (CTMP) as the underlying stochastic process, as opposed to the DBN discrete step process (Howard, 1960). The continuous time model also relaxes the directed acyclic requirement of the traditional Bayesian network.

### 2.1. Risk-Based PHM

PHM is characterized by monitoring system health and degradation, as well as forecasting future failures and associated maintenance problems (Pecht, 2008). Historically a forecast (prognosis) would be given by a domain expert. Domain expert-based models, however, present several drawbacks. The first is that domain expert knowledge and experience can vary greatly between experts, causing the accuracy their prognosis also to vary widely. Secondly, by using a domain expert, prognoses are difficult to replicate with many aspects of the process being a black-box combination of expertise and intuition.

To be able to standardize PHM across sites, physics-based models are a better choice. Such an approach removes the variability of the human experience by replacing it with a standardized model used in all locations. This standardized model is more effectively testable. All decisions are made through the same process, so a failure in a piece of equipment can then be investigated with the certainty that the historical prognosis was consistent for the same failure across the organization. Unfortunately, such models rarely scale to real-world systems due to the computational complexity associated with processing them (Kim, An, & Choi, 2017).

Due to the complexities associated with creating, validating, and using physics-based models in real-world systems, alternative methods based on data-driven processes have been proposed. These methods often employ techniques

from machine learning to capture regularities in historical data. A common data-driven approach is based on sparse Bayesian learning, incorporating so-called “relevance vector machines” (Tipping, 2001). Alternative Bayesian approaches incorporate particle filtering as a way of performing inference over these models to handle some of the inefficiencies associated with exact inference (Goebel, Saha, Saxena, Celaya, & Christophersen, 2008; Orchard, Kacprzyński, Goebel, Saha, & Vachtsevanos, 2008; Lall, Lowe, & Goebel, 2012).

We have developed a model that attempts to exploit the advantages of both physics-based models and data-driven models while minimizing the disadvantages. To that end, we employ methods from probabilistic graphical models, namely the CTBN that incorporate information from engineering models (fault trees and D-matrices) while also supporting learning from historical data when available. We also support methods of applying uncertain evidence as a means of incorporating information provided from an external (e.g., physics-based) model as input to our CTBN (Sturlaugson & Sheppard, 2016; J. W. Sheppard, Gorton, Kalgren, & Sturlaugson, 2018).

As a discipline, PHM focuses on prediction and prevention of faults and failures. However, not all faults and failures have equal risk. For example, we modeled the cascade distillation system for the water recovery unit on board the International Space Station. The system can fail in a variety of ways, including a motor failure in the centrifuge or a valve failure in the chemical processing plant. The risk associated with a motor failure is substantially lower than a valve failure since motor failure delays the production of potable water. A valve failure, on the other hand, could lead to toxic chemicals being released into the closed atmosphere of the station, thus resulting in the immediate loss of the crew (Ghoshal & Sheppard, 2016). To address this inequality in the effects of failures, we use rPHM. In rPHM, we extend the PHM model to include risks associated with failures to more efficiently allocate resources toward prevention.

The approach of incorporating risk into the predictive process is already being used by the United States Nuclear Regulatory Commission (USNCR), which uses the term “risk-informed decision making.” USNCR implements this through Systems Analysis Programs for Hands-on Integrated Reliability Evaluations (SAPHIRE), a system that uses minimal cut fault trees to analyze the relationship of effects to failures (Smith, Knudsen, Kvarfordt, & Wood, 2008). This approach, however, involves a static analysis and is not able to adapt readily to evidence gained from observation and testing. The CTBN-based approach, on the other hand, is able to do both, as well as incorporate utility functions and decision points, supporting optimization of work plans (Perreault, Thornton, & Sheppard, 2016b; Sturlaugson, Perreault, & Sheppard, 2017).

## 2.2. D-Matrices

The first step in any fault analysis is collecting data on the health of various components in the system. However, we cannot generally observe the state of a component directly, so we must devise a series of tests whose outcomes give a probability of a fault. In this framework, the component is called the unit under test (UUT) and has a series of tests  $t \in \mathbf{T}$  associated to the UUT that test for faults  $f \in \mathbf{F}$ . More formally the relationships between faults  $\mathbf{F}$  and tests  $\mathbf{T}$  are represented by a  $|\mathbf{T}| \times |\mathbf{F}|$  matrix  $\mathbf{D}$ , where  $\mathbf{D} = \{d : \mathbb{1}_{t \text{ tests for } f}(t, f) \quad \forall t \in \mathbf{T}, \forall f \in \mathbf{F}\}$ . We call  $\mathbf{D}$  a D-matrix. Since the fault's state changes the probability of a test outcome, we consider the test-to-fault relationship as corresponding to a conditional probability density,  $P(t_i|\mathbf{F})$ , though these densities are generally unknown in the model (Simpson & Sheppard, 1994).

## 2.3. Fault Trees

Although many other methods exist for analyzing the effects of failure or faults in a system, such as Fault Isolation Manuals (FIM), Failure Modes Effects and Criticality Analysis (FMECA), and Fault Detection and Isolation (FDI) systems (Kallesøe, 2005), this paper combines D-matrices with the results of performing fault tree analysis (FTA), (Watson, 1961). FTA is a common method in the reliability literature to encode the relationships between component faults and system failures. The fault tree is a directed acyclic graph whose leaves are the set of faults and interior vertices are the set of failures. The edges feed into logic gates, *AND* and *OR*, which are placed between vertices to indicate causal relationships.

While this structure allows for the same component to appear in many locations in the tree so as to minimize complexity, these are sometimes removed by using the minimal cut set of the fault tree. In our method, however, we eliminate redundancies but keep the hierarchical structure to facilitate tracking intermediate failures (J. Sheppard et al., 2019).

Additionally, we assume that the intermediate “failures” in the fault tree correspond to hazards associated with dependent faults occurring. Going forward, we use the notion of a failure and a hazard interchangeably.

## 3. PUMP FAULTS AND FAILURES

In this section, we introduce the pump model we have developed to illustrate the rPHM process. Figure 1 shows the top level portion of the fault tree we developed mapping pump faults to system failures. As the figure indicates, we have divided the pump complex into two assemblies—the motor and the pump. The motor has two subassemblies—electrical and mechanical. The pump has a shaft, an impeller, and mechanical subassemblies. With each subassembly, we have identi-

fied common faults and their associated mean times between failure (MTBF), which we estimated in hours. We have also estimated mean repair times and associated costs. Both mean repair times and MTBFs are treated as the parameters for exponentially distributed random variables. We chose to use a noisy-OR construct where, if any fault occurs, the subassembly fails, and if any subassembly fails, the main assembly fails (Perreault, Thornton, Sheppard, & DeBruycker, 2017). All faults and effects were adapted from (Kallesøe, 2005). Rates were taken, when available, from (Mays, 1989). Costs and unknown rates were estimated in the best judgment of the authors.

The model faults are described below with complete descriptions of rates and costs in Table 1. Since many faults are unable to be tested directly, we have also formulated a series of tests with estimated false alarm and non-detect rates in Table 2. A description of the tests with non-detect and false alarm rates are available here, and the associated D-matrix is shown in Table 3.

### 3.1. Motor

Kliman *et al.* estimate that induction motor faults are comprised of bearing faults (40%-50%), stator faults (30%-40%), and rotor faults (5%-10%), with the remaining faults accounting for only a small percentage of total faults (Kliman, Premerlani, Kogel, & Hoeweler, 1996). Both stator faults and rotor faults are considered electrical faults.

We have included five electrical faults in the motor: loss of supply voltage, short circuit between windings, short circuit to ground, broken rotor bar, and air grab due to motor shaft issues. Electrical issues are detected through changes in torque oscillation, pack transform current, and in the case of a short circuit, the motor temperature.

We model two mechanical faults: wear of bearings in the motor and rub impact between stator and rotor. The bearing faults are tested in the same way as the short circuits by monitoring changes in the pack transform current, the torque oscillations and the engine temperature. The rub impact between stator and rotor is also detected by changes in the harmonic oscillations.

### 3.2. Pump

We divide the pump into three subassemblies: impeller, mechanical, and shaft. Common shaft problems involve a bent or broken shaft or a shaft misalignment between the motor and the pump. Faults on the shaft are detected through changes in harmonic vibrations and changes in torque oscillation. Common impeller faults include impurities in the water ranging from sand to larger obstacles, dry running, and wear or complete loss of the impeller. The faults are detected through

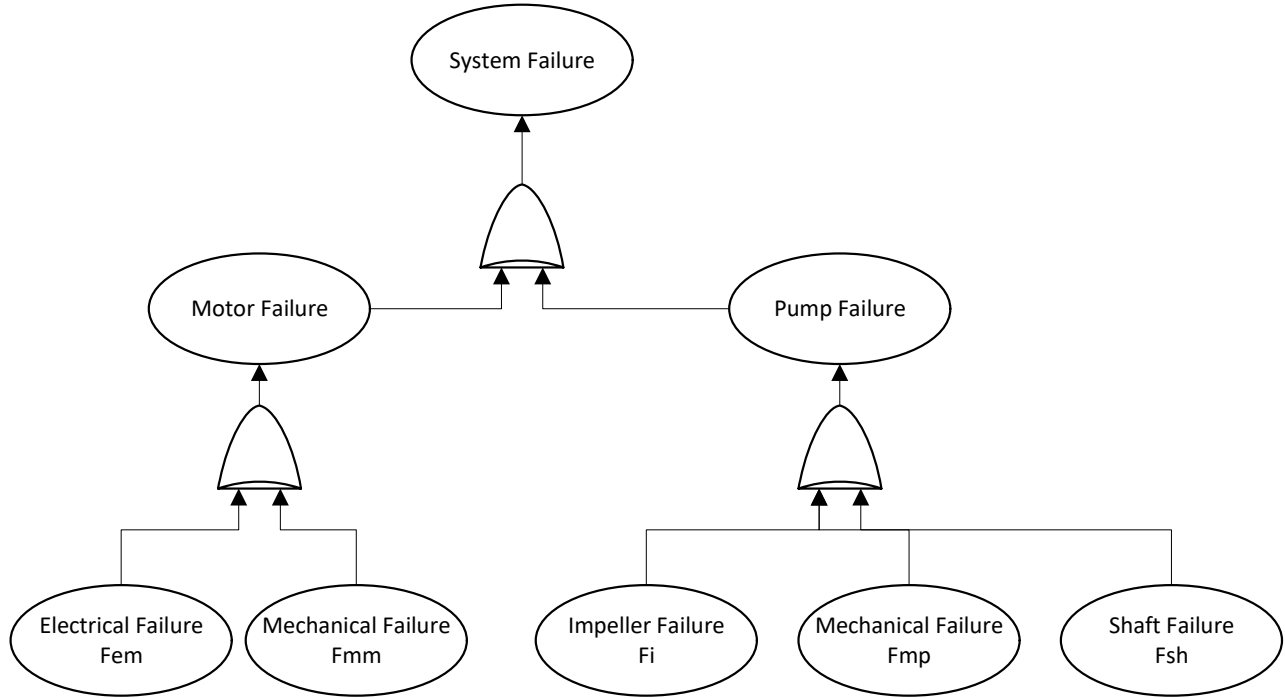


Figure 1. Fault Tree with High-Level Faults

Table 1. Faults with associated failure rates and repair rates and costs.

Faults	Part Name	MTBF(hrs)	MTTR(hrs)	Repair Costs
Fem1	Loss of supply voltage	183,000	2	1000
Fem2	Short circuit between windings	746,000	2	1000
Fem3	Short circuit to ground	70,100	2	1000
Fem4	Broken rotor bar	701,280	8	5000
Fem5	Air grab due to motor shaft issue	600,000	8	5000
Fmm1	Wear of the bearings in the motor	11,000	6	2500
Fmm2	Rub impact between stator and rotor	8,015,000	8	4000
Fsh1	Bent shaft	36,000	12	3200
Fsh2	Misalignment between the motor and the pump	679,000	1.5	800
Fi1	Dry running	32,500	1	400
Fi2	Impurities fixed on the impeller	205,800	4	400
Fi3	Wear of the impeller	205,800	8	1800
Fi4	Large obstacles in the liquid	205,800	4	400
Fi5	Sand and other impurities in the water	205,800	4	400
Fi6	Missing sealing ring	205,800	2	250
Fi7	Loss of the impeller	205,800	8	1800
Fmp1	Dry running	60,400	1	100
Fmp2	Inlet flow equal to zero	60,400	1	100
Fmp3	Wear of the bearings in the pump	28,700	6	2600
Fmp4	Wear of seals	28,200	2	250
Fmp5	Rub impact between the impeller and the casing	135,900	8	9800

changes in pressure or pressure harmonics, impeller torque, motor torque oscillations and harmonic vibrations.

The mechanical faults in the pump are wear of the seal and

bearings, no inlet flow, and rub impact between the impeller and the casing. In the case of a seal fault, the discerning test is leakage flow from the pump. In the case of dry running or

Table 2. Tests Abbreviations, Descriptions with No-Detect and False Alarm rates

Id	Full Name	Description	ND	FA
TOT	Torque Oscillations Test	Changes in harmonic torque oscillations	.02	.03
PT	Pressure Test	Checks system pressure	.01	.008
ITS	Impeller torque sensor	Monitors the torque load from the impeller	.02	.01
HPT	Harmonic Pressure Test	Tests for harmonic pressure oscillations	.005	.01
HVT	Harmonic Vibrations Test	Checks harmonic vibrations of the pump	.02	.015
BT	Bearing Temperature Test	Monitors bearing temperature	.005	.008
MTG	Motor temperature gauge	Monitors motor temperature	.008	.02
LT	Leak test	Tests leakage flow from pump	.03	.01
PTCA	FFT pack transform current	Monitors oscillations in the current length	.004	.03
SCG	Stator current gauge	Checks stator current balance	.005	.03
SPT	Shaft speed test	Checks shaft speed	.01	.004

Table 3. D-Matrix Mapping Tests to Faults

	TOT	PT	ITS	HPT	HVT	BT	MTG	LT	PTCA	SCG	SPT
Fem1	0	0	0	0	0	0	0	0	1	0	0
Fem2	0	0	0	0	0	0	1	0	1	0	0
Fem3	1	0	0	0	0	0	1	0	1	1	0
Fem4	1	0	0	0	0	0	0	0	1	0	0
Fem5	1	0	0	0	0	0	0	0	0	1	0
Fi1	0	1	1	0	0	0	0	0	0	0	0
Fi2	1	0	0	1	1	0	0	0	0	0	0
Fi3	1	1	1	1	1	0	0	0	0	0	0
Fi4	1	1	1	1	1	0	0	0	0	0	0
Fi5	0	1	1	0	0	0	0	0	0	0	0
Fi6	0	1	1	0	0	0	0	0	0	0	0
Fi7	0	1	1	0	0	0	0	0	0	0	0
Fmm1	1	0	0	0	0	0	1	0	1	0	0
Fmm2	1	0	0	0	1	0	0	0	0	0	0
Fmp1	0	0	0	0	0	1	0	0	0	0	0
Fmp2	0	0	0	0	0	1	0	0	0	0	0
Fmp3	1	0	0	0	1	1	0	0	0	0	0
Fmp4	0	0	0	0	0	0	0	1	0	0	0
Fmp5	1	0	0	0	1	0	0	0	0	0	0
Fsh1	1	0	0	0	1	0	0	0	0	0	0
Fsh2	1	0	0	0	1	0	0	0	0	0	0

no inlet flow, the bearing temperature can be tested. Worn-out bearings can also be tested by bearing temperature and through the harmonic vibrations test or torque oscillation test. The rub impact between impeller and the casing is tested by torque oscillation and harmonic vibrations analysis.

#### 4. MODELING SYSTEM EVOLUTION

In the following, we provide background information on the main model type for our application. We introduce the basics from the perspective of a CTMP and explain how the CTMP can be factored to yield a CTBN. We then explain the basic inference process employed.

#### 4.1. Continuous-Time Markov Process (CTMP)

To begin to understand the CTBN, we introduce the Continuous Time Markov Process (CTMP). A CTMP over some variable  $X$  with  $n$  discrete states is defined by a row vector initial distribution at time,  $t = 0$ , namely  $P_X^0$ , and an intensity matrix  $\mathbf{Q}$  (also called a infinitesimal generator) representing the transitions between states.

$$\mathbf{Q}_X = \begin{bmatrix} -q_{x_1} & q_{x_{12}} & \cdots & q_{x_{1n}} \\ q_{x_{21}} & -q_{x_2} & \cdots & q_{x_{2n}} \\ \vdots & \vdots & \ddots & \vdots \\ q_{x_{n1}} & q_{x_{n2}} & \cdots & -q_{x_n} \end{bmatrix}$$

The amount of time that the process stays in some state  $x_i$  is distributed exponentially with parameter  $q_{x_i}$  with a corresponding expected time to transition of  $\frac{1}{q_{x_i}}$ . A multinomial distribution can then be parameterized with  $\frac{q_{x_{i,j}}}{q_{x_i}}$  representing the process of transitioning from state  $x_i$  to state  $x_j$ . The distribution of the CTMP by some time  $t$  is then given as

$$P_X(t) = P_X^0 \exp(\mathbf{Q}_X t)$$

where  $\exp(\cdot)$  is the matrix exponential. The sufficient statistics for a CTMP are the amount of time that a process variable stays in a state  $x$ , commonly denoted by  $T[x]$ , and the number of times that  $X$  transitions from state  $x$  to state  $x'$ , denoted  $M[x, x']$  (U. Nodelman, Shelton, & Koller, 2002a).

#### 4.2. Continuous Time Bayesian Network

A CTBN is a compact graphical representation of a factored CTMP. More formally, a CTBN is faithful to a probability distribution over trajectories  $\sigma$ , of a set of random process variables  $\mathbf{X}$ , which are distributed according to a conditional CTMP. A trajectory represents a continuous sequence of state changes of  $\mathbf{X}$ , recording both the state change and duration within a state. A conditional Markov process is an inhomogeneous Markov process whose distribution varies based on the set of instantiated values of the conditioning process variables  $\mathbf{U}$ . The distribution is represented by an intensity matrix, now called a conditional intensity matrix (CIM), where each set of instantiated values has its own CIM  $\mathbf{Q}_{\mathbf{x}|\mathbf{U}}$  (U. Nodelman et al., 2002a; Saria, Nodelman, & Koller, 2012).

We can now define a CTBN as an initial distribution based on a Bayesian network  $\mathcal{B}^0$  over  $\mathbf{X}$  with continuous-time transition model represented by a directed graph  $\mathcal{G}$  whose nodes represent random process variables with a node's parents being the set of conditioning process variables  $\mathbf{U}$ . Each variable has a CIM associated with it (U. Nodelman et al., 2002a; Saria et al., 2012).

More succinctly, given a set  $\mathbf{X}$  of discrete random variables  $X_1, \dots, X_N$ , a CTBN over  $\mathbf{X}$  can be described with an initial distribution  $P_X^0$ , which is a Bayesian network, a directed

(possibly) cyclic graph  $\mathcal{G}$ , whose nodes represent  $\mathbf{X}$ , and a set of parent random variables  $Pa(X_i)$  that dictates the corresponding set of intensity matrices, and finally a set of conditional intensity matrices, denoted  $\mathbf{Q}_{\mathbf{x}|\mathbf{U}}$  for each discrete random variable ( $\mathbf{U}$ . Nodelman et al., 2002a).

#### 4.3. Inference

Inference is process of estimating the state of the stochastic process as it evolves through time based on the observation of an instantiated partial trajectory. The two main applications of inference in CTBNs are estimating posterior densities over a given time interval for a known parameterization of the CTBN or learning the parameterization of a CTBN from data (Fan, Xu, & Shelton, 2010). Our focus here is the former.

Exact inference in CTBNs is intractable since it requires ‘‘amalgamating’’ the CIMs into a single intensity matrix which describes the CTMP. In fact, approximate inference is also intractable (Sturlaugson & Sheppard, 2014). Even so, multiple efficient ‘‘approximate’’ inference methods have been proposed that prevent the need for amalgamation (U. Nodelman et al., 2002a). One common form of approximate inference is importance sampling, which we discuss below.

##### 4.3.1. Importance Sampling

Suppose we want to know  $\theta = \mathbb{E}_f[h(\mathbf{x})]$ , where  $\mathbf{x}$  is a high dimensional random vector that is difficult to simulate, or alternatively where the variance of the density  $h(\mathbf{x})$  is high. One method we could use is to calculate

$$\theta = \mathbb{E}_g \left[ \frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})} \right]$$

with the restriction that  $f(\mathbf{x}) = 0$  wherever  $g(\mathbf{x}) = 0$ . We generate successive random vectors,  $\mathbf{x}_i$  from  $g(\cdot)$  and then use  $\frac{1}{n} \sum_{i=1}^n \frac{h(\mathbf{x}_i)f(\mathbf{x}_i)}{g(\mathbf{x}_i)}$  to estimate  $\theta$ . If the random variable,  $\frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}$  has low variance, then this results in an efficient estimator,  $\hat{\theta}$ . This method is called importance sampling (Ross, 1990).

Noting that  $f(\mathbf{x}_i)$  and  $g(\mathbf{x}_i)$  are the probability of  $\mathbf{x}_i$  occurring according to their respective densities, it is usual that  $f(\cdot)$  is small where  $g(\cdot)$  is big, which implies that the likelihood ratio  $\frac{f(\mathbf{x})}{g(\mathbf{x})}$  is often small. However, we need for  $\mathbb{E}_g \left[ \frac{f(\mathbf{x})}{g(\mathbf{x})} \right] = 1$ , so the likelihood ratio must sometimes be very large and the variance of  $\frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}$  will often be large, unless we pick  $g(\cdot)$  where likelihood is large and  $h(\cdot)$  is small. This will stabilize  $\frac{h(\mathbf{x})f(\mathbf{x})}{g(\mathbf{x})}$  (Ross, 1990).

### 4.3.2. CTBN Importance Sampling

Importance sampling in the context of the CTBN is a method to incorporate evidence into the model. Evidence corresponds to observations of the state of the system and inferring how these observations modify beliefs in the resulting trajectories. Two basic observations are the state at a specific point in time (point evidence) and observing the duration that the system was in the state (continuous-time evidence). Evidence has also been extended to include observations where state cannot be observed directly and therefore probability densities are defined over the states and over the duration<sup>1</sup> (Sturlaugson & Sheppard, 2016).

Originally proposed by (Fan & Shelton, 2008), importance sampling modifies a forward sampling procedure proposed by (U. Nodelman, Shelton, & Koller, 2002b) to incorporate the application of evidence of different durations and at different intervals. The method generates samples that conform to the observed evidence. Since the observed evidence modifies the state space, we correct for this modification of the sampling space by using likelihood weighting. The prediction is the likelihood weighted average of the samples. A more thorough treatment can be found in (Fan et al., 2010) and (Sturlaugson & Sheppard, 2016).

## 5. CTBN REPRESENTATION OF PUMP RELIABILITY

In this section, we walk through the process of converting the information about a pump, captured in terms of D-matrices and fault trees, into a corresponding CTBN.

### 5.1. Mapping a D-matrix to a CTBN

Since the CTBN is focused on failure progression, we first need to understand the interaction between tests and faults. We map a D-matrix to a CTBN and then perform inference. Each fault is an independent random variable in our model, and a test random variable is conditionally dependent on the faults that it monitors. Creating a CTBN from a D-matrix is straightforward and capitalizes on the performance gains of the independent graphical model representation. Details of the conversion process can be found in (Perreault, Thornton, Strasser, & Sheppard, 2015). Other graphical model representations have similar performance gains, but since the fault events are also dependent on time with no time step appropriate for all faults, we believe the CTBN with its associated conditional CTMPs to be a more accurate representation.

We build the CTBN graph,  $\mathcal{G}$ , so that its vertices correspond to  $\mathbf{T}$  and  $\mathbf{F}$  with a directed edge  $(f_i, t_j)$  if and only if  $t_j$  tests for fault  $f_i$ . The conditional intensity matrices,  $\mathbf{Q}_{t_j|\mathbf{F}_j}$  are then determined by the state of the faults  $\mathbf{F}_j$  monitored by  $t_j$ . In most scenarios, the number of discrete states for both tests

and faults is small, which moves tractability of this problem to depend upon the number of intensity matrices. Each fault has exactly one intensity matrix as it has no parents; however, for tests, an intensity matrix is needed for each state of each fault it detects, giving an upper bound on time complexity of  $O(c^m)$  where  $c$  is the maximum number of states of the faults the test detects, and  $m$  is number of faults that the test detects. This demonstrates the importance of unit tests or tests that detect faults on a small number of components, (Perreault, Thornton, Strasser, & Sheppard, 2015). An example conversion of the D-matrix from Table 3 to CTBN is shown in Figure 2.

### 5.2. Mapping a Fault Tree to a CTBN

We view failures (or hazards) as being conditioned upon faults, such that the joint distribution of failures is decomposed into factors, where each factor is independent of its nondescendants in the fault tree given its parents. Since the faults are the leaves of the fault tree, they become the parents of the failures and as such are parameterized using unconditional intensity matrices, i.e. homogeneous intensity matrices. These are the same unconditional intensity matrices used for the D-matrix conversion.

The failures, which have both faults or other failures in its parent set, are considered conditionally dependent on its parents, and a failure will have an intensity matrix that is conditional on the instantiation of the set of parents. The intensity matrices in the binary case will represent known failure and repair rates, which can be found easily or estimated from historical data. The failure rate,  $\lambda$ , differs from mean time between failures (MTBF) and, in our model, is the parameter of the exponential density indicating the probability that the failure will occur. If a failure occurs, then the repair rate,  $\mu$ , parameterizes an exponential density indicating the probability at which time the failure will be fixed. The repair rate can be omitted if the failure is unrepairable (Perreault, Thornton, & Sheppard, 2016a). An example conversion of the fault tree in Figure 1 is shown in Figure 3.

One of the obvious limitations of the CTBN model is the reliance on the exponential distribution when we know that very few components are truly Markovian. One successful approach to handle this problem is to relax this restriction in Markov processes into a semi-Markov process using a phase-type distribution. This relaxation using a phase-type distribution is also possible for the CTBN, and the process has been demonstrated with both Weibull and log-normal failure distributions (Perreault, Thornton, Goodman, & Sheppard, 2015). The CTBN can then be used to approximate non-exponential distributions that provide similar modeling flexibility to semi-Markov processes.

<sup>1</sup>There is also negative evidence about a state in which the system cannot be, but this is outside the scope of this paper

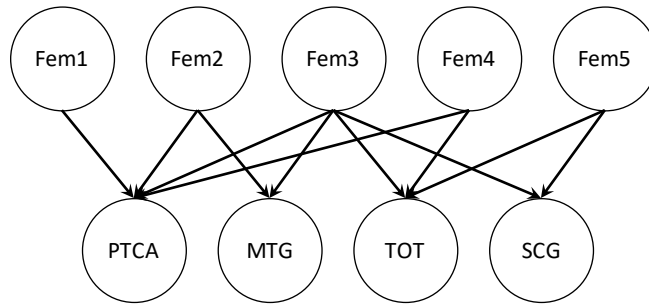


Figure 2. Example CTBN Structure for Fault to Test Relationships

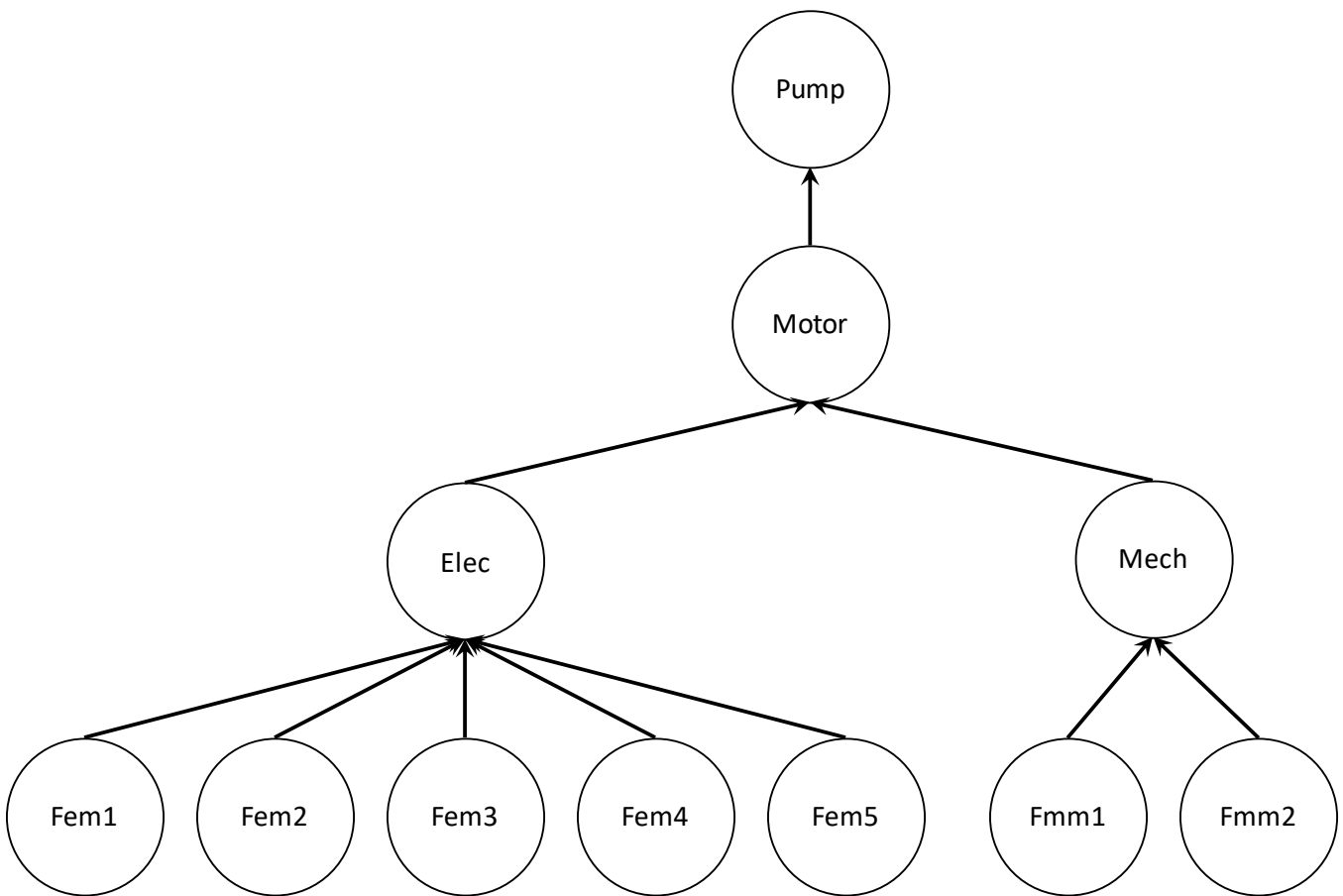


Figure 3. CTBN parameterization of a Fault Tree Subsystem



Table 4. Predicted Uptimes and Failures per 10,000 hours

Component	Uptime	Failures
<b>system</b>	0.998402	3.1305
<b>pump</b>	0.999013	1.9845
<b>motor</b>	0.998774	1.1488
fem*	0.996600	0.2416
fem1	0.994869	0.0567
fem2	0.988081	0.0119
fem3	0.996640	0.142
fem4	0.992915	0.0138
fem5	0.994878	0.0172
fmm*	0.998839	0.9074
fmm1	0.998841	0.9062
fmm2	0.994301	0.0012
fsh*	0.999563	0.2833
fsh1	0.999565	0.2688
fsh2	0.983878	0.0145
fi*	0.999470	0.5917
fi1	0.999614	0.3069
fi2	0.996803	0.0465
fi3	0.998068	0.0483
fi4	0.998428	0.0495
fi5	0.998115	0.0511
fi6	0.999475	0.045
fi7	0.999444	0.0445
fmp*	0.999474	1.1108
fmp1	0.999484	0.1667
fmp2	0.998290	0.1656
fmp3	0.999264	0.3497
fmp4	0.999780	0.3495
fmp5	0.999455	0.0799

## 6. EXPERIMENTS

In the following selection, we present experiments to demonstrate the utility of using a CTBN for risk-based PHM.

The first experiment verifies that the CTBN can correctly approximate the rates given by a traditional exponential parameter-based reliability analysis. Using the rates described above to estimate MTBF of the individual components by simulating the components trajectories using the importance sampling algorithm proposed by (Fan & Shelton, 2008). A Monte Carlo simulation of 1000 trials using the disjunctive CTBN model (Perreault et al., 2017) was performed. Each trial sampled failure time for each of the five components of the electrical system and recorded the first failure. This failure estimated the first time to failure of the motor electrical subassembly.

Using the properties of the order statistics of the exponential distribution, we know that if each individual component failure is independently exponentially distributed, then the system will fail on average every 41,300 hours. The average failure time in the Monte Carlo simulation was 41,700 hours, which is consistent with the underlying assumptions of the model. The Monte Carlo simulation also predicts the frequency of different types of faults and the corresponding

average cost of repair for the subsystem. The expected cost of the repair of the subassembly is 1524 units with an expected repair time of 2.7 hours. We can see the breakdown of the frequency of component faults in Figure 4.

In the second experiment, we model uptime and failures per 10,000 hours for entire system. We performed 10,000,000 runs of 10,000 hour simulation averaging uptime, failures and expected cost of the failures. The experiments were performed again using the noisy-OR model and the CTBN importance sampling algorithm. We then use the repair cost information from Table 1 as a means of estimating the expected repair cost over the 10,000 hour period.

## 7. RESULTS AND ANALYSIS

As we can see in Figure 5a, the cost of maintaining the pump is roughly 60% more than maintaining the motor. If we dive slightly deeper into the model in Figures 5b, 5c, 5d, we can see that the most costly faults are occurring in the mechanical subassembly of the pump, particularly, the rub impacted between the impeller and casing and the wear of bearings in the pump. The rub impacted between the impeller and casing is costly, because the entire assembly needs to be replaced and the wear of bearings in the pump is commonly repaired through pump replacement. In this analysis, by performing preventative maintenance on both the bearings and the impeller, we can reduce our unexpected costs in the pump. The next important fault to address is the broken shaft, while impeller problems, though common, contribute less to the cost of running the pump.

Results for each component, subassembly, assembly, and system are available in Table 4. Overall we expect the system to exhibit 3.13 failures per 10,000 hours, with 1.98 in the pump assembly and 1.15 in the motor assembly; although, the motor assembly has slightly lower uptime because of longer repair times in the assembly. If we evaluated this without cost associated, we see the mechanical pump subassembly is still a common source of faults with 1.11 faults. A close second is the motor mechanical subassembly with 0.9 faults per 10,000 hours. To increase uptime, improving performance of these two subassemblies would be an appropriate focus.

## 8. CONCLUSION

The CTBN model is capable of modeling over both state and time, providing an improvement over basic hazard models. It simulates not only that a fault occurs, but also predicts the duration of the fault and repair, providing a model of the cyclic process of failure and repair. The understanding that we gain from analyzing this structure can be used to build better work plans and perform maintenance in a cost effective manner. An sample simulation of the assemblies' life-cycle is included in Figure 6.

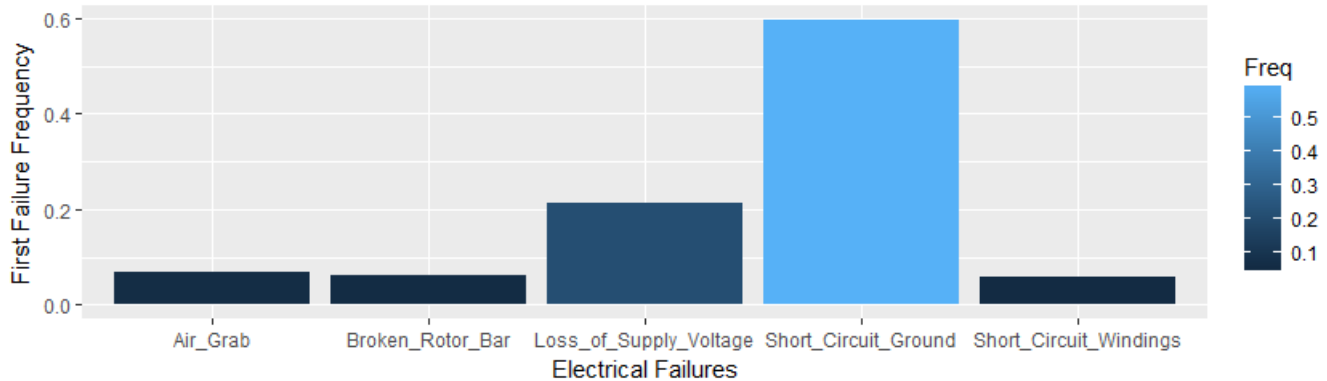


Figure 4. Probability of First Component Failure in Fem

For example by placing the inspections on intervals in the model, we are able to model system degradation based on the inspection's outcome. One advantage of the CTBN model is that it supports both continuous inspections and just-in-time inspections at variable intervals, improving assessment over traditional interval based inspection strategies. These different types of inspection are a more appropriate procedure to design work-plans, giving maintenance the ability to choose more flexible inspections for each fault.

Additionally through the use of risk analysis, the CTBN gives advice about the allocation of resources, thereby improving mission readiness. Using Monte-Carlo techniques, we predict when failures occur, how long they will last, the cost of failure, and the probability of a failure occurring. This provides excellent visibility into system behavior. Also, by applying evidence from tests to the system, the program models the future performance of individual components and subassemblies under uncertainty.

## 9. FUTURE WORK

Several avenues of future work are available. An obvious extension is to build workplans automatically from the risk and reliability analysis. Ideally, we would like to be able to design efficient workplans algorithmically through the CTBN framework. We would also like to start to work with more complex systems using dynamic fault trees.

The second extension involves improving the efficiency of the computation when doing inference. We believe there are optimizations to our inference engine for performance gains. This would increase the tractability of the CTBN framework on new systems and speed-up the experimental life cycle. Some of these improvements may be based on the specific nature of modeling system failure, where other improvements may draw on inference techniques, for example, based on variational methods.

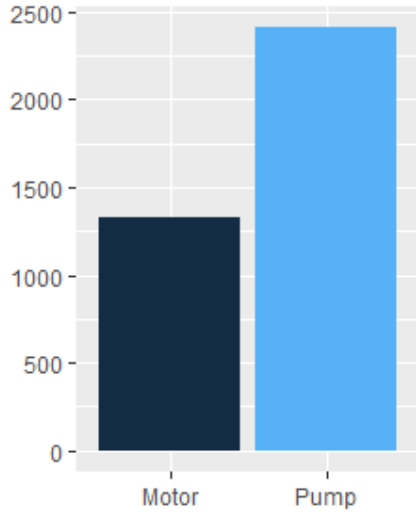
The third extension focuses on developing methods for quan-

tifying uncertainty in the system. By studying uncertainty in the model, we can understand where the algorithm is operating with less certainty and then focus our attention on those parts of the system, because study of the system assemblies with less certainty will provide larger gains in readiness than studying systems with small error bars.

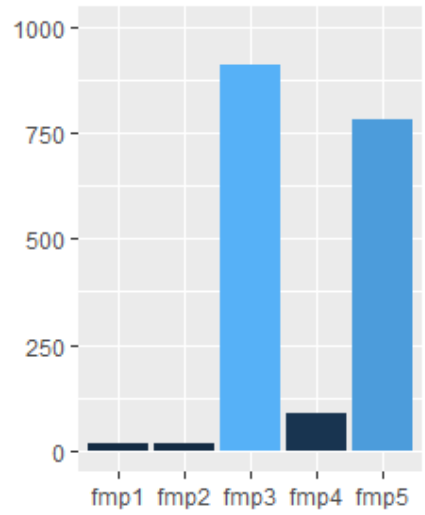
The fourth area to consider is improving the risk analysis framework. Calculating the present value of risk over a dynamic system is a challenging problem with widespread application. Improving this risk analysis process yields large gains in readiness.

## REFERENCES

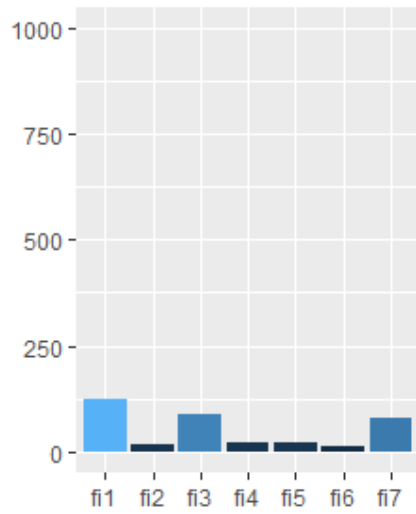
- Behrends, L., Houke, L., Bailey, E., Jansen, P., & Brown, D. (2001). Reciprocating constructed wetlands for treating industrial, municipal and agricultural wastewater. *Water science and technology*, 44(11-12), 399–405.
- Fan, Y., & Shelton, C. R. (2008). Sampling for approximate inference in continuous time Bayesian networks. In *Proceedings of ISAIM*.
- Fan, Y., Xu, J., & Shelton, C. R. (2010). Importance sampling for continuous time Bayesian networks. *Journal of machine learning research*, 11(Aug), 2115–2140.
- Ghoshal, S., & Sheppard, J. (2016). *Nasa sttr t13.01-9887: Risk engineering, sciences, computation and informed decisions (rescaid) phase ii final report* (Tech. Rep. No. QSI-REP-16-021). Qualtech Systems Inc.
- Goebel, K., Saha, B., Saxena, A., Celaya, J. R., & Christophersen, J. P. (2008). Prognostics in battery health management. *IEEE instrumentation & measurement magazine*, 11(4), 33–40.
- Gülich, J. F. (2008). *Centrifugal pumps* (Vol. 2). Springer.
- Howard, R. A. (1960). *Dynamic programming and Markov processes*. John Wiley.
- Kallesøe, C. S. (2005). *Fault detection and isolation in cen-*



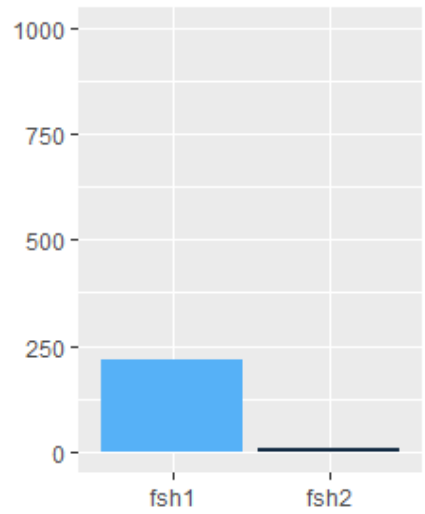
(a) Assemblies



(b) Pump Mechanical

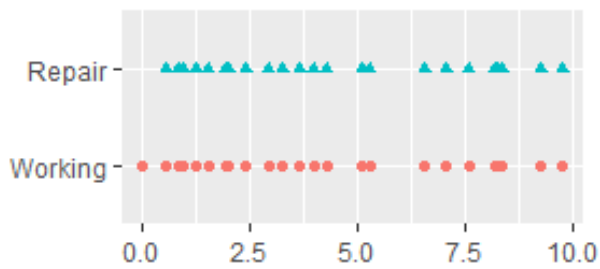


(c) Impeller

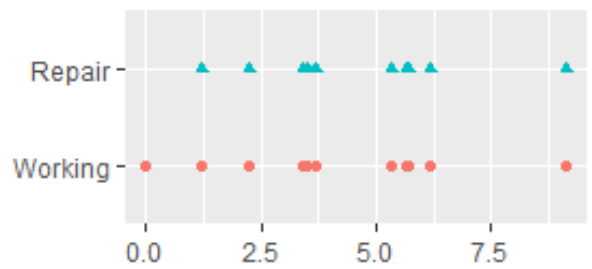


(d) Shaft

Figure 5. Expected Repair Cost per  $10^5$  hours



(a) Pump



(b) Motor

Figure 6. Simulated Assembly Life-cycle over  $10^6$  Hours in  $10^5$  Hour Increments

- trifugal pumps*. Carsten Skovmose Kallesøe.
- Kim, N.-H., An, D., & Choi, J.-H. (2017). Prognostics and health management of engineering systems. *Switzerland: Springer International Publishing*.
- Kliman, G., Premerlani, W., Koegl, R., & Hoeweler, D. (1996). A new approach to on-line turn fault detection in ac motors. In *Conference record of the IEEE 31st industry applications conference* (Vol. 1, pp. 687–693).
- Lall, P., Lowe, R., & Goebel, K. (2012). Prognostics health management of electronic systems under mechanical shock and vibration using Kalman filter models and metrics. *IEEE Transactions on industrial electronics*, 59(11), 4301–4314.
- Mays, L. W. t. (1989). *Reliability analysis of water distribution systems*. ASCE.
- Murphy, K. P. (2002). *Dynamic Bayesian networks: Representation, inference and learning* (Unpublished doctoral dissertation). University of California Berkeley.
- Nodelman, U., Shelton, C. R., & Koller, D. (2002a). Continuous time Bayesian networks. In *Proceedings of the eighteenth conference on uncertainty in artificial intelligence* (pp. 378–387).
- Nodelman, U., Shelton, C. R., & Koller, D. (2002b). Learning continuous time Bayesian networks. In *Proceedings of the nineteenth conference on uncertainty in artificial intelligence* (pp. 451–458).
- Nodelman, U. D. (2007). *Continuous time Bayesian networks* (Unpublished doctoral dissertation). Stanford University.
- Orchard, M., Kacprzynski, G., Goebel, K., Saha, B., & Vachtsevanos, G. (2008). Advances in uncertainty representation and management for particle filtering applied to prognostics. In *2008 international conference on prognostics and health management* (pp. 1–6).
- Pearl, J. (1988). *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Elsevier.
- Pecht, M. (2008). Prognostics and health management of electronics. *Encyclopedia of structural health monitoring*.
- Perreault, L., Thornton, M., Goodman, R., & Sheppard, J. W. (2015). Extending continuous time Bayesian networks for parametric distributions. In *IEEE swarm intelligence symposium*.
- Perreault, L., Thornton, M., & Sheppard, J. (2016a). Deriving prognostic continuous time Bayesian networks from fault trees. In *Proceedings of the annual conference of the prognostics and health management society*.
- Perreault, L., Thornton, M., Sheppard, J., & DeBruycker, J. (2017). Disjunctive interaction in continuous time Bayesian networks. *International journal of approximate reasoning*, 90, 253–271.
- Perreault, L., Thornton, M., & Sheppard, J. W. (2016b, September). Valuation and optimization for performance based logistics using continuous time Bayesian networks. In *IEEE AUTOTESTCON conference record*.
- Perreault, L., Thornton, M., Strasser, S., & Sheppard, J. W. (2015). Deriving prognostic continuous time Bayesian networks from D-matrices. In *IEEE AUTOTESTCON conference record* (pp. 152–161).
- Ross, S. M. (1990). *A course in simulation*. Prentice Hall PTR.
- Saria, S., Nodelman, U., & Koller, D. (2012). Reasoning at the right time granularity. *arXiv preprint arXiv:1206.5260*.
- Sheppard, J., Perrault, L., & Forrester, T. (2019). Risk-based prognostics and health management. *In preparation*.
- Sheppard, J. W., Gorton, J., Kalgren, P. W., & Sturlaugson, L. E. (2018, May). *Integrated model for failure diagnosis and prognosis*. (US Patent Appl. 15/573,265)
- Simpson, W. R., & Sheppard, J. W. (1994). *System test and diagnosis*. Springer Science & Business Media.
- Smith, C., Knudsen, J., Kvarfordt, K., & Wood, T. (2008). Key attributes of the sapphire risk and reliability analysis software for risk-informed probabilistic applications. *Reliability engineering & system safety*, 93(8), 1151–1164.
- Sturlaugson, L., Perreault, L., & Sheppard, J. W. (2017). Factored performance functions and decision making in continuous time bayesian networks. *Journal of applied logic*(22), 28–45.
- Sturlaugson, L., & Sheppard, J. W. (2014, July). Inference complexity in continuous time bayesian networks. In *Proceedings of the conference on uncertainty in artificial intelligence (UAI)* (pp. 772–779).
- Sturlaugson, L., & Sheppard, J. W. (2016). Uncertain and negative evidence in continuous time bayesian networks. *International journal of approximate reasoning*, 70, 99–122.
- Tipping, M. E. (2001). Sparse bayesian learning and the relevance vector machine. *Journal of machine learning research*, 1(Jun), 211–244.
- Watson, H. (1961). *Launch control safety study* (Tech. Rep.). Murray Hill, NJ: Bell Telephone Laboratories.