AFRL-HE-WP-TR-2006-0059

STINFO COPY

AIR FORCE RESEARCH LABORATORY

Maintenance Decision-Making Under Prognostic and Diagnostic Uncertainty

C. Richard Cassady
Heather L. Nachtmann
Edward A. Pohl
Alejandro Mendoza
Letitia Pohl
Nick Rew

University of Arkansas
Department of Industrial Engineering
4207 Bell Engineering Center
Fayetteville, AR 72701

January 2005

Interim Report for November 2002 to January 2005

20060814291

Human Effectiveness Directorate
Warfighter Readiness Research Division
Logistics Readiness Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

AFRL-HE-WP-TR-2006-0059

STINFO COPY

AIR FORCE RESEARCH LABORATORY

Maintenance Decision-Making Under Prognostic and Diagnostic Uncertainty

C. Richard Cassady
Heather L. Nachtmann
Edward A. Pohl
Alejandro Mendoza
Letitia Pohl
Nick Rew

University of Arkansas
Department of Industrial Engineering
4207 Bell Engineering Center
Fayetteville, AR 72701

January 2005

Interim Report for November 2002 to January 2005

Human Effectiveness Directorate
Warfighter Readiness Research Division
Logistics Readiness Branch
2698 G Street
Wright-Patterson AFB OH 45433-7604

NOTICES

TECHNICAL REVIEW AND APPROVAL

AFRL-HE-WP-TR-2006-0059

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER

//SIGNED//

DANIEL R. WALKER, Colonel, USAF
Chief, Warfighter Readiness Research Division
Human Effectiveness Directorate

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*
January 2005

2. REPORT TYPE
Interim

3. DATES COVERED *(From - To)*
November 2002 - January 2005

4. TITLE AND SUBTITLE
Maintenance Decision-Making Under Prognostic and Diagnostic Uncertainty

5a. CONTRACT NUMBER
F33615-99-D-6001

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER
63231F

6. AUTHOR(S)
C. Richard Cassady, Heather L. Nachtmann, Edward A. Pohl, Alejandro Mendoza, Letitia Pohl, Nick Rew

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER
49230208

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
University of Arkansas
Department of Industrial Engineering
4207 Bell Engineering Center
Fayetteville, AR 72701

8. PERFORMING ORGANIZATION REPORT NUMBER
SMM0301

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Air Force Materiel Command
Air Force Research Laboratory
Human Effectiveness Directorate
Warfighter Readiness Research Division
Logistics Readiness Branch
Wright-Patterson AFB OH 45433-7604

10. SPONSOR/MONITOR'S ACRONYM(S)
AFRL/HEAL

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
AFRL-HE-WP-TR-2006-0059

12. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES
DO #26, Task 2 Cleared as AFRL/WS-06-1578, 21 Jun 06.

14. ABSTRACT
A key challenge faced by USAF maintenance personnel is the uncertainty associated with the information provided by diagnostic tools. This uncertainty results from accuracy issues associated with individual diagnostic tools, as well as inconsistencies across different diagnostic tools. This uncertainty can make it very difficult for maintenance technicians to choose an appropriate course of action. The end result is the possible omission of necessary maintenance actions and performance of unnecessary actions. Both of these potential mistakes cause additional delays in returning an aircraft to the fleet and increased requirements for spare parts in the supply chain. Therefore, the objective of this project is to develop a methodology based on mathematical modeling that can be used to synthesize diagnostic information and provide a recommended course of action to a technician. This methodology potential could be incorporated into a decision-support tool for maintenance technicians. The activities required to achieve the objective of this project are applied to a hypothetical system. However, the definition of this hypothetical system is such that the system possesses fundamental characteristics like those systems utilized by the US Air Force (and many other organizations).

15. SUBJECT TERMS
Aircraft Maintenance, Maintenance Technicians, Diagnostic Tool, Prognostic Tool, Decision-Support Tool, Artificial Neural Networks

16. SECURITY CLASSIFICATION OF: a. REPORT	b. ABSTRACT	c. THIS PAGE	17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR	26	Cheryl L. Batchelor
					19b. TELEPHONE NUMBER *(include area code)*

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. 239.18

THIS PAGE LEFT INTENTIONALLY BLANK

Executive Summary

A key challenge faced by USAF maintenance personnel is the uncertainty associated with the information provided by diagnostic tools. This uncertainty can make it very difficult for maintenance technicians to choose an appropriate course of action. The end result is the possible omission of necessary maintenance actions and performance of unnecessary actions. Both of these potential mistakes cause additional delays in returning an aircraft to the fleet and increased requirements for spare parts in the supply chain. Therefore, the objective of this project is to develop a methodology based on mathematical modeling that can be used to synthesize the diagnostic information and provide a recommended course of action to the technician.

For a hypothetical system that possesses fundamental characteristics like those systems utilized by the US Air Force (and many other organizations), we develop a two modeling-based methodologies for synthesizing diagnostic information and providing an estimated assessment of the system. First, we define a probabilistic approach for synthesizing imperfect and conflicting diagnostic information. We define the characteristics of the system of interest and the diagnostics applied to this system. We demonstrate how probabilistic analysis can be used to provide an assessment of system status, and, using a numerical example, we demonstrate the potential effectiveness of the approach.

The probabilistic approach shows great promise as a means of compiling imperfect and conflicting diagnostic information. However, the approach requires exact monitoring of component aging and perfect life distribution estimation. Furthermore, our approach requires an assumption of independent component failures. Therefore, we explore an alternative approach based on artificial neural networks (ANN). This approach does not suffer from either of the identified limitations of the probabilistic approach. However, the numerical results associated with this new approach are not as promising.

Table of Contents

List of Tables iv

1. Introduction 1

2. Research Literature Review 2

3. A Probabilistic Approach 7

   3.1 System Characteristics 7

   3.2 The Probabilistic Analysis 9

   3.3 A Simulation-Based Assessment 11

4. A Neural Network Approach 15

   4.1 System Characteristics 15

   4.2 Data Generation 17

   4.3 The Use of the Artificial Neural Network 18

References 22

List of Tables

Table 3.1 Analysis of the Probabilistic Approach 14

Table 4.1 Back-propagation Network Settings 21

1. Introduction

A key challenge faced by USAF maintenance personnel is the uncertainty associated with the information provided by diagnostic tools. This uncertainty results from accuracy issues associated with individual diagnostic tools, as well as inconsistencies across different diagnostic tools. This uncertainty can make it very difficult for maintenance technicians to choose an appropriate course of action. The end result is the possible omission of necessary maintenance actions and performance of unnecessary actions. Both of these potential mistakes cause additional delays in returning an aircraft to the fleet and increased requirements for spare parts in the supply chain. Therefore, the objective of this project is to develop a methodology based on mathematical modeling that can be used to synthesize diagnostic information and provide a recommended course of action to a technician. This methodology potentially could be incorporated into a decision-support tool for maintenance technicians.

The activities required to achieve the objective of this project are applied to a hypothetical system. However, the definition of this hypothetical system is such that the system possesses fundamental characteristics like those systems utilized by the US Air Force (and many other organizations). First, we define the system structure and the reliability and maintainability characteristics of each component in the system. Second, we identify the characteristics of the diagnostic tools applied to the system. This identification includes a description of the accuracy of diagnostic information. Third, we develop a set of mathematical and logical models which synthesize the diagnostic information and provide an estimated assessment of system status. Finally, we utilize numerical experiments for assessing the capabilities of the defined models.

The remainder of this report is summarized as follows. In Section 2, we summarize the relevant research literature. Section 3 contains the development an analysis of a probabilistic approach to synthesizing imperfect and conflicting diagnostic information. In Section 4, we explore an alternative approach based on the use of artificial neural networks.

2. Research Literature Review

The purpose of this literature review is to identify existing mathematical modeling techniques used in the area of diagnostics. Diagnostics is the first step in the repair process and involves identifying the cause of a failure. Typically, the goal is to isolate the failure to a faulty module and/or component, and this is done based on system observations and available test data. Often, the determination that a failure has occurred is one step (e.g., the failure of a built-in test) and the isolation of that failure is a second step. In other applications, however, failure detection and isolation are not separable. For example, the diagnostic problem may be formulated as a classification problem, where the system state is classified as either normal operation, or as one of several possible failure modes [6]. We are particularly interested in techniques that take into consideration imperfect test results, which introduce uncertainty into the diagnosis. The two main types of test error include (1) the test indicates a pass, when in fact, the unit under test has failed, and (2) the test indicates a fail, when the unit is working properly (a false alarm).

Fault diagnosis in large-scale systems has been a major research area for several decades and there is considerable literature available. The inter-disciplinary problem of diagnostics is a concern in all stages of the product life cycle, but particularly during manufacturing and field maintenance [3]. It has therefore been approached from the perspective of the electronics design engineer, the diagnostics software developer, the reliability engineer, and others. Many of these techniques require specific information about the system design, and in fact, the models may be constructed during the design phase. Because the diagnostics process has traditionally been dependent on human involvement, the development of automated diagnostics systems has also frequently relied on artificial intelligence (AI) techniques.

A well established approach to diagnostics is the Bayesian process. Bayesian inference can be used to determine the probability that a diagnosis is correct. However, it has the disadvantage of requiring a priori probability distributions, which may not always be available [8].

Fenton [3], in his review of AI approaches to diagnostics, states that model-based diagnosis involves using the model to predict faults from observations and information on the real device or system. He identifies four types of models and provides numerous references as examples of their application: fault models (or fault dictionaries), causal models, models based on structure and behavior, and diagnostic inference models.

Fault models anticipate the types of faults that may occur and only model those. Each fault type is inserted into the system and the system behavior is monitored. From this, a list of fault/symptom pairs or fault dictionary is produced. This method has been used primarily for digital circuit diagnosis. These models are unable to handle unanticipated faults.

A causal model is a directed graph, where nodes represent symptoms and faults, and the links represent the relationships between them. The strength of each link is often defined using a numerical weight or probability. The fault hypotheses are ranked or eliminated using Bayesian techniques. Bayesian networks are a variation on this approach.

Models based on structure and behavior require detailed information on the system components, their interconnections, and the behavior pattern for each component. This type of model can theoretically diagnose any fault type, which overcomes the disadvantage of a fault model, which cannot detect unanticipated errors [3].

Diagnostic inference models represent the problem as a flow of diagnostic information. The model consists of two basic elements: tests and conclusions. Tests may be any source of diagnostic information, including observable symptoms, logistics history and results from diagnostic tests. Conclusions typically represent faults or units to replace. The dependency relationship between tests and conclusions is represented using a directed graph.

In [6], the types of models used in diagnostics are identified as physical models, reliability models, machine learning models, and dependency models. Physical models are based on the natural laws governing system operation, e.g., material properties (solid, liquid, gas), finite-element models, thermodynamics, etc. A physics-based failure model usually needs to be built for each failure mode, and requires intricate knowledge by area experts. Reliability modeling requires knowledge of the system structure and failure probability distributions. Machine learning models are purely data dependent models and require historical training data. Neural networks are the primary example of machine learning models. Dependency models capture cause and effect relationships. An example of a failure dependency model is provided in [6].

Deb *et al.* [2] describe four modeling techniques for diagnosing faults in complex systems: quantitative, qualitative, structural and dependency. Quantitative models require highly detailed system information and provide an exact simulation of the system. Qualitative models are simplified quantitative models. Structural models represent the connectivity and failure propagation direction in the form of a directed graph. An example is found in [4], where a directed graph is used to represent the propagation of a fault through the system. Each node represents a unit (or its failure mode) and a link between two nodes indicates that a fault can propagate from one to the other. Dependency models (similarly defined in [6]) represent the cause-effect relationships in the form of a directed graph, and can deviate significantly from structural models. According to [2], dependency modeling, which is also referred to elsewhere as inference modeling, is the primary modeling technique employed in testability analysis tools.

Fenton [3] summarizes the use of fuzzy logic and artificial neural networks in diagnostics. Fuzzy logic can be used to represent uncertainty and inaccurate data in a diagnostics environment - approximations rather than exact measurements. It can also be used to incorporate qualitative judgments from experts into an automated diagnostics system. In traditional sets, membership is either true (1) or false (0), and there is no concept of partial membership. In fuzzy sets, partial membership is allowed, so membership is represented by a value between 0 and 1. Fuzzy logic is typically combined with other modeling approaches. One such application is found in [8] and several more are identified in [3].

Artificial neural networks (ANNs) are used for a variety of applications, including diagnostics. ANNs are basically directed graphs with nodes, or neurons, connected by weighted links. Each link has an associated weight, which typically multiplies the signal transmitted along that link. Each neuron applies an activation function to its net input (sum of weighted input signals) to determine its output signal. The net can be single layer (containing only a set of input units and a set of output units, with a single set of weighted links), or more commonly, multilayer (one or more layers of nodes between the input and output units). The process of establishing the weights for each link is called training. The neural net is "trained" with data to perform a function. In the case of diagnostics, the input data may be the results of diagnostic tests and the output could be an indication of which subsystem has failed. An ANN is characterized by its structure of nodes and links, method of training, and activation function.

In [8], methods to combine system information (such as test results) to improve the confidence and accuracy of diagnostics are examined. One of the data fusion approaches proposed uses neural networks. An example is given using engine test cell data, where the output is a determination of the validity of the sensor signals, and at times, diagnosis of a sensor fault. In [10], a neural network is presented that attempts to shrink the confidence bounds around failure prediction. In [11], a self-organizing feature map (SOM) neural network, combined with fuzzy logic, is implemented. The types of neural networks most commonly used in diagnostics are multilayer, feed-forward networks with back-propagation training (see [5], [7], and [9]). Fenton [3] states that ANNs are most useful for their ability to recognize patterns and have shown promise in application where noise and error is present. Mather *et al.* [6] acknowledge that neural nets are useful for modeling phenomena that are hard to model using parametric/analytical equations. However, they are difficult to validate and do not enhance the basic understanding of the system under study.

Finally, a method for evaluating the performance of automatic diagnostic systems is presented in [1]. Three measures of effectiveness for a diagnostics system are defined, which include the false positive and false negative errors previously mentioned, plus a third measure defined as false alarm correction. The false alarm correction measures the ability of the diagnostics system to correct its actions after indicating a false alarm. For the purpose of comparing various diagnostics systems, the paper develops a method for evaluating the life cycle cost of a diagnostics system, based on the three measures of effectiveness.

3. A Probabilistic Approach

In this section, we define a probabilistic probability approach for synthesizing imperfect and conflicting diagnostic information. We begin by defining the characteristics of the hypothetical system of interest, as well as the diagnostics applied to this system. Then, we demonstrate how probabilistic analysis can be used to provide an assessment of system status based on component time to failure behavior and the diagnostic results. Using a set of numerical examples, we then demonstrate the potential effectiveness of the approach.

3.1 System Characteristics

Consider a system comprised of M independent, binary-state (functioning, failed) components that is required to perform a sequence of missions each having a length of l. During each mission, the system is subject to one or more individual component failures. Failed components can only be replaced, and these replacements (system maintenance) take place only between missions. Note that functional components neither age nor fail during system maintenance. Let T_m denote the time to individual failure of a new copy of component m, $m = 1, 2, \dots, M$, and note that T_m is governed by a Weibull probability distribution having shape parameter $\theta_m \geq 1$ and scale parameter $\eta_m > 0$. Therefore, the cumulative distribution function of T_m is given by

$$G_m(t) = 1 - \exp\left(-\left(t/\eta_m\right)^{\theta_m}\right) \tag{3.1}$$

Note that the fact that $\theta_m \geq 1$, $m = 1, 2, \dots, M$, implies that components have either constant or increasing failure rates.

Upon completion of each mission, some or all of the components may be failed. A built-in-test is used to determine if there is one or more failed components, and this test is assumed to be perfect. However, the test does not identify which components are failed. Note that if there are no failed components, then the system starts its next mission.

If the built-in test reveals that at least one component failed during the previous mission, then a set of D independent diagnostics are used in an attempt to determine the status of each component. Each diagnostic provides an independent assessment of the status of some subset of the components. Let

$$c_{d,m} = \begin{cases} 1 & \text{if diagnostic } d \text{ assesses component } m \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

$d = 1, 2, \ldots, D$, $m = 1, 2, \ldots, M$. Furthermore, let

$$X_{d,m} = \begin{cases} 1 & \text{if diagnostic } d \text{ indicates that component } m \text{ is failed} \\ 0 & \text{otherwise} \end{cases} \tag{3.3}$$

$d = 1, 2, \ldots, D$, $m = 1, 2, \ldots, M$. Unfortunately, each diagnostic is subject to Type I (false positive) and Type II (false negative) errors. Let

$$Y_m = \begin{cases} 1 & \text{if component } m \text{ is failed} \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

$m = 1, 2, \ldots, M$. Then,

$$\alpha_{d,m} = \Pr\left(X_{d,m} = 1 \middle| Y_m = 0\right) \tag{3.5}$$

is the probability that diagnostic d produces a false positive regarding component m and

$$\beta_{d,m} = \Pr\left(X_{d,m} = 0 \middle| Y_m = 1\right) \tag{3.6}$$

is the probability that diagnostic d fails to detect the failure of component m, $d = 1, 2, \ldots, D$, $m = 1, 2, \ldots, M$.

We assume that, eventually, the failed components are correctly identified and replaced and the system starts its next mission. However, our focus in this study is on the first attempt at diagnosing the failed components.

3.2 The Probabilistic Analysis

Suppose a mission has just ended and the built-in test indicates that there is one or more failed components. A critical assumption required for the effective use of this probabilistic approach is that the system manager can track the age of each component in the system. Note that the age of a component refers to the elapsed mission time since the component was last replaced. Let a_m denote the age of component m at the beginning of the last mission, $m = 1, 2, \ldots, M$. Then, the probability that component m failed during the last mission is given by

$$\rho_m = \frac{G_m(l + a_m) - G_m(a_m)}{1 - G_m(a_m)} \tag{3.7}$$

$m = 1, 2, \ldots, M$.

Since the built-in test revealed at least one component failure, the set of D diagnostics are applied to the system. Let D_m denote the set of diagnostics that assess component m, i.e.

$$D_m = \{d \in \{1, 2, \ldots, D\} \mid c_{d,m} = 1\} \tag{3.8}$$

$m = 1, 2, \ldots, M$. Then,

$$\Pr(X_{d,m} = x_{d,m} \mid Y_m = 1) = (1 - \beta_{d,m})^{x_{d,m}} \beta_{d,m}^{(1-x_{d,m})} \tag{3.9}$$

$x_{d,m} = 0, 1$; $m = 1, 2, \ldots, M$; $d \in D_m$. Also,

$$\Pr(X_{d,m} = x_{d,m} \mid Y_m = 0) = \alpha_{d,m}^{x_{d,m}} (1 - \alpha_{d,m})^{(1-x_{d,m})} \tag{3.10}$$

$x_{d,m} = 0, 1$; $m = 1, 2, \ldots, M$; $d \in D_m$. Let $\vec{X}_m$ denote the vector of diagnostic results associated with component m, and let $\vec{x}_m$ denote a specific realization of $\vec{X}_m$, $m = 1, 2, \ldots, M$, i.e,

$$\vec{X}_m = [X_{d,m} \mid d \in D_m] \tag{3.11}$$

and

$$\vec{x}_m = [x_{d,m} \mid d \in D_m] \tag{3.12}$$

Since the D diagnostics are independent,

$$\Pr\left(\vec{X}_m = \vec{x}_m \mid Y_m = 1\right) = \prod_{d \in D_m} \Pr\left(X_{d,m} = x_{d,m} \mid Y_m = 1\right) \tag{3.13}$$

and

$$\Pr\left(\vec{X}_m = \vec{x}_m \mid Y_m = 0\right) = \prod_{d \in D_m} \Pr\left(X_{d,m} = x_{d,m} \mid Y_m = 0\right) \tag{3.14}$$

$m = 1, 2, \ldots, M$. Applying the law of total probability yields

$$\Pr\left(\vec{X}_m = \vec{x}_m\right) = \Pr\left(\vec{X}_m = \vec{x}_m \mid Y_m = 1\right)\rho_m + \Pr\left(\vec{X}_m = \vec{x}_m \mid Y_m = 0\right)\left(1 - \rho_m\right) \tag{3.15}$$

$m = 1, 2, \ldots, M$. Finally, application of Bayes' Theorem yields

$$\pi_m\left(\vec{x}_m\right) = \Pr\left(Y_m = 1 \mid \vec{X}_m = \vec{x}_m\right) = \frac{\Pr\left(\vec{X}_m = \vec{x}_m \mid Y_m = 1\right)\rho_m}{\Pr\left(\vec{X}_m = \vec{x}_m\right)} \tag{3.16}$$

which can be rewritten as

$$\pi_m\left(\vec{x}_m\right) = \frac{\rho_m \prod_{d \in D_m} \left(1 - \beta_{d,m}\right)^{x_{d,m}} \beta_{d,m}^{\left(1 - x_{d,m}\right)} + \left(1 - \rho_m\right) \prod_{d \in D_m} \alpha_{d,m}^{x_{d,m}} \left(1 - \alpha_{d,m}\right)^{\left(1 - x_{d,m}\right)}}{\prod_{d \in D_m} \left(1 - \beta_{d,m}\right)^{x_{d,m}} \beta_{d,m}^{\left(1 - x_{d,m}\right)}} \tag{3.17}$$

$m = 1, 2, \ldots, M$. Thus, $\pi_m\left(\vec{x}_m\right)$ provides a Bayesian update to the probability of component m failure based on the diagnostic results.

Based on this probabilistic analysis, we propose the following policy. First, compute ρ_m, $m = 1, 2, \ldots, M$. Second, perform the diagnostics. Third, compute $\pi_m\left(\vec{x}_m\right)$, $m = 1, 2, \ldots, M$. Finally, if

$$\pi_m\left(\vec{x}_m\right) > \pi_0 \tag{3.18}$$

then conclude that component m is failed, $m = 1, 2, \ldots, M$. Note that the value of π_0 is specified by the decision-maker.

3.3 A Simulation-Based Assessment

To facilitate study of the probabilistic policy, we constructed a discrete-event simulation model of system performance. The model, coded in Visual Basic, mimics the operation, failure, testing, and initial diagnosis of the system over a sequence of a user-specified number of missions. The Visual Basic form containing the simulation code, a Visual Basic module required to run the simulation, and a dll file required to run the simulation are included on the CD accompanying this report. The inputs to the model include: the number of components, the Weibull life distribution parameters for each component, the mission length, the number of diagnostics, the coverage of each diagnostic, and the Type I and II error probabilities for each component/diagnostic combination.

For the first simulated mission, Weibull random variates are generated and set as the initial time to failure for each component. The time to failure values are compared to the mission length to determine if each component can complete the mission. For components that survived the mission, the remainder of their time to failure is stored. If all components survived the mission, then the next mission is initiated. If the system suffered at least one component failure, then initial diagnostics are conducted. Monte Carlo analysis is used to determine the diagnostic results. The diagnostic results and actual system status are stored as the output of the model. Prior to starting the next mission, all failed components are renewed and given a new time to failure drawn from the appropriate Weibull probability distribution. Note that to avoid initial condition bias, a set of a user-specified number of "warm-up" missions are simulated before data collection begins.

As a numerical example, we simulated 5000 missions (after 500 warm-up missions) for a system having M = 10 components that performs sequential missions of length l = 0.5. This system is analyzed using D = 5 diagnostics. The remaining system parameters are:

$$\vec{\theta} = \begin{pmatrix} 1.0 & 1.5 & 2.5 & 1.0 & 2.0 & 1.0 & 1.5 & 1.0 & 2.0 & 2.5 \end{pmatrix} \tag{3.19}$$

$$\vec{\eta} = \begin{pmatrix} 1.0 & 4.0 & 1.5 & 5.5 & 4.5 & 3.0 & 2.0 & 2.5 & 3.5 & 5.0 \end{pmatrix} \tag{3.20}$$

$$\vec{c} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{3.21}$$

$$\vec{\alpha} = \begin{pmatrix} 0.01 & 0.08 & 0.02 & 0.04 & 0.03 & 0.07 & 0.1 & 0.05 & 0.06 & 0.09 \\ 0.04 & 0.02 & 0.01 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.08 & 0.02 & 0.05 & 0 & 0 & 0 & 0 & 0.07 \\ 0.02 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0.10 & 0.05 & 0.03 \\ 0 & 0 & 0 & 0 & 0 & 0.02 & 0.05 & 0.04 & 0 & 0 \end{pmatrix} \tag{3.22}$$

$$\vec{\beta} = \begin{pmatrix} 0.04 & 0.10 & 0.01 & 0.02 & 0.03 & 0.08 & 0.09 & 0.05 & 0.06 & 0.07 \\ 0.03 & 0.05 & 0.07 & 0.06 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0.06 & 0.02 & 0 & 0 & 0 & 0 & 0.04 \\ 0.06 & 0 & 0 & 0 & 0.09 & 0 & 0 & 0.01 & 0.03 & 0.04 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0.04 & 0.07 & 0 & 0 \end{pmatrix} \tag{3.23}$$

Of the 5000 missions simulated, there were 4416 during which the system experienced at least one failure. For those 4416 missions, there were 9350 component failures (out of a possible 44,160 component-missions). When applied using a threshold of π_0 = 0.5, the probabilistic approach resulted in 151 false positives, 9003 true positives, 347 false negatives and 34,659 true negatives. To provide a comparison, two additional algorithms were considered: a voting algorithm and a signal algorithm. With the voting algorithm, a component is deemed to have failed if a majority of the diagnostics applied to that component indicate failure. With the signal algorithm, a component is deemed to have failed if any of the diagnostics applied to the component indicate failure. For the same example, the voting algorithm produced 1629 false positives and 25 false negatives. The signal algorithm produced 4095 false positives and 9 false negatives.

To further investigate the capability of the probabilistic approach, we conducted a more thorough numerical experiment using twelve combinations of M and D. These combinations are enumerated in Table 3.1. For each combination, we randomly generated 1000 scenarios as follows:

> The scenario included 5000 missions (after a warm-up period of 50 missions). Each mission was of length 0.5.
>
> The scale parameter of the Weibull probability distribution for each component was randomly selected from the set {1.0, 1.5, 2.0, 2.5}.
>
> The shape parameter of the Weibull probability distribution for each component was randomly selected from the set {1.0, 1.5, ... , 5.5}.
>
> The first diagnostic assesses all components. For all other diagnostics, there is a 40% chance that the diagnostic covers each component.
>
> For each component/diagnostic combination, the probability of a Type I error was randomly selected from the range (0.01, 0.05).
>
> For each component/diagnostic combination, the probability of a Type II error was randomly selected from the range (0.05, 0.10).

For each scenario, we compared the performance of the probabilistic approach (with $\pi_0 = 0.5$) to a voting algorithm. The results are summarized in Table 3.1 and suggest that the probabilistic approach can improve upon a voting algorithm.

			Probabilistic Approach		Voting Algorithm	
Combination	*M*	*D*	False Positives	False Negatives	False Positives	False Negatives
1	5	2	1.21%	29.54%	1.39%	32.40%
2	5	3	0.94%	20.08%	1.62%	23.34%
3	5	4	0.77%	13.59%	1.85%	16.58%
4	10	3	1.61%	21.66%	1.79%	26.33%
5	10	5	0.99%	10.53%	1.90%	13.60%
6	10	7	0.60%	5.28%	2.03%	7.42%
7	15	5	1.14%	11.07%	1.68%	14.28%
8	15	7	0.69%	5.64%	1.67%	8.02%
9	15	10	0.30%	2.15%	1.83%	3.55%
10	20	5	1.20%	11.21%	1.51%	14.63%
11	20	10	0.32%	2.35%	1.47%	3.82%
12	20	15	0.08%	0.48%	1.84%	1.03%

Table 3.1 Analysis of the Probabilistic Approach

4. A Neural Network Approach

The probabilistic approach shows great promise as a means of compiling imperfect and conflicting diagnostic information. However, the approach we use is limited in two ways. First, our approach requires exact monitoring of component aging and perfect life distribution estimation. Second, our approach requires an assumption of independent component failures. In this section, we explore an alternative approach based on artificial neural networks (ANN). This approach does not suffer from either of the identified limitations of the probabilistic approach. However, the numerical results associated with this new approach are not as promising.

4.1 System Characteristics

Consider a system comprised of M binary-state (functioning, failed) components that is required to perform a sequence of missions each having a length of l. During each mission, the system is subject to one or more individual component failures as well as some number of common-cause failures. Failed components can only be replaced, and these replacements (system maintenance) take place only between missions. Note that functional components do not age or fail during system maintenance. Let T_m denote the time to individual failure of a new copy of component m, $m = 1, 2, \ldots, M$, and note that T_m is governed by a Weibull probability distribution having shape parameter $\theta_m \geq 1$ and scale parameter $\eta_m > 0$. Therefore, the cumulative distribution function of T_m is given by

$$G_m(t) = 1 - \exp\left(-\left(t/\eta_m\right)^{\theta_m}\right) \tag{4.1}$$

Note that the fact that $\theta_m \geq 1$, $m = 1, 2, \ldots, M$, implies that components have either constant or increasing failure rates.

The system is also subject to F types of random, common-cause failures. Let γ_f denote the probability that common-cause failure type f occurs during a single mission, $f = 1, 2, \ldots, F$. Furthermore, let

$$\phi_{f,m} = \begin{cases} 1 & \text{if common-cause failure } f \text{ affects component } m \\ 0 & \text{otherwise} \end{cases} \tag{4.2}$$

$f = 1, 2, \ldots, F$, $m = 1, 2, \ldots, M$.

Upon completion of each mission, some or all of the components may be failed. A built-in-test is used to determine if there is one or more failed components, and this test is assumed to be perfect. However, the test does not identify which components are failed. Note that if there are no failed components, then the system starts its next mission.

If the built-in test reveals that at least one component failed during the previous mission, then a set of D independent diagnostics are used in an attempt to determine the status of each component. Each diagnostic provides an independent assessment of the status of some subset of the components. Let

$$c_{d,m} = \begin{cases} 1 & \text{if diagnostic } d \text{ assesses component } m \\ 0 & \text{otherwise} \end{cases} \tag{4.3}$$

$d = 1, 2, \ldots, D$, $m = 1, 2, \ldots, M$. Furthermore, let

$$X_{d,m} = \begin{cases} 1 & \text{if diagnostic } d \text{ indicates that component } m \text{ is failed} \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

$d = 1, 2, \ldots, D$, $m = 1, 2, \ldots, M$. Unfortunately, each diagnostic is subject to Type I (false positive) and Type II (false negative) errors. Let

$$Y_m = \begin{cases} 1 & \text{if component } m \text{ is failed} \\ 0 & \text{otherwise} \end{cases} \tag{4.5}$$

$m = 1, 2, \ldots, M$. Then,

$$\alpha_{d,m} = \Pr\left(X_{d,m} = 1 \middle| Y_m = 0\right) \tag{4.6}$$

is the probability that diagnostic d produces a false positive regarding component m and

$$\beta_{d,m} = \Pr\left(X_{d,m} = 0 \middle| Y_m = 1\right) \tag{4.7}$$

is the probability that diagnostic d fails to detect the failure of component m, $d = 1, 2, \ldots, D$, $m = 1, 2, \ldots, M$.

We assume that, eventually, the failed components are correctly identified and the system starts its next mission. However, our focus in this study is on the first attempt at diagnosing the failed components.

4.2 Data Generation

To facilitate study of the ANN-based policy, we constructed a discrete-event simulation model of system performance. The model, coded in Visual Basic, mimics the operation, failure, testing, and initial diagnosis of the system over a sequence of a user-specified number of missions. The Visual Basic form containing the simulation code, a Visual Basic module required to run the simulation, and a dll file required to run the simulation are included on the CD accompanying this report. The inputs to the model include: the number of components, the Weibull life distribution parameters for each component, the mission length, the number of common-cause failures, the probability of and components affected by each common-cause failure, the number of diagnostics, the coverage of each diagnostic, and the Type I and II error probabilities for each component/diagnostic combination.

For the first simulated mission, Weibull random variates are generated and set as the initial time to failure for each component. The time to failure values are compared to the mission length to determine if each component can complete the mission. Monte Carlo analysis is used to determine if each type of common-cause failure occurs. If a common-cause failure occurs, each affected component is failed.

For components that survived the mission, the remainder of their time to failure is stored. If all components survived the mission, then the next mission is initiated. If the system suffered at least one component failure, then initial diagnostics are conducted. Monte Carlo analysis is used to determine the diagnostic results. The diagnostic results and actual system status are stored as the output of the model. Prior to starting the next mission, all failed components are renewed and given a new time to failure drawn from the appropriate Weibull probability distribution. Note that to avoid initial condition bias, a set of a user-specified number of "warm-up" missions are simulated before data collection begins.

4.3 The Use of the Artificial Neural Network

As a numerical example, we simulated 5000 missions (after 500 warm-up missions) for a system having $M = 10$ components that performs sequential missions of length $l = 0.5$. In addition to individual component failures, the system is subject to $F = 2$ common-cause failures. Upon failure, this system is analyzed using $D = 5$ diagnostics. The remaining system parameters are:

$$\vec{\theta} = \begin{pmatrix} 1.0 & 1.5 & 2.5 & 1.0 & 2.0 & 1.0 & 1.5 & 1.0 & 2.0 & 2.5 \end{pmatrix} \tag{4.8}$$

$$\vec{\eta} = \begin{pmatrix} 1.0 & 4.0 & 1.5 & 5.5 & 4.5 & 3.0 & 2.0 & 2.5 & 3.5 & 5.0 \end{pmatrix} \tag{4.9}$$

$$\vec{\gamma} = \begin{pmatrix} 0.05 & 0.02 \end{pmatrix} \tag{4.10}$$

$$\vec{\phi} = \begin{pmatrix} 0 & 0 & 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 \end{pmatrix} \tag{4.11}$$

$$
\vec{c} = \begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 \end{pmatrix} \tag{4.12}
$$

$$
\vec{\alpha} = \begin{pmatrix} 0.01 & 0.08 & 0.02 & 0.04 & 0.03 & 0.07 & 0.1 & 0.05 & 0.06 & 0.09 \\ 0.04 & 0.02 & 0.01 & 0.05 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.08 & 0.02 & 0.05 & 0 & 0 & 0 & 0 & 0.07 \\ 0.02 & 0 & 0 & 0 & 0.01 & 0 & 0 & 0.10 & 0.05 & 0.03 \\ 0 & 0 & 0 & 0 & 0 & 0.02 & 0.05 & 0.04 & 0 & 0 \end{pmatrix} \tag{4.13}
$$

$$
\vec{\beta} = \begin{pmatrix} 0.04 & 0.10 & 0.01 & 0.02 & 0.03 & 0.08 & 0.09 & 0.05 & 0.06 & 0.07 \\ 0.03 & 0.05 & 0.07 & 0.06 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.01 & 0.06 & 0.02 & 0 & 0 & 0 & 0 & 0.04 \\ 0.06 & 0 & 0 & 0 & 0.09 & 0 & 0 & 0.01 & 0.03 & 0.04 \\ 0 & 0 & 0 & 0 & 0 & 0.01 & 0.04 & 0.07 & 0 & 0 \end{pmatrix} \tag{4.14}
$$

Of the 5000 missions simulated, there were 4404 during which the system experienced at least one failure.

Artificial neural networks (ANN) are mathematical algorithms designed to emulate the biological neuron learning process. These purely data-driven algorithms can be used for function approximation when the explicit form of the variable relationship (i.e. linear, exponential, etc.) is unknown. Artificial neural networks use a set of processing elements (or nodes) that are loosely analogous to neurons in the brain. These nodes are interconnected in a network consisting of multiple layers. The ANN identifies a pattern of connections between the nodes and uses a training algorithm to determine weights on these connections. The algorithm transitions from a random state to a final model through iterative training.

A common concern with the application of ANN is that they require large amounts of data, which is randomly partitioned into training and testing sets. In addition, there is a high learning curve associated with setting the parameters of the training algorithm. In addition to setting the network architecture, these parameters determine how quickly the network learns, the learning and transfer functions, and the number of training iterations. The typical benefits of this approach include the ability to capture data nonlinearities, discontinuities, and interactions and to accept a very large number of input and output variables.

A back-propagation ANN was developed using NeuralWorks to predict component status as a function of diagnostic test results. The back-propagation network learns by calculating the error between desired and actual output and propagating this error information back to each node in the network. This back-propagated error is used to drive the learning at each node. A variety of architecture and parameter settings were tested in this research. Based on minimum root mean squared error, the selected architecture and parameter settings of the implemented ANN are described in Table 4-1. The implemented ANN results in root mean squared errors of 0.0720 and 0.0747 respectively for the training and testing sets.

For the 2404 missions used to test the ANN, there were 5093 component failures (out of a possible 24,040 component-missions). When applied to the test set, the ANN approach resulted in 3967 false positives, 1240 true positives, 3853 false negatives and 14,980 true negatives. To provide a comparison, two additional algorithms were considered: a voting algorithm and a signal algorithm. With the voting algorithm, a component is deemed to have failed if a majority of the diagnostics applied to that component indicate failure. With the signal algorithm, a component is deemed to have failed if any of the diagnostics applied to the component indicate failure. For the same example, the voting algorithm produced 875 false positives and 18 false negatives. The signal algorithm produced 2165 false positives and 5 false negatives.

Network Setting	Description	Implemented Setting
Input layer (IL)	Layer consisting of one node per input variable	26 nodes
Output layer (OL)	Layer consisting of one node per output variable	10 nodes
Hidden layer (HL)	Single or multiple layers of nodes positioned between the input and output layers that determine the number of connections between these two layers	HL 1: 10 nodes HL 2: 10 nodes HL 3: 5 nodes
Training data set	Subset of data records (input and output observations) used to train the ANN	2000 records
Testing data set	Subset of data observations (input and output observations) used to test the ANN	2404 records
Learning rule	Rule that specifies how connection weights are changed during the learning process	Delta-rule
Learning rate	Coefficients that determine the rate of learning for each layer	IL: 0.30 HL 1: 0.25 HL 2: 0.20 HL 3: 0.15

Table 4.1 Back-propagation Network Settings

References

[1] N. A. E. Aly and A. A. Aly, "Measures of Testability for Automatic Diagnostic Systems," *IEEE Transactions on Reliability*, vol 37, no 5, 1988, pp 531-538.

[2] S. Deb, K. R. Pattipati, V. Raghavan, M. Shakeri and R. Shrestha, "Multi-Signal Flow Graphs: A Novel Approach for System Testability Analysis and Fault Diagnosis," *IEEE AES Systems Magazine*, vol 10, no 5, 1995, pp 14-25.

[3] W. G. Fenton, T. M. McGinnity and L. P. Maguire, "Fault Diagnosis of Electronic Systems Using Intelligent Techniques: A Review," *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, vol 31, no 3, 2001, pp 269-281.

[4] J. Guan and J. H. Graham, "Diagnostic Reasoning with Fault Propagation Digraph and Sequential Testing," *IEEE Transactions on Systems, Man and Cybernetics*, vol 24, no 10, 1994, pp 1552-1558.

[5] A. A. AL-Jumah and T. Arslan, "Artificial Neural Network Based Multiple Fault Diagnosis in Digital Circuits," *Proceedings ICCAS*, vol 2, 1998, pp 304-307.

[6] A. Mathur, K. F. Cavanaugh, K. R. Pattipati, P. K. Wilett and T. R. Galie, "Reasoning and Modeling Systems in Diagnosis and Prognosis," *Proceedings of the 2001 SPIE*, vol 4389, 2001, pp 194-203.

[7] J. H. Murphy and B. J. Kagle, "Neural Network Recognition of Electronic Malfunctions," *Journal of Intelligent Manufacturing*, 1992, pp 205-216.

[8] M. J. Roemer, G. J. Kacprzynski and M. H. Schoeller, "Improved Diagnostic and Prognostic Assessments Using Health Management Information Fusion," *AUTOTESTCON (Proceedings)*, 2001, pp 365-377.

[9] K. A. E. Totton and P. R. Limb, "Experience in Using Neural Networks for Electronic Diagnosis," *Proceedings of 2nd International Conference on Artificial Neural Networks*, 1991, pp 115-118.

[10] G. Vachtsevanos, "Performance Metrics for Fault Prognosis of Complex Systems," *AUTOTESTCON (Proceedings)*, 2003, pp 341-345.

[11] H. Wu, Y. Liu and Y. Ding, "A Method of Aircraft Unit Fault Diagnosis," *Aircraft Engineering and Aerospace Technology*, vol 75, no 1, 2003, pp 27-32.