Prognostics and Health Monitoring in the Presence of Heterogeneous Information

Gregory Bartram[1], Sankaran Mahadevan[2]

[1,2]*Vanderbilt University, Nashville, TN, 37212, USA*

gregory.w.bartram@vanderbilt.edu
sankaran.mahadevan@vanderbilt.edu

ABSTRACT

Diagnosis and prognosis methodologies have used dynamic Bayesian networks (DBNs) to fuse many types of information. These methodologies, however, fuse problem-specific information and focus only on a subset of information types. By using only a subset of information, the interactions between or individual behaviors of subsystems, components, and faults may not be fully realized. In this paper, a general framework for system level diagnosis and prognosis of a mechanical system in the presence of heterogeneous information using dynamic Bayesian network (DBN) is proposed. Due to their ability to fuse heterogeneous information — information in a variety of formats from various sources — and give a probabilistic representation of a system, DBNs provide a platform naturally suited for diagnosis and uncertainty quantification. In the proposed methodology, a DBN is first constructed via an established machine learning algorithm from heterogeneous information. The DBN is then used to track the system and detect faults by monitoring the Bayes' factor of the system state estimate. When a fault occurs, the underlying system model changes, and the Bayes' factor of the DBN system model decreases. The state estimate provided by tracking indicates the most likely fault scenario and quantifies the diagnosis uncertainty. Estimation of remaining useful life and quantification of uncertainty in prognosis can then proceed based upon the diagnosis results. The proposed methodology is then demonstrated using a cantilever beam example with a possible loose bolt at the connection or a crack in the middle of the span.

1. INTRODUCTION

Diagnosis and prognosis have become increasingly important in the quest for safer, more intelligent, more efficient, and more cost effective systems. Accurate diagnosis and prognosis are integral parts of system maintenance, operation, and design. Condition-based maintenance using automated diagnosis has gained traction in the belief that such a program can improve the safety and minimize the maintenance costs of a system. Mission-level decision making can also benefit from improved diagnosis and prognosis capabilities. Systems designed for prognosis and health management (PHM) also stand to benefit from improved diagnosis and prognosis.

One important requirement for a PHM methodology is the ability to utilize existing information about a system. Such information may be found in the form of expert opinion, operational and laboratory data, reliability data, and mathematical models. This information needs to be integrated into a system-level approach to better understand interactions between subsystems and components and make all of the information available for system-level diagnosis and prognosis procedures.

In previous work, static Bayesian networks (BNs) have commonly been used for diagnosis, particularly in medicine (e.g. Shwe et al. (Shwe et al., 1991)). Sahin et al. (Sahin, Yavuz, Arnavut, & Uluyol, 2007) learned a static BN of an aircraft engine using particle swarm optimization and used the learned BN for diagnosis. Examples of DBNs in diagnosis and prognosis are Jha et al. (Jha, Wenchao Li, & Seshia, 2009) who learn a DBN and use it to find transient faults, Przytula and Choi (Przytula & Choi, 2008) who use DBNs to obtain discrete qualitative measures of prognosis, McNaught and Zagorecki (McNaught & Zagorecki, 2009) who use a DBN of discrete variables to aid maintenance decisions, and Camci and Chinnam (Camci & Chinnam, 2005) who use hierarchical hidden Markov models for diagnosis and prognosis of machine parts. Straub (Straub, 2009) integrates model parameters into a DBN for modeling crack growth However, existing DBN-based diagnosis and prognosis methodologies have focused on problems with a limited universe of information. In mechanical applications, many diverse sources of information must be integrated into the DBN.

The primary contribution of this paper is a methodology for performing diagnosis and prognosis of a mechanical system in the presence of uncertainty and in the presence of heterogeneous information. Heterogeneous information consists of information from various sources in varying formats such as expert opinion, laboratory and operational data, existing reliability data, and mathematical models. The methodology uses dynamic Bayesian networks (DBNs) for system modeling.

The remainder of the paper is organized as follows. First, DBNs and the process of building a DBN model from heterogeneous information are briefly explained. Model-based diagnosis using the DBN model and uncertainty quantification in diagnosis are then discussed. The proposed methodology is demonstrated for a cantilever beam.

2. Proposed Methodology

In this paper a methodology for performing diagnosis and prognosis of a mechanical system in the presence of heterogeneous information is proposed. First, a DBN model of the system is constructed. This model is used in conjunction with a particle filter to track the system. Binary faults are detected based Bayes' factors and isolated using the state estimate of the system. The future trajectory of the system is then estimated via particle filtering

2.1. Bayesian Networks & Dynamic Bayesian Networks

A static BN, also referred to as a belief network and directed acyclic graph (DAG), 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. Unconnected nodes are conditionally independent of each other. The acyclic requirement means that no paths exist in the graph where, starting at node X_i, it is possible to return to node X_i.

A BN is a representation of the joint probability space U containing variables $X_1, \ldots, X_n$. The joint probability distribution is represented in factored form as

$$p(X_1, \ldots, X_n) = \prod_{i=1}^{n} p(X_i | \Pi_i) \tag{1}$$

where Π_i is the set of nodes on which X_i is conditionally dependent, so $p(X_i | X_1, \ldots, X_{i-1}) = p(X_i | \Pi_i)$. The nodes in Π_i are commonly referred to as the parents of node i. Note that this definition of a BN depends on the variable ordering.

The factored formulation in Eq. (1) is readily extended to handle different types of multivariate continuous distributions such as the Gaussian as well as distributions consisting of Gaussian and discrete variables.

In a DBN, the variable $\mathbf{X}[t]$ is the value of $\mathbf{X}$ at time t. The probability distribution describing $\mathbf{X}$ on the interval $[0, \infty)$ is very complex, as it is over $\bigcup_{t=0}^{t=\infty} \mathbf{X}[t]$. Using the Markov assumption simplifies this distribution by assuming that only the present state of the variable $\mathbf{X}[t]$ is necessary to estimate $\mathbf{X}[t+1]$ and thus $p(\mathbf{X}[t+1] | \mathbf{X}[0] \ldots \mathbf{X}[t]) = p(\mathbf{X}[t+1] | \mathbf{X}[t])$. Additionally, the process is assumed to be stationary, meaning that $p(\mathbf{X}[t+1] | \mathbf{X}[t])$ is independent of t. This approach to modeling DBNs is developed by Friedman et al. (Friedman, Murphy, & Russell, 1998).

A DBN may be composed of all discrete variables, all continuous variables, or hybrid set of discrete and continuous variables. For modeling systems with faults, it is advantageous to consider a hybrid system, typically with the continuous variables being modeled as Gaussian and the faults being discrete. Theory for such networks is developed in Heckerman and Geiger (Heckerman & Geiger, 1995) and Lauritzen (Lauritzen, 1992).

2.2. Heterogeneous Information

An important feature of Bayesian analysis is the ability to encode 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. Existing research has not fully exploited heterogeneous information, especially with respect to building DBN models for diagnosis and prognosis. The following sections summarize heterogeneous information and its integration into a DBN model.

2.2.1. Expert Opinion

The first information type used in BN and DBN learning is expert opinion. This information is typically elicited from a domain expert who can provide assumptions about the system variables, such as what variables to be modeled and what distributions to model them with. An expert may even be able to provide a network structure and/or distribution parameters.

2.2.2. 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. Each case of operational data is a set of values for each variable. A set of cases forms a database. This data is included in the network via Bayesian network structure and parameter learning algorithms.

2.2.3. 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 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. This data is included in the network via Bayesian network structure and parameter learning algorithms.

2.2.4. Published Reliability Data

In some instances, reliability data may be available for certain system components. Reliability data is used to establish prior distributions for specific system faults or update distribution parameters of faults in a DBN as time passes. 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.

2.2.5. Mathematical Behavior Models

Mathematical models may contain useful information for improving the system model. Mathematical models may either be physics-based (e.g. finite element or bond graph) or data-based/empirical (neural network, BN/DBN). They can be used to obtain reliability data, calculate values for immeasurable system variables (e.g. stress intensity factor or structural damping), or otherwise make modifications to the network.

An example mathematical model is the two-term exponential model,

$$y(x) = Ae^{Bx} + Ce^{Dx} \tag{2}$$

where A, B, C, D are model parameters to be estimated from data. To integrate this model into a DBN, the parameters for this model must be included as variables in the DBN. Through simulation, it has been determined that this model provides a suitable fit for modeling the length of a fatigue crack as it grows with the number of load cycles or the deflection of a cantilever beam with a growing crack.

2.2.6. Summary of Heterogeneous Information Integration

The information types discussed are used in concert with a DBN learning procedure to create a system model. Prior domain knowledge is the starting point for determining basic assumptions about the network, such as whether to use all discrete variables, all Gaussian variables, or a hybrid network. Any structural rules (e.g. banning certain nodes from being parent nodes) for the network need to be defined at this point. Once the basic assumptions are in place, operational and laboratory data and calculated data from mathematical models may then be used to determine the best structure for the DBN and the distribution parameters of the network. Reliability data and mathematical reliability models can then be used to augment probability distributions for some faults.

2.3. Learning Bayesian Networks & Dynamic Bayesian Networks

Learning BNs and DBNs consists of learning the structure of a network which defines the conditional independence relationships between variables and learning the distribution parameters of the network. Algorithms have been developed that can learn the structure and parameters for networks composed of discrete and continuous variables as well as hidden variables and missing data.

2.3.1. Learning a Static BN

Initially, it is assumed that all the variables are independent with unknown distributions. After gathering all available expert opinion about the variables (distribution types, constraints on the network structure), the structure learning algorithm evaluates any laboratory and experimental data available to determine the system structure of the system BN. The laboratory and experimental data along with any reliability data or mathematical models are used to determine the distribution parameters of the network.

One of the first algorithms for learning a BN is the K2 algorithm by Cooper and Herskovits (Cooper & Herskovits, 1992). The algorithm requires placing the variables in a fixed order. The variable that is first in the order can have no parents and subsequent variables in the ordering may have the preceding nodes as parents. The algorithm analyzes each node, attempting to determine which set of parents maximizes some scoring metric for the local structure (node + parents). The scoring metric computes the probability of observing the data given that local structure (essentially, a Bayes' factor). Once each node has been evaluated, the collection of local structures is combined into a global structure for the network. Other algorithms are available such as conditional independence testing (de Campos, 2006) or expectation-maximization (EM) methods (Friedman, 1998). EM methods can handle situations where data is missing or variables are unobservable.

2.3.2. Learning a Dynamic BN

DBN learning generally consists of learning the structure of a transition network which defines the relationships between variables in different time slices and learning the parameters of this network. Variations of several algorithms have been developed that can handle various combinations of discrete and continuous variables. The basic methodology for DBN structure learning followed in this paper is that given by Friedman et al. (Friedman et al., 1998).

Structure learning in a DBN is similar to that in a static BN. The observations for a dynamic system may include multiple time histories of instantiations of the system. By lagging each time history individually, data for a variable at t and $t + 1$ are obtained. After each time history is lagged, all the time histories can be combined into a database. In this database, a data case represents simultaneous observations of all the variables at a discrete point in time. Note that for the purpose of learning, it is here assumed that all the variables are observable. In general this may not be true.

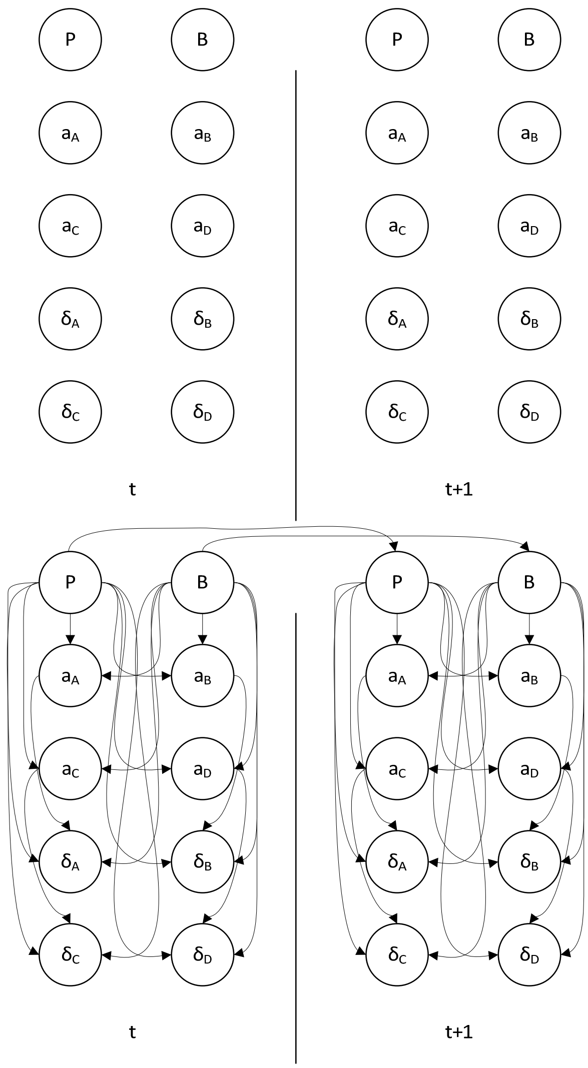

Figure 1. DBN before and after learning.

Consider the cantilever beam model. Time series of measurements are collected for each beam. In this case, it is desirable not only to seed the faults (i.e. loosen the bolt manually), but also to observe healthy beams over time as they are loaded and then begin to form cracks or have a bolt gradually loosen. Say evaluating 100 beams results in 100 time series of data. Each time series of data is lagged individually so as to create e.g. B(t) and B(t+1). Instead of 5 variables per beam, there are now 10. Then, the data for 100 sets of 10 variables are combined into a database for learning the DBN structure. Fig. 1 shows the initial assumption of the variables being independent and the structure after learning.

2.4. Diagnosis

Diagnosis is the process of detecting and isolating faults in a system and quantifying their magnitudes. Detection and isolation can be performed using a BN or DBN model of the system which treats faults as binary (true or false).

In detection, it is desired to determine the state of the system, including the states of potentially unobservable variables (faults). First, the case of a static system is considered followed by the dynamic case.

2.4.1. Diagnosis of a Static System

Diagnosis of a static system using a BN is a relatively straightforward problem. The purpose of diagnosis is to determine values of the unknown parameters (hidden state) from the measurements. Given observations of observable variables, the state of unobservable variables, which are hidden, can be inferred using Bayes' Theorem. The state of the fault values that best explains the observations is taken as the fault state. Inference is automated using algorithms such as the junction tree algorithm for exact inference or Monte Carlo-based approaches for approximate inference. Static BN diagnosis has been implemented in e.g. medical diagnosis.

2.4.2. Diagnosis of a Dynamic System

Diagnosis of a dynamic system is built around the concept of tracking. First, the tracking problem is explained as well as a Monte Carlo solution to the problem. Then, it is explained how the results of tracking can be used to detect and isolate faults in the system.

2.4.2.1 Filtering

Diagnosis of a dynamic system reduces to the problem of filtering the state of the system over time, as opposed to inferring the state at once, as for a static system. Filtering combines theoretical system state estimates with noisy

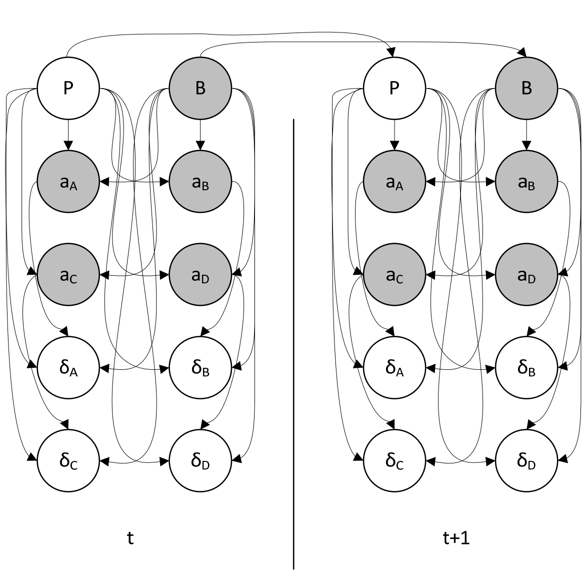

Figure 2. Cantilever beam DBN in filtering.
Gray nodes are unobservable.

measurements to estimate the true state of the system. The state estimate includes estimates of the states of faults, whose states are otherwise unknown, which is why tracking can be used for diagnosis. The general tracking problem, explained by Arulampalam et al. (Arulampalam, Maskell, & Gordon, 2002) takes the form of a recursive Bayesian filter, where the prior pdf of the current state estimate, having been constructed from previous state estimates, is updated using a current measurement. This is also known as a predictor-corrector algorithm. The state estimate is often called the belief state due to the Bayesian nature of the estimation.

More formally, consider the k[th] state estimate $\mathbf{x}_k$ given by the state model $\mathbf{f}_{k-1}(\mathbf{x}_{k-1}, \mathbf{v}_{k-1})$, where $\mathbf{x}_{k-1}$ is the previous state estimate and $\mathbf{v}_{k-1}$ is an i.i.d process noise sequence. It is desired to determine $\mathbf{x}_k$ using the measurements z_k given by the measurement model $\mathbf{h}_k(\mathbf{x}_k, \mathbf{n}_k)$ where $\mathbf{n}_k$ is an i.i.d measurement noise sequence. In a Bayesian sense, this translates to finding the belief state, the pdf $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$, where $\mathbf{Z}_{1:k}$ are the measurements from 1:*k*. The prior pdf $p(\mathbf{x}_0|\mathbf{z}_0) \equiv p(\mathbf{x}_0)$ since $\mathbf{z}_0$ is a set of no measurements.

If $p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})$ is available, the pdf $p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$ may be estimated by the Chapman-Kolmogorov equation

$$p(\mathbf{x}_k|\mathbf{Z}_{1:k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})d\mathbf{x}_{k-1} \tag{3}$$

where $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ is the probabilistic model defined by $\mathbf{f}_{k-1}$ using the known statistics of the process noise $\mathbf{v}_{k-1}$. Eq. (1) makes use of the Markov property by having $p(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{Z}_{1:k-1}) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$ The estimation of the prediction pdf $p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})$ is known as the prediction stage.

Once the measurement $\mathbf{z}_k$ is available, $p(\mathbf{x}_k|\mathbf{Z}_{1:k})$ is determined in the update stage using Bayes' rule as

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_{1:k}) &= p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{1:k-1}) \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{1:k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{1:k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_{k-1}|\mathbf{Z}_{1:k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{1:k-1})} \end{aligned} \tag{4}$$

where $p(\mathbf{z}_k|\mathbf{Z}_{1:k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{1:k-1})d\mathbf{x}_k$ is a normalizing constant. The likelihood function $p(\mathbf{z}_k|\mathbf{x}_k)$ is the probabilistic measurement model defined by $\mathbf{h}_k$ and the known statistics of the measurement noise $\mathbf{n}_k$. The optimal state estimated may then be computed as the maximum a posteriori (MAP) estimate, $argmax_{\mathbf{x}_k} p(\mathbf{x}_k|\mathbf{Z}_{1:k})$, or the minimum mean-square error (MMSE), $E\{\mathbf{x}_k|\mathbf{Z}_{1:k}\} = \int \mathbf{x}_k \cdot p(\mathbf{x}_k|\mathbf{Z}_{1:k})d\mathbf{x}_k$ (Ristic & Arulampalam, 2004).

At first glance it appears that the structure of a DBN might allow for easy tracking of the belief state, perhaps similar to a static BN, and providing system-level diagnosis with minimal effort. Unfortunately, even with a structured process, the belief state quickly becomes very complex after only a couple of updates, except in degenerate cases, as over time the system becomes highly correlated (Boyen & Koller, 1998). Only specific implementations of the recursive filter such as the Kalman filter or grid-based methods provide analytical solutions. Even more, the computational expense of exact inference algorithms is exponential in the number of nodes for a discrete DBN and even worse for hybrid or continuous distribution DBNs. Exact inference in the conditional Gaussian distributions used in this paper is NP-hard (Lerner, 2001). Thus, it is necessary to pursue approximate inference algorithms. Particle filtering is one technique that makes the tracking problem tractable.

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. (Ristic & Arulampalam, 2004).

A summary of the SIS algorithm for one time step is as follows. A previous (or initial if k = 1) set of N weights w_{k-1}^i and N corresponding particles $\mathbf{x}_{k-1}^i$ are given initially or known from the previous time step, where i denotes the i^{th} particle. N samples are drawn from the importance distribution, $q\left(\mathbf{x}_k^i \middle| \mathbf{x}_{k-1}^i, \mathbf{z}_k\right)$, where $\mathbf{z}_k$ are the measurements at the k^{th} time step. Unnormalized weights $\widetilde{w}_k^i$ are then computed up to a normalizing constant using a ratio of the distribution of the state estimate to the distribution of the importance sampling density. The weights $\widetilde{w}_k^i$ are then normalized so that their sum is equal to 1. The normalized weights w_k^i and points $\mathbf{x}_k^i$ form an approximation to the state estimate $p(\mathbf{x}_k|\mathbf{z}_k)$. Two common choices of the importance density $q\left(\mathbf{x}_k^i \middle| \mathbf{x}_{k-1}^i, \mathbf{z}_k\right)$ are the prior $p\left(\mathbf{x}_k^i \middle| \mathbf{x}_{k-1}^i\right)$ and the likelihood $p\left(\mathbf{z}_k \middle| \mathbf{x}_k^i\right)$.

Many variations of the SIS PF exist using different priors, resampling techniques, etc. For the conditional Gaussian system model employed in this paper, also called a jump Markov model, hybrid state model, or switching system, a dynamic multiple model (MM) filter is appropriate for tracking. The multiple model particle filter (MMPF) given in Ristic et al. (Ristic & Arulampalam, 2004) implements this filter. This particle filter is a generalization of a standard particle filter and is equivalent to a standard particle filter when there is only one underlying discrete state. Each discrete state in $\mathrm{r}_k \in \mathrm{R}_k$ represents an operating mode. Before doing SIS for the continuous components of the state, an operating mode is selected based on the probabilities of those operating modes occurring.

Considering the DBN dynamic beam model in Fig. 5, the state of the unobservable (gray) nodes at t + 1 is first predicted based on the state of the unobservable nodes at t. Next, the measurements $\mathrm{P} = p_{t+1}$ and $\delta = \delta_{t+1}$ are used to update (correct) the predictions and obtain the state estimate of the unobservable nodes. The process is implemented using a MMPF, which allows tracking of the discrete fault B as well as the continuous variables crack length, a, and stiffness, K.

2.4.2.2 Fault Detection and Isolation

Given a DBN which can track the system, fault detection and isolation may occur in a couple ways. Roychoudhury et al. (Roychoudhury, Biswas, & Koutsoukos, 2006) use a DBN derived from a bond graph with SIS of the healthy system for tracking. The belief state estimate estimates are compared with 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. (Jha et al., 2009) who use a discrete DBN to determine the most probable explanation of transient faults. However, the approach is offline and does not require any tracking. Still, it is successful in combining consistency and abduction-based diagnosis methods.

Bayes' factors (and related ratios) may be used to detect faults. Because the transition model $\mathrm{p}(\mathrm{x_k}|\mathrm{x_{k-1}})$ is not updated until a fault is diagnosed, after the fault occurs the model and measurements diverge and the Bayes' factor of the state estimate decreases. Thus, the Bayes' factor may be used to indicate the presence of a fault. Vaswani et al. (Vaswani, 2004) calculate $-log\left(Pr(\mathbf{z}_k|\mathbf{z}_{0:k-1}, M)\right)$ where M is the transition model (not updated for the current fault) for abrupt faults and use the posterior expected negative log-likelihood of the state under the prior distribution of the state for incipient faults. In both cases a threshold for detection is established. Sankararaman and Mahadevan (Sankararaman & Mahadevan, 2011) use Bayes' factors to indicate faults and update the Bayes' factor as new data is collected. Again, if the Bayes' factor crosses a predetermined threshold, the fault is then isolated by checking the state estimate of the system. Following the approach of Sankararaman and Mahadevan (Sankararaman & Mahadevan, 2011), the Bayes' factor is taken as

$$B = \frac{Pr(D|H_1)}{Pr(D|H_0)} \tag{5}$$

where D is is the set of measurements, H_0 is the no damage hypothesis, H_1 is the damage hypothesis, and $Pr(D|H_1) = 1 - Pr(D|H_0)$. The probability of damage is

$$Pr(H_1|D) = \frac{Pr(D|H_1)P(H_1)}{Pr(D|H_0)P(H_0) + Pr(D|H_1)P(H_1)} \tag{6}$$

If the prior probabilities of H_0 and H_1 are assumed to be 0.5, Eq. (5) reduces to

$$Pr(H_1|D) = \frac{B}{B+1} \tag{7}$$

If a new Bayes' factor is calculated based on new measurement information becoming available, the updated probabilities of H_0 and H_1 are

$$Pr(H_1|D) = \frac{B'B}{1+B'B} \tag{8}$$

and

$$Pr(H_0|D) = \frac{1}{1+B'B}. \tag{9}$$

Following the results of Vaswani et al. (Vaswani, 2004), for detection of abrupt faults, H_0 and H_1 refer to healthy and unhealthy realizations of the transition model, $p(\mathbf{x}_k^n|\mathbf{x}_{k-1}^n, \mathrm{r}_k^n)$, and we denote the corresponding Bayes' factor as B_t. For incipient faults, H_0 and H_1 refer to a baseline and current value of the state estimate and we denote the corresponding Bayes' factor as B_t. By updating both Bayes' factors and comparing to a threshold probability, fault detection can occur. Fault isolation is then achieved by determining the MMSE or MAP estimate from the state estimated, as approximated by samples and weights. After isolation, any parameter changes that need to be made to the state transition or measurement models are made.

2.5. Prognosis

In prognosis, it is desired to make an inference about the future distribution of the state of a system given its current state and possible future scenarios. As no new measurements are available, a Bayesian recursive filter which integrates measurement data is no longer necessary. The last set of state estimates and measurements are propagated through the DBN (as was done with the particle filter). The result is an estimate of the future distribution of the state variables.

3. Numerical Example

To illustrate the diagnosis methodology, fault detection, isolation, and prognosis is performed on a cantilever beam system subjected repeatedly to a point load at the free end. In this system, a crack is growing near the support according to Paris' law (Paris, Gomez, & Anderson, 1961)

$$\frac{da}{dN} = C(\Delta K)^m. \tag{10}$$

where C and m are material dependent constants, a is the crack length, N is the number of load cycles, and ΔK is the stress intensity factor.

Additionally, a bolt may become loose at the support, which is modeled by removing constraints on finite element model boundary conditions for nodes at the support up to a distance d from the bottom of the beam.

3.1.1. Model Construction

To obtain operational data, 50 load histories were constructed containing 100 load cycles each. In these load histories, it was assumed that crack length, deflection, load, and whether the bolt is loose or tight are observable. For each load history, a load P, Paris law parameter m,

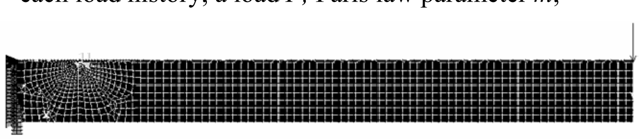

Figure 3. Meshed cantilever beam finite element model

Variable	Symbol	Distribution
Bolt state	B	Categorical, binary (tight, loose)
Load	P	Categorical, ternary (-.5 kN, -1 kN, -1.5 kips)
Crack length parameter A	a_A	Gaussian
Crack length parameter B	a_B	Gaussian
Crack length parameter C	a_C	Gaussian
Crack length parameter D	a_D	Gaussian
Deflection parameter A	δ_A	Gaussian
Deflection parameter B	δ_B	Gaussian
Deflection parameter C	δ_C	Gaussian
Deflection parameter D	δ_D	Gaussian

Table 1. Cantilever beam model variables.

exponential failure rate for the bolt loosening, and elastic modulus E (finite element model input) are selected. Paris' law parameter C was held constant across all experiments. A crack is initiated near the support of the beam. The beam is then repeatedly subjected to load P. With each application of P, the crack grows slightly larger, according to Paris' law. Additionally, there is a growing chance with each application of P that the bolt will become loose. The severity of damage is determined by d, which is selected before each load history. Table 1 describes the distributions used to model these variables. A rendering of the system in ANSYS is shown in Fig. 6.

A DBN model of the beam (Fig. 3) is systematically constructed from heterogeneous information. First expert opinion is solicited to determine the variables to be used in modeling the system and the structure of the DBN. The variables selected to represent the system model are the loose bolt indicator variable, B, the load P, the parameters of the two-term exponential model describing the crack length, a, and the deflection, δ. The two-term parameters of the exponential models are estimated over a sliding 10 load cycle window from smoothed crack length and deflection data. The parameters of the DBN are then calculated via the principle of maximum likelihood.

3.1.2. Diagnosis

To perform diagnosis, a 100 cycle load history was synthesized with P = -.5 kN and known deflection parameters, $\delta_A, \delta_B, \delta_C, \delta_D,$. The bolt condition, B, crack length parameters, a_A, a_B, a_C, a_D were assumed to be unobservable. The DBN is used in conjunction with a MMPF with 200 particles to obtained filtered estimates of the system state. After each load cycle, detection probabilities corresponding to the loose bolt condition are

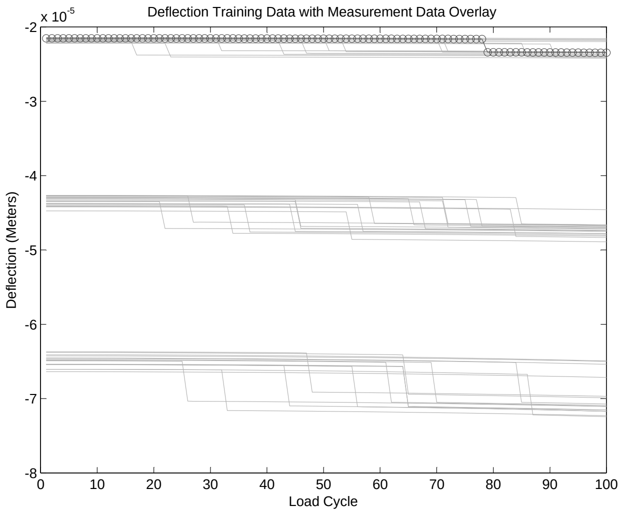

Figure 4. Deflection data for training DBN (gray lines) compared against measurement data.

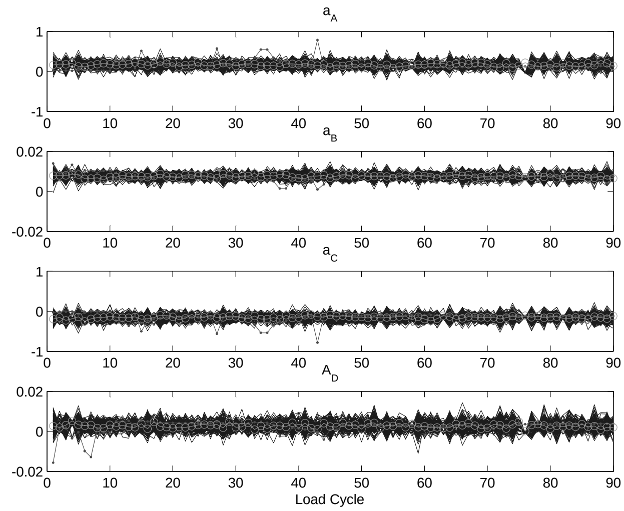

Figure 5. Overlay of MAP filtered estimate (gray with circles), true values (gray), and filtered estimates (black) for crack length parameters

updated as per Section II.D. The threshold probability of detection used is 95%. As the fault considered was abrupt, the Bayes' factor corresponding to the transition model as opposed to the belief state was used in detection.

3.1.3 Prognosis

After the last measurement data has been filtered, the system state is estimated 100 cycles into the future. As no new measurements are available, a Bayesian recursive filter which integrates measurement data is no longer necessary. The last set of state estimates and measurements are propagated through the DBN (as was done with the particle filter) with one estimate per particle. The result is an estimate of the future distribution of the model parameters. By using Eq. (2) with the model parameters, estimates of the crack length and deflection distributions are obtained. The model parameters are smoothed offline before estimating crack length and deflection so that trends are easier to visualize.

3.1.3. Results and Discussion

Data for the DBN for crack length and deflection used in training the DBN are overlaid with the measurement data used in diagnosis in Fig. 4. Considering the measurement data, the first item to discuss is the smoothness of the crack length and deflection curves versus the jagged model parameter curves. The jagged curves of particle values for the crack length parameters overlaid with the MAP estimate and true value are shown in Fig. 5. This is due to several uncertainties in estimating the model parameters. First, the curves produced by the FEM model may not be perfectly smooth. Locally, parts of the curve may not fit the two-term exponential model as well as others. A limited window of data is used in estimating these parameters, resulting infurther errors. Also, the optimization procedure used to estimate the model parameters induces errors.

The operational data lies in three bands, one for each value of P. When the bolt becomes loose, it is important to note that the deflection does not fall far outside the range of healthy deflection curves. This highlights the importance of integrating a mathematical model into this system. The parameters of the mathematical model, on the other hand, are able to isolate the loose bolt fault shortly after occurrence as well as estimate the crack length and deflection. The MAP estimate of the bolt condition, the true value of the bolt condition, and the measured values of δ_D, which is the variable most strongly connected with the bolt condition, are shown in Fig. 6.

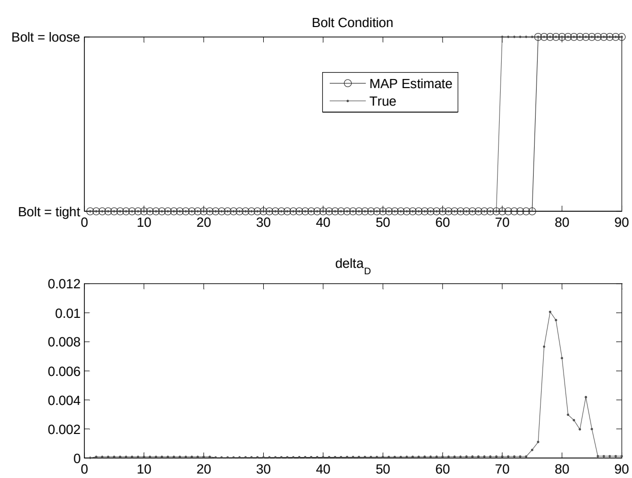

Figure 6. MAP estimate vs. true value of bolt condition and plot of $\boldsymbol{\delta_D}$ when a fault occurs.

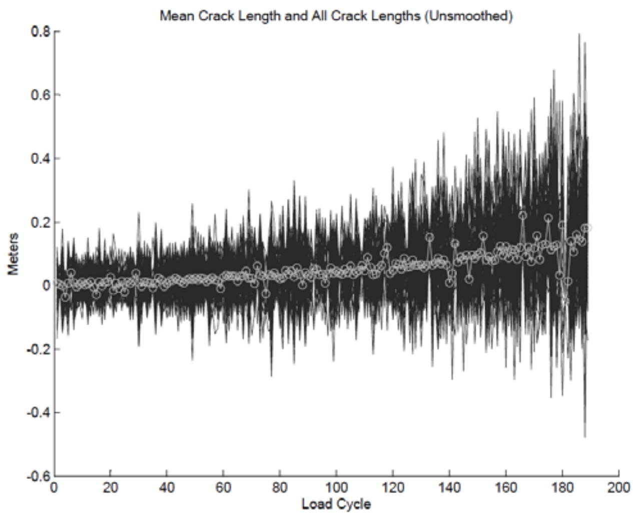

Figure 7. All crack length estimates (black) and mean crack length estimate (gray) computed from crack length mathematical model.

Prognosis of the system shows how small changes in the model parameters result in large amounts of uncertainty in prognosis. The prognosis results (Fig. 7) are in line with expectations, giving the trends evident in the training data (Fig. 4). Figure 8 shows histograms of crack length data after 115 and 175 load cycles, showing a gradual shift to towards larger crack lengths. Using MAP estimates and offline smoothing of the crack length (Fig. 9) and deflection parameters enhances the legibility of the plots but can induce some error, particularly near the beginning and end of a time series.

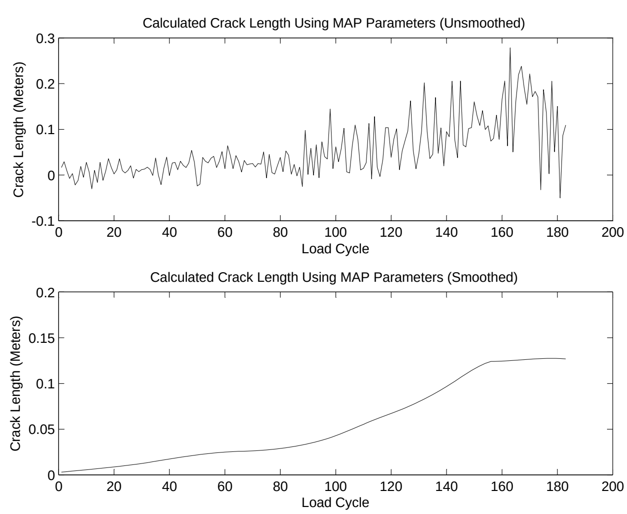

Figure 9. Crack length estimate using smoothed and unsmoothed MAP parameters.

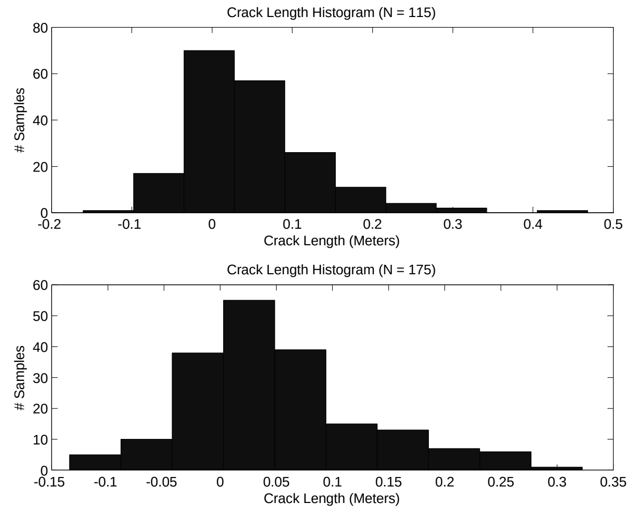

Figure 8. Histograms of crack length after 115 and 175 load cycles).

The results show that a DBN-based methodology can be used for diagnosis and prognosis of a mechanical system in the presence of many different types of information. Where other methodologies may exclusively model crack growth or detect faults, the DBN-based methodology is capable of doing both as it integrates expert opinion, a database of observational information, failure rate information, and a mathematical model for crack growth and deflection.

4. CONCLUSION

In this paper, a methodology for diagnosis and prognosis of a mechanical system in the presence of heterogeneous information has been proposed. First, a DBN is constructed using heterogeneous information such as expert opinion, operational and laboratory data, published reliability data, and mathematical models. Next, the DBN is employed for filtering the system. Filtering is performed via a particle filter. Faults are detected using Bayesian hypothesis testing. Then, by determining the MAP estimate of the state, faults are isolated. After the diagnosis has been performed, a prognosis estimate of the future distribution of the system state is obtained. In future work, a hydraulic actuator system constructed from heterogeneous information and containing multiple fault scenarios will be considered for PHM.

ACKNOWLEDGEMENT

This work is partly supported by the U. S. Air Force Research Laboratory through subcontract to General Dynamics Information Technology (Contract No. USAF-0060-43-0001, Project Monitor: Mark Derriso). The support is gratefully acknowledged.

NOMENCLATURE

B	=	Bayes' factor
DBN	=	dynamic Bayesian network
MAP	=	Maximum a posteriori estimate
MMSE	=	Minimum mean-square estimate
PF	=	Particle filter
SIS	=	Sequential importance sampling

REFERENCES

Arulampalam, M. S., Maskell, S., & Gordon, N. (2002). A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking. *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, *50*, 174–188.

Boyen, X., & Koller, D. (1998). Tractable inference for complex stochastic processes. *Proceedings of the Fourteenth conference on Uncertainty in artificial intelligence*, UAI'98 (pp. 33–42). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc. Retrieved from http://dl.acm.org/citation.cfm?id=2074094.2074099

Camci, F., & Chinnam, R. B. (2005). Dynamic Bayesian networks for machine diagnostics: hierarchical hidden Markov models vs. competitive learning. *2005 IEEE International Joint Conference on Neural Networks, 2005. IJCNN '05. Proceedings* (Vol. 3, pp. 1752– 1757 vol. 3). Presented at the 2005 IEEE International Joint Conference on Neural Networks, 2005. IJCNN '05. Proceedings, IEEE. doi:10.1109/IJCNN.2005.1556145

Cooper, G. F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, *9*(4), 309–347. doi:10.1007/BF00994110

de Campos, L. M. (2006). A Scoring Function for Learning Bayesian Networks based on Mutual Information and Conditional Independence Tests. *J. Mach. Learn. Res.*, *7*, 2149–2187.

Friedman, N. (1998). The Bayesian Structural EM Algorithm. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.24.1555

Friedman, N., Murphy, K., & Russell, S. (1998). Learning the Structure of Dynamic Probabilistic Networks. *UAI'98 Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence* (pp. 139–147). Morgan Kaufmann Publishers Inc. Retrieved from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.32.6615

Heckerman, D., & Geiger, D. (1995). Learning Bayesian Networks: A unification for discrete and Gaussian domains. *PROCEEDINGS OF ELEVENTH CONFERENCE ON UNCERTAINTY INARTI CIAL INTELLIGENCE*. Retrieved from http://citeseer.ist.psu.edu/viewdoc/summary?doi=10.1.1.156.7976

Jha, S., Wenchao Li, & Seshia, S. A. (2009). Localizing transient faults using dynamic bayesian networks. *High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International* (pp. 82–87). Presented at the High Level Design Validation and Test Workshop, 2009. HLDVT 2009. IEEE International, IEEE. doi:10.1109/HLDVT.2009.5340170

Lauritzen, S. L. (1992). Propagation of Probabilities, Means, and Variances in Mixed Graphical Association Models. *Journal of the American Statistical Association*, *87*(420), 1098–1108. doi:10.2307/2290647

Lerner, U. (2001). Exact inference in networks with discrete children of continuous parents. *in: J. Breese, D. Koller (Eds.), Uncertainty in Artificial Intelligence* (pp. 319–328). Morgan Kaufmann.

McNaught, K. R., & Zagorecki, A. (2009). Using dynamic Bayesian networks for prognostic modelling to inform maintenance decision making. *IEEE International Conference on Industrial Engineering and Engineering Management, 2009. IEEM 2009* (pp. 1155–1159). Presented at the IEEE International Conference on Industrial Engineering and Engineering Management, 2009. IEEM 2009, IEEE. doi:10.1109/IEEM.2009.5372973

Paris, P. C., Gomez, M. P., & Anderson, W. E. (1961). A Rational Analytic Theory of Fatigue. *The Trend in Engineering*, *13*, 9–14.

Przytula, K. W., & Choi, A. (2008). An Implementation of Prognosis with Dynamic Bayesian Networks. *2008 IEEE Aerospace Conference* (pp. 1–8). Presented at the 2008 IEEE Aerospace Conference, IEEE. doi:10.1109/AERO.2008.4526616

Ristic, B., & Arulampalam, S. (2004). *Beyond the Kalman filter : particle filters for tracking applications*. Boston, MA: Artech House.

Roychoudhury, I., Biswas, G., & Koutsoukos, X. (2006). A Bayesian approach to efficient diagnosis of incipient faults. *IN PROC. 17TH INT. WORKSHOP PRINCIPLES OF DIAGNOSIS*, 243–250.

Sahin, F., Yavuz, M. Ç., Arnavut, Z., & Uluyol, Ö. (2007). Fault diagnosis for airplane engines using Bayesian networks and distributed particle swarm optimization. *Parallel Computing*, *33*(2), 124–143. doi:16/j.parco.2006.11.005

Sankararaman, S., & Mahadevan, S. (2011). Uncertainty quantification in structural damage diagnosis.

Structural Control and Health Monitoring, *18*(8), 807–824. doi:10.1002/stc.400

Shwe, M. A., Middleton, B., Heckerman, D. E., Henrion, M., Horvitz, E. J., Lehmann, H. P., & Cooper, G. F. (1991). Probabilistic Diagnosis Using a Reformulation of the INTERNIST-1/QMR Knowledge Base. *Methods of Information in Medicine*, *30*, 241–55.

Straub, D. (2009). Stochastic Modeling of Deterioration Processes through Dynamic Bayesian Networks. *Journal of Engineering Mechanics*, *135*(10), 1089–1099. doi:10.1061/(ASCE)EM.1943-7889.0000024

Vaswani, N. (2004). *Bound on Errors in Particle Filtering with Incorrect Model Assumptions and Its Implication for Change Detection*.