SYSTEM HEALTH DIAGNOSIS AND PROGNOSIS USING DYNAMIC BAYESIAN NETWORKS

By

Gregory W. Bartram

Dissertation

Submitted to the Faculty of the

Graduate School of Vanderbilt University

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

in

Civil Engineering

August, 2013

Nashville, Tennessee

Approved:

Prof. Sankaran Mahadevan

Prof. Prodyot K. Basu

Prof. Gautam Biswas

Prof. Mark P. McDonald

To my wife and family

ACKNOWLEDGEMENTS

This dissertation would not be possible without the support of many individuals and institutions. The utmost gratitude is due to my advisor, Prof. Sankaran Mahadevan. Prof. Mahadevan's belief in me, expertise, advice, time, and whip cracking have proven indispensable in my academic and life journey. For this, I am thankful, and I look forward to continuing our relationship for many years to come.

I am also grateful to my committee members Prof. Prodyot K. Basu, Prof. Gautam Biswas, and Prof. Mark P. McDonald. Their insight, feedback, advice, and faith in me have motivated and guided me to new heights. I would also like to thank my various sponsors, including the National Science Foundation's Integrative Graduate Education and Research Traineeship (IGERT) program, the U. S. Air Force Research Laboratory, WPAFB, NASA Ames Research Center, and the Vanderbilt Department of Civil and Environmental Engineering for teaching assistant support.

The faculty, staff, and students at Vanderbilt have all provided me with much help and support, for which I am thankful. Dr. Shankar Sankararaman and You Ling have provided me a great amount of technical guidance, friendship, and support as have Dr. Robert D. Crouch, Joshua R. Arnold, Joseph H. Rustick, and Paul A. Sparks.

I am also thankful to my mother and late father. Without their incredible love, guidance, emotional and financial support, I would not be where I am. Additionally, I would like to thank my sister Jeanne, my late grandfather Thomas W. Bartram, Jr., and my grandmother Marianne Bartram for their support and encouragement.

Finally, I owe the greatest thanks to my patient wife Barrett, who was expecting me to be finished two years ago. Her unconditional love and support has kept me on task and is my true motivation.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

APF: auxiliary particle filter

CBM: condition based maintenance

DBN: dynamic Bayesian network

EoL: end of life

EoUP: end of useful prognosis

MC: Monte Carlo

PH: prognostic horizon

PF: particle filter

RUL: remaining useful life

SIF: stress intensity factor

SIR: sequential importance resampling filter

TTF: time to failure

CHAPTER I

1. INTRODUCTION

1.1 Motivation

Engineering applications such as aerospace, transportation, communication, energy systems, and healthcare demand high reliability, safety, and performance. To meet these criteria, such systems rely on many interacting subsystems, e.g. aircraft enlist structural, propulsion, communication, navigation, and flight control subsystems, among others. The combination of these subsystems leads to a complex system that is expected to satisfy the reliability, safety, and performance requirements for which it was designed. However, through manufacturing/construction, usage, changing environment and demands, systems experience changes that compromise their reliability, safety, and performance. Naturally, it is desired to take steps that prevent any sort of failure of the system to meet its demands. These steps may be maintenance, reinforcement, repurposing, or even retirement of the system. However, deciding the course of action is no simple task, especially in light of system complexity.

The information involved in such a decision consists of prior knowledge of the system, present knowledge of the system, and future estimates of the system. Prior knowledge comes in many formats from many sources. It may be combined and represented as a probabilistic system model. Present knowledge of the system is based on the most recent observations, and is provided by diagnosis in the form of an estimate of the system's state. The state is the set of values of all variables that describe the system. Diagnosis infers the (unobservable) system state using observations of the system. Future knowledge of the system is provided by prognosis. Prognosis

estimates future states of the system and provides a quantitative assessment of the future health of the system, typically by estimating the remaining useful life (RUL) of the system. Prognosis estimates also require some knowledge of the demands that a system will experience in the future and degradation of system capacity with time.

The prior, present, and future knowledge necessary for decision-making may be more easily understood and manipulated when in a structured format. Dynamic Bayesian networks (DBNs) provide a means for aggregating heterogeneous prior information into a system model, integrating the latest measurements into that model, and predicting future states of the system. As such, DBNs are the backbone of the methodology investigated in this dissertation. Subsections 1.1.1 and 1.1.2 provide a brief introduction to Bayesian networks and dynamic Bayesian networks.

1.1.1 Bayesian Networks

A static BN is a probabilistic graphical representation of a set of random variables and their conditional dependencies. Variables are represented by nodes (vertices) and conditional dependence is represented by directed edges. BNs are acyclic, meaning that no paths exist in the graph where, starting at node X_i, it is possible to return to node X_i.

Theory for BNs has been developed by Pearl [1] and is explained here using notation from Heckerman et al. [2] and Friedman et al [3]. Consider the domain U consisting of n variables $X_1, \dots, X_n \in \boldsymbol{X}$. The joint probability distribution over U may be written using the chain rule of probability as

$$p(X_1, \dots, X_n) = \prod_{i=1}^{n} p(X_i | X_1, \dots, X_{i-1}) \qquad \textbf{Eq. 1}$$

Two variables A and B are conditionally independent of each other given a third variable C when $Pr(A|C) = Pr(A|B, C)$. That is, once the value of C is known, any evidence about B does not affect the belief about C and vise-versa. Figure 1 shows the graphical representation of this assumption.

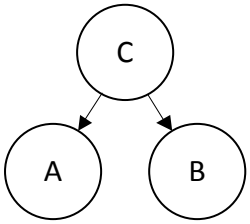


Figure 1. A and B are conditionally independent given C

From Figure 1, it is readily apparent that when A and B are conditionally independent of each other given a third variable C, A is not a parent of B and B is not a parent of A. Thus, conditional independence assertions for each variable X_i, are contained in the set of parent variables X_i denoted by Π_i. Π_i contains only the variables in $\{X_1, \dots, X_{i-1}\}$ on which X_i is conditionally dependent. Thus,

$$p(X_i|X_1, \dots, X_{i-1}) = p(X_i|\Pi_i) \quad \textbf{Eq. 2}$$

The BN structure which has nodes corresponding to each variable X_i in U and parent nodes Π_i for each variable X_i is denoted B_s. The collection of local distributions $p(X_i \mid \Pi_i)$ at each node in U is the Bayesian network probability set, B_p. A BN for U, denoted by B, is given by the pair (B_s, B_p) with a uniquely determined joint probability distribution for U,

$$p(X_1, \dots, X_n) = \prod_{i=1}^{n} p(X_i|\Pi_i) \quad \textbf{Eq. 3}$$

It should be noted that this definition of a BN depends on the variable ordering, which must be chosen carefully to ensure that conditional independence assertions are represented properly. The formulation in Eq. 3 is readily extended to handle different types of distributions such as the

Gaussian. A further extension (see Lauritzen [4]) is a hybrid network where Gaussian variables may have discrete parents.

1.1.2 Dynamic Bayesian Networks

A DBN can be thought of as a series of BNs, one for each instant of time and whose state depends on the BNs at previous time steps when all nodes are observable and the database is complete. The variable $X[t]$ is the value of X at time t. The probability distribution describing X on the interval [0, ∞) is very complex, as it is over $\bigcup_{t=0}^{t=\infty} X[t]$. Using the Markov assumption simplifies this distribution by assuming that only the present state of the variable $X[t]$ is necessary to estimate $X[t+1]$ and $p(X[t+1] \mid X[0], \ldots X[t]) = p(X[t+1] \mid X[t])$. Additionally, it is necessary to assume that the process is stationary, meaning that $p(X[t+1] \mid X[t])$ is independent of t. This approach to modeling DBNs is used by Friedman et al. [3].

The DBN is composed of two parts, a prior network $B[0]$ specified by $(B_s[0], B_p[0])$ and a transition network B_{tr} over X[t] U X[t+1] which specifies $p(X[t+1] \mid X[t])$ over all t. The probability distribution for the transition network is

$$p_{B_{tr}}(\boldsymbol{X}[t+1]|\boldsymbol{X}[t]) = \prod_{i=1}^{n} p_{B_{tr}}\left(X_i[t+1]\middle|\Pi_{X_i[t+1]}\right) \qquad \textbf{Eq. 4}$$

where $\Pi_{X_i[1]}$ are the parents of X_i[t+1]. In reality, t is on the finite interval [0, T] instead of [0, ∞). The joint distribution over this network is then

$$p_B(\boldsymbol{X}[0], \ldots, \boldsymbol{X}[T]) = p_{B[0]}(\boldsymbol{X}[0]) \prod_{t=0}^{T-1} p_{B_{tr}}(\boldsymbol{X}[t+1]|\boldsymbol{X}[t]) \qquad \textbf{Eq. 5}$$

where $p_{B_{tr}}(\boldsymbol{X}[t+1]|\boldsymbol{X}[t])$ is determined from the transition model.

A DBN may be composed of all discrete variables, all continuous variables, or hybrid set of discrete and continuous variables. A conditional probability distribution (CPD) is chosen for each variable, e.g. Gaussian, tabular (multinomial), softmax, deterministic, logic, etc. See Koller and Friedman [5] for a detailed explanation of CPDs. For modeling systems with faults, it is advantageous to consider a hybrid system, typically with the continuous variables being modeled as continuous and the faults being discrete. Theory for networks with Gaussian continuous variables is developed in Heckerman and Geiger [6] and Lauritzen [4].

DBNs provide a flexible modeling framework, allowing integration of expert opinion, reliability data, mathematical models (including system state space, surrogate, and physics of failure models), existing databases of operational and laboratory data, and online measurement information. Bartram and Mahadevan [7] have proposed a methodology for integration of such heterogeneous information into DBN system models. In the next section, that discussion is extended to consider physics of failure models, which are of particular importance in prognosis.

The remainder of this chapter establishes the objectives of this dissertation. Section 1.2 presents the goal and objectives of this research. Sections 1.4 – 1.5 discuss the key steps in the methodology that consist of system modeling, diagnosis, prognosis, and decision-making. Section 1.6 explains the goals of the research while Section 1.7 presents the highlights and contributions of this dissertation. Section 1.8 outlines the rest of the dissertation.

1.2 Research Goal and Objectives

The primary goal of this research is to develop a methodology that provides information to make optimal decisions with respect to the mission and maintenance of a system. A great amount of

information is needed about a system to make such decisions, including its current condition and predictions of its future state. The problem is broken into four objectives.

1) Develop a system modeling approach for use when the available information is heterogeneous, i.e. available in various formats from various sources. The resulting system model should account for uncertainty and be amenable to the system health management tasks of diagnosis, prognosis, and decision-making.
2) Develop a diagnosis approach for systems when the available information is heterogeneous. The approach should account for and quantify uncertainty in damage detection, isolation, and quantification.
3) Develop a methodology for system prognosis when the available information is heterogeneous. The methodology should account for uncertainty in diagnosis and be subject to validation.
4) Develop a methodology for decision-making when the available information is heterogeneous and when assigning multiple systems to multiple missions. The methodology should account for uncertainty in diagnosis and prognosis.

The following subsections describe the approaches used to address each objective.

1.2.1 System Modeling with Heterogeneous Information

The methodology has several important features. First, by using dynamic Bayesian networks (DBNs) for modeling the system, uncertainty in the system may be accounted for, as DBNs are a representation of the joint probability distribution over a set of variables. The DBN-based approach also allows information in various formats and from various sources (heterogeneous information),

including expert opinion, mathematical models, reliability data, laboratory data, and historical and online operational data to be included in the model.

1.2.2 Damage Diagnosis

The DBN is then used in online diagnosis, which estimates the distribution over the current state of the system. When new measurements become available, the beliefs about the distributions of other variables in the network are updated via statistical inference. In a DBN, this is performed using a sequential Monte Carlo approximation called particle filtering. The particle filter represents the system state as an approximate distribution consisting of samples and weights. This representation is useful for understanding diagnosis uncertainty and detecting faults. It also provides a convenient initial condition for prognosis.

1.2.3 Prognosis

Prognosis predicts future states of a particular system and determines the outlook for the system, expressed quantitatively as the remaining useful life (RUL). The initial conditions for prognosis are established during diagnosis (state estimation). The particle-based state representation created during diagnosis seamlessly transitions to the sequential Monte Carlo prediction used for prognosis and allows a particle-based approximation of the RUL distribution.

1.2.4 Decision-making

Decision-making relies on the RUL distribution obtained during prognosis. The prognosis is performed for various situations using expected loads and inputs to the system. An optimization problem is solved to determine how to deploy the system in the case of multiple instances of the

system and multiple possible mission assignments, which can include maintenance and rest. The optimization solves the assignment problem by minimizing the risk of failure across all units.

1.3 Highlights

1. A methodology for system modeling under heterogeneous information is proposed. The methodology integrates expert opinion, reliability data, mathematical models, laboratory data, and operational data into a dynamic Bayesian network system model.

2. DBN-based diagnosis of a mechanical system under heterogeneous information is developed. The methodology is a general approach that uses the existing DBN model of the system. Additionally, the methodology can handle systems with multiple faults.

3. Diagnosis uncertainty is quantified in the context of particle filtering by estimating the probability of detecting any fault, estimating the isolation probabilities of all fault combinations, and estimating the distribution of damage parameters.

4. A methodology for prognosis of a system under heterogeneous information is developed. The methodology seamlessly integrates with the diagnosis procedure and utilizes the same DBN model of the system.

5. An optimization problem is formulated for solving the decision-making problem with multiple systems and multiple assignments. The formulation considers multiple limit states and accounts for probabilistic diagnostic and prognostic information.

1.4 Organization of the Dissertation

Chapter 2 presents a methodology for the integration of heterogeneous information into a Dynamic Bayesian network system model. Here, the difficulty is that information is available in disparate forms and from various sources that are generally not easily reconciled. Additionally, physics based models may be unknown or too complex for practical use. Bayesian networks provide a structured approach for combining information as well as the ability to learn data-driven models to explain complex interactions. First, the semantics and mathematics of Bayesian networks are presented. Next, inclusion of expert opinion, reliability data, mathematical models, laboratory data, and operational data is explained. The methodology is demonstrated for a hydraulic actuator system with multiple possible faults.

In Chapter 3, the problem of damage diagnosis is considered. Damage diagnosis is the process of detecting, isolating, and quantifying damage in a system. After reviewing previous work in diagnosis, Bayesian recursive filtering and approximate filtering via the particle filter are discussed as approximate inference methods in a dynamic Bayesian network that is constructed as in Chapter 2. The state estimate provided by filtering contains the necessary information to detect, isolate, and quantify damage in the system. Quantification of diagnosis uncertainty in the context of particle filtering is developed. Diagnosis is illustrated on a cantilever beam with possible damage at the support and a midspan crack.

The subject of Chapter 4 is damage prognosis. Prognosis consists of state estimation, prediction, and remaining useful life (RUL) estimation. State estimation is the result of diagnosis (Chapter 3). In Chapter 4, DBN-based prediction beginning with the state estimate from diagnosis is first considered. Next, physics of failure modeling is also discussed, in particular, the wear phenomenon. The RUL density estimate is then explained and steps for validating a prognostic algorithm are

given. The prognosis methodology is applied to a hydraulic actuator with a progressive seal leak resulting from wear.

In Chapter 5, a decision-making problem is solved via optimization. The problem is to assign multiple systems of the same type but with different health states to different tasks and to minimize the chance that any system exceeds its RUL. First, the tasks; including operation (with different load profiles and inputs), rest, and maintenance; are defined. Next, the load profiles, inputs, and current health state of the systems are used to estimate the RUL, as in Section 4, for each system. The procedure is demonstrated on a group of hydraulic actuators.

Each chapter contains a review of relevant literature. The notation is defined for each chapter separately. Complete citations are provided at the end of the dissertation.

CHAPTER II.

2. INTEGRATION OF HETEROGENEOUS INFORMATION IN DYNAMIC BAYESIAN NETWORKS

2.1 Introduction

System health monitoring (SHM) is a challenge that has been approached with a broad spectrum of approaches. On one end of the spectrum are data-driven methods, relating observable symptoms directly to failure modes. On the other end of the spectrum are model-based approaches, which indicate failure by finding differences between expected healthy system responses and measured system responses. In some instances of model-based diagnosis/prognosis, a mathematical model may be well-established from physics or exhaustive study. However, in other cases, mathematical models for some components may be nonexistent (e.g. cyber-physical systems with hardware and software components). Probabilistic models may be desired in order to account for many sources of uncertainty while the probability distributions of system variables and the conditional probability distributions between them may be unknown. These problems may be due to the complexity of the physics required to model the system or a lack of existing quantitative observational data about the system.

An important issue in SHM is that the information pertaining to the system may be heterogeneous, i.e. available in various formats from multiple sources, such as literature, experts, model predictions, experiments, and operational data. The integration of such heterogeneous information is possible for individual quantities through Bayesian analysis; however, diagnosis and prognosis of systems with multiple levels of components and subsystems, multiple physics, and multiple damage

mechanisms is not straightforward in the presence of heterogeneous information. System-level integration of information obtained during subsystem and component level analysis is important for SHM of real-world systems whose subsystems and components have complex interrelationships. Many different SHM techniques have been developed for diagnosis at subsystem and component levels or specific failure modes (e.g. acoustic emission testing [8], wireless sensor networks [9], crack growth measurement [10], online damage monitoring using input error functions [11], fast mode identification [12], and many others). In a realistic system, multiple techniques may be applied for diagnosis of different faults, providing another instance of heterogeneous information that needs to be integrated in system-level prognosis.

A machine learning approach is an intuitive solution to this problem. Machine learning approaches based on Bayes networks (BNs) and dynamic Bayes networks (DBNs) can interpret heterogeneous information and construct a probabilistic model of a system, making sense of information that humans would normally find overwhelming and indecipherable. BN and DBN system models can support a hybrid data-driven/model-based SHM approach due to their ability to simultaneously incorporate many types of data and serve as a system model. Hybrid SHM approaches and probabilistic information have both been found to decrease the complexity of diagnosis by decreasing the size of the diagnosis search space [13]. Thus BN and DBN system models make an ideal platform for SHM. Many applications of DBNs in SHM are available, such as Choudhury et al. [14] and Jha et al. [15]. However, these approaches use DBNs generated from known state space models or learn them from a homogeneous data source instead of tapping all available heterogeneous information.

In this chapter, a Bayesian methodology is presented for creating a dynamic system model for probabilistic diagnosis and prognosis in the presence of multiple types of data and components.

Heterogeneous information, available from multiple sources and in various formats — operational (observational) data, laboratory (interventional) data, published failure rates, mathematical models, and expert opinion — is used to construct a dynamic Bayes network. The dynamic Bayes network structure allows for the integration of various types of information and aids diagnosis/prognosis with both forward and backward reasoning. The methodology is demonstrated for a hydraulic actuator used in an aircraft flight control system.

2.2 Previous Work

2.2.1 Learning DBNs

Research in learning system models based on BNs and DBNs has been ongoing for years and a vast amount of literature related to learning methods and applications is available. Particular focus has been given to static BNs but this work is often extensible to DBNs. Buntine [16] provides a good survey of BN learning methods. As methods for transforming a database of observational data have matured, work has been done to accommodate BN system modeling in the presence of specific types of information. Kipersztok [17] proposes combining reliability data with learned static BNs for aircraft maintenance decision support. Challagulla et al. [18] explore software architecture reliability by analyzing datasets. They use a BN to combine reliability data from analysis and operations to dynamically predict system reliability.

Various approaches for handling prior information in BNs have been developed, although they focus on determining a structural prior. Leach [19] aggregates expert opinion (domain knowledge) of varying quality to create a structural prior for a BN describing interactions between genes. Langseth et al. [20] exploit prior information provided by repetitive structures in object-oriented

domains with help from domain experts. Richardson and Domingos [21] combine opinions of multiple experts to establish a structural prior for a BN.

Recently, interest has been growing in the machine learning community on the task of combining data from multiple sources [22]. The problem is often framed as an inductive transfer learning problem (or multi-task learning), where a model is learned by using information from related models. Early work on the problem of inductive transfer learning was done by Thrun [23], Caruana [24], and Baxter [25]. More recently, Luis et al. [26] proposed a transfer learning methodology using conditional independence tests to learn Bayesian networks. Dai et al. [27] used labeled data from similar domains in a transfer learning procedure for text classification. Silver et al. [28] developed a context-sensitive multiple task learning method. Niculescu-mizil and Caruana [29] proposed a method for learning similar tasks by using similar structures for each task. Roy and Kaelbling [30] proposed an efficient algorithm for multi-task learning using a Dirichlet process approximation. Szafranksi et al. [31] used a multiple kernel method to learn a kernel from an ensemble of basis kernels.

Other related problems have been considered, such as combining data from multiple experts for learning (Richardson and Domingos [21]). Wu and Dietterich [32], Dai et al. [27], and Ben-David et al. [33] consider learning when auxiliary data is drawn from distributions other than the underlying distribution. Amini and Gouette [34] use a multi-view learning approach for categorizing text from multilingual documents. An online learning framework proposed by Dredze et al. [35] combines parameters from multiple classifiers to learn across domains.

2.2.2 Data Fusion

Data fusion is a term describing the combination of any two or more sources of data. For example, two sensor measurements may be combined into a "best" estimate using data fusion. Updating a distribution upon obtaining a measurement is also data fusion. DBNs, BNs, and Bayesian recursive filtering are common tools used in data fusion. Here, DBNs in data fusion shall be considered.

The data fusion concept has been useful in a wide range of problems and has integrated several types of information. Qu et al. [36] use a DBN-based data approach to integrate data generated by mathematical models (which were themselves fitted from historical data), and remote sensor information. Das et al. [37] use DBNs built from expert opinion and observational data in a factored particle filtering scheme for situational awareness in military operations. Cappelle et al. [38] fuse sensor data with DBNs for geo-location. Chen and Ji [39] integrate temporal and spatial data in DBNs for improved online labeling. Lerner et al. [40] make use of expert opinion and mathematical models in DBNs for diagnosis but do not consider operational/laboratory data or reliability data. Roychoudhury et al. [41] use expert opinion and mathematical models (two tank system, incipient faults and abrupt faults) but no operational/laboratory data or reliability data. Przytula and Choi [42] integrate expert opinion and observational data with DBNs to estimate probability of component failure. Arroyo-Figueroa and Sucar [43] include expert opinion and operational data in a diagnosis methodology using temporal Bayesian networks of events.

Data fusion applications for mechanical systems are relatively infrequent in the literature, even as a large variety of heterogeneous information may be available for such systems. Some examples include Straub [44], who includes deterioration model parameters in a DBN, and Dong and Yang [45], who use DBNs to integrate expert opinion and operational data in estimating remaining useful life of drill bits. Straub and Der Kiureghian [46] use BNs for enhanced structural reliability analysis.

BNs have also been used to combine sources of uncertainty in structural damage prognosis, as in Sankararaman et al. [47] and Ling and Mahadevan[48].

Some fusion efforts have considered problems with (or constrain problems to) two or three sources of information. For example, Straub [44] uses expert opinion, a mathematical model, and inspection data. In machine learning, a class of fusion approaches is inductive transfer learning (or multi-task learning), where a model is learned by using information from related models [22]. Early work on the problem of inductive transfer learning was done by Thrun [23], Caruana [24], and Baxter [25]. Other related problems have been considered, such as combining data from multiple experts for learning (Richardson and Domingos [21]). An online learning framework proposed by Dredze et al. [35] combine parameters from multiple classifiers to learn across domains. Because interactions, known and unknown, between components and subsystems may exist, it is important to consider multiple streams of information which may reveal such interactions while also characterizing the behavior of isolated components.

2.3 Contributions of this Chapter

Previous work in machine learning has focused on learning better system models using limited varieties of information, but generally do not consider the application of the model to probabilistic diagnosis and prognosis where heterogeneous information is available. Accurate predictive and diagnostic capabilities are needed for prognosis of high-risk systems. DBNs provide a causal and probabilistic description of a system, thus providing more complete information than deterministic or acausal system models, and support multiple tasks including probabilistic diagnosis and prognosis.

The primary contribution of this chapter is a methodology for creating a system-level model intended for probabilistic diagnosis and prognosis when the probability distributions describing system variables and the conditional probability relationships between them are unknown, and when heterogeneous information is available. The methodology is general and can be applied in any domain. A diagnosis/prognosis probabilistic model should not only be capable of predicting nominal system behavior but also be capable of approximating system behavior under many damage scenarios with limited information from various sources.

The proposed methodology applies existing machine learning approaches to heterogeneous information likely available to system modelers. Specifically, the use of operational data, laboratory data, published reliability data, expert opinion and mathematical models is investigated and implemented on a dynamic system. The methodology allows the model to capture the behavior of the system in healthy and faulty operational states.

Operational and laboratory data are key elements in building a model for diagnosis and prognosis. Operational data is obtained from the system operating in service and is necessary to understand how subsystems and components interact. Laboratory data, often called interventional data, come from experiments designed to test system behavior. For example, a fault may be seeded in a prototype of the system to determine the faulty state operational profile of the system. This type of experiment is very important in model construction, as it may take excessive amounts of operational data to quantify the effect on system behavior of a fault that occurs infrequently.

If the system is constructed from known components (e.g. valves, fittings, etc.), reliability databases may be available for individual components or be obtainable relatively cheaply through testing,

even if the relationships between components may be unknown. Failure rates can be used in determination of the parameters of the distribution of that fault.

A similar approach can be used if a mathematical model is available for a subsystem or component. The mathematical model indicates whether the subsystem is in a failed state and the faults responsible for the failure. A collection of realizations of the CDF can be used to determine Weibull distribution parameters. The parameters of the corresponding fault distribution are then computed. The remainder of the chapter is organized as follows. First, DBNs are introduced. Then, learning the structure and parameters of DBNs is explained. After these preliminaries, the proposed methodology for the inclusion of heterogeneous information from laboratory and experimental data, expert opinion, a reliability database, and a subsystem mathematical model to learn a DBN is detailed. The methodology is demonstrated on a hydraulic actuator controlled by a spool valve.

2.4 System Modeling Framework

The proposed methodology seeks to develop a system diagnosis/prognosis model in the presence of heterogeneous information. This requires a framework capable of learning the model by analyzing heterogeneous information sources. Many frameworks exist but not all are capable of handling many different types of information. A simple method for learning is a regression model where the coefficients are estimated by minimizing the sum of squares of differences between observed and predicted values. Other, more complex learning platforms include Gaussian process (GP) models [49], neural networks (NNs) [50], hidden Markov models (HMMs)[51], Bayesian networks (BNs)[1] and dynamic Bayesian networks (DBNs) [52], and support vector machines [53]. Regression models are popular and user friendly but according to Korb [54], cannot model interventions “in ordinary usage,” which are required to determine the correct causal structure of a system. GPs and NNs may be efficiently discovered but do not reveal much about the causal

structure of a system, which is desirable for diagnosis, prognosis, and design. HMMs and BNs are special cases of DBNs. However, since the proposed methodology models dynamic systems, DBNs are necessary.

DBNs provide a useful framework for this dissertation due to their (1) ability to illuminate the underlying dependence structure within a system and predict consequences of intervention, (2) ability to handle prior knowledge, and (3) ability to avoid overfitting the data [2]. DBNs are suitable for online monitoring applications in conjunction with particle filtering. Further, the ability of DBNs to incorporate heterogeneous information into the system model is of considerable importance. Due to these benefits, a methodology which learns a DBN representation of the system is pursued in this dissertation.

First, expert opinion is invoked to determine the basic problem setup and initial modeling assumptions. This includes determining important system variables, selecting learning algorithms, and choosing a validation metric. Data from observation and controlled experiments, reliability data, mathematical models, expert opinion must then be carefully integrated into the DBN model. The process of integration is governed by the DBN formalisms and modeling assumptions and requires an understanding of BNs, DBNs, and the algorithms used to learn them.

2.5 Integrating Heterogeneous Information

An important need in system diagnosis and prognosis is the ability to incorporate mixed information types. The graphical and probabilistic nature of BNs provides many opportunities to integrate information into the model. This allows them to benefit from heterogeneous information, i.e. information entrained in a variety of sources and formats. These sources include observational and experimental data, published reliability data, mathematical behavior models of components or

subsystems, or expert opinion. Table 1 describes typical sources and formats of heterogeneous information. Existing research has not fully exploited heterogeneous information, especially with respect to building DBN models for diagnosis and prognosis.

Table 1. Heterogeneous information and sources

Information Type	Sources	Formats
Expert opinion	domain experts, scholarly publications, technical reports	distribution types, parameter estimates, network structures, system assumptions
Operational data	field data	databases
Laboratory data	controlled laboratory experiments	databases
Reliability data	scholarly publications, technical reports domain experts	distribution types, failure rates
Mathematical models	scholarly publications, technical reports, domain experts	physics-based and data-driven system models

First, expert opinion is invoked to determine the basic problem setup and initial modeling assumptions. This includes determining important system variables, selecting learning algorithms, and choosing a validation metric. Data from observation and controlled experiments, reliability data, mathematical models, expert opinion are then carefully integrated into the DBN model, as described in the following subsections.

2.5.1 Expert Opinion

Various approaches for handling expert opinion in BNs have been developed, and they focus on determining a structural prior. Leach [19] aggregates expert opinion (domain knowledge) of varying quality to create a structural prior for a BN describing interactions between genes. Langseth et al. [20] exploit prior information provided by repetitive structures in object-oriented domains with help from domain experts. Richardson and Domingos [21] combine opinions of multiple experts to establish a structural prior for a BN.

Expert opinion could include domain experts, scholarly publications, and technical reports that can provide assumptions about system variables and their distributions. Sometimes, the expert can provide a network structure and conditional probabilities, which may be used as priors in structure learning algorithms. An expert-provided network structure can serve as a prior in DBN learning algorithms.

Unfortunately, specific parameter values or even the structure of a network may be especially difficult to elicit from an expert. However, other assumptions provided by an expert can be helpful. For example, an assumption that some faults should have no parent nodes, or that the current state of a variable affects the next state of that or another variable, may be enforced regardless of what the structure learning algorithm suggests. A structural search algorithm is explained in Section 3.2.

Inclusion of expert opinion in BNs has been explored in the literature, particularly with the aim of determining a structural prior. Leach [19] aggregates expert opinion, Richardson and Domingos [21] combine opinions from multiple experts, while Langseth et al. [20] have experts extract information from existing structures.

2.5.2 Reliability Data

In some instances, reliability data may be available for certain system components. In this chapter, reliability data is used to establish prior distributions for specific system faults that are then updated using operational data or laboratory data. Reliability data may take on several forms including mean time between unscheduled removal (MTBUR) and mean time between unscheduled failures (MTBUF) or more simple measures such as failure rate and failure mode distribution. Better data types yield better modeling results but may be more difficult to obtain, as in the case of obtaining MTBUF instead of MTBUR [17]. MTBUF is used in this chapter.

The Reliability Information Analysis Center (RIAC) provides databases of many electronic and non-electronic parts. Generally, these databases include average failure rate data (NPRD-95) and failure mode distribution data (FMD-97) [55]. Estimates of component lifetime are modeled as exponential random variables. Confidence intervals for exponential distributions are typically calculated using the Chi-squared distribution. However, the data sets from which the exponential distribution parameter (failure rate) is calculated are non-homogenous, so the Chi-squared distribution cannot be used to determine confidence bounds. RIAC indicates that the published average failure rate may be treated as a random variable, the natural logarithm of which, for all components in the catalog, is normally distributed with a standard deviation of 1.5.

Consider a discrete variable $A(k+1)$ with parent $A(k)$. If $A(k+1)$ is a fault indicator variable for fault A, when $A(k+1) = 0$ fault A has not occurred. When $A(k+1) = 1$, fault A has occurred. In the simplest case, $A(k+1)$ only has one parent, $A(k)$. Thus, $A(k+1)$ has a conditional probability as in Table 2.

Table 2. Conditional probability table for A(k+1)

state	$\Pr(A(k+1) = 1 \mid \text{state})$	$\Pr(A(k+1) = 0 \mid \text{state})$
$A(k) = 1$	$p_{1,1}$	$p_{1,2}$
$A(k) = 0$	$p_{2,1}$	$p_{2,2}$

The prior $p_{i,j}$ may be determined using functions of time that utilize known failure rates and can be updated when laboratory or operational data is available (operational and laboratory data are discussed in Sections 2.5.4-2.5.5). These functions of time are typically time to failure CDFs, often characterized by the exponential or Weibull distribution.

If possible, reliability data may be updated with operational data for the system in question. If the time to failure (TTF) is described by a Gamma family distribution, such as the exponential distribution, a conjugate prior is available and Bayesian updating is straightforward. Updating is

also possible with Weibull parameters. However, the two parameter Weibull distribution does not have a natural conjugate prior, making Bayesian updating complicated. However, a solution has been proposed by Kaminskiy and Krivtsov [56]. More generally, if a prior distribution over TTF is available, Markov chain Monte Carlo (MCMC) may be used to update the distribution.

The assumption of parameter independence of a Bayesian network [5] makes it possible to update the conditional probability distributions of individual variables in the network without affecting the distributions of other variables in the network. Parameter independence means that the values of the distribution parameters of one variable in the network do not depend on the values of another distribution parameter. This is a key assumption in updating distribution parameters based on reliability data.

Some complications may arise during the process of determining $p_{i,j}(k+1)$. The network structure must be established before any $p_{i,j}(k+1)$ may be calculated from reliability data. The number of rows in a conditional probability table of a binary variable is 2^n (where n is the number of parents). The fault in question may have additional causes, such as other faults, resulting in a larger number of probabilities to estimate. The question is, which probabilities can be estimated with the given reliability data. If the conditions under which the reliability data are collected are known, the appropriate $p_{i,j}(k+1)$ may be calculated. However, these conditions are generally unknown and it is difficult to know which conditional probability to update. Naively, it can be assumed that the data applies to all conditions. For example, in the cantilever beam, unless otherwise specified, time until crack occurrence could be assumed to have been collected under conditions where either damage at the support does or does not exist.

2.5.3 Mathematical Models

Mathematical models derived from first principles (e.g. finite element or bond graph) or empirical relationships (neural network, regression, BN/DBN) are used for generating predictions about system operation and for simulating reliability data (e.g. failure rates).

The mathematical model's advantage is the ability to include system variables and faults that are not directly observable. For example, damping may not be measureable but a suitable expression may be available to estimate it. Another situation is the case of an unobservable fault. Fault identification models may be able to determine the presence of that fault. With estimates available for the unobservable system variables or faults at a particular time, the data case at the particular time can be completed. In Table 3 in Section 2.5.4, this means that if A(k) is an unobservable fault for which no data can be collected, estimates obtained from a fault identification model for A(k) may be used in lieu of actual observations.

Some models may provide insight into the structure of the DBN. For example, in the approach of Roychoudhury et al. [14], the structure of the network is derived from the bond graph model of the system. In addition, the qualitatively derived fault signatures, which indicate increases or decreases in model parameters due to faults, provide a means of checking the behavior of the DBN.

In the context of this chapter, one process for integrating a mathematical model into a DBN is as follows. First, generate predictions of the states of the variables in the model. Organize the predictions as operational data as per Section 2.5.4. The model predictions are then included in the DBN following the method for operational data. A key benefit of using a mathematical model in this way is the ability to model system variables that are not directly observable. Further, mathematical models can provide data on system failures that are too expensive for repeated experiments.

An alternative use of a mathematical model is to evaluate the reliability of a subsystem. Limit state functions can be used to estimate the time-dependent failure probability of a system [57].

Eqn. 3 defines the i^{th} limit-state function for a system with m failure modes ($i = 1 \ldots m$).

$$g_i(\boldsymbol{X}) = a_i \quad \textbf{Eq. 6}$$

where $g_i(\boldsymbol{X})$ are the system responses, $\boldsymbol{X}$ are system inputs, and a_i are threshold values for the system response which defines failure. When $g_i(\boldsymbol{X}) \leq a_i$ for $i = 1, 2, \ldots, m$, the system is in a safe state. When any $g_i(\boldsymbol{X}) > a_i$, the system is said to have failed. $p_f = \bigcap_{i=1}^{m} Pr(g_i(\boldsymbol{X}) > a_i)$ is the failure probability of a parallel system and $p_f = \bigcup_{i=1}^{m} Pr(g_i(\boldsymbol{X}) > a_i)$ is the failure probability for a series system. For a static system, finding p_f reduces to analysis using Monte Carlo or first/second order reliability methods (FORM/SORM) [58].

For a dynamic system, Eq. 6 becomes time-dependent, and it is desired to find the probability of the response staying within the safe region over a duration of time. Extensive work has been done in time-dependent reliability, e.g. Li and Melchers [57], Mahadevan and Dey [59], Kuschel and Rackwitz [60].

A second process for integrating a mathematical model into a DBN is as follows. First, using the system model, obtain $p_f(t)$ from time-dependent reliability analysis. $p_f(t)$ is then directly included in the DBN as per Section 3.4. Both processes for integrating a mathematical model into a DBN are shown in Figure 2.

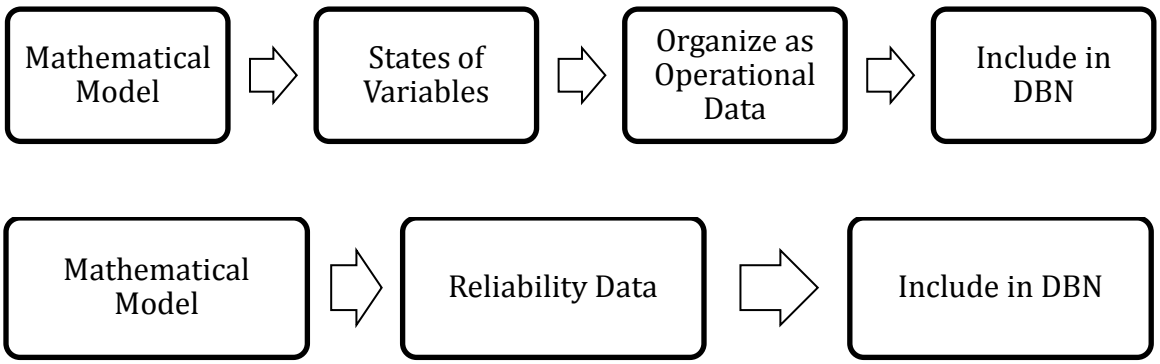


Figure 2. Processes for integrating a mathematical model into a DBN

2.5.4 Operational Data

Operational (observational) data is data collected through passive observation of a system during its operation. The system is allowed to operate naturally without external intervention. Operational data provides significant insight about the causal structure of a system and the conditional probability relationships between variables. Most BN and DBN learning methodologies rely heavily on operational data, practically making it a prerequisite for learning, although exploiting other types of information could potentially reduce the exclusive dependence on operational data.

In order to integrate operational data into the DBN, the data is organized in a time-lagged format as shown in Table 3, using data drawn from a network consisting of binary variables A and B and a ternary variable C spanning time k and $k + 1$. Although the example shown here uses discrete variables, the variables could also be continuous (e.g. Gaussian) or a combination of discrete and continuous [61]. Each row in Table 3 is a data case and the collection of data cases is the database, D, a m by n matrix where m is the number of data cases and n is the number of variables in the DBN. More specifically, a data case consists of realizations of all the system variables corresponding to a discrete value of time. Note that the last data case is incomplete and may be discarded because

there is no data at $k_{max} + 1$. If the database is to be constructed using data from multiple time series, each time series should be lagged and truncated independently before being included in the database.

Table 3. Time-lagged database

Case	A(k)	B(k)	C(k)	A(k+1)	B(k+1)	C(k+1)
k=1	0	0	0	0	1	1
k=2	0	1	1	0	1	1
k=3	0	1	1	1	1	1
k=4	1	1	1	0	1	2
k=5	0	1	2	1	1	0
k=…						
$k=k_{max}$	0	1	0	-	-	-

Learning Bayesian network parameters

For a defined BN structure, a BN has parameters θ consisting of the distribution parameters of each conditional probability distribution. These parameters may be estimated using either maximum likelihood estimation or Bayesian estimation [5].

Learning Bayesian network structure

A considerably more difficult problem than learning network parameters is learning the network structure. Given the database, the problem is to determine the network structure with the largest likelihood (out of many possible candidate network structures) and the distribution parameters of that network. This amounts to an optimization problem, and it is usually approached with heuristic optimization procedures.

One early algorithm for finding the best network structure is the K2 algorithm by Cooper and Herskovitz [62]. In K2, the N nodes (variables) of the joint probability space are arranged in a fixed order. The i^{th} node in the ordering can have any of the following nodes in the order as parents (but

none of the preceding nodes). For example, the set of nodes in Table 3 is {$A(k)$, $B(k)$, $C(k)$, $A(k+1)$, $B(k+1)$, $C(k+1)$}. Their respective position in the ordering could be {4 5 6 1 2 3} or {5 4 6 3 1 2} but not {1 5 6 4 3 2}, because a possible parent of $A(k)$ cannot be $A(k+1)$.

The K2 algorithm finds the best network structure by first determining the local structure of each node and then amalgamating the local structures to determine the global network structure. Given the i^{th} node in the node ordering, the K2 algorithm compares various candidate sets of parent nodes of i using a scoring function (e.g. the Bayesian Information Criteria (BIC) [5]) and selects the highest scoring candidate set given the database as the local structure for node i. To avoid overfitting the data and decrease computational effort, the maximum number of parents a variable may have can be limited.

Suppose it is desired to find the parents of $C(k+1)$. c_1 = {$C(k)$ $B(k)$} is one candidate parent set and c_2 = {$A(k)$ $B(k)$} is a second candidate parent set. If score(data|c_1) > score(data|c_2), c_1 is the best of the two candidate sets and must be compared against any other possible parent candidate sets for node $B(k+1)$. In order to estimate the score, the max likelihood of the parameters of conditional probabilities $P(C(k+1)|c_1)$ and $P(C(k+1)|c_2)$ must be determined. For example, the distribution parameters $p_{i,j}$ of $P\big(C(k+1)\big|A(k), C(k)\big)$ are shown in Table 4. Because $C(k+1)$ is ternary, it has three possible values, 0, 1, and 2. The distribution parameters $p_{i,j}$ define the probabilities of $C(k+1)$ = 0, $C(k+1)$ = 1, and $C(k+1)$ = 2 given some state of the parent variables, $A(k)$ and $C(k)$.

Table 4. Conditional probability table for C(t+1) with structure $B(t) \rightarrow C$(t+1) $\leftarrow C$(t)

Parent state	$\pi(C(k+1)=0 \mid$ *parent state*$)$	$\pi(C(k+1)=1 \mid$ *parent state*$)$	$\pi(C(k+1)=2 \mid$ *parent state*$)$
$B(k) = 0, C(k) = 0$	$p_{1,1}$	$p_{1,2}$	$p_{1,3} = 1 - p_{1,1} - p_{1,2}$
$B(k) = 1, C(k) = 0$	$p_{2,1}$	$p_{2,2}$	$p_{2,3} = 1 - p_{2,1} - p_{2,2}$
$B(k) = 0, C(k) = 1$	$p_{3,1}$	$p_{3,2}$	$p_{3,3} = 1 - p_{31} - p_{32}$
$B(k) = 1, C(k) = 1$	$p_{4,1}$	$p_{4,2}$	$p_{4,3} = 1 - p_{4,1} - p_{4,2}$
$B(k) = 0, C(k) = 2$	$p_{5,1}$	$p_{5,2}$	$p_{5,3} = 1 - p_{5,1} - p_{5,2}$
$B(k) = 1, C(k) = 2$	$p_{6,1}$	$p_{6,2}$	$p_{6,3} = 1 - p_{6,1} - p_{6,2}$

To find $p_{1,1}$, the database is sorted to find all cases where $B(k) = 0$ and $C(k) = 0$. If there are N_1 such cases and if $C(k+1) = 0$ is observed $n_{1,1}$ times out of those N_1 cases, then $p_{1,1} = n_{1,1}/ N_1$. Of course, strength of belief in the values $p_{i,j}$ varies with the size of the database. To account for this, the $p_{i,j}$ are treated as a r-dimensional Dirichlet random variables for fixed i, e.g. $p_{1,j}$ is a $r = 3$ dimensional Dirichlet random variable. The Dirichlet distribution is

$$f(p_{i,1} \dots p_{i,r}; n_{i,1} \dots n_{i,r-1}) = \frac{1}{B(\boldsymbol{n}_{i,j})} \prod_{j=1}^{r} p_{i,j}^{n_{i,j}-1} \qquad \textbf{Eq. 7}$$

where $\boldsymbol{n}_{i,j} = \{n_{i,1}, n_{i,2}, \dots n_{i,r}\}$, B is the Beta function, $p_{i,1}, \dots p_{i,r-1} > 0$, $p_{i,1} + \cdots + p_{i,r-1} < 1$, $p_{i,r} = 1 - p_{i,1} - \cdots - p_{i,r-1}$, and i corresponds to a fixed parent state. For $r = 2$, the Dirichlet distribution is the Beta distribution. The Dirichlet distribution is widely considered suitable for modeling probabilities, as it bounds each probability between 0 and 1 and allows for a reasonable amount of flexibility. For discrete variables, use of the Dirichlet distribution to account for the uncertainty of the parameters and updating of the parameters using $p_{i,j}$ estimates from data as in Table 3 and Table 4 is standard practice [5].

After each iteration of the structure learning algorithm, the local structure for node i is noted in a global structural matrix, S, with rows $i = 1 \dots N$ and columns $j = 1 \dots N$. $S(i,j) = 1$ when node i (represented by the i^{th} row in $S(i,j)$) is a parent of the node j (represented by the j^{th} column in $S(i,j)$), as determined by the local structure of i and $S(i,j) = 1$ and is zero otherwise. For example,

assume nodes $\{A(k),\ B(k),\ C(k),\ A(k{+}1),\ B(k{+}1),\ C(k{+}1)\}$ with node orderings {4 5 6 1 2 3} respectively. Initially, S is the 6x6 identity matrix. If K2 first finds that $A(k)$ and $B(k)$ are the only parents of $A(t+1)$, then $S(4,1) = 1$ and $S(5,1) = 1$. After the next iteration, if $C(k)$ and $B(k)$ are found to be the only parents of $B(k+1)$, then $S(4,2) = 1$ and $S(3,2) = 1$.

Due to the large number of parent candidate sets encountered throughout learning a network, the distribution parameters of each are generally not kept in memory. Thus, it is necessary to re-estimate the distribution parameters for the network after learning the structure.

More recent hill-climbing algorithms allow arc addition, arc reversal, and arc removal operations [5]. They may also incorporate TABU lists, which keep track of previously tested local structures [63]. Alternative approaches to hill-climbing/greedy algorithms are constraint based methods, which test for conditional independence between variables [64]. Hybrid methodologies such as max-min hill-climbing (MMHC) [65] combine both approaches. For hybrid networks (networks with continuous and discrete variables), approaches such as MMHC [65]. MMHC is also suitable for large networks with thousands of variables.

Although the structure learning algorithm produces a single best structure as output, it must be noted that operational data generally cannot uniquely define the causal structure of a BN. That is, another structure may exist which explains the data equally well. This is evident by considering a basic network composed of variables *A* and *B*. From the same set of data, the networks $A \rightarrow B$ and $B \rightarrow A$ both can represent the same assertions of conditional independence. Their joint distributions are the same and the networks can be said to be equivalent [2], [61], [66]. An equivalence class of network structures may be said to exist. Heckerman et al. [2], [61] call this property likelihood equivalence. The property must always hold for acausal networks of variables with a generalized

probability distribution function and is often found to hold for causal networks of variables with a generalized probability distribution function [61]. Heckerman et al. [2], [61] consider the use of pure operational data as an assumption of likelihood equivalence. In developing a model for prognosis, this means that the use of pure operational data (the likelihood assumption) may not reveal the underlying causal structure of a system which best explains system behavior.

One practical concern associated with operational data is how well it captures system behavior under the influence of faults. If a fault does not occur and is therefore not observed, its effects on the system cannot be observed. Since faults are (hopefully) rare occurrences, it could require a large amount of operational data to encounter the fault and learn how it affects the BN structure. Another practical concern is that operational data needs to be used for model validation. Because of this, some operational data should not be used to train the model and instead saved for model validation.

2.5.5 Laboratory Data

Laboratory (experimental, interventional) data is obtained while observing the system under outside intervention. Outside intervention comes in the form of fixing the value of one or more of the system or input variables. An intervention shows how a particular variable assuming a particular value affects the behavior of the system but does not provide any insight as to how likely that variable is to naturally assume a particular value. For instance, in a laboratory setting, the effect of a loose bolt on the resistance provided by a structural system would be discovered by loosening a bolt and then observing the system. The frequency of the bolt loosening, however, cannot be established. Combined with knowledge of the frequency of the bolt loosening or using a variety of experiments, the laboratory data could provide enough information to learn the system model without operational data.

Integration of laboratory data requires creating a formatted database, as with operational data. A database of operational and laboratory data may combined into one database. Laboratory data, additionally, requires information as to which variables are fixed and for which data case. Other non-fixed variables in the system, including children of the fixed variable, may be characterized by experimental data.

To track which data points have been fixed through experiment, an indicator matrix L with dimensions equal to the size of the database, D, is constructed. If a value $L(i, j)$ has been fixed experimentally, $L(i, j) = 1$. Otherwise, $L(i, j) = 0$. When $L(i, j) = 1$, the corresponding value in the database $D(i,j)$ is useful for determining the structure of the network and the distribution of the downstream variables. The use of L is to block the data such that the conditional probabilities are properly calculated.

Consider the database in Table 3. If the value of $B(k)$ is fixed at 1 in the 2nd and 3rd data cases, L is given by Table 4. Say it is desired to find parents of $C(k+1)$ as in Section 3.2. Once a candidate parent state is fixed, the distribution parameters of $C(k+1)$ are calculated without regard for how often that parent state occurs. Thus, fixed variables in laboratory data can help determine the structure of the network and downstream conditional probabilities.

Table 4. Matrix L indicating which values have been experimentally fixed

Case	A(k)	B(k)	C(k)	$A(k+1)$	$B(k+1)$	$C(k+1)$
k=1	0	0	0	0	1	0
k=2	0	1	0	0	1	0
k=3	0	1	0	0	0	0
k=4	0	0	0	0	0	0
k=5	0	0	0	0	0	0
k=…						
k=k_{max}				-	-	-

In some experiments, a variable may be fixed for part of the duration. Consider a dynamic system operating in time $k \in [0\ \infty)$ with a perfect intervention where $x_i = k$, beginning at time t_I. The intervention lasts as long as $x_i = k$ is fixed, say from k_I to k_{If}. Under appropriate environmental conditions, the data collected on the interval $[0\ k_{If})$ may be considered operational. However, the data collected after k_{If} must be treated as laboratory data until any transient effects of the intervention on the system disappear.

2.5.6 Summary

The integration of heterogeneous information into a unified system model is discussed in this section. The procedures provided show how information from a variety of sources and in varying formats can be leveraged in a model suitable for SHM applications. The framework provides a flexible workflow for construction of the system model. Figure 3 describes a possible workflow for constructing the system model. Typically, expert opinion is the starting point for determining basic assumptions about the network, such as whether to use all discrete variables, all continuous variables, or a hybrid network. Once the basic assumptions are in place, published reliability data is drawn upon to determine probabilities to model the occurrence of faults in the system. Mathematical models are then selected and used to generate prior data in formats similar to operational, laboratory, or reliability data. Finally, operational and laboratory data are used to update the structure (as necessary) and conditional probabilities of the network. The availability of information may affect the workflow. For example, if plenty of operational data is available, defining mathematical models to generate prior data may be an unnecessary second step. The process is iterative, requiring the system designer to tune and refine the network multiple times, with a high percentage of correct diagnosis and accurate damage quantification being the primary goal.

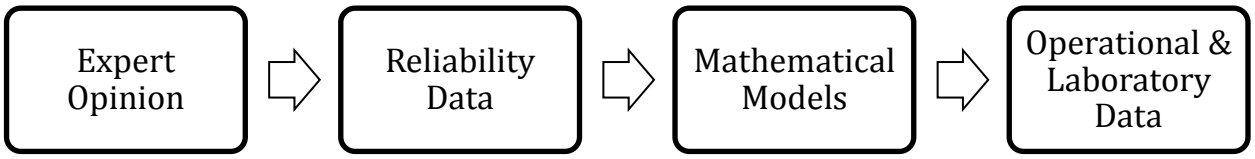


Figure 3. Possible workflow for integrating heterogeneous information

Several caveats about the methodology are worth emphasizing. A great deal of uncertainty is associated with various types of information from multiple sources. The environmental conditions under which different information types are collected may vary widely and fail to match the conditions under which the system ultimately will operate. Testing methods of varying quality may have been used to generate data. Important information about failure modes may be missing. An expert may supply erroneous domain knowledge. On a positive note, using diverse information types and sources may be seen as a way to minimize the risk of producing an unacceptable model. With non-diversified information sources, if information from a particular source or of a particular kind is corrupt, it follows that a model constructed from this information will also be corrupt.

Learning a DBN is computationally expensive. It can require a large amount of observational and experimental data, especially when the system contains many variables and possible faults. Although many algorithms for learning DBNs are available, none are cheap. Inference in DBNs can also be a large computational burden but can be eased with approximate inference methods such as particle filters. The sparsity of the DBN representation of the joint probability space of the variables also helps to alleviate the computational burden. Further, by using expert opinion to determine the network structure of a subset of the variables, the computational burden of structure learning on the remaining variables is drastically less than if structure learning were performed over all variables.

Key advantages of the learned model are in risk and reliability analysis, diagnosis, and prognosis i.e. its forward and backward reasoning abilities, and its ability to transform various types of information into a usable format. The use of experimental data and reliability information helps ensure that the model can successfully simulate faults and aid in their diagnosis.

2.6 Example Problem

A hydraulic actuator system is considered to demonstrate the proposed methodology. Such a system is often used to manipulate the control surfaces of aircraft. The system consists primarily of three subsystems: a hydraulic actuator, critical center spool valve, and an axial piston pump (Figure 4). Using procedures explained in Section 3, expert opinion, operational data, laboratory data, reliability data, and mathematical behavior models were used to construct a DBN model of the spool valve and hydraulic subsystems.

First, expert opinion is invoked to establish an initial DBN and priors for the conditional probability distributions. Next, the mathematical model of the system is used to generate predictions of the system variables. The predictions are treated as operational and laboratory data. Then, a structure learning algorithm determines the remaining structure of the network. Finally, the parameters of the network are estimated and reliability data updated using data from the system model.

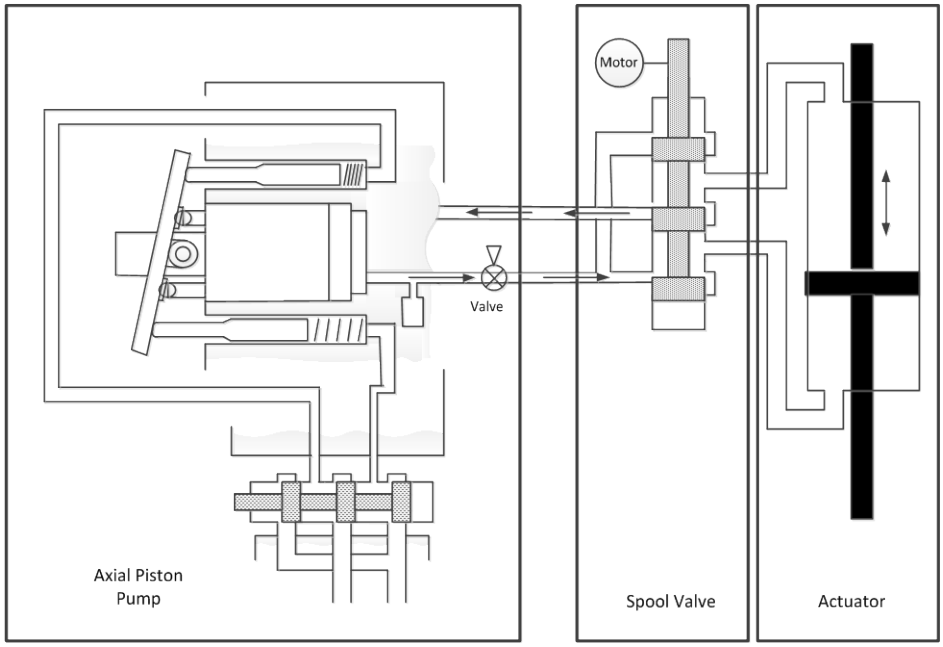


Figure 4. Hydraulic Actuator System

2.6.1 Expert Opinion

Expert opinion must be considered first to define the basic parameters of the problem. A DBN representation of the system was chosen because heterogeneous information sources were available, the intended use of the model is diagnosis and prognosis, and the system is dynamic. Seven state variables and six discrete faults were selected to model the behavior of the system.

A generic initial structure for the DBN is first selected (Figure 5) based on expert opinion. This generic two time slice structure consists of the set of faults, $\boldsymbol{F}$, model parameters, $\boldsymbol{\theta}$, system state, $\boldsymbol{y}$, and measurements, $\boldsymbol{z}$. In this structure, faults cause changes in system parameters, which then cause changes in system responses, which are observed. $\boldsymbol{F}$ contains the persistent variables in the DBN – their future values depend upon their present values. The observations, $\boldsymbol{z}$, while not connected across time slices, are nonetheless *not* independent across time slices, but correlated via $\boldsymbol{\theta}$.

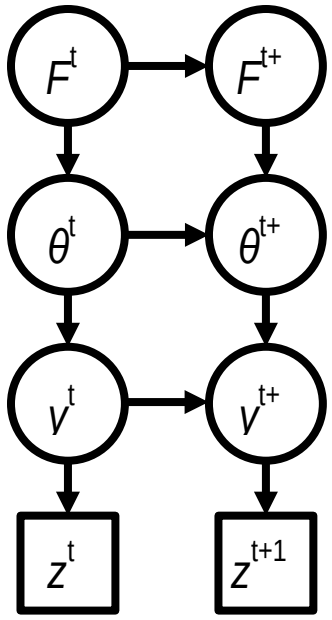


Figure 5. Generic DBN structure

Table 5. List of faults and affected parameters

Fault	Parameter Affected
Electrical Fault	Control Signal
Seal Leak	Leakage Area
Water Leak into System	Hydraulic Fluid Bulk Modulus
Air Leak Into System	Hydraulic Fluid Bulk Modulus
Pressure Valve Malfunction	Supply Pressure
Pump Fault	Supply Pressure

Table 5 lists the faults considered in the actuator system and the parameter affected by that fault (the faults are described further in Section 2.6.2). For each fault, a binary variable is added to the network at time k and $k + 1$. The parameters are assumed to have Gaussian distributions, whose mean and variance depend on the health state of the system. The leakage area parameter is a

special case, as wear and leakage are assumed to be present at all times. The discrete variable for leakage area indicates that the leakage area has increased beyond some threshold value.

Parameters from the current time step and initial conditions from the previous time step are input into a physics-based model of the actuator (see Section 2.6.2), which estimates the system responses, assumed to be Gaussian variables. Measurements are then connected to the corresponding system response. Links are also drawn between like faults at time k and $k + 1$ and like parameters at time k and $k + 1$. Finally, a Gaussian variable is added at time k and $k + 1$ for each measurement available. The resultant DBN is shown in Figure 6 with parameters described in Table 6.

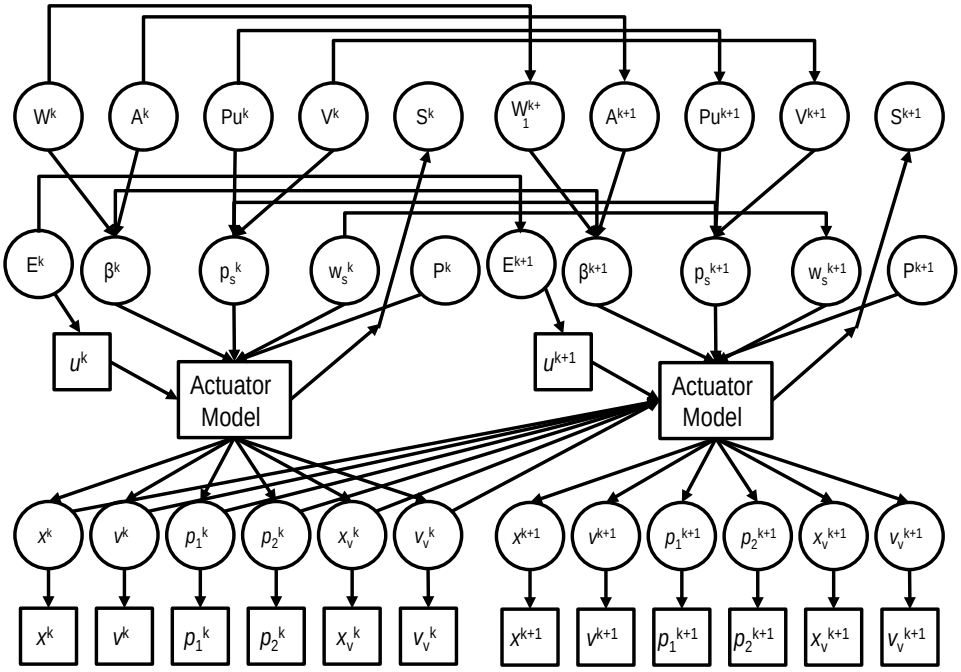


Figure 6. Initial DBN structure as a result of expert opinion

Table 6. DBN Variables

State Variable	Symbol	Unit	Type	Note
Actuator position	x_{act}	m	continuous	measured
Actuator velocity	v_{act}	m/s	continuous	measured
Pressure in chamber 1	P_1	Pa	continuous	measured
Pressure in chamber 2	P_2	Pa	continuous	measured
Valve position	x_{valve}	m	continuous	measured
Valve velocity	v_{valve}	m/s	continuous	measured
Control signal	u	V	continuous	input
Water leak	W	-	binary	inferred
Air leak	A	-	binary	inferred
Pump sensor fault	P	-	binary	inferred
Valve fault	V	-	binary	inferred
Seal leak	S	-	binary	inferred
Electrical fault	E	-	binary	inferred
Fluid bulk modulus	β	MPa	continuous	inferred
Supply pressure	p_s	MPa	continuous	inferred
Wear rate	w_s	mm^3/Nm	continuous	inferred

As a precursor to the data-based portion of the methodology, assumptions were made to facilitate the learning process. The intra-slice arcs on the graph were assumed to be the same in slice k and $k + 1$. Because the system responses are conditionally independent given the actuator model, the structure learning was not performed between the response variables. Finally, the actuator model parameters are assumed to be independent. This saves a great amount of computational expense compared to considering all system variables, as only relationships between the binary fault variables need to be explored.

This step of model construction involves making many choices about tunable model and algorithmic parameters. These choices are made based on data which is not used for parameter and structure learning.

2.6.2 Published Reliability Data

The DBN model of the system should be able to simulate multiple faults and extrapolate system behavior multiple steps into the future for the model to be a useful diagnosis and prognosis tool. The overall failure rate for an actuator may be determined by estimating the base failure rate and making empirical corrections for temperature and fluid contamination [67]. The RIAC Databook [55] and the Handbook of Reliability Prediction Procedures for Mechanical Equipment [67] give failure rates for many mechanical systems. For illustration of the methodology, a handful of the possible faults for the actuator system are considered in this chapter. Table 7 lists the faults that have been considered, the subsystem where they are located, the internal failure rate used in the simulation model, and the information source for that fault.

An electrical fault that can interfere with electronics is mentioned in [67]. The exponential distribution is used to model this fault. It is assumed that this fault can be modeled by adding noise to the input signal.

The actuator seal leak is mentioned by Sepeheri et al. [68] and results in hydraulic fluid passing between chambers of the actuator, thereby reducing its performance.

Air and water leaks are common problems which change the effective bulk modulus of the hydraulic fluid as is a pressure relief valve malfunction [68]. The failure rates for these faults are estimated by multiplying the base failure rate for a hydraulic actuator by the percentage of faults considered hose failures (6.1%).

The pressure relief valve is used to adjust the supply pressure of the hydraulic fluid after it leaves the axial piston pump. A fault with this valve is assumed to change the supply pressure provided to the servovalve.

The pump has a pressure sensor used to modulate its output [67]. It is assumed that an electrical fault could result in malfunctioning of this sensor, resulting in a change in supply pressure provided to the servovalve.

The failure rates were then used to calculate probabilities as in Section 2.5.2. These probabilities are adjusted with time and override the maximum likelihood parameters calculated from laboratory and operational data.

Table 7. Faults Considered

Fault	Subsystem	Information Source	Failure Rate
Electrical Fault	Electronics	RIAC Databook [55]	1.510 failures /10^6 hours
Seal Leak	Actuator	RIAC Databook [55], Literature [68]	NA
Water Leak into System	Piping/Fittings	RIAC Databook [55], Literature [68]	2.736 failures /10^6 hours
Air Leak Into System	Piping/Fittings	RIAC Databook [55], Literature [68]	0.167 failures /10^6 hours
Pressure Valve Malfunction	Pressure Valve	RIAC Databook [55], Literature [68]	15.575 failures /10^6 hours
Pump Pressure Sensor	Piston Pump	Mathematical Model [69], Literature [70]	103.567 failures /10^6 hours

2.6.3 Mathematical Behavior Models

Mathematical models are used for modeling the load, actuator, and seal leakage area. These mathematical models are used to generate model predictions for single fault and no fault scenarios.

A nominal (healthy) set of predictions is also generated to estimate the residuals of the system responses.

The hydraulic actuator is described in Eq. 8 - Eq. 20 following Kulakowski et al. [71] and Thompson et al. [72]. Variables and parameters for the system are given in Table 8.

$$\dot{x}_{act} = v_{act} \quad \textbf{Eq. 8}$$

$$\dot{v}_{act} = \frac{1}{m}\left[(P_1 - P_2)A_{pist} - b_{act}v_{act} - k_{act}x_{act} - F_{ext}\right] \quad \textbf{Eq. 9}$$

$$\dot{P}_1 = \frac{1}{C_{f1}}\left(Q_1 - A_{pist}Q_2 + Q_{leak}\right) \quad \textbf{Eq. 10}$$

$$\dot{P}_2 = \frac{1}{C_{f2}}(V_2 x_{act} - Q_2 - Q_{leak}) \quad \textbf{Eq. 11}$$

$$\dot{x}_{valve} = v_{valve} \quad \textbf{Eq. 12}$$

$$\dot{v}_{valve} = a_1 v_{valve} + a_0 x_{valve} + b_0 e_{command} \quad \textbf{Eq. 13}$$

$$C_{f1} = \frac{V_1(x_{act})}{\beta} \quad \textbf{Eq. 14}$$

$$C_{f2} = \frac{V_2(x_{act})}{\beta} \quad \textbf{Eq. 15}$$

$$\text{If } x_{valve} > 0, \begin{cases} Q_1 = C_d w_{valve} x_{valve} sgn(P_s - P_1)\sqrt{\frac{2}{\rho}|P_s - P_1|} & \textbf{Eq. 16} \\ Q_2 = C_d w_{valve} x_{valve}\sqrt{\frac{2}{\rho}(P_2)} & \textbf{Eq. 17} \end{cases}$$

$$\text{If } x_{valve} < 0, \begin{cases} Q_1 = C_d w_{valve} x_{valve}\sqrt{\frac{2}{\rho}(P_1)} & \textbf{Eq. 18} \\ Q_2 = C_d w_{valve} x_{valve} sgn(P_s - P_2)\sqrt{\frac{2}{\rho}|P_s - P_2|} & \textbf{Eq. 19} \end{cases}$$

$$Q_{leak} = C_d a_{leak}\sqrt{\frac{2}{\rho}|P_2 - P_1|}sign(P_2 - P_1) \quad \textbf{Eq. 20}$$

Table 8. Model parameters and variables for a spool valve and a hydraulic actuator

Parameter/variable	Symbol	Nominal Value/ Unit
Actuator position	x_{act}	m
Actuator velocity	v_{act}	m/s
Servovalve position	x_{valve}	m
Servovalve velocity	v_{valve}	m/s
Pressure in chamber 1	P_1	Pa
Pressure in chamber 2	P_2	Pa
Combined mass of actuator and load	m_{act}	40 kg
Combined damping of actuator and load	b_{act}	800 Ns/m
Combined stiffness of actuator and load	k_{act}	10^6 N/m
Piston annulus area	A_{pist}	0.0075 m^2
Valve port width	w_{valve}	0.0025 m
Spool valve model coefficients	b_0	90 m/Vs^2
	a_0	360,000 $1/s^2$
	a_1	1/s
Hydraulic fluid bulk modulus	β	1000 MPa
Hydraulic fluid density	ρ	847 kg/m^3
Discharge coefficient	C_d	0.7
Supply pressure	P_{supply}	20 MPa
Chamber 1 volume	V_1	m^3
Chamber 2 volume	V_2	m^3
Chamber 1 fluid capacitance	C_{f1}	$m^3/(kg/s)$
Chamber 2 fluid capacitance	C_{f2}	$m^3s/(kg/s)$
Volumetric flow rate into chamber 1	Q_1	m^3/s
Volumetric flow rate out of chamber 2	Q_2	m^3/s
Externally applied force	F_{ext}	0 N
Input voltage	$e_{command}$	Sin(2*pi*t) V
Leakage volumetric flow rate	Q_{leak}	0 m^3/s
Leakage area	a_{leak}	0 m^2

The seal leakage area may be understood by considering the actuator cross section in Figure 7, the surface area of the seal is $(r_2^2 - r_1^2)$, where r_1 and r_2 are the inner and outer radii of the seal, respectively.

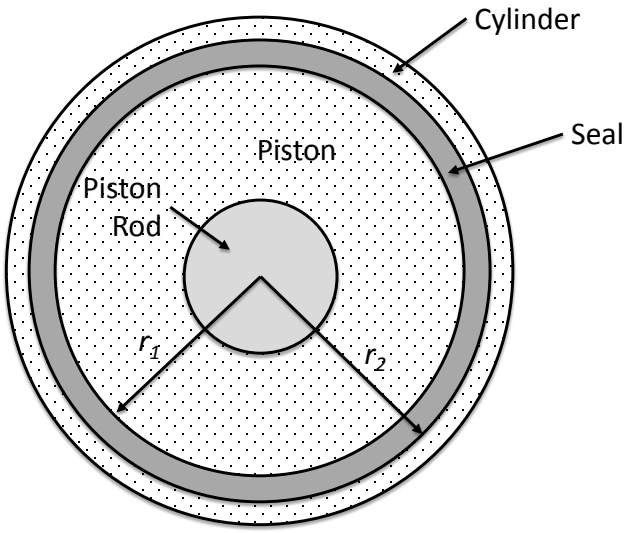


Figure 7. Actuator cross section

For demonstration of the prognosis methodology, the load is synthesized using an ARIMA (autoregressive integrated moving average) model, which is treated as a deterministic conditional probability distribution in the DBN. In reality the load on a flight control actuator is depends on many variables related to the dynamics of the aircraft and the desired flight path (for e.g. see Mahulkar et al. [73], Karpenko and Sepeheri [74], and McCormick [75]).

Finally, the physics of failure model for the seal leak is considered. The seal leakage area is modeled as in Section 2.2.

2.6.4 Operational Data

In this example, the operational data is synthetic, generated by the mathematical model in Section 2.6.3 with additive noise. The actuator system was simulated in the Matlab/Simulink environment

such that faults could randomly occur according to predefined failure rates (different from those provided as prior information for learning the system model). 25 sequences of operational data lasting 20 seconds each were simulated. All data, operational and laboratory, was collected at a sampling rate of 2 samples/sec.

Operational (observational) data is collected when the system is operating under field conditions with no external intervention or idealized loads or environmental conditions. Operational data can provide insight into network structure and parameters and is heavily relied upon in any BN or DBN learning methodology.

2.6.5 Laboratory Data

Laboratory data was simulated in the Matlab/Simulink environment. 10 simulation runs of 20 seconds for each of six single fault scenarios were performed. Fault times were introduced at random time instants. This resulted in 60 series of laboratory data. Data from experiments which was collected before the intervention occurred was treated as operational data.

Each series of operational and laboratory data was then individually lagged, allowing for analysis between time slice k and $k + 1$. The time series were then combined into one large data set to be fed to the learning algorithm. It is important that the time series be lagged separately first and then combined into one data set for training instead of being combined into one data set first and then lagged. If lagged after being combined, data mismatches will occur where one time series ends and another begins. Although structure learning was limited to one time slice, parameter learning still requires data over both time slices.

After generation of laboratory and operational data, the relationship between faults was explored. For each pair of faults i and j ($i \neq j$), the conditional probability $\theta_{ij} = Pr(fault_i = true | fault_j = true)$ was calculated, resulting in 30 probabilities (6 faults, 5 possible parents each). Large values of θ_{ij} indicate relationships between faults. The values of θ_{ij} greater than a heuristically determined threshold are then selected and arcs are added between the corresponding faults.

Once the full structure of the network is determined, the distribution parameters were filled in by calculating their maximum likelihood estimates from the data.

2.6.6 Results and Discussion

The result of this work is a DBN which can be used in diagnosis, prognosis, and decision making. The posterior DBN of the system is shown in Figure 8. The model clearly links faults to other faults and system responses, which is a direct result of using the laboratory data. Expert opinion has provided the links between faults and parameters as well as between time slices.

Changes in the bulk modulus do not manifest themselves in the available system measurements. Additional measurements or more advanced features may be required to handle these faults.

Not shown in Figure 8 are the conditional probability distributions for the DBN. Due to space limitations, only a sampling of the Gaussian and discrete conditional probability distributions are provided. Table 9 shows a Gaussian conditional probability distribution of the voltage amplitude at time *t*. Based on the results of the variance, it may be beneficial to train with more data on the fault scenario. Table 10 is a conditional probability table showing the effect of time on the expected conditional probabilities for the electrical fault.

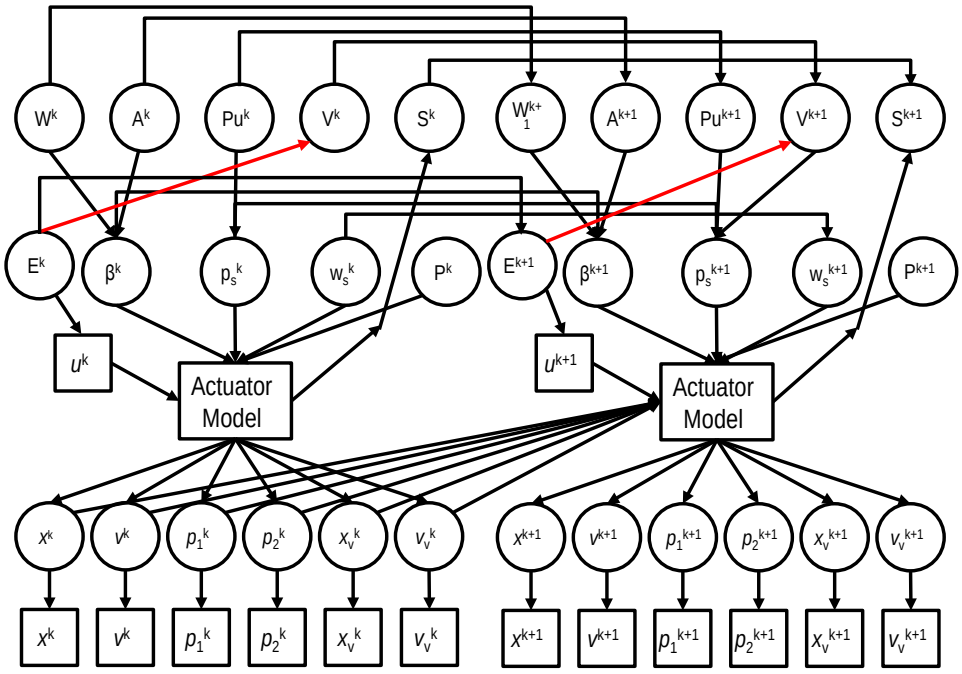


Figure 8. DBN with two time slices. Note: Red arrows indicate a learned connection.

Table 9. Gaussian distribution parameters for voltage amplitude at time *k*. Variance prior is 0.001

Parent State	Mean (Volts)	Variance
Electrical Fault = False	4.9993	0.007039
Electrical Fault = True	4.7500	0.001983

Table 10. Electrical fault (k+1) conditional probability tables at k = 0.25 and k = 15 sec

state	*Pr*(Elec Fault (*k*+1) = *False*|*state*)	*Pr*(Elec Fault (*k*+1) = *True*|*state*)	*state*	*Pr*(Elect Fault (*k*+1) = *False* | *state*)	*Pr*(Elec Fault (*k*+1) = *True* | *state*)
Elec Fault (*k*) = *False*	0.999723	0.000277	Elec Fault (*k*) = *False*	0.999136	0.000864
Elec Fault (*k*) = *True*	0	1	Elec Fault (*k*) = *True*	0	1

The effect of published reliability data is not apparent by observing the graphical structure of the DBN. However, Table 10 illustrates how a fault gradually becomes more likely, which would have implications in diagnosis and prognosis.

The model is evaluated in two steps. First, the model should be inspected for consistency with prior information about the system. Next, using a new set of test data, the performance of the model at its prescribed tasks may be evaluated. This may be repeated to establish, e.g. the model's diagnostic accuracy.

Because the structure of the actuator DBN is relatively simple, it is possible to manually inspect this network to determine if it is consistent with existing knowledge of the actuator system. Finding the structure in Figure 8 to be satisfactory, the model can be evaluated on its true objectives, diagnosis and prognosis.

Here, the diagnosis capabilities of the model are considered. In conjunction with a sequential importance resampling (SIR) particle filter [76], diagnosis using the DBN and measurements of an actuator with seeded faults and additive Gaussian noise has been tested. As expected, all faults except those related to fluid bulk modulus are diagnosed rapidly after occurrence (using 250 particles). Changes in bulk modulus do not tend to manifest themselves once the system is in a steady state and may require additional measurements to diagnose. Figure 9, Figure 10 and Figure 11 show the values of the discrete fault variables, the parameters, and observations for a valve malfunction at k = 6 sec. The diagnosis system is able to distinguish between faults affecting the same parameter and provide estimates of the magnitude of the fault. Maximum a posteriori (MAP) [76] estimates from the particle filter are shown where applicable.

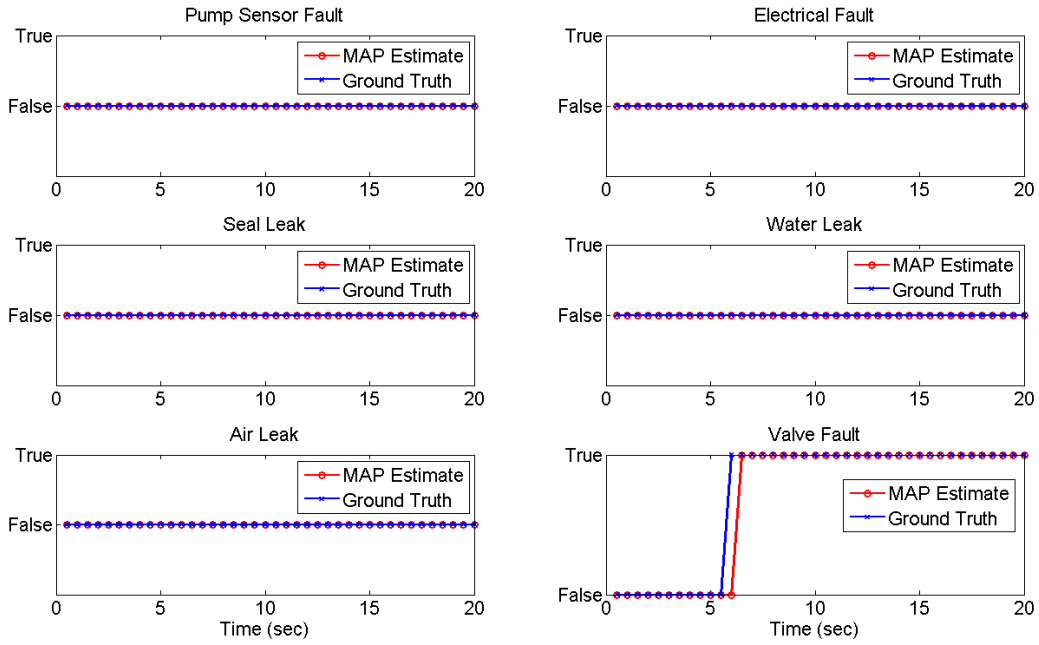


Figure 9. True fault value and particle filter MAP estimates

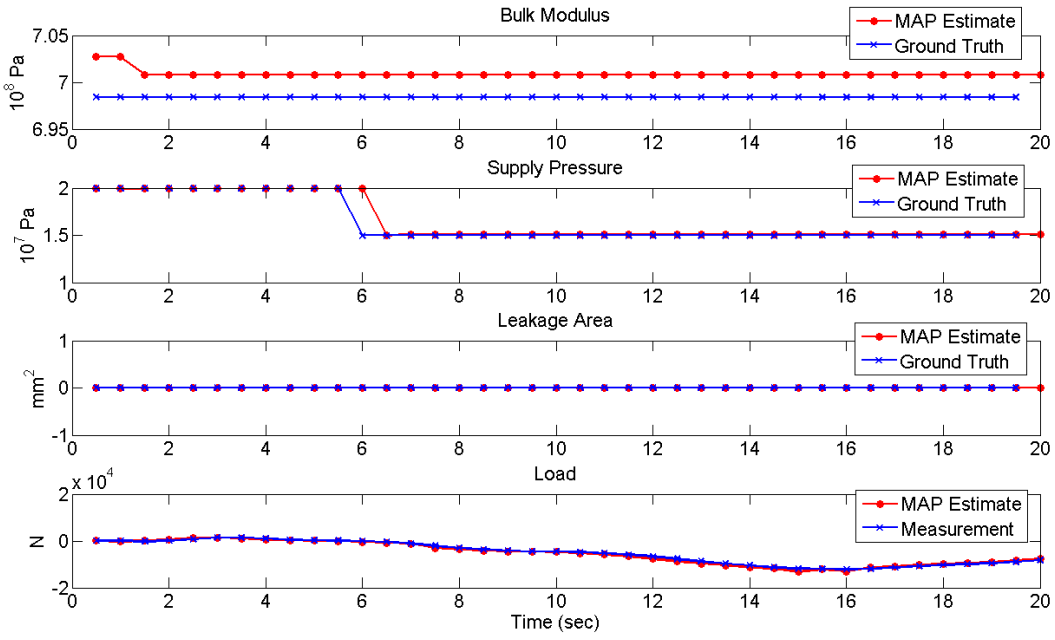


Figure 10. True parameter values and particle filter MAP estimates

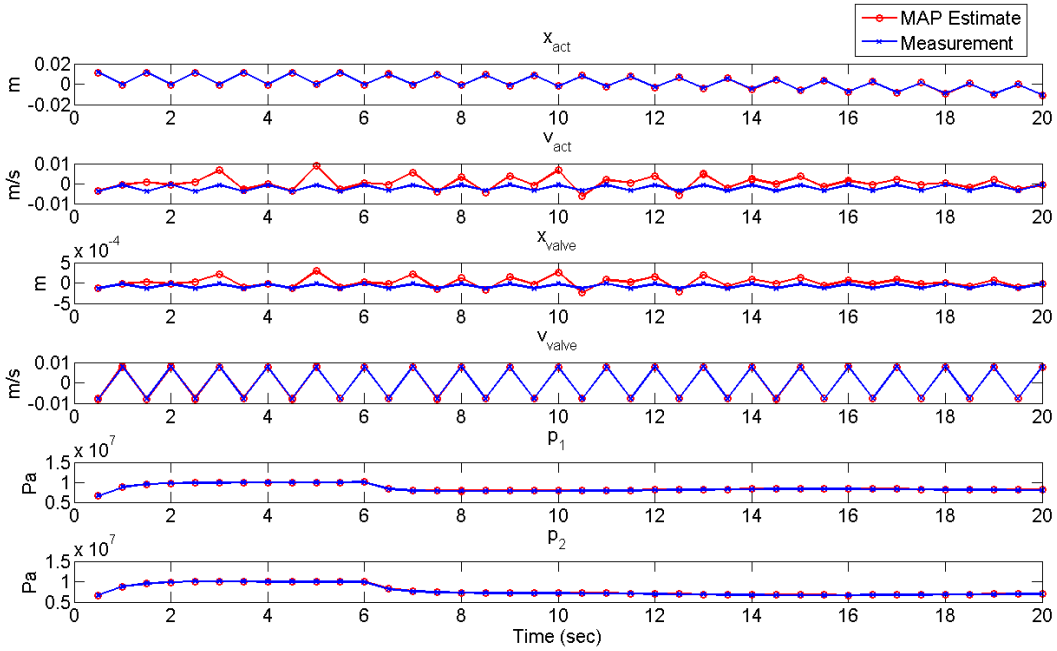


Figure 11. Observations used in fault detection

The effort to include heterogeneous information in the model appears to have resulted in an accurate diagnosis (and prognosis, and decision making) tool. Such accuracy, of course, may not always be possible. Faults may not have unique signatures (or any signature at all) and require additional measurements to be diagnosable. The use of synthetic, model-generated data may give much more predictable and cleaner data than could be expected from any true sensor measurement. Further, modeling assumptions about fault magnitudes parameter variance, or noise can strongly affect the performance of the system. Real-world data is still needed to test these assumptions. However, as the results are promising, ongoing work is focusing on utilizing the SHM model in quantifying diagnosis uncertainty, prognosis, and decision making.

Although the distribution types used affect the results of the methodology, these were used only for the sake of illustration and do not limit the methodology.

2.7 Conclusion

A methodology for building a dynamic system model for diagnosis and prognosis in the presence of heterogeneous information has been presented in this chapter. In particular, the methodology helps to learn the conditional probability relationships between system variables when the information available is heterogeneous. Several types of information (prior domain knowledge/expert opinion, operational and laboratory data, published reliability data, mathematical behavior models) have been considered in this chapter, and methods are developed for integrating them into a DBN model of a system. The methodology is general and may be applied to any domain. Here, a hydraulic actuator system was used to demonstrate the methodology.

In future work, it will be useful to address concerns related to computational effort and modeling assumptions. Computational effort reduction techniques, such as approximate inference or qualitative methods utilizing the DBN structure (e.g. temporal causal graphs), should hasten the application of the model to diagnosis and prognosis while making the method more computationally affordable. The use of hybrid networks where some variables have continuous distributions while other variables may have discrete distributions should be considered as many dynamic systems are composed of both continuous and discrete variables.

CHAPTER III.

3. System Damage Diagnosis with Heterogeneous Information

3.1 Introduction

3.1.1 Diagnosis Background

Diagnosis is the process of determining the root causes of abnormal system behavior based on knowledge and observation of the system. Diagnostic procedures have been variously described as a dichotomy between abduction or consistency-based methods [77], [78], models of correct behavior or fault models [13], classification methods or diagnostic reasoning strategies [79], traditional or model-based approaches [80], and experiential or first principles approaches [81]. The logic-based models of diagnosis, abduction and consistency-based reasoning, as discussed by Poole [82] and Console and Torasso [13], are used in this chapter.

Abduction methods use existing knowledge of faults and symptoms to determine which faults the observations imply. In abductive diagnosis methods where only faults and symptoms are known, possible faults, which are parameterized by values they depend on, are first hypothesized. Rules are then developed which show how symptoms (observations) are a consequence of the faults. These rules may be facts or merely indicate possible hypotheses. Under normality assumptions when faults and symptoms are unknown, the abduction method is the same except that the possible hypotheses consist of the system being faulty or normal. These hypotheses must be parameterized sufficiently to imply the observations.

Abduction methods include diagnostic testing (verification), fault dictionaries, rule-based methods, and decision trees. They are best when a system model is complete with all its faults known or when the system is very complex and difficult to model and basic observation of behavior to create associative diagnosis rules is easiest, a pure abduction method is beneficial. However, pure abduction has its difficulties. If the model happens to be incomplete, some reasonable explanations may be excluded. Diagnostic testing and fault dictionaries require faults be pre-selected. Rule-based methods and decision trees are system specific and difficult to update for new systems. Further, rule-based systems may be obsolete by the time the rules are learned. Decision trees offer an easy way to perform diagnosis but do little to explain how the answer was reached.

Consistency-based methods indicate faults by showing deviance from healthy behavior and need not enumerate every possible fault. In consistency-based methods with knowledge only of normal system behavior, for each component C which may be faulty, a hypothesis of C is not abnormal is given. Based on these normality assumptions, the correct behavior is predicted. When incorrect behavior is observed, it is concluded that an abnormality must have occurred. In consistency-based methods with knowledge of possible faults but no knowledge of normal system behavior, the negation of faults is the normal case. Rules are then developed which show how faults are implied by symptoms.

Consistency-based methods are best used when system models are incomplete and not all faults are known [13]. Additional benefits for these methods are described by Davis and Hamscher [83]. Models built from first principles can often be easily adjusted for new systems, decreasing the time for creating a diagnostic method and saving costs. Consistency-based methods using first principles models are also well-suited for containing structural information about the system in a useful format.

Console and Torasso [13] expand the dichotomous notion of logic-based diagnosis to a spectrum of diagnostic methods. More specifically, they define the general diagnostic problem as an abduction problem with consistency constraints. In this formulation, a diagnosis (explanation) must explain a group of observations while the behavioral model must indicate abnormal behavior consistent with all observations. The spectrum results from choosing the group of observations that the diagnosis must explain. By blending consistency-based with abduction approaches, the space of plausible solutions can be enlarged or shrunk for added flexibility [13]. Other approaches (de Kleer and Williams, 1989 [84]) use probabilistic information or exoneration knowledge (Friedrich et al., 1990 [85] to perform this function.

Hybrid consistency-based and abductive diagnosis methodologies often use consistency-based approaches for fault detection or symptom generation (although this can also be performed by diagnostic testing, inspection, etc.) then invoke an abductive methodology to isolate the fault. System models may further be used in quantification of the fault. Such a process is seen in Isermann [79], [86], Mosterman and Biswas [87], Moustafa et al. [88], Tandon and Choudhury [89], and Ding et al. [90]. Typical approaches in automated diagnosis rely on limit value checking of measurements, signal analysis, and process analysis to detect faults. Using symptoms, isolation of the fault is performed via classification and reasoning methods such as neural networks, fault-tree analysis, Bayesian networks, possibilistic reasoning, adaptive fuzzy systems, and backward chaining.

Another powerful tool is system modeling with dynamic Bayesian networks (DBNs). Bayesian networks (BNs) and dynamic Bayesian networks (DBNs) are statistical tools that can be used to integrate various sources of information. Due in part to this ability to integrate information, DBNs have been studied for many applications ranging from geo-location [38], maintenance decision making [91], and fault identification [92] to describing forest density [36]. A feature of previous

DBN-based studies in the literature is that the information used in constructing the DBN is typically from two or three sources. While this in many cases may provide acceptable descriptions of a particular system component or phenomenon, real systems contain many complex interactions, known and unknown, between subsystems, components, and phenomena that must be included in the DBN. By fusing heterogeneous from multiple information sources, these interactions may be discovered and quantified.

3.1.2 DBNs in Diagnosis

DBN-based diagnosis methodologies have gained traction in the literature, especially as computing power has increased. Arroyo-Figueroa and Sucar [93] provide an early example of performing inference in a DBN to diagnose faults. Jha et al. [92] use DBNs in identifying transient faults in a circuit. Data about correct system behavior was used to learn a DBN model of a healthy circuit. For each possible fault, a fault model based on expert knowledge is used to modify the DBN.

Lerner et al. [40] extract a DBN from bond graph models. The DBN is then used for diagnosis as part of a tracking and smoothing algorithm capable of handling burst and incipient faults. Their algorithm maintains the tractability of the problem by collapsing multiple hypotheses into one belief state. The primary drawback is that a model, fault modes, and how the fault modes affect the system are assumed to be known a priori.

Roychoudhury et al. [41] use a DBN, extracted from a bond graph and capable of modeling incipient faults, with a particle filter to detect faults in a two tank system. A qualitative procedure using the temporal causal graph given in Lerner et al. [40] narrows the number of fault candidates. The DBN model is then augmented and used to isolate the fault.

Other examples of DBNs in diagnosis (and prognosis) include Przytula and Choi [42] who use DBNs to obtain discrete qualitative measures of prognosis, McNaught and Zagorecki [91] who use a DBN of discrete variables to aid maintenance decisions, and Camci and Chinnam [94] who use hierarchical hidden Markov models for diagnosis and prognosis of machine parts.

As with data fusion, DBN-based diagnosis methodologies tend to focus on 2 or 3 sources of information. Lerner et al. [40] and Roychoudhury et al. [41] use bond graph models to define the interactions between components and subsystems. However, the approaches do not consider the availability of existing databases of information or reliability data. The interactions are defined by the bond graph and expert opinion. An approach utilizing data mining may yield additional insight into the behavior of the system.

3.1.3 System Health Monitoring with Heterogeneous Information

System health monitoring (SHM) involves many forms of heterogeneity. The systems under consideration and the SHM approaches for monitoring them are heterogeneous on many levels, including the mechanics, materials, sensors, measurements, system models employed, and time scales. However, this usually does not imply heterogeneous information as discussed in this dissertation.

Heterogeneous sensors extend simple sensors by adding capabilities such as energy storage or communications ability. Yarvis et al. [95] exploit heterogeneity in sensor networks to improve network reliability and lifetime. Kijewski-Correa et al. [96] propose a multi-scale wireless sensing methodology.

The measurements themselves may be heterogeneous. Kulcu et al. [97] describe integration of data streams from many sensors, including slow and high speed strain gages and a camera, on the Commodore Barry Bridge. Sidek et al. [98] fuse acoustic, electro-mechanical, and optical sensors to monitor deterioration in concrete structures.

Some methodologies combine system models with measurements. Kacprzynski and Muench [99] combine a corrosion model with measurements for adaptive prognosis. Viana et al. [100] combine experimental data from a sensor data with a structural model for aircraft SHM. Others incorporate expert opinion. Taha and Lucero [101] use fuzzy sets in combining sparse data and expert opinion for SHM.

3.1.4 Contributions

The above review indicates a large number of SHM methodologies and technologies that integrate various sources of information on a local scale, often relying on a particular mathematical model, focusing on a particular component, or isolating a particular failure mechanism. However, more generalized SHM frameworks are needed which are capable of integrating many types of information encountered in a complex system and able to combine localized approaches for total system SHM.

The primary contributions of this chapter are a DBN-based methodology for diagnosis of a mechanical system in the presence of multiple types of data and components and diagnosis uncertainty quantification for a particle filter. Heterogeneous information, available from multiple sources and in various formats — observational data, laboratory (interventional) data, published failure rates, mathematical models, expert opinion — is used to construct a dynamic Bayesian network. The DBN captures the relationships between components and subsystems in addition to

describing individual components and their behaviors. The dynamic Bayesian network structure is used with a particle filter to obtain filtered estimates of the system state, thus integrating system measurements and providing a diagnosis of the system. From the sample-based state estimated, diagnosis uncertainty is quantified.

The methodology is illustrated with a cantilever beam with a mid-span crack and potential damage at the support. The problem of estimating crack length after a number of cycles, starting with a random initial crack length is addressed through the integration of a mathematical model into the DBN. Additionally, the sensitivities of system responses to inputs of the system are explored.

Details of the methodology for integrating heterogeneous information are presented in Section 3.2 and Chapter 2. Section 3.3 explains how the DBN is used in diagnosis and how diagnosis uncertainty is quantified. The cantilever beam example is then explored in Section 3.4.

3.2 System Modeling

In Chapter 2, a methodology for integration of heterogeneous information into a DBN system model was proposed. Expert opinion, reliability data, mathematical models, operational data, and laboratory data are pulled together into the probabilistic DBN. The DBN allows updating of the system state as new measurements are obtained, making it ideal for diagnosis.

To expand upon the methodology presented in Chapter 2, sensitivity analysis is briefly considered in this section.

3.2.7 Sensitivity Analysis

Sensitivity analysis is a useful tool for understanding and modifying the system model, and can be done before and after diagnosis, as needed. The sensitivity results before diagnosis are based on prior information and after diagnosis are based on the updated system model. By varying input quantities, the effects on system responses are observed. This information indicates which inputs strongly affect the outputs and require further attention. Sensitivity results that are contrary to expectations may reveal errors in modeling or assumptions, or reveal that other phenomena may deserve additional attention. On the other hand, inputs which do not contribute significantly to the uncertainty of the response may be good candidates for dimensional reduction, thus reducing computational effort.

In this chapter, the sensitivity information is provided by standardized linear regression coefficients. To calculate standardized regression coefficients, before fitting, for each variable (input and response), subtract the mean and divide by the standard deviation of that variable. A standardized regression coefficient indicates how many standard deviations an output variable changes for a unit standard deviation change in the input variable Additional details are available in [102], [103], [104]. (Variance-based decomposition is also used for sensitivity analysis in uncertainty quantification studies [105]). Insignificant values of regression coefficients may indicate that variables are immaterial to the response. If all coefficients are insignificant, important variables may be missing from the model. This approach is suitable when the data being fit is in fact linear. A coefficient of determination, R^2, that is nearly 1, indicates that linearity is an acceptable assumption.

3.3 Diagnosis

Diagnosis is the process of determining the state of a system for the purpose of identifying and quantifying damage. As measurements are obtained, the belief about the system state changes and the existence and extent of any damage can be reviewed. This section describes diagnosis of a dynamic system and the quantification of uncertainty in diagnosis.

3.3.1 Diagnosis of a Dynamic System

DBNs update the belief about the values of system variables during diagnosis by integrating new measurements into the system state estimate. However, to successfully use a DBN to perform this update, the system must be viewed in the context of Bayesian recursive filtering. This is because, as more measurements are observed, the hidden variables that are desired to be estimated become highly correlated and traditional inference becomes increasingly expensive, except in degenerate cases [106]. Bayesian recursive filtering addresses this problem by maintaining a distribution over the states of the hidden variables that is easily updated as new measurements become available.

Bayesian Recursive Filtering

Consider a system whose state at time $k+1$, $\mathbf{x}^{k+1}$, is described by

$$\mathbf{x}^{k+1} = f^k\left(\mathbf{x}^k, \mathbf{v}^k\right) \qquad \textbf{Eq. 21}$$

where $\mathbf{x}^k$ is the state vector at time k, $\mathbf{v}^k$ is an independently and identically distributed (i.i.d) process noise vector, and f^k is a possibly nonlinear and time-varying function of $\mathbf{x}^k$ and $\mathbf{v}^k$. The problem is to estimate $\mathbf{x}^{k+1}$ given $\mathbf{x}^k$ and measurements $\mathbf{z}^{k+1}$, where

$$\mathbf{z}^{k+1} = h^k\left(\mathbf{x}^k, \mathbf{n}^k\right) \qquad \textbf{Eq. 22}$$

$\mathbf{n}^k$ is i.i.d measurement noise and $h^k$is a possibly nonlinear and time-varying function of $\mathbf{x}^k$ and $\mathbf{n}^k$. Of particular interest is estimating $\mathbf{x}^{k+1}$ using all measurements $\mathbf{z}^{1:k+1}$.

To view the problem in terms of Bayesian updating, a probability distribution for $p(\mathbf{x}^{k+1}|\mathbf{z}^{1:k+1})$ is constructed. Because Bayesian statistics is used, this distribution is often called the belief state, denoted as $\sigma^{k+1}(\mathbf{x}^{k+1})$. It is assumed that a prior belief state $\sigma^k(\mathbf{x}^k)$ is available. Additionally, $p(\mathbf{x}^0|\mathbf{z}^0) \equiv p(\mathbf{x}^0)$. Thus, $\sigma^{k+1}(\mathbf{x}^{k+1})$ may be obtained first by obtaining a prior estimate of $\sigma^{k+1}(\mathbf{x}^{k+1})$ (prediction) and then updating that prior to obtain the posterior $\sigma^{k+1}(\mathbf{x}^{k+1})$.

Before obtaining $\mathbf{z}^{k+1}$, the belief state $\sigma^{k+1}(\mathbf{x}^{t+1})$ at $k + 1$ is

$$\sigma^{k+1}(\mathbf{x}^{k+1}) = p(\mathbf{x}^{k+1}|\mathbf{z}^{1:k}) \quad \textbf{Eq. 23}$$

which may be expanded by summing over the states of $\mathbf{x}^k$ as

$$\sigma^{k+1}(\mathbf{x}^{k+1}) = \sum_{\mathbf{x}^k} p(\mathbf{x}^{k+1}|\mathbf{x}^k, \mathbf{z}^{1:k}) p(\mathbf{x}^k|\mathbf{z}^{1:k}) \quad \textbf{Eq. 24}$$

Using the Markov assumption, which says that the future $\mathbf{x}^{k+1}$ is independent of all else given the previous state $\mathbf{x}^k$, the term $\mathbf{z}^{1:k}$ may be eliminated from $p(\mathbf{x}^{k+1}|\mathbf{x}^k, \mathbf{z}^{1:k})$, resulting in

$$\sigma^{k+1}(\mathbf{x}^{k+1}) = \sum_{\mathbf{x}^k} p(\mathbf{x}^{k+1}|\mathbf{x}^k) p(\mathbf{x}^k|\mathbf{z}^{1:k}) \quad \textbf{Eq. 25}$$

where $p(\mathbf{x}^k|\mathbf{z}^{1:k}) = \sigma^k(\mathbf{x}^k)$. Upon receiving the measurement at time k+1, Bayes' rule may be used to update the belief state, and Eq. 25 becomes

$$\sigma^{k+1}(\mathbf{x}^{k+1}) = p(\mathbf{x}^{k+1}|\mathbf{z}^{1:k}, \mathbf{z}^{k+1}) \quad \textbf{Eq. 26}$$

By Bayes' rule expansion of the right hand side of Eq. 26,

$$\sigma^{k+1}(\mathbf{x}^{k+1}) = \frac{p(\mathbf{z}^{k+1}|\mathbf{x}^{k+1}, \mathbf{z}^{1:k}) p(\mathbf{x}^{k+1}|\mathbf{z}^{1:k})}{p(\mathbf{z}^{k+1}|\mathbf{z}^{1:k})} \quad \textbf{Eq. 27}$$

Because the measurements $\mathbf{z}^{k+1}$ and $\mathbf{z}^{1:k}$ are conditionally independent given $\mathbf{x}^{k+1}$, $p(\mathbf{z}^{k+1}|\mathbf{x}^{k+1}, \mathbf{z}^{1:k}) = p(\mathbf{z}^{k+1}|\mathbf{x}^{k+1})$, resulting in

$$\sigma^{k+1}(\mathbf{x}^{k+1}) = p(\mathbf{x}^{k+1}|\mathbf{z}^{1:k+1}) = \frac{p(\mathbf{z}^{k+1}|\mathbf{x}^{k+1})p(\mathbf{x}^{k+1}|\mathbf{z}^{1:k})}{p(\mathbf{z}^{k+1}|\mathbf{z}^{1:k})} \quad \textbf{Eq. 28}$$

where $p(\mathbf{x}^{k+1}|\mathbf{z}^{1:k})$ is equivalent to Eq. 25.

Unfortunately, only specific implementations of the Bayesian recursive filter such as the Kalman filter or grid-based methods for domains consisting of discrete variables provide analytical methods of evaluating Eq. 28. Thus it is necessary to pursue approximate inference algorithms. Particle filtering is one technique that makes the filtering problem tractable. Details of Bayesian recursive filtering and particle filters are available in [107].

Particle Filtering

Particle filtering, i.e. sequential Monte Carlo (MC), is a method for approximating the distribution of the belief state. Common particle filtering method are based on sequential importance sampling (SIS), which improves upon the basic sequential MC by weighting point masses (particles) according to their importance sampling density, thus focusing on the samples that affect the posterior belief state the most. A comprehensive tutorial on particle filters is given by Ristic et al. [107] and is the basis of the following explanation on the SIS algorithm.

A SIS filter builds upon the basic ideas of MC integration and importance sampling. In MC integration, to evaluate $I = \int \boldsymbol{g}(\boldsymbol{x})d\boldsymbol{x}, \mathbf{x} \in R^{n_x}$. $\boldsymbol{g}(\boldsymbol{x})$ may be factored into $\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) \cdot \pi(\boldsymbol{x})$ such that $\pi(\boldsymbol{x})$ is a probability density function where $\pi(\boldsymbol{x}) \geq 0$ and $\int \pi(\mathbf{x})d\boldsymbol{x} = 1$. By drawing enough samples $\{\mathbf{x}_i; i = 1, \dots, N_s\}$ from $\pi(\boldsymbol{x})$, the MC estimate of the integral is $I_{N_s} = \frac{1}{N_s}\sum_{i=1}^{N_s} \boldsymbol{f}(\mathbf{x}_i)$.

$\pi(\boldsymbol{x})$ may not be invertible and may be impossible to sample. However, there may be a similar density $q(\boldsymbol{x})$, called the importance or proposal density, which is easily sampled. The original integral may be rewritten as

$$I = \int \boldsymbol{f}(\boldsymbol{x})\pi(\boldsymbol{x})d\boldsymbol{x} = \int \boldsymbol{f}(\boldsymbol{x})\frac{\pi(\boldsymbol{x})}{q(\boldsymbol{x})}q(\boldsymbol{x})d\boldsymbol{x}. \quad \textbf{Eq. 29}$$

Some $\mathbf{x}_i$ may be sampled from $q(\boldsymbol{x})$. The value $q(\mathbf{x}_i)$ is related to $\pi(\mathbf{x}_i)$ by a weight, $\widetilde{w}(\mathbf{x}_i)$, so that $q(\mathbf{x}_i)\widetilde{w}(\mathbf{x}_i) = \pi(\mathbf{x}_i)$. The weights are calculated by

$$\widetilde{w}(\mathbf{x}_i) = {\pi(\mathbf{x}_i)}\Big/{q(\mathbf{x}_i)} \quad \textbf{Eq. 30}$$

and the value of the MC integral becomes

$$I_N = \frac{1}{N}\sum_{i=1}^{N_s} \boldsymbol{f}(\mathbf{x}_i)\,\widetilde{w}(\mathbf{x}_i). \quad \textbf{Eq. 31}$$

If the normalizing factor of $\pi(\boldsymbol{x})$ is not known, the importance weights must be normalized by

$$w(\mathbf{x}_i) = {\widetilde{w}(\mathbf{x}_i)}\Big/{\sum_{i=1}^{n}\widetilde{w}(\mathbf{x}_i)}. \quad \textbf{Eq. 32}$$

The value of the estimated integral is then $I_N = \sum_{i=1}^{N_s}\mathbf{f}(\mathbf{x}_i)w(\mathbf{x}_i)$

$$I_N = \sum_{i=1}^{N_s}\mathbf{f}(\mathbf{x}_i)w(\mathbf{x}_i). \quad \textbf{Eq. 33}$$

Now, to apply the MC approach to the Bayesian filtering problem of estimating $\sigma^{k+1}\left(\mathbf{x}^{k+1}\right)$, consider the joint posterior density at time $k + 1$, $p\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)$. Assuming $\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right) = \int \boldsymbol{g}(\boldsymbol{x})d\boldsymbol{x}$, a sample-based approximation of the posterior at $k + 1$, $p\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)$, may be taken as

$$p\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right) \approx \sum_{i=1}^{N_s}\delta\left(\mathbf{x}^{0:k+1} - \mathbf{x}_i^{1:k+1}\right)w_i^{k+1}. \quad \textbf{Eq. 34}$$

with support points $\mathbf{x}^{0:k+1}$ having been sampled from some importance density $q\left(\mathbf{x}^{k+1}\middle|\mathbf{z}^{k+1}\right)$ and normalized weights w_i^{k+1}. The normalized weights are

$$w_i^{k+1} \propto \frac{p\left(\mathbf{x}_i^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)}{q\left(\mathbf{x}_i^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)}. \quad \textbf{Eq. 35}$$

If at time k there are samples and weights approximating $p\left(\mathbf{x}^{0:k}\middle|\mathbf{z}^{1:k}\right)$, after observing $\mathbf{z}_k$ the density $p\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)$ may be approximated by a new set of samples. The importance density may be factorized as

$$q\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right) = q\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{0:k}, \mathbf{z}^{1:k+1}\right) q\left(\mathbf{x}^{0:k}\middle|\mathbf{z}^{1:k}\right) \quad \textbf{Eq. 36}$$

which allows for the new samples $\mathbf{x}_i^{k+1} \sim q\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)$ to be generated simply by augmenting the previous state $\mathbf{x}_i^k \sim q\left(\mathbf{x}^{0:k}\middle|\mathbf{z}^{1:k}\right)$with the new state $\mathbf{x}_i^k \sim q\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{0:k}, \mathbf{z}^{1:k+1}\right)$.

Now, if $p\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)$ is expressed in terms of $p\left(\mathbf{x}^{0:k}\middle|\mathbf{z}^{1:k}\right)$, $p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}^{k+1}\right)$, and $p\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{k}\right)$, an expression to update the weights can be derived. $p\left(\mathbf{x}^{1:k+1}\middle|\mathbf{z}^{1:k+1}\right)$ may be expressed as

$$\begin{aligned} p\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right) &= \frac{p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}^{0:k+1}, \mathbf{z}^{1:k}\right) p\left(\mathbf{x}^{0:k+1}\middle|\mathbf{z}^{1:k}\right)}{p\left(\mathbf{z}^{t+1}|\mathbf{z}^{1:t}\right)} \\ &= \frac{p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}^{0:k+1}, \mathbf{z}^{1:k}\right) p\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{0:k}, \mathbf{z}^{1:k}\right)}{p\left(\mathbf{z}^{k+1}|\mathbf{z}^{1:k}\right)} p\left(\mathbf{x}^{0:k}\middle|\mathbf{z}^{1:k}\right) \\ &= \frac{p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}^{k+1}\right) p\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{k}\right)}{p\left(\mathbf{z}^{k+1}|\mathbf{z}^{1:k}\right)} p\left(\mathbf{x}^{0:k}\middle|\mathbf{z}^{1:k}\right) \\ &\propto p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}^{k+1}\right) p\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{k}\right) p\left(\mathbf{x}^{0:k}\middle|\mathbf{z}^{1:k}\right). \end{aligned} \quad \textbf{Eq. 37}$$

Knowing the expression for $p\left(\mathbf{x}_i^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)$ and $q\left(\mathbf{x}_i^{0:k+1}\middle|\mathbf{z}^{1:k+1}\right)$, w_i^{k+1} becomes

$$\begin{aligned} w_i^{k+1} &\propto \frac{p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}_i^{k+1}\right)p\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{k}\right)p\left(\mathbf{x}_i^{0:k}\middle|\mathbf{z}^{1:k}\right)}{q\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{0:k},\mathbf{z}^{1:k+1}\right)q\left(\mathbf{x}_i^{0:k}\middle|\mathbf{z}^{1:k}\right)} \\ &= \frac{p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}_i^{k+1}\right)p\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{k}\right)}{q\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{0:k},\mathbf{z}^{1:k+1}\right)}w_i^k \end{aligned} \quad \textbf{Eq. 38}$$

Additionally, if $q\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{0:k},\mathbf{z}^{1:k+1}\right) = q\left(\mathbf{x}^{k+1}\middle|\mathbf{x}^{k},\mathbf{z}^{k+1}\right)$, the weight becomes

$$w_i^{k+1} = \frac{p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}_i^{k+1}\right)p\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{k}\right)}{q\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{k},\mathbf{z}^{k+1}\right)}w_i^k \quad \textbf{Eq. 39}$$

And the posterior filtered density is

$$p\left(\mathbf{x}^{k+1}\middle|\mathbf{z}^{k+1}\right) \approx \sum_{i=1}^{N_s}\delta\left(\mathbf{x}^{k+1}-\mathbf{x}_i^{k+1}\right)w_i^{k+1}. \quad \textbf{Eq. 40}$$

Thus, a recursive Bayesian filter may be implemented by maintaining a set of points and weights that approximate $p\left(\mathbf{x}^{k+1}\middle|\mathbf{z}^{k+1}\right)$.

A summary of the SIS algorithm for one time step is as follows. A previous (or initial if k=1) set of N_s weights w_i^t and N corresponding particles $\mathbf{x}_i^k$ are known. N samples are drawn from the importance distribution, $q\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{k},\mathbf{z}^{k+1}\right)$. The unnormalized weights $\widetilde{w}_i^{k+1}$ are then computed up to a normalizing constant using Eq. 39. The weights $\widetilde{w}_i^{k+1}$ are then normalized so that their sum is equal to 1. The normalized weights w_i^{t+1} and points $\mathbf{x}_i^{k+1}$ form an approximation to $p\left(\mathbf{x}^{k+1}\middle|\mathbf{z}^{k+1}\right)$. Two common choices of the importance density $q\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{k},\mathbf{z}^{k+1}\right)$ are the prior $p\left(\mathbf{x}_i^{k+1}\middle|\mathbf{x}_i^{k}\right)$ and the likelihood $p\left(\mathbf{z}^{k+1}\middle|\mathbf{x}_i^{k+1}\right)$. Either of these choices simplifies the weight update in Eq. 39.

Many variations of the SIS particle filter exist using different importance sampling densities, resampling techniques, and combinations of discrete and continuous variables. Resampling is a technique used to prevent a phenomenon called degeneracy, where a single particle accounts for a large proportion of the total weight of all particles. The choice of importance sampling density also can influence degeneracy. Ristic et al. [107] provide algorithms for many variations of the basic SIS particle filter. The auxiliary particle filter (APF) [108] is used in this chapter, as it is more efficient than a basic SIS filter with resampling. An improvement of the APF for systems with distinct operating modes has been developed by Andrieu et al. [109].

The DBN contains information necessary for particle filtering including the likelihood distribution and a transition model between load steps k and $k + 1$. These distributions are used as the importance sampling density and to estimate the weights of particles. Thus, any information integrated in the DBN plays a role in online damage diagnosis. Initial assumptions from expert opinion, operational and laboratory data, reliability data, and mathematical models all impact the resulting DBN and the outcome of diagnosis.

3.3.2 Fault Detection and Isolation

The particle filter allows the state estimate of the system to evolve whether faults are present or not. However, hundreds or thousands of particles each with constantly changing, multidimensional state estimates, are not in and of themselves conducive to maintenance or mission-level decision making. To make sense of the information provided in the particles, a procedure that alerts the presence of a fault, isolates the fault, and quantifies the damage is desirable.

Fault detection and isolation may occur in a couple of ways. Roychoudhury et al. [41] use a DBN derived from a bond graph with SIS of the healthy system for tracking. The belief state estimates

are compared against measurements using a statistical hypothesis test to detect faults. A qualitative, bond-graph based procedure is used to isolate fault candidates. Separate DBNs for each fault candidate are then used to quantify the faults and determine the most likely fault candidate. Another example is provided by Jha et al. [92] who use a DBN of discrete variables to determine the most probable explanation of transient faults. However, the approach is offline and does not require any filtering.

Conveniently, the DBN and state estimates of the system provided by particle filtering contain the information necessary for the three stages of diagnosis. The state estimate given by the SIR particle filter, S, consists of N_s samples with associated weights. The weights are the normalized likelihood of each particle. Here it is assumed that the samples are of equal weight, which occurs after resampling in a particle filter. The marginal distribution over each of the k combinations of faults, including the healthy case, is a k-dimensional multinomial distribution. $k = 2^n$, where n is the number of faults. The i^{th} fault combination occurs with expected probability $p_i = N_i/N_s$, where N_i is the number of occurrences of the i^{th} fault combination in S and N_s is the number of samples.

The probability of detection is the probably of observing any fault. If the healthy scenario has an expected value of $p_1 = N_1/N_s$, the detection probability is taken as $P_D = 1 - p_1$. If P_D exceeds the detection threshold, an alarm is triggered and isolation is performed by finding the largest p_i. To quantify the damage, the mean value of the damage parameter(s), q_i, is calculated over all particles corresponding to i^{th} fault combination. Note that q_i may be a vector if the i^{th} fault combination has multiple damage parameters.

3.3.3 Diagnosis Uncertainty

While the particle filter inherently accounts for the uncertainty of the state estimate by providing a distribution of the state, the uncertainty in this estimate is in general difficult for a human decision-maker to discern. Lack of understanding of the diagnosis uncertainty could result in the wrong maintenance actions being performed or prematurely taking a system offline for repairs. Thus, providing a measure of the overall diagnosis uncertainty for the decision-maker is important.

Overall diagnosis uncertainty is determined by the uncertainty in the existence of damage, location of damage, and extent of damage. These uncertainties may be combined to determine the overall uncertainty in diagnosis. Following Sankararaman and Mahadevan [110], the overall probability of correctly diagnosing a fault in parameter θ_i and correctly obtaining the value of θ_i, denoted by q_i, is

$$F_q(q_i) = (1 - P_D) * F_q\left(q_{healthy}\right) + P_D * \left(1 - P_{iso,i}\right) * F_q\left(q_{healthy}\right) + P_D P_{iso,i} * F_q(q_i | Damage\ is\ true\ and\ in\ \theta_i) \quad \textbf{Eq. 41}$$

where F_q is the unconditional CDF of q_i, P_D is the fault detection probability, $1 - P_D$ is the probability of a false alarm, $q_{healthy}$ is the nominal value of θ_i, and $P_{iso,i}$ is the probability of correct fault isolation given damage.

Sankararaman and Mahadevan [110] use a Bayesian methodology to estimate the various terms in Eq. 41. In the context of the particle filter, an estimate of the belief state already contains this information. The unconditional pdf of damage parameter q, $f_q(q_i)$ — the derivative of the unconditional cdf, $F_q(q_i)$, given in Eq. 41 — is conveniently provided by the particles. Each particle contains an estimate of q, and given hundreds or thousands of particles, an estimate of the distribution $f_q(q_i)$ may be constructed. This estimate accounts for the probability of incorrect isolation by allotting more particles to the most likely fault combinations. The probability of a false

alarm is included because the healthy state is one of the combinations considered. This distribution, along with a plot or distribution parameters of the healthy distribution of q, alerts the decision maker to the severity of damage.

The probability of correct isolation for combination i is the probability that the i^{th} fault combination is true, which is $p_i = N_i/N_s$. Thus, $P_{iso,i} = p_i = N_i/N_s$. The detection probability $P_D = 1 - \sum_{i=2}^{n} p_i$, where $i = 2..n$ represent the system configurations with at least one fault. The detection probability and faults ranked according to isolation probability may be presented to the decision maker.

Confidence in the diagnosis uncertainty depends on the number of samples, N_s, and may be increased by increasing N_s. The variance of p_i follows from the Dirichlet distribution as

$$Var[p_i] = \frac{N_i(N_s - N_i)}{{N_s}^2(N_s + 1)} \quad \textbf{Eq. 42}$$

Confidence intervals for p_i may be calculated as in [111] (or approximated with the normal distribution). This information allows the decision maker to assess how meaningful the detection and isolation probabilities are.

Three factors that affect the diagnosis (and prognosis) of a system are the sensitivity of the system measurements to damage, noise in the training and measurement data, and the sampling frequency at which measurements are obtained. If the sensitivity to damage is low and the noise large, diagnosis may be difficult or impossible. A low sampling frequency can deteriorate the quality of state estimates in systems that depend on previous states, such as nonlinear dynamic systems like the actuator system presented in Chapter 2. The states estimates amount to a linearized approximation of the state across the sampling interval (reciprocal of sampling frequency). As

sampling rate decreases, the interval becomes wider and the approximation worsens. A greater sampling frequency decreases the sampling interval and reduces the error due to linearization, thus improving states estimates.

3.3.4 Summary

Chapter 2 proposed a methodology for integrating multiple types of information from multiple sources into a DBN. The current section presented a framework for the diagnosis of faults and quantification of diagnosis uncertainty, when using a particle filter and in the presence of multiple faults. The results of uncertainty quantification, namely the probability of damage, isolation probabilities, distributions of damage estimates, and sensitivity analysis provide information regarding confidence in the system model and the diagnosis to a decision maker.

3.4 Illustrative Example

To illustrate the methodology, diagnosis is performed on the cantilever beam structure of Figure 12 in the presence of heterogeneous information. The available information consists of expert opinion, synthetic operational and laboratory data generated by an ANSYS finite element model, published failure rates for damage occurring at the support and a crack occurring, and Paris' law for modeling crack growth. To account for uncertainty, some assumptions have been made about the distributions of some model parameters. However, the methodology is agnostic to such assumptions.

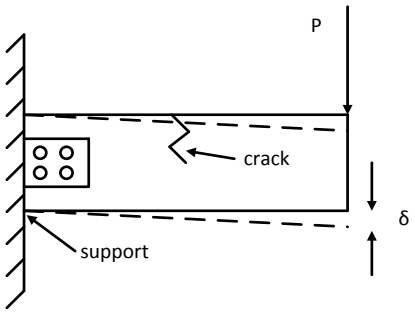


Figure 12. Cantilever beam system

The example system consists of a 1 meter long cantilever beam of depth 100 mm and uniform cross section made of 7075-T6 aluminum alloy subjected to a point load at the free end. The beam is subject to K cycles of load P, and the interest is in calculating/measuring deflection δ at the free end. Suppose that a mid-span crack in the beam initiates at the k^{th} load step, and that this crack grows as additional cycles of the load P are applied. Also, suppose the support has the possibility of suddenly weakening. Both of these faults have the effect of decreasing the stiffness of the system, resulting in a larger deflection magnitude. The stiffness is a function of the support condition, geometry (length, cross section), material properties (Young's modulus), and crack size/shape. The support damage alters the support from the most stiff condition (fully fixed) to a less stiff condition (partially fixed). The crack effectively changes the cross section of the beam and reduces the moment of inertia, thereby decreasing the ability of the beam to resist bending. Discrete variables D and Cr represent, respectively, whether damage at the support has occurred and whether a crack has occurred. It is desired to create a DBN for this system that can diagnose the existence of damage at the support or a crack and quantify the amount of damage.

3.4.1 DBN Construction from Heterogeneous Information

Expert Opinion

Suppose an expert has decided to treat the condition of the support and the crack independently, while the crack length depends on the load. This network structure can be updated using operational data if desired. (However, the network structure is not updated in the numerical example below, because the data does not indicate the need for updating). The expert further designates *P* as a time varying load that can be modeled with an ARIMA model, and treats the continuous variables, stress intensity factor and deflection, as Gaussian. The crack length, *a,* is to be calculated using Paris' law. Note that two variables, *Cr* and *a*, are being used to describe the presence of the crack (true/false) and the crack length, respectively. The expert does not define any of the conditional probabilities within the network.

The expert then determines the structure of the DBN (Figure 13) along with the system variables and their distributions (Table 11), based on knowledge of the system. With the network structure and assumptions about variable types in place, new types of information, such as operational data, become the focus of attention.

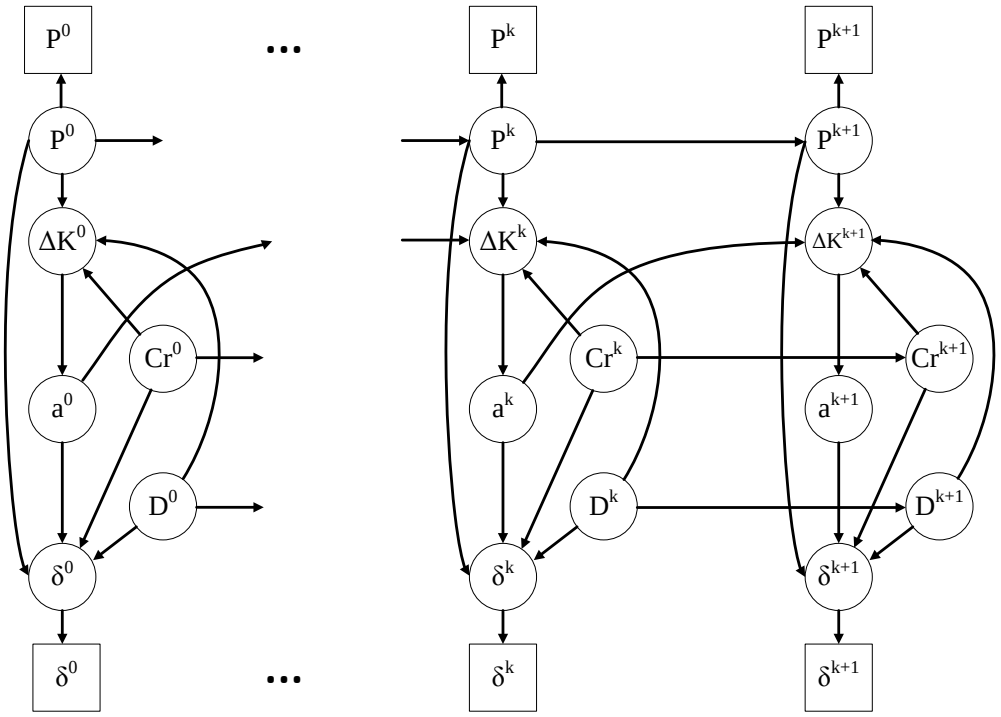


Figure 13. Cantilever Beam DBN, initial and transition network. Gray nodes are measurements.

Table 11. Cantilever beam DBN (Figure 13) model variables.

Variable	Symbol	Type	Distribution	Role
Damage at support indicator	*D*	Discrete	Binomial (false, true)	Model parameter
Crack indicator	*Cr*	Discrete	Binomial (false, true)	Model parameter
Load	*P*	Continuous	Gaussian	Input (measured)
Stress Intensity Factor	*ΔK*	Continuous	Gaussian	Model parameter
Crack length	*a*	Continuous	Gaussian	Model parameter
Deflection	δ	Continuous	Gaussian	Output (measured)

Mathematical Models

The beam has possible damage at the support and possibly a crack that grows with the load cycle. A (two-dimensional) ANSYS finite element model of the beam is created to generate synthetic data. The ANSYS model is used to calculate the structural response (deflection) given *I* load histories, each containing *J* load cycles. For each load history, model parameters are sampled from

distributions given in Table 12. The load P is described by an ARIMA model, following Ling and Mahadevan [48].

Each load history is initially applied to a "healthy" version of the model, and the deflection of the free end of the beam recorded at each load cycle. If damage at the support occurs or a crack forms at a particular load cycle, a "faulty" version of the model is used for the rest of the load history. The "faulty" model may either have damage at the support or a crack, or both. Because both faults may occur, and one fault will likely occur before the other, this may result in switching from a "healthy" model to a "single fault" model to a "both faults" model within one load history. In total, four models are available, one for each configuration of the system.

Table 12. Finite element model parameters

Parameter, unit	Symbol	Distribution Type	Distribution Parameters	
Initial crack length, mm	a_0	lognormal	Mean 0.23 Std. 0.05	
Paris' law parameter	C	lognormal	Mean 6.54E-13 Std. 4.01E-13	
Young's modulus, N/mm^2	E	Constant	71705.48	
Damage at support, mm	d	Gaussian	Mean 20 Std. 0.00075	
Paris' law exponent	m	constant	3.8863	
Load, N	P	Gaussian	(2,1,0) ARIMA specification	
Exponential parameter for occurrence of damage at support	λ_d	Gaussian	If uncracked Mean 2E-4 Std. 1E-5	If cracked Mean 4E-4 Std. 1E-5
Exponential parameter for occurrence of crack	λ_c	Gaussian	If no damage at support Mean 3E-4 Std. 1E-5	If damage at support Mean 6E-4 Std. 1E-5

Damage at the support is modeled by relaxing the constraints in the ANSYS model. This is done for nodes lying on the fixed end of the beam and on the interval [0 d] from the bottom edge of the beam, where d is sampled from a normal distribution with parameters given in Table 12. d is held constant until the end of the end of the load history. The distribution of d has been heuristically selected for demonstration of the diagnosis methodology.

The crack is modeled as a straight crack as shown in Figure 14 and Figure 15. A finer mesh is used in the vicinity of the crack tip. The length of the crack is increased in the model according to Paris' law [112], $^{da}/_{dN} = C(\Delta K)^m$, where C and m are material dependent constants and ΔK is the stress intensity factor (SIF) calculated within ANSYS at a particular load step. The crack is initialized with a length a_0 that is determined by sampling from the equivalent initial flaw size (EIFS) [113] distribution (Table 12). Distributions of the model parameters C, and m are found in [113] and given in Table 12.

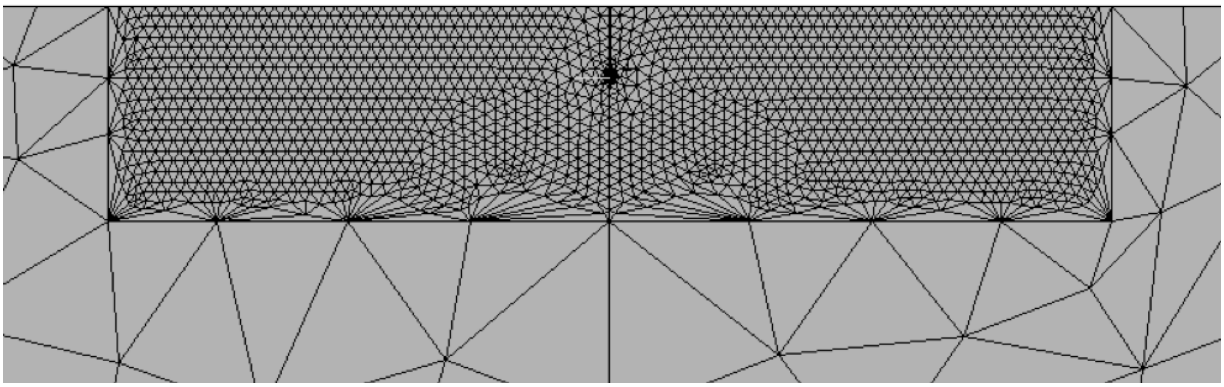


Figure 14. Mesh in vicinity of crack

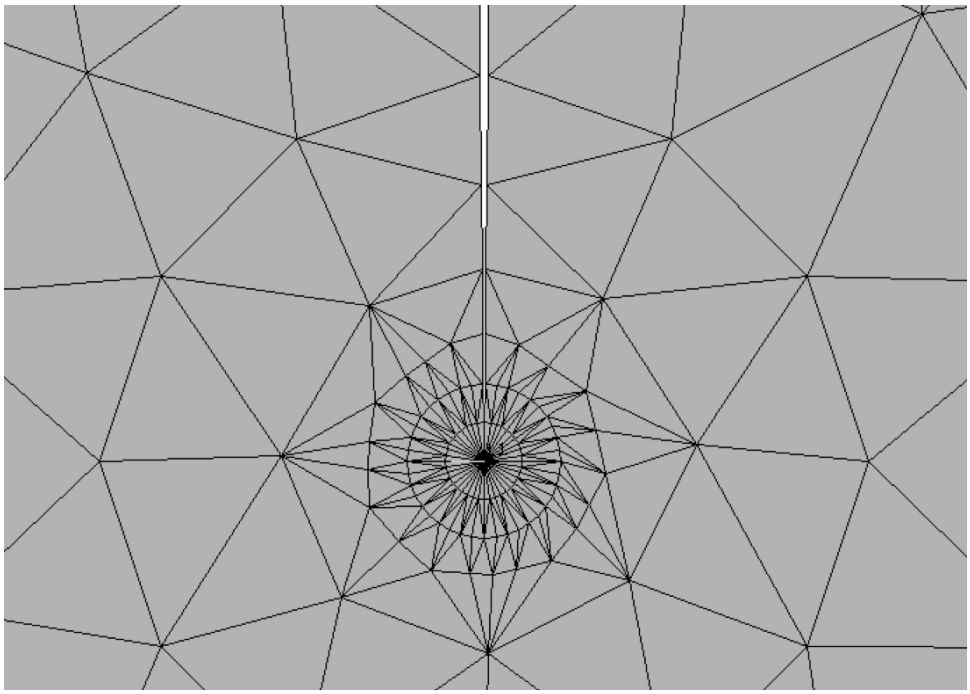


Figure 15. Mesh at crack tip

The load, P, is generated by an ARIMA model (1,1,0), which is trained from online operational data as it becomes available. The one step ahead prediction is considered to have a Gaussian distribution. The stress intensity factor, $\Delta K(k+1)$, is determined based on the previous crack length, a(k), existence of the crack, $Cr(k+1)$, and the load $P(k+1)$ using a Gaussian process (GP) model built using the Netlab toolbox in Matlab [114] with a rational quadratic covariance function (see Appendix for details of GP modeling). GP models have been used because of their ability to effectively approximate arbitrary nonlinear functions probabilistically, without assuming a particular functional form. (What is assumed is the covariance structure). Further, GP models can be deployed efficiently in particle filtering, requiring at most one matrix inversion operation per possible state of the discrete parents of the variable being modeled by the GP. The discrete parent of $\Delta K(k+1)$, $Cr(k+1)$, is binary, indicating two operating modes and a maximum of two matrix inversions per

cycle in particle filtering. In reality, the computational effort may be even less. For example, if $Cr(k+1) = true$ does not occur in any of the particles, there is no need to invert a matrix for the GP of $\Delta K(k+1)$ corresponding to this condition. Further, in the case when $Cr(k+1) = false$, $\Delta K = 0$ and no GP is required at all, further reducing computational effort. When $Cr(k+1) = true$, $\Delta K > 0$ and GP is trained. Similarly, the crack length, $a(k+1)$, is estimated using a GP model, with previous crack length, a(k), and stress intensity factor, $\Delta K(k+1)$, as parents. $a(k+1)$ has no discrete parents, so only one GP is required to be trained. The GPs for $\Delta K(k+1)$ and $a(k+1)$ are trained using a time-lagged database as in Table 3 in Chapter 2.

The deflection, $\delta(k+1)$, is modeled using linear regression of the form $P(k+1) = b_0 + b_1 P(k+1) + b_2 a(k+1)$, b_0 is the intercept, b_1 is the load coefficient, and b_2 is the crack length coefficient. Note that b_2 is only required when the crack is present. Because the deflection is additionally conditioned on the presence of damage at the support, D, and the existence of a crack, Cr, four different sets of regression coefficients are estimated corresponding to each unique system configuration. Test statistics for the regression models at the 99% confidence level are listed in Table 13. (Modulus of elasticity is not included in the regression model, as it is assumed constant).

The linear relationship is appropriate and adequate in this case as the system is assumed to stay within the elastic regime. The linear regression also enables accurate extrapolation outside of training values (within the elastic regime). In contrast, a GP model increases the uncertainty in between training points, making deflection estimates noisier and diagnosis more difficult, and a GP requires more computational effort to train and evaluate. (Thus when there is good reason to assume a functional form and constant variance, a simple regression model may be adequate; when the functional form is not obvious and the fluctuation between training points needs to be modeled,

a GP model might be more appropriate. Note that the GP model also consists of a mean term and a variance term; the mean term is very similar to a regression model).

Table 13. Linear regression coefficients and test statistics

Configuration	Regression Coefficients	99% conf. interval (t-dist)	Standardized Regression Coefficients	R^2
D = false Cr = false	b_0=-1.2488E-5 b_1=5.3141E-4 b_2=0	[-1.2589E-5, -1.2387E-5] [5.3141E-4, 5.3141E-4] [0, 0]	b^*_0=-0.7071 b^*_1=0.7071	0.999
D = true Cr = false	b_0=-1.8231E-4 b_1=5.5932E-4 b_2=0	[-1.8252E-4, -1.8210E-4] [5.5932E-4, 5.5932E-4] [0, 0]	b^*_0=-0.7071 b^*_1=0.7071	0.999
D = false Cr = true	b_0=-2.3332E-4 b_1=5.3143E-4 b_2=0.0013	[-2.3448E-4, -2.3216E-4] [5.3143E-4, 5.3143E-4] [0.0013, 0.0013]	b^*_0=-1.0000 b^*_1=0.0001 b^*_2=1.0000	0.999
D = true Cr = true	b_0=-0.0015 b_1=5.5933E-4 b_2=0.0056	[-0.0015, -0.0015] [5.5933E-4, 5.5933E-4] [0.0056, 0.0056]	b^*_0=-0.8407 b^*_1=-0.2651 b^*_2=1.1058	0.999

Note: D – support damage state; Cr – crack state.

Reliability Data

Existence of damage at the support, *D*, is governed by an exponential distribution with rate parameter λ_d that is sampled before each load history from a normal distribution with parameters given in Table 12. Because it is assumed that damage at the support is more likely if a crack is present, λ_d is increased if a crack occurs.

Before each load application, an appropriate ANSYS model of the beam is selected. A random number on the interval [0 1] is generated. If the random number exceeds $exp(-\lambda_d n)$, where n indicates the *nth* loading in the load history, then a ANSYS model with damage at the support is used.

The presence of the crack in the beam, *Cr*, is assumed to be governed by an exponential distribution, whose rate parameter, λ_c, is sampled from a normal distribution (mean, variance) before

simulation. λ_c is increased if there is damage at the support already. As with damage at the support, a random number is drawn on the interval [0 1], and, if the number exceeds $exp(-\lambda_c n)$, an ANSYS model for a cracked beam is used. The crack is initialized with a length a_0 that is determined by sampling from the equivalent initial flaw size (EIFS) [113] distribution (Table 12).

To illustrate the inclusion of reliability data in a DBN, consider the variable $Cr(k+1)$ whose conditional probability table is shown in Table 14. $Cr(k+1)$ indicates whether a crack has occurred at load step k + 1. When $Cr(k+1) = false$, there is no crack, and when $Cr(k+1) = true$, there is a crack. $Cr(k+1)$ has one parent, $Cr(k)$.

Table 14. Conditional probability table for Cr(k+1).

Crack state at cycle k	Pr($Cr(k$+1) = false | state)	Pr($Cr(k$+1) = true | state)
$Cr(k)$ = false	$p_{1,1}(k$+1)	$p_{1,2}(k$+1)
$Cr(k)$ = true	$p_{2,1}(k$+1)	$p_{2,2}(k$+1)

The probabilities $p_{i,j}(k$+1) may be determined using reliability data (failure rates) found in the literature. In particular, Pr($Cr(k$+1) = true | $Cr(k)$ = false) is of interest, as this is the probability of the fault occurring given that the fault has not occurred at the previous load cycle.

Operational and Laboratory Data

Synthetic operational data was generated using the ANSYS finite element model. To obtain enough data to estimate network parameters, 50 load histories each containing 200 load cycles were used to generate responses from the beam model.

After generating synthetic data, the data was organized in a table. Sample data for one load history is given in Table 15.

Table 15. Example data for a single load and response history

Load Cycle, *k*	Load, *N*	Support Damage	Crack Existence	Crack Length, *mm*	Deflection, *mm*
1	6	False	False	0	11
2	6	False	False	0	11
3	6	True	False	0	13
4	6	True	True	0.23	13.05
5	6	True	True	0.231	13.1

Because it is desired to know transition probabilities between time slices *k* and *k* + 1 in the DBN, the data is then lagged so that each data case includes data from load cycles *k* and *k* + 1. Lagged data from Table 15 is shown in Table 16.

Table 16. Lagged data for a single history

Load Cycle *k*	Load N	Support Damage	Crack Existence	Crack Length *mm*	Defl. *mm*	Load Cycle *k* + 1	Load *N*	Supp. Dama-ge	Crack Existence	Crack Length *mm*	Defl. *mm*
1	6	False	False	0	11	2	6	False	False	0	11
2	6	False	False	0	11	3	6	True	False	0	13
3	6	True	False	0	13	4	6	True	True	0.23	13.05
4	6	True	True	0.23	13.05	5	6	True	True	0.231	13.1

Sensitivity Analysis

The sensitivity of the deflection to the load, crack length, and damage at support condition (forward sensitivity) may be ascertained by examining the regression coefficients in Table 7. The coefficient of determination, R^2, is nearly 1 for all models, indicating that the linear regression model is a good fit. Further, the 99% confidence intervals for the coefficients are narrow, indicating little uncertainty in their values. Due to the confidence in the regression model, when searching for the source of uncertainty in deflection estimates, the ARIMA and GP models should be scrutinized.

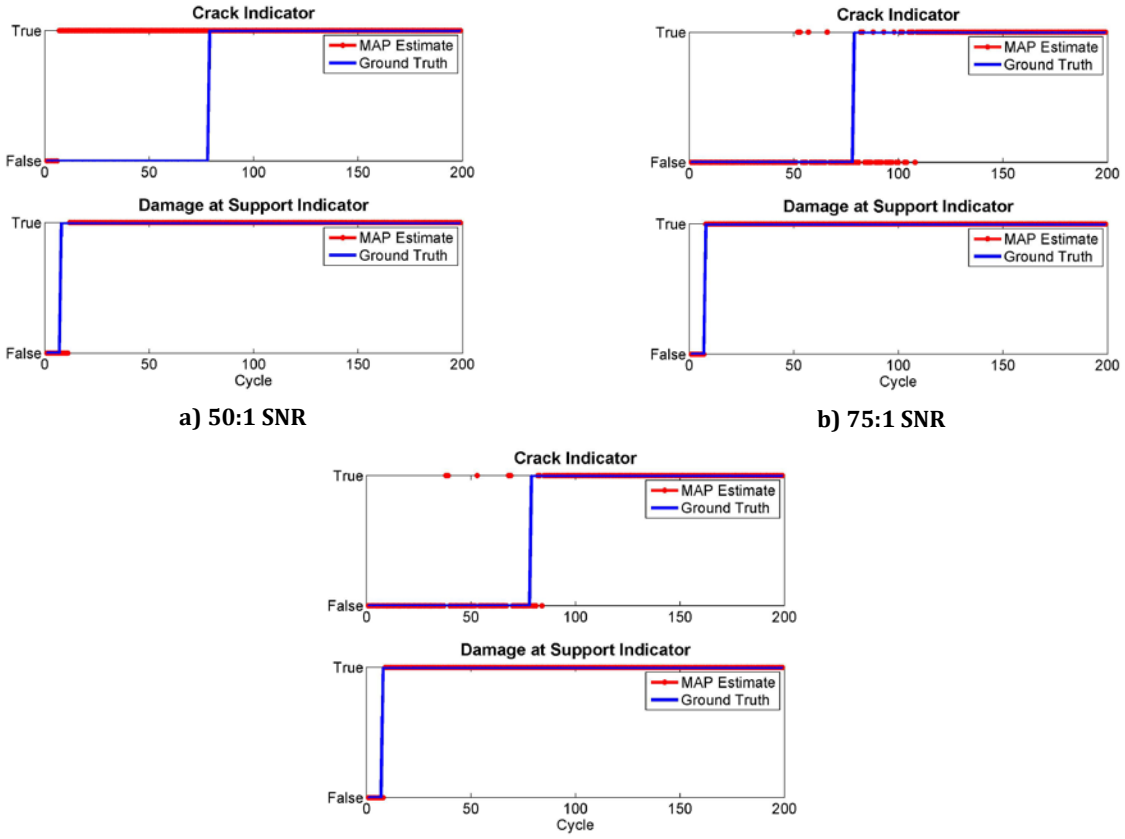
By examining the normalized regression coefficients, it is seen that when damage at the support is present, the effect of crack length on the deflection is magnified (i.e., the normalized magnitude of b_2 increased from 1.0000 to 1.1058) as is the effect of the load on the deflection (normalized magnitude of b_1 increases from 0.0001 to 0.2651), as expected.

3.4.2 Diagnosis

To perform diagnosis, a 200 cycle load history was simulated with damage at the support occurring first followed by a crack. The load and deflection were assumed to be measurable while the crack and damage at the support were assumed to be unobservable. Inference via particle filtering (N_s = 15,000) using an auxiliary particle filter was performed on the DBN to obtain filtered estimates of the system state. The scenario was considered for the separate cases of 50:1, 75:1, and 100:1 signal to noise ratio (SNR) using Gaussian white noise added to the synthetic operational and laboratory data. Noise was added to data used to construct the DBN (including GP, regression, and ARIMA models), and to the synthetic measurement data.

After obtaining each state estimate at each load step, the probability of damage was calculated as in Section 3. If the probability of damage exceeded 95%, an alarm was triggered. The fault was then

isolated and quantified. Figure 16 shows maximum a posteriori (MAP) estimates of the variables *Cr* and *D* against their ground truth values under the various noise scenarios. Note that the MAP value only reflects the most likely particle. Larger amounts of noise affect the ability to accurately diagnose damage as well as the delay in determining the presence of damage. At the 50:1 SNR level, the ability to detect the crack is completely lost, as the noise drowns out the subtle changes in deflection caused by the crack.



a) 50:1 SNR

b) 75:1 SNR

c) 100:1 SNR

Figure 16. MAP estimate of indicator variables for damage at the support and crack length with ground truth values

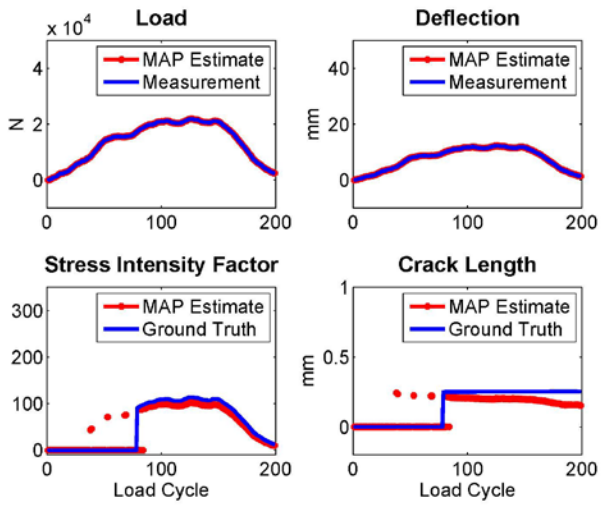


Figure 17. MAP estimate of state variables with measurement or ground truth values (SNR 100:1)

Figure 17 shows the MAP estimates of the load, deflection, stress intensity factor, and crack length with SNR 100:1. Further, the unobservable quantities stress intensity factor and crack length have some uncertainty in their estimates, as evidenced by the MAP values in Figure 17.

3.5 Diagnosis Uncertainty

Diagnosis uncertainty was quantified after obtaining the state estimate at each load step and performing the diagnostic tasks of detection, isolation, and quantification. Figure 18 shows a kernel density estimate of the crack length distribution from the particles at the 97th cycle.

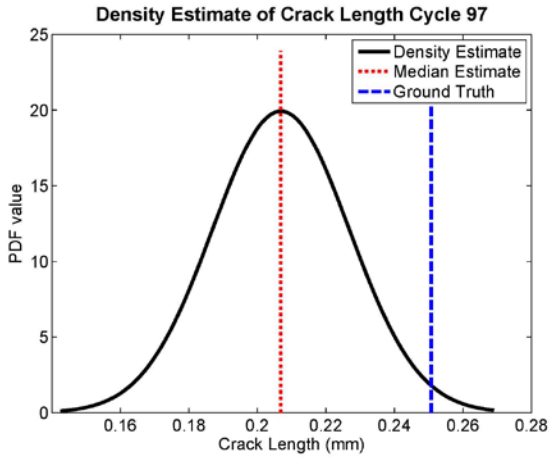
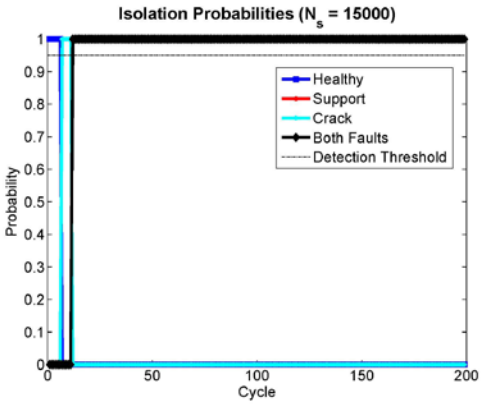
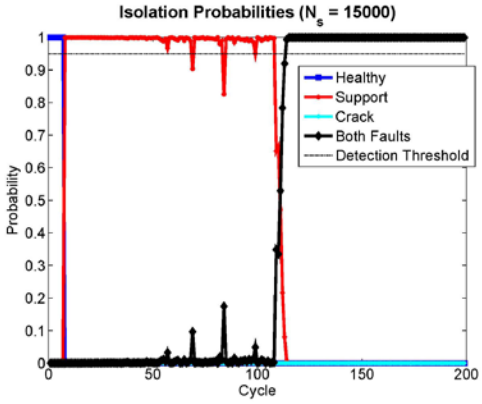


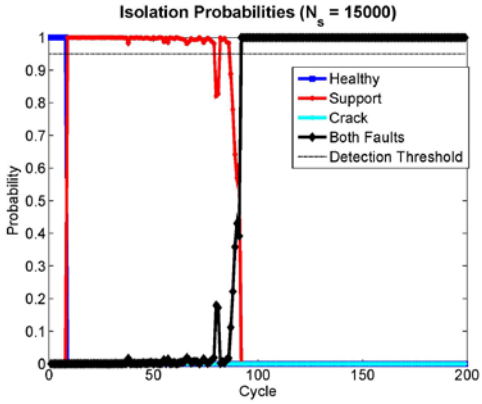
Figure 18. Kernel density estimate of crack length estimate from particle filter with SNR 100:1



a) 50:1 SNR



b) 75:1 SNR



c) 100:1 SNR

Figure 19. Fault isolation probabilities

Figure 19 shows the isolation probability for each of the four system states (healthy, one fault or the other, both faults) as they evolve with load cycle under each noise scenario. The probabilities of all combinations must sum to 1 at each time instant. When a particular fault combination is isolated, its probability approaches 1. In all cases, damage at the support was correctly isolated at the 8^{th} load cycle. However, with 50:1 SNR, the crack was incorrectly isolated well before its occurrence at the 79^{th} cycle (Figure 19 a). As the noise is decreased (Figure 19 b, c), the crack is isolated with less delay.

The ability to diagnose the system is dependent on the amount of noise and the sensitivity of measurements to parameters. Noise and sensitivity to damage vary by system and do not depend on the diagnosis methodology. The diagnosis methodology proposed in this chapter is general and is applicable to many systems. Further investigation is needed to determine the sources of uncertainty in diagnosis.

3.7 Conclusion

This chapter developed a methodology for diagnosis of a mechanical system in the presence of heterogeneous information and uncertainty. First, expert opinion is used to establish the system definition and basic assumptions. Reliability data is used to calculate conditional probabilities for fault indicator variables for damage at the support and a crack. Operational and laboratory data generated from an ARIMA model and ANSYS model are organized in a table and used for estimating a Gaussian process and regression model. Next, the DBN is employed in obtaining filtered estimates of the system state via an auxiliary particle filter. Then, by using information provided during particle filtering, faults are detected and isolated. Diagnosis uncertainty is quantified using the DBN and information acquired during particle filtering. The effectiveness of the methodology needs to be investigated with real-world data and for high-dimensional systems.

The next chapter extends this approach to *prognosis* of future system health and remaining useful life under heterogeneous and uncertain information.

CHAPTER IV.

4. PROBABILISTIC PROGNOSIS USING DYNAMIC BAYESIAN NETWORKS

4.1 Introduction

4.1.1 Background

The rise of complex and costly systems for use in extreme environments has resulted in new challenges in maintenance, planning, decision-making and monitoring for these systems. To reliably execute the missions they were designed for, these systems must be meticulously maintained. Traditional schedule-based maintenance results in unnecessary system downtime and the potential for serious problems to develop between routine maintenance. The alternative, condition-based maintenance (CBM) [115], monitors systems as they operate, alerting personnel when faults occur. Maintenance is performed on-demand, resulting in less downtime and lower costs. Further, online system measurements may occur on different time scales from one another or only be available in particular system configurations. This necessitates seamless integration of current state estimation and predictive techniques, which are part of a prognosis methodology.

Prognosis is the process of predicting the future state of a system coupled with information about the implications of that estimate of the system health state. The quantitative prognosis of a system is commonly expressed through the remaining useful life (RUL). RUL quantifies the amount of time until a system reaches some failure criterion, e.g. fault magnitude or performance metric crosses a threshold or system is no longer operable. Ideally, the uncertainty in RUL is quantified by estimating the distribution of RUL, resulting in a probabilistic prognosis. Importantly, probabilistic

prognosis assesses the outlook for a specific instantiation of a system, or a particular unit under test (UUT). Measurement data updates the belief about the present state and RUL of the particular UUT. In this way, probabilistic prognosis differs from probabilistic reliability analysis, which aggregates data to obtain probabilistic reliability data for a population as opposed to an individual. Such population statistics may be suitable for tasks such as system design, but less helpful for operational and maintenance decisions that focus on individual units, as is the case in CBM. Prognosis, on the other hand, tracks the health of an individual unit.

A prognosis methodology should thus have several important characteristics. It should provide a distribution of RUL as opposed to a point estimate, thus accounting for the uncertainty coming from many sources (variability, information uncertainty, and model uncertainty). In most situations, prognosis about the future is based on diagnosis of the current state; therefore it should account for uncertainty in diagnosis (i.e. uncertainty in damage probability, detection, isolation, and quantification). It should allow easy transition between situations when measurements are available and when they are unavailable. Finally, the methodology should survive rigorous validation.

Prognosis methodologies may be divided into statistical, data-based, model-based, and hybrid approaches (see e.g. Jardine et al. [115], [116]). Statistical approaches include statistical process control [117], logistic regression [118], survival models [116], [119], and stochastic process models [120], [121], [122].

Data-based approaches consist of machine learning methods (support vector machines [123], relevant vector machines [124], neural networks [10,12,13,14,15,16]) and graphical models such as dynamic Bayesian networks (DBNs) hidden Markov models (HMMs) [129], [130]. Liu et al. [131]

use adaptive recurrent neural networks for the estimation of battery RUL. Goebel et al. [132] compare relevance vector machines (RVMs), Gaussian process regression (GPR) and neural network (NN) methods for prognosis. Gebraeel [133] uses NNs for degradation modeling and test the methodology on ball bearings. Saha et al. [134] compare relevance vector machines (RVMs, a Bayesian implementation of support vector machines) and particle filtering to estimate RUL distributions for batteries.

In model-based approaches, system models are used to estimate RUL or other relevant metrics. Such methods rely on accurate physics-based models for prediction. These include physical failure models [135], filtering models [136],[137],[138][139], and statistical models. Orchard and Vachtsevanos [136] use state estimation models combined with particle filtering for diagnosis and estimation of the RUL distribution of a planetary gear. Lorton et al. [137] combine the differential equations of a system with system measurements via particle filtering for probabilistic model-based prognosis.

Hybrid methodologies combine multiple approaches, i.e., a combination of data-driven and model-based approaches. E.g. Kozlowski [140] uses ARMA models, neural networks, and fuzzy logic for estimation of the state of health, state of charge, and state of life of batteries.

DBNs are probabilistic graphical models with diagnostic and prognostic capabilities. They have shown promise in several recent applications. Dong and Yang [45] use DBNs combined with particle filtering to estimate the RUL distribution of drill bits in a vertical drilling machine. While very useful, particle filtering is not the only inference method available for prognosis. Jinlin and Zhengdao [141] use DBNs modeling discrete variables and the Boyen-Koller algorithm for prognosis. Tobon-Mejia et al. [142] use mixtures of Gaussian HMMs (a form of DBN) to estimate the

RUL distributions for bearings. The junction tree algorithm is used for exact inference. The prognosis methodology is validated using the hierarchical metrics proposed by Saxena et al. [143].

4.1.2 Motivation

While the preceding literature review represents a number of prognosis approaches, prognosis is still an emerging research area with room for much additional work. The combination of DBNs and particle filtering has many qualities that are attractive for prognosis:

1) The graphical representation of a problem provided by DBNs aids understanding of interactions in a system.

2) DBNs provide a probabilistic model of the system that accounts for uncertainty due to natural variability, measurement error, and modeling error.

3) DBNs can integrate many types of information that may be encountered during prognosis (including expert opinion, reliability data, mathematical models, operational data, and laboratory data) into a unified system model.

4) DBNs can update the distributions of all variables in the network when observations are obtained for any one or more variables. This allows the most recent system measurements to be accounted for in prognosis.

5) The probabilistic state estimate generated during particle filtering contains information about diagnosis uncertainty that can be used in prognosis and decision making.

6) Particle filtering algorithms can take advantage of parallel computing, thus reducing computation time.

Additionally, prognosis methodologies reported in the literature are mostly application-specific. There is a need for a prognosis methodology that can make use of heterogeneous information and be applied to a wide range of problems.

4.1.3 Contributions

In this chapter, a framework for probabilistic prognosis is proposed. The methodology advances the use of DBNs in prognosis by integrating heterogeneous information. Further, the DBN-based methodology addresses the need for a general prognosis framework for developing validated prognosis methodologies for any system. The methodology provides a means for quantifying diagnosis uncertainty that can be used in prognosis and decision making.

The DBN is constructed from prior information, including physics of failure models. The DBN is a store of prior information, but also provides the means for integrating current measurements into a probabilistic estimate of the current state of a system. In this chapter, particle filter-based inference is used for diagnosis, and forward sampling in the DBN is used for recursive prediction. Diagnosis uncertainty is quantified probabilistic state estimate of the system that results from particle filtering. Sample-based estimates of detection and isolation probabilities as well as density estimates of system damage parameters are developed based on marginal distributions in the state. This information can be used in decision making for improvement of the health monitoring system. The probabilistic state estimate also provides a seamless transition from diagnosis to future state prediction using recursive sampling. The ability of the methodology to estimate RUL is validated using metrics from Saxena et al. [143]. The methodology is illustrated for a hydraulic actuator with a seal leak.

The remainder of this paper is organized as follows. Section 2 details the proposed prognosis methodology, including system modeling, diagnosis, prediction, and validation. In Section 3, the proposed methodology is demonstrated on a hydraulic actuator system with a progressive internal leak. Section 4 discusses conclusions and future work.

4.2 Proposed Prognosis Framework

The challenge of prognosis is to minimize the uncertainty in the estimated distribution of RUL given constraints on available information about the system, operating environment and loading conditions, computational resources, and time horizon. In this chapter, a DBN-based prognosis framework is proposed. The prognosis framework first constructs a DBN-based system model using heterogeneous information sources. Expert opinion, reliability data, mathematical models, and operational and laboratory data are used in the construction of the DBN model. In particular, inclusion of physics of failure models is important in prognosis. The evolution of phenomena such as cracking, wear, and corrosion play a large role in determining the health of a system. The system model is used for diagnosis to obtain information about the current state of the system. A sequential Monte Carlo then predicts future system states and estimates the RUL distribution. Finally, the prognosis capability of the resulting system model, diagnostic, and predictive algorithms are validated using a four step hierarchical procedure. The prognosis procedure is shown in Figure 20.

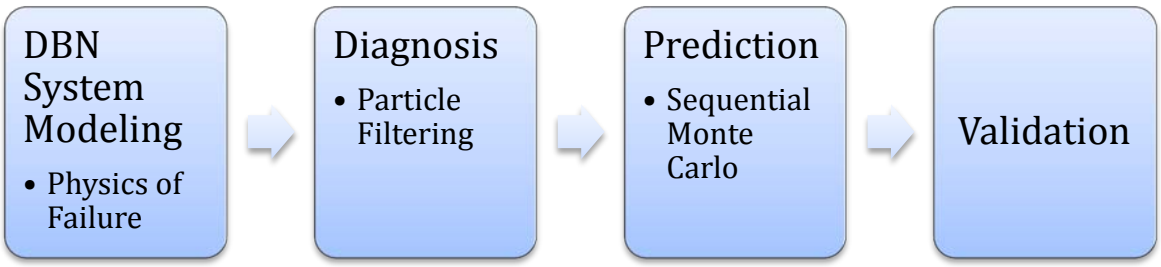


Figure 20. Proposed Prognosis Methodology

4.2.1 DBN System Modeling

The proposed prognosis methodology first builds a DBN system model from heterogeneous information. A methodology for building such a model is discussed in Chapter 2.

4.2.2 Physics of Failure Models

A key distinction between a system model capable of diagnosis and one capable of prognosis is that a prognostic model can estimate the evolution of damage in the future while a diagnosis model only needs the ability to infer the current state of damage. Diagnostic procedures based on fault signatures or pattern recognition are examples of this. While they may be able to detect and isolate damage, quantification can be done using least-squares based estimation, they do not necessarily have any ability to model progressive damage mechanisms such as crack growth, wear, and corrosion. One of the challenges of prognosis is developing accurate and comprehensive physics of failure models. These damage mechanisms are complex, varying with system design and dynamics, and can interact in many ways.

For illustration, the example problem in this chapter considers a dynamic seal in a hydraulic actuator. Seal failure is discussed in great detail in [67].

The volume of material removed from the seal per cycle depends on the friction force and sliding distance per cycle and may be calculated by

$$V(t) = w_{seal}(t)F(t)d(t) \qquad \textbf{Eq. 43}$$

where w_{seal} is the wear rate of the seal in $mm^3/N/m$, F is the frictional force on the seal, and d is the total sliding distance, and t refers to the load cycle.

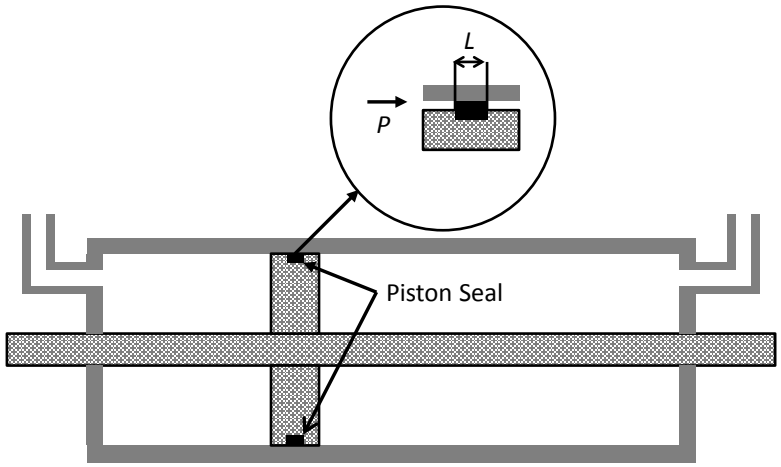


Figure 21. Hydraulic actuator diagram showing dynamic seals

For the seal shown in Figure 21, where L is the contact length of the seal and P is pressure, the leakage area (considered in Eq. 8 - Eq. 20 as in Thompson et al. [72]) based on the volume of material removed is assumed to be $a_{leak} = V(t)/L$.

The wear rate itself varies with factors such as the age of the seal, temperature, contaminants in the fluid. The load experienced by the seal also varies as does the velocity of the actuator. As a result the volume of material removed and leakage area are nonlinear functions. However, for the sake of demonstration, it is assumed that the leakage area and volume of material removed vary linearly. This implies that the wear rate is steady, which is possible outside of the initial wear-in phase and under constant environmental conditions. Additional information on seal wear is provided in Appendix B.

4.2.3 Diagnosis

Diagnosis is the process of detecting and isolating damage in a system and quantifying the magnitude of damage. When the probability of a fault occurring crosses the detection threshold, a fault isolation procedure finds fault set candidates to further analyze. To isolate candidate fault sets, statistical inference computes the probability of each fault set. The magnitude of the fault may then be estimated.

In the context of prognosis, diagnosis (or more generally, filtering) provides the initial conditions for prognosis of a UUT. The initial condition for prognosis has a large impact on the accuracy and precision of the RUL distribution. As such, it is important to understand and account for diagnosis uncertainty. Diagnosis under heterogeneous information using DBNs is discussed in Chapter 3. Details of Bayesian recursive filtering and particle filtering are given.

4.2.4 Prediction

In probabilistic prognosis, possible future states of the system are generated and the remaining useful life (RUL) distribution, $r(t)$, of the particular unit under test (UUT) is estimated. RUL is the amount of time a UUT is usable until corrective action is required and may be measured in hours, minutes, cycles, etc. Measurements are unavailable and the system model is assumed to be valid under future operating conditions. Prediction can be initiated at any time in the life of a system based on the last available state estimate. However, in this chapter, the time of prognosis, t_P, the first time point for which a prognosis estimate is obtained, is after the time of fault detection, t_D. Figure 22 illustrates these important prognosis time indices.

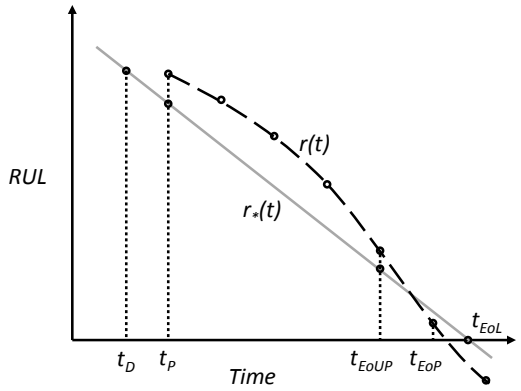


Figure 22. Prognosis time indices: $r_*(t)$ is the ground truth RUL, t_{EoUP} is the end of useful prognosis, dashed line depicts mean $r(t)$.

One approach to prediction when performing particle filtering on a DBN is a basic sequential Monte Carlo. Starting with the last belief state estimate (with measurements available), particles are recursively sampled through the two time slice DBN until some termination criteria is met, such as $Pr(r(t) = 0)$ is above some target threshold. Thus, there are N_s trajectories of the variables of interest beginning at time t, $\{\boldsymbol{\Phi}(t)\}_{i=1}^{N_s}$. Each trajectory consists of a series of predictions for the variables of interest, $\boldsymbol{\Phi}(t) = \{\boldsymbol{\varphi}(t|t), \boldsymbol{\varphi}(t+1|t), \ldots \boldsymbol{\varphi}(EoP|t)\}$, where the end of prediction (EoP) is the time index of the last prediction before the end of life (EoL) is reached. Particle weights are fixed from the last available measurement, as there is no basis for updating the weights (Eq. 39). This results in a particle-based approximation of RUL (similar to the belief state approximation), using the last available set of weights. When a new measurement is obtained, a new RUL distribution is estimated.

To further tailor the prognosis to a particular UUT, the conditional probability distributions in the DBN may be updated as measurements become available. This may be performed via Bayesian

updating of the distributions. If a conjugate prior is available, the update can be performed analytically. Otherwise, techniques such as Markov chain Monte Carlo (MCMC) may be required.

The RUL distribution is sensitive to many aspects of the problem. The initial state estimate provided by the diagnosis algorithm must be accurate. As such, the filtering algorithm and number of particles are important algorithmic design decisions. These decisions also involve a tradeoff between accuracy and computational effort, which must be considered. Optimal sensor placement and improved sensor reliability also impact the accuracy of the diagnosis.

The accuracy of predictive models, including those for inputs (loads) and physics of failure models, is another large source of uncertainty in the RUL estimate. Because the prediction is recursive with no measurements available to correct the prediction, errors in prediction compound quickly and must be minimized.

Three additional factors that affect the diagnosis (and prognosis) of a system are the sensitivity of the system measurements to damage, noise in the training and measurement data, and the sampling frequency at which measurements are obtained. If the sensitivity to damage is low and the noise large, diagnosis may be difficult or impossible. A low sampling frequency can deteriorate the quality of state estimates in systems that depend on previous states, such as nonlinear dynamic systems like the actuator system presented in Chapter 2. The state estimates amount to a linearized approximation of the state across the sampling interval (reciprocal of sampling frequency). As sampling rate decreases, the interval becomes wider and the linear approximation worsens. A higher sampling frequency decreases the sampling interval and reduces the error due to linearization, thus improving state estimates.

Measurement Gaps

Systems may experience periods of times where measurements are unavailable. This may be a result of the system configuration, availability of measurements, failure of sensing systems, or the desire to have system state estimates at a higher frequency than the available measurements. For example, offline inspection data may be available for an aircraft on the ground, while onboard sensing provides a steady stream of information about temperature, altitude, windspeed, pressure, etc. These onboard measurements may only be available for portions of a flight (perhaps during cruising but not takeoff or landing).

Using the same recursive sampling used for RUL estimate, predictions may be produced and used to fill in the information gaps. When a measurement becomes available, the particle filtering algorithm is used to update the last predicted system state. The particle filter update may be performed as long as at least one measurement is available. The process is shown in Figure 23.

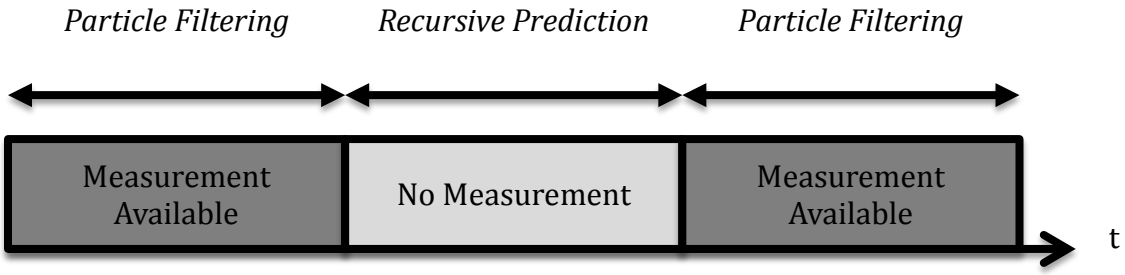


Figure 23. Handling measurement gaps

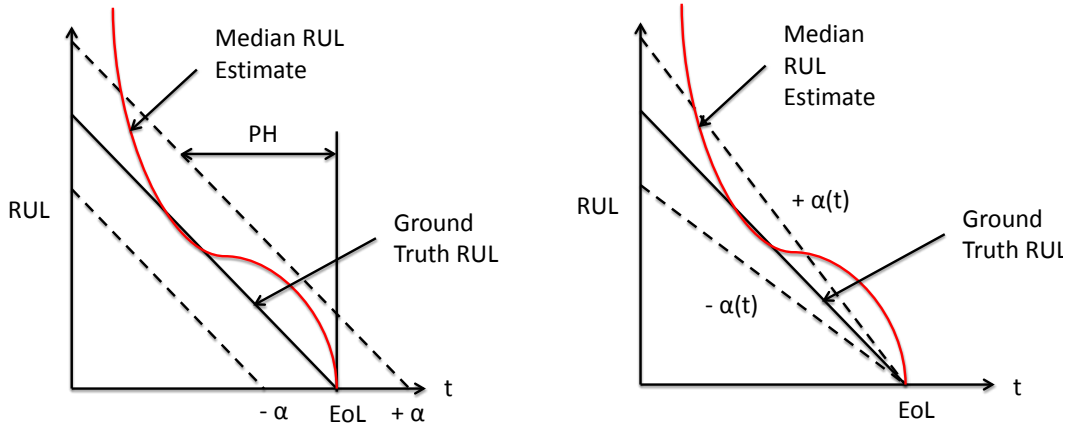
4.2.5 Prognosis Validation

Prognosis validation is essential to establish confidence in the RUL estimate. Many sources of uncertainty, including modeling errors, sensor faults, data noise, and unpredictable loading conditions and operating environments, strongly affect prognosis. Therefore, validation of a prognosis procedure must be done using strong performance metrics. These metrics must be carefully chosen, as many issues arise when evaluating prognosis algorithms, such as time scales or

the ability to improve accuracy as more measurements are obtained [143]. Saxena et al. [143] propose a standard offline four metric hierarchical test to evaluate a prognosis algorithm. This hierarchical test assumes that prognosis improves as more measurements become available. Combined, these four metrics provide a means for testing and comparing prognostic algorithms.

The first two metrics examine the accuracy of the RUL estimates by determining the probability p that the RUL estimate is between $\pm\alpha$ of the ground truth RUL. This probability p is compared to a threshold value, β. It is desirable for p to be greater than β. The primary difference between the first two metrics is in how α is determined, which results in a stricter test for the second metric than the first.

In the first metric, prognostic horizon (PH) is considered. Prognostic horizon indicates the time at which RUL estimates using a particular prognostic algorithm for a particular system are within acceptable limits. The upper and lower limits are the ground truth RUL plus or minus a constant α, which is some percentage of the EoL value. PH is the difference between the true EoL time and the time when the prognostic algorithm attains this acceptable level of accuracy ($p > \beta$). As this is a validation metric, the true EoL is known. A longer PH is indicative of a better prognostic algorithm. Figure 24a provides a visual representation of prognostic horizon.



a) b)

Figure 24. a) Prognostic horizon with +/- α bounds about the ground truth RUL b) +/- α bounds for evaluating α-λ accuracy

Prognostic horizon maintains upper and lower bounds that are always the same distance from the true RUL. The second validation metric, $\alpha - \lambda$ accuracy, utilizes a stricter criterion that gradually tightens the limits about the RUL estimate (Figure 24b). Additionally, the accuracy of the RUL is considered at time t_λ, where $0 \leq \lambda \leq 1$, $t_\lambda = t_P + \lambda(t_{EoL} - t_P)$, and t_P is the time at which a prognosis estimate is first obtained. This metric reflects the idea that, as more information is collected about the system, the RUL estimate is expected to improve, and thus the accuracy requirement for the RUL estimate should become more stringent. The $\alpha - \lambda$ accuracy is equal to 1 when the increasingly stringent accuracy requirements are met, and zero otherwise.

In step three, the relative accuracy (RA) of the prognostic algorithm is calculated. Instead of merely indicating that accuracy requirements have been met, the accuracy of the RUL estimates are quantified. At t_λ

$$RA_\lambda = 1 - \frac{|r_*(t_\lambda) - r(t_\lambda)|}{r_*(t_\lambda)} \qquad \textbf{Eq. 44}$$

where $r(t_\lambda)$ is a central tendency point such as the mean or median of the RUL estimate at t_λ and $r_*(t_\lambda)$ is the ground truth RUL. The RA is computed separately for each time step at which RUL is estimated. RA is a value between 0 and 1, and values closer to 1 indicate better accuracy.

Finally, if the prognostic algorithm satisfies all the previous metrics, a final metric to compute is convergence. Convergence is a measure of how quickly a metric, such as RA, improves with time. A high rate of convergence is desirable and leads to a larger PH. To estimate convergence of a prognosis algorithm based on some metric *M,*

$$C_M = [(x_c - t_P)^2 + {y_c}^2]^{1/2} \quad \textbf{Eq. 45}$$

where

$$x_c = \frac{\frac{1}{2}\sum_{i=P}^{EoUP}({t_{i+1}}^2 - {t_i}^2)M(t_i)}{\sum_{i=P}^{EoUP}(t_{i+1} - t_i)M(t_i)} \quad \textbf{Eq. 46}$$

and

$$y_c = \frac{\frac{1}{2}\sum_{i=P}^{EoUP}({t_{i+1}}^2 - {t_i}^2)M(t_i)^2}{\sum_{i=P}^{EoUP}(t_{i+1} - t_i)M(t_i)} \quad \textbf{Eq. 47}$$

$M(\mathrm{t_i})$ is the non-negative prediction accuracy, EoUP is the end of useful prediction, and P is the time at which the prognosis algorithm makes its first prediction. End of useful prediction is the time after which corrective action cannot be performed before EoL. A high rate of convergence is better and leads to a larger PH.

The effects of noise and sampling rate, as discussed in Chapter 3, Section 3.5, are apparent during prognosis and prognosis validation. Poor initial conditions provided from diagnosis limit the accuracy of prognosis from the beginning. Further, accumulated error due to low sampling rate may cause noticeable deviations from ground truth RUL estimates.

4.2.6 Summary of Prognosis Framework

This section presented a framework for probabilistic prognosis. DBNs are used as a system modeling paradigm due to their ability to handle uncertainty and to integrate many types of information, both in the offline model construction phase and the online belief state updating phase. For prognosis, it is of particular importance to model complex physics of failure phenomena and integrate such models into the DBN. After the DBN model is established, the model is used for tracking the state of a particular UUT. Particle filtering is used to update the belief state as new measurements are obtained. Uncertainty in the state estimate (diagnosis) is quantified, and when a fault is detected, estimation of RUL via recursive prediction begins. The result is an estimate of the distribution of RUL. Section 2.4 considers the situation when there are gaps in the availability of measurements.

When a prognosis procedure (DBN model of system combined with available measurements and filtering algorithm), is designed for a particular system, it is then validated using the 4 step hierarchical procedure outlined in Section 4.2.5.

4.3 Illustrative Example

The hydraulic actuator system described in Chapter 1 was considered to demonstrate the proposed methodology. Such a system is often used to manipulate the control surfaces of aircraft. The system consists primarily of three subsystems: a hydraulic actuator, critical center spool valve, and an axial piston pump (Figure 4, Chapter 2). Expert opinion, reliability data, mathematical models, operational data, and laboratory data were used to construct a DBN model of the spool valve and hydraulic subsystems. Next, the DBN model was used to diagnosis a seal leak in the actuator. The RUL of the system was then estimated.

4.3.1 DBN Model Construction

The DBN model for the actuator is constructed from heterogeneous information as in Chapter 2. The resulting DBN is shown in Figure 8.

4.3.2 Diagnosis

Diagnosis is performed as described in Section 2.3. The actuator was operated for 20 seconds with a leak occurring after 6 seconds. At this point, the system has already reached the steady state. Synthetic measurement data was generated using the Simulink model for two cases. In the first case, the data was not resampled (i.e. the sampling frequency remained 4 samples per second). In the second case, data was resampled at 2 samples per second. This was done to compare the effect of sampling rate on diagnosis and prognosis. The system responses and load were assumed to be measurable while the system parameters including wear rate and leakage area were assumed to be unobservable. Inference via particle filtering (N_s = 250) was performed on the DBN to obtain filtered estimates of the system state.

After obtaining the state estimate at cycle t, the probability of detection was calculated as in Section 2.2. If the probability of detection exceeded 95%, an alarm was triggered. The fault was then isolated and quantified. Figure 25 shows maximum a posteriori (MAP) estimates of the system responses against their measured values for the 2 samples/sec series (not shown are responses for the 4 samples/sec series that appear similar). It is seen that the MAP system responses track the measured values closely. Figure 26 shows the MAP estimates of the system parameters for the 4 samples/sec series, including the leakage area, and load against the ground truth values. This figure shows how the leakage area changes with time and how well the filtering procedure can infer the value of the leakage area. The system responses in Figure 25 are sensitive to changes in the supply pressure and leakage area, but insensitive to changes in the fluid bulk modulus. Changes in bulk

modulus, however, may result in effects such as changes in wear rate that have not been included in the ground truth model. In both Figure 25 and Figure 26, the good estimates may be attributed to the use of an accurate physics-based model, but also to the use of synthetic measurement data whose sampling rate matched the internal sampling rate of the generative model. Thus, the performance of filtering is favorably biased.

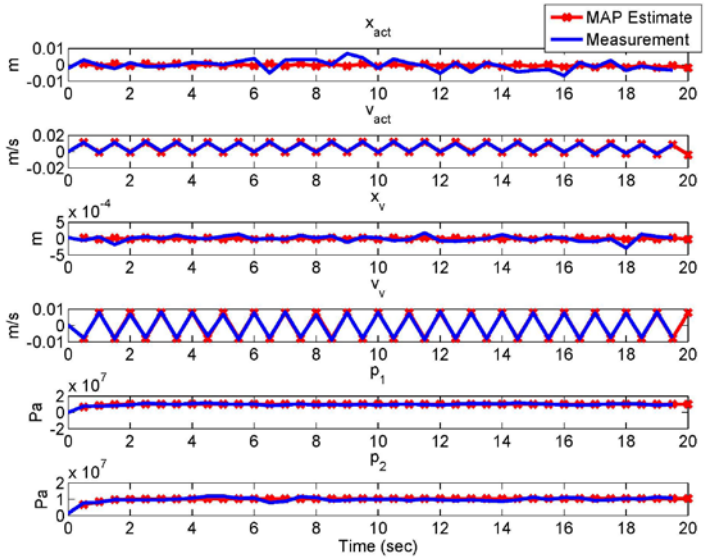


Figure 25. MAP estimates and measured values of actuator position and velocity, servovalve position and velocity, and pressure in each actuator chamber (2 samples/sec)

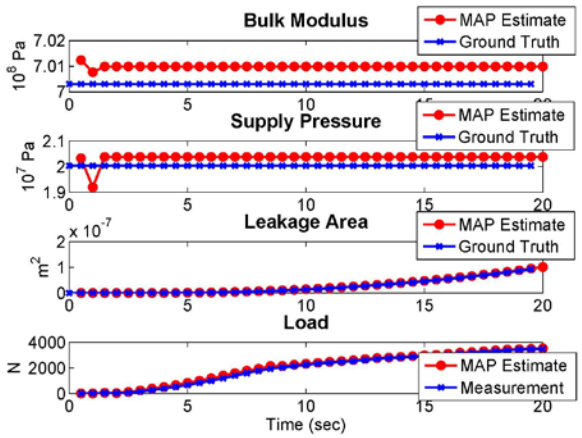


Figure 26. MAP estimate system parameters and load with ground truth and measured values (2 samples/sec)

4.3.3 Diagnosis Uncertainty

Diagnosis uncertainty was quantified during filtering as in Section 3.3.3. Figure 27 shows the damage probability as it evolves with time for the 2 samples/sec case. The damage probability passes the alarm threshold soon shortly after the actual occurrence of the fault.

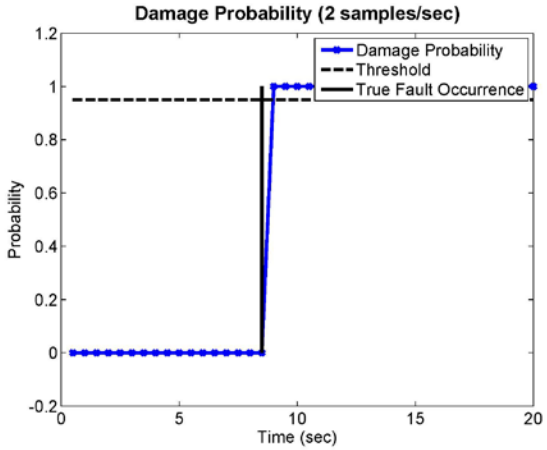


Figure 27. Damage probability with actual fault time (2 samples/sec)

At the time of detection (t = 18 cycles / 9 secs), the all particles for the 2 samples/sec series have a value of 1.038E-8 for the leakage area. This is in contrast to a ground truth value of 9.339E-9. Note that all particles have the same value due to resampling. If this damage is judged to be severe

(having a high probability of exceeding some threshold value), this may provide a decision-maker with cause to take immediate action. If damage is considered less severe, a decision-maker may consider prognostic information and delay action. In any case, these leakage area values are the initial condition for prognosis starting at t = 9 sec. Figure 27 may be used to signal an alarm that a fault has occurred. In some cases, this alarm can be used to initiate RUL estimation or other tasks such as inspection.

4.3.4 Prediction

After alerting the presence of a leak above the leakage threshold of 7.578E-9 m^2, estimation of the RUL distribution was performed as in Section 4.2.4. The RUL distribution assumes a failure threshold for leakage area of 4.123E-8 m^2. The RUL distribution at the 25th cycle is shown in Figure 28 for the 2 samples/sec series.

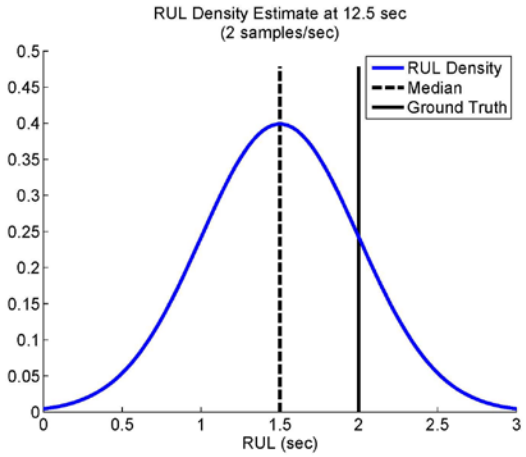


Figure 28. RUL density estimate at t = 25 sec

4.3.5 Computational Effort

The above diagnosis has been performed with 12 Intel Nehalem processor cores working in parallel. Each belief state update in diagnosis takes approximately 30 seconds. The majority of the

computational effort is directed towards solving the differential equations for the actuator in the Matlab Simulink environment.

4.3.6 Prognosis Validation

By continuing to estimate the new RUL distribution as new measurements become available, the performance of the prognostic algorithm may be evaluated. In Figure 29, median RUL estimates are plotted against the ground truth RUL with +/- α bounds for the 2 samples/sec series. The +/- α bounds are selected to be +/-10% of the ground truth EoL about the current ground truth RUL. Figure 30 indicates whether the probability of the RUL estimate being between the +/- α bounds at a particular time is greater than a

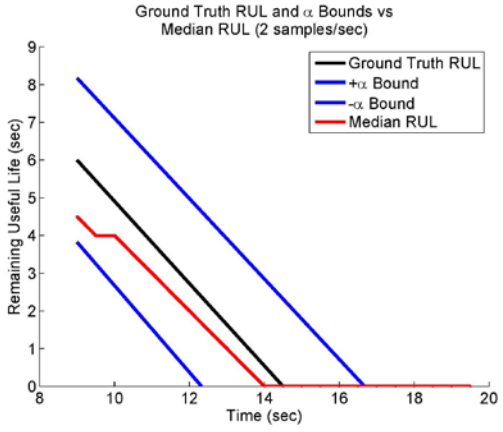


Figure 29. Ground truth RUL, median RUL, and α bounds with α = 0.10 (2 samples/sec)

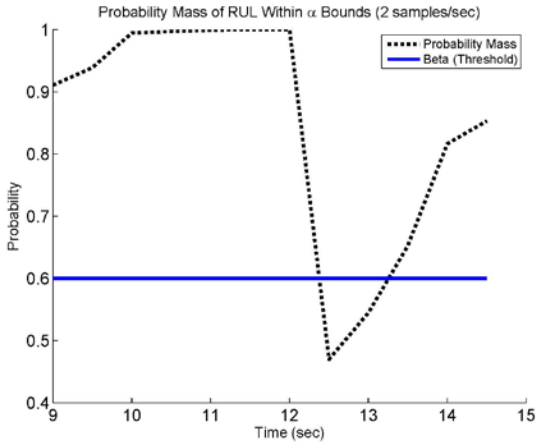


Figure 30. Probability that RUL is within α bounds with α = 0.10 (2 samples/sec)

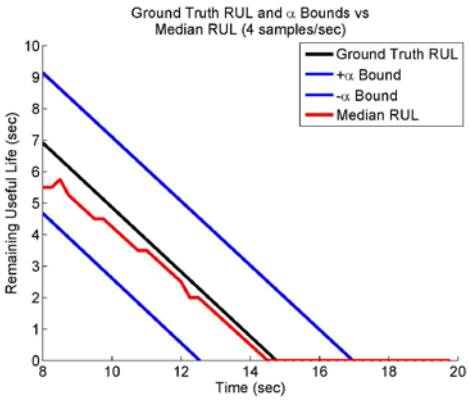


Figure 31. Ground truth RUL, median RUL, and α bounds with α = 0.10 (4 samples/sec)

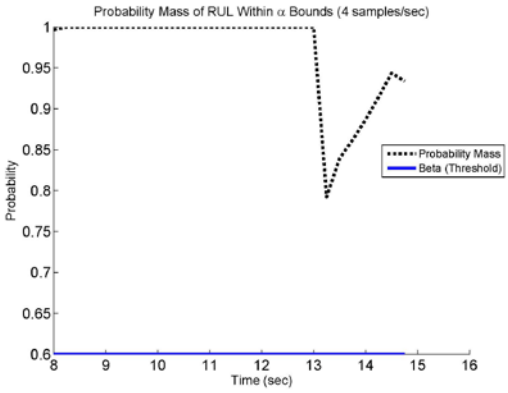


Figure 32. Probability that RUL is within α bounds with α = 0.10 (4 samples/sec)

threshold value, taken as 0.8. From this information, it is also determined that the prognostic horizon is about 2.5 cycles (1.3 seconds) because the first time that $0.6 \leq \pi\left[r\left(t_j\right)\right]_{-\alpha}^{+\alpha}$ is at t = 7, the EoL is t = 26.5. This is 2.5 cycles before the EoL. Similarly, Figure 31 and Figure 32 show the same plots for the 4 samples/sec series and have a PH of 27.5 cycles (6.9 seconds). The increase in PH in the 4 samples/sec case over the 2 samples/sec case may be attributed to the sampling rate, as all else is the same. A higher sampling rate ensures nonlinear effects are not missed.

+/- α bounds that narrow as the EoL approaches are considered for both sampling rates in Figure 33 and Figure 34 for λ = 0.5 and α = 0.20. λ = 0.5 considers the accuracy of the RUL estimate halfway between the time of prognosis and end of life, termed t_λ. The bottom plots of Figure 33 and Figure 34 shows the λ-α accuracy, which is a binary value that indicates whether the probability of the RUL estimate being between the +/- α bounds at a particular time is greater than a threshold value, taken as 0.8 here. Note that the red line only indicates the median RUL estimate and that λ-α accuracy is calculated using the RUL distribution. The λ-α accuracy is generally zero for the 2 sample/sec series, indicating that the RUL estimate is too diffuse to pass this test. The higher sampling rate results in much better performance on this metric for the 4 sample/sec case than the 2 sample/sec case. Performance may be further improved by using additional samples.

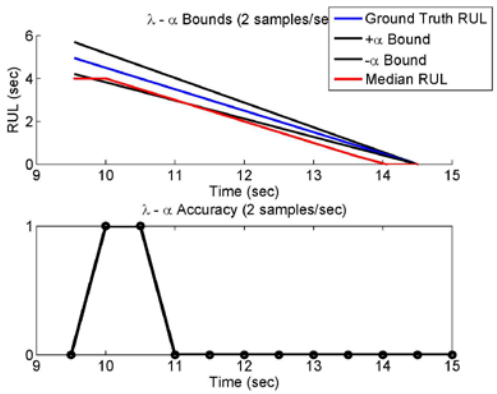


Figure 33. Bounds used for calculating λ-α accuracy with α = 0.20 (top) and λ-α accuracy (bottom) for 2 samples/sec

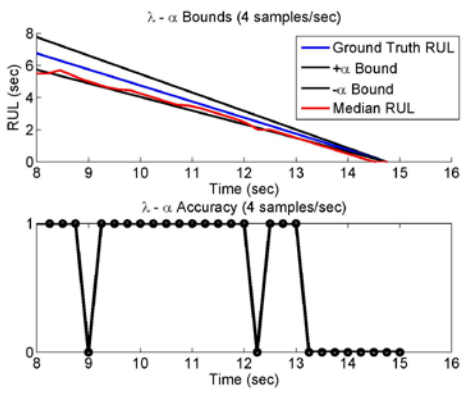


Figure 34. Bounds used for calculating λ-α accuracy with α = 0.20 (top) and λ-α accuracy (bottom) for 4 samples/sec

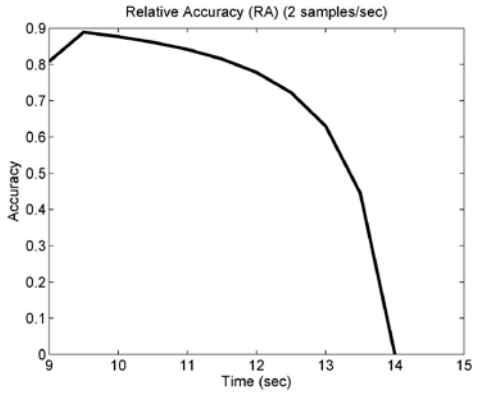


Figure 35. Relative accuracy based on median RUL estimate 2 samples/sec

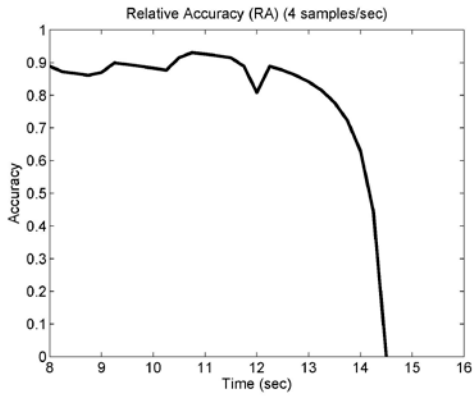


Figure 36. Relative accuracy based on median RUL estimate 4 samples/sec

Figure 35 and Figure 36 show the relative accuracy of the RUL density estimate based on the median RUL value, and shows that the median values are accurate. The 2 sample/sec series again lags the 4 sample/sec series in performance. The drop towards zeros accuracy is more a mathematical artifact than an indicator of poor RUL estimation. Considering Eq. 44, if the median RUL is zero, the RA is zero, regardless of the absolute difference between the median RUL and ground truth. Based on the relative accuracy and excluding any zero values, the convergence is estimated to be 19.95 for the 4 sample/sec series. Convergence is not calculated for the 2

sample/sec series because it fails on λ-α accuracy. When comparing prognostic algorithms, larger convergence values are desirable.

4.3.7 Discussion

The DBN-based methodology successfully integrates heterogeneous sources of information to diagnose the system and estimate RUL. Particle-filter based inference provides a seamless method for switching between probabilistic diagnosis and prediction while facilitating uncertainty quantification.

The prognosis validation results indicate that the methodology provides reasonable median estimates of RUL, even as the RUL density estimates are sometimes diffuse. The sampling rate of the measurements is a large factor in whether or not prognosis is successful. Inclusion of inspection data may reduce the uncertainty in the leak area estimate and thus the RUL estimate. The accuracy of prognosis, of course, will vary depending on the system, available information, loading conditions, and environmental conditions.

Computational effort is a persistent issue in particle-based methodologies, affected by the complexity of the system, models involved, simplifying assumptions, filtering algorithms, etc. The prognosis methodology described in this paper is flexible with respect to these decisions, so computational effort will vary. Reductions in computational effort may be achieved by using reduced order models (e.g. linearized model of actuator), feature selection and dimensional reduction techniques, and improved particle filtering techniques (e.g. Rao-Blackwellized particle filter [5]). Additionally, the ability to massively parallelize a particle filter using modern central processing units (CPUs) and graphics processing units (GPUs) provides the potential for greatly

decreased computation times. The actuator problem would see immediate decreases in computational time with additional processor cores.

Thus far, the methodology has only been demonstrated using synthetic data, and needs to be tested further using real-world data. Further, more complex physics of failure models should be considered.

4.4 Conclusion

A methodology for DBN-based probabilistic prognosis is presented in this chapter, considering heterogeneous information sources and diagnosis uncertainty. First, expert opinion is used to establish the system definition and basic assumptions. Reliability data is used to calculate conditional probabilities for fault indicator variables for damage at the support and a crack. Operational and laboratory data are organized in a database and used for estimating a polynomial regression model. This system model is used in online diagnosis via particle filter-based inference. The particles resulting from filtering integrate seamlessly into a sequential Monte Carlo predictive procedure, used for estimating RUL distribution. The prognosis results are validated using a four step hierarchical procedure. In the future, the methodology needs to be extended to systems of larger dimension, thus requiring feature selection, dimensional reduction, and more efficient inference.

CHAPTER V.

5. RISK-INFORMED MAINTENANCE AND MISSION DECISION-MAKING

5.1 Introduction

Decision making methodologies in engineering system health management take remaining useful life (RUL) estimates derived from prognosis to develop and select maintenance policies or perform condition-based maintenance. While advanced diagnostics and prognostics provide vital information, good decision making with that information requires a methodology that effectively utilizes available information.

In this chapter, mission-level decision-making when uncertain diagnosis and prognosis estimates are available is considered. Mission-level decisions decide between various usage scenarios that also include maintenance as an option. Additionally, it is desirable that the diagnosis and prognosis be based on all possible relevant information about the system. This information may be heterogeneous, including expert opinion, operational and laboratory data, mathematical models, and reliability data.

For example, an aircraft ends a mission in a particular heath state, perhaps with diagnosed faults. A decision maker must decide if that aircraft is suitable for its next mission, a reduced mission, or requires immediate maintenance based on the results of diagnosis and prognosis, which is based on the expected load profiles for the possible missions. Additionally, other aircraft with their own health states, diagnoses and prognoses may be available, resulting in an assignment problem.

This chapter proposes a diagnosis prognosis based decision-making methodology that assigns systems to tasks based upon their current and estimated health. First a DBN system model is constructed from heterogeneous information for each unique system. The DBNs are then used to diagnose the current state of each system. The diagnosis and prognosis results provided by this methodology then feed into an optimization problem that determines the best allocation of systems to mission alternatives.

This primary contribution of this chapter is a methodology for decision-making that 1) combines heterogeneous information in a useful manner within a DBN system model and 2) enables optimal decision making based on diagnostics as well as remaining useful life (RUL) estimates. Previous work in the area of decision-making has seen RUL estimates and diagnostics used primarily for scheduling maintenance of a system, generally without considering multiple systems and non-maintenance options. There is a need to consider resource allocation with multiple systems and possible missions while maintaining the health of the systems.

The following sections review the methodology for constructing a DBN model from heterogeneous information, diagnosis, and prognosis. After reviewing these topics, details of a procedure for optimal mission and maintenance decision-making are provided.

5.2 Model Construction

A DBN model of the system is constructed as in Chapter 2. This probabilistic model integrates heterogeneous information (Figure 37) and enables diagnosis via particle filtering and prognosis. The availability of prognostic information is an important requirement for decision-making.

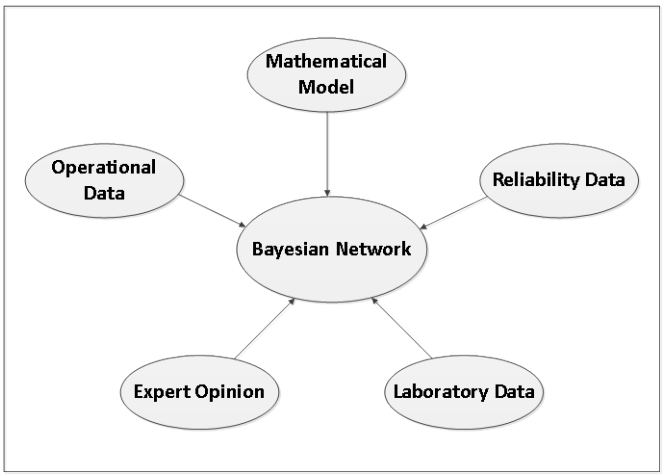


Figure 37. Integration of heterogeneous information

5.5 Decision-Making

The power of diagnosis and prognosis information is its ability to improve the safety and reduce the cost of operating a system or collection of systems. Suppose there is a set of *S* systems and another set of possible mission alternatives, *A*. Each system *i* and mission pairing *j* has a cost *C*(*i,j*). This is a classic assignment problem. The cost function that is minimized is

$$f(x) = \sum_{i \in S} \sum_{j \in A} C(i,j)\, x_{ij} \qquad \textbf{Eq. 48}$$

x_{ij} is the assignment of system *i* to mission alternative *j* and takes a value of 1 if assigned and 0 if not.

The restraints are typically that each system must have a single assignment

$$\sum_{j \in A} x_{ij} = 1 \text{ for } i \in S, \qquad \textbf{Eq. 49}$$

Each mission must have a single system assigned to it

$$\sum_{i \in S} x_{ij} = 1 \text{ for } j \in A, \qquad \textbf{Eq. 50}$$

and that

$$x_{ij} \geq 0 \text{ for } i, j \in S, A. \quad \textbf{Eq. 51}$$

Further, the standard problem assumes an equal number of systems and mission alternatives. However, it need not be the case that there be the same number of systems and mission alternatives and that every mission alternative have a system assigned to it. There may be limits on the number of systems that can be assigned to a particular mission. In this chapter, only one system may be assigned to a mission alternative. However, the constraints may vary by situation. Further, the value x_{ij} is limited to 0 or 1, making the optimization discrete.

If multiple limit states are present, the cost function may be expanded by considering a cost function and weight for each limit state in the set of limit states, *G.* The cost function is then

$$f(x) = \sum_{k \in G} \sum_{i \in S} \sum_{j \in A} w_k C_i(j,k)\, x_{ij} \quad \textbf{Eq. 52}$$

where w_k is a weight given to each limit state. Note that the cost C now varies depending on the limit state.

The crux of the optimization is determining the values $C_i(j,k)$. Each system in S has a remaining useful life (RUL) distribution for each limit state, π_{ijk}. Each mission alternative has a threshold RUL value g_{jk} for each limit state.

$$C_i(j,k) = p_{ijk} = Pr\left(RUL_{ijk} < g_{jk}\right) \quad \textbf{Eq. 53}$$

Thus, the optimization seeks to find the assignments x_{ij} which minimize the probability that any of the limit states are violated for any of the mission alternatives.

Note that, if limit states are characterized by abrupt faults, the probabilities p_{ijk} may be calculated without need for prognosis. For example, p_{ijk} may be calculated directly if the time to failure distribution is known.

5.6 Numerical Example

5.6.1 Problem Description

To illustrate the methodology, a hydraulic actuator system is considered. Hydraulic actuators provide the force to move critical flight control surfaces on many aircraft. In the hydraulic actuator described in Chapter 2, an axial piston pump pressurizes hydraulic fluid that moves a piston and moves the control surface. The position of the pump is modulated by a servo-valve, which controls the flow of fluid into the chambers of the actuator.

Several potential faults are considered for the actuator, based on faults discussed in [68] and the [55]. In this section, the seal leak is exclusively considered, as it is a progressive fault that can be used to estimate RUL.

For the sake of illustration, consider three aircraft whose airworthiness is determined by their actuators. Suppose these three such aircraft are to be assigned to missions A, B, and C, where missions A and B correspond to different durations and load profiles and mission C is a maintenance option. Each mission has an allowable limit for the volume of leakage as well as upper and lower limits on the supply pressure and hydraulic fluid bulk modulus.

5.6.2 Actuator Health States

Each of the three actuators is subjected to different load profiles, control signals, system parameters (supply pressure, bulk modulus), leakage, faults, and total time of operation. Due to these different histories, each actuator has a different health state (initial condition for prognosis) at the time the assignment problem is performed.

5.6.3 Mission

Each mission has a unique duration, allowable leakage, probabilistic loading, and deterministic control signal. Prognosis is performed for each actuator for each possible mission, resulting in 9 instances of prognosis. The RUL distribution π_{ijk} was estimated by determining how long it would take an actuator to exceed the threshold leakage for a particular mission. Based on the RUL distribution, the probability of the RUL being less than the mission duration, p_{ijk}, was estimated.

5.6.4 Cost Function and Assignments

With the probabilities p_{ijk} available, the cost matrix $C_i(j,k)$ was then populated by Eq. 53. For the 5 abrupt faults, exponentially distributed time to failure densities were used to estimate p_{ijk}. Table 17, Table 18, and Table 19 show the cost matrix for each actuator for each of the three missions (M1, M2,M3). A value of 1 for an abrupt fault indicates that the fault has already occurred. The failure rate for each abrupt fault is assumed to be the same regardless of the system or mission, however the amount of operational time for each system varies, resulting in different probabilities.

Weights were assigned for each of the six limit states (0.8 for the seal leak and 0.04 for each of the other 5 limit states). The cost function (Eq. 52) was then minimized to find the optimal assignment using Microsoft Excel. The resulting assignment is shown in Table 20.

Table 17. Actuator 1 cost matrix

Actuator 1	M1 $Pr(RUL_{11k} < g_{1k})$	M2 $Pr(RUL_{12k} < g_{2k})$	M3 $Pr(RUL_{13k} < g_{3k})$
Seal Leak	0.84	0.72	.7
Electrical Fault	.022	.025	.027
Valve Fault	1	1	1
Pump Fault	.054	.061	.065
Water Leak	.104	.118	.139
Air Leak	.124	.139	.150

Table 18. Actuator 2 cost matrix

Actuator 2	M1 $Pr(RUL_{21k} < g_{1k})$	M2 $Pr(RUL_{22k} < g_{2k})$	M3 $Pr(RUL_{23k} < g_{3k})$
Seal Leak	0.929	0.870	0.830
Electrical Fault	0.024	0.027	0.029
Valve Fault	0.092	0.102	0.110
Pump Fault	0.058	0.065	0.070
Water Leak	0.113	0.126	0.135
Air Leak	0.134	0.150	0.160

Table 19. Actuator 3 cost matrix

Actuator 3	M1 $\Pr(RUL_{31k} < g_{1k})$	M2 $\Pr(RUL_{32k} < g_{2k})$	M3 $\Pr(RUL_{33k} < g_{3k})$
Seal Leak	0.830	0.750	0.710
Electrical Fault	1.000	1.000	1.000
Valve Fault	0.081	0.092	0.099
Pump Fault	0.051	0.058	0.063
Water Leak	0.100	0.113	0.122
Air Leak	0.118	0.134	0.144

Table 20. Assignments

	M1	M2	M3
A1		X	
A2	X		
A3			X

5.6.5 Discussion

The example decision making problem has illustrated the basic steps in allocating systems to missions using diagnosis and prognosis results. The procedure can be expanded to handle more systems, limit states, and missions – including maintenance. Additional work needs to be done to solve the decision making problem for a system composed of various components, with some of these components being redundant in parallel and series configurations. More realistic systems and mission need to be considered.

5.7 Conclusion

A methodology for mission-level decision making has been described in this chapter. The methodology incorporates information from expert opinion, operational and laboratory data, reliability data, and mathematical models by constructing a DBN model of a system. Through a particle filter, the most recent measurements from the system are included in the most recent estimate of the system's health. In prognosis, an estimate of the RUL distribution is obtained. Thus in decision making, past information about the system, the current health state, and predictions about the future health state are included.

In future work, the methodology needs to be tested on a realistic system. The methodology should also be expanded to consider series and parallel system configurations. Subsequently, testing with real-world data and real-world applications should be performed to develop confidence in the methodology.

CHAPTER VI.

6. CONCLUSION

6.1 Summary

The demand for more reliable and more cost effective systems has resulted in a need for advanced system analysis methods. These methods support mission and maintenance decision-making and aim to increase system availability, uptime, and optimize resource allocation. Ideally, these methods take advantage of all information available about a system. However, the heterogeneous nature of information requires a framework for decision support that can integrate information in many formats from various sources and account for many types of uncertainty.

This dissertation proposed a dynamic Bayesian network (DBN) based methodology for decision-making under uncertainty and in the presence of heterogeneous information. The methodology is divided into the tasks of system modeling, diagnosis, prognosis, decision-making. The system-modeling approach in Section 2 constructs a DBN model of the system that incorporates expert opinion, reliability data, mathematical models, operational data, and laboratory data. The probabilistic nature of DBNs allows them to handle uncertainty in the information used to build the DBN. Further, the ability to update and sample the distributions of the variables in light of new information makes DBNs suitable for diagnosis and prognosis. The system modeling methodology was demonstrated for a hydraulic actuator.

After establishing a methodology for building a DBN system model in the presence of heterogeneous information, the diagnosis problem was then considered in Section 3. Once the DBN

model of the system is constructed, approximate inference via particle filtering was pursued to update the system variables in light of new information. The online state estimates provided during particle filtering provides important information that enables quantification of the uncertainty in fault detection, isolation, and quantification. Diagnosis is demonstrated for a cantilever beam with potential damage at the support and a midspan crack.

When a fault is diagnosed, it is necessary to determine how that fault affects system performance. In the prognosis methodology of Section 4, the future state of the system is predicted and the remaining useful life (RUL) estimated. Conveniently, prediction may be done using a sequential Monte Carlo beginning with the sample-based state density estimate obtained during diagnosis. This state estimate reflects the knowledge contained in the DBN model as well as information contained in measurements. Special consideration is given for physics of failure models, as accurate modeling of progressive damage is crucial in estimating RUL. Prognosis is then validated using an offline procedure that considers the accuracy of the RUL estimate. The methodology is illustrated on a hydraulic actuator with a seal leak.

With prognosis and diagnosis information available, decision-making may commence. Chapter 5 considered an optimization problem for assigning a set of like systems to a set of missions, including maintenance. The cost function is based on the RUL estimates obtained for each system on each mission. The optimization minimizes the total probability of failure across all systems and missions. Thus, RUL information is turned into an actionable plan for a decision maker. The decision-making methodology is demonstrated for a set of hydraulic actuators with multiple possible missions.

Some of the key contributions of this work are:

1. Proposed a methodology for system modeling under heterogeneous information. The methodology integrates expert opinion, reliability data, mathematical models, laboratory data, and operational data into a dynamic Bayesian network system model.

2. Developed a methodology for DBN-based diagnosis of a mechanical system under heterogeneous. The methodology is a general approach that uses the existing DBN model of the system. Additionally, the methodology can handle systems with multiple faults.

3. Quantified diagnosis uncertainty in the context of particle filtering by estimating the probability of detecting any fault, estimating the isolation probabilities of all fault combinations, and estimating the distribution of damage parameters.

4. Developed a methodology for prognosis of a system under heterogeneous information. The methodology seamlessly integrates with the diagnosis procedure and utilizes the same DBN model of the system.

5. Formulated an optimization problem for solving the decision-making problem with multiple systems and multiple assignments. The formulation considers multiple limit states and accounts for probabilistic diagnostic and prognostic information.

6.2 Future Work

The methodology presented in this dissertation provides many different opportunities to direct future research efforts. In general, these efforts are focused on testing the methodology and improving supporting technologies.

Synthetic data has been used to illustrate the methodology using a hydraulic actuator system and cantilever beam. However, the methodology still needs to be tested using real-world data to demonstrate its viability under actual operating conditions. Testing should be done starting with relatively simple systems, as superior supporting technologies may be required to consider more complex systems.

The primary hurdle for the methodology proposed in this dissertation is computational effort when learning DBNs and performing diagnosis and prognosis. Supporting technologies such as dimensional reduction, feature selection, and particle filtering influence the required amount of computational effort. For example, Rao-Blackwellized particle filtering can provide a significant reduction in computational effort. Utilization of more advanced technologies in these areas would allow the proposed methodology to be used for more complex systems.

Research into the DBN representation of dynamic systems may also prove beneficial. Dynamic systems, such as the hydraulic actuator, that are described by ordinary differential equations (ODEs), have been modeled using deterministic nodes that call an ODE solver. A more elegant and efficient solution may potentially be found by exploring the relationship between DBNs and ODEs, thus enabling rapid adaptation of the proposed methodology to a wide variety of systems.

There are many possible extensions to the methodology presented in this dissertation. As new technologies are developed, there will be a need to test and validate them and to compare their performance to existing technologies, on their own and in the context of system modeling, diagnosis, prognosis, and decision-making.

APPENDIX

A. Gaussian Process Regression

An introduction to Gaussian processes by Rasmussen [144] is summarized in this appendix. A Gaussian process (GP) is a set of random variables where the distribution over any subset of the variables is a joint Gaussian distribution, consistent with the joint Gaussian distributions over other subsets of variables. A GP is fully specified by a mean function $m(x)$ and covariance function $k(x, x')$. A function f may be distributed as a GP, $f(x) \sim GP(m, k)$.

The difference between a GP and a Gaussian distribution is that a GP is specified by functions instead of vectors of values. Gaussian distributions are thus indexed by the positions of the values in the vectors while GPs are indexed by the argument x, which has a corresponding value $f(x)$.

Given a set of input data, $\mathbf{x} = \{x_1 \dots x_n\}$, the mean and covariance functions may be evaluated such that $\mu_i = m(x_i)$ for $i = 1 \dots n$ and $\Sigma_{ij} = k(x_i, x_j)$. With the set of corresponding responses $f(\mathbf{x})$ available, $\mathbf{f} \sim N(\mu, \Sigma)$. The process is now a multivariate Gaussian distribution.

The purpose of the GP regression is given test inputs, some training cases $\mathbf{x}$ with function values $\mathbf{f}$ and test inputs $\mathbf{x}_*$, estimate the function values $\mathbf{f}_*$. The joint distribution over all of these quantities of interest is

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim N\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^T & \Sigma_{**} \end{bmatrix} \right) \tag{1}$$

where $\boldsymbol{\mu} = m(x_i), i = 1 \dots n$ for the training means and similarly for the test means $\boldsymbol{\mu}_*$. Σ is the matrix of training set covariances, Σ_* the training-test covariances, and Σ_{**} the test set covariances. The conditional distribution of $\mathbf{f}_*$ given $\mathbf{f}$ is

$$\mathbf{f}_* | \mathbf{f} \sim N\left(\boldsymbol{\mu}_* + \Sigma_*^T \Sigma^{-1} (\mathbf{f} - \boldsymbol{\mu}), \Sigma_{**} - \Sigma_*^T \Sigma^{-1} \Sigma_* \right) \tag{2}$$

The posterior process given the data D is

$f|D \sim GP(m_D, k_D)$,

$$m_D(x) = m(x) + \Sigma(\mathbf{x}, x)^T \Sigma^{-1}(\mathbf{f} - \boldsymbol{\mu}) \tag{3}$$

$$k_D(x, x') = k(x, x') - \Sigma(\mathbf{x}, x)^T \Sigma^{-1} \Sigma(\mathbf{x}, x')$$

where $\Sigma(\mathbf{x}, x)$ is the vector of covariances between the training data **x** and the test data point *x*. Generally speaking, the mean function $m(x)$ and covariance function $k(x, x')$ have some set of hyperparameters θ. These hyperparameters may be estimated by maximizing the the log marginal likelihood,

$$L = log\big(p(\mathbf{f}|\mathbf{x}, \theta)\big) = -\frac{1}{2} log|\Sigma| - \frac{1}{2}(\mathbf{f} - \boldsymbol{\mu})^T \Sigma^{-1}(\mathbf{f} - \boldsymbol{\mu}) - \frac{n}{2} log(2\pi) \tag{4}$$

B. Seal Wear

A dynamic seal prevents leakage when there is relative motion between two surfaces. The seal under consideration prevents leakage between the two chambers of the actuator. Modeling the failure of a seal can become complicated very quickly, as a number of factors influence seal failure, including, material characteristics, amount of seal compression, surface irregularities, seal size, fluid pressure, pressure pulses, temperature, fluid viscosity, fluid contamination, fluid/material compatibility, allowable leakage levels, and assembly and quality control procedures. The failure symptoms include excessive leakage and slow mechanical response. Many mechanisms and causes of these symptoms are described in [67].

In this dissertation, the wear mechanism is considered for a seal in a hydraulic actuator. Generally, seal leakage is due to wear caused by friction between the seal and piston, which removes seal material and allows fluid to pass between the chambers of the actuator. There are multiple wear mechanisms including adhesive wear, abrasive wear, surface fatigue, fretting wear, and erosive wear [145]. Lancaster [146] explains many of the complexities of abrasive wear while Briscoe and Sinha [147] and Briscoe [148] review wear of polymers. Due to the complexity of the mechanisms of wear, wear is typically modeled through the use of an experimentally determined wear rate.

Nikas [149] has written an extensive literature review on seal wear in actuators. The leakage area is the result of the removal of seal material — typically a polytetrafluoroethylene (PTFE) polymer — which is a function of load, distance traveled, material properties of the actuator and seal, geometry of the actuator, temperature, hydraulic fluid viscosity, and contaminants. Experimentally determined wear rates ($mm^3/m/N$) are available for PTFE composites used in hydraulic actuators e.g. Sawyer et al. [150] and Khedkar et al. [151].

REFERENCES

[1] J. Pearl, *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

[2] D. Heckerman, D. Geiger, and D. M. Chickering, "Learning Bayesian networks: The combination of knowledge and statistical data," *Mach. Learn.*, vol. 20, pp. 197–243, 1995.

[3] N. Friedman, K. Murphy, and S. Russell, "Learning the Structure of Dynamic Probabilistic Networks," in *UAI'98 Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, 1998, pp. 139–147.

[4] S. L. Lauritzen, "Propagation of Probabilities, Means, and Variances in Mixed Graphical Association Models," *J. Am. Stat. Assoc.*, vol. 87, no. 420, pp. 1098–1108, Dec. 1992.

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

[6] D. Heckerman and D. Geiger, "Learning Bayesian Networks: A unification for discrete and Gaussian domains," *Proc. Elev. Conf. Uncertain. Inarti Cial Intell.*, 1995.

[7] G. Bartram and Mahadevan, S., "Integration of Heterogeneous Information in SHM Models," *Struct. Control Heal. Monit.*, p. Accepted, 2013.

[8] A. Carpinteri, G. Lacidogna, and G. Niccolini, "Damage analysis of reinforced concrete buildings by the acoustic emission technique," *Struct. Control Heal. Monit.*, vol. 18, no. 6, pp. 660–673, Oct. 2011.

[9] J. A. Rice, K. A. Mechitov, S. H. Sim, B. F. Spencer Jr, and G. A. Agha, "Enabling framework for structural health monitoring using smart sensors," *Struct. Control Heal. Monit.*, vol. 18, no. 5, pp. 574–587, Aug. 2011.

[10] K. S. Yen and M. M. Ratnam, "2-D crack growth measurement using circular grating moiré fringes and pattern matching," *Struct. Control Heal. Monit.*, vol. 18, no. 4, pp. 404–415, Jun. 2011.

[11] B.-H. Koh, P. Dharap, S. Nagarajaiah, and M. Phan, "Real-Time Structural Damage Monitoring by Input Error Function," *Aiaa J.*, vol. 43, no. 8, pp. 1808–1814, Aug. 2005.

[12] J. M. Caicedo and J. Marulanda, "Fast mode identification technique for online monitoring," *Struct. Control Heal. Monit.*, vol. 18, no. 4, pp. 416–429, Jun. 2011.

[13] L. Console and P. Torasso, "A spectrum of logical definitions of model-based diagnosis," *Comput. Intell.*, vol. 7, no. 3, pp. 133–141, Aug. 1991.

[14] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "Comprehensive diagnosis of continuous systems using dynamic Bayes nets," in *Proceedings of the 19TH International Workshop on Principles of Diagnosis*, 2008, pp. 151–158.

[15] S. Jha, W. Li, and S. A. Seshia, "Localizing Transient Faults Using Dynamic Bayesian Networks."

[16] W. Buntine, "A guide to the literature on learning probabilistic networks from data," *Knowl. Data Eng. Ieee Trans.*, vol. 8, no. 2, pp. 195–210, 1996.

[17] O. Kipersztok, "Diagnosis Decision Support for Airplane Maintenance," in *Proceedings of the 5th WSES International Conference on Circuits, Systems, Communications and Computers*, Rethymno, Greece, 2001.

[18] V. U. B. Challagulla, F. B. Bastani, R. A. Paul, W.-T. Tsai, and Y. Chen, "A Machine Learning-Based Reliability Assessment Model for Critical Software Systems," in *Computer Software and Applications Conference, Annual International*, Los Alamitos, CA, USA, 2007, vol. 1, pp. 79–86.

[19] S. M. Leach, "Informed structural priors for bayesian networks: applications in molecular biology using heterogeneous data sources," Brown University, Providence, RI, USA, 2006.

[20] H. Langseth, T. D. Nielsen, and R. Dybowski, "Fusion of Domain Knowledge with Data for Structural Learning in Object Oriented Domains," 2003.
[21] M. Richardson and P. Domingos, "Learning with Knowledge from Multiple Experts," *Icml 20*, pp. 624–631, 2003.
[22] N. Cesa-Bianchi, D. R. Hardoon, and G. Leen, "Guest Editorial: Learning from multiple sources," *Mach. Learn.*, vol. 79, no. 1–2, pp. 1–3, Feb. 2010.
[23] S. Thrun, "Is Learning The n-th Thing Any Easier Than Learning The First?," in *Advances in Neural Information Processing Systems*, vol. 9, The MIT Press, 1995, pp. 640–646.
[24] R. Caruana, "Multitask Learning," *Mach. Learn.*, vol. 28, pp. 41–75, 1997.
[25] J. Baxter, "A Bayesian/information theoretic model of learning to learn via multiple task sampling," *Mach. Learn.*, vol. 28, pp. 7–39, 1997.
[26] R. Luis, L. E. Sucar, and E. F. Morales, "Inductive transfer for learning Bayesian networks," *Mach. Learn.*, vol. 79, no. 1–2, pp. 227–255, Dec. 2009.
[27] W. Dai, G. Xue, Q. Yang, and Y. Yu, "Transferring naive Bayes classifiers for text classification," in *Proceedings of the 22nd National Conference on Artificial Intelligence*, 2007, vol. 1, pp. 540–545.
[28] D. L. Silver, R. Poirier, and D. Currie, "Inductive transfer with context-sensitive neural networks," *Mach. Learn.*, vol. 73, no. 3, pp. 313–336, Oct. 2008.
[29] A. Niculescu-mizil and R. Caruana, "Inductive transfer for Bayesian network structure learning," in *Proceedings of the Eleventh International Conference on Artificial Intelligence and Statistics*, 2007.
[30] D. M. Roy and L. P. Kaelbling, "Efficient Bayesian Task-Level Transfer Learning," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, 2007, pp. 2599–2604.
[31] M. Szafranski, Y. Grandvalet, and A. Rakotomamonjy, "Composite kernel learning," *Mach. Learn.*, vol. 79, no. 1–2, pp. 73–103, Oct. 2009.
[32] P. Wu and T. G. Dietterich, "Improving SVM accuracy by training on auxiliary data sources," in *Proceedings of the Twenty-first International Conference on Machine Learning*, 2004.
[33] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. W. Vaughan, "A theory of learning from different domains," *Mach. Learn.*, vol. 79, no. 1–2, pp. 151–175, Oct. 2009.
[34] M.-R. Amini and C. Goutte, "A co-classification approach to learning from multilingual corpora," *Mach. Learn.*, vol. 79, no. 1–2, pp. 105–121, Oct. 2009.
[35] M. Dredze, A. Kulesza, and K. Crammer, "Multi-domain learning by confidence-weighted parameter combination," *Mach. Learn.*, vol. 79, no. 1–2, pp. 123–149, Oct. 2009.
[36] Y. Qu, Y. Zhang, and J. Wang, "A dynamic Bayesian network data fusion algorithm for estimating leaf area index using time-series data from in situ measurement to remote sensing observations," *Int. J. Remote Sens.*, vol. 33, no. 4, pp. 1106–1125, 2012.
[37] S. Das, D. Lawless, B. Ng, and A. Pfeffer, "Factored particle filtering for data fusion and situation assessment in urban environments," in *2005 8th International Conference on Information Fusion*, 2005, vol. 2, p. 8 pp.
[38] C. Cappelle, M. E. El Najjar, D. Pomorski, and F. Charpillet, "Multi-sensors data fusion using Dynamic Bayesian Network for robotised vehicle geo-localisation," in *2008 11th International Conference on Information Fusion*, 2008, pp. 1 –8.
[39] J. Chen and Q. Ji, "Online Spatial-temporal Data Fusion for Robust Adaptive Tracking," in *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, 2007, pp. 1 –8.
[40] U. Lerner, R. Parr, D. Koller, and G. Biswas, "Bayesian Fault Detection and Diagnosis in Dynamic Systems," *Proc Aaai*, pp. 531–537, 2000.
[41] I. Roychoudhury, G. Biswas, and X. Koutsoukos, "A Bayesian approach to efficient diagnosis of incipient faults," *Proc 17th Int Work. Princ. Diagn.*, pp. 243–250, 2006.

[42] K. W. Przytula and A. Choi, "An Implementation of Prognosis with Dynamic Bayesian Networks," in *2008 IEEE Aerospace Conference*, 2008, pp. 1–8.
[43] G. Arroyo-Figueroa and L. E. Sucar, "Temporal Bayesian Network of Events for Diagnosis and Prediction in Dynamic Domains," *Appl. Intell.*, vol. 23, no. 2, pp. 77–86, 2005.
[44] D. Straub, "Stochastic Modeling of Deterioration Processes through Dynamic Bayesian Networks," *J. Eng. Mech.*, vol. 135, no. 10, pp. 1089–1099, 2009.
[45] M. Dong and Z. Yang, "Dynamic Bayesian network based prognosis in machining processes," *J. Shanghai Jiaotong Univ. Sci.*, vol. 13, no. 3, pp. 318–322, 2008.
[46] D. Straub and A. Der Kiureghian, "Bayesian Network Enhanced with Structural Reliability Methods: Methodology," *J. Eng. Mech.*, vol. 136, no. 10, pp. 1248–1258, 2010.
[47] S. Sankararaman, Y. Ling, and S. Mahadevan, "Uncertainty quantification and model validation of fatigue crack growth prediction," *Eng. Fract. Mech.*, vol. 78, no. 7, pp. 1487–1504, May 2011.
[48] Y. Ling and S. Mahadevan, "Integration of structural health monitoring and fatigue damage prognosis," *Mech. Syst. Signal Process.*, vol. 28, pp. 89–104, Apr. 2012.
[49] J. L. Doob, "The Elementary Gaussian Processes," *Ann. Math. Stat.*, vol. 15, no. 3, pp. 229–282, 1944.
[50] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities.," *Proc. Natl. Acad. Sci. U. S. A.*, vol. 79, no. 8, pp. 2554–2558, Apr. 1982.
[51] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proc. Ieee*, vol. 77, no. 2, pp. 257–286, Feb. 1989.
[52] T. Dean and K. Kanazawa, "Probabilistic Temporal Reasoning," in *AAAI-88 Proceedings*, 1988.
[53] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, Jun. 1998.
[54] K. B. Korb, L. R. Hope, A. E. Nicholson, and K. Axnick, "Varieties of Causal Intervention," in *PRICAI 2004: Trends in Artificial Intelligence*, vol. 3157, C. Zhang, H. W. Guesgen, and W. K. Yeap, Eds. Auckland, New Zealand: Springer Berlin / Heidelberg, 2004, pp. 322–331.
[55] *RIAC Automated Databook*. Reliability Information Analysis Center, 2006.
[56] M. P. Kaminskiy and V. V. Krivtsov, "A simple procedure for Bayesian estimation of the Weibull distribution," *Ieee Trans. Reliab.*, vol. 54, no. 4, pp. 612– 616, Dec. 2005.
[57] C. Q. Li and R. E. Melchers, "Outcrossings from Convex Polyhedrons for Nonstationary Gaussian Processes," *J Engrg Mech J. Eng. Mech.*, vol. 119, no. 11, 1993.
[58] A. Haldar and S. Mahadevan, *Probability, reliability, and statistical methods in engineering design*. New York: John Wiley, 2000.
[59] S. Mahadevan and A. Dey, "Adaptive Monte Carlo Simulation for Time-Variant Reliability Analysis of Brittle Structures," *Aiaa J.*, vol. 35, no. 2, pp. 321–326, 1997.
[60] N. Kuschel and R. Rackwitz, "Optimal design under time-variant reliability constraints," *Struct. Saf.*, vol. 22, no. 2, pp. 113–127, Jun. 2000.
[61] D. Heckerman and D. Geiger, "Learning Bayesian Networks: A unification for discrete and Gaussian domains," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 274–284.
[62] G. F. Cooper and E. Herskovits, "A Bayesian method for the induction of probabilistic networks from data," *Mach. Learn.*, vol. 9, no. 4, pp. 309–347, Oct. 1992.
[63] N. Friedman, I. Nachman, and D. Peér, "Learning bayesian network structure from massive datasets: the «sparse candidate «algorithm," in *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, 1999, pp. 206–215.
[64] P. Spirtes, C. N. Glymour, and R. Scheines, *Causation, prediction, and search*. MIT Press, 2000.
[65] I. Tsamardinos, L. E. Brown, and C. F. Aliferis, "The max-min hill-climbing Bayesian network structure learning algorithm," *Mach. Learn.*, vol. 65, no. 1, pp. 31–78, Mar. 2006.

[66] D. M. Chickering, "A Transformational Characterization of Equivalent Bayesian Network Structures," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, 1995, pp. 87–98.
[67] "Handbook of Reliability Prediction Procedures for Mechanical Equipment," Naval Surface Warfare Center, West Bethesda, Maryland 20817-5700, May 2011.
[68] N. Sepeheri, M. Karpenko, L. An, and S. Karam, "A test rig for experimentation on fault tolerant control and condition monitoring algorithms in fluid power systems: from designing through implementation," *Trans. Can. Soc. Mech. Eng.*, vol. 29, no. 3, pp. 441–458, 2005.
[69] G. Zeiger and A. Akers, "Dynamic analysis of an axial piston pump swashplate control," *Arch. Proc. Inst. Mech. Eng. Part C Mech. Eng. Sci. 1983-1988 Vols 197-202*, vol. 200, no. 13, pp. 49–58, Jun. 1986.
[70] L. Zeliang, "Condition Monitoring of Axial Piston Pump," Master's Thesis, University of Saskatchewan, Saskatoon, Saskatchewan, Canada, 2005.
[71] B. T. Kulakowski, J. F. Gardner, and J. L. Shearer, *Dynamic modeling and control of engineering systems*. Cambridge University Press, 2007.
[72] D. F. Thompson, J. S. Pruyn, and A. Shukla, "Feedback design for robust tracking and robust stiffness in flight control actuators using a modified QFT technique," *Int. J. Control*, vol. 72, no. 16, pp. 1480 – 1497, 1999.
[73] V. Mahulkar, H. McGinnis, M. Derriso, and D. E. Adams, "Fault Identification in an Electro-Hydraulic Actuator and Experimental Validation of Prognosis Based Life Extending Control," DTIC Document, 2010.
[74] M. Karpenko and N. Sepehri, "Robust Position Control of an Electrohydraulic Actuator With a Faulty Actuator Piston Seal," *J. Dyn. Syst. Meas. Control*, vol. 125, no. 3, pp. 413–423, 2003.
[75] B. W. MacCormick, *Aerodynamics, Aeronautics, and Flight Mechanics*. John Wiley & Sons, Incorporated, 1995.
[76] M. S. Arulampalam, S. Maskell, and N. Gordon, "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Ieee Trans. Signal Process.*, vol. 50, pp. 174–188, 2002.
[77] D. Poole and A. Mackworth, *Artificial Intelligence: Foundations of Computational Agents*. Cambridge University Press, 2010.
[78] D. T. Dupre, "Abductive and Consistency-Based Diagnosis Revisited: a Modeling Perspective," *Proc 8th Int. Work. Non-Monotonic Reason. Nmr2000*, 2000.
[79] R. Isermann, "Supervision, fault-detection and fault-diagnosis methods -- An introduction," *Control Eng. Pr.*, vol. 5, no. 5, pp. 639–652, May 1997.
[80] W. G. Fenton, T. M. McGinnity, and L. P. Maguire, "Fault diagnosis of electronic systems using intelligent techniques: a review," *Ieee Trans. Syst. Man Cybern. Part C Appl. Rev.*, vol. 31, no. 3, pp. 269–281, Aug. 2001.
[81] R. Reiter, "A theory of diagnosis from first principles," *Artif. Intell.*, vol. 32, no. 1, pp. 57–95, Apr. 1987.
[82] D. Poole, "Normality and Faults in Logic-Based Diagnosis," 1985.
[83] R. Davis and W. C. Hamscher, "Model-Based Reasoning: Troubleshooting," Massachusetts Institute of Technology, Artificial Intelligence Laboratory, A. I. Memo No. 159, Jul. 1988.
[84] J. de Kleer and B. C. Williams, "Diagnosis with behavioral modes," in *Proceedings of the 11th International Joint Conference on Artificial Intelligence*, Detroit, MI, 1989, pp. 1324–1330.
[85] G. Friedrich, G. Gottlob, and W. Nejdl, "Physical impossibility instead of fault models," in *Proceedings of the eighth National conference on Artificial intelligence - Volume 1*, 1990, pp. 331–336.
[86] R. Isermann, "Model-based fault-detection and diagnosis - status and applications," *Annu. Rev. Control*, vol. 29, no. 1, pp. 71–85, 2005.

[87] P. J. Mosterman and G. Biswas, "Diagnosis of continuous valued systems in transient operating regions," *Ieee Trans. Syst. Man Cybern. Part Syst. Humans*, vol. 29, no. 6, pp. 554–565, Nov. 1999.
[88] A. Moustafa, S. Mahadevan, M. Daigle, and G. Biswas, "Structural and sensor damage identification using the bond graph approach," *Struct. Control Heal. Monit.*, vol. 17, no. 2, pp. 178–197, Mar. 2010.
[89] N. Tandon and A. Choudhury, "A review of vibration and acoustic measurement methods for the detection of defects in rolling element bearings," *Tribol. Int.*, vol. 32, no. 8, pp. 469–480, Aug. 1999.
[90] Y. Ding, D. Ceglarek, and J. Shi, "Fault Diagnosis of Multistage Manufacturing Processes by Using State Space Approach," *J. Manuf. Sci. Eng.*, vol. 124, no. 2, pp. 313–322, May 2002.
[91] K. R. McNaught and A. Zagorecki, "Using dynamic Bayesian networks for prognostic modelling to inform maintenance decision making," in *IEEE International Conference on Industrial Engineering and Engineering Management, 2009. IEEM 2009*, 2009, pp. 1155–1159.
[92] S. Jha, Wenchao Li, and S. A. Seshia, "Localizing transient faults using dynamic bayesian networks," in *High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International*, 2009, pp. 82–87.
[93] G. Arroyo-Figueroa and L. Sucar, "A Temporal Bayesian Network for Diagnosis and Prediction," in *Proceedings of the Proceedings of the Fifteenth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-99)*, San Francisco, CA, 1999, pp. 13–20.
[94] F. Camci and R. B. Chinnam, "Dynamic Bayesian networks for machine diagnostics: hierarchical hidden Markov models vs. competitive learning," in *2005 IEEE International Joint Conference on Neural Networks, 2005. IJCNN '05. Proceedings*, 2005, vol. 3, pp. 1752– 1757 vol. 3.
[95] M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, and S. Singh, "Exploiting heterogeneity in sensor networks," in *Proceedings IEEE INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, 2005, vol. 2, pp. 878–890 vol. 2.
[96] T. Kijewski-Correa, M. Haenggi, and P. Antsaklis, "Wireless sensor networks for structural health monitoring: a multi-scale approach," in *ASCE Structures 2006 Congress*, 2006.
[97] E. Kulcu, X. Qin, J. Barrish, and A. E. Aktan, "Information technology and data management issues for health monitoring of the Commodore Barry Bridge," pp. 98–111, Jun. 2000.
[98] O. Sidek, S. Kabir, and S. Quadri, "Multi-sensor Data Fusion System for Enhanced Analysis of Deterioration in Concrete Structures," in *Progress in electromagnetics research symposium proceedings*, 2011.
[99] G. J. Kacprzynski, "Sensor/Model Fusion for Adaptive Prognosis of Structural Corrosion Damage," 2006.
[100] J. C. Viana, P. J. Antunes, R. J. Guimara, N. J. Ferreira, M. A. Baptista, G. R. Dias, and C. Materials, "Combining experimental and computed data for effective SHM of critical structural components," in *2011 IEEE Aerospace Conference*, 2011, pp. 1–10.
[101] M. M. R. Taha and J. L. Lucero, "A Generic Fuzzy Metric for Damage Recognition in Structural Health Monitoring Systems," in *2005 IEEE International Conference on Systems, Man and Cybernetics*, 2005, vol. 2, pp. 1518–1523.
[102] H. M. Blalock, "Causal inferences, closed populations, and measures of association," *Am. Polit. Sci. Rev.*, vol. 61, no. 1, pp. 130–136, 1967.
[103] L. L. Hargens, "A note on standardized coefficients as structural parameters," *Sociol. Methods Res.*, vol. 5, no. 2, pp. 247–256, 1976.
[104] J.-O. Kim and C. W. Mueller, "Standardized and unstandardized coefficients in causal analysis An expository note," *Sociol. Methods Res.*, vol. 4, no. 4, pp. 423–438, 1976.
[105] A. Saltelli, K. Chan, and E. M. Scott, *Sensitivity Analysis*. Wiley, 2009.

[106] X. Boyen and D. Koller, “Tractable inference for complex stochastic processes,” in *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, San Francisco, CA, USA, 1998, pp. 33–42.
[107] B. Ristic and S. Arulampalam, *Beyond the Kalman filter : particle filters for tracking applications*. Boston, MA: Artech House, 2004.
[108] M. K. Pitt and N. Shephard, “Filtering via Simulation: Auxiliary Particle Filters,” *J. Am. Stat. Assoc.*, vol. 94, no. 446, pp. 590–599, 1999.
[109] C. Andrieu, M. Davy, and A. Doucet, “Efficient particle filtering for jump Markov systems. Application to time-varying autoregressions,” *Ieee Trans. Signal Process.*, vol. 51, no. 7, pp. 1762–1770, 2003.
[110] S. Sankararaman and S. Mahadevan, “Bayesian methodology for diagnosis uncertainty quantification and health monitoring,” *Struct. Control Heal. Monit.*, vol. 20, no. 1, pp. 88–106, 2013.
[111] A. Genz and K.-S. Kwong, “Numerical evaluation of singular multivariate normal distributions,” *J. Stat. Comput. Simul.*, vol. 68, no. 1, pp. 1–21, 2000.
[112] P. C. Paris, M. P. Gomez, and W. E. Anderson, “A Rational Analytic Theory of Fatigue,” *Trend Eng.*, vol. 13, pp. 9–14, 1961.
[113] Y. Liu and S. Mahadevan, “Probabilistic fatigue life prediction using an equivalent initial flaw size distribution,” *Int. J. Fatigue*, vol. 31, no. 3, pp. 476–487, Mar. 2009.
[114] I. Nabney and C. Bishop, “Netlab neural network software,” *Matlab Toolbox*, vol. 71, 2003.
[115] A. K. S. Jardine, D. Lin, and D. Banjevic, “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006.
[116] D. Banjevic and A. K. S. Jardine, “Calculation of reliability function and remaining useful life for a Markov failure time process,” *Ima J. Manag. Math.*, vol. 17, no. 2, pp. 115–130, Apr. 2006.
[117] K. B. Goode, J. Moore, and B. J. Roylance, “Plant machinery working life prediction method utilizing reliability and condition-monitoring data,” *Proc. Inst. Mech. Eng. Part E J. Process Mech. Eng.*, vol. 214, no. 2, pp. 109–122, May 2000.
[118] J. Yan, M. Koç, and J. Lee, “A prognostic algorithm for machine performance assessment and its application,” *Prod. Plan. Control*, vol. 15, no. 8, pp. 796–801, 2004.
[119] P.-J. Vlok, M. Wnek, and M. Zygmunt, “Utilising statistical residual life estimates of bearings to quantify the influence of preventive maintenance actions,” *Mech. Syst. Signal Process.*, vol. 18, no. 4, pp. 833–847, Jul. 2004.
[120] D. Lin and V. Makis, “Filters and parameter estimation for a partially observable system subject to random failure with continuous-range observations,” *Adv. Appl. Probab.*, vol. 36, no. 4, pp. 1212–1230, Dec. 2004.
[121] W. Wang, P. A. Scarf, and M. a. J. Smith, “On the application of a model of condition-based maintenance,” *J. Oper. Res. Soc.*, vol. 51, no. 11, pp. 1218–1227, Oct. 2000.
[122] W. Wang, “A model to predict the residual life of rolling element bearings given monitored condition information to date,” *Ima J. Manag. Math.*, vol. 13, no. 1, pp. 3–16, Jan. 2002.
[123] C. R. Farrar and K. Worden, *Structural Health Monitoring: A Machine Learning Perspective*. John Wiley & Sons, 2012.
[124] M. E. Tipping, “Sparse Bayesian learning and the relevance vector machine,” *J. Mach. Learn. Res.*, vol. 1, pp. 211–244, 2001.
[125] S. Zhang and R. Ganesan, “Multivariable Trend Analysis Using Neural Networks for Intelligent Diagnostics of Rotating Machinery,” *J. Eng. Gas Turbines Power*, vol. 119, no. 2, pp. 378–384, Apr. 1997.
[126] G. Vachtsevanos and P. Wang, “Fault prognosis using dynamic wavelet neural networks,” in *AUTOTESTCON Proceedings, 2001. IEEE Systems Readiness Technology Conference*, 2001, pp. 857–870.

[127] R. C. M. Yam, P. W. Tse, L. Li, and P. Tu, "Intelligent Predictive Decision Support System for Condition-Based Maintenance," *Int. J. Adv. Manuf. Technol.*, vol. 17, no. 5, pp. 383–391, Feb. 2001.
[128] W. Q. Wang, M. F. Golnaraghi, and F. Ismail, "Prognosis of machine health condition using neuro-fuzzy systems," *Mech. Syst. Signal Process.*, vol. 18, no. 4, pp. 813–831, Jul. 2004.
[129] R. B. Chinnam and P. Baruah, "Autonomous diagnostics and prognostics through competitive learning driven HMM-based clustering," in *Proceedings of the International Joint Conference on Neural Networks, 2003*, 2003, vol. 4, pp. 2466–2471 vol.4.
[130] C. Kwan, X. Zhang, R. Xu, and L. Haynes, "A novel approach to fault diagnostics and prognostics," in *IEEE International Conference on Robotics and Automation, 2003. Proceedings. ICRA '03*, 2003, vol. 1, pp. 604–609 vol.1.
[131] J. Liu, A. Saxena, K. Goebel, B. Saha, and W. Wang, "An Adaptive Recurrent Neural Network for Remaining Useful Life Prediction of Lithium-ion Batteries," DTIC Document, 2010.
[132] K. Goebel, B. Saha, A. Saxena, N. Mct, and N. Riacs, "A comparison of three data-driven techniques for prognostics," in *62nd Meeting of the Society For Machinery Failure Prevention Technology (MFPT)*, 2008, pp. 119–131.
[133] N. Z. Gebraeel and M. A. Lawley, "A neural network degradation model for computing and updating residual life distributions," *Autom. Sci. Eng. Ieee Trans.*, vol. 5, no. 1, pp. 154–163, 2008.
[134] B. Saha, K. Goebel, and J. Christophersen, "Comparison of prognostic algorithms for estimating remaining useful life of batteries," *Trans. Inst. Meas. Control*, vol. 31, no. 3–4, pp. 293–308, 2009.
[135] G. J. Kacprzynski, A. Sarlashkar, M. J. Roemer, A. Hess, and B. Hardman, "Predicting remaining life by fusing the physics of failure modeling with diagnostics," *JOM*, vol. 56, no. 3, pp. 29–35, Mar. 2004.
[136] M. E. Orchard and G. J. Vachtsevanos, "A particle-filtering approach for on-line fault diagnosis and failure prognosis," *Trans. Inst. Meas. Control*, Jun. 2009.
[137] A. Lorton, M. Fouladirad, and A. Grall, "A methodology for probabilistic model-based prognosis," *Eur. J. Oper. Res.*, vol. 225, no. 3, pp. 443–454, Mar. 2013.
[138] B. Saha, J. R. Celaya, P. F. Wysocki, and K. F. Goebel, "Towards prognostics for electronics components," in *2009 IEEE Aerospace conference*, 2009, pp. 1–7.
[139] T. Khan, L. Udpa, and S. Udpa, "Particle filter based prognosis study for predicting remaining useful life of steam generator tubing," in *Prognostics and Health Management (PHM), 2011 IEEE Conference on*, 2011, pp. 1 –6.
[140] J. D. Kozlowski, "Electrochemical cell prognostics using online impedance measurements and model-based data fusion techniques," in *2003 IEEE Aerospace Conference, 2003. Proceedings*, 2003, vol. 7, pp. 3257–3270.
[141] Z. Jinlin and Z. Zhengdao, "Fault prognosis for data incomplete systems: A dynamic Bayesian network approach," in *Control and Decision Conference (CCDC), 2012 24th Chinese*, 2012, pp. 2244–2249.
[142] D. A. Tobon-Mejia, K. Medjaher, N. Zerhouni, and G. Tripot, "A Data-Driven Failure Prognostics Method Based on Mixture of Gaussians Hidden Markov Models," *Reliab. Ieee Trans.*, vol. 61, no. 2, pp. 491–503, 2012.
[143] A. Saxena, J. Celaya, B. Saha, S. Saha, and K. Goebel, "Metrics for Offline Evaluation of Prognostic Performance," *Int. J. Progn. Heal. Manag.*, no. 1, 2010.
[144] C. E. Rasmussen, "Gaussian processes for machine learning," 2006.
[145] M. H. Jones, Ed., *Industrial Tribology: The Practical Aspects of Friction, Lubrication and Wear*. North Holland, 1983.
[146] J. K. Lancaster, "Abrasive wear of polymers," *Wear*, vol. 14, no. 4, pp. 223–239, Oct. 1969.
[147] B. J. Briscoe and S. K. Sinha, "Wear of polymers," *Proc. Inst. Mech. Eng. Part J J. Eng. Tribol.*, vol. 216, no. 6, pp. 401–413, Jun. 2002.

[148] B. Briscoe, “Wear of polymers: an essay on fundamental aspects,” *Tribol. Int.*, vol. 14, no. 4, pp. 231–243, Aug. 1981.
[149] G. K. Nikas, “Eighty years of research on hydraulic reciprocating seals: Review of tribological studies and related topics since the 1930s,” *Proc. Inst. Mech. Eng. Part J J. Eng. Tribol.*, vol. 224, no. 1, pp. 1–23, Jan. 2010.
[150] W. G. Sawyer, K. D. Freudenberg, P. Bhimaraj, and L. S. Schadler, “A study on the friction and wear behavior of PTFE filled with alumina nanoparticles,” *Wear*, vol. 254, no. 5–6, pp. 573–580, Mar. 2003.
[151] J. Khedkar, I. Negulescu, and E. I. Meletis, “Sliding wear behavior of PTFE composites,” *Wear*, vol. 252, no. 5–6, pp. 361–369, Mar. 2002.