The Reduction of No Fault Found Incidents Using Posterior Knowledge Integration

Dr Shaun Marriott
Robert F Harrison
Department of Automatic Control & Systems Engineering
The University of Sheffield
Mappin Street
Sheffield S1 3JD

Research Report No. 758

The financial support of Rolls-Royce plc and the EPSRC (GR/L16651) is gratefully acknowledged

Project collaborators: Dr. Peter Cowley and Dr Steve King (Rolls Royce Applied Science Laboratory, Derby)

Indexing Terms: Condition Monitoring, Fault Diagnosis/Isolation, No Fault Found Incidents, Posterior Knowledge Integration, Aircraft Engine Maintenance, Weighted Maintenance Costs, Bayesian Belief Networks.

Table of Contents

1 Overview

The final project documentation considers the problem of the integration of 'posterior knowledge' into condition monitoring systems from both the theoretical and practical points of view. The work is presented in the context of aircraft engine maintenance. Initially, the problem is framed in the context of elementary probability theory where the task of posterior knowledge representation is examined. A methodology for updating posterior condition probabilities is proposed for cases where fault conditions are rejected or retained on the basis of external knowledge supplied by an end-user—the posterior knowledge. A possible fault class ranking is generated following the specification of fault class posterior probability functions.

It is shown that a simple renormalisation of existing probabilities does not apply in the dependent condition-class case and can lead to erroneous results; the condition-class ranking may change following the exclusion of condition-classes known *not* to have occurred. An artificial example is used to illustrate the theoretical principles. Simulated fault data are then used to explore the posterior probability estimation problem through the use of radial basis function networks. A validated aircraft jet engine model is used which allows for the injection of faults (conditions). A simple, model-based range-checking methodology is applied to the data to provide a quick method of generating verified condition data for condition-class prediction and probability estimation. It is shown that the maximum possible accuracy can be achieved when the most probable fault is chosen in each case.

Context-free simulations are used to show the effect of posterior knowledge as part of a maintenance strategy. Being context-free, the simulations are independent of any specific condition-monitoring situation. Preliminary results indicate that posterior knowledge reduces the number of sub-unit inspections required for isolation of all faults. This has the potential to result in real maintenance cost savings.

One of the key objectives of this project work is to reduce the number of no fault found (NFF) incidents during aircraft maintenance. Stated simply, a NFF incident happens when a fault is indicated in a sub-unit which turns out to be a false alarm on subsequent inspection. The techniques of posterior knowledge integration (PKI) have been applied to the NFF reduction problem as detailed in this report. Simulations indicate that the application of PKI will not only reduce the overall number of sub-unit inspections during the maintenance cycle, but will also reduce the number of NFF incidents. Thus, the application of PKI may result in tangible reductions in maintenance costs.

A cost-weighted version of the context-free simulations is also presented in the final documentation. Cost weightings may be important in that maintenance strategies are governed by costs such as component accessibility, inspection costs and r information may not lead to identifying the system state as quickly but the overall cost will be lower; such considerations are important in the commercial world.

Both the unweighted and weighted PKI systems lead to an overall minimum in the number of sub-unit inspections required during maintenance as shown by the simulations. This fact can be proved theoretically. The gain is in the reduction of NFF incidents comprising sub-unit inspections over and above the theoretical and practical minimum encountered when identifying *all* faults.

Prototype PKI code is also presented and discussed together with further possible modifications and directions for research. The overall structure of the final project documentation is as follows:

Chapter 1 considers the theory underlying the integration of posterior knowledge in the context of condition monitoring. PK representation is discussed together with its integration using a posterior probability update equation. This equation governs the changes in posterior probabilities for a given fault in a given sub-unit and is determined by posterior knowledge. This subject is considered further in section 1.6 of Chapter 1 which looks at the resulting fault rankings and the effects of condition classes being included or excluded. A discussion of probability estimation and the use of radial basis function networks is also included in section 1.7; the reliance upon estimated probabilities necessitates the investigation of the probability estimation problem. Another approach, that of Bayesian belief networks (BBN's), is considered in Chapter 3 which examines the PKI problem from a different angle from that of total reliance upon empirically-derived data. Indeed, future work may stem from combining the two approaches.

Section 1.7 also introduces the turbo-jet engine. Simulation of fault conditions and the subsequent fault detection method are both discussed in this section along with some of the issues involved in fault diagnosis

Chapter 2 presents the simulation studies and considers the simulation protocol and performance assessment measures. It is shown empirically that PKI is always at least a good as, or possibly superior to, a realistic comparison method. The theoretical explanation of these results is discussed throughout this chapter.

Chapter 3 introduces Bayesian belief networks and discusses them in some detail. The networks are related to the PKI problem and pointers to future work are given. In particular, section 3.4 considers the application of BBN's in PKI.

Some conclusions are drawn in Chapter 4, together with a discussion of possible directions for future work. The prototype Matlab code is documented in Chapter 5.

2 Condition Monitoring: The Posterior Knowledge Integration Approach

2.1 Introduction

Over the past decade or so, research effort has been directed at developing methods of identifying faults or conditions in dynamical systems using statistical classifiers based upon historical data. Much effort—and possibly expense—can go into the development of such classifiers which form the basis of diagnostic systems. A set of fault-condition (FC) posterior probabilities is generated upon which diagnoses are made for example using maximum *a posteriori* (MAP) or Bayesian risk weighted decision criteria (e.g. Melsa and Cohn, 1978). In short, this is a *statistical* viewpoint on condition monitoring. When given a ranked set of FC probabilities representing the most likely FCs to have occurred, if the most probable FC is known not to have occurred then what should further decisions be based upon? Does it make sense always to choose the next most likely FC or set of FCs?

The knowledge that FCs have (or have not) occurred is deterministic, not available to the statistical classifier and is *specific* to the current situation. It cannot be made part of the historical data until the *complete* set of FCs is known for that particular input vector. Furthermore, the situation-specific data may become 'swamped' by the rest of the historical set in which it will be included. The main issue then, is the problem of integrating deterministic situation-specific data with historical, probabilistic data in a condition monitoring context. The aim, then, is to provide a rationale, which leads, in some sense, to the shortest or least costly route to the discovery of *all* faults that have occurred.

This document addresses the issue of the post-processing of statistical condition monitoring information when external evidence is available to inform the fault diagnosis process. The key objective is to devise a mechanism for the integration of such evidence into predictive systems to allow the update of FC probabilities that have been generated without reference to that knowledge. Posterior knowledge integration has potentially widespread application in the field of condition monitoring (fault detection and isolation) as explored in this document. Incorporation of deterministic situation-specific knowledge about a monitored plant, not available in developing the condition monitoring system, will facilitate a more informed choice of maintenance strategy. Such a post-processing system could augment available condition monitoring systems which generate probabilistic data following fault classification by statistical pattern recognition.

There is a growing interest in automated condition monitoring systems as the number and complexity of monitored plants increases to keep pace with the demands of modern technology. This interest is reflected in the number of fault detection and features—with or without a reference model—for pre-defined anomalies or novel operating conditions.

The so-called "classical" methods are based upon limit checking (Isermann, 1997) and involve the monitoring of measurable variables to detect pre-defined range violations. The monitoring system may initiate appropriate control actions immediately and alert the operator. Other systems may alert the operator only. Such systems are often simple and reliable (Isermann,1997) but may only be suitable for detecting relatively large changes. Furthermore, the detection is not dynamic in that changes in operating profile over time may indicate possible faults much earlier, prior to failure. Condition monitoring systems may be model-based where a reference model is used in comparison with the real plant behaviour (e.g. Trave-Massuyes and Milne, 1997; Karsai and DeCaria,1997; Milne et al, 1996; Gomm, 1994). Other systems may involve the use of rule-bases and expert systems (e.g. Wang, Lu, and McGreavy, 1997; Bogunovic, and Mesic,1996; Keravnou and Johnson,1986; Liu, Singonahalli, and Iyer, 1996; McDonald, Burt, and Moyes, 1996; Wang Xue and Yang Shuzi, 1996). Novelty detection provides another way of detecting anomalous conditions by training an artificial neural network (or other adaptive system) to recognise normal operating modes; anomalous conditions are those which deviate from the learned regions of parameter space (outliers) (Tarassenko, 1996;1997). Various types of artificial neural network have been applied to condition monitoring (e.g. Dimla, Lister and Leighton, 1997; Wilson, Irwin, and Lightbody, 1997; Boudoud and Masson, 1996; Li, Wong, and Nee, 1996; Patel, *et al*,1996; Perrott and Perryman, 1995; Zhang; Ganesan, and Sankar, 1995).

Other condition monitoring methods include the use chaos theory (correlation dimension), (e.g. Logan and Matthew, 1996), statistical methods (e.g. Weighell, Martin, and Morris, 1997; Korbicz, and Kus, 1996; Ma Yizhong, 1996; Zhang,1996), Fourier Transforms and Wavelets (e.g. Pan, Sas. and van Brussel, 1996; MacIntyre and O'Brien, 1995), nonlinear observers (e.g. Preston, Shields, and Daley, 1996; Yang and Saif,1996; Krishnaswami and Rizzoni 1994.), hybrid approaches (e.g. Hines, Miller, and Hajek,1995; Eryurek, and Upadhyaya, (1995) Lianhui Chen and Ho 1994; Ding, and Wach,1994; Isermann, 1994) analytical redundancy (e.g. Dorr, et al, 1997) and evolutionary methods (e.g. Bilchev and Parmee, 1996; Korbicz, and Kus,1995)

The emphasis of our work is the post-processing of probabilistic fault data *regardless* of the fault detection and isolation methods employed, i.e. we assume a particular set of flags has been flagged with a certain probability. Where the condition-monitoring stage is required to illustrate probability estimation, the classical range-checking detection method is used in the simulations, for the sake of simplicity The range checking method provides adequate data for the demonstration of post-detection methods of the type explored here. The CM system can be thought of as a "black box" with measurements as inputs and probabilities of conditions as outputs; this information is then used by the post-processing method presented here. In general, condition monitoring systems are confined to the actual tasks of detecting and isolating faults and alerting an end-user to their possible existence and location. These systems may or may not give probabilistic estimates of FC probabilities to allow the end-user to decide an appropriate course of action. It is clear that such a methodology is 'open-loop' in that end-users are given a final analysis, upon which to base operational decisions, without having the opportunity to feed their observations or knowledge back into the process.

What if the end user has external information (not available to the condition monitoring system) which would alter specific fault diagnoses? It is obviously desirable to maximise the use of available information. The feedback of external information to a condition monitoring process makes it a 'closed-loop' process as shown in Figure 1.

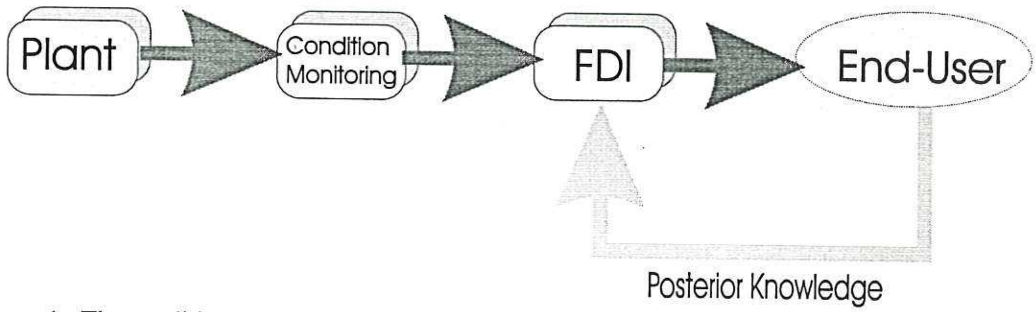


Figure 1. The condition monitoring feedback-loop. Posterior knowledge supplied by an end-user may be integrated into the condition monitoring process to improve FC isolation. The fault diagnosis and isolation (FDI) block is where the decisions are made.

In a condition monitoring situation, the end-user may say, 'The condition monitoring system indicates the possibility of faults x,y and z. I have just checked y and can discount the possibility of a fault there. How does this affect the probability of faults x and z having occurred?' The checking of y is not included in the monitored plant features and occurs after the condition monitoring system has made its predictions concerning possible fault scenarios. This external knowledge is given the name 'posterior knowledge' to distinguish it from any other knowledge about the monitored plant. Posterior knowledge is *knowledge* about the outcome supplied by an operator, or some other source, and which is not available to the predictive system at the time of prediction. It is new evidence about the posterior probabilities which have been predicted for the current classification in the form of an updated output classification and differs from the new evidence about the state of the system which is typically encountered in sequential decision theory (e.g. Melsa and Cohn, 1978), i.e. updated evidence vectors. Posterior knowledge is deterministic; it is about *known* outcomes. Subjective probabilities could be used but are not considered here. The incorporation of this knowledge into the feedback loop of condition monitoring allows fault analysis to be adjusted towards a more accurate picture of the current plant status.

The main principle behind posterior knowledge integration is the fact that dynamical systems usually consist of a set of interconnected sub-systems which are *causally related* in some way. This means that information about a particular sub-system may have an effect on the prediction probabilities of other sub-system faults mediated by the causal connection, i.e. that multiple fault scenarios may be indicated simultaneously, as a result of a single cause. This differs from the usual assumptions of exclusivity or of conditional independence made in conventional pattern recognition situations.

One particular application area for fault diagnosis and isolation methods is that of aircraft gas turbine engines (e.g. Patton and Chen, 1997; Nairac *et al*, 1997). These are complex systems comprising distinct interacting sub-units which include electronic feedback control and monitoring devices. The posterior knowledge integration problem, as considered in this document, is discussed in the context of aircraft gas turbine engine monitoring. The integration of posterior knowledge into condition monitoring systems applied to gas turbine engines is motivated by a need to reduce costly *no fault found* (NFF) conditions. NFF conditions occur when one or more faults are flagged and subsequent tests of sub-units fail to locate a problem. For example, a fault may be logged in-flight and when the plane lands at airport x, the supposedly faulty unit is replaced. The same fault is flagged during the onward flight and the unit is replaced at airport y. Subsequent analyses of both replaced units show no signs of malfunction because the alarm was triggered, perhaps, by to a faulty connection. However, the units have to be re-certified for future use which is a very expensive process. The generation of fault rankings—capable of being updated by posterior knowledge—may allow better-informed decisions about which sub-units and/or components to remove and test.

The aircraft engineering context is illustrated by the use of the Trent 700 engine model developed by Patel (Patel *et al*, 1996). This SIMULINK^{tm} model consists of the engine and accessories and is used to generate fault data for the estimation of fault probabilities as discussed in Section 1.7. The accessories include the electronic engine control (EEC) to monitor engine performance and make necessary adjustments.

2.2 Posterior knowledge integration from an engineer's point of view

A condition monitoring system will typically provide an end-user with a set of predictions indicating one or more possible FCs. Merely choosing a single FC, on the basis of its associated probabilities, may be too simplistic. Furthermore, the end-user's knowledge may come to bear on the problem as posterior knowledge, and be used to modify the original condition monitoring system diagnosis. A simple example will illustrate this (Marriott and Harrison, 1998a):

A gas turbine vibration monitoring system has detected several features that correspond to one of three conditions: 'Bearing wear in IP shaft' with probability 0.65, 'Out-of-balance in LP compressor' with probability 0.20, and 'Out-of-balance in HP compressor' with probability 0.15. However, the user knows from additional knowledge that a recent change of bearing rules out condition 'Bearing wear in IP shaft'. Is the most likely diagnosis now 'Out of balance in LP compressor'?[1]

If the above conclusions are based on *statistically dependent* probability distributions then it may not be sufficient simply to renormalise the probabilities associated with FCs '*Out-of-balance in LP compressor*' and '*Out-of-balance in HP compressor*'; this issue will be discussed further in Section 1.3. Indeed it is possible that the suggestion *'Out-of-balance in LP compressor'* is based on vibration phenomena attributed to bearing wear that also produces the out-of-balance as a side-effect. Eliminating

[1] Suggested by Dr. Steven King of Rolls-Royce plc, Applied Science Laboratory, Derby

bearing wear as a possible diagnosis could remove the possibility of the LP out-of-balance. The engineer may, therefore, conclude that the correct diagnosis is '*Out-of-balance in the HP compressor*'. This example illustrates some of the issues concerning the manner in which this posterior knowledge can be incorporated by the system for re-evaluation and future reference.

2.3 Posterior knowledge: representation and integration

It has been stated that posterior knowledge is deterministic knowledge external to the condition monitoring system. Two questions naturally arise from this: how can posterior knowledge be quantified and how is it to be integrated with the information contained within the condition monitoring system based upon the key plant features? This document explores these two questions and then presents simulation evidence to show that posterior knowledge integration represents a technique with potential application in the condition-monitoring field. As stated in the introduction, the key objective is to develop a method of automating the knowledge integration and updating process that follows logically from the fault detection and isolation tasks.

There are many possible ways of representing posterior knowledge. The representation problem is solved here by representing the posterior knowledge of possible system states and associated FCs as FC probabilities with discrete values of 1 or 0 depending upon whether a FC is known to occurred or not. Thus, although the posterior knowledge is deterministic, it is represented as a new set of FC probabilities, that is, a revised probability for each class that is influenced by external observations of the current situation only. The externally obtained information is then used to update the predicted FC ranking for the remaining probabilistic FCs. In other words, posterior knowledge about an FC is represented in the form of a probability indicating the occurrence or non-occurrence of that FC. In this way, deterministic data has been represented within a probabilistic framework. Note that this is not to be confused with the *posterior probability* of a fault occurring. For example, a set of class posterior probabilities will be predicted for a single input datum (feature vector). If it is then possible to *exclude* one or more classes (i.e. the probability of those FCs occurring is zero) on the basis of knowledge or reasoning not available to the predictive system, then the current list of predicted FC probabilities may be revised. This will give a more accurate estimation of new FC *posterior* probabilities in the form of a revised probabilistic ranking. Similarly for the *inclusion* of known FCs.

Here, the integration of posterior knowledge is given in terms of classes that are known *not to have occurred* (excluded) or are known *to have occurred* (included) as indicated by the external knowledge. It is convenient to represent the updated posterior probabilities in terms of probabilities estimated from previous observations of system FCs, i.e. classes that *have* occurred; these probabilities are probabilities of *occurrence* and they can be estimated from empirical fault data. The limitation of probabilities involved in the representation of posterior knowledge to the class of null and certain events is a design choice made to facilitate implementation of the posterior knowledge integration process. This does not preclude the generalisation of the

1. the probability associated with the posterior knowledge that a FC has or has not occurred i.e. 1 or 0 representing the deterministic knowledge,
2. the exclusive (i.e. non-overlapping) singleton and joint probabilities generated by the condition monitoring system or FC frequencies of occurrence,
3. the (possibly) overlapping posterior probabilities of FCs reconstructed from the condition monitoring probabilities of (2).

2.3.1 Conditional Probabilities: a Basis for PKI

Posterior knowledge integration is based upon conditional probability. The probability of a sub-unit failure is conditioned upon alarm readings and the failure or correct operation of other sub-units. The latter relationship exists unless the sub-units are totally independent of each other. It will be shown that the PKI method is superior to the baseline comparison method unless all sub-units are independent; in this case, the performances are equal.

This section (1.3) will introduce the concepts underlying PKI. The basic theory will be developed and illustrated by examples to provide a justification for the *posterior probability update equation* (PPUE) which governs the integration of posterior knowledge. Issues related to the use of the PPUE will also be covered.

The *input space* or *sample space* of a condition monitoring situation (e.g. Grimmet and Stirzaker,1992) may be divided into *N*, possibly overlapping, classes given by $U_f = C_1 \cup C_2,\ldots,\cup C_N$. This space is exhaustive and contains all possible outcomes or conditions A four class example is represented by a Venn diagram in Figure 2.

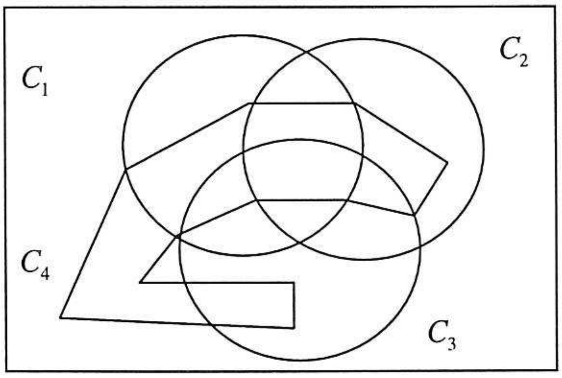


Figure 2. Abstract representation of a four class problem showing a number of overlapped regions. Note that some of the possible regions of overlap may contain no members and, thus, would not exist.

Here, the universal set, U is given by $U = U_f \cup U_n$ where U_n signifies normal or non-faulty operation. The remainder of the universal set may be taken to represent the normal operation. Venn diagrams provide a useful way of representing the probabilities involved in updating class predictions. Figure 3 represents schematically the probability of class 1 faults occurring in a four-class problem.

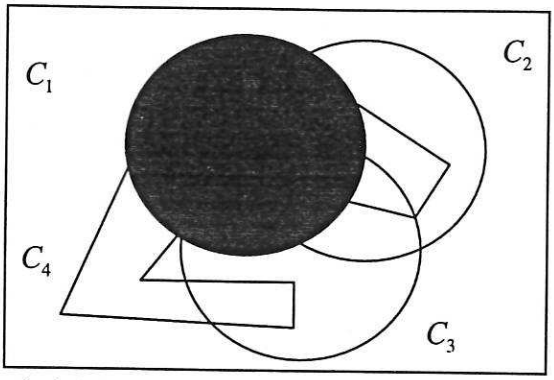


Figure 3. The shaded circle represents the total probability of class 1 occurring.

The representation of probabilities by Venn diagrams can be justified by appealing to the frequentist interpretation of probability (e.g. Kneale, 1949). Letting n_i represent the number of elements in set i (class i) and N represent the total number of elements, then $P(C_i)$ can be defined as

$$P(C_i) = \lim_{N \to \infty} \frac{n_i}{N}.$$

From this definition (e.g. Durrett, 1994) many other expressions involving the union and conjunction of sets and their respective probability definitions can be derived.

Fault classes may be:

i) independent
ii) dependent and exclusive
iii) dependent and non-exclusive

These three cases will be examined within this document in the context of PKI. Classes (i) and (ii) are special cases of PKI whilst class (iii) is the most informative as PKI utilises joint information.

Independence in this context is taken to be *conditional independence* (Bernardo and Smith, 1994; Grimmet and Stirzaker, 1992). More detail is given in the relevant parts of later sections.

In essence, the posterior knowledge integration process involves the modification of the FC posterior probabilities by the exclusion or inclusion of FC frequencies based upon the known occurrence or non-occurrence of FCs. Both the inclusion and exclusion of FCs by posterior knowledge integration will be dealt with here. The revised posterior probabilities include the integration of external knowledge explicitly in the notation. The revised posterior FC probabilities, given some posterior knowledge, are represented by the expression $P(C_p | \varepsilon \cap \mathbf{x})$ where the posterior knowledge is given by $\varepsilon = \left(\bigcap_i C_{\delta_i} \right) \cap \left(\bigcap_e C^c_{\delta_e} \right)$ and $\boldsymbol{x}$ is the feature vector. The subscripts δ_i and δ_e represent the indices of the sets of FC given FC, $C^c_{\delta_e}$, is the complement with respect to all other classes (including the normal operation class). The set of FC occurrences and the set of non-occurrences are known as the *included* and *excluded* classes respectively. The expression $P\left(C_p \middle| \varepsilon \cap \mathbf{x}\right)$ denotes the predicted probability that FC p has occurred given the observation vector, x, and the posterior knowledge, ε. The following discussion will illustrate the motivation behind this representation of PKI.

2.3.2 Excluded Faults: a Special Case

When faults are known not to occur in sub-units (normal operation), the FCs are *excluded*. That is, a sub-unit may be known not to be faulty and so it can be excluded from the search for faults. This is a special case of the general PPUE and will be used to illustrate the PKI process. Included (known) faults will be discussed later. Without loss of generality, the conditioning on the reading vector will be ignored in the early stages of the discussion.

Using the definition of conditional probability for discrete events (e.g. Durrett, 1994, Grimmet and Stirzaker, 1992)

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \quad P(B) > 0$$

where event A is conditional upon event B, the integration of posterior knowledge in the form of excluded classes, that is where a given system condition is known not to have occurred, can be represented by statements such as

$$P(C_1|C_2^c) = \frac{P(C_1 \cap C_2^c)}{P(C_2^c)} \quad \text{and} \quad P(C_1|C_2^c \cap C_4^c) = \frac{P(C_1 \cap C_2^c \cap C_4^c)}{P(C_2^c \cap C_4^c)}$$

the superscripted c indicates the complement operation with respect to the *universal set*, (Grimmet and Stirzaker, 1992) thus C_2^c and C_4^c signify that fault classes two and four respectively have been excluded; this constitutes the new knowledge that those classes are now known not to have occurred. The revised probabilities $P(C_1|C_2^c)$ and $P(C_1|C_2^c \cap C_4^c)$ now represent the state of knowledge regarding the occurrence of FC 1, after external knowledge has been incorporated, in the form of revised posterior probabilities. In other words, it is known that $P(C_2) = 0$, or $P(C_2) = 0$ and $P(C_4) = 0$, respectively.

Formally, the revised posterior probabilities require that the inclusion of external knowledge be explicitly included in the notation e.g. $P(C_1|\varepsilon)$, with $\varepsilon = C_2^c$ and $P(C_1|\varepsilon)$ $\varepsilon = C_2^c \cap C_4^c$ respectively, where the symbol 'ε' denotes the external knowledge or evidence. In later sections, ε_n is used to denote the sequential nature of PKI but this refinement is not important here. So $\varepsilon = C_2^c$ and $\varepsilon = C_2^c \cap C_4^c$ respectively in the above examples.

Here, probabilities are required of the form $P\left(C_i | C_j^c\right)$ and $P\left(C_i | C_j^c \cap C_k^c\right)$. The general form given by $P\left(C_i \middle| \bigcap_{k \in \Delta_e} C_k^c\right)$ where Δ_e denotes the set of indices of the excluded FCs; the exclusion being based upon external knowledge. Here $\varepsilon = \bigcap_{k \in \Delta_e} C_k^c$ and the inclusion of posterior knowledge is given in terms of classes which are known *not to have occurred* as indicated by the external knowledge. It is convenient to represent the updated posterior probabilities in terms of probabilities estimated from previous observations of system conditions, i.e. classes which have occurred; these probabilities we call *positive* probabilities and they can be estimated from empirical data.

The probability of class 1 occurring given three possible classes and given the *posterior* information that class 2 has not occurred is denoted by

$$P\left(C_1 | C_2^c\right) = \frac{P\left(C_1 \cap C_2^c\right)}{P\left(C_2^c\right)} = \frac{P\left(C_1 \cup C_2\right) - P\left(C_2\right)}{P\left(C_1 \cup C_2 \cup C_3\right) - P\left(C_2\right)}$$

using the definition of conditional probability and the identity
$P\left(A \cap B^c\right) \equiv P(A \cup B) - P(B)$

This situation is shown schematically in Figure 4.

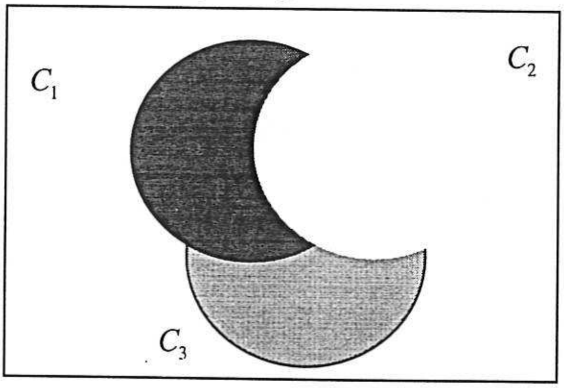


Figure 4. The diagrammatic representation of $P\left(C_1 | C_2^c\right)$ for three dependent classes. It is the probability of the remainder of C_1 (without C_2) divided by the probability of C_1 and C_3 combined (without C_2).

The probability of class 1 occurring given a total of four possible classes and given the posterior information that class 2 has not occurred is also denoted by

$$P\left(C_1 | C_2^c\right) = \frac{P\left(C_1 \cap C_2^c\right)}{P\left(C_2^c\right)}.$$

This is illustrated in Figure 5.

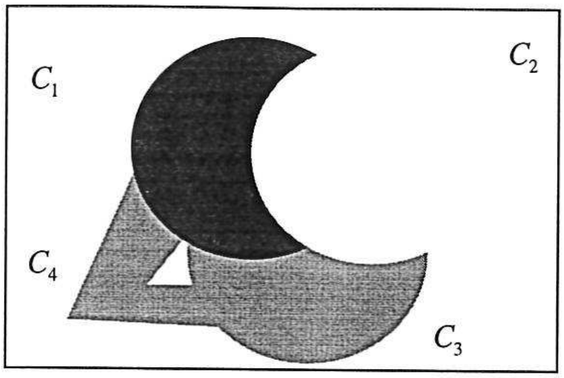


Figure 5. The diagrammatic representation of $P(C_1|C_2^c)$ for four dependent classes. The probability of the remainder of class 1 occurring is divided by the total remaining probability excluding class 2 to give the remaining probability of class 1 occurring.

For dependent FCs, the underlying general pattern appears to consist of finding the difference of unions of those sets involved in the numerator and denominator of the conditional probability expression and finding the ratio of the respective probabilities. For example, for the four class problem, $P(C_1|C_2^c)$ can be written as

$$P(C_1|C_2^c) = \frac{P(C_1 \cap C_2^c)}{P(C_2^c)} = \frac{P(C_1 \cup C_2) - P(C_2)}{P\left(\bigcup_{l=1}^{4} C_l\right) - P(C_2)}$$

where the posterior information has been included in terms of *positive* probabilities, that is, those which can be estimated directly, or computed from estimated probabilities. Similarly, the inclusion of further posterior information that class C_4 has also been excluded can be written as

$$P(C_1|C_2^c \cap C_4^c) = \frac{P(C_1 \cap C_2^c \cap C_4^c)}{P(C_2^c \cap C_4^c)} = \frac{P(C_1 \cup C_2 \cup C_4) - P(C_2 \cup C_4)}{P\left(\bigcup_{l=1}^{4} C_l\right) - P(C_2 \cup C_4)}$$

Here, the class unions have increased by a single member (FC 4) which is to be excluded to give the revised probabilities. This is shown in Figure 6.

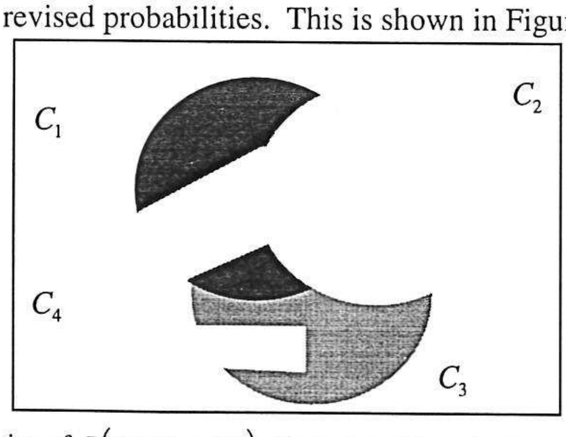


Figure 6. A representation of $P(C_1|C_2^c \cap C_4^c)$ where class 4 has also been excluded from Figure 5.

The set of fault classes is taken to be exhaustive with the universal set given by the union of all classes including a "no-fault" class.

2.3.3 The Generalised Form of Posterior Knowledge integration Where One Or More Dependent Classes Are Excluded.

For the more general case, where a *set* of dependent fault classes is *excluded*, the following notation is introduced:, Δ_r and Δ_ε denote the index sets of remaining and excluded sets respectively where $\Delta_r = \{\delta_1, \delta_2, \ldots, \delta_{Nr}\}$ and $\Delta_\varepsilon = \{\delta_{Nr+1}, \delta_{Nr+2}, \ldots, \delta_N\}$ N_r is the number of remaining classes, N is the total number of possible classes. The delta notation is used to denote that the class indices are not necessarily selected on the basis of ordering e.g. it could be that for a five class problem, $\Delta_r = \{1,3,5\}$ and $\Delta_\varepsilon = \{2,4\}$ in which case $\delta_1 = 1$, $\delta_2 = 3$, $\delta_3 = 5$, $\delta_4 = 2$ and $\delta_5 = 4$ where classes two and four have been excluded.

In general, to calculate the updated probabilities, given *posterior* knowledge on excluded classes,

$$P\left(C_{\delta_i} | C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N}\right) = \frac{P\left(C_{\delta_i} \cap C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N}\right)}{P\left(C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N}\right)}$$

$$= \frac{P\left(\bigcup_j C_{\delta_j}\right) - P\left(\bigcup_k C_{\delta_k}\right)}{P\left(\bigcup_l C_{\delta_l}\right) - P\left(\bigcup_k C_{\delta_k}\right)} \qquad (1)$$

where δ_i is the i th index, $\delta_i \in \{1,2,\ldots,N\}$ signifying the fault class under investigation, $j \in \{\delta_i\} \cup \Delta_\varepsilon$, is the union of this class with the excluded set, $k \in \Delta_\varepsilon$, signifies the excluded set and $l \in \Delta_r \cup \Delta_\varepsilon$ is the totality of fault classes.
For the numerator, the probability of the union of all classes omitted is subtracted from the probability of this union augmented by the class of interest i.e. $P\left(C_{\delta_i} \cup \left(\bigcup_k C_{\delta_k}\right)\right) - P\left(\bigcup_k C_{\delta_k}\right)$. For the denominator, the probability of the union of all excluded classes is subtracted from the probability of the total number of classes i.e. the certain event. This calculation suggests that an incremental procedure is possible . Indeed, the simulations discussed in this document use incremental PKI. The proof of equation (1) is deferred until sub-section 1.3.7 when the dependence of the class on the data is included. A general version that also allows included fault class knowledge to be integrated is discussed in Section 1.4. That is, faults known to have occurred are included in the PKI process.

2.3.4 Union of Overlapping Sets.

Now that the general form of the updated posterior probabilities $P\left(C_{\delta_i} | C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N}\right)$ is given in terms of unions of sets, the general form of $P\left(\bigcup_{s=1}^{K} C_{\delta_s}\right)$ is required (e.g Durrett, 1994,Grimmet and Stirzaker, 1992) where K is the number of sets involved in the union. This is expanded to give

$$
\begin{aligned}
P\left(\bigcup_{s=1}^{K} C_{\delta_s}\right) = & \sum_{i=1}^{K} P\left(C_{\delta_i}\right) \\
& - \sum_{i<j}^{K} P\left(C_{\delta_i} \cap C_{\delta_j}\right) \\
& + \sum_{i<j<k}^{K} P\left(C_{\delta_i} \cap C_{\delta_j} \cap C_{\delta_k}\right) \qquad (2) \\
& \vdots \\
& + (-1)^{K+1} P\left(C_{\delta_1} \cap C_{\delta_2} \cap \ldots \cap C_{\delta_K}\right)
\end{aligned}
$$

in terms of positive probabilities. Recall that the δ notation is used to reflect the fact that the indices *i,j,k*... are not necessarily consecutive or ordered sequentially. For example, $\delta_1 = 1$, $\delta_2 = 3$, etc as above.

2.3.5 Examples of Posterior Knowledge Inclusion in Terms of Positive Probabilities.

2.3.5.1 For Three Classes where a Single Class has Been Excluded:

$$
\begin{aligned}
P\left(C_1 | C_2^c\right) &= \frac{P\left(C_1 \cap C_2^c\right)}{P\left(C_2^c\right)} = \frac{P\left(C_1 \cup C_2\right) - P\left(C_2\right)}{P\left(C_1 \cup C_2 \cup C_3\right) - P\left(C_2\right)} \\
&= \frac{P\left(C_1\right) - P\left(C_1 \cap C_2\right)}{P\left(C_1\right) + P\left(C_3\right) - P\left(C_1 \cap C_2\right) - P\left(C_1 \cap C_3\right) - P\left(C_2 \cap C_3\right) + P\left(C_1 \cap C_2 \cap C_3\right)}
\end{aligned}
$$

2.3.5.2 For four classes where a Single Class has Been Excluded:

$$
\begin{aligned}
P\left(C_1 | C_2^c\right) &= \frac{P\left(C_1 \cap C_2^c\right)}{P\left(C_2^c\right)} = \frac{P\left(C_1 \cup C_2\right) - P\left(C_2\right)}{P\left(C_1 \cup C_2 \cup C_3 \cup C_4\right) - P\left(C_2\right)} \\
&= P\left(C_1\right) - P\left(C_1 \cap C_2\right) \Big/ \left(
\begin{aligned}
& P\left(C_1\right) + P\left(C_3\right) + P\left(C_4\right) \\
& - P\left(C_1 \cap C_2\right) - P\left(C_1 \cap C_3\right) - P\left(C_1 \cap C_4\right) \\
& - P\left(C_2 \cap C_3\right) - P\left(C_2 \cap C_4\right) - P\left(C_2 \cap C_4\right) \\
& + P\left(C_1 \cap C_2 \cap C_3\right) + P\left(C_1 \cap C_2 \cap C_4\right) \\
& + P\left(C_1 \cap C_3 \cap C_4\right) + P\left(C_2 \cap C_3 \cap C_4\right) \\
& - P\left(C_1 \cap C_2 \cap C_3 \cap C_4\right)
\end{aligned}
\right)
\end{aligned}
$$

2.3.5.3 The Exclusion of *two* Classes in the Four class Example:

$$
\begin{aligned}
P\left(C_1 | C_2^c \cap C_4^c\right) &= \frac{P\left(C_1 \cap C_2^c \cap C_4^c\right)}{P\left(C_2^c \cap C_4^c\right)} \\
&= \frac{P\left(C_1 \cup C_2 \cup C_4\right) - P\left(C_2 \cup C_4\right)}{P\left(C_1 \cup C_2 \cup C_3 \cup C_4\right) - P\left(C_2 \cup C_4\right)}
\end{aligned}
$$

$$
= \frac{\begin{array}{l} P(C_1) - P(C_1 \cap C_2) \\ - P(C_1 \cap C_4) + P(C_1 \cap C_2 \cap C_4) \end{array}}{\begin{array}{l} P(C_1) + P(C_3) \\ - P(C_1 \cap C_2) - P(C_1 \cap C_3) - P(C_1 \cap C_4) \\ - P(C_2 \cap C_3) - P(C_3 \cap C_4) \\ + P(C_1 \cap C_2 \cap C_3) + P(C_1 \cap C_2 \cap C_4) \\ + P(C_1 \cap C_3 \cap C_4) + P(C_2 \cap C_3 \cap C_4) \\ - P(C_1 \cap C_2 \cap C_3 \cap C_4) \end{array}}
$$

2.3.6 All Probabilities are Conditional Upon the Input Space

So far, the dependence of the classes on the underlying sample space has been omitted. In actuality, these classes are labels attached to regions of the space. Denoting the underlying data variable by the n-dimensional vector $\mathbf{x}$, and assuming for now that the elements of $\mathbf{x}$ are discrete random variables, the expression that the updated class probabilities including posterior information can be stated as

$$
\begin{aligned}
P\left(C_{\delta_i} | C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N} \cap \mathbf{x}\right) &= \frac{P\left(C_{\delta_i} \cap C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N} \cap \mathbf{x}\right)}{P\left(C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N} \cap \mathbf{x}\right)} \\
&= \frac{P\left(\left(\bigcup_j C_{\delta_j}\right) \cap \mathbf{x}\right) - P\left(\left(\bigcup_k C_{\delta_k}\right) \cap \mathbf{x}\right)}{P\left(\left(\bigcup_l C_{\delta_l}\right) \cap \mathbf{x}\right) - P\left(\left(\bigcup_k C_{\delta_k}\right) \cap \mathbf{x}\right)} \qquad (3)
\end{aligned}
$$

where $j \in \{\delta_i\} \cup \Delta_\varepsilon$, $k \in \Delta_\varepsilon$, and $l \in \Delta_r \cup \Delta_\varepsilon$. From equation (3) we get

$$P\left(C_{\delta_i} | C^c_{\delta_{Nr+1}} \cap C^c_{\delta_{Nr+2}} \cap \ldots \cap C^c_{\delta_N} \cap \mathbf{x}\right) = \frac{P\left(\left(\bigcup_j C_{\delta_j}\right) \cap \mathbf{x}\right) - P\left(\left(\bigcup_k C_{\delta_k}\right) \cap \mathbf{x}\right)}{P\left(\left(\bigcup_l C_{\delta_l}\right) \cap \mathbf{x}\right) - P\left(\left(\bigcup_k C_{\delta_k}\right) \cap \mathbf{x}\right)}$$

$$= \frac{P\left(\bigcup_j C_{\delta_j} \middle| \mathbf{x}\right) P(\mathbf{x}) - P\left(\bigcup_k C_{\delta_k} \middle| \mathbf{x}\right) P(\mathbf{x})}{P\left(\bigcup_l C_{\delta_l} \middle| \mathbf{x}\right) P(\mathbf{x}) - P\left(\bigcup_k C_{\delta_k} \middle| \mathbf{x}\right) P(\mathbf{x})} = \frac{P\left(\bigcup_j C_{\delta_j} \middle| \mathbf{x}\right) - P\left(\bigcup_k C_{\delta_k} \middle| \mathbf{x}\right)}{P\left(\bigcup_l C_{\delta_l} \middle| \mathbf{x}\right) - P\left(\bigcup_k C_{\delta_k} \middle| \mathbf{x}\right)}$$

where the rule $P(A \cap B) = P(A|B)P(B)$ has been applied.
The above expression reduces to

$$\frac{P\left(\mathbf{x} \cap \bigcup_j C_{\delta_j}\right) - P\left(\mathbf{x} \cap \bigcup_k C_{\delta_k}\right)}{P(\mathbf{x}) - P\left(\mathbf{x} \cap \bigcup_k C_{\delta_k}\right)}$$

if *all* input vectors have class labels associated with them i.e. the set of classes is exhaustive. The final expression for the revised probabilities is given by:

$$P\left(C_{\delta_i} \middle| \bigcap_k C^c_{\delta_k} \cap \mathbf{x}\right) = \frac{P\left(\left(\bigcup_j C_{\delta_j}\right) \middle| \mathbf{x}\right) - P\left(\left(\bigcup_k C_{\delta_k}\right) \middle| \mathbf{x}\right)}{P\left(\left(\bigcup_l C_{\delta_l}\right) \middle| \mathbf{x}\right) - P\left(\left(\bigcup_k C_{\delta_k}\right) \middle| \mathbf{x}\right)} \qquad (4)$$

Equation (4) can be proved formally as follows:

$$P\left(C_{\delta_i} \middle| \left(\bigcap_k C^c_{\delta_k}\right) \cap \mathbf{x}\right) = \frac{P\left(C_{\delta_i} \cap \left(\bigcap_k C^c_{\delta_k}\right) \cap \mathbf{x}\right)}{P\left(\left(\bigcap_k C^c_{\delta_k}\right) \cap \mathbf{x}\right)}$$ by definition of conditional probability.

$$= \frac{P\left(C_{\delta_i} \cap \left(\bigcap_k C^c_{\delta_k}\right) \middle| \mathbf{x}\right) P(\mathbf{x})}{P\left(\left(\bigcap_k C^c_{\delta_k}\right) \middle| \mathbf{x}\right) P(\mathbf{x})}$$ by $P(A \cap B) = P(A|B)P(B)$

$$= \frac{P\left(C_{\delta_i} \cap \left(\bigcap_k C^c_{\delta_k}\right) \middle| \mathbf{x}\right)}{P\left(\left(\bigcap_k C^c_{\delta_k}\right) \middle| \mathbf{x}\right)}$$ by cancellation of $P(\mathbf{x})$

$$= \frac{P\left({}^{c}C_{\delta_i} \cap \left[\left(\bigcap_k C_{\delta_k}^c \right)^c \right] \middle| \mathbf{x} \right)}{P\left(U \cap \left[\left(\bigcap_k C_{\delta_k}^c \right)^c \right]^c \middle| \mathbf{x} \right)} \text{ by } (A^c)^c = A \text{ and } U \cap A = A$$

$$= \frac{P\left(C_{\delta_i} \cup \left(\bigcap_k C_{\delta_k}^c \right)^c \middle| \mathbf{x} \right) - P\left(\left(\bigcap_k C_{\delta_k}^c \right)^c \middle| \mathbf{x} \right)}{P\left(U \cup \left(\bigcap_k C_{\delta_k}^c \right)^c \middle| \mathbf{x} \right) - P\left({}^{c}\left(\bigcap_k C_{\delta_k}^c \right) \middle| \mathbf{x} \right)} \text{ by } P(A \cap B^c) = P(A \cup B) - P(B)$$

$$= \frac{P\left(C_{\delta_i} \cup \left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right) - P\left(\left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right)}{P\left(U \cup \left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right) - P\left(\left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right)} \text{ by de Morgan's law(e.g. Applebaum, 1996)}$$

$$= \frac{P\left(C_{\delta_i} \cup \left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right) - P\left(\left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right)}{P(U|\mathbf{x}) - P\left(\left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right)}$$

$$= \frac{P\left(\left(\bigcup_j C_{\delta_j} \right) \middle| \mathbf{x} \right) - P\left(\left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right)}{P\left(\left(\bigcup_l C_{\delta_l} \right) \middle| \mathbf{x} \right) - P\left(\left(\bigcup_k C_{\delta_k} \right) \middle| \mathbf{x} \right)}$$

where the fact that the union of the classes is exhaustive has been used.

If $\boldsymbol{x}$ is real-valued (continuous) then appropriate probability density functions of the form

$$p\left(\mathbf{x} \cap \left(\bigcup_j C_{\delta_j} \right) \right) = P\left(\left(\bigcup_j C_{\delta_j} \right) \middle| \mathbf{x} \right) p(\mathbf{x})$$

are substituted into the previous argument where $p(.)$ denotes a probability *density* function.

After including the dependency upon $\boldsymbol{x}$, equation (2) is now written in terms of probabilities conditional upon the input

$$P\left(\bigcup_{s=1}^{K} C_{\delta_s} \middle| \mathbf{x}\right) = \sum_{i=1}^{K} P\left(C_{\delta_i} \middle| \mathbf{x}\right)$$

$$- \sum_{i<j}^{K} P\left(C_{\delta_i} \cap C_{\delta_j} \middle| \mathbf{x}\right)$$

$$+ \sum_{i<j<k}^{K} P\left(C_{\delta_i} \cap C_{\delta_j} \cap C_{\delta_k} \middle| \mathbf{x}\right) \qquad (5)$$

$$\vdots$$

$$+ (-1)^{K+1} P\left(C_{\delta_1} \cap C_{\delta_2} \cap \ldots \cap C_{\delta_K} \middle| \mathbf{x}\right)$$

to include the conditional probabilities of equation (4). Equation (5) can be proved easily by using the distributivity of set relations and substituting $C_{\delta_i} \cap \mathbf{x}$ for C_{δ_i} in equation (2).

2.3.6.1 A Simple Example in Terms of Relative Frequencies.

In principle, the probability estimation problem can be solved by counting the number of occurrences of classes on a case-by-case basis. Here, the probabilities are represented by relative frequency observations. In practice, this method requires a large amount of data (Bishop, 1995) and may not be accurate thus necessitating the use of a continuous (i.e. not discrete-valued) estimator. The example presented here is intended to illustrate the principles discussed in the previous section in a simple context.

The probabilities of single classes are approximated by

$$P(C_i|\mathbf{x}) = \frac{P(C_i \cap \mathbf{x})}{P(\mathbf{x})}$$

$$= \frac{n(C_i \cap \mathbf{x}) / N(\mathbf{x})}{n(\mathbf{x}) / N(\mathbf{x})}$$

$$= \frac{n(C_i \cap \mathbf{x})}{n(\mathbf{x})}$$

where $N(\mathbf{x})$ is the total number of condition occurrences within a given region of the data space $\boldsymbol{x}$. Similarly for two classes occurring simultaneously

$$P(C_i \cap C_j|\mathbf{x}) = \frac{P(C_i \cap C_j \cap \mathbf{x})}{P(\mathbf{x})}$$

$$= \frac{n(C_i \cap C_j \cap \mathbf{x})}{n(\mathbf{x})}$$

or for all classes:

$$P\left(C_1 \cap C_2 \cap \ldots \cap C_N | \mathbf{x}\right) = \frac{P\left(C_1 \cap C_2 \cap \ldots \cap C_N \cap C_{\mathbf{x}}\right)}{P(\mathbf{x})}$$

$$= \frac{n\left(C_1 \cap C_2 \cap \ldots \cap C_N \cap \mathbf{x}\right)}{n(\mathbf{x})}$$

and are associated with a reading vector, **x**.

A numerical example of overlapping classes with all elements associated with x is shown in Figure 7.

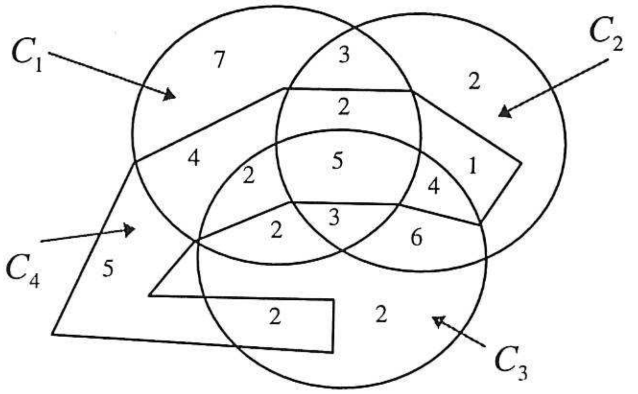


Figure 7. A diagrammatic representation of the numerical example.

The class frequency counts are given by

$$n\left(C_1 \cap \mathbf{x}\right) = 28,\ n\left(C_2 \cap \mathbf{x}\right) = 26,\ n\left(C_3 \cap \mathbf{x}\right) = 26,\ n\left(C_4 \cap \mathbf{x}\right) = 25,$$

$$n\left(C_1 \cap C_2 \cap \mathbf{x}\right) = 13,\ n\left(C_1 \cap C_3 \cap \mathbf{x}\right) = 12,\ n\left(C_1 \cap C_4 \cap \mathbf{x}\right) = 13,$$
$$n\left(C_2 \cap C_3 \cap \mathbf{x}\right) = 18,\ n\left(C_2 \cap C_4 \cap \mathbf{x}\right) = 12,\ n\left(C_3 \cap C_4 \cap \mathbf{x}\right) = 13,$$

$$n\left(C_1 \cap C_2 \cap C_3 \cap \mathbf{x}\right) = 8,\ n\left(C_1 \cap C_2 \cap C_4 \cap \mathbf{x}\right) = 7,\ n\left(C_1 \cap C_3 \cap C_4 \cap \mathbf{x}\right) = 7,$$
$$n\left(C_2 \cap C_3 \cap C_4 \cap \mathbf{x}\right) = 9,$$

$$n\left(C_1 \cap C_2 \cap C_3 \cap C_4 \cap \mathbf{x}\right) = 5$$

The total number of condition occurrences across the four classes is given by

$$N(\mathbf{x}) = 28 + 26 + 26 + 25 - 13 - 12 - 13 - 18 - 12 - 13 + 8 + 7 + 7 + 9 - 5$$
$$= 105 - 81 + 31 - 5$$
$$= 50$$

Now the relative frequencies (probabilities) of the *singleton* classes (the total probability of each class as a whole) can be calculated by

$$P\left(C_1 | \mathbf{x}\right) = \frac{n\left(C_1 \cap \mathbf{x}\right)}{N(\mathbf{x})} = \frac{28}{50} = 0.56$$

Similarly,

$$P\left(C_2 | \mathbf{x}\right) = \frac{26}{50} = 0.52,\ P\left(C_3 | \mathbf{x}\right) = \frac{26}{50} = 0.52,\ \text{and}\ P\left(C_4 | \mathbf{x}\right) = \frac{25}{50} = 0.5$$

Note that $\sum_{i=1}^{4} P(C_i) \neq 1$ because the classes are not exclusive, instead, the union of the four classes $P\left(\bigcup_{i=1}^{4} C_i\right) = 1$.

The class pairs are given by

$$P(C_1 \cap C_2 | \mathbf{x}) = \frac{n(C_1 \cap C_2 \cap \mathbf{x})}{N(\mathbf{x})} = \frac{13}{50} = 0.26$$

$$P(C_1 \cap C_3 | \mathbf{x}) = \frac{12}{50} = 0.24, \; P(C_1 \cap C_4 | \mathbf{x}) = \frac{13}{50} = 0.26, \; P(C_2 \cap C_3) = \frac{18}{50} = 0.36$$

$$P(C_2 \cap C_4 | \mathbf{x}) = \frac{12}{50} = 0.24, \; P(C_3 \cap C_4 | \mathbf{x}) = \frac{13}{50} = 0.26$$

the class triples by

$$P(C_1 \cap C_2 \cap C_3 | \mathbf{x}) = \frac{n(C_1 \cap C_2 \cap C_3 \cap \mathbf{x})}{N(\mathbf{x})} = \frac{8}{50} = 0.16$$

$$P(C_1 \cap C_2 \cap C_4 | \mathbf{x}) = \frac{7}{50} = 0.14, \; P(C_1 \cap C_3 \cap C_4) = \frac{7}{50} = 0.14$$

$$P(C_2 \cap C_3 \cap C_4 | \mathbf{x}) = \frac{9}{50} = 0.18$$

and finally, $P(C_1 \cap C_2 \cap C_3 \cap C_4 | \mathbf{x}) = \frac{n(C_1 \cap C_2 \cap C_3 \cap C_4 \cap \mathbf{x})}{N(\mathbf{x})} = \frac{5}{50} = 0.1$.

The revised probabilities given the *posterior* knowledge, C_2^c, (i.e. not class 2 in this case) are given by

$$P(C_1 | C_2^c \cap \mathbf{x}) = \frac{P(C_1 \cup C_2 | \mathbf{x}) - P(C_2 | \mathbf{x})}{1 - P(C_2 | \mathbf{x})} = \frac{P(C_1 | \mathbf{x}) - P(C_1 \cap C_2 | \mathbf{x})}{1 - P(C_2 | \mathbf{x})}$$

$$= \frac{\frac{28}{50} - \frac{13}{50}}{\frac{50}{50} - \frac{26}{50}} = \frac{15}{24} = 0.625$$

Similarly,

$$P(C_3 | C_2^c \cap \mathbf{x}) = \frac{\frac{26}{50} - \frac{18}{50}}{\frac{50}{50} - \frac{26}{50}} = \frac{8}{24} = 0.333$$

$$P(C_4 | C_2^c \cap \mathbf{x}) = \frac{\frac{26}{50} - \frac{13}{50}}{\frac{50}{50} - \frac{26}{50}} = \frac{13}{24} = 0.5417$$

If class 4 is also excluded, the revised probabilities become

$$P\left(C_1 | C_2^c \cap C_4^c \cap \mathbf{x}\right) = \frac{P\left(C_1 \cap C_2^c \cap C_4^c \cap \mathbf{x}\right)}{P\left(C_2^c \cap C_4^c\right)}$$

$$= \frac{P\left(C_1 \cup C_2 \cup C_4 | \mathbf{x}\right) - P\left(C_2 \cup C_4 | \mathbf{x}\right)}{P\left(C_1 \cup C_2 \cup C_3 \cup C_4 | \mathbf{x}\right) - P\left(C_2 \cup C_4 | \mathbf{x}\right)}$$

$$= \frac{\begin{array}{l} P(C_1|\mathbf{x}) - P(C_1 \cap C_2|\mathbf{x}) \\ -P(C_1 \cap C_4|\mathbf{x}) + P(C_1 \cap C_2 \cap C_4|\mathbf{x}) \end{array}}{\begin{array}{l} P(C_1|\mathbf{x}) + P(C_3|\mathbf{x}) \\ -P(C_1 \cap C_2|\mathbf{x}) - P(C_1 \cap C_3|\mathbf{x}) - P(C_1 \cap C_4|\mathbf{x}) \\ -P(C_2 \cap C_3|\mathbf{x}) - P(C_3 \cap C_4|\mathbf{x}) \\ +P(C_1 \cap C_2 \cap C_3|\mathbf{x}) + P(C_1 \cap C_2 \cap C_4|\mathbf{x}) \\ +P(C_1 \cap C_3 \cap C_4|\mathbf{x}) + P(C_2 \cap C_3 \cap C_4|\mathbf{x}) \\ -P(C_1 \cap C_2 \cap C_3 \cap C_4|\mathbf{x}) \end{array}}$$

$$= \frac{\frac{28}{50} - \frac{13}{50} - \frac{13}{50} + \frac{7}{50}}{\frac{28}{50} + \frac{26}{50} - \frac{13}{50} - \frac{12}{50} - \frac{13}{50} - \frac{18}{50} - \frac{13}{50} + \frac{8}{50} + \frac{7}{50} + \frac{7}{50} + \frac{9}{50} - \frac{5}{50}}$$

$$= \frac{9}{11} = 0.8182$$

and $P\left(C_3 | C_2^c \cap C_4^c \cap \mathbf{x}\right) = \dfrac{\frac{26}{50} - \frac{18}{50} - \frac{13}{50} + \frac{9}{50}}{\frac{11}{50}} = \dfrac{4}{11} = 0.3636$

Note that had a simple renormalisation been used following the exclusion of class 2, the results would have been given by

$$P\left(C_1 | \mathbf{x}\right) = \frac{P\left(C_1 | \mathbf{x}\right)}{P\left(C_1 \cup C_3 | \mathbf{x}\right)} = \frac{P\left(C_1 | \mathbf{x}\right)}{P\left(C_1 | \mathbf{x}\right) + P\left(C_3 | \mathbf{x}\right) - P\left(C_1 \cap C_3 | \mathbf{x}\right)}$$

$$= \frac{\frac{28}{50}}{\frac{28}{50} + \frac{26}{50} - \frac{12}{50}} = \frac{28}{42} = 0.6666$$

Similarly, $P\left(C_3 | \mathbf{x}\right) = \dfrac{26}{42} = 0.6190$

A simple renormalisation is not sufficient because the classes are not exclusive or independent and, consequently, the exclusion of classes 2 and 4 affects the probabilities of occurrence of classes 1 and 3 depending upon the extent of 'coupling' between the respective classes (see Table 1).

Class	Adjusted for overlap	Simple Renormalisation
1	0.8182	0.6666
3	0.3636	0.6190

Table 1 The effects of adjusting the new posterior probabilities by taking into account the overlaps with the excluded classes. Note that the probabilities obtained using a simple renormalisation are only valid for exclusive or independent classes.

2.3.7 The Use of Bayes Theorem.

As will be discussed in Section 1.7, posterior probabilities can be estimated directly if certain techniques are used. In some cases, however, it may be more appropriate to use Bayesian decision theory, and compute the posterior probabilities indirectly rather than directly estimating them. Bayesian decision theory is a framework for calculating the required conditional probabilities from other empirically derivable probabilities (e.g. Duda and Hart, 1973; Gelman *et al*, 1995). Bayes' theorem for real valued data variables is of the form

$$P(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)P(C_i)}{p(\mathbf{x})} \qquad (6)$$

where $P(C_i|\mathbf{x})$, is the *posterior probability*, $p(\mathbf{x}|C_i)$ is the *likelihood*, $P(C_i)$, is the *prior probability* of class i occurring and $p(\mathbf{x})$ is the *unconditional density function*. These probabilities are estimated from the data.

For a set of *exclusive* classes, the form of $p(\mathbf{x})$ is given by

$$p(\mathbf{x}) = \sum_i^N p(\mathbf{x} \cap C_i) = \sum_i^N p(\mathbf{x}|C_i)P(C_i) \qquad (7)$$

as $\boldsymbol{x}$ is generated by a single FC only. Equation (7) ensures that the posterior probabilities sum to unity, i.e.,

$$\sum_{i=1}^N P(C_i|\mathbf{x}) = 1 \qquad (8)$$

Equation (7) is a special case of the more general case involving non-exclusive classes given by

$$\begin{aligned} p(\mathbf{x}) &= p(\mathbf{x} \cap U) = \\ p\left(\mathbf{x} \cap \left(\bigcup_{r=1}^N C_i\right)\right) &= \sum_{i=1}^N p(\mathbf{x} \cap C_i) \\ &\quad - \sum_{i<j}^N p(\mathbf{x} \cap C_i \cap C_j) \\ &\quad + \sum_{i<j<k}^N p(\mathbf{x} \cap C_i \cap C_j \cap C_k) \\ &\quad \vdots \\ &\quad + (-1)^{N+1} p(\mathbf{x} \cap C_1 \cap C_2 \cap \ldots \cap C_N) \end{aligned}$$

$$
\begin{aligned}
&= \sum_{i=1}^{N} p(\mathbf{x}|C_i)P(C_i) \\
&- \sum_{i<j}^{N} p(\mathbf{x}|C_i \cap C_j)P(C_i \cap C_j) \\
&+ \sum_{i<j<k}^{N} p(\mathbf{x}|C_i \cap C_j \cap C_k)P(C_i \cap C_j \cap C_k) \qquad (9) \\
&\vdots \\
&+ (-1)^{N+1} p(\mathbf{x}|C_1 \cap C_2 \cap \ldots \cap C_N)P(C_1 \cap C_2 \cap \ldots \cap C_N)
\end{aligned}
$$

where equation (9) ensures that the probability of the union of the classes conditional upon $\boldsymbol{x}$ is unity, i.e. every input is classified.

$$
\begin{aligned}
&P(U|\mathbf{x}) = \\
&P\left(\left(\bigcup_{r=1}^{N} C_i\right)|\mathbf{x}\right) = \sum_{i=1}^{N} P(C_i|\mathbf{x}) \\
&\qquad - \sum_{i<j}^{N} P(C_i \cap C_j|\mathbf{x}) \\
&\qquad + \sum_{i<j<k}^{N} P(C_i \cap C_j \cap C_k|\mathbf{x}) \qquad (10) \\
&\qquad \vdots \\
&\qquad + (-1)^{N+1} P(C_1 \cap C_2 \cap \ldots \cap C_N|\mathbf{x}) \\
&= 1
\end{aligned}
$$

where $P(C_i \cap C_j|\mathbf{x}) = \dfrac{p(\mathbf{x}|C_i \cap C_j)P(C_i \cap C_j)}{p(\mathbf{x})}$ etc.

Equation (10) reduces to Equation (8) when $C_i \cap C_j = \phi$, i.e. the classes are exclusive giving rise to the usual definition of Bayes' theorem (e.g. Walpole and Myers, 1989):

Given a partition of the event space, $\{B_1, \ldots, B_N\}$ that is $B_i \cap B_j = \phi, \quad \forall i \neq j$, and a set A such that $A \subseteq \bigcup_{k=1}^{N} B_k$, the conditional probability, $P(B_i|A)$ can be written as

$$
P(B_i|A) = \frac{P(B_i|A)}{P(A)} = \frac{P(B_i)P(A|B_i)}{\sum_{j}^{N} P(B_j)P(A|B_j)}
$$

Note that the condition that $B_i \cap B_j = \phi, \quad \forall i \neq j$ is required.

2.3.8 Posterior Knowledge Inclusion For Exclusive Classes

The next four sections examine the inclusion of posterior knowledge for the exclusive, independent and dependent class cases where only members of a single type of class are to be excluded. The more general case is examined in Section 1.4

where it is shown that the three class types can be decoupled and, thus, treated individually.

For exclusive classes, the situation is shown schematically in Figure 8.

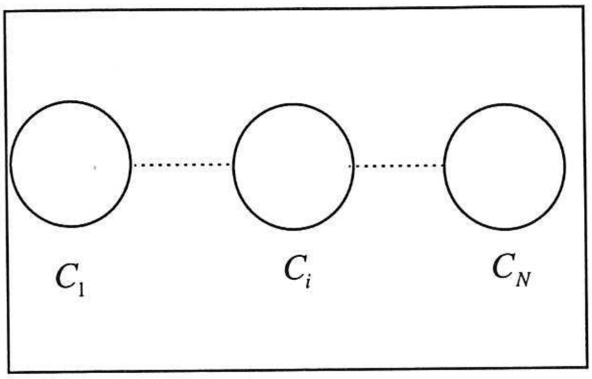


Figure 8. A set of exclusive classes.

Because the classes are exclusive, $P(C_i \cap C_j|\mathbf{x}) = 0 \ \ \forall i, j$ i.e. all probabilities of joint classes are zero, only the single class probabilities $P(C_i|\mathbf{x})$ are required to calculate the class union probability, $P\left(\bigcup_{r=1}^{N} C_r|\mathbf{x}\right)$ in Equation (5). This fact leads to the rule given in 1.5.2.

2.3.8.1 Exclusive FC Example:

A three class problem is specified as follows:
The input variable $\mathbf{x} = x$ is one dimensional.

Priors: $P(C_1) = P(C_2) = P(C_3) = \frac{1}{3}$
Likelihoods: $P(\mathbf{x} \mid C_1) \sim N(3,2)$, $P(\mathbf{x} \mid C_2) \sim N(6,3)$, $P(\mathbf{x} \mid C_3) \sim N(8,2)$.
Where $N(.,.)$ denotes the normal distribution.

The likelihoods are shown in Figure 9. Note that there are no occurrences of two or more classes together because the classes are exclusive.

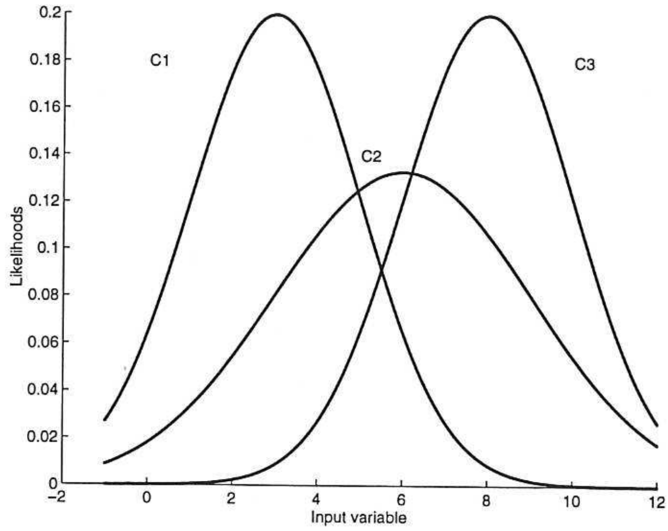


Figure 9 Likelihoods for a three class example where the sets are exclusive.

The posterior probabilities are given by Bayes' theorem

$$P(C_i|\mathbf{x}) = \frac{p(\mathbf{x}|C_i)P(C_i)}{p(\mathbf{x})}, \; i = 1,2,3 \text{ where}$$

$$p(\mathbf{x}) = \sum_i^3 p(\mathbf{x} \cap C_i) = \sum_i^3 p(\mathbf{x}|C_i)P(C_i)$$

The posterior probabilities are shown in Figure 10.

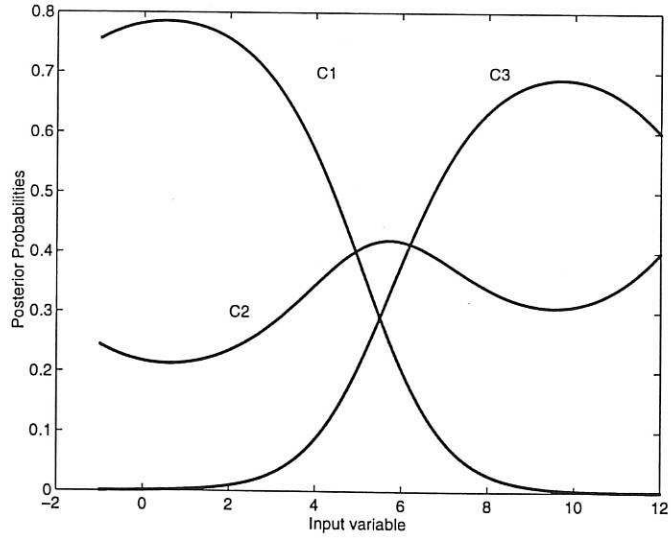


Figure 10.Posterior probabilities for the three class example of Figure 10.

At the point $x = 6$, $P(C_1|6) = 0.2032$, $P(C_2|6) = 0.4172$, $P(C_3|6) = 0.3796$, and $\sum_{i=1}^{3} P(C_i|\mathbf{x}) = 1$ as expected.

Given the *posterior* knowledge that class three has been excluded, in this case, the updated posterior probabilities are given by equation (4) and shown in Figure 11.

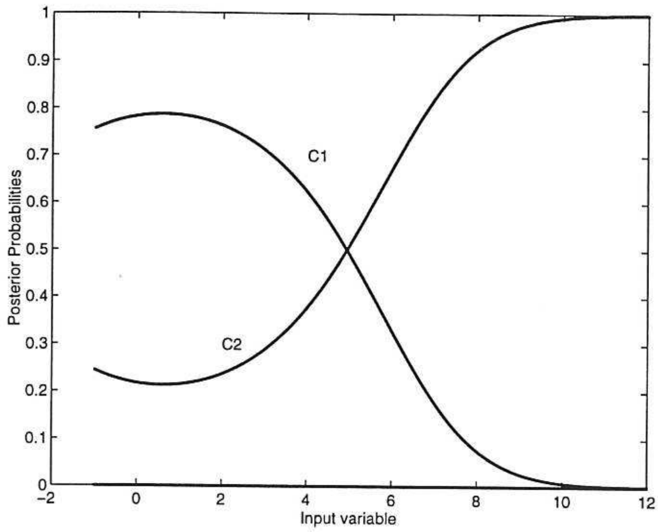


Figure 11. Posterior probabilities where class three has been excluded. At each point, the remaining class probabilities are renormalised.

At the point $x = 6$, $P(C_1|6) = 0.3275$, $P(C_2|6) = 0.6725$.

From the original posterior probabilities, owing to the exclusiveness of the classes, there is a simple renormalisation giving

$$P(C_1|6) = \frac{0.2032}{0.2032 + 0.4172} = 0.3275$$

and

$$P(C_1|6) = \frac{0.4172}{0.2032 + 0.4172} = 0.6725$$

as expected. The renormalisation of exclusive classes is covered further in section 1.5 (rule 1.5.2)

2.3.8.2 Example: a five FC problem:

Two fault classes C_{f1} and C_{f2} whose posterior probabilities will be updated when posterior knowledge becomes available. Three classes C_e, C_i and C_d which are exclusive, independent and dependent respectively (that is, $C_e \cap C_i = \phi$ etc.) are to be excluded on the basis of external knowledge.

2.3.8.2.1 1.3.8.2.1 Excluding the Exclusive Class

Where the fault classes are *excluded* by PKI, the updated probability for class C_{f1} is given by Equation (4)

$$P\left(C_{f1} \middle| C_e^c \cap C_i^c \cap C_d^c \cap \mathbf{x}\right) = \frac{P\left(C_{f1} \cup C_e \cup C_i \cup C_d \middle| \mathbf{x}\right) - P\left(C_e \cup C_i \cup C_d \middle| \mathbf{x}\right)}{P\left(C_{f1} \cup C_{f2} \cup C_e \cup C_i \cup C_d \middle| \mathbf{x}\right) - P\left(C_e \cup C_i \cup C_d \middle| \mathbf{x}\right)}$$

$$= \frac{P\left(\left(C_{f1} \cup C_e\right) \cup \left(C_i \cup C_d\right) \middle| \mathbf{x}\right) - P\left(C_e \cup \left(C_i \cup C_d\right) \middle| \mathbf{x}\right)}{P\left(\left(C_{f1} \cup C_{f2} \cup C_e\right) \cup \left(C_i \cup C_d\right) \middle| \mathbf{x}\right) - P\left(C_e \cup \left(C_i \cup C_d\right) \middle| \mathbf{x}\right)}$$

$$= \frac{P\left(\left(C_{f1} \cup C_e\right) \cup \left(C_{id}\right) \middle| \mathbf{x}\right) - P\left(C_e \cup \left(C_{id}\right) \middle| \mathbf{x}\right)}{P\left(\left(C_{f1} \cup C_{f2} \cup C_e\right) \cup \left(C_{id}\right) \middle| \mathbf{x}\right) - P\left(C_e \cup \left(C_{id}\right) \middle| \mathbf{x}\right)}$$

$$= \frac{\begin{array}{c} P\left(C_{f1} \cup C_e \middle| \mathbf{x}\right) + P\left(C_{id} \middle| \mathbf{x}\right) - P\left(\left(C_{f1} \cup C_e\right) \cap C_{id} \middle| \mathbf{x}\right) - \\ \left\{P\left(C_e \middle| \mathbf{x}\right) + P\left(C_{id} \middle| \mathbf{x}\right) - P\left(C_e \cap C_{id} \middle| \mathbf{x}\right)\right\} \end{array}}{\begin{array}{l} P\left(C_{f1} \cup C_{f2} \cup C_e \middle| \mathbf{x}\right) + P\left(C_{id} \middle| \mathbf{x}\right) - \\ P\left(\left(C_{f1} \cup C_{f2} \cup C_e\right) \cap C_{id} \middle| \mathbf{x}\right) - \left\{P\left(C_e \middle| \mathbf{x}\right) + P\left(C_{id} \middle| \mathbf{x}\right) - P\left(C_e \cap C_{id} \middle| \mathbf{x}\right)\right\} \end{array}}$$

where C_{id} is used to represent $C_i \cup C_d$ for convenience, giving

$$(\;_{f1}| \;^c_e \;^c_i \;^c_d \;)$$

$$\frac{(\;_{f1} \;_e|) \quad ((\;_{f1} \;_e) \;_{id}|) \quad (\;_e|) \quad (\;_e \;_{id}|)}{((\;_{f1} \;_{f2}) \;_e|) \quad ((\;_{f1} \;_{f2} \;_e) \;_{id}|) \quad (\;_e|) \quad (\;_e \;_{id}|)}$$

$$\frac{(\;_{f1} \;_e|) \quad ((\;_{f1} \;_e) \;_{id}|) \quad (\;_e|) \quad ( )}{((\;_{f1} \;_{f2}) \;_e|) \quad ((\;_{f1} \;_{f2} \;_e) \;_{id}|) \quad (\;_e|) \quad ( )}$$

$$(\;_{f1} \;_e|) \quad ((\;_{f1} \;_e) \;_{id}|) \quad (\;_e|)$$

$$(\;_{f1} \;_{f2})$$

This job requires more memory than is available in this printer.
Try one or more of the following, and then print again:

In the PostScript dialog box, click Optimize For Portability.

In the Device Options dialog box, make sure the Available Printer Memory is accurate.

Reduce the number of fonts in the document.

Print the document in parts.

Air is compressed by the compressor prior to combustion. A triple spool (axial flow) compressor is shown below. Each spool is driven by its own turbine and consists of multiple stages, each of which increases the air pressure by a small amount.

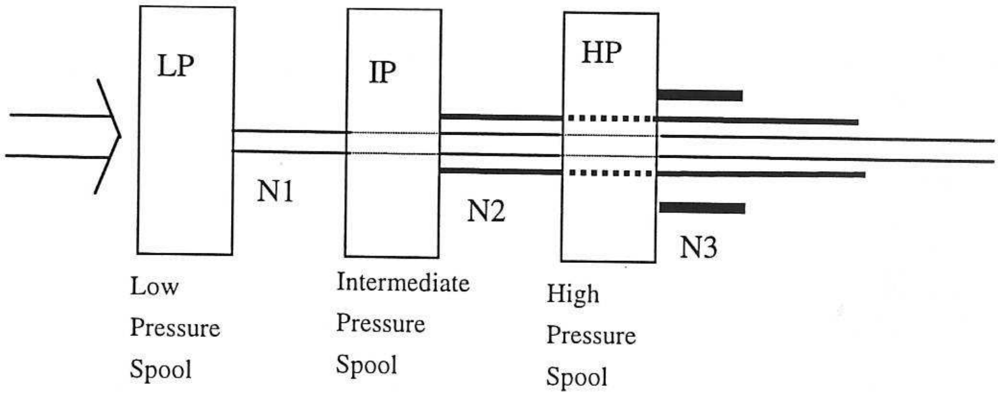


Figure 18. The compressor N1,N2, N3 low, intermediate and high pressure shaft speeds respectively

2.7.2 Trent 700 model fault induction:

The model is used for the induction of simple faults so that data can be generated to investigate the probability estimation problem. An overview of the fault induction process is:

Nominal operating points (NOPs) are determined by the Mach and altitude (MA) settings. The Mach and altitude settings are chosen at random from a uniform distribution in the ranges [0.3,0.8] and [20000,80000] ft respectively. This method is used for simplicity. A more realistic flight envelope could be chosen and is a possible extension.

The MA settings give rise to a series of dependent parameters that include the sub-set:

1. WF: Fuel flow,
2. TGT: Turbine gas temperature sensed by thermocouple, at entry to the LP turbine
3. N3: HP shaft speed,
4. P30: HP compressor delivery pressure (total),
5. T30: HP compressor delivery temperature (total).

The TGT thermocouple measures the core gas temperature at its entry to the LP turbine.

This sub-set has been chosen specifically as a first approximation. This the minimal set of useful parameters chosen in conjunction with engineers from Rolls Royce plc. An extended, richer set of parameters is a subject for further investigation.

Induced fault cases are chosen to be:

1. Fuel decay
2. TGT increase
3. N3 decay
4. Normal (N) Fault not induced ie running at NOP
5. No fault found (NFF): fault not induced but flagged as faulty

This set has been chosen as a first approximation to the problem. The fault severities are fixed but may be varied in an extended model. Fuel decay is realistic in that partial blockages may occur in fuel pipes or filters. TGT increase and N3 decay may reflect a decrease in engine efficiency. No fault found reflects the condition in which one or more faults are flagged by the system but no apparent cause can be found. Faults 1, 2 and 3 may also occur in binary combination giving eight fault conditions in total. These are f1, f2, f3, f1&f2, f1&f3, f2&f3, N, and NFF. The condition f1&f2&f3 has been omitted for simplicity.

Priors are chosen for each of the eight cases to reflect the mix of actual faults, no fault found and normal conditions. For case 5, no fault found, a single fault is chosen at random from faults 1 to 3 depending upon the priors.

2.7.2.1 Fault Diagnosis Issues

The engine and accessories comprise a dynamical system and fault detection / diagnosis is not a straightforward task. Fault diagnosis is a research area in itself to determine both what constitutes a fault and how to detect faults. When faults occur, the EEC attempts to compensate for the problem and maintain the desired set-point further complicating the problem. Many design choices have had to be made to allow the generation of data of sufficient complexity and realism to investigate methods of posterior knowledge inclusion which is the primary remit of this work.

Even seemingly straightforward fault diagnosis methods such as range checking pose several problems. How are the ranges to be determined across the operating envelope? What constitutes a fault? How are fault labels to be associated with sets of parameters? Will fault conditions in steady-state mode following EEC intervention be mistaken for NOPs? In the latter case, will range checking be of any use? How are static faults to be detected and represented? How are dynamical faults to be detected and represented?

The steady-state reached following the injection of a fault may give a parameter vector commensurate with normal operation in a different region of the operating envelope. The labelling of the final parameter vector as indicating a fault may provide misinformation to a FD system. It may be possible to prevent this by including the Mach number and altitude information in the parameter vector.

The N3 decay fault is a case in point. Whilst operating at a NOP during an experiment, (fixed Mach number and altitude) a decay was introduced into the HP shaft speed (which may indicate bearing faults for example). An expected increase in fuel flow (WF) occurred as the EEC attempted to compensate for the problem. There was an initial surge in fuel flow followed by reversion to a steady-state value of fuel flow not significantly above the original. The final parameter vector may be indicative of a normal (N) state. In this case, the FD problem is one of detecting a fuel surge not accounted for by normal operation or allowed transients. This involves the detection of dynamical anomalies—a research topic in itself. Such considerations belong to the domain of fault diagnosis proper. Here, the concern is with post-processing of fault occurrence probabilities, hence our crude simulation and detection methods. The main thrust of this work is in the direction of post-processing techniques and, hence, the introduction of context-free simulations as discussed in Chapter 2.

2.7.3 A Neural Network Approach to Probability Estimation

A common method of estimating posterior probabilities is to use an artificial neural network (e.g. Bishop, 1995; Richard and Lippmann, 1991). Where the FCs are exclusive, given N classes, there arises the 1 from N estimation problem, that is, for each input, one FC will be chosen on the basis of the posterior probabilities. Where the classes are non-exclusive, more than one FC can occur simultaneously giving rise to an M from N estimation problem. It has been shown (e.g. Bishop, 1995; Richard and Lippmann, 1991) for both the mean squared error (MSE) and cross entropy (CE) performance measures, that feedforward neural networks (such as the multilayer Perceptron or Radial Basis Function Network) will estimate the total Bayesian posterior probabilities of the form $P(C_i|\mathbf{x})$ only. Thus, although joint class information (M from N) is available in the training vectors, a conventional neural network classifier will not be able to estimate the joint probability function unless the output space is expanded to give an equivalent 1 from *many* problem. To capture class combination information in general, an augmented output vector consisting of 2^N outputs is required. The expansion is valid if the output space is treated as a collection of disjoint sets or *partitions* (Halmos, 1974). The desired probabilities may then be reconstructed from the members of the partition. The posterior probability update equation can be considered as a renormalisation of such partitions. Figure 19 illustrates how a set of overlapping classes is broken down into a set of disjoint classes for probability estimation.

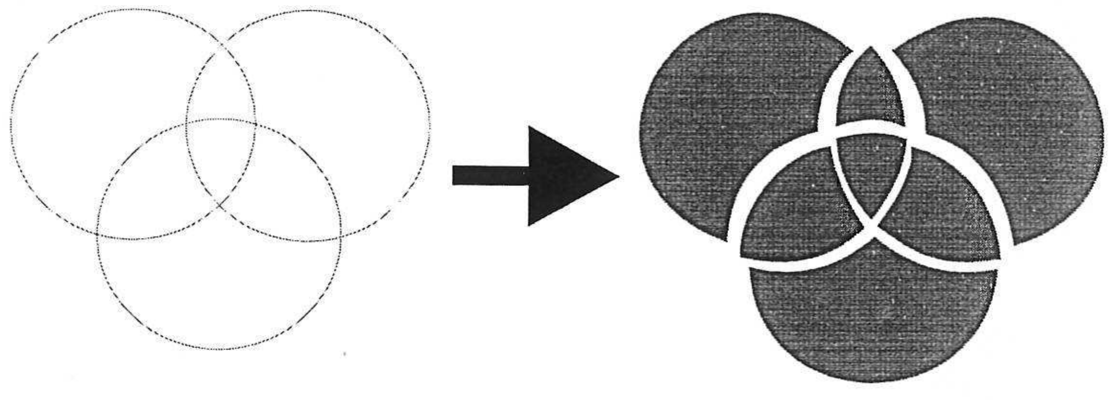


Figure 19. A set of overlapping fault classes is treated as a partition for convenience. The exclusive class probabilities are estimated and used to reconstruct the non-exclusive fault probabilities.

Figure 20 shows the neural network training situation schematically. If the desired class training vector was $[1...1...1]'$ for example, joint class information would be available but would not be learned by the network. It is equivalent to having N decoupled networks with no correlation between the outputs, hence the M from N to a 1 from 2^N expansion at the worse possible case. Thus, the apparently complex tasks of estimating overlapping class probabilities and updating the posterior probabilities have been simplified by using a partition which divides the fault-space into exclusive regions. A renormalisation of the remaining probabilities, following posterior knowledge integration, is then carried out to give the updated posteriors. Posterior probabilities of fault occurrence are used to guide engineers to individual faulty sub-units in a given plant.

The whole process of posterior knowledge integration is shown schematically in figure 21. Not all of the joint probabilities will be nonzero unless the worst case scenario occurs. The subset of relevant probabilities is chosen, forming a partition, and estimated using a neural network or other method. A possible fault scenario identification cycle is then entered.

The desired FC posteriors are reconstructed giving a ranking of sub-unit fault probabilities. This information is used to make a sub-unit inspection. If the fault scenario is identified—i.e. there are no other faults to be found—then the cycle ends. If the scenario is not yet identified, then the new inspection information is fed in as further posterior knowledge and the cycle continues. For the purpose of our simulation, the number of faults in a scenario is known in advance to provide the stopping criterion. However, in reality, a posterior probability threshold may be used to determine when the fault search is halted.

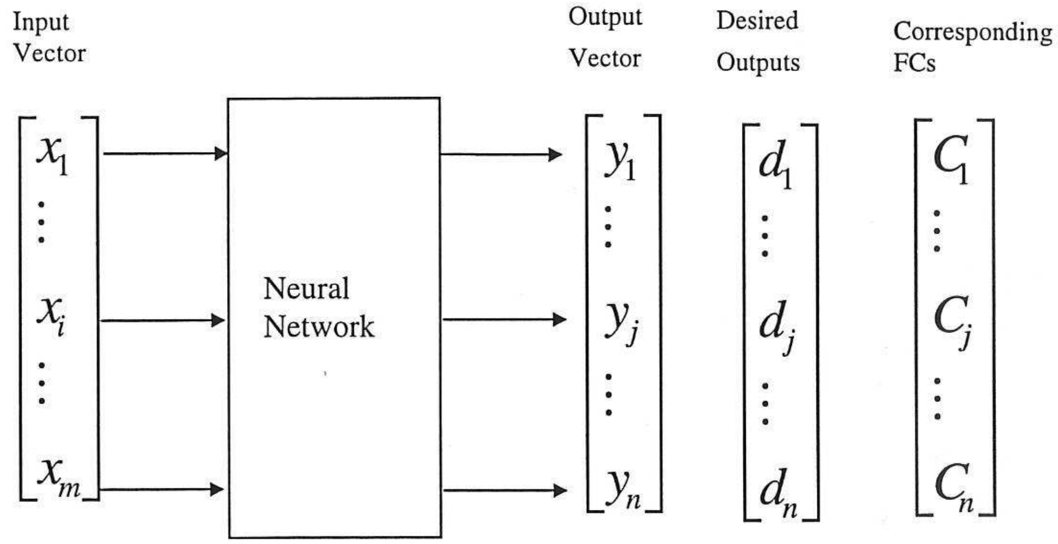


Figure 20. A schematic illustration of the process of using a neural network as a classifier. If a class is indicated, the relevant desired output is set to 1, otherwise it is left at 0.

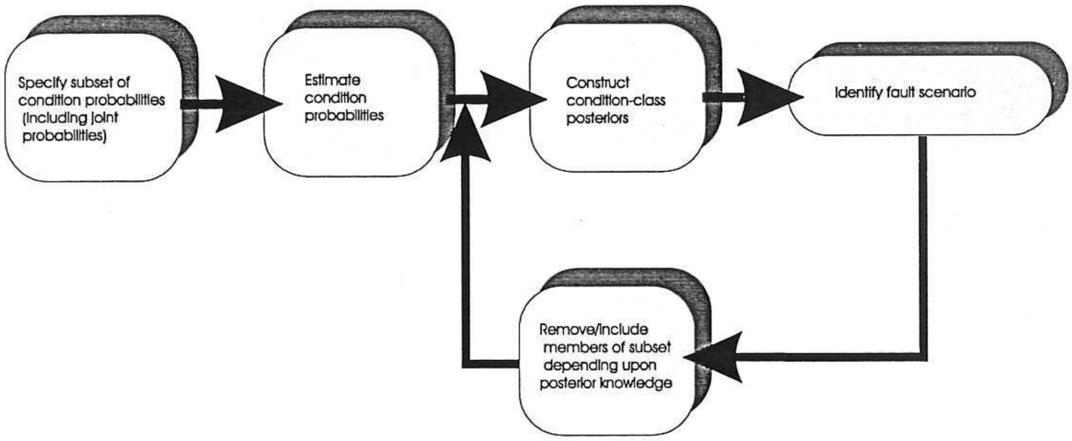


Figure 21. The posterior knowledge integration cycle. The posterior knowledge feedback occurs until all faults are isolated.

The analysis given here is general and applies to both regression and classification problems and involves minimising the mean square error (MSE). For classification problems, however, the cross-entropy measure is more useful and a similar result for cross-entropy will be found in Richard and Lippman (1991).

Assuming discrete outputs indicating class membership, d_{ij} where i signifies the output and j signifies the discrete output value, for L output values, the (MSE) of classification can be calculated by

$$E = \lim_{N_d \to \infty} \frac{1}{N_d} \sum_{n=1}^{N_d} \sum_{i=1}^{N} \sum_{j=1}^{L} \left[y_i - d_{ij} \right]^2 P\left(d_{ij} \cap \mathbf{x}\right) \qquad (12)$$

for N_d data points

For continuous output values and applying the law of large numbers (e.g. Bishop, 1995)

$$E = \sum_{i=1}^{N} \iint [y_i - d_i]^2 p(d_i \cap \mathbf{x}) dd_i d\mathbf{x}$$

$$= \sum_{i=1}^{N} \iint [y_i - d_i]^2 p(d_i | \mathbf{x}) p(\mathbf{x}) dd_i d\mathbf{x}$$

where d_i is a continuous variable.

From equation (12) using the law of large numbers and assuming that $P(d_{ij} \cap \mathbf{x}) = P(C_j \cap \mathbf{x})$ i.e any network output value depends upon class membership,

$$E = \int \sum_{i=1}^{N} \sum_{j=1}^{N} [y_i - d_{ij}]^2 p(\mathbf{x} \cap C_j) d\mathbf{x}$$

$$= \sum_{i=1}^{N} \sum_{j=1}^{N} \int [y_i^2 - 2y_i d_{ij} + d_{ij}^2] p(\mathbf{x} \cap C_j) d\mathbf{x}$$

$$= \sum_{i=1}^{N} \int \left[y_i^2 \sum_{j=1}^{N} p(\mathbf{x} \cap C_j) - 2y_i \sum_{j=1}^{N} d_{ij} p(\mathbf{x} \cap C_j) + \sum_{j=1}^{N} d_{ij}^2 p(\mathbf{x} \cap C_j) \right] d\mathbf{x} \qquad (13)$$

For 1 from *N* classification $d_{ii} = 1$, for $\mathbf{x} \in C_i$ and $d_{ij} = 0$, for $\mathbf{x} \notin C_j$.

Furthermore, as the classes are exclusive (1 from N) and x belongs to one of the classes $\sum_{j=1}^{N} p(\mathbf{x} \cap C_j) = p(\mathbf{x})$, can be substituted into equation (13) to give

$$E = \sum_{i=1}^{N} \int [y_i^2 p(\mathbf{x}) - 2y_i p(\mathbf{x} \cap C_i) + p(\mathbf{x} \cap C_i)] d\mathbf{x}$$

$$= \sum_{i=1}^{N} \int [y_i^2 p(\mathbf{x}) - 2y_i P(C_i | \mathbf{x}) p(\mathbf{x}) + P(C_i | \mathbf{x}) p(\mathbf{x})] d\mathbf{x}$$

$$= \sum_{i=1}^{N} \int [y_i^2 - 2y_i P(C_i | \mathbf{x}) + P(C_i | \mathbf{x}) + P^2(C_i | \mathbf{x}) - P^2(C_i | \mathbf{x})] p(\mathbf{x}) d\mathbf{x}$$

$$= \sum_{i=1}^{N} \int [y_i^2 - 2y_i P(C_i | \mathbf{x}) + P(C_i | \mathbf{x}) + P^2(C_i | \mathbf{x}) - P^2(C_i | \mathbf{x})] p(\mathbf{x}) d\mathbf{x}$$

and, finally,

$$E = \sum_{i=1}^{N} \int \left[y_i - P(C_i|\mathbf{x}) \right]^2 p(\mathbf{x}) d\mathbf{x} + \int P(C_i|\mathbf{x})(1 - P(C_i|\mathbf{x})) p(\mathbf{x}) d\mathbf{x}$$

To minimise E with respect to the parameters, the second term can be ignored because it is not a function of the parameters. This leaves the first term which gives $y_i = p(C_i|\mathbf{x})$ for a minimum to occur as the integral will always be positive.

A number of assumptions have been made in the above derivation (Bishop, 1995):

i) a large data set which approximates to an infinite set, is available
ii) parameters (weights) exist such that $y_i(\mathbf{x},\mathbf{w}) \rightarrow P(C_i|\mathbf{x})$ i.e. the approximating function is able to approximate the required probabilities, and
iii) the optimisation procedure finds the appropriate minimum.

It is also assumed that the classes are exclusive to ensure $\sum_{j=1}^{K} p(\mathbf{x} \cap C_j) = p(\mathbf{x})$. This can be written as

$\sum_{j=1}^{K} P(C_j|\mathbf{x}) p(\mathbf{x}) = p(\mathbf{x})$ which implies that $\sum_{j=1}^{K} P(C_j|\mathbf{x}) = 1$.

2.7.4 Conditional Expectation of Vector Output (M from N)

It cannot be assumed that all classes will be exclusive. Where more than one class is likely at any one time, the problem becomes an m from n estimation problem. A procedure analogous to the one above for deriving the result $y_i = p(C_i|\mathbf{x})$ is given below for the more general m from n estimation problem where it is convenient to formulate the class membership problem in terms of vector output.

It will be shown that although joint class information (m from n) is available in the training vectors, a neural network will not be able to estimate the joint probability function unless the output space is expanded to give an equivalent 1 from n problem.

For the continuous valued vector output case (regression problem) the MSE is given by

$$E = \iint \|\mathbf{y} - \mathbf{d}\|^2 p(\mathbf{d} \cap \mathbf{x}) d\mathbf{d} d\mathbf{x} = \iint \|\mathbf{y} - \mathbf{d}\|^2 p(\mathbf{d}|\mathbf{x}) p(\mathbf{x}) d\mathbf{d} d\mathbf{x}$$

which implies that for a minimum MSE,

$$\mathbf{y} = \langle \mathbf{d}|\mathbf{x} \rangle = \int \mathbf{d} p(\mathbf{d}|\mathbf{x}) d\mathbf{d}$$

For the discrete output case (classification) which is of more relevance to the m from n problem, the minimum MSE is given by

$\mathbf{y} = \langle \mathbf{d}|\mathbf{x} \rangle = \sum_{i=1}^{2^N-1} \mathbf{d}_i P(\mathbf{d}_i|\mathbf{x})$ where $\mathbf{d}_i$ is a binary output vector.

Proof:

$$E = \int \sum_{i=1}^{2^N-1} \|\mathbf{y} - \mathbf{d}_i\|^2 P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$= \int \sum_{i=1}^{2^N-1} \|(\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle) + (\langle \mathbf{d}|\mathbf{x} \rangle - \mathbf{d}_i)\|^2 P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$= \int \sum_{i=1}^{2^N-1} (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle)^2 P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$
$$+ 2\int \sum_{i=1}^{2^N-1} (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle)(\langle \mathbf{d}|\mathbf{x} \rangle - \mathbf{d}_i) P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$
$$+ \int \sum_{i=1}^{2^N-1} (\langle \mathbf{d}|\mathbf{x} \rangle - \mathbf{d}_i)^2 P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \qquad (14)$$

For term 1:

$$\int \sum_{i=1}^{2^N-1} (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle)^2 P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$= \int (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle)^2 \sum_{i=1}^{2^N-1} P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$= \int (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle)^2 .1. p(\mathbf{x}) d\mathbf{x}$$

$$= \int (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle)^2 p(\mathbf{x}) d\mathbf{x}$$

For term 2:

$$2\int \sum_{i=1}^{2^N-1} (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle)(\langle \mathbf{d}|\mathbf{x} \rangle - \mathbf{d}_i) P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x}) d\mathbf{x}$$

$$= 2\int (\mathbf{y} - \langle \mathbf{d}|\mathbf{x} \rangle) \sum_{i=1}^{2^N-1} \{\langle \mathbf{d}|\mathbf{x} \rangle P(\mathbf{d}_i|\mathbf{x}) - \mathbf{d}_i P(\mathbf{d}_i|\mathbf{x})\} p(\mathbf{x}) d\mathbf{x}$$

$$= 2\int \left(\mathbf{y} - \langle \mathbf{d}|\mathbf{x}\rangle\right)\left\{\langle \mathbf{d}|\mathbf{x}\rangle \sum_{i=1}^{2^N-1} P(\mathbf{d}_i|\mathbf{x}) - \sum_{i=1}^{2^N-1} \mathbf{d}_i P(\mathbf{d}_i|\mathbf{x})\right\} p(\mathbf{x})d\mathbf{x}$$

$$= 2\int \left(\mathbf{y} - \langle \mathbf{d}|\mathbf{x}\rangle\right)\left\{\langle \mathbf{d}|\mathbf{x}\rangle .1 - \langle \mathbf{d}|\mathbf{x}\rangle\right\} p(\mathbf{x})d\mathbf{x}$$

$$= 0$$

For term 3:

$$\int \sum_{i=1}^{2^N-1} \left(\langle \mathbf{d}|\mathbf{x}\rangle - \mathbf{d}_i\right)^2 P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x})d\mathbf{x}$$

$$= \int \sum_{i=1}^{2^N-1} \left(\langle \mathbf{d}|\mathbf{x}\rangle^2 - 2\langle \mathbf{d}|\mathbf{x}\rangle \mathbf{d}_i + \mathbf{d}_i^2\right) P(\mathbf{d}_i|\mathbf{x}) p(\mathbf{x})d\mathbf{x}$$

$$= \int \left\{\langle \mathbf{d}|\mathbf{x}\rangle^2 \sum_{i=1}^{2^N-1} P(\mathbf{d}_i|\mathbf{x}) - 2\langle \mathbf{d}|\mathbf{x}\rangle \sum_{i=1}^{2^N-1} \mathbf{d}_i P(\mathbf{d}_i|\mathbf{x}) + \sum_{i=1}^{2^N-1} \mathbf{d}_i^2 P(\mathbf{d}_i|\mathbf{x})\right\} p(\mathbf{x})d\mathbf{x}$$

$$= \int \left\{\langle \mathbf{d}|\mathbf{x}\rangle^2 .1 - 2\langle \mathbf{d}|\mathbf{x}\rangle\langle \mathbf{d}|\mathbf{x}\rangle + \langle \mathbf{d}^2|\mathbf{x}\rangle\right\} p(\mathbf{x})d\mathbf{x}$$

$$= \int \left\{\langle \mathbf{d}^2|\mathbf{x}\rangle - \langle \mathbf{d}|\mathbf{x}\rangle^2\right\} p(\mathbf{x})d\mathbf{x}$$

Substituting the above terms into equation (14) gives the expression for the MSE

$$E = \int \left(\mathbf{y} - \langle \mathbf{d}|\mathbf{x}\rangle\right)^2 p(\mathbf{x})d\mathbf{x} + \int \left(\langle \mathbf{d}^2|\mathbf{x}\rangle - \langle \mathbf{d}|\mathbf{x}\rangle^2\right) p(\mathbf{x})d\mathbf{x}$$

As the second term is determined by the data, the minimum MSE will be where $\mathbf{y} = \langle \mathbf{d}|\mathbf{x}\rangle$ in the first term

For three classes there will be $2^3 - 1 = 7$ different binary output vectors and the expected conditional output will be given by

$\langle \mathbf{d}|\mathbf{x}\rangle = \sum_{i=1}^{7} \mathbf{d}_i P\left(\left(C_{\gamma 1} \cap \ldots \cap C_{\gamma n}|\mathbf{x}\right)'\right)$ $n \le 3$ where $\gamma_1 \ldots \gamma n \in \Delta_i$ the set of class indices involved for pattern i. The dash denotes the probability of any class or set of classes occurring exclusively i.e. $P\left(\left(C_3|\mathbf{x}\right)'\right)$ does not include $P\left(\left(C_1 \cap C_3|\mathbf{x}\right)'\right)$ etc.

$$\langle \mathbf{d}|\mathbf{x}\rangle = \mathbf{d}_1 P\left((C_3 \mid \mathbf{x})'\right) + \mathbf{d}_2 P\left((C_2 \mid \mathbf{x})\right) + \mathbf{d}_3 P\left((C_2 \cap C_3 \mid \mathbf{x})'\right) + \mathbf{d}_4 P\left((C_1 \mid \mathbf{x})'\right)$$

$$+\mathbf{d}_5 P\left((C_1 \cap C_3 \mid \mathbf{x})'\right) + \mathbf{d}_6 P\left((C_1 \cap C_2 \mid \mathbf{x})'\right) + \mathbf{d}_7 P\left((C_1 \cap C_2 \cap C_3 \mid \mathbf{x})'\right)$$

$$\langle \mathbf{d}|\mathbf{x}\rangle = \begin{bmatrix}0\\0\\1\end{bmatrix} P\left((C_3 \mid \mathbf{x})'\right) + \begin{bmatrix}0\\1\\0\end{bmatrix} P\left((C_2 \mid \mathbf{x})'\right) + \begin{bmatrix}0\\1\\1\end{bmatrix} P\left((C_2 \cap C_3 \mid \mathbf{x})'\right) + \begin{bmatrix}1\\0\\0\end{bmatrix} P\left((C_1 \mid \mathbf{x})\right)$$

$$+\begin{bmatrix}1\\0\\1\end{bmatrix} P\left((C_1 \cap C_3 \mid \mathbf{x})'\right) + \begin{bmatrix}1\\1\\0\end{bmatrix} P\left((C_1 \cap C_2 \mid \mathbf{x})'\right) + \begin{bmatrix}1\\1\\1\end{bmatrix} P\left((C_1 \cap C_2 \cap C_3 \mid \mathbf{x})'\right)$$

This implies that

$$\begin{aligned}
y_1 &= P\left((C_1|\mathbf{x})'\right) + P\left((C_1 \cap C_3|\mathbf{x})'\right) + P\left((C_1 \cap C_2|\mathbf{x})'\right) + P\left((C_1 \cap C_2 \cap C_3|\mathbf{x})'\right) \\
&= \left\{P(C_1|\mathbf{x}) - P(C_1 \cap C_3|\mathbf{x}) - P(C_1 \cap C_2|\mathbf{x}) + P(C_1 \cap C_2 \cap C_3|\mathbf{x})\right\} \\
&\quad + \left\{P(C_1 \cap C_3|\mathbf{x}) - P(C_1 \cap C_2 \cap C_3|\mathbf{x})\right\} \\
&\quad + \left\{P(C_1 \cap C_2|\mathbf{x}) - P(C_1 \cap C_2 \cap C_3|\mathbf{x})\right\} \\
&\quad + P(C_1 \cap C_2 \cap C_3|\mathbf{x}) \\
&= P(C_1|\mathbf{x})
\end{aligned}$$

Similarly, $y_2 = P(C_2|\mathbf{x})$ $y_3 = P(C_3|\mathbf{x})$.

In general, for calculating

$$\mathbf{y} = \langle \mathbf{d}|\mathbf{x}\rangle = \sum_{i=1}^{2^N-1} \mathbf{d}_i P(\mathbf{d}_i|\mathbf{x})$$

for any output, y_i, the class C_i will occur in $\frac{1}{2}2^N = 2^{n-1}$ terms in the summation because the other class intersections form a partition of C_i i.e.

$$P(C_i|\mathbf{x}) = P(C_i'|\mathbf{x}) + P\left((C_i \cap C_j)'|\mathbf{x}\right) + P\left((C_i \cap C_j \cap C_k)'|\mathbf{x}\right) + \ldots$$

$$+ P\left((C_i \cap C_j \cap C_k \cap \ldots \cap C_{N-1} \cap C_N)'|\mathbf{x}\right)$$

Figure 22 shows an example of the results obtainable using a multilayer Perceptron to estimate the posterior probabilities of a given a set of distributions. A data set consisting of 1000 data points was used to generate the graph. Note that the classes are not exclusive or independent. The MLP is an instantiation of the estimation problem and is only able to estimate the singleton class posterior probabilities although joint class data is available (i.e. more than one desired output bit may be active at any one time).

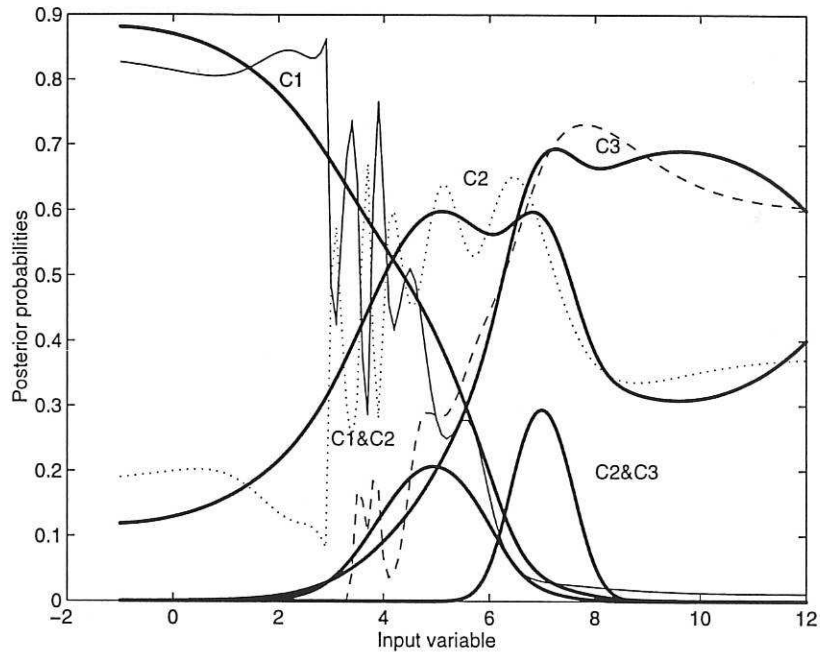


Figure 22 A graph showing the estimation of posterior probabilities by a Multilayer Perceptron. Note that only the singleton class probabilities have been estimated as expected.

It is clear that the probability of class 1 occurring contains some occurrences of class 1 paired with class 2 i.e. $P(C_1 \cap C_2|\mathbf{x}) \neq 0$. Thus, even though the training data incorporates examples of two classes occurring together, any method using binary outputs to indicate class membership based upon error minimisation as described in the previous discussion (including cross-entropy) is not able to extract this information using N outputs alone where N indicates the number of classes.

To capture class combination information in general, an augmented output vector consisting of $2^N - 1$ outputs is required.

Note the non-smooth approximation of $P(C_1|\mathbf{x})$ and $P(C_2|\mathbf{x})$. This problem and a possible solution, known as regularisation, is discussed in the context of radial basis function networks in the next sub-section.

2.7.5 Using Radial Basis Function Networks to Estimate the Posterior Probabilities

One way of estimating posterior probabilities is to use a radial basis function network (RBFN) (e.g. Powell, 1987, Broomhead and Lowe, 1988; Moody and Darken, 1989; Bishop, 1995). Radial basis function networks are capable of interpolating between data points to approximate a given noisy function (regression) or probability density function (classification).

A basic RBFN consists of a weighted linear sum of basis functions. This will not be gone into in detail here as there are many references dealing with this subject (e.g. Bishop, 1993, 1995; Haykin, 1994; Wasserman, 1993).

This documentation deals with classification problems which necessitates the use of the *softmax function* (e.g Bishop, 1995). To prevent over-learning of the training data, *regularisation* (Bishop, 1991, 1993,1995) may be used. The total cost function for any error-driven neural network using regularisation will be given by

$$C = E + \nu\Omega$$

where E is the original error function, ν is the regularisation constant and Ω is the regularisation function.

For the simulations given below, the second-order differential regularisation function is given by

$$\Omega = \sum_{i=1}^{N} \sum_{l=1}^{L} \left(\frac{\partial^2 y_i}{\partial x_l^2} \right)^2 .$$

Details of the implementation of an RBFN network with second-order differential regularisation applied to a standard network configuration with a softmax layer will be found in Appendix E.

Second-order differential regularisation penalises large changes in the curvature of the output function thus smoothing the resultant function.

The following dependent condition classes were generated using Gaussian distributions for the likelihoods of: C_1, C_2, C_3, $C_1 \cap C_2$, and $C_2 \cap C_3$. The RBFN is expected to approximate the posterior probabilities $P(C_1|\mathbf{x})$, $P(C_2|\mathbf{x})$, $P(C_3|\mathbf{x})$, $P(C_1 \cap C_2|\mathbf{x})$, and $P(C_2 \cap C_3|\mathbf{x})$. The RBFN used had an expanded output set consisting of 5 outputs, each output signifying that case alone e.g. $P(C_1|\mathbf{x})$ gives the posterior probability of class 1 occurring alone. To be consistent with earlier notation: $P(C_i|\mathbf{x}) = P(C_i'|\mathbf{x})$ and $P(C_i \cap C_j|\mathbf{x}) = P\left((C_i \cap C_j)'\middle|\mathbf{x}\right)$.

A Radial Basis Function network of the sort discussed previously was used in the simulations of this document. The network had a cross-entropy cost-function and incorporated a softmax layer to reflect the output probabilities. Second-order differential regularisation was used to reduce the rate of curvature of the output to prevent over-fitting to the data.

Figure 23 shows the estimated posterior probabilities without regularisation.

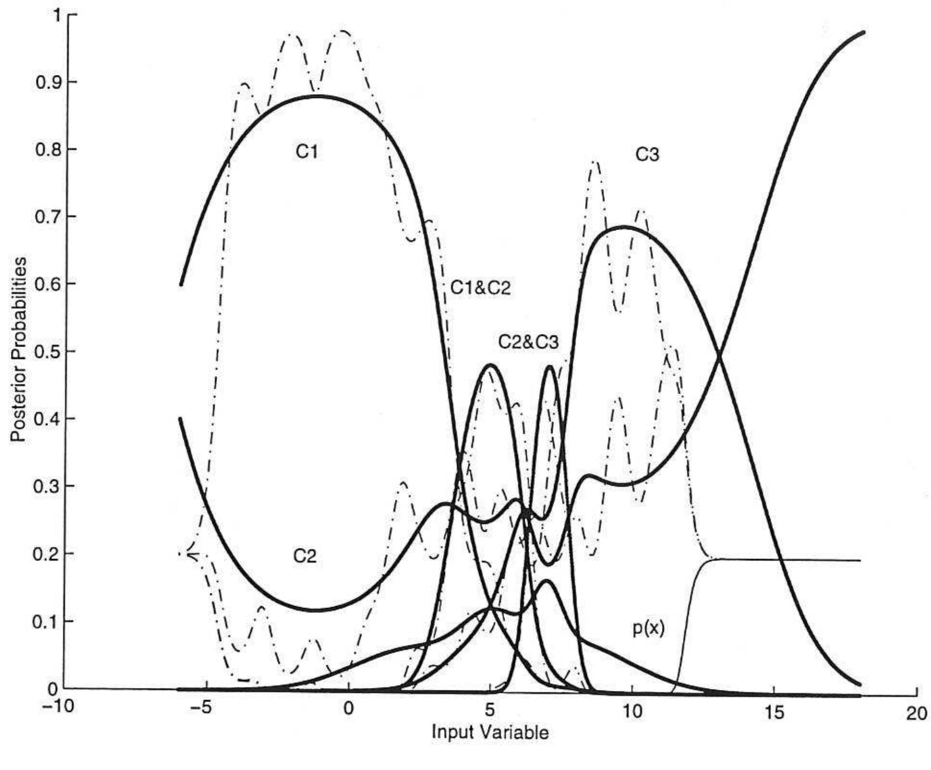


Figure 23. A graph showing the estimation of posterior probabilities by a radial basis function network. Note that only the output space has been partitioned to allow the joint probability functions to be estimated.

The data density outside of the range $[-3,+12]$ is low, giving inaccurate predictions of the posterior probability functions as expected. The lack of regularisation allows over-learning of the data and is indicated by the considerable curvature of the estimated probability functions.

Figure 24 shows the estimated posterior probabilities with second-order derivative regularisation.

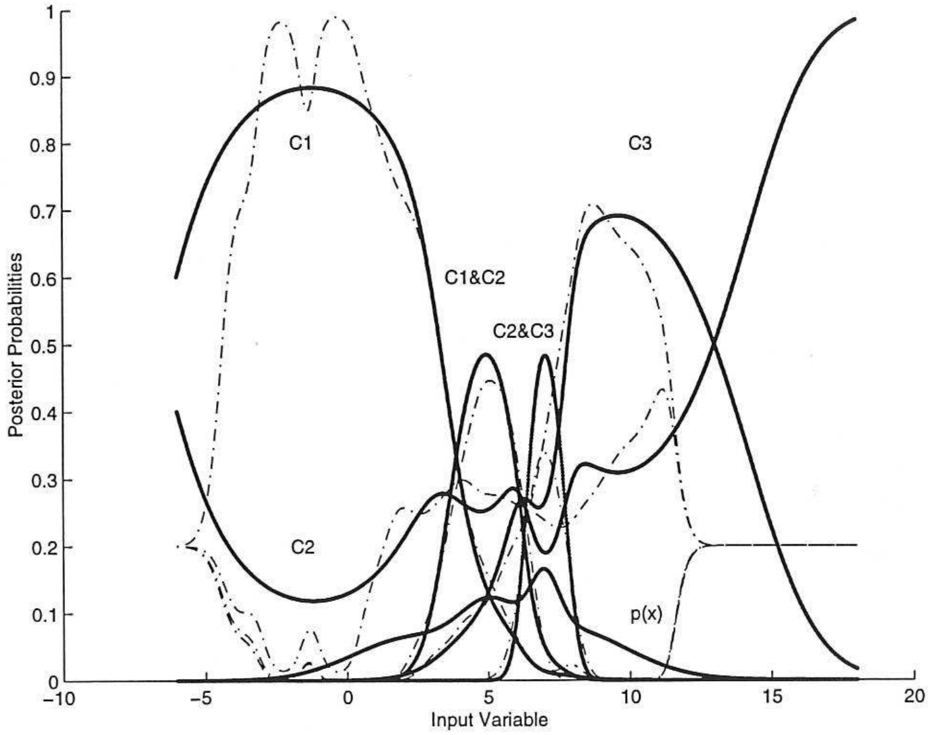


Figure 24. A graph showing the estimation of posterior probabilities by a radial basis function network using second-order differential regularisation as explained in the text.

Note that the approximated functions are considerably smoother in the region of higher data density.

2.7.6 Fault Protocol

The protocol used for generating the fault data used in exploring the probability estimation and update problems is:

for each choice of Mach-Altitude co-ordinates,

> i) run the model at the NOP without any faults,
> ii) run the model with faults induced based upon the priors
> iii) compare NOP run with the possible fault run, then
> iv) find the maximum absolute percentage deviation across the time-trace of the variable for each indicator; if this exceeds the limit set for that particular variable, flag a fault.

For the experiments detailed here a data set comprising 800 training patterns and 300 test patterns was used. The faults actually induced were:

i) Fuel decay (increase), C1

ii) TGT increase, C2
iii) N3 decay, C3
iv) Fuel decay (increase) and TGT increase, C1 & C2
v) Fuel decay (increase) and N3 decay, C1 & C3
vi) TGT increase and N3 decay, C2 & C3

Faults are not induced for a number of cases which are thus considered normal. From these normal cases, a fraction is assigned a false alarm or no fault found. The faults are induced according to the prior probabilities. The following list of priors was used in the preliminary experiments featured in this document. These could be adjusted to give a more realistic spread of fault/normal conditions.

Condition-Class	Prior Probabilities
C1	0.15
C2	0.15
C3	0.15
C1 & C2	0.05
C1 & C3	0.05
C2 & C3	0.05
No fault / normal(N):	0.3
No fault found (NFF)	0.1

Table 4. The induced condition-class priors

As mentioned in step iv) of the protocol, faults are assigned on the basis of range checking compared with the fault-free model for the five parameters detailed above. The fault ranges of all five variables are all set at ($\pm$10%) for simplicity; they can all be set at separate limits if required. We remark that such a detection scheme is crude and is simply used for illustration.

2.7.6.1 The fault vector coding scheme

The fault vectors are coded using a five bit input string which indicates the occurrence or non-occurrence of a limit-trip on each of the monitored variables. A seven-bit output is used to indicate the following fault conditions: C1, C2, C3, C1&C2, C1&C3, C2&C3, N. The case C1&C2&C3 is prevented from happening by not allowing all three faults to be induced at any one time, i.e. the condition is ignored. This is using one from many coding. Thus, the three condition-classes can be indicated separately or in pairs or the plant can be operating normally. The target vector represents verified fault / no fault occurrences. This form of input/output data indicates a binary heteroassociative problem.

An example data vector is (10111, 0000100). The first set of five bits is the alarm indicator set which signifies that there are trip deviations in fuel flow, N3, P30, and T30 but TGT is within range. It forms the input pattern to the neural network. The second set of seven bits indicates the verified status. It represents the training input and represents a verified fault in C1 & C3, i.e., there is a verified fuel problem and a problem with the HP turbine (N3). The network has to develop a mapping between input alarms and verified faults. The network used is a RBFN with soft-max output layer and second order derivative regularisation, trained using simple, steepest gradient descent.

2.7.6.2 Results

The empirical training set and test set probabilities were computed from the data files. These were found by computing the relative frequencies for each input vector.

2.7.6.2.1 1.7.6.2.1 The Theoretical Maximum Accuracy

A linear network, incorporating a softmax output layer to allow for the representation of probabilities, was trained and tested with the 800/300 set using a variety of initial weights. The network was used to predict the most likely fault(s). The best performance was a prediction accuracy of 64% and this varied very little for different choices of the initial weight set; this indicated that the minimum mean-squared error for such a system had been achieved. Thus, comparing with the theoretical maximum prediction accuracy of 87.7% (calculated directly), it is clear that this is not a trivial problem solvable using a simple, linear network. As the complexity of the fault data increases, it is likely that the linear system will have an even poorer performance. Measurement noise and quantisation of inputs may possibly reduce accuracy further.

A single run of the regularised RBFN was carried out with the 800/300 data set to assess the network's accuracy in probability estimation. The prediction accuracy for the test set for the most likely fault scenario was 87.7%, the maximum possible; that is, if faults were chosen on the basis of probability magnitudes. How can this be? This is because the RBFN models the probability distribution and only the maximum probability for each of the binary input vectors is required. There may be a large error in the estimates of the probabilities which does not affect the MAP decision as long as the probability of the most probable prediction exceeds the others by a small margin. In other words, the winning probability only has to be largest. Thus, the probability density function may not be very accurate or representative of the underlying distribution of fault vectors but still allow the maximum achievable accuracy.

The test set results are shown in Table 3. Only 11 out of a possible 32 states occurred with this run; NB this would change for different values of the fault detection thresholds. For each input, the actual probabilities (relative frequencies) of occurrence are shown together with those predicted by the network. The column labelled p(x) shows the data distribution (relative frequency) of the patterns. This is included to illustrate the variation of accuracy with data density.

The RBFN network was then trained and tested with an 800/300 data set based upon the 11/32 binary input vectors encountered in the above fault-induction experiments. This set was devised for use as a calibration check. This time, a single *unambiguous* input was assigned to each input on the basis of the maximum probabilities encountered in the previous experiments. The input vectors were distributed approximately according to the frequency of occurrence encountered above. Thus, the max theoretical accuracy of correct diagnoses was 100% The RBFN achieved this 100% target.

		C1	C2	C3	C1&C2	C1&C3	C2&C3	N	p(x)
input	actual	0	0	0	0	0	0	1	0.3100
00000	predicted	0.005029	0.012104	0	0	0	0	0.982867	
input	actual	0	0	0	0	0	0	1	0.0433
00100	predicted	0.164490	0.202020	0.008506	0.007712	0.006416	0.006438	0.604418	
input	actual	0	0.7910	0	0	0	0	0.2090	0.2233
01000	predicted	0.000097	0.883493	0.000068	0.000187	0.000001	0.000009	0.116146	
input	actual	0	0	0.6667	0.3333	0	0	0	0.0100
01011	predicted	0.010755	0.014644	0.850851	0.008370	0.011507	0.092996	0.010877	
input	actual	0	0	0.8750	0	0.1250	0	0	0.0267
01111	predicted	0.000516	0.000375	0.922690	0.000667	0.001522	0.073864	0.000366	
input	actual	0.8750	0	0	0	0	0	0.1250	0.1333
10000	predicted	0.941444	0.000090	0.000002	0.000078	0.000001	0.000001	0.058384	
input	actual	1	0	0	0	0	0	0	0.0133
10010	predicted	0.558011	0.053072	0.025421	0.051029	0.029564	0.015743	0.267161	
input	actual	0	0	0	0	1	0	0	0.0500
10111	predicted	0.000875	0.000359	0.005837	0.000367	0.991319	0.000811	0.000432	
input	actual	0	0	0	1	0	0	0	0.0533
11000	predicted	0.000629	0.000737	0.000058	0.998410	0.000010	0.000011	0.000144	
input	actual	0	0	0	1	0	0	0	0.0033
11010	predicted	0.087575	0.085436	0.094690	0.576361	0.042954	0.054367	0.058617	
input	actual	0	0	0.6000	0	0.0500	0.3500	0	0.1333
11111	predicted	0.000094	0.000036	0.719065	0.000333	0.021754	0.258647	0.000070	

Table 5. Test set results. The actual probabilities are the relative frequencies of the classes in the data set. The predicted results are those of the RBFN run. P(x) indicates the relative data frequency.

Note that not all of the condition-class rankings are correct; where they are incorrect (e.g. alarm condition 01111) the relative frequency of data is low. Some type of error measure is needed which allows comparison of rankings between experiments.

2.7.6.2.2 1.7.6.2.2 Discussion

As expected, the underlying statistics of the training set population are estimated by the RBFN system; where the data relative frequency is low the estimated probabilities become inaccurate. For both training and test sets, the same decision as predicted by the actual probabilities will be taken in all cases using the predicted probabilities. Some states may not occur in the fault diagnosis procedure, e.g. 00001 which signifies that T30 has changed without any concomitant changes in other monitored variables.

The eleven observed states were as follows:
State 00000 indicates that no fault has occurred.
State 00100 indicates that a fault has occurred in N3 only. The probability estimations point to the fact that a NFF condition is actually the case because an actual N3 decay fault usually has an effect on other parameters such as fuel flow and TGT. Note the low data density.
State 01000 indicates that a fault has occurred with TGT only. It is highly likely that the fault lies with the TGT thermocouple because no other parameter changes have been noted. There is also the possibility that no fault has occurred.
State 01011 indicates that, as well as the thermocouple changes, there are concomitant changes in P30 and T30 indicating that the N3 shaft may be involved (N3 or N3&TGT). In both cases, the N3 activity does not show up as a fault. Note that the data density is low in this case.
State 01111 indicates that all faults are triggered except for the fuel flow. This is highly indicative of an N3 fault but the data density is low indicating that a fuel flow problem usually occurs as well. This is supported by the higher data density associated with state 11111.
State 10000 indicates that there is a fuel flow problem. When this occurs alone it is rarely a consequence of any other actual fault. However, there is a possibility that a NFF condition has occurred.
State 10010 indicates both a fuel flow and P30 problem. According to the test set statistics, it is always a fuel flow problem.
State 10111 indicates that all faults are triggered except for TGT. A low data density indicates that TGT is usually associated with N3 (alarm pattern 11111). Where TGT is omitted, WF and N3 together are expected according to the training or test data.
State 11000 indicates that WF and TGT have occurred together. The actual faults are WF and TGT because the occurrence of N3 usually has a 'knock-on' effect.
State 11010 indicates that it is again a conjunction of WF and TGT but the P30 fault is anomalous as shown by the data density.
State 11111 is either indicative of N3 alone or N3 and TGT. Note the ratio of occurrences of approximately 3:1 is commensurate with the ratio of prior probabilities of 0.15:0.05 or 3:1.

This preliminary empirical investigation indicates that fault induction and detection using the aircraft engine model will provide data suitable for testing and extending the posterior knowledge inclusion model. The fault induction and detection process is to be refined so that meaningful posterior probability hierarchies will be generated.

2.8 Representing Non-Exclusive Probabilities by Exclusive Probabilities

The motivation for seeking a partition of the input space is that we need to expand the space to estimate all of the probabilities required for the update equation. In other words, the class dependencies indicated by more than one 'on bit' in the target vector.

A partition of classified input space may be achieved by specifying that the class intersections are pairwise disjoint, for example C_i' only contains data points that belong to C_i and not $C_i \cap C_j$ etc. Similarly, $\left(C_i \cap C_j\right)'$ only contains data points that belong to $C_i \cap C_j$ and not $C_i \cap C_j \cap C_k$ etc. This will ensure a partition of the space with disjoint sets as required (e.g. $C_i' \cap \left(C_i \cap C_j\right)' = \phi$). The 'dash' notation is used throughout to indicate partition members which comprise the entire sample space.

Now, the original formula for the union of sets in terms of set intersections can be specified in purely additive terms:

$$\begin{aligned} P\left(\bigcup_{r=1}^{N} C_r \mid \mathbf{x}\right) = & \sum_{i=1}^{N} P\left(C_i' \mid \mathbf{x}\right) \\ & + \sum_{\substack{i=1,j=2 \\ j \neq k}}^{N} P\left(\left(C_i \cap C_j\right)' \mid \mathbf{x}\right) \\ & + \sum_{\substack{i=1,j=2,k=3 \\ i \neq j \neq k}}^{N} P\left(\left(C_i \cap C_j \cap C_k\right)' \mid \mathbf{x}\right) \\ & \vdots \\ & + P\left(\left(C_1 \cap C_2 \cap \ldots \cap C_N\right)' \mid \mathbf{x}\right) \end{aligned} \qquad (15)$$

It is required to prove that the two representations are formally equivalent.

For $P\left(\bigcup_{r=1}^{N} C_r \mid \mathbf{x}\right)$, it must be shown that the probability term representing each disjoint region only occurs once in the sum.

For each C_i', $C_i' \subseteq C_i$ and $P\left(C_i \mid \mathbf{x}\right)$ occurs only once in the summation $\sum_{i=1}^{N} P\left(C_i \mid \mathbf{x}\right)$ and in the expression $P\left(\bigcup_{r=1}^{N} C_r \mid \mathbf{x}\right)$ because all other class segments consist of two or more intersecting classes and, hence, do not have single class sets as subsets. The first set of terms of $P\left(\bigcup_{r=1}^{N} C_r \mid \mathbf{x}\right)$ become $\sum_{i=1}^{N} P\left(C_i' \mid \mathbf{x}\right)$ where $\sum_{i=1}^{N} P\left(C_i' \mid \mathbf{x}\right) \leq \sum_{i=1}^{N} P\left(C_i \mid \mathbf{x}\right)$. Introducing the notation $C(n,k)$ which signifies a combination of k objects selected from n. So for C_i' there is only a single set and a single way of selecting that set so n=1 and k=1 giving $C(1,1) = 1$.

For two or more intersecting classes the non-overlapping region of interest is $\left(C_i \cap C_j\right)'$. Now, $\left(C_i \cap C_j\right)' \subset C_i, C_j, C_i \cap C_j$ so, for

$P(C_i|\mathbf{x}) + P\left(C_j|\mathbf{x}\right) - P\left(C_i \cap C_j|\mathbf{x}\right)$

where all three terms all include the term $P\left(\left(\left(C_i \cap C_j\right)' |\mathbf{x}\right)\right)$, the resultant term will be $2P\left(\left(C_i \cap C_j\right)' \middle| \mathbf{x}\right) - P\left(\left(C_i \cap C_j\right)' \middle| \mathbf{x}\right) = P\left(\left(C_i \cap C_j\right)' \middle| \mathbf{x}\right)$ i.e. the term only occurs once. Here, the number of terms is given by $C(2,1) - C(2,2) = 1$ where each of the singleton terms $P(C_i|\mathbf{x})$ and $P\left(C_j|\mathbf{x}\right)$ can be selected once from a set of 2 (because $\left(C_i \cap C_j\right)' \subset C_i, C_j$), hence $C(2,1)$,and the term $P\left(\left(C_i \cap C_j\right)' \middle| \mathbf{x}\right)$ involving 2 sets can only be selected once from $P\left(C_i \cap C_j|\mathbf{x}\right)$, hence $C(2,2)$. Continuing this argument for $P\left(\left(C_i \cap C_j \cap C_k\right)' \middle| \mathbf{x}\right)$,

$\left(C_i \cap C_j \cap C_k\right)' \subset C_i, C_j, C_k, C_i \cap C_j, C_i \cap C_k, C_j \cap C_k, C_i \cap C_j \cap C_k$ and so the number of terms including $P\left(\left(C_i \cap C_j \cap C_k\right)' \middle| \mathbf{x}\right)$ will be given by

$C(3,1) - C(3,2) + C(3,3) = 3 - 3 + 1 = 1$. For the general case, n class intersection terms occur $N_n = C(n,1) - C(n,2) + C(n,3) - C(n,4) + \ldots + (-1)^{n+1} C(n,n)$ times. It is required to prove that $N_n = 1$, that is, each term only occurs once.

$$
\begin{aligned}
N_n &= \sum_{k=1}^{n}(-1)^{k+1}C(n,k) \\
&= (-1)\sum_{k=1}^{n}(-1)^{k}C(n,k) \\
&= (-1)\sum_{k=1}^{n}(1)^{n-k}(-1)^{k}C(n,k) \\
&= 1+(-1)+(-1)\sum_{k=1}^{n}(1)^{n-k}(-1)^{k}C(n,k) \\
&= 1+(-1)(-1)^{0}+(-1)\sum_{k=1}^{n}(1)^{n-k}(-1)^{k}C(n,k) \\
&= 1+(-1)+(-1)\sum_{k=0}^{n}(1)^{n-k}(-1)^{k}C(n,k) \\
&= 1+(1-1)^{n} \\
&= 1
\end{aligned}
$$

Here, the expansion of $(a-b)^n = \sum_{k=1}^{n}(-1)^k a^{n-k}b^k$ has been used with $a=b=1$.

Now any probabilistic function of the possibly overlapping classes $U=\{C_1,\ldots C_N\}$ can be replaced with an equivalent disjoint set $U'=\left\{C_1',\ldots,C_N',(C_1\cap C_2)',\ldots,(C_{N-1}\cap C_N)',\ldots,(C_1\cap,\ldots,\cap C_N)'\right\}$ which forms a partition of the input space. Equation (15) can now be written in terms of Bayes theorem:

$$
\begin{aligned}
P\left(\bigcup_{r=1}^{N} C_r|\mathbf{x}\right) &= \sum_{i=1}^{N}\frac{P(C_i')P(\mathbf{x}|C_i')}{P(\mathbf{x})} \\
&+ \sum_{\substack{i=1,j=2\\ j\neq k}}^{N}\frac{P\left((C_i\cap C_j)'\right)P\left((C_i\cap C_j)'|\mathbf{x}\right)}{P(\mathbf{x})} \\
&+ \sum_{\substack{i=1,j=2,k=3\\ i\neq j\neq k}}^{N}\frac{P\left((C_i\cap C_j\cap C_k)'\right)P\left(\mathbf{x}|(C_i\cap C_j\cap C_k)'\right)}{P(\mathbf{x})} \\
&\vdots \\
&+\frac{P\left((C_1\cap C_2\cap\ldots\cap C_N)'\right)P\left(\mathbf{x}|(C_1\cap C_2\cap\ldots\cap C_N)'\right)}{P(\mathbf{x})}
\end{aligned}
$$

2.9 Interim Review

It has been stated that, in general, condition monitoring involves the detection of anomalous conditions that arise during the operation of some plant or process. The indication of the most likely fault and its estimated probability by a fixed pattern recognition system is not necessarily the end-point. Condition monitoring is, or should be, a closed-loop process involving an end-user. The end-user ultimately decides how to use the information generated by the condition monitoring system. The end-user may, in turn, require a mechanism of incorporating his or her observations into the condition monitoring system for a more accurate diagnosis. The incorporation and utilisation of posterior knowledge presents a difficult problem. This research has attempted both to articulate the problem and to provide a framework for its solution. It is clear that more work is required in this area. The contrived example illustrates some of the issues involved in the integration of posterior knowledge within the human / machine diagnostic cycle as fault evidence is accumulated. Three phases of the fault diagnosis cycle have been identified:

(i) fault diagnosis and isolation to provide fault prediction data,
(ii) probability estimation to provide the fault hierarchy, and
(iii) posterior knowledge inclusion to provide a revised fault hierarchy.

Phases (i) and (ii) are covered by many condition monitoring schemes. Some of the theory of phase (iii) has been explored so far in this document. Chapter 2 will expand upon the theme and discuss simulation studies.

The problem of posterior knowledge representation is a difficult one and further work is needed to increase the scope beyond just excluding classes on the basis of external observations. Both the method of knowledge representation and the posterior probability update problem are independent of the method used for probability estimation; this is because of the general framework based upon set theory.

Using the specified probabilistic framework, the posterior knowledge integration problem has been reduced to an m from n *estimation* problem. Furthermore, The m from n estimation problem has been reduced to a 1 from 2^n problem by expansion of the output space. Pre-processing may be required to reduce the combinatorial explosion. It is clear that probability distribution estimation methods must give sufficiently accurate estimates to maintain condition class hierarchies; the estimation problem has been explored further using an established neural network technology. The radial basis function network has the combined features of cross-entropy cost, a soft-max layer and second order derivative regularisation.

2.10 Assessing the utility of posterior knowledge integration

Of primary interest here is the development of more informed maintenance strategies which reduce the amount of maintenance required. By using posterior knowledge, revised fault probabilities will lead to a more efficient sub-unit checking order. Without posterior knowledge integration, the probability estimates—generated from the condition monitoring data—give a fixed fault *ranking* via the posterior probabilities of fault occurrences. The theory discussed so far indicates that revised posterior probabilities may alter the fault ranking and give a more accurate prediction of the current fault scenario. How can posterior knowledge integration enhance maintenance strategies in practice? Furthermore, how effective is the use of posterior knowledge and how can this be quantified? These questions and related issues will be explored in the remainder of this document.

In general, in a condition-monitoring situation, there will be a *search path* followed by maintenance engineers to detect and isolate all current faults. In terms of aircraft maintenance, this entails using all available fault indicators and maintenance experience/procedures to detect the faulty line-replaceable units (LRUs). The posterior knowledge integration technique has been developed to reduce search-path lengths during maintenance.

Posterior knowledge integration is an abstract technique designed, in theory, to be a post-processing stage with general applicability to a wide-range of condition monitoring techniques which produce probabilistic FC data. Consequently, the assessment of this technique should be, at least initially, *context-free*. In other words, its utility should be indicated without reference to a *specific* condition monitoring situation. A technique for context-free simulation has been developed for this purpose. Using context-free simulations means that the results are not limited to a specific set of FC relative frequencies. Using such a specific set may give a misleading impression of the possible utility of posterior knowledge integration.

Context-free simulations use a number of individual sets of relative frequencies to explore how the posterior knowledge technique functions across a range of conceivable condition monitoring situations. Each simulation is based upon a single set of relative frequencies generated at random; this set represents some possible set of condition monitoring data for the lifetime of a single plant such as an individual aircraft. By applying the integration technique to each of the relative frequency sets, an *ensemble* of results is obtained which can be summarised using appropriate ensemble statistics. Performance measures will be discussed in Section 2.2. The ensemble results may represent many individual items of the same plant type (e.g. a *fleet* of aircraft) or, more generally as applied in this work, a heterogeneous set of plants. For the purposes of simulation, multiple instantiations of a plant may be characterised by using a narrow probability distribution for the relative frequency vectors. In other words, relative frequencies of individual plant items of the same basic type are not expected to vary to the extent of those of different plants.

For a *single simulation* (realisation) from an ensemble, a set of FC frequencies are generated at random which represent a possible estimated set from the real-world. The real-world counterpart is shown schematically in figure 25. Here, in the context-free simulations, the input features do not exist explicitly because the FC frequencies may be associated with any possible input and represent some possible situation. The relative frequencies represent the probabilities of fault *scenarios* for the plant. An example scenario may be when sub-units (LRUs) 1 and 5 are faulty and all the rest are operating normally.

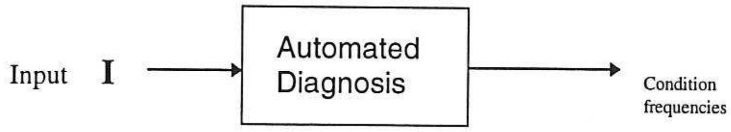


Figure 25. A schematic diagram of a 'real-world' counterpart to a single context-free simulation. A set of observations is fed into a pattern recognition system which generates a set of FC frequencies corresponding to the predicted probabilities of fault scenarios. These probabilities will then be used to identify the actual fault scenario.

A fixed set of FC frequencies are generated for a given simulation from the ensemble. Each of the fault scenarios represented by a non-zero probability is taken in-turn as the actual fault scenario to be detected. The FC frequencies are used systematically—in conjunction with the probability update equation—to identify the actual fault scenario. Finally, after each scenario, relevant information is recorded which allows the single simulation measures (e.g. for a single aircraft) or the ensemble measures (e.g. for a fleet) to be calculated. Each fault scenario of the individual simulation entails a fault search which results in a fault path representing the number of inspections required before all faults are identified.

Figure 26 illustrates what happens for a single fault scenario awaiting identification using posterior knowledge integration. The posterior probabilities of occurrence for each fault are reconstructed using the scenario frequencies. These posterior probabilities are ranked in descending order of magnitude and represent the probability of a given fault occurring. The sub-unit with the highest fault probability is chosen and inspected for that fault. The posterior knowledge following inspection is then used to specify the actual form of the probability update equation which depends upon the FCs to be included or excluded. The probability update equation is used to give the revised posteriors. If the fault scenario has been identified then the search is halted. Otherwise, the revised posteriors are used and the process is continued until the scenario is identified or the maximum number of sub-units is inspected.

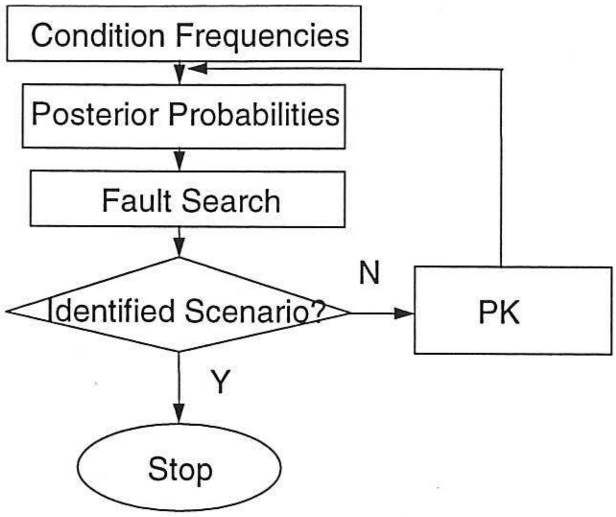


Figure 26. A flow diagram of the identification process for a single binary-coded fault scenario. A number of scenarios is identified for a given set of FC frequencies generated for a single simulation.

In order to show any possible utility of the posterior knowledge integration technique, a comparison has to be made with a baseline method which does not use this external evidence. A simple comparison method is to fix the FC posterior probabilities as they are obtained from the condition monitoring system. The fault search entails inspecting the sub-units in descending order which is equivalent to a simple renormalisation of the MAP decision. Here, the comparison technique is referred to as the *baseline, or MAP, method.*

For comparison purposes, the posterior knowledge is included *systematically*, that is, the fault status of an inspected unit is fed into the probability update equation to give the new posteriors after each inspection. Both the posterior knowledge and baseline methods are carried out for each scenario until all faults are found, i.e. the scenario is identified. In practice, the search could be halted when all posterior probabilities fall below a given threshold.

3 Simulation Studies: Theory

3.1 Simulation Protocol.

Binary vectors are used to represent the *reading status* and *actual fault status* of system sub-units. The reading status is a set of sub-system alarms coded as a binary string in which '1' entries represent triggered alarms and '0' entries represent correct operation. The reading vector, $\mathbf{x}_k$ provides the input to a probability estimation/classification sub-system (as part of an FDI system) for training an estimation system or probabilistic fault isolation. In the latter case, $\mathbf{x}_k$ is applied and a set of probabilities are generated (see Figure 27) which assist the user in making the optimal choice (given some criterion of optimality) of system sub-unit to be inspected. In other words, the set of fault readings $\mathbf{x}_k$ gives rise to a set of condition probabilities indicating the relative frequency of faults associated with the readings.

The actual fault status of a system is indicated by a binary vector, $\mathbf{S}_i$, which has a '1' entry for each sub-unit shown (on inspection) to be faulty; the binary vector of actual fault states is known as a fault *scenario* and provides the target for training or fault isolation. The probabilistic FDI system is based upon a mapping, $\mathbf{x}_k \rightarrow \mathbf{S}_i$ between the binary reading or alarm space and the binary scenario space (actual fault status space).. Thus, there is a set of data representing a mapping between input sensors and verified actual faults. Statistical information can be gathered from this data which indicates the probability of fault scenarios given fault indicator readings. For example, the pattern pair (1100, 1000) indicates faults in sub-system 1 and 2 but, subsequently, only faults were found in sub-system 1 as represented by the scenario 1000. The fault indicated in sub-system 2 in this case is an example of a NFF situation. The output for a given input, $\mathbf{x}_k$ is a set of posterior scenario probabilities, $P(\mathbf{S}_i|\mathbf{x}_k)$ conditional upon the input. Figure 27 illustrates the association of scenario probabilities with a given reading vector $\mathbf{x}_k$. $\mathbf{S}_{\delta_{kj}}$ is the jth scenario associated with the kth reading vector. Scenarios represent the disjoint fault classes of figure 19 in Section 1.7. To get the *total* posterior probability of a FC given an input vector $P(C_i|\mathbf{x}_k)$, sum the exclusive posterior scenario probabilities of scenarios containing the required fault class. Thus

$$P(C_i|\mathbf{x}_k) = \sum_j P(\mathbf{S}_{\delta_j}|\mathbf{x}_k)$$

where each scenario, $\mathbf{S}_{\delta_j}$ contains the ith sub-unit represented as faulty.

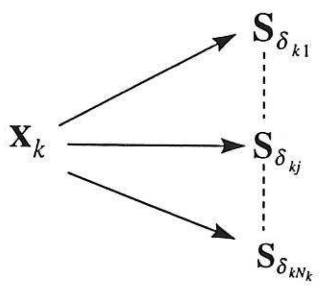


Figure 27. The kth reading vector gives rise to a number of associated scenarios. Each scenario has a conditional probability associated with it.

For a binary NFF implementation, the number of readings must equal the number of sub-units, i.e. there is a one to one correspondence between the alarms and the sub-units. This is assumed here without loss of generality because it is must be possible to find a suitable mapping between the measured variables and alarm states, $\mathbf{x}_k$ otherwise certain alarm states would be indistinguishable and therefore redundant.

Here a NFF is taken to be when a reading indicates a faulty sub-unit but the corresponding sub-unit is not faulty, i.e a bit is set in the alarm vector $\mathbf{x}_k$ which is not set in the corresponding scenario vector $\mathbf{S}_i$ which currently represents the actual fault scenario (system fault status).

Binary data is generated in the form of pattern pairs $(\mathbf{x}_k, \mathbf{S}_i)$ where an input reading, $\mathbf{x}_k$ is associated with an output vector $\mathbf{S}_i$ representing the associated *actual* fault state of the system. If a *forward simulation* is used to generate data, the scenario prior probabilities, $P(\mathbf{S}_i)$ are used to generate fault scenarios and the likelihoods, $P(\mathbf{x}_k|\mathbf{S}_i)$ are used to choose the reading vectors. Bayes' Theorem:

$$P(\mathbf{S}_i|\mathbf{x}_k) = \frac{P(\mathbf{x}_k|\mathbf{S}_i)P(\mathbf{S}_i)}{P(\mathbf{x}_k)}, \qquad P(\mathbf{x}_k) = \sum_{i=1}^{N} P(\mathbf{x}_k|\mathbf{S}_i)P(\mathbf{S}_i)$$

can be used to calculate the expected posterior probabilities as the number of pattern pairs $(\mathbf{x}_k, \mathbf{S}_i)$ grows countably large. The exclusive scenario probabilities are represented in matrix form,

$$P(\mathbf{S}|\mathbf{x}) = \begin{bmatrix} P(\mathbf{S}_1|\mathbf{x}_1) & \cdots & P(\mathbf{S}_i|\mathbf{x}_1) & \cdots & P(\mathbf{S}_N|\mathbf{x}_1) \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ P(\mathbf{S}_1|\mathbf{x}_k) & \cdots & P(\mathbf{S}_i|\mathbf{x}_k) & \cdots & P(\mathbf{S}_N|\mathbf{x}_k) \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ P(\mathbf{S}_1|\mathbf{x}_{M1}) & \cdots & P(\mathbf{S}_i|\mathbf{x}_M) & \cdots & P(\mathbf{S}_N|\mathbf{x}_M) \end{bmatrix}$$

with each row,

$$P(\mathbf{S}|\mathbf{x}_k) = \left[P(\mathbf{S}_1|\mathbf{x}_k) \quad \ldots \quad P(\mathbf{S}_i|\mathbf{x}_k) \quad \cdots \quad P(\mathbf{S}_N|\mathbf{x}_k)\right]$$

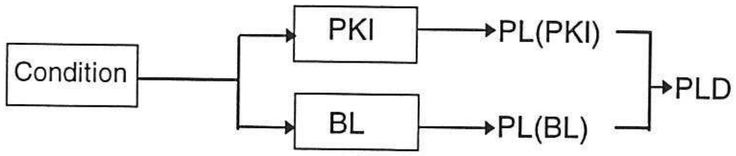
representing the scenarios associated with a single reading vector. The reading priors are represented by $P(\mathbf{x}) = \left[P(\mathbf{x}_1) \quad \cdots \quad P(\mathbf{x}_k) \quad \cdots \quad P(\mathbf{x}_M)\right]$.

Thus, there is a vector of scenario posterior probabilities $P(\mathbf{S}|\mathbf{x}_k)$ associated with each alarm vector $\mathbf{x}_k$ which indicate the possible current scenario. These exclusive posterior probabilities are converted into non-exclusive fault class probabilities $P(C_p|\mathbf{x}_k)$ for each sub-unit. The PKI and baseline methods use these probabilities to make informed choices about probable faulty sub-units. The simulations presented in this document use the calculated posteriors for convenience to illustrate the use of PKI without loss of generality. The calculated posteriors represent some population of fault classes with parameters which will be estimated from a sample.

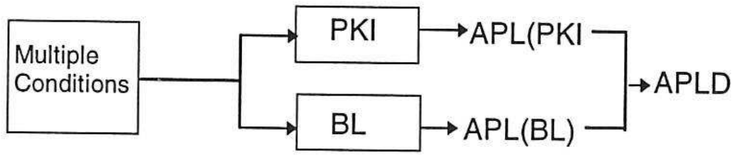
3.2 Performance measures

A number of performance measures are possible which allow a comparison between methods of scenario identification. Of direct interest and applicability are measures involving the *path length* (PL) or *path length difference* (PLD) between the same scenario identified using the two techniques of PKI and the baseline comparison method. The path-length is the number of sub-unit inspections required to find all faults. The path length difference is the difference between the baseline and posterior knowledge integration techniques for a given scenario. A positive path length difference indicates that the MAP method, henceforth referred to as the baseline method, took more steps (sub-unit inspections) to identify the scenario as compared to the PKI method and vice versa. Each sub-unit inspection increases the path length by one. One of the objectives here is to reduce the path-length to a minimum thereby possibly reducing maintenance costs. The other objective—the most important—is to reduce the number of no fault found (NFF) incidents. The measure of no fault found number (NFFN) indicates the number of NFF incidents encountered during the identification of a fault scenario. The overall measures of a FDI technique (PKI or baseline) are the average path-length (APL) and the average number of NFF incidents encountered per scenario (ANFFN). These measures are obtained from data covering every possible alarm situation and every possible fault scenario which occurred in a given (simulated) data set.

Results are recorded for each scenario, each simulation and each ensemble for a given set of simulation FCs. Thus, in the context of aircraft maintenance, a set of measures is calculated for each situation (FC), each aircraft and each fleet. Figure 28 shows this schematically for a single FC and a single simulation.



(a)



(b)

Figure 28. (a) for a single FC, the posterior knowledge and baseline methods are applied to give path-lengths signifying the number of sub-unit (LRU) inspections required to identify the scenario. The path length difference is then calculated from the two path lengths. (b) For multiple FCs, the average path lengths and average path length differences may be calculated.

The path length difference ranges between $-(n-1)$ through 0 to $(n-1)$ where n is the number of sub-units. One *ensemble* measure is to count the individual PLD's to give quasi-histograms of PLD frequencies. However, the counts must be weighted to reflect the relative frequencies of the scenarios which gave rise to those path length differences. For a single scenario, the path length difference is calculated and weighted by the scenario probability. The weighted PLD counts are then presented in a bar chart as a quasi-histogram. Positive PLD's indicate that the baseline method requires more sub-unit inspections to identify the scenario. Thus, a quasi-histogram skewed to the right indicates that the posterior knowledge method is more efficient in scenario identification, requiring fewer sub-unit inspections. This may translate into maintenance savings. In the aircraft industry, this would mean fewer LRU inspections. LRU's are usually removed and replaced which can be a costly process in terms of LRU recertification. NFF situations mean that non-faulty LRU's have to be tested thoroughly prior to re-use.

3.3 Context-Free Simulation: The Algorithm

Each possible alarm (reading) vector is gone through in turn and taken to be the current reading vector which would occur with a relative frequency of approximately $P(\mathbf{x}_k)$ as given by the calculated reading prior. For a given reading vector, each scenario is treated in turn as the *actual* fault scenario. Both the PKI and baseline methods are then used to identify the actual scenario.

The optimal situation is when the scenarios are identified by choosing each faulty unit in turn; thus the *minimum* path length for a *single* scenario—to be identified by locating *all* of the faulty sub-units, if any—is the total number of faults in that particular scenario. Each scenario will have an associated minimum path-length. Furthermore, each scenario, which has non-zero probability, will have an associated path-length for both of the PKI and BL methods respectively when they are applied in the scenario identification process. The vectors $\mathbf{l}_{opt}^{k}$ $\mathbf{l}_{pki}^{k}$ and $\mathbf{l}_{bl}^{k}$ represent the scenario path-lengths for the kth reading vector with the optimal, PKI and BL cases respectively. For example $\mathbf{l}_{pki}^{k} = \left[l_{pki,1}^{k} \quad \cdots \quad l_{pki,i}^{k} \quad \cdots \quad l_{pki,N}^{k}\right]$ where $l_{pki,i}^{k}$ is the path-length (number of sub-unit inspections) required to find all faults when the PKI technique is applied to the ith fault scenario given the kth reading vector, $\mathbf{x}_k$

The path-length vectors $\mathbf{l}_{opt}^{k}$ $\mathbf{l}_{pki}^{k}$ and $\mathbf{l}_{bl}^{k}$ are then weighted using the posterior vector $P(\mathbf{S}|\mathbf{x}_k)$ for the particular reading vector to give

$$apl_{opt}^{k} = \left(\mathbf{l}_{opt}^{k}\right)^{t} P(\mathbf{S}|\mathbf{x}_k)$$

$$apl_{pki}^{k} = \left(\mathbf{l}_{pki}^{k}\right)^{t} P(\mathbf{S}|\mathbf{x}_k)$$

$$apl_{bl}^{k} = \left(\mathbf{l}_{bl}^{k}\right)^{t} P(\mathbf{S}|\mathbf{x}_k)$$

where apl_{opt}^{k} apl_{pki}^{k} and apl_{bl}^{k} are the average path lengths for the optimal path, the PKI derived path and the baseline method derived path respectively given the kth input vector.

There are analogous measures for the number of NFF incidents. Prior to applying PKI and the baseline method for a given reading and a given scenario, the possible NFFs are flagged. The total number of possible NFFs for a scenario is weighted by the scenario posterior (conditioned upon the input) to give the contribution to the average NFF occurrence for that scenario. This gives a comparison for the PKI and baseline techniques which provides more information about performance other than relative to each other. NFF incidents for a given scenario are also counted for the PKI and baseline techniques. For each reading, there is an associated NFF vector (each entry corresponding to a given scenario) indicating how many NFF units have been checked for that technique. The vectors $\mathbf{n}_{\max}^{k}$ $\mathbf{n}_{pki}^{k}$ and $\mathbf{n}_{bl}^{k}$ represent the scenario NFF counts (associated with the input vector $\mathbf{x}_k$) for the maximum possible numbers of NFF incidents, the numbers for the PKI technique and the numbers for the baseline technique. When a sub-unit is checked (search-path incremented), if it is a NFF, the corresponding count for that reading and scenario is incremented. For example $\mathbf{n}_{pki}^{k} = \left[n_{pki,1}^{k} \quad \cdots \quad n_{pki,i}^{k} \quad \cdots \quad n_{pki,N}^{k}\right]$ where $n_{pki,i}^{k}$ is the number of sub-unit inspections occurring as NFF incidents when the PKI technique is applied to the ith fault scenario given the kth reading vector, $\mathbf{x}_k$

The path-length vectors $\mathbf{n}_{\max}^{k}$ $\mathbf{n}_{pki}^{k}$ and $\mathbf{n}_{bl}^{k}$ are then weighted using the posterior vector $P(\mathbf{S}|\mathbf{x}_k)$ for the particular reading vector to give

$$anff_{\max}^{k} = (\mathbf{n}_{\max}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k)$$
$$anff_{pki}^{k} = (\mathbf{n}_{pki}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k)$$
$$anff_{bl}^{k} = (\mathbf{n}_{bl}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k)$$

where $anff_{\max}^{k}$ $anff_{pki}^{k}$ and $anff_{bl}^{k}$ are the average path lengths for the optimal path, the PKI derived path and the baseline method derived path respectively given the kth input vector.

Both the path-length and NFF incident measures for the three cases are conditioned upon the current input vector $\mathbf{x}_k$ To get a meaningful measure across the whole range of system conditions, the path-length and NFF incident measures are weighted by the calculated priors, $P(\mathbf{x}_k)$ associated with the inputs. The resulting measures give the overall averages per scenario for a given system. The final measures are:

the average scenario optimum path-length $aspl_{opt}$ given by

$$aspl_{opt} = E[apl_{opt}] = \sum_{k}^{M} apl_{opt}^{k} P(\mathbf{x}_k) = \sum_{k}^{M} (\mathbf{l}_{opt}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k) P(\mathbf{x}_k),$$

the average scenario PKI path-length $aspl_{pki}$ given by

$$aspl_{pki} = E[apl_{pki}] = \sum_{k}^{M} apl_{pki}^{k} P(\mathbf{x}_k) = \sum_{k}^{M} (\mathbf{l}_{pki}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k) P(\mathbf{x}_k),$$

the average scenario baseline path-length $aspl_{bl}$ given by

$$aspl_{bl} = E[apl_{bl}] = \sum_{k}^{M} apl_{bl}^{k} P(\mathbf{x}_k) = \sum_{k}^{M} (\mathbf{l}_{bl}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k) P(\mathbf{x}_k),$$

the average scenario maximum NFF incident number $asnff_{\max}$ given by

$$asnff_{\max} = E[anff_{\max}] = \sum_{k}^{M} anff_{\max}^{k} P(\mathbf{x}_k) = \sum_{k}^{M} (\mathbf{n}_{\max}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k) P(\mathbf{x}_k)$$

the average scenario PKI NFF incident number $asnff_{pki}$ given by

$$asnff_{pki} = E[anff_{pki}] = \sum_{k}^{M} anff_{pki}^{k} P(\mathbf{x}_k) = \sum_{k}^{M} (\mathbf{n}_{pki}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k) P(\mathbf{x}_k)$$

the average scenario PKI NFF incident number $asnff_{bl}$ given by

$$asnff_{bl} = E[anff_{bl}] = \sum_{k}^{M} anff_{bl}^{k} P(\mathbf{x}_k) = \sum_{k}^{M} (\mathbf{n}_{bl}^{k})^{t} P(\mathbf{S}|\mathbf{x}_k) P(\mathbf{x}_k)$$

3.4 PKI Optimality

For both the PKI method and the BL method, the optimum case is when all faults are found in unbroken sequence, that is $aspl_{pki} = aspl_{bl} = aspl_{opt}$ It follows that where the methods will differ is in the number of LRU inspections where a fault is not found.

There is thus a common core of the average scenario path lengths where the probability-weighted path lengths do not differ for the PKI and BL methods. The remaining ASPL contributions can be defined by,

$R_{pki} = aspl_{pki} - aspl_{opt}$

and

$R_{bl} = aspl_{bl} - aspl_{opt}$

respectively.

The fundamental idea behind PKI (in the unweighted case) is to look for the sub-unit with the highest posterior probability at each choice stage. The evidence at stage n of the PKI process has narrowed down the fault scenario to within the set, ε_n. Now, ε_n can be broken down into two exclusive sets (a partition) given new evidence at the n+1 th stage. Thus, the scenario either belongs to the set with a fault at a given position or otherwise as shown in Figure 29.

The stage n evidence is given by

$\varepsilon_n = \varepsilon_{n+1} \cup \varepsilon_{n+1}^c$

at stage n+1 and so,

$$\begin{aligned} P(\varepsilon_n) &= P(\varepsilon_{n+1}) + P(\varepsilon_{n+1}^c) - P(\varepsilon_{n+1} \cap \varepsilon_{n+1}^c) \\ &= P(\varepsilon_{n+1}) + P(\varepsilon_{n+1}^c) - P(\phi) \\ &= P(\varepsilon_{n+1}) + P(\varepsilon_{n+1}^c) \end{aligned}$$

where $\varepsilon_0 = U$ and $P(\varepsilon_0) = 1$.

Sub-units are chosen on the basis of having the largest posterior probability of being faulty. This implies that the probability of the chosen sub-unit being faulty is the lowest. Therefore, if the sub-unit is non-faulty, it adds the lowest weighting to the APL for NFF situations.

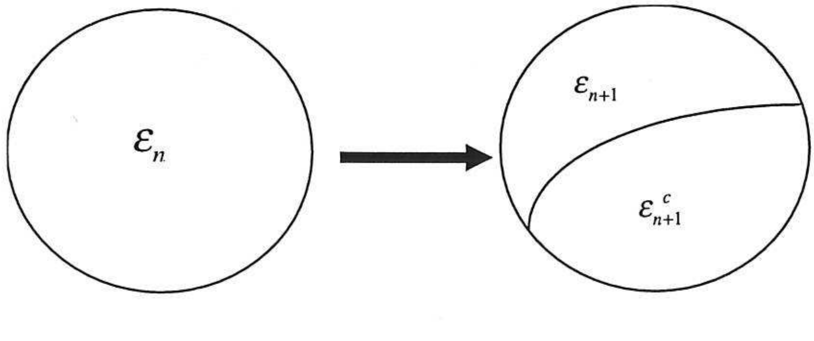


Figure 29. The scenario space is reduced at each stage until the scenario is identified (all faulty sub-units are isolated).

3.4.1 A Four Sub-Unit Example

Four sub-units were used in the following example. The *ambiguity* was chosen to be 100% (that is, all scenarios are possible) For PKI the path lengths were

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PL	2	1	4	3	3	3	4	4	4	2	4	4	4	3	4

for each scenario given a particular reading vector. S and PL signify the scenario number (index) and the path-length respectively. For example, S=6 signifies scenario 6 with a binary representation of 0110; this indicates that sub-units (LRUs) 2 and 3 are faulty. The probability weighted sum of path lengths or APL is 2.2905 in this case.

For the same parameters the baseline (MAP) approach gives

S	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
PL	4	1	4	2	4	2	4	3	4	3	4	3	4	3	4

giving an APL of 3.3758.

Notice that, in individual cases involving a single scenario, MAP can be superior. This is because of misdirection (incorrect choice in a specific case based upon probabilities) by the PKI method. For example, consider scenario 4 where PKI takes 3 LRU inspections and the MAP approach requires only 2 inspections. However, for a given reading vector, the probability (relative frequency) weighted total (APL) for PKI is always smaller (or equal to) that of the MAP method.

In the weighted PKI case (WPKI), discussed in section 1.7, the sub-unit with the lowest cost weighted probability of a fault is chosen; this is equivalent to the above definition in the unity weighted (unweighted) case. PKI superiority in both the unweighted and weighted cases is discussed in section 2.8.

3.5 Simulation Studies: Actual Simulations

3.5.1 Single Seed Simulations

One thousand simulations were carried out where each simulation generated a set of FC frequencies and applied the posterior knowledge and baseline methods to the scenarios within each simulation. For the simulations described here, a total of eight sub-units was used to represent a hypothetical plant; this gave 256 outcomes where the fault scenarios were represented by 8-bit binary strings. Here, the 256 fault scenarios were equally likely. Multiple seed simulations of Section 2.6 use different distributions. Figure 30 shows a quasi-histogram for a set of simulations in which *all* 255 scenarios having one or more faults are possible. The 256th scenario has zero faults.

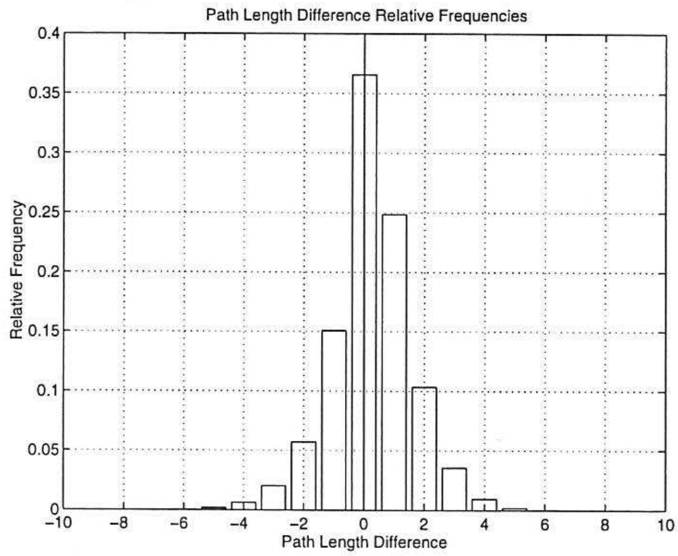


Figure 30 A quasi-histogram of path length differences over a thousand simulations. Each simulation provided a number of scenarios for identification using both the posterior knowledge and baseline methods. The histogram is skewed towards positive path length indicating that posterior knowledge integration results in shorter path lengths overall. The height of the bar indicates the relative frequency of occurrence for a particular PLD

Comparing related columns (PLD of same *magnitude*) such as +1 and -1 sub-unit inspections respectively reveals that the quasi-histogram is skewed towards positive path length differences, i.e. that positive PLDs are more likely. This means that the sequential integration of posterior knowledge has reduced, on average, the number of required sub-unit inspections. As 255 fault scenarios were possible, the scenario predictions were maximally ambiguous, that is, when an actual scenario is to be identified, it can be any 1 of 255 possible scenarios.

Figure 31 shows the same protocol but with the number of scenarios reduced to *any* 64 out of 256. Comparison of figures 30 and 31 reveals that the quasi-histogram skewing is more pronounced—as expected—because the number of possible scenarios for a given diagnosis is reduced. In reality, the number of fault scenarios predicted for a given feature vector will invariably be lower than the maximum possible; the prior distribution of scenario frequencies will be dependent upon the dynamical system being monitored. For example it is conceivable that multiple fault scenarios will be much less prevalent than simple fault scenarios thereby reducing the number of fault scenarios associated with a given input. Furthermore, associations between fault scenarios and input vectors depend upon the key features monitored. *If ambiguity is high, then it is likely that the choice of monitored features is not optimal for predictive disambiguation of fault scenarios.*

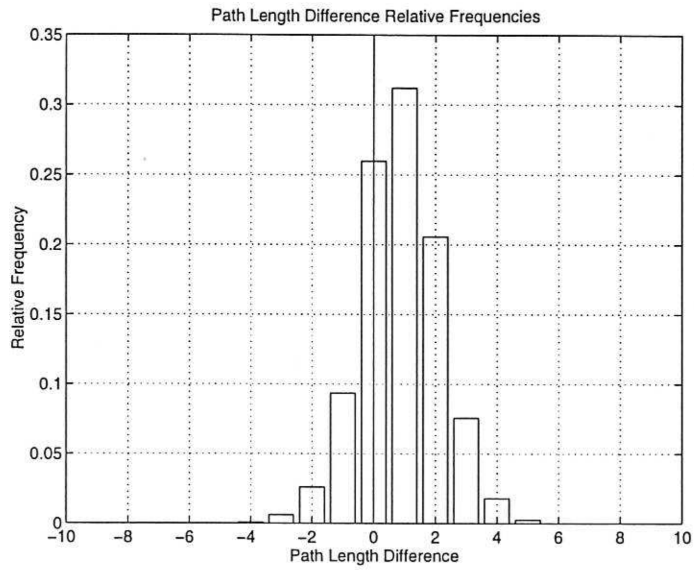


Figure 31. The results of the same simulations carried out using a reduced number of possible scenarios of 64 out of 256. The skewing is more pronounced. See text for details.

Figure 32 shows the same protocol again but this time with only 8 out of a possible 256 scenarios. Note that the skewing is even more pronounced than in the previous simulations. This results from a further substantial reduction in target scenario ambiguity.

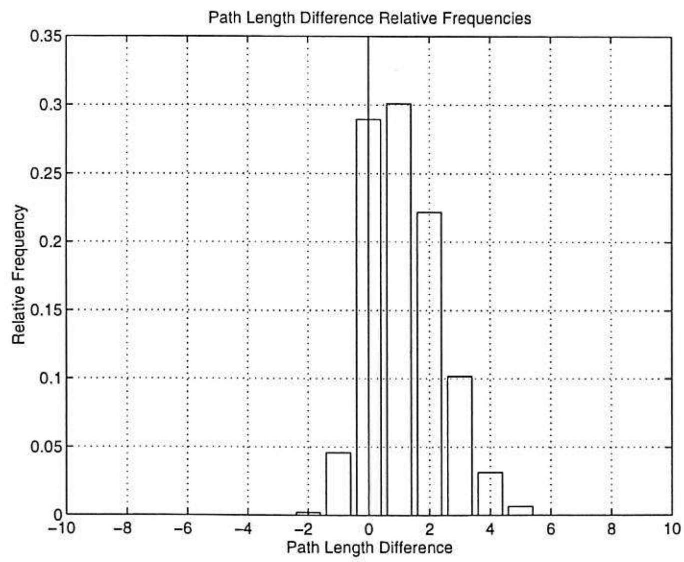


Figure 32. The results of the same simulations but using a further reduced number of possible scenarios (8 out of 256). The skewing is yet more pronounced. See text for details.

Figure 33 shows the total path-length-difference relative frequencies of the left and right sides of the quasi-histograms for an increasing number of scenarios. These sums represent the total relative frequencies of the left and right sides of the quasi-histograms. A larger positive PLD relative frequency indicates that the path-length for posterior knowledge integration is shorter than for the baseline method. Note that the positive PLD sum is consistently larger. As the number of scenarios allowed for each simulation increases, the difference between the positive and negative PLD relative frequencies diminishes. The number of possible scenarios for a given input increases and represents an increase in predictive ambiguity. An expected consequence is that the effectiveness of the posterior knowledge technique diminishes. A high degree of predictive ambiguity is not expected in a 'real world' situation because it would indicate a problem with fault resolution. A consistently larger *positive* PLD relative frequency indicates that the posterior knowledge integration technique always outperforms the baseline technique. The main point is that the posterior knowledge integration technique is superior even with high predictive ambiguity.

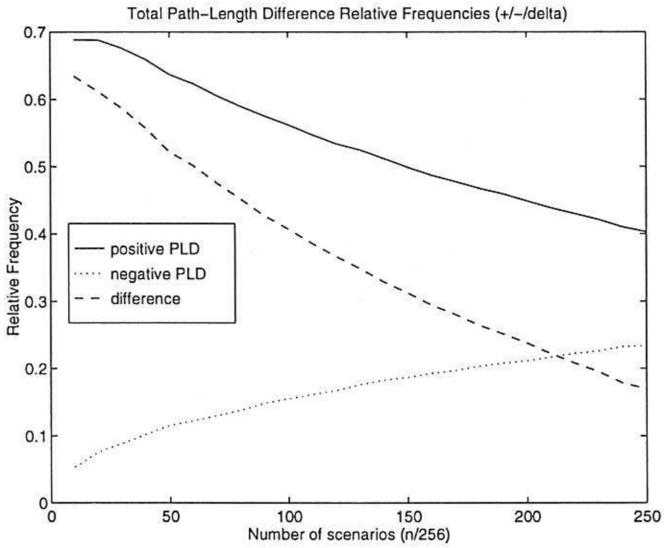


Figure 33. A graph of the PLD sum relative frequency versus the number of scenarios allowed for each simulation. The positive PLD sum relative frequency is a total relative frequency reflecting how many times the baseline method path length has exceeded that of the posterior knowledge integration technique (positive PLD) i.e. the total relative frequency for the right-hand side of the histogram. This shows where the integration of posterior knowledge has shortened the path-length.

Where there are only singleton classes, i.e. no joint probabilities, there will be no gain using the posterior knowledge integration method. This is because no information is given about other FCs or sub-units in the form of joint probabilities. For individual scenarios, the posterior knowledge integration method may result in a longer path length. However, on average, the overall path length is shorter owing to the larger probability weightings of shorter path lengths arising from the posterior knowledge integration choice mechanism. Thus, on average, the posterior knowledge integration method is at least as good as, if not better than, the baseline method. This may lead to financial savings for maintenance.

3.6 *Multiple seed simulations*

3.6.1 An Illustrative Set of Simulations

Figure 34 shows the results of a 'calibration' run where equal fault scenario priors are used to check the PKI-BL system. The results for the average number of faults climbs linearly at 0.5 faults per extra sub-unit, as expected.

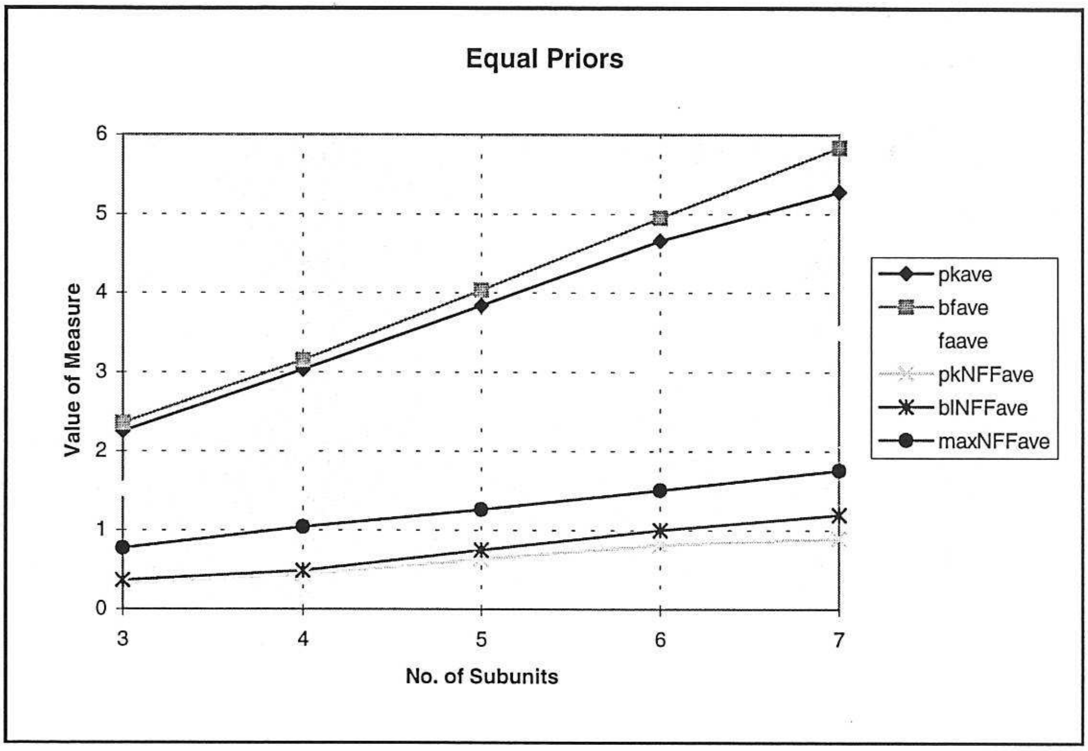


Figure 34. A calibration run indicating that the PKI system is working properly. The increase of 0.5 faults per added sub-unit is as expected for equal fault scenario priors.

The first illustrative simulation shown in Figure 35 has an ambiguity of 0.25, that is, each reading vector has only 25% of the total number of scenarios associated with it. The code A0.25unr/u indicates 25% ambiguity, unbiased but random scenario priors and a random unweighted likelihood matrix. The numerical values are shown in Table 6.

Numsub	pkave	bfave	faave	pkNFFave	blNFFave	maxNFFave
3	1.5463	1.5463	1.4797	0.0582	0.0582	0.5006
4	1.4856	1.5879	1.2687	0.1351	0.1618	1.1897
5	2.4595	2.6955	1.9748	0.2537	0.3504	1.451
6	2.9887	3.4476	2.2131	0.4133	0.6482	1.8439
7	3.9228	4.5092	2.7875	0.6098	0.9285	2.0395
8	4.9123	5.6407	3.3487	0.818	1.1891	2.2835

Table 6. The actual numerical values for the simulation of Figure 35

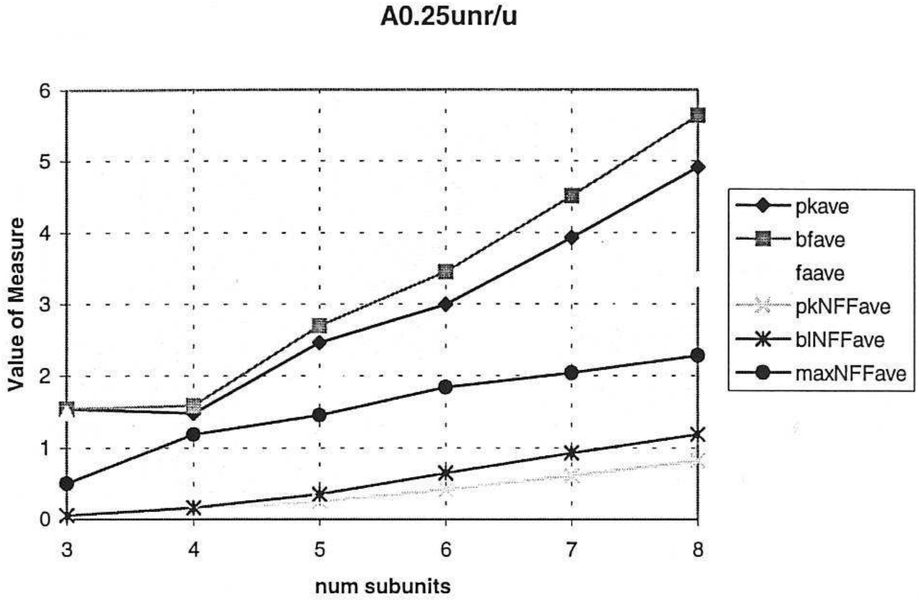


Figure 35 A simulation of PKI showing the increase in the average path lengths for PKI, BL and optimal situations respectively. Note that the PKI method is between the optimal and the BL method for any number of sub-units.

The code A1.0unr/u indicates 100% ambiguity and the PKI method is still better than that of the baseline method as expected.

Numsub	pkave	bfave	faave	pkNFFave	blNFFave	maxNFFave
3	1.4821	1.6351	1.0286	0.2217	0.326	1.1209
4	1.8917	2.0187	1.28	0.2804	0.3235	1.3545
5	2.4129	2.7334	1.5484	0.4263	0.578	1.7414
6	2.9751	3.3263	1.8286	0.5945	0.7252	2.0926
7	3.5221	3.9105	2.1165	0.6981	0.8757	2.4713

Table 7. The actual numerical values for the simulation of Figure 36.

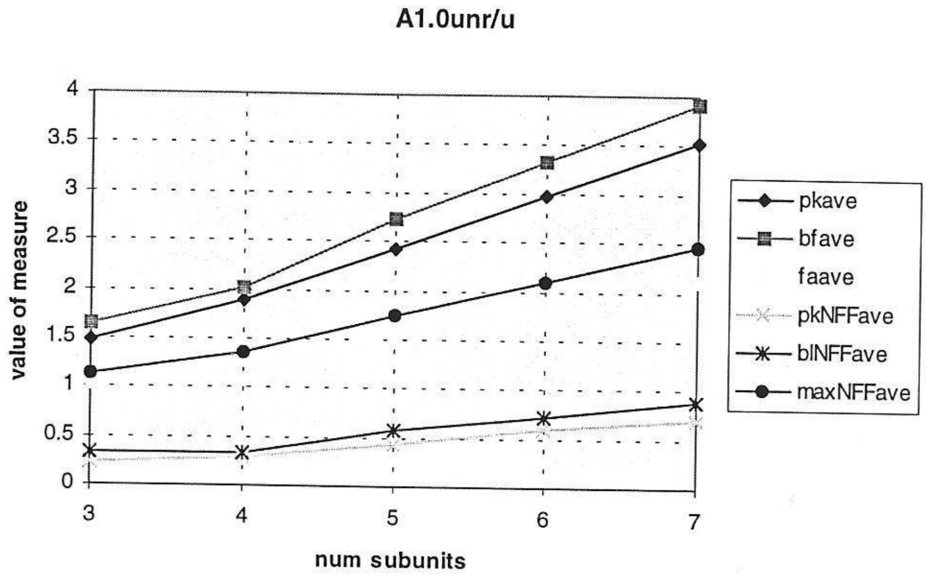


Figure 36. Again, a simulation of PKI showing the increase in the average path lengths for PKI, BL and optimal situations respectively. Here the ambiguity is 100% but note that the PKI method is still between the optimal and the BL method for any number of sub-units.

Figure 37 shows the average path length reduction of PKI compared with the baseline method for an *ensemble* of 9 runs. The variation results from the use of 9 different random number seeds to generate the probabilities involved. The percentage APL reduction depends upon the number of sub-units in the system which provide the joint events which PKI exploits.

These results are obtained using an ambiguity parameter of 0.25, that is, only 25% of the total number of scenarios are possible for a given input. An ambiguity of 100% would mean that all fault scenarios are possible to some degree which would indicate that the alarm reading coding used to generate the probabilistic FDI information was not very effective. A one to one mapping between the alarms and the fault scenarios would be the ideal situation with 0% ambiguity.

For the current set of operational parameters, the *average ensemble* APL reduction is as high as 13-14% in some cases. Even with a small number of sub-units, it is 8% or greater.

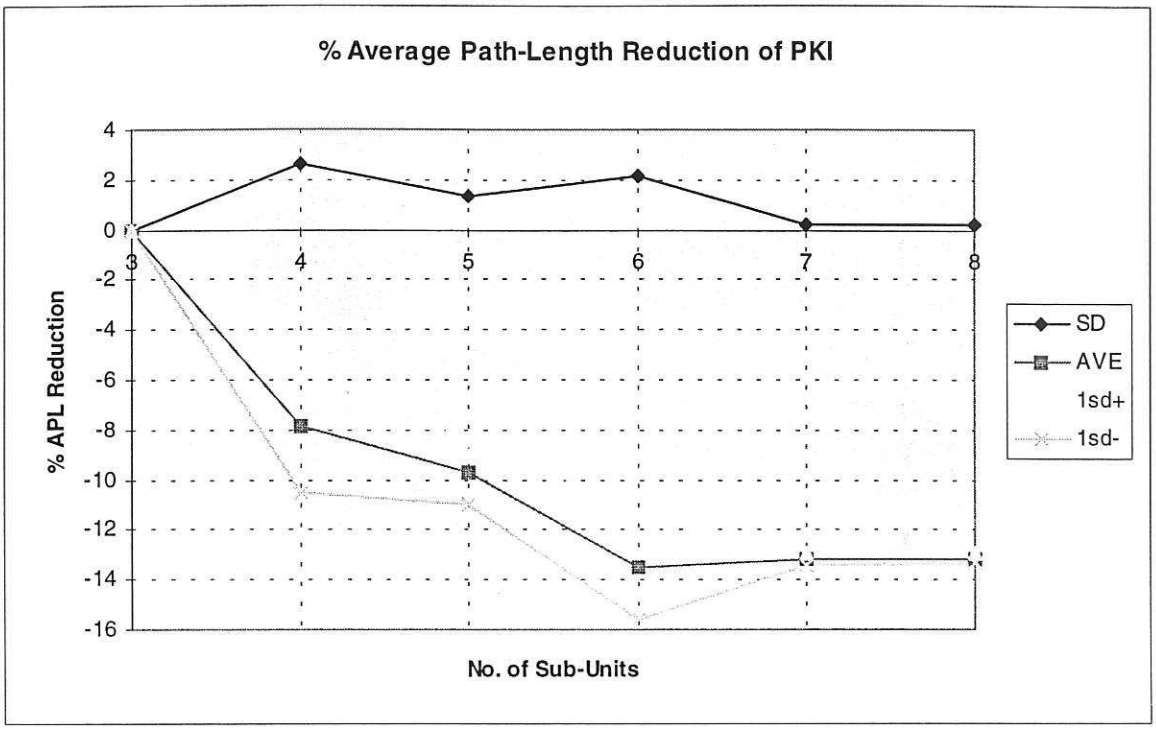


Figure 37. Percentage reduction in average path-length per scenario of PKI compared to the baseline method.

Figure 38 shows even greater gains in performance by the PKI method for the NFF incident case. The PKI method has reduced the *average ensemble* ANFFN to between 25-35% in some cases. This means that, across a number of systems, the expected NFF number per scenario is about 30% less, i.e. 30% fewer NFF incidents are expected.

The reduction of NFF incidents is a consequence of the PKI method tending to look for the sub-units most likely to be faulty. This means that non-faulty sub-units are more likely to be avoided by PKI than the baseline method. As the sub-units exhibiting NFF incidents form a subset of the set of non-faulty sub-units, the NFF incidents are either the same or reduced by the PKI method.

Note that there is no gain in the results for 3 sub-units. Here there is very little scope for PKI to reduce path length when the paths are so small. However, note that there is improvement as the number of sub-units increases and, consequently, the scope for longer path-lengths.

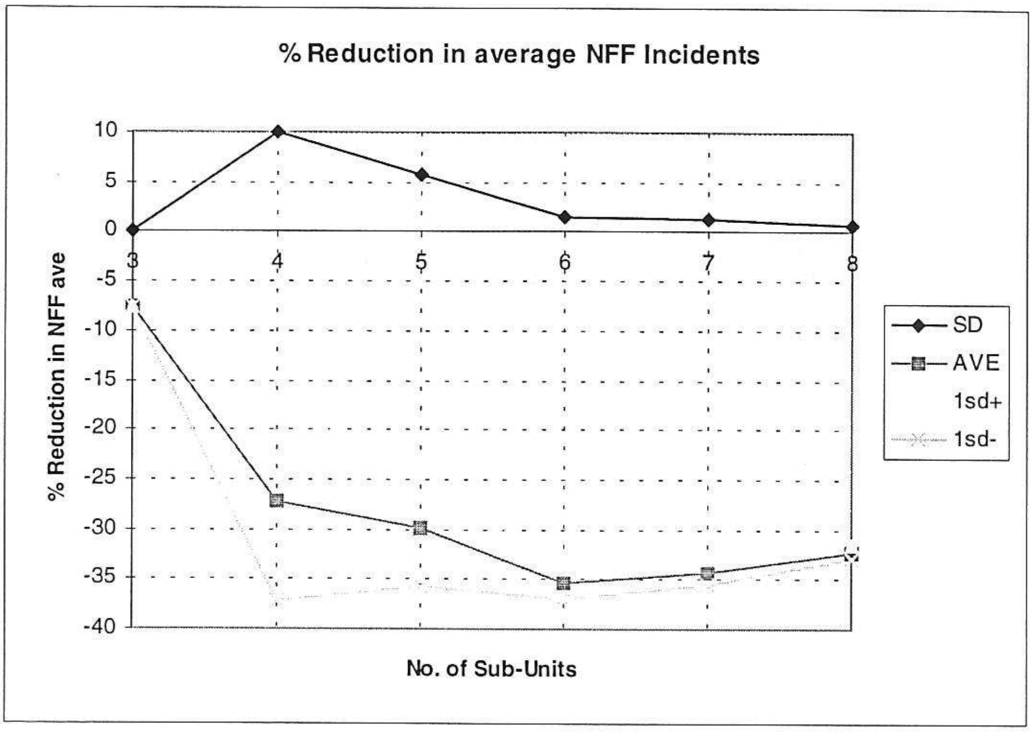


Figure 38. Percentage reduction in average NFF incidents per scenario by the PKI method compared to the baseline method.

3.7 Weighted Multiple seed simulations

The empirical results suggest that unweighted PKI using the PPUE is as good as or superior to unweighted MAP. The superiority rests upon the availability of joint information. This relationship between PKI performance and the availability of joint information still has to be investigated. The improvement of PKI over MAP depends upon the amount of joint information available. Total independence of sub-units implies that no gain would be possible because joint information is used to increase or decrease the influence of other sub-units through inclusion or exclusion of fault scenarios following evidence propagation.

The number of faults to be found during LRU inspections determines the minimum APL. This is given by the fault average and is the optimum overall APL value which cannot be improved upon by either the PKI or the MAP methods. Where PKI is superior, it is through reducing the APL component originating from LRU inspections resulting in NFF incidents. In other words, PKI seeks to reduce the number of NFF incidents by using PK to avoid inspecting LRUs where it is likely that faults will *not* be found.

3.7.1 Weighted PKI

In the simulations so far, the sub-units have all been weighted evenly (unity weighting). In reality, the inspection and repair costs of sub-units (LRUs) differ widely depending upon component costs, accessibility, time costs and so on. A refinement of the current PKI system is to include these weightings to give WPKI.

Now, the unweighted version of PKI relies upon the fact that *maximising* the posterior probability of a fault *minimises* the probability of a NFF. This connection is broken in the weighted case and so the weighted posteriors are minimised directly, that is, sub-units with the lowest weighted posterior are chosen for inspection This allows the *weighted* NFF component of the weighted APL to be reduced by the PKI process providing that joint information exists. This is discussed formally in Appendix G. An informal discussion is given in sub-section 2.7.2 below.

Figure 39 shows the average path length reduction of PKI compared with the baseline method for an *ensemble* of 9 runs using weighting. The variation results from the use of 9 different random number seeds to generate the probabilities involved. The percentage APL reduction depends upon the amount of joint information in the system that PKI exploits. These results are obtained using an ambiguity parameter of 0.25. The weighting is a permutation of $\{1,\ldots,n\}$ where n is the number of sub-units. For fair comparison, we weight the probabilities of the MAP classifier in exactly the same way as for WPKI. Thus all references to MAP in the weighted context imply weighted MAP.

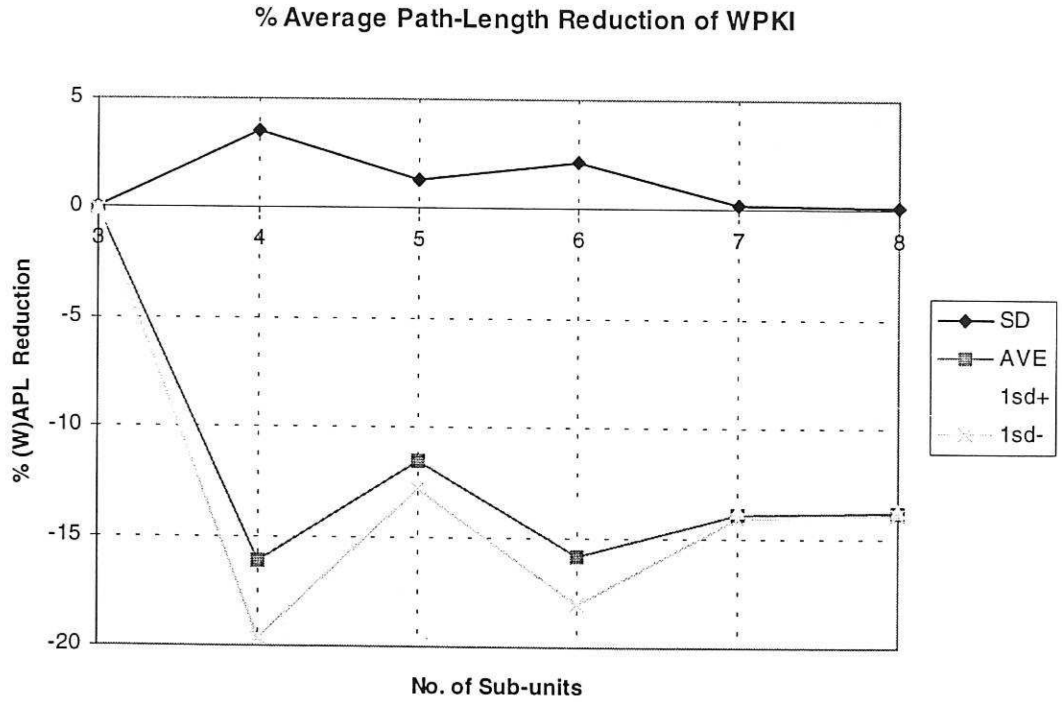


Figure 39 The average percentage path length reduction of WPKI compared with weighted MAP.

Figure 40 shows even greater gains in performance by the PKI method for the NFF incident case. The PKI method has reduced the *average ensemble* (W)NFF to between 35-45% in some cases. This means that, across a number of systems, the expected (W)NFF number per scenario is about 40% less, i.e. 40% fewer (W)NFF incidents are expected. The reduction of (W)NFF incidents is a consequence of the (W)PKI method tending to look for the sub-units with the lowest (W)NFF probability.

Proof of superiority relies on the fact that, where there are faults, both weighted PKI and MAP cannot be better than the weighted optimum. The gain of weighted PKI over MAP is where the weighted no-fault contributions to the PL measures are minimised. This is equivalent to maximising the fault probabilities in the unweighted version.

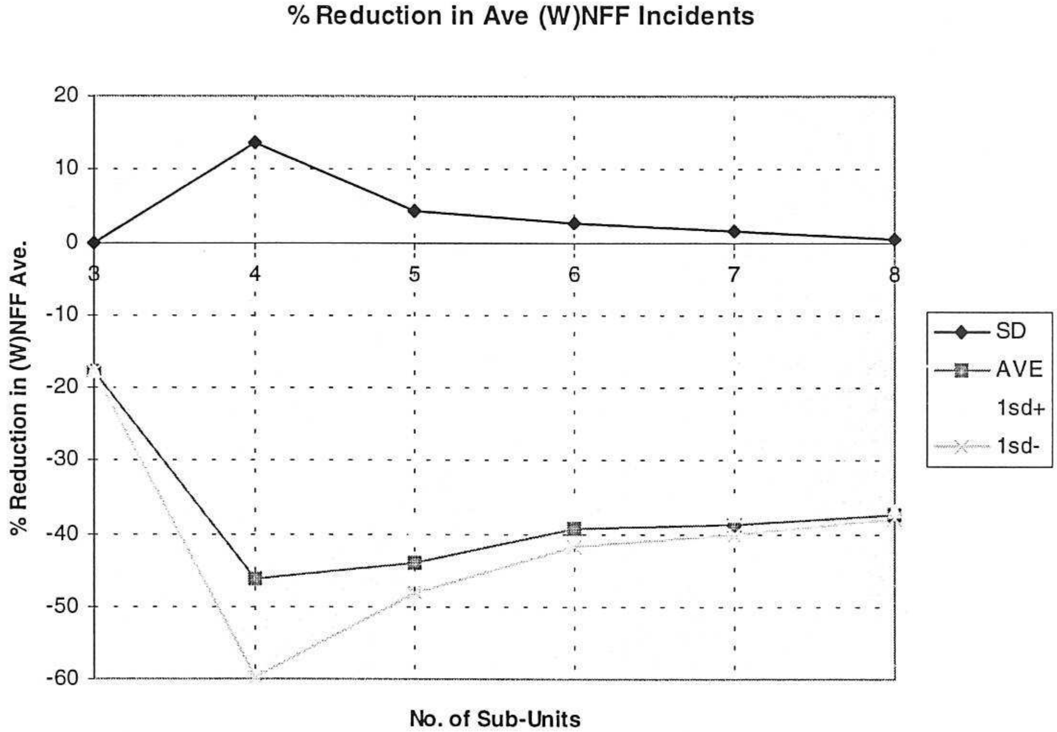


Figure 40.The average percentage NFF incident reduction of WPKI compared with weighted MAP.

3.7.2 Direct connection of APL minimisation with NFF reduction

In the previous discussion of the context-free simulation algorithm, it was apparent that there is a common core of the average scenario path lengths where the probability-weighted path lengths do not differ for the PKI and BL methods. This is true in both the weighted and unweighted cases because *all* faults must be found in both cases and the weighting is the same for any approach. The faults can be chosen in any order and incur the same inspection penalty regardless of approach. The remaining APL contributions are given by a 'residue' over and above the method-independent theoretical optimum. The residue arises where LRUs are inspected and no faults are found. Thus, the residue determines the overall APL for any technique and APL minimisation is concerned with avoiding NFF incidents in both the weighted and unweighted cases. PKI acts to reduce the probability or weighted probability of NFF incidents by choosing LRUs with the lowest (weighted) probabilities of no-fault LRU states. Thus LRUs exhibiting no faults or those highly weighted will be less likely to be chosen for inspection.

A greater percentage reduction in NFF incidents alone is noted for PKI than overall (including faulty LRU inspection) because reduced APL is gained by minimising NFF incidents only. There is a component of the APL *common* to both PKI and MAP (or any possible technique) and is irreducible because ALL faults have to be found.

Now, the unweighted approach relies on the fact that $P(C_i) + P(C_i^c) = 1$ so that when the probability of a fault is maximised, i.e. $P(C_i) > P(C_j)$ in all cases where $i \neq j$

The relationship $P(C_j) + P(C_j^c) = 1$ ensures that $P(C_i^c) < P(C_j^c)$ is the case. Thus, the probability-weighted residue will be smaller where a NFF incident occurs.

Now, in the weighted case, $W_i\left(P(C_i) + P(C_i^c)\right) = W_i$ and $W_j\left(P(C_j) + P(C_j^c)\right) = W_j$ and if $W_j \neq W_i$ in *all* cases, then the relationship between LRU fault probabilities is broken.

So, $W_i P(C_i) > W_j P(C_j)$ does not imply $W_i P(C_i^c) < W_j P(C_j^c)$.

As it does not matter which order the faulty LRUs are found and the residue is solely dependent upon the occurrence of NFF incidents, the choice of weighted LRUs is compromised. If the relationship $W_i P(C_i^c) < W_j P(C_j^c)$ cannot be guaranteed, then the *minimisation* of the residue by PKI is not assured.

The way to minimise the residue is to use the relationship $W_i P(C_i^c) < W_j P(C_j^c)$ directly. Where unity weighting is used, the relationship $P(C_i) > P(C_j)$ is preserved.

To summarise, the order in which faulty LRUs are found is irrelevant in both the weighted and unweighted cases. The APL will always have a minimum value which depends solely upon the faulty LRUs because all faulty LRUs have to be inspected. The optimisation of maintenance strategies arises from minimising the NFF incidents and thus the APL contributions from inspecting non-faulty LRUs. Now, in the weighted case, it is better to reduce the APL contributions directly by minimising the weighted probability of a fault *not* being found. This guarantees that the residue, over and above the optimum APL, will be a minimum.

3.8 The PKI Performance Rule: A Discussion

In both the unweighted and weighted PKI cases, PKI appears to be consistently better than the MAP approach when using the path-length and NFF incident measures. The empirical investigations lead to a conjecture that PKI is always at least as good as, if not superior to the MAP approach. Theoretical investigations have confirmed this conjecture and lead to the following theorem:

Theorem: The PKI performance theorem.

The performance of PKI is always at least as good as that of the baseline method and possibly better.

A proof of this Theorem for the unweighted case is given in Appendix F. An analogous proof for the weighted version is found in Appendix G.

4 Bayesian Belief Networks

This Chapter will introduce the ideas of Bayesian belief networks (BBNs) and how they relate to the task of posterior knowledge integration. It will be shown that the PPUE and BBNs represent two extremes of a continuum of information regarding a FDI situation. The PPUE represents the extreme of relying solely on empirical data to provide the basis for probability updates; this is the empirical approach. The structural approach, using BBNs is the opposite extreme where probabilities are given (and updated) in terms of known causal links arising from engineering knowledge. It will be shown that the two approaches are interchangeable in the general context of PKI. Figure 41 shows the situation schematically.

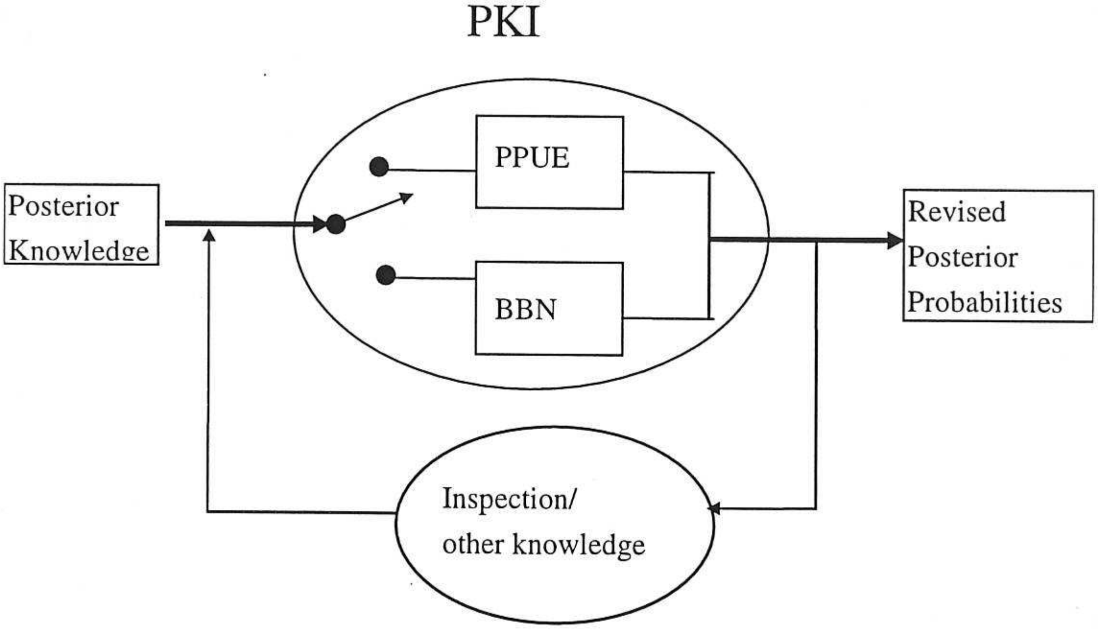


Figure 41. Either the PPUE or the BBN approaches can be used to update the posterior probabilities as part of the PKI process. Further work may uncover a way of combining the two approaches.

4.1 Fault distinction and causality.

- faults can be classified conveniently into two types:
    1. Sensor faults (SFs), and
    2. Actual faults (AFs)

- Sensor faults are faults with the sensing devices as opposed to actual deviations in sub-system behaviour. A schematic example is shown in Figure 42.

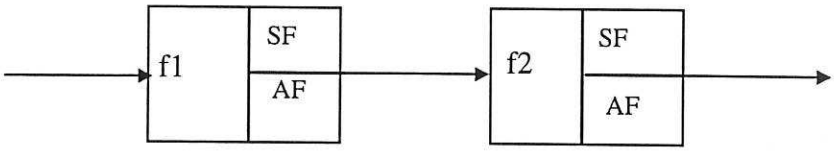


Figure 42. Two potential faults in a causal link. Faults may be sensor-based or actual.

If actual fault f1 "causes" actual fault f2, extra information may be gained through a knowledge of causality. For example, if f1 occurs without f2, it is highly likely that a SF has occurred. If f2 occurs then information further downstream is required to differentiate between SFs and AFs. These considerations introduce the idea of fault causality.

4.2 Cause and Effect

Figure 43 shows a possible sub-system of a larger system to illustrate the ideas of cause and effect. Conditions in components A and C occur independently of those in component B and vice-versa. However, the occurrence of a condition in component A entails a condition in component C. A condition in C can also occur independently of conditions in A. Thus, there is a causal link between A and C.

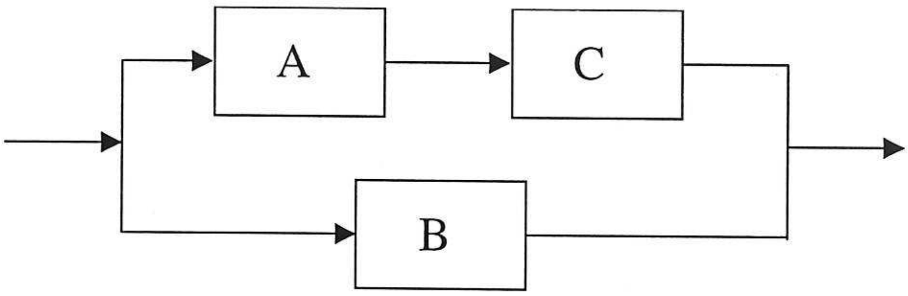


Figure 43. A schematic representation of a group of interconnected sub-systems which form part of a system model. Here, sub-systems A and C are causally connected.

Figure 44 illustrates this situation in terms of sets. For $A \subset C, \quad (A \neq C)$ the condition denoted by class C will always occur when the condition denoted by class A occurs. Thus,

$$P(C|A) = \frac{P(A \cap C)}{P(A)} = \frac{P(A)}{P(A)} = 1$$

This is not necessarily the case the other way round where

$P(A|C) = \frac{P(A \cap C)}{P(C)} = \frac{P(A)}{P(C)} < 1$ if class A and class C are distinct (not equal). The case that $A = C$ is excluded because two such classes would be indistinguishable.

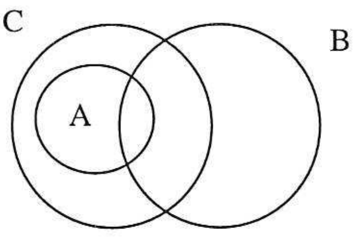


Figure 44. A Venn diagram showing the relationships between the event sets of figure 42. When class A occurs it causes class C to occur. Note that class C events do not necessarily generate a class A event.

In many cases, determining the direction of causality will not be straightforward when using the PPUE. For example, given $P(C_1|C_2 \cap \mathbf{x}_k)$ is sub-unit affected in the sense of "causality" or in the sense of "information passing" as discussed below. $P(C_1|C_2 \cap \mathbf{x}_k)$ can give rise to $P(C_2|C_1 \cap \mathbf{x}_k)$ by the use of Bayes theorem.

Figure 45 shows actual causality between sub-units A and B. That is, sub-unit A can cause changes in sub-unit B. Here, the sub-units will be represented as circles or nodes as in Figure 46.

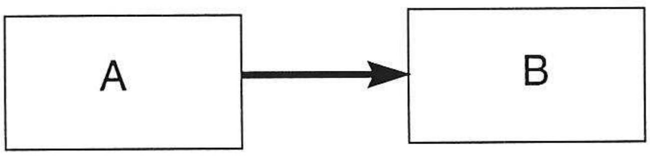


Figure 45. A schematic illustration of actual causality between sub-units.

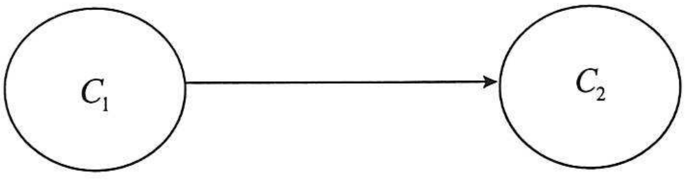


Figure 46. The representation of causality as a directed link between nodes representing sub-units.

Causality reflects the ability of one sub-system state to alter another and information about the "causal" state gives information about the "recipient" state in the direction of causality. The idea of actual causality is related to classical deductive logic of the form $A, A \Rightarrow B \therefore B$

However, what if information is available about the "recipient" sub-unit as shown in Figure 47?

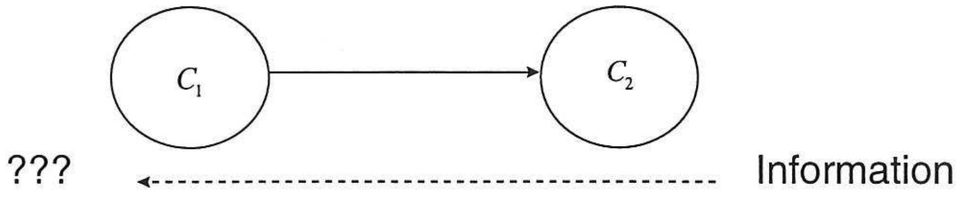


Figure 47. The passing of non-causal information in the reverse direction. Observed events may give rise to information concerning the causes of those events.

Information about a recipient sub-unit given information about the causal sub-unit or event but this is NOT causality. The information (AFTER the causal "event") reflects the likely state of the causal sub-unit. This can be symbolized by
$B, A \Rightarrow B \therefore ???$
In other words, what do fault readings tell us about the possible causes of faults?
In this situation, the causality is from faults to readings and readings *do not* "cause" faults but give information about likely causes. This is abductive logic or reasoning from conclusions to causes. It is "acceptance of a conclusion on the grounds that it explains the available evidence" (Krause and Clarke, 1993) and is used in Bayesian belief networks to update probabilities of "parent" variables. This will be covered in the remainder of this chapter.

4.3 Illustrative discussion problem:

Three fault classes, C_1,C_2,C_3 given prototype $\hat{\mathbf{x}}$.

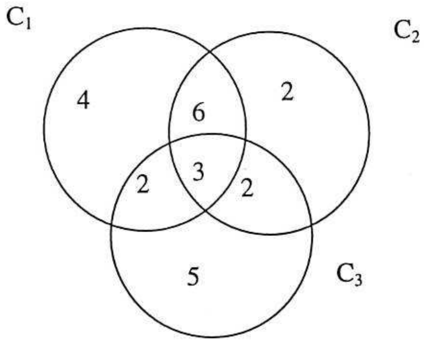


Figure 48 A simple numerical example to illustrate the issues of investigating causality using empirical data.

The total number of occurrences of each FC are given by

$n(C_1 \cap \hat{\mathbf{x}}) = 15$

$n(C_2 \cap \hat{\mathbf{x}}) = 13$

$n(C_3 \cap \hat{\mathbf{x}}) = 12$

The total number of scenarios is given by

$$n(\hat{\mathbf{x}}) = n(C_1 \cap \hat{\mathbf{x}}) + n(C_2 \cap \hat{\mathbf{x}}) + n(C_3 \cap \hat{\mathbf{x}})$$
$$- n(C_1 \cap \hat{\mathbf{x}} \cap C_2) - n(C_1 \cap \hat{\mathbf{x}} \cap C_3) - n(C_2 \cap \hat{\mathbf{x}} \cap C_3)$$
$$+ (C_1 \cap \hat{\mathbf{x}} \cap C_2 \cap C_3)$$

$$n(\hat{\mathbf{x}}) = 15 + 13 + 12 - 9 - 5 - 5 + 3 = 24$$

Assumption of exaustivity:

$$\forall \mathbf{x} \in category(\hat{\mathbf{x}}) \Rightarrow \mathbf{x} \in \bigcup_{i=1}^{i=3} C_i$$

Now,

$$p(C_1|\hat{\mathbf{x}}) = \frac{n(\hat{\mathbf{x}} \cap C_1)}{n(\hat{\mathbf{x}})} = \frac{15}{24} = 0.625$$
$$p(C_2|\hat{\mathbf{x}}) = \frac{n(\hat{\mathbf{x}} \cap C_2)}{n(\hat{\mathbf{x}})} = \frac{13}{24} = 0.5417$$
$$p(C_3|\hat{\mathbf{x}}) = \frac{n(\hat{\mathbf{x}} \cap C_3)}{n(\hat{\mathbf{x}})} = \frac{12}{24} = 0.5$$

The posterior probabilities, given by vector $\mathbf{x}$ belonging to input category (cluster) denoted by prototype $\hat{\mathbf{x}}$, give a prediction as shown in Figure 49.

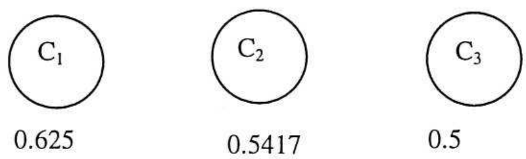


Figure 49. The posterior probabilities of FCs occurring given a reading vector.

Note that $p(C_1 \cap \hat{\mathbf{x}}) + p(C_2 \cap \hat{\mathbf{x}}) + p(C_3 \cap \hat{\mathbf{x}}) \neq 1$ unless the classes are exclusive.

Probabilities of two classes occurring together given the evidence:

$$p(C_1 \cap C_2|\hat{\mathbf{x}}) = \frac{n(\hat{\mathbf{x}} \cap C_1 \cap C_2)}{n(\hat{\mathbf{x}})} = \frac{9}{24} = 0.375$$
$$p(C_1 \cap C_3|\hat{\mathbf{x}}) = \frac{n(\hat{\mathbf{x}} \cap C_1 \cap C_3)}{n(\hat{\mathbf{x}})} = \frac{5}{24} = 0.2083$$
$$p(C_2 \cap C_3|\hat{\mathbf{x}}) = \frac{n(\hat{\mathbf{x}} \cap C_2 \cap C_3)}{n(\hat{\mathbf{x}})} = \frac{5}{24} = 0.2083$$

Interdependence of classes:

$$p(C_1|\hat{\mathbf{x}} \cap C_2) = \frac{n(\hat{\mathbf{x}} \cap C_1 \cap C_2)}{n(\hat{\mathbf{x}} \cap C_2)} = \frac{9}{13} = 0.6923$$

$$p(C_2|\hat{\mathbf{x}} \cap C_1) = \frac{n(\hat{\mathbf{x}} \cap C_1 \cap C_2)}{n(\hat{\mathbf{x}} \cap C_1)} = \frac{9}{15} = 0.6$$

Correlation between classes 1 and 2 Class 2 is a slightly better indicator of class 1 than vice versa. Note that, $p(C_1|\hat{\mathbf{x}} \cap C_2) \neq p(C_1|\hat{\mathbf{x}})$ and $p(C_2|\hat{\mathbf{x}} \cap C_1) \neq p(C_2|\hat{\mathbf{x}})$ indicating the dependence. The empirical data does not indicate the direction of causality which needs to be indicated explicitly.
Similarly,

$$p(C_1|\hat{\mathbf{x}} \cap C_3) = \frac{n(\hat{\mathbf{x}} \cap C_1 \cap C_3)}{n(\hat{\mathbf{x}} \cap C_3)} = \frac{5}{12} = 0.4167$$

$$p(C_3|\hat{\mathbf{x}} \cap C_1) = \frac{n(\hat{\mathbf{x}} \cap C_1 \cap C_3)}{n(\hat{\mathbf{x}} \cap C_1)} = \frac{5}{15} = 0.3333$$

$$p(C_2|\hat{\mathbf{x}} \cap C_3) = \frac{n(\hat{\mathbf{x}} \cap C_2 \cap C_3)}{n(\hat{\mathbf{x}} \cap C_3)} = \frac{5}{12} = 0.4167$$

$$p(C_3|\hat{\mathbf{x}} \cap C_2) = \frac{n(\hat{\mathbf{x}} \cap C_3 \cap C_2)}{n(\hat{\mathbf{x}} \cap C_2)} = \frac{5}{13} = 0.3486$$

Integration of *posterior* knowledge:

The information that class 2 has not occurred has now been given. How is this knowledge to be incorporated?

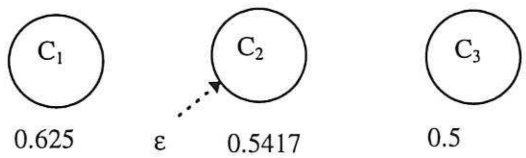


Figure 50. Apply evidence concerning FC2.

$$p(C_1|\hat{\mathbf{x}} \cap \overline{C}_2) = \frac{n(\hat{\mathbf{x}} \cap C_1 \cap \overline{C}_2)}{n(\hat{\mathbf{x}} \cap \overline{C}_2)} = \frac{6}{11} = 0.5455$$

$$p(C_3|\hat{\mathbf{x}} \cap \overline{C}_2) = \frac{n(\hat{\mathbf{x}} \cap C_3 \cap \overline{C}_2)}{n(\hat{\mathbf{x}} \cap \overline{C}_2)} = \frac{7}{11} = 0.6374$$

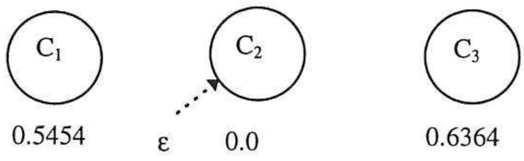


Figure 51. The effects of evidence concerning FC2 on the other posterior probabilities.

Now, the probability of class 1 occurring has decreased and that of class three has increased. All this has been achieved using empirical data. However, the causality of the system consisting of the sub-units is unknown. What if causal information is available from a causal model? Incorporating this into the PKI process may reduce the probability estimation overheads. The technique of Bayesian Belief networks is a possible way of incorporating prior engineering knowledge into an FDI system which is related to the PPUE. The remainder of this chapter will explore this relationship.

4.4 PPUE and Bayesian Belief Networks

4.4.1 Introduction: Causal Networks

Systems can be represented by *causal networks*. (Jensen,1996) A causal network consists of a set of *variables*, which represent sub-unit states of a system, and *directed links* between variables, which represent causal relationships between system sub-units. In formal mathematical terms, a causal network is a *digraph* or *directed graph* (Wilson and Watkins, 1990). Figure 52 shows a causal network which represents a simple three sub-unit system.

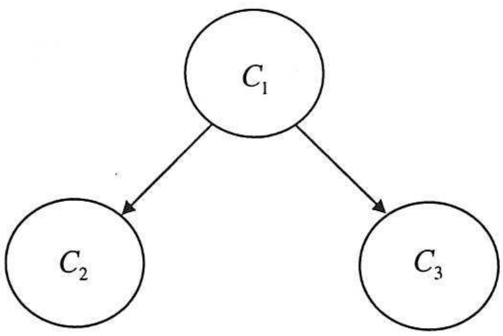


Figure 52. A simple causal network.

4.4.2 Bayesian Networks

In order to quantify the relationships between the system state variables, a probabilistic calculus is required. One such calculus is the *Bayesian probability* *calculus* or classical probability calculus. Bayesian probability calculus allows the quantification of causal links by attaching values to them in accordance with the fundamental definitions and axioms.

The probability of a hypothesis, h given evidence, e has a number of possible axiomatisations. One axiomatisation of probability as a continuous monotonic function P(.) is

A1 $0 \leq P(h|e) \leq 1$

A2 $P(True|e) = 1$

A3 $P(h|e) + P(\neg h|e) = 1$

A4 $P(gh|e) = P(h|ge).P(g|e)$

(Krause and Clark, 1993)

The following theorem can be derived from axiom A4:

$P(h|e) = \frac{P(e|h)P(h)}{P(e)}$ (Bayes' Theorem).

It will be useful throughout this paper.

The notion of conditional independence is crucial to the process of transmitting evidence in Bayesian belief networks as discussed in later sections.
Conditional independence of A and C given the evidence e and B.

The variables A and C are independent given the variable B if

$$P(A|B,C,e) = P(A|B,e)P(B|e)$$

not forgetting the evidential context signified by e.

Marginalisation:

$$P(A|e) = P(A,X|e) + P(A,X^c|e)$$

where X is some event set and the superscript c signifies the complement.
e.g. let $X = B \cap C$

$$P(A|e) = P(A \cap (B \cap C)|e) + P(A \cap (B \cap C)^c|e)$$

Where, for the network of figure 51

$$P(U) = P\left(\bigcup_{i=1}^{3} C_i \cup \left[\bigcup_{i=1}^{3} C_i\right]^c\right) =$$
$$P(C_1) + P(C_2) + P(C_3) + P(NF) - P(C_1 \cap C_2) - P(C_1 \cap C_3) - P(C_2 \cap C_3) + P(C_1 \cap C_2 \cap C_3) = 1$$
and $P(NF) = P\left(\left[\bigcup_{i=1}^{3} C_i\right]^c\right)$.

This is the unity constraint. In other words, the event space is closed.

For *PKI*, we are given
$P(C_1), P(C_2), P(C_3), P(NF), P(C_1 \cap C_2), P(C_1 \cap C_3), P(C_2 \cap C_3),$
$P(C_1 \cap C_2 \cap C_3)$
These probabilities can be constructed from exclusive probabilities estimated from empirical data (Marriott and Harrison a,b). NB, the dependence upon $\boldsymbol{x}$ has been omitted without loss of generality. Other constraints are also effective and can be extracted from the causal network as discussed below.

4.4.3 Connection Types

4.4.3.1 Serial Connection

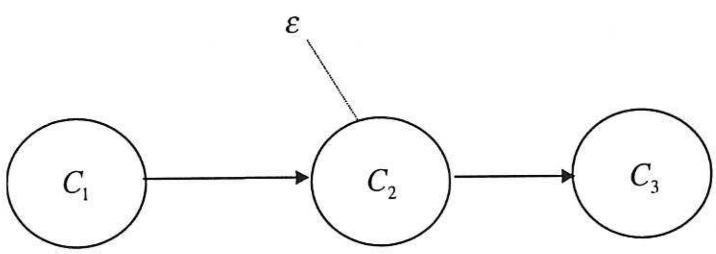


Figure 53. A serial connection of a Bayesian belief network.

Evidence may be transmitted through a *serial* connection unless the state of the variable in the connection is known. In that case, C_1 and C_3 are rendered independent as illustrated in Figure 53.

4.4.3.1.1 3.3.3.1.1 Forward Flow of Evidence in a Serial Connection

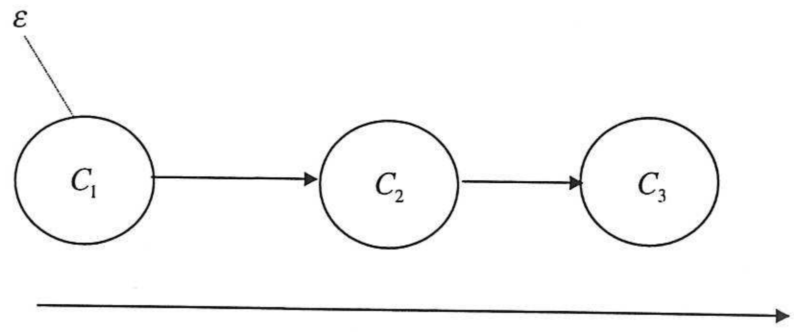


Figure 54. The forward flow of evidence.

Here, the flow of evidence is $P^*(C_1) \to P^*(C_2) \to P^*(C_3)$ and follows the direction of causality as shown in Figure 54.

First, $P^*(C_1)$ is specified which determines the complement, $P^*(C_1^c) = 1 - P^*(C_1)$. Now, the quantities, $P(C_2|C_1)$, $P(C_2|C_1^c)$ are specified for the given BBN. To calculate $P^*(C_2)$, the revised joint probabilities $P^*(C_1 \cap C_2)$ and $P^*(C_1^c \cap C_2)$ have to be calculated so that a marginalisation can be carried out.

Now,

$$
\begin{aligned}
P^*(C_1 \cap C_2) &= P(C_2|C_1)P^*(C_1) \\
&= \frac{P(C_1 \cap C_2)}{P(C_1)} P^*(C_1) \\
&= P(C_1 \cap C_2)\frac{P^*(C_1)}{P(C_1)}
\end{aligned}
$$

and,

$$
\begin{aligned}
P^*(C_1^c \cap C_2) &= P(C_2|C_1^c)P^*(C_1^c) \\
&= \frac{P(C_1^c \cap C_2)}{P(C_1^c)} P^*(C_1^c) \\
&= P(C_1^c \cap C_2)\frac{P^*(C_1^c)}{P(C_1^c)}
\end{aligned}
$$

Marginalising gives,

$$P^*(C_2) = P^*(C_1 \cap C_2) + P^*(C_1^c \cap C_2).$$

Now similarly,

$$
\begin{aligned}
P^*(C_2 \cap C_3) &= P(C_3|C_2)P^*(C_2) \\
&= \frac{P(C_2 \cap C_3)}{P(C_2)} P^*(C_2) \\
&= P(C_2 \cap C_3)\frac{P^*(C_2)}{P(C_2)}
\end{aligned}
$$

and,

$$
\begin{aligned}
P^*(C_2^c \cap C_3) &= P(C_2|C_2^c)P^*(C_2^c) \\
&= \frac{P(C_2^c \cap C_3)}{P(C_2^c)} P^*(C_2^c) \\
&= P(C_2^c \cap C_3)\frac{P^*(C_2^c)}{P(C_2^c)}
\end{aligned}
$$

Marginalising,

$$P^*(C_3) = P^*(C_2 \cap C_3) + P^*(C_2^c \cap C_3).$$

4.4.3.1.2 3.3.3.1.2 Reverse Flow of Evidence in a Serial Connection

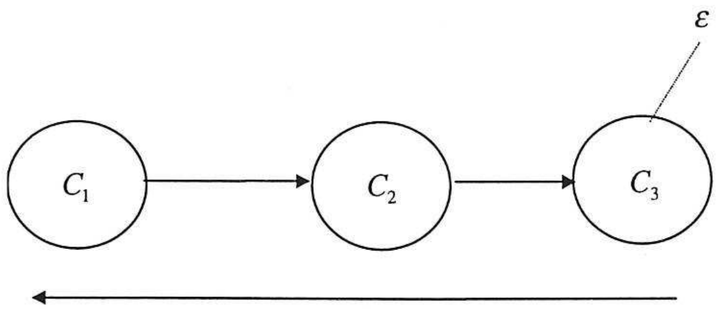


Figure 55 Reverse flow of evidence.

In this case, the flow of evidence is in the opposite direction to causality. This is abductive reasoning from effect to cause. The flow of evidence, $P^*(C_3) \rightarrow P^*(C_2) \rightarrow P^*(C_1)$ is illustrated in Figure 55.

Here, $P^*(C_3)$ is specified which determines $P^*(C_3^c) = 1 - P^*(C_3)$.

Now, $P\left(C_3 \middle| C_2\right)$ is specified for the BBN. This determines $P\left(C_3^c \middle| C_2\right) = 1 - P\left(C_3 \middle| C_2\right)$.

To calculate $P^*\left(C_2\right)$, the revised joint probabilities, $P^*\left(C_2 \cap C_3\right)$ and $P^*\left(C_2 \cap C_3^c\right)$ have to be calculated so that a marginalisation can be carried out.

$$P^*\left(C_2 \cap C_3\right) = P\left(C_2 \middle| C_3\right) P^*\left(C_3\right)$$

$$P^*\left(C_2 \cap C_3^c\right) = P\left(C_2 \middle| C_3^c\right) P^*\left(C_3^c\right)$$

The conditional probability, $P\left(C_2 \middle| C_3\right)$ is not specified in the BBN so the Bayes' rule is used,

$$P\left(C_2 \middle| C_3\right) = \frac{P\left(C_3 \middle| C_2\right) P\left(C_2\right)}{P\left(C_3\right)}$$

Similarly with

$$P\left(C_2 \middle| C_3^c\right) = \frac{P\left(C_3^c \middle| C_2\right) P\left(C_2\right)}{P\left(C_3^c\right)}$$

where $P\left(C_3^c \middle| C_2\right) = 1 - P\left(C_3 \middle| C_2\right)$.

Marginalisation

$$P^*\left(C_2\right) = P^*\left(C_2 \cap C_3\right) + P^*\left(C_2 \cap C_3^c\right)$$

$$P^*\left(C_1 \cap C_2\right) = P\left(C_1 \middle| C_2\right) P^*\left(C_2\right)$$

$$P^*\left(C_1 \cap C_2^c\right) = P\left(C_1 \middle| C_2^c\right) P^*\left(C_2^c\right)$$

The conditional probability, $P\left(C_1 \middle| C_2\right)$ is not specified in the BBN so the Bayes' rule is used,

$$P\left(C_1 \middle| C_2\right) = \frac{P\left(C_2 \middle| C_1\right) P\left(C_1\right)}{P\left(C_2\right)}$$

Similarly with

$$P(C_1|C_2^c) = \frac{P(C_2^c|C_1)P(C_1)}{P(C_2^c)}$$

where $P(C_2^c|C_1) = 1 - P(C_2|C_1)$.

Marginalisation

$$P^*(C_1) = P^*(C_1 \cap C_2) + P^*(C_1 \cap C_2^c)$$

4.4.3.1.3 3.3.3.1.3 Bi-directional Flow Following the Instantiation of a Connecting Node

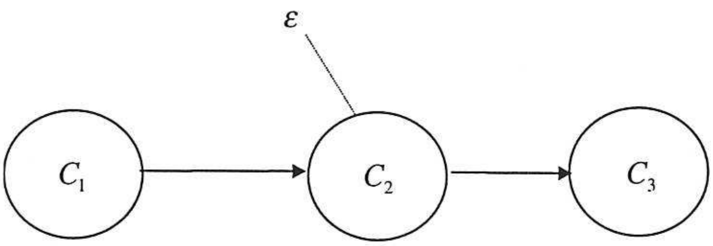


Figure 56 Instantiation of a connecting node.

Here, $P^*(C_2)$ is specified. This determines the complement, $P^*(C_2^c) = 1 - P^*(C_2)$. The flow of evidence is $P^*(C_2) \rightarrow P^*(C_1), P^*(C_3)$. The state of the variable in the connection is known and so blocks transmission of evidence between C_1 and C_3 rendering them independent.

$$P^*(C_1 \cap C_2) = P(C_1|C_2)P^*(C_2)$$
$$P^*(C_1 \cap C_2^c) = P(C_1|C_2^c)P^*(C_2^c)$$

The conditional probability, $P(C_1|C_2)$ is not specified in the BBN so the Bayes' rule is used,

$$P(C_1|C_2) = \frac{P(C_2|C_1)P(C_1)}{P(C_2)}$$

Similarly with

$$P(C_1|C_2^c) = \frac{P(C_2^c|C_1)P(C_1)}{P(C_2^c)}$$

where $P(C_2^c|C_1) = 1 - P(C_2|C_1)$.

Marginalisation

$$P^*(C_1) = P^*(C_1 \cap C_2) + P^*(C_1 \cap C_2^c)$$

$$\begin{aligned} P^*(C_2 \cap C_3) &= P(C_3|C_2)P^*(C_2) \\ &= \frac{P(C_2 \cap C_3)}{P(C_2)} P^*(C_2) \\ &= P(C_2 \cap C_3)\frac{P^*(C_2)}{P(C_2)} \end{aligned}$$

and,

$$\begin{aligned} P^*(C_2^c \cap C_3) &= P(C_2|C_2^c)P^*(C_2^c) \\ &= \frac{P(C_2^c \cap C_3)}{P(C_2^c)} P^*(C_2^c) \\ &= P(C_2^c \cap C_3)\frac{P^*(C_2^c)}{P(C_2^c)} \end{aligned}$$

Marginalising,

$$P^*(C_3) = P^*(C_2 \cap C_3) + P^*(C_2^c \cap C_3).$$

4.4.4 Diverging Connection

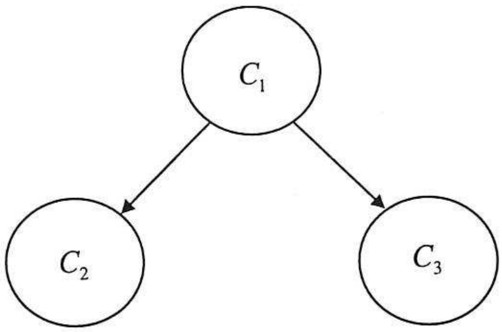


Figure 57 Diverging connection.

Evidence may be transmitted through a diverging connection unless it is instantiated. That is, evidence may be applied to a child node which then affects the other child nodes.

Instantiate a single child node and the parent can be instantiated, thus, rending the remaining child nodes independent and instantiated.

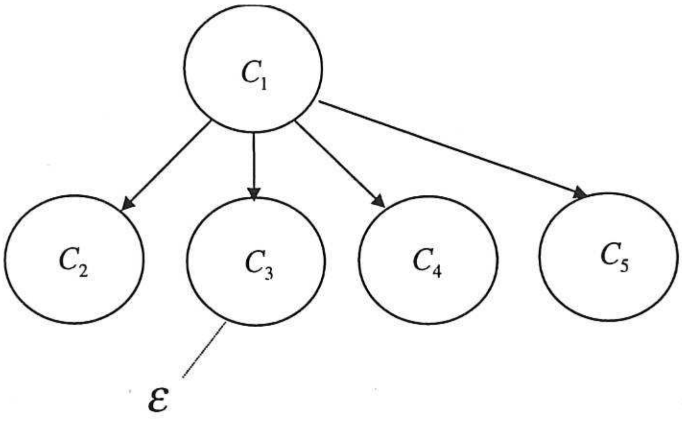


Figure 58 Multiple children in a diverging connection.

Influence can pass between all the children of C_1 unless the state of C_1 is known.

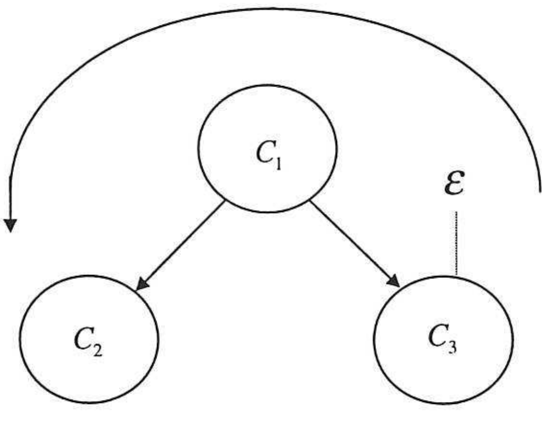


Figure 59 Transmission of evidence between child nodes.

4.4.4.1 Reverse Flow of Evidence in a Diverging Connection

Here, the flow of evidence is
$P^*(C_3) \to P^*(C_1) \to P^*(C_2)$

thus,
$P^*(C_1 \cap C_3) = P(C_1|C_3)P^*(C_3)$
and
$P^*(C_1 \cap C_3^c) = P(C_1|C_3^c)P^*(C_3^c)$

The conditional probability, $P(C_1|C_3)$ is not specified in the BBN so the Bayes' rule is used,

$$P(C_1|C_3)=\frac{P(C_3|C_1)P(C_1)}{P(C_3)}$$

Similarly with

$$P(C_1|C_3^c)=\frac{P(C_3^c|C_1)P(C_1)}{P(C_3^c)}$$

where $P(C_3^c|C_1)=1-P(C_3|C_1)$.

Marginalisation

$$P^*(C_1)=P^*(C_1\cap C_3)+P^*(C_1\cap C_3^c)$$

To calculate $P^*(C_2)$, the revised joint probabilities, $P^*(C_1\cap C_2)$ and $P^*(C_1^c\cap C_2)$ have to be calculated so that a marginalisation can be carried out.

Now,

$$\begin{aligned}P^*(C_1\cap C_2)&=P(C_2|C_1)P^*(C_1)\\&=\frac{P(C_1\cap C_2)}{P(C_1)}P^*(C_1)\\&=P(C_1\cap C_2)\frac{P^*(C_1)}{P(C_1)}\end{aligned}$$

and,

$$\begin{aligned}P^*(C_1^c\cap C_2)&=P(C_2|C_1^c)P^*(C_1^c)\\&=\frac{P(C_1^c\cap C_2)}{P(C_1^c)}P^*(C_1^c)\\&=P(C_1^c\cap C_2)\frac{P^*(C_1^c)}{P(C_1^c)}\end{aligned}$$

Marginalising,

$$P^*(C_2)=P^*(C_1\cap C_2)+P^*(C_1^c\cap C_2).$$

4.4.4.1.1 Bi-directional Flow Following the Instantiation of a Parent Node

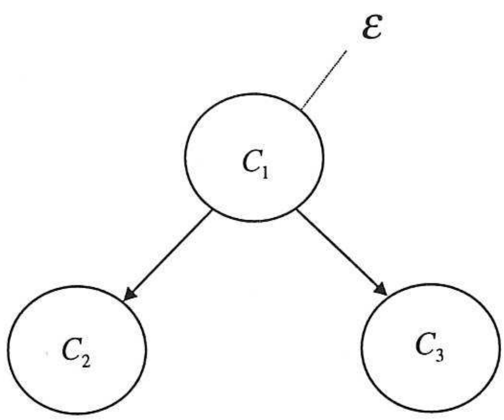


Figure 60. When the parent node is instansiated, the children are independent.

Here, the independence property is used to simplify the calculation (Krause and Clarke, 1993). Thus,

$$P(C_1 \cap C_2 \cap C_3) = P(C_2|C_1 \cap C_3)P(C_1|C_3)P(C_3)$$
$$= P(C_2|C_1)P(C_1|C_3)P(C_3)$$

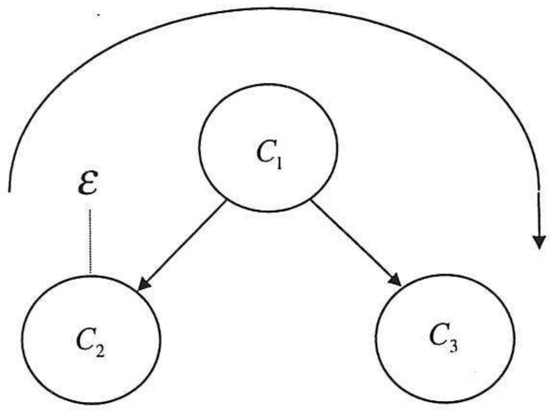


Figure 61.

Figure 61 shows the other child node as being instantiated in the "two child" example. This case is analogous to that illustrated in Figure 59.

4.4.5 Converging Connection

For a *converging connection*, The notion of conditional dependence is required. Here, evidence may be transmitted through a converging connection if either the variable in the connection or one of its descendants has received evidence.

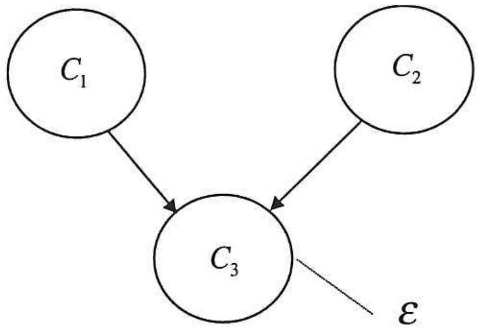


Figure 62. Evidence is applied to the node in the converging connection.

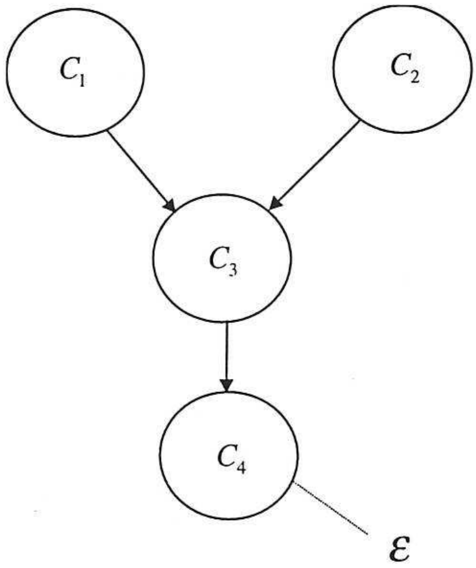


Figure 63. Applying evidence to a descendent of the connecting node.

If the state of C_3 is unknown, then C_1 and C_2 are independent. The certainty of C_3 depends upon all parent nodes. If evidence is applied to C_3, either directly, or indirectly through one of its child nodes, then the parent nodes may communicate.

All methods of transmitting evidence through variables is covered by the three cases. Given, $P(C_1)$ and $P(C_2)$, $P(C_1^c)$ and $P(C_2^c)$ are specified. The child probability may then be calculated using marginalisation. Thus,

$$P(C_3) = P(C_1 \cap C_2 \cap C_3) + P(C_1^c \cap C_2 \cap C_3) + P(C_1 \cap C_2^c \cap C_3) + P(C_1^c \cap C_2^c \cap C_3)$$

Now,

$$
\begin{aligned}
P(C_1 \cap C_2 \cap C_3) &= P(C_3 | C_1 \cap C_2) P(C_1 \cap C_2) \\
&= P(C_3 | C_1 \cap C_2) P(C_1 | C_2) P(C_2) \\
&= P(C_3 | C_1 \cap C_2) P(C_1) P(C_2)
\end{aligned}
$$

because node 3 is not instantiated.

$$
\begin{aligned}
P(C_1^c \cap C_2 \cap C_3) &= P(C_3 | C_1^c \cap C_2) P(C_1^c \cap C_2) \\
&= P(C_3 | C_1^c \cap C_2) P(C_1^c | C_2) P(C_2) \\
&= P(C_3 | C_1 \cap C_2) [1 - P(C_1 | C_2)] P(C_2) \\
&= P(C_3 | C_1 \cap C_2) [1 - P(C_1)] P(C_2) \\
&= P(C_3 | C_1 \cap C_2) P(C_1^c) P(C_2)
\end{aligned}
$$

4.4.6 An Example Network

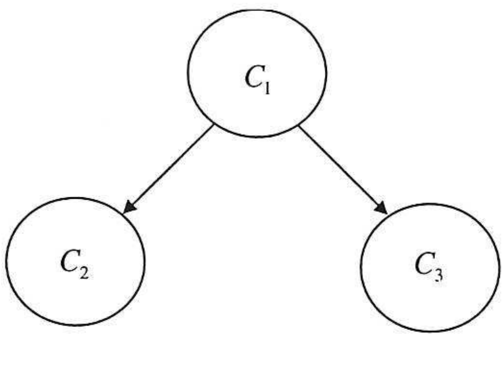


Figure 64. An example belief network.

From the network diagram, there is no causal relationship between C_2 and C_3. This gives a further two constraints as C_2 does not depend upon C_3 and vice versa.

For example the following probabilities may be estimated:

$P(C_1) = 0.6$, $P(C_2) = 0.5$, $P(C_3) = 0.4$, $P(C_1 \cap C_2) = 0.4$, $P(C_1 \cap C_3) = 0.3$

Leaving
$P(NF)$, $P(C_2 \cap C_3)$, and $P(C_1 \cap C_2 \cap C_3)$ which can be calculated using information from the Bayesian network.

4.4.6.1 The PPUE

Given the sub-unit class probabilities, the conditional probabilities can be calculated to show the effect of a subset of sub-units upon another subset of sub-units.

For example

$$P(C_2|C_1) = \frac{P(C_2 \cap C_1)}{P(C_1)} = \frac{0.4}{0.6} = 0.6667,$$

which quantifies the effect of sub-unit 1 on sub-unit 2. This is a special case of the PPUE. Note that some of these causal relationships may be quantified as part of prior information and may vitiate the need for the estimation of some of the probabilities.

Dependencies upon the complement of a sub-unit event may also be calculated; for example

$$P(C_2|C_1^c) = \frac{P(C_2 \cap C_1^c)}{P(C_1^c)} = \frac{P(C_2) - P(C_2 \cap C_1)}{1 - P(C_1)} = \frac{0.5 - 0.4}{0.4} = 0.25$$

which is also a special case of the PPUE. These quantities allow us to construct a table for $P(C_2|C_1)$ viz.:

$P(C_2\|C_1)$	C_1	C_1^c
C_2	0.6667	0.25
C_2^c	0.3333	0.75

Table 8

Where $P(C_2^c|C_1) = 1 - P(C_2|C_1)$ and $P(C_2^c|C_1^c) = 1 - P(C_2|C_1^c)$

Similarly,

$$P(C_3|C_1) = \frac{P(C_3 \cap C_1)}{P(C_1)} = \frac{0.3}{0.6} = 0.5.$$

$$P(C_3|C_1^c) = \frac{P(C_3 \cap C_1^c)}{P(C_1^c)} = \frac{P(C_3) - P(C_3 \cap C_1)}{1 - P(C_1)} = \frac{0.4 - 0.3}{0.4} = 0.25$$

These quantities allow us to construct a table for $P(C_3|C_1)$ viz:

$P(C_3\|C_1)$	C_1	C_1^c

C_3	0.5	0.25
C_3^c	0.5	0.75

Table 9

4.4.6.2 The Bayesian Belief Network

For *Bayesian networks*, the prior $P(C_1)$,would be given together with $P(C_2|C_1)$, $P(C_2|C_1^c)$, $P(C_3|C_1)$ and $P(C_3|C_1^c)$.

The tables would allow us to calculate

$$P(C_1 \cap C_2) = P(C_2|C_1).P(C_1) = (0.6667)(0.6) = 0.4\,,$$
$$P(C_1 \cap C_2^c) = P(C_2^c|C_1).P(C_1) = (0.3333)(0.6) = 0.2\,,$$
$$P(C_1^c \cap C_2) = P(C_2|C_1^c).P(C_1^c) = (0.25)(0.4) = 0.1\,,$$
$$P(C_1^c \cap C_2^c) = P(C_2^c|C_1^c).P(C_1^c) = (0.75)(0.4) = 0.3,$$

giving

$P(C_2 \cap C_1)$	C_1	C_1^c
C_2	0.4	0.1
C_2^c	0.2	0.3

Table 10

and similarly,

$$P(C_1 \cap C_3) = P(C_3|C_1).P(C_1) = (0.5)(0.6) = 0.3\,,$$
$$P(C_1 \cap C_3^c) = P(C_3^c|C_1).P(C_1) = (0.5)(0.6) = 0.3\,,$$
$$P(C_1^c \cap C_3) = P(C_3|C_1^c).P(C_1^c) = (0.25)(0.4) = 0.1\,,$$
$$P(C_1^c \cap C_3^c) = P(C_3^c|C_1^c).P(C_1^c) = (0.75)(0.4) = 0.3,$$

giving

$P(C_3 \cap C_1)$	C_1	C_1^c
C_3	0.3	0.1
C_3^c	0.3	0.3

Table 11

For the *Bayesian network*, we marginalise to get:

$$P(C_2) = P(C_1 \cap C_2) + P(C_1^c \cap C_2) = 0.4 + 0.1 = 0.5$$
$$P(C_3) = P(C_1 \cap C_3) + P(C_1^c \cap C_3) = 0.3 + 0.1 = 0.4$$

as expected.

4.4.6.3 Relationships Between the Network Node Variables

$$P(C_2 | C_1 \cap C_3) = P(C_2 | C_1)$$

Proof:

From conditional independence:

$$P(C_2 | C_1 \cap C_3) = \frac{P(C_1 \cap C_2 \cap C_3)}{P(C_1 \cap C_3)}$$
$$= \frac{P(C_2 \cap C_3 | C_1) P(C_1)}{P(C_3 | C_1) P(C_1)}$$
$$= \frac{P(C_2 | C_1) P(C_3 | C_1)}{P(C_3 | C_1)}$$
$$= P(C_2 | C_1)$$

$$P(C_1 \cap C_2 \cap C_3) = P(C_2 | C_1 \cap C_3) P(C_1 | C_3) P(C_3)$$
$$= P(C_2 | C_1) P(C_1 | C_3) P(C_3)$$

$$P(C_1 \cap C_2^c \cap C_3) = P(C_2^c | C_1 \cap C_3) P(C_1 | C_3) P(C_3)$$
$$= (1 - P(C_2 | C_1 \cap C_3)) P(C_1 | C_3) P(C_3)$$
$$= (1 - P(C_2 | C_1)) P(C_1 | C_3) P(C_3)$$
$$= P(C_2^c | C_1) P(C_1 | C_3) P(C_3)$$

$$P\left(C_1^c \cap C_2 \cap C_3\right) = P\left(C_2 \cap C_3 \middle| C_1^c\right)P\left(C_1^c\right)$$
$$= P\left(C_2 \middle| C_1^c\right)P\left(C_3 \middle| C_1^c\right)P\left(C_1^c\right)$$
$$= P\left(C_2 \middle| C_1^c\right)P\left(C_1^c \middle| C_3\right)P\left(C_3\right)$$

$$P\left(C_1^c \cap C_2^c \cap C_3\right) = P\left(C_2^c \cap C_3 \middle| C_1^c\right)P\left(C_1^c\right)$$
$$= P\left(C_2^c \middle| C_1^c\right)P\left(C_3 \middle| C_1^c\right)P\left(C_1^c\right)$$
$$= P\left(C_2^c \middle| C_1^c\right)P\left(C_1^c \middle| C_3\right)P\left(C_3\right)$$

For both Bayesian networks and PKI:

$$P\left(C_1 \cap C_2 \cap C_3\right) = P\left(C_2 \middle| C_1\right)P\left(C_1 \middle| C_3\right)P(C_3) = (0.6667)(0.75)(0.4) = 0.2$$
$$P\left(C_1 \cap C_2^c \cap C_3\right) = P\left(C_2^c \middle| C_1\right)P\left(C_1 \middle| C_3\right)P(C_3) = (0.3333)(0.75)(0.4) = 0.1$$
$$P\left(C_1^c \cap C_2 \cap C_3\right) = P\left(C_2 \middle| C_1^c\right)P\left(C_1^c \middle| C_3\right)P(C_3) = (0.25)(0.25)(0.4) = 0.025$$
$$P\left(C_1^c \cap C_2^c \cap C_3\right) = P\left(C_2^c \middle| C_1^c\right)P\left(C_1^c \middle| C_3\right)P\left(C_3\right) = (0.75)(0.25)(0.4) = 0.075$$

Marginalise:

$$P\left(C_2 \cap C_3\right) = P(C_1 \cap C_2 \cap C_3) + P(C_1^c \cap C_2 \cap C_3) = 0.2 + 0.025 = 0.225$$
$$P\left(C_2^c \cap C_3\right) = P(C_1 \cap C_2^c \cap C_3) + P(C_1^c \cap C_2^c \cap C_3) = 0.1 + 0.075 = 0.175$$

The values $P\left(C_2 \cap C_3\right)$ and $P\left(C_1 \cap C_2 \cap C_3\right)$ are added to the probability list giving:

$P\left(C_1\right) = 0.6$, $P\left(C_2\right) = 0.5$, $P\left(C_3\right) = 0.4$, $P\left(C_1 \cap C_2\right) = 0.4$, $P\left(C_1 \cap C_3\right) = 0.3$

$P\left(C_2 \cap C_3\right) = 0.225$, $P\left(C_1 \cap C_2 \cap C_3\right) = 0.2$

$P\left(NF\right) = ????$

Applying the 'unity' constraint gives:

$0.6 + 0.5 + 0.4 + P(NF) - 0.4 - 0.3 - 0.225 + 0.2 = 1$

Which implies that $P(NF) = 0.225$.

4.4.6.4 A Fault Occurs in C3

Say C3 has a fault; this means that evidence is introduced at the point shown in Figure 65. We want $P\left(C_1 \middle| C_3\right)$.

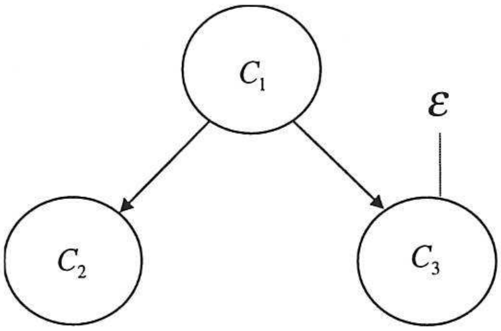


Figure 65. Evidence is applied to the child node representing fault class 3.

By Bayes' rule,

$$P(C_1|C_3) = \frac{P(C_3|C_1)P(C_1)}{P(C_3)} = \frac{(0.5)(0.6)}{0.4} = 0.75.$$

For PKI, (by PPUE)

$$P(C_1|C_3) = \frac{P(C_1 \cap C_3)}{P(C_3)} = \frac{0.3}{0.4} = 0.75$$

The probability of C1 being faulty has now increased from 0.6 to 0.75. Note that this relationship between C3 and C1 is not *causal*!!!! It is about the passing of information.

For BBN,

$$P^*(C_1 \cap C_3) = P(C_1|C_3)P^*(C_3) = (0.75)(1.0) = 0.75$$
$$P^*(C_1^c \cap C_3) = P(C_1^c|C_3)P^*(C_3) = (0.25)(1.0) = 0.25$$

or

$$P^*(C_1 \cap C_3) = P(C_1|C_3)P^*(C_3) = P(C_1|C_3) = 0.75$$
$$P^*(C_1^c \cap C_3) = P(C_1^c|C_3)P^*(C_3) = P(C_1^c|C_3) = 0.25$$

because $P^*(C_3) = 1$.

(Bayes on C3 complement)

Also, $P(C_1|C_3^c) = \dfrac{P(C_3^c|C_1)P(C_1)}{P(C_3^c)} = \dfrac{(0.5)(0.6)}{0.5} = 0.6$

$$P^*(C_1 \cap C_3^c) = P(C_1|C_3^c)P^*(C_3^c) = (0.6)(0) = 0$$
$$P^*(C_1^c \cap C_3^c) = P(C_1^c|C_3^c)P^*(C_3^c) = (0.25)(0) = 0$$
because $P^*(C_3^c) = 0$.

$$P^*(C_1) = P^*(C_1 \cap C_3) + P^*(C_1 \cap C_3^c) = 0.75 + 0 = 0.75$$

and

$$P^*(C_{1_1}^c) = P^*(C_{1_1}^c \cap C_3^c) + P^*(C_1 \cap C_3^c) = 0.25 + 0 = 0.75$$

Here, $P^*(C_1 \cap C_3)$ and $P^*(C_1^c \cap C_3)$ are the new values for $P(C_1)$ and $P(C_1^c)$ because C3 has occurred!!!!

So,

$$P^*(C_3) = 1$$
$$P^*(C_1) = P^*(C_1 \cap C_3) = 0.75$$

$$P^*(C_1 \cap C_2) = P(C_2|C_1).P^*(C_1) = (0.6667)(0.75) = 0.5,$$

$$P^*(C_1^c \cap C_2) = P(C_2|C_1^c).P^*(C_1^c) = (0.25)(0.25) = 0.125$$

$$P^*(C_2) = P^*(C_1 \cap C_2) + P^*(C_{,1}^c \cap C_2) = 0.5 + 0.125 = 0.625$$

Now,

$$P^*(C_1 \cap C_2 \cap C_3) = P(C_2|C_1 \cap C_3)P(C_1|C_3)P^*(C_3)$$
$$= P(C_2|C_1)P(C_1|C_3).1$$
$$= P(C_2|C_1)P(C_1|C_3)$$

giving
$$P^*(C_1 \cap C_2 \cap C_3) = P(C_2|C_1)P(C_1|C_3)P^*(C_3) = (0.6667)(0.75)(1.0) = 0.5$$

similarly
$$P^*(C_1 \cap C_2^c \cap C_3) = P(C_2^c|C_1)P(C_1|C_3)P^*(C_3) = (0.3333)(0.75)(1.0) = 0.25$$

$$P^*(C_1^c \cap C_2 \cap C_3) = P(C_2 | C_1^c) P(C_1^c | C_3) P^*(C_3) = (0.25)(0.25)(1.0) = 0.0625$$
$$P^*(C_1^c \cap C_2^c \cap C_3) = P(C_2^c | C_1^c) P(C_1^c | C_3) P^*(C_3) = (0.75)(0.25)(1.0) = 0.1875$$

Marginalise:

$$P^*(C_2 \cap C_3) = P^*(C_1 \cap C_2 \cap C_3) + P^*(C_1^c \cap C_2 \cap C_3) = 0.5 + 0.0625 = 0.5625$$

$$P^*(C_2^c \cap C_3) = P^*(C_1 \cap C_2^c \cap C_3) + P^*(C_1^c \cap C_2^c \cap C_3) = 0.25 + 0.1875 = 0.4375$$

The value $P^*(C_2 \cap C_3)$ is the new probability of C2, $P^*(C_2)$, because C3 has occurred.
So,

$$P^*(C_3) = 1$$
$$P^*(C_1) = P^*(C_1 \cap C_3) = 0.75$$
$$P^*(C_1 \cap C_2) = 0.5$$
$$P^*(C_2) = P^*(C_2 \cap C_3) = 0.5625$$
$$P^*(C_1 \cap C_2 \cap C_3) = 0.5$$

As C3 has occurred, $P(NF) = 0$ which is confirmed by

$$P(C_1) + P(C_2) + P(C_3) + P(NF) - P(C_1 \cap C_2) - P(C_1 \cap C_3) - P(C_2 \cap C_3) + P(C_1 \cap C_2 \cap C_3) = 1$$

$$0.75 + 0.5625 + 1 + P(NF) - 0.5 - 0.75 - 0.5625 = 1$$
$$\Rightarrow P(NF) = 0$$

4.4.7 What about when probabilities are changed in the range (0,1)?

Now,

$$P^*(C_1 \cap C_3)$$
$$= P(C_1 | C_3) P^*(C_3)$$
$$= \frac{P(C_1 \cap C_3)}{P(C_3)} P^*(C_3)$$
$$P(C_1 \cap C_3) \frac{P^*(C_3)}{P(C_3)}$$

4.4.7.1 The Probability of C3 is Reduced.

Start with original *values* and substitute

$$P^*(C_3) = 0.2$$

$$P^*(C_1 \cap C_3) = P(C_1|C_3)P^*(C_3) = (0.75)(0.2) = 0.15.$$

Equivalently,

$$P^*(C_1 \cap C_3) = P(C_1 \cap C_3)\frac{P^*(C_3)}{P(C_3)}$$

$$= (0.3)\left(\frac{0.2}{0.4}\right)$$

$$= 0.15$$

Using the 'old' values (applying the PPUE) gives

$$P(C_1|C_3^c) = \frac{P(C_1 \cap C_3^c)}{P(C_3^c)} = \frac{P(C_1) - P(C_1 \cap C_3)}{1 - P(C_3)} = \frac{0.6 - 0.3}{1 - 0.4} = 0.5$$

Therefore,

$$P^*(C_1 \cap C_3^c) = P(C_1|C_3^c)P^*(C_3^c) = (0.5)(0.8) = 0.4$$

$$P^*(C_1 \cap C_3^c) = P(C_1 \cap C_3^c)\frac{P^*(C_3^c)}{P(C_3^c)}$$

$$= (0.3)\left(\frac{0.8}{0.6}\right)$$

$$= 0.4$$

Marginalising,

$$P^*(C_1) = P^*(C_1 \cap C_3) + P^*(C_1 \cap C_3^c) = 0.15 + 0.4 = 0.55$$

$$P^*(C_3) = 0.2$$

$$P^*(C_1 \cap C_3) = 0.15$$

$$P^*(C_1) = 0.55$$

$$P(C_1 \cap C_2) = P(C_2|C_1).P(C_1) = (0.6667)(0.55) = 0.3667,$$

$$P(C_1^c \cap C_2) = P(C_2|C_1^c).P(C_1^c) = (0.25)(0.45) = 0.1125$$

Again,

$$P^*(C_1 \cap C_2) = P(C_1 \cap C_2)\frac{P^*(C_1)}{P(C_1)}$$
$$= (0.4)\left(\frac{0.55}{0.6}\right)$$
$$= 0.3667$$

$$P^*(C_1^c \cap C_2) = P(C_1^c \cap C_2)\frac{P^*(C_1^c)}{P(C_1^c)}$$
$$= (0.1)\left(\frac{0.45}{0.4}\right)$$
$$= 0.1125$$

Marginalising:

$$P^*(C_2) = P^*(C_1 \cap C_2) + P^*(C_1^c \cap C_2) = 0.3667 + 0.1125 = 0.4792$$

$$P^*(C_1 \cap C_2 \cap C_3) = P(C_2|C_1)P(C_1|C_3)P^*(C_3) = (0.6667)(0.75)(0.2) = 0.1$$

$$P^*(C_1^c \cap C_2 \cap C_3) = P(C_2|C_1^c)P(C_1^c|C_3)P^*(C_3) = (0.25)(0.25)(0.2) = 0.0125$$

Marginalising:

$$P^*(C_2 \cap C_3) = P^*(C_1 \cap C_2 \cap C_3) + P^*(C_1^c \cap C_2 \cap C_3) = 0.1 + 0.0125 = 0.1125$$

$$P^*(C_3) = 0.2$$

$$P^*(C_1 \cap C_3) = 0.15$$

$$P^*(C_1) = 0.55$$
$$P(C_1 \cap C_2) = 0.3667$$
$$P(C_2) = 0.4792$$

$$P^*(C_2 \cap C_3) = 0.1125$$
$$P^*(C_1 \cap C_2 \cap C_3) = 0.1$$

Applying the 'unity' constraint gives:

$$0.55 + 0.4792 + 0.2 + P(NF) - 0.3667 - 0.15 - 0.1125 + 0.1 = 1$$

Which implies that

$$P^*(NF) = 0.3.$$

$$P^*(NF) = P(NF)\left(\frac{P^*(C_3^c)}{P(C_3^c)}\right) = (0.225)\left(\frac{0.8}{0.6}\right) = 0.3$$

4.4.7.2 How would this be done by PKI (PPUE)?

Given the original values,

$$P(C_1) = 0.6$$
$$P(C_2) = 0.5$$
$$P(C_3) = 0.4$$
$$P(NF) = 0.225.$$
$$P(C_1 \cap C_2) = 0.4$$
$$P(C_1 \cap C_3) = 0.3$$
$$P(C_2 \cap C_3) = 0.225$$
$$P(C_1 \cap C_2 \cap C_3) = 0.2$$

the PPUE can be applied.

$$P(C_1|C_3) = \frac{P(C_1 \cap C_3)}{P(C_3)} = \frac{0.3}{0.4} = 0.75$$

$$P(C_1|C_3^c) = \frac{P(C_1) - P(C_1 \cap C_3)}{P(U) - P(C_3)} = \frac{0.6 - 0.3}{1 - 0.4} = 0.5$$

$$P(C_2|C_3) = \frac{P(C_2 \cap C_3)}{P(C_3)} = \frac{0.225}{0.4} = 0.5625$$

$$P\left(C_2\middle|C_3^c\right)=\frac{P(C_2)-P(C_2\cap C_3)}{P(U)-P(C_3)}=\frac{0.5-0.225}{1-0.4}=0.4583$$

$$P^*(C_1\cap C_3)=P\left(C_1\middle|C_3\right)P^*(C_3)=(0.75)(0.2)=0.5$$

$$P^*\left(C_1\cap C_3^c\right)=P\left(C_1\middle|C_3^c\right)P^*\left(C_3^c\right)=(0.5)(0.8)=0.4$$

Marginalise:

$$P^*(C_1)=P^*(C_1\cap C_3)+P^*\left(C_1\cap C_3^c\right)=0.15+0.4=0.55$$

$$P^*(C_2\cap C_3)=P\left(C_2\middle|C_3\right)P^*(C_3)=(0.5625)(0.2)=0.1125$$

$$P^*\left(C_2\cap C_3^c\right)=P\left(C_2\middle|C_3^c\right)P^*\left(C_3^c\right)=(0.4583)(0.8)=0.3667$$

Marginalising:

$$P^*(C_2)=P^*(C_2\cap C_3)+P^*\left(C_2\cap C_3^c\right)=0.4792$$

$$P\left(C_1\cap C_2\middle|C_3\right)=\frac{P(C_1\cap C_2\cap C_3)}{P(C_3)}=\frac{0.2}{0.4}=0.5$$

$$P^*(C_1\cap C_2\cap C_3)=P\left(C_1\cap C_2\middle|C_3\right)P^*(C_3)=(0.5)(0.2)=0.1$$

$$P\left(C_1\cap C_2\middle|C_3^c\right)=\frac{P(C_1\cap C_2)-P(C_1\cap C_2\cap C_3)}{P(U)-P(C_3)}=\frac{0.4-0.2}{0.6}=\frac{0.2}{0.6}=0.3333$$

$$P^*\left(C_1\cap C_2\cap C_3^c\right)=P\left(C_1\cap C_2\middle|C_3^c\right)P^*\left(C_3^c\right)=(0.3333)(0.8)=0.2667$$

Marginalising:

$$P^*(C_1\cap C_2)=P^*(C_1\cap C_2\cap C_3)+P^*\left(C_1\cap C_2\cap C_3^c\right)=0.3667$$

Taken together, these results again imply that

$P^*(NF)=0.3$.

PPUE

$P(C_i^c|\varepsilon) = 1 - P(C_i|\varepsilon)$

Proof:

$P(C_i^c|\varepsilon) = P(U \cap C_i^c|\varepsilon) = P(U|\varepsilon) - P(U \cap C_i|\varepsilon) = 1 - P(C_i|\varepsilon)$

What about knowledge of C1?

4.4.8 A Slightly More Complex Example

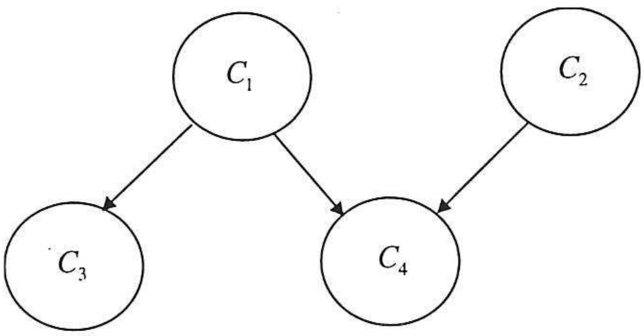


Figure 66 A more complex example.

The network can be decomposed into two sub-nets connected by C_1. One sub-net consists of a single diverging connection and the other sub-net consists of a single converging connection.

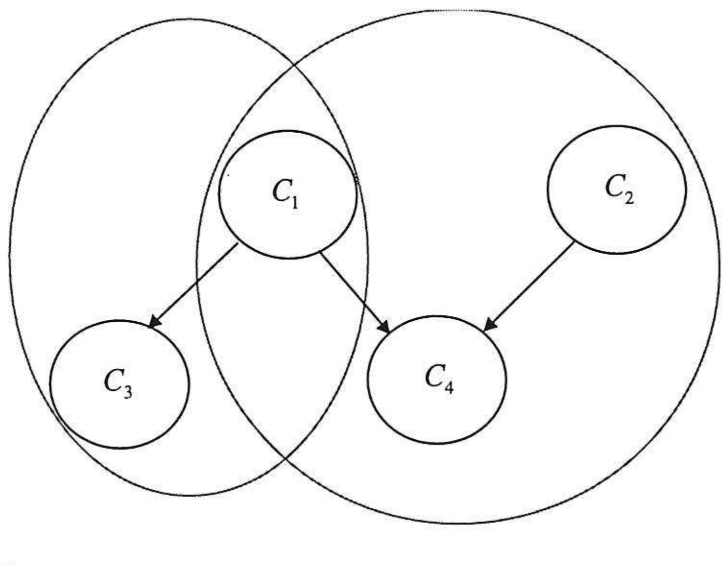


Figure 67. The BBN is split into two sub-networks connected by C1.

Given the values of the two parent nodes, the values of the child nodes may be derived.

For example,
$P(C_1) = 0.3$
$P(C_2) = 0.2$
which imply that $P(C_1^c) = 0.7$ and $P(C_2^c) = 0.8$ respectively.
In order to propagate the information on the state of the two parent nodes through the network, the causal relations governing the links are required.

For the first sub-network, $P(C_3|C_1) = 0.8$ and $P(C_3|C_1^c) = 0.4$ which imply that $P(C_3^c|C_1) = 0.2$ and $P(C_3^c|C_1^c) = 0.6$ respectively as shown in table

$P(C_3\|C_1)$	C_1	C_1^c
C_3	0.8	0.4
C_3^c	0.2	0.6

Table 12

Also,

$P(C_4|C_1 \cap C_2) = 0.9$
$P(C_4|C_1 \cap C_2^c) = 0.6$
$P(C_4|C_1^c \cap C_2) = 0.5$
$P(C_4|C_1^c \cap C_2^c) = 0.3$
Which implies that
$P(C_4^c|C_1 \cap C_2) = 0.1$
$P(C_4^c|C_1 \cap C_2^c) = 0.4$
$P(C_4^c|C_1^c \cap C_2) = 0.5$
$P(C_4^c|C_1^c \cap C_2^c) = 0.7$

respectively.

$P(C_4\|C_1, C_2)$	C_2	C_2^c
C_1	(0.9,0.1)	(0.6,0.4)
C_1^c	(0.5,0.5)	(0.3,0,7)

Table 13

First, calculate the certainty of node 3 in the first sub-net by finding the 'double products' $P(C_1 \cap C_3)$ and $P(C_1^c \cap C_3)$ and then marginalising.

$$P(C_1 \cap C_3) = P(C_3|C_1)P(C_1) = (0.8)(0.3) = 0.24$$
$$P(C_1^c \cap C_3) = P(C_3|C_1^c)P(C_1^c) = (0.4)(0.7) = 0.28$$

$$P(C_1 \cap C_3^c) = P(C_3^c|C_1)P(C_1) = (0.2)(0.3) = 0.06$$
$$P(C_1^c \cap C_3^c) = P(C_3^c|C_1^c)P(C_1^c) = (0.6)(0.7) = 0.42$$

Marginalising:

$$P(C_3) = P(C_1 \cap C_3) + P(C_1^c \cap C_3) = 0.24 + 0.28 = 0.52$$

Next, the 'triple products' are calculated in the second sub-net

$$P(C_4 \cap C_1 \cap C_2) = P(C_4|C_1 \cap C_2)P(C_1)P(C_2) = (0.9)(0.3)(0.2) = 0.054$$
$$P(C_4 \cap C_1 \cap C_2^c) = P(C_4|C_1 \cap C_2^c)P(C_1)P(C_2^c) = (0.6)(0.3)(0.8) = 0.144$$
$$P(C_4 \cap C_1^c \cap C_2) = P(C_4|C_1^c \cap C_2)P(C_1^c)P(C_2) = (0.5)(0.7)(0.2) = 0.07$$
$$P(C_4 \cap C_1^c \cap C_2^c) = P(C_4|C_1^c \cap C_2^c)P(C_1^c)P(C_2^c) = (0.3)(0.7)(0.8) = 0.168$$

the complements:

$$P(C_4^c \cap C_1 \cap C_2) = P(C_4^c|C_1 \cap C_2)P(C_1)P(C_2) = (0.1)(0.3)(0.2) = 0.006$$
$$P(C_4^c \cap C_1 \cap C_2^c) = P(C_4^c|C_1 \cap C_2^c)P(C_1)P(C_2^c) = (0.4)(0.3)(0.8) = 0.096$$
$$P(C_4^c \cap C_1^c \cap C_2) = P(C_4^c|C_1^c \cap C_2)P(C_1^c)P(C_2) = (0.5)(0.7)(0.2) = 0.07$$
$$P(C_4^c \cap C_1^c \cap C_2^c) = P(C_4^c|C_1^c \cap C_2^c)P(C_1^c)P(C_2^c) = (0.7)(0.7)(0.8) = 0.392$$

Marginalising:

$$P(C_4 \cap C_1) = P(C_4 \cap C_1 \cap C_2) + P(C_4 \cap C_1 \cap C_2^c) = 0.054 + 0.144 = 0.198$$
$$P(C_4 \cap C_1^c) = P(C_4 \cap C_1^c \cap C_2) + P(C_4 \cap C_1^c \cap C_2^c) = 0.07 + 0.168 = 0.238$$

$$P(C_4) = P(C_4 \cap C_1) + P(C_4 \cap C_1^c) = 0.198 + 0.238 = 0.436$$

4.4.8.1 The Probability of C4 is Increased.

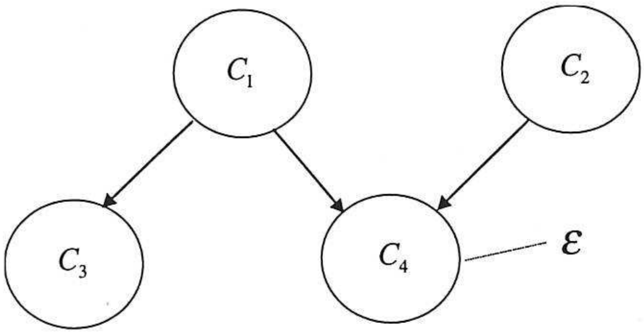


Figure 68

Here, evidence is applied to node 4.

$$P^*(C_4) = 0.9$$

$$R_4^* = \frac{P^*(C_4)}{P(C_4)} = \frac{0.9}{0.436} = 2.0642$$

and

$$RC_4^* = \frac{P^*(C_4^c)}{P(C_4^c)} = \frac{0.1}{0.564} = 0.1773$$

The second sub-net consists of a divergent connection making C_1 and C_2 dependent when C_4 is instantiated. For the first sub-net, C_4 can influence C_1 and, hence, C_3 owing to an uninstantiated parent node in a divergent connection.
Abductive reasoning is required to go from C_4 to C_1 and C_2.

For a convergent node, the child node is specified in terms of the link probabilities conditioned on the parents.

For a marginalisation to be carried out following evidence applied at the child node, the relevant joint probability components have to be calculated. For example,

$$P^*(C_1) = P^*(C_4 \cap C_1 \cap C_2) + P^*(C_4^c \cap C_1 \cap C_2) + P^*(C_4 \cap C_1 \cap C_2^c) + P^*(C_4^c \cap C_1 \cap C_2^c)$$

Where,

$$P^*(C_4 \cap C_1 \cap C_2) = P(C_1 \cap C_2 | C_4) P^*(C_4)$$

and so on.

Now,

$$P\left(C_1 \cap C_2 \middle| C_4\right) = \frac{P\left(C_4 \middle| C_1 \cap C_2\right) P\left(C_1 \cap C_2\right)}{P\left(C_4\right)}$$

By Bayes' rule.

Therefore,

$$\begin{aligned} P^*\left(C_4 \cap C_1 \cap C_2\right) &= P\left(C_1 \cap C_2 \middle| C_4\right) P^*\left(C_4\right) \\ &= \frac{P\left(C_4 \cap C_1 \cap C_2\right) P^*\left(C_4\right)}{P\left(C_4\right)} \\ &= P\left(C_4 \cap C_1 \cap C_2\right) \frac{P^*\left(C_4\right)}{P\left(C_4\right)} \\ &= P\left(C_4 \cap C_1 \cap C_2\right) R_4^* \end{aligned}$$

where,

$$P\left(C_4 \cap C_1 \cap C_2\right) = P\left(C_4 \middle| C_1 \cap C_2\right) P\left(C_1\right) P\left(C_2\right)$$

from before.

For a marginalisation to give $P^*\left(C_2\right)$, a further four joint probabilities are required giving 8 in all. Following a similar line of reasoning for the remaining 7 cases gives:

$$P^*\left(C_4 \cap C_1 \cap C_2\right) = P\left(C_4 \cap C_1 \cap C_2\right) R_4^* = (0.054)(2.0642) = 0.1115$$
$$P^*\left(C_4 \cap C_1 \cap C_2^c\right) = P\left(C_4 \cap C_1 \cap C_2^c\right) R_4^* = (0.144)(2.0642) = 0.2972$$
$$P^*\left(C_4 \cap C_1^c \cap C_2\right) = P\left(C_4 \cap C_1^c \cap C_2\right) R_4^* = (0.07)(2.0642) = 0.1445$$
$$P^*\left(C_4 \cap C_1^c \cap C_2^c\right) = P\left(C_4 \cap C_1^c \cap C_2^c\right) R_4^* = (0.168)(2.0642) = 0.3468$$
$$P^*\left(C_4^c \cap C_1 \cap C_2\right) = P\left(C_4^c \cap C_1 \cap C_2\right) RC_4^* = (0.006)(0.1773) = 0.0011$$
$$P^*\left(C_4^c \cap C_1 \cap C_2^c\right) = P\left(C_4^c \cap C_1 \cap C_2^c\right) RC_4^* = (0.096)(0.1773) = 0.0170$$
$$P^*\left(C_4^c \cap C_1^c \cap C_2\right) = P\left(C_4^c \cap C_1^c \cap C_2\right) RC_4^* = (0.07)(0.1773) = 0.0124$$
$$P^*\left(C_4^c \cap C_1^c \cap C_2^c\right) = P\left(C_4^c \cap C_1^c \cap C_2^c\right) RC_4^* = (0.392)(0.1773) = 0.0695$$

Marginalising:

$$P^*(C_1 \cap C_2) = P^*(C_4 \cap C_1 \cap C_2) + P^*(C_4^c \cap C_1 \cap C_2) = 0.1115 + 0.0011 = 0.1126$$
$$P^*(C_1 \cap C_2^c) = P^*(C_4 \cap C_1 \cap C_2^c) + P^*(C_4^c \cap C_1 \cap C_2^c) = 0.2972 + 0.0170 = 0.3142$$

Marginalising again:

$$P^*(C_1) = P^*(C_1 \cap C_2) + P^*(C_1 \cap C_2^c) = 0.1126 + 0.3142 = 0.4268$$

and finally,

$$P^*(C_1^c \cap C_2) = P^*(C_4 \cap C_1^c \cap C_2) + P^*(C_4^c \cap C_1^c \cap C_2) = 0.1445 + 0.0124 = 0.1569$$

giving

$$P^*(C_2) = P^*(C_1 \cap C_2) + P^*(C_1^c \cap C_2) = 0.1126 + 0.1569 = 0.2695$$

Now, a change in C_1 will affect C_3 within sub-net 1:

$$P^*(C_1) = 0.4268$$

$$R_1^* = \frac{P^*(C_1)}{P(C_1)} = \frac{0.4628}{0.3} = 1.5427$$

and

$$RC_1^* = \frac{P^*(C_1^c)}{P(C_1)} = \frac{0.5372}{0.7} = 0.7674$$

$$P^*(C_1 \cap C_3) = P(C_1 \cap C_3)R_1^* = (0.24)(1.5427) = 0.3702$$
$$P^*(C_1^c \cap C_3) = P(C_1^c \cap C_3)RC_1^* = (0.28)(0.7674) = 0.2149$$

$$P^*(C_1 \cap C_3^c) = P(C_1 \cap C_3^c)R_1^* = (0.06)(1.5427) = 0.0926$$
$$P^*(C_1^c \cap C_3^c) = P(C_1^c \cap C_3^c)RC_1^* = (0.42)(0.7674) = 0.3223$$

$$P^*(C_3) = P^*(C_1 \cap C_3) + P^*(C_1^c \cap C_3) = 0.3702 + 0.2149 = 0.5851$$

So, the initial state of the system given by

$$P(C_1) = 0.3$$

$P(C_2) = 0.2$
$P(C_3) = 0.52$
$P(C_4) = 0.436$

gives rise to the state

$P^*(C_1) = 0.4268$
$P^*(C_2) = 0.2695$
$P^*(C_3) = 0.5851$
$P^*(C_4) = 0.9$

when evidence is applied to node 4.

4.4.8.2 A Fault Occurs in C3 (again)

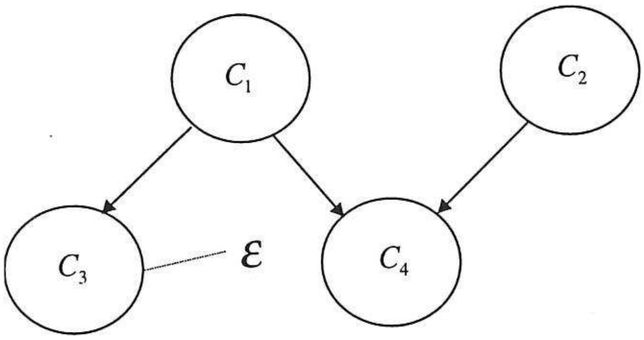


Figure 69

$P^{**}(C_3) = 1.0$

$$R_3^{**} = \frac{P^{**}(C_3)}{P^*(C_3)} = \frac{1.0}{0.5851} = 1.7091$$

$$RC_3^{**} = \frac{P^{**}(C_3^c)}{P^*(C_3^c)} = \frac{0.0}{0.4149} = 0.0$$

$P^{**}(C_1 \cap C_3) = P^*(C_1 \cap C_3)R_3^{**} = (0.3702)(1.7091) = 0.6327$
$P^{**}(C_1 \cap C_3^c) = P^*(C_1 \cap C_3^c)RC_3^* = (0.06)(0.0) = 0.0$

Marginalising:

$P^{**}(C_1) = P^{**}(C_1 \cap C_3) + P^{**}(C_1 \cap C_3^C) = 0.6327 + 0.0 = 0.6327$

4.5 The Application of BBNs in PKI.

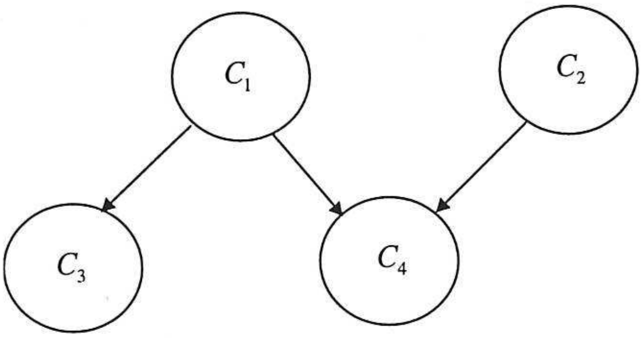


Figure 70

Assume that the reading vector is [0011] and that the scenario vector is [0010]

$P(C_3 \mid C_1)$	C_1	C_1^c
C_3	0.8	0.4
C_3^c	0.2	0.6

Table 14

$P(C_4 \mid C_1, C_2)$	C_2	C_2^c
C_1	(0.9,0.1)	(0.6,0.4)
C_1^c	(0.5,0.5)	(0.3,0,7)

Table 15

The state of the system prior is determined before PKI is applied. Given the values of the two parent nodes, the values of the child nodes may be derived.

For example,

$P(C_1) = 0.6$

$P(C_2) = 0.7$

which imply that $P(C_1^c) = 0.4$ and $P(C_2^c) = 0.3$ respectively.

First, calculate the certainty of node 3 in the first sub-net by finding the 'double products' $P(C_1 \cap C_3)$ and $P(C_1^c \cap C_3)$ and then marginalising.

$$P(C_1 \cap C_3) = P(C_3 \mid C_1)P(C_1) = (0.8)(0.6) = 0.48$$

$$P(C_1^c \cap C_3) = P(C_3 \mid C_1^c)P(C_1^c) = (0.4)(0.4) = 0.16$$

Marginalising:

$$P(C_3) = P(C_1 \cap C_3) + P(C_1^c \cap C_3) = 0.48 + 0.16 = 0.64$$

Check:

$$P(C_1 \cap C_3^c) = P(C_3^c | C_1)P(C_1) = (0.2)(0.6) = 0.12$$
$$P(C_1^c \cap C_3^c) = P(C_3^c | C_1^c)P(C_1^c) = (0.6)(0.4) = 0.24$$

$$P(C_3^c) = P(C_1 \cap C_3^c) + P(C_1^c \cap C_3^c) = 0.12 + 0.24 = 0.36$$

Next, the 'triple products' are calculated in the second sub-net

$$P(C_4 \cap C_1 \cap C_2) = P(C_4 | C_1 \cap C_2)P(C_1)P(C_2) = (0.9)(0.6)(0.7) = 0.378$$
$$P(C_4 \cap C_1 \cap C_2^c) = P(C_4 | C_1 \cap C_2^c)P(C_1)P(C_2^c) = (0.6)(0.6)(0.3) = 0.108$$
$$P(C_4 \cap C_1^c \cap C_2) = P(C_4 | C_1^c \cap C_2)P(C_1^c)P(C_2) = (0.5)(0.4)(0.7) = 0.14$$
$$P(C_4 \cap C_1^c \cap C_2^c) = P(C_4 | C_1^c \cap C_2^c)P(C_1^c)P(C_2^c) = (0.3)(0.4)(0.3) = 0.036$$

the complements:

$$P(C_4^c \cap C_1 \cap C_2) = P(C_4^c | C_1 \cap C_2)P(C_1)P(C_2) = (0.1)(0.6)(0.7) = 0.042$$
$$P(C_4^c \cap C_1 \cap C_2^c) = P(C_4^c | C_1 \cap C_2^c)P(C_1)P(C_2^c) = (0.4)(0.4)(0.3) = 0.048$$
$$P(C_4^c \cap C_1^c \cap C_2) = P(C_4^c | C_1^c \cap C_2)P(C_1^c)P(C_2) = (0.5)(0.4)(0.6) = 0.12$$
$$P(C_4^c \cap C_1^c \cap C_2^c) = P(C_4^c | C_1^c \cap C_2^c)P(C_1^c)P(C_2^c) = (0.7)(0.4)(0.3) = 0.084$$

Marginalising:

$$P(C_4 \cap C_1) = P(C_4 \cap C_1 \cap C_2) + P(C_4 \cap C_1 \cap C_2^c) = 0.378 + 0.108 = 0.486$$
$$P(C_4 \cap C_1^c) = P(C_4 \cap C_1^c \cap C_2) + P(C_4 \cap C_1^c \cap C_2^c) = 0.14 + 0.036 = 0.176$$

$$P(C_4) = P(C_4 \cap C_1) + P(C_4 \cap C_1^c) = 0.486 + 0.176 = 0.662$$

The posterior probabilities for sub-unit faults (given the reading vector) are

$P(C_1) = 0.6$

$P(C_2) = 0.7$

$P(C_3) = 0.64$

$P\left(C_4\right) = 0.662$

The MAP order is therefore

sub-units 2,4,3,1

Both MAP and PKI advise checking sub-unit 2 first as there is no posterior knowledge yet.

Inspecting sub-unit 2 reveals that it is not faulty, thus $P^*\left(C_2\right) = 0$.
The first evidence is now applied to node 2 as shown in figure 70.

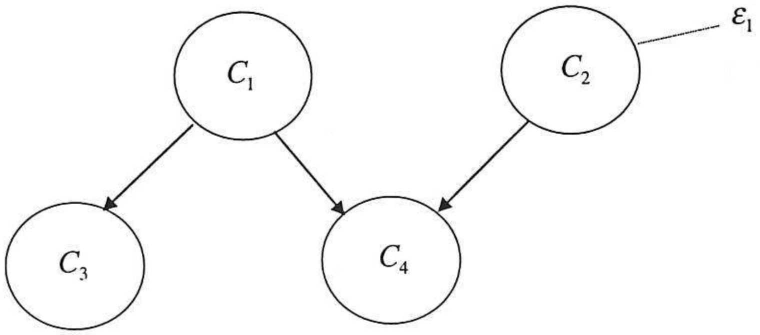


Figure 71

$$R_2^* = \frac{P^*\left(C_2\right)}{P\left(C_4\right)} = \frac{0.0}{0.7} = 0.0$$

and

$$RC_2^* = \frac{P^*\left(C_2^c\right)}{P\left(C_2^c\right)} = \frac{1.0}{0.3} = 3.3333$$

Now, node 2 is a parent node in a converging connection where the variable in the connection has not been instantiated. It follows that sub-units 1 and 2 are independent giving

$P^*\left(C_1\right) = P\left(C_1\right) = 0.6$,

and,

$P^*(C_3) = P(C_3) = P(C_1 \cap C_3) + P(C_1^c \cap C_3) = 0.48 + 0.16 = 0.64$

as before.

$$P^*\left(C_4 \cap C_1 \cap C_2\right) = P\left(C_4 \cap C_1 \cap C_2\right)R_2^* = (0.378)(0.0) = 0.0$$
$$P^*\left(C_4 \cap C_1 \cap C_2^c\right) = P\left(C_4 \cap C_1 \cap C_2^c\right)RC_2^* = (0.108)(3.3333) = 0.36$$

$$P^*(C_4 \cap C_1^c \cap C_2) = P(C_4 \cap C_1^c \cap C_2)R_2^* = (0.14)(0.0) = 0.0$$
$$P^*(C_4 \cap C_1^c \cap C_2^c) = P(C_4 \cap C_1^c \cap C_2^c)RC_2^* = (0.036)(3.3333) = 0.12$$
$$P^*(C_4^c \cap C_1 \cap C_2) = P(C_4^c \cap C_1 \cap C_2)R_2^* = (0.042)(0.0) = 0.0$$
$$P^*(C_4^c \cap C_1 \cap C_2^c) = P(C_4^c \cap C_1 \cap C_2^c)RC_2^* = (0.048)(3.3333) = 0.16$$
$$P^*(C_4^c \cap C_1^c \cap C_2) = P(C_4^c \cap C_1^c \cap C_2)R_2^* = (0.12)(0.0) = 0.0$$
$$P^*(C_4^c \cap C_1^c \cap C_2^c) = P(C_4^c \cap C_1^c \cap C_2^c)RC_2^* = (0.084)(3.3333) = 0.28$$

$$P^*(C_4 \cap C_1) = P^*(C_4 \cap C_1 \cap C_2) + P^*(C_4 \cap C_1 \cap C_2^c) = 0.0 + 0.36 = 0.36$$
$$P^*(C_4 \cap C_1^c) = P^*(C_4 \cap C_1^c \cap C_2) + P^*(C_4 \cap C_1^c \cap C_2^c) = 0.0 + 0.12 = 0.12$$

$$P^*(C_4) = P^*(C_4 \cap C_1) + P^*(C_4 \cap C_1^c) = 0.36 + 0.12 = 0.48$$

$$P^*(C_1) = 0.6,$$
$$P^*(C_2) = 0$$
$$P^*(C_3) = 0.64$$
$$P^*(C_4) = 0.48$$

The MAP order is still sub-units 2,4,3,1 but the PKI order has changed from sub-units 2,4,3,1 to sub-units 2,3,1,4.

The PKI maintenance methodology indicates that sub-unit 3 is to be checked next. Sub-unit 3 is found to be faulty. After replacement of LRU 3, the fault indicators are reset and the system appears to be working normally.

What if the process has to continue?

Now $P^{**}(C_3) = 1.0$.

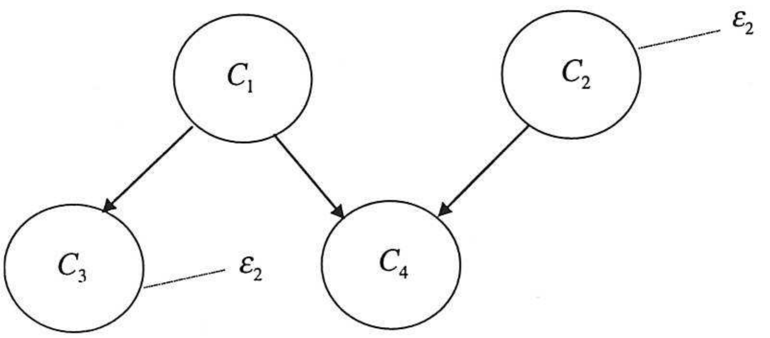


Figure 72

$$R_3^{**} = \frac{P^{**}(C_3)}{P^*(C_3)} = \frac{1.0}{0.64} = 1.5625$$

$$RC_3^{**} = \frac{P^{**}(C_3^c)}{P^*(C_3^c)} = \frac{0.0}{0.36} = 0.0$$

$$P^{**}(C_1 \cap C_3) = P^*(C_1 \cap C_3)R_3^{**} = (0.48)(1.5625) = 0.75$$
$$P^{**}(C_1 \cap C_3^c) = P^*(C_1 \cap C_3^c)RC_3^{**} = (0.16)(0.0) = 0.0$$

Marginalising:

$$P^{**}(C_1) = P^{**}(C_1 \cap C_3) + P^{**}(C_1 \cap C_3^C) = 0.75 + 0.0 = 0.75$$

$$P^{**}(C_2) = 0$$

$$R_1^{**} = \frac{P^{**}(C_1)}{P^*(C_1)} = \frac{0.75}{0.6} = 1.25$$

$$RC_1^{**} = \frac{P^{**}(C_1^c)}{P^*(C_1^c)} = \frac{0.25}{0.4} = 0.625$$

$$P^*(C_4 \cap C_1 \cap C_2) = P(C_4 \cap C_1 \cap C_2)R_1^{**} = (0.0)(1.25) = 0.0$$
$$P^*(C_4 \cap C_1 \cap C_2^c) = P(C_4 \cap C_1 \cap C_2^c)R_1^* = (0.36)(1.25) = 0.45$$
$$P^*(C_4 \cap C_1^c \cap C_2) = P(C_4 \cap C_1^c \cap C_2)RC_1^* = (0.0)(0.625) = 0.0$$
$$P^*(C_4 \cap C_1^c \cap C_2^c) = P(C_4 \cap C_1^c \cap C_2^c)RC_1^* = (0.12)(0.625) = 0.075$$
$$P^*(C_4^c \cap C_1 \cap C_2) = P(C_4^c \cap C_1 \cap C_2)R_1^* = (0.0)(1.25) = 0.0$$
$$P^*(C_4^c \cap C_1 \cap C_2^c) = P(C_4^c \cap C_1 \cap C_2^c)RC_2^* = (0.16)(1.25) = 0.2$$
$$P^*(C_4^c \cap C_1^c \cap C_2) = P(C_4^c \cap C_1^c \cap C_2)RC_1^* = (0.0)(0.625) = 0.0$$
$$P^*(C_4^c \cap C_1^c \cap C_2^c) = P(C_4^c \cap C_1^c \cap C_2^c)RC_1^* = (0.28)(0.625) = 0.175$$

$$P^{**}(C_4 \cap C_1) = P^{**}(C_4 \cap C_1 \cap C_2) + P^{**}(C_4 \cap C_1 \cap C_2^c) = 0.0 + 0.45 = 0.45$$
$$P^*(C_4 \cap C_1^c) = P^*(C_4 \cap C_1^c \cap C_2) + P^*(C_4 \cap C_1^c \cap C_2^c) = 0.0 + 0.075 = 0.075$$

$$P^{**}(C_4) = P^*(C_4 \cap C_1) + P^*(C_4 \cap C_1^c) = 0.45 + 0.075 = 0.525$$

$$P^{**}(C_1) = 0.75$$
$$P^{**}(C_2) = 0$$

$$P^{**}(C_3) = 1.0$$
$$P^{**}(C_4) = 0.525$$

The search order has now become 2,3,1,4

Note that in individual cases, MAP will outperform PKI; this is the problem of misdirection.

Overall, PKI will always be as least as good as MAP. It will be better if joint probabilities occur through causality represented by the links of the Bayesian belief network

The utility of Bayesian Belief Networks also rests upon the availability of joint information. For example, the links between nodes are governed by conditional probability expressions. If all nodes are independent in a proto-BBN, then there are no causal or informative links which would allow propagation of evidence and so no BBN would be possible.

Note that the weighting can be used on the PPUE or BBN methods; it is used only in the LRU *choice process* and not in the probability update process.

5 Conclusions

In general, condition monitoring involves many processes including FDI; FD is the detection of anomalous conditions that arise during the operation of some plant or process. FDI techniques usually end at the point of providing information about which sub-units of a given plant are suspected as being faulty. The indication of the most likely fault and its estimated probability by a fixed pattern recognition system is not necessarily the end-point. In reality, condition monitoring is or should be a closed-loop process involving an end-user who ultimately decides how to use the information generated by the condition monitoring system. The end-user may, in turn, require a mechanism of incorporating his or her observations or knowledge into the condition monitoring system for a more accurate diagnosis. The incorporation and utilisation of posterior knowledge presents a difficult problem. This research has attempted both to articulate the problem and to provide a framework for its solution.

It has been demonstrated that posterior knowledge integration, as a post-processing technique, improves, on average, fault scenario identification. It is general in that it is applicable to condition monitoring systems which provide probabilistic fault scenario data. The end-user is able to feed back information into the condition monitoring process effectively, thus closing the loop. Context-free simulations provide a clear indication that, on average, posterior knowledge integration reduces path lengths in faulty sub-unit identification. This has potential payoffs in terms of maintenance costs, both direct and indirect. The skewing effect on the quasi-histograms is dependent upon the number of non-zero scenario probabilities. Here, the posterior knowledge integration is sequential, that is, it is included after individual sub-units were inspected. There are other possible strategies including using all currently available posterior knowledge together to give modified posteriors in contrast to using it sequentially. As additional posterior knowledge becomes available, it may be integrated sequentially, as per the method described in this documentation.

The above results are preliminary but they show that posterior knowledge integration has potential use in condition monitoring. Furthermore, the 'closed-loop' method is independent of any predictive condition monitoring system. This stage follows on from the prediction of faults given a set of monitored features. The method requires a set of fault scenarios and their corresponding relative frequencies regardless of how they are estimated.

Now that the possible utility of the posterior knowledge integration technique has been demonstrated, a number of issues remain to be addressed. The probability update equation has been applied to sets of FC frequencies as specified in the simulations. These FC frequencies determine both the initial fault scenario ranking and subsequent changes. In reality, the probabilities will be estimated from condition monitoring data and, as such, will be subject to estimation errors. The combined effects of these estimation errors may alter the scenario ranking and, consequently, change the maintenance strategy. The effect of estimation errors on the update equation must therefore be investigated.

At present, the path lengths are weighted only with respect to the scenario relative frequencies and are not weighted with respect to maintenance cost. In reality, the costs may rise significantly as time goes on; in the case of aircraft, for example, long down-times can incur extra costs. The effect of cost weightings will be taken into account. Further weightings will also apply, e.g. the financial cost of replacing one LRU may be very much higher than replacing another. The simulations presented within this documentation have equally weighted scenarios. This means that the prior probabilities of fault scenarios are the same and that scenarios with many faults are as equally likely as those with fewer faults. In practice, scenarios with multiple faults are less likely. This will be represented by using various, non-uniform prior distributions for the scenario frequencies.

The simulations presented here assume that the number of faults occurring in each simulation is known *a priori*. This is to ensure that performance comparisons between posterior knowledge integration and the baseline methods can be made. In the real-world, the number of faults will be unknown. Another possible benefit of posterior knowledge integration is that the modified probabilities may indicate whether or not it is sensible to search for other possible fault FCs. The baseline method will not supply any further information as to whether or not more faults remain. With posterior knowledge integration, a probability threshold may be used, below which any further search is terminated.

Posterior knowledge is currently included sequentially following each sub-unit inspection. In practice, information about one or more sub-units may be available prior to the current stage of the fault search. A facility for 'en masse' posterior knowledge integration could be included. There is also the possibility of bringing joint probability information into the system derived from engineering knowledge and practice, i.e. *subjective probabilities.*

The properties of exclusivity and independence may be used to pre-process data before using the probability update equation. Indeed, exclusivity can be detected from the raw data. Furthermore, by identifying and simplifying dependencies, the probability estimation problem may be reduced. It is anticipated that prior engineering knowledge will also be used. It can be proved that the average path length for the PKI technique *always* lies somewhere between the optimal APL and the best baseline method APL as illustrated in Figure 73. Thus, there is the same number or fewer sub-unit inspections for the PKI method as compared to the baseline method.

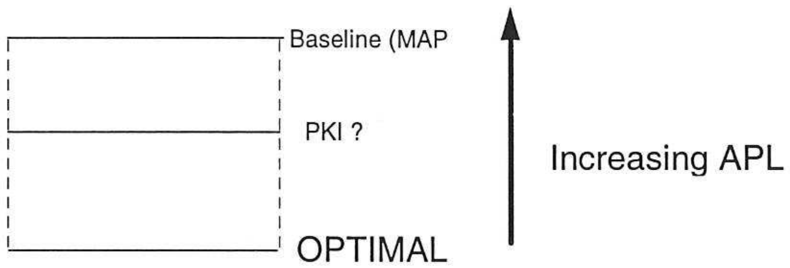


Figure 73. The PKI approach (WPKI) will always give an average path length which lies somewhere between the baseline approach and the theoretical optimum with zero residue owing to NFF incidents.

Empirical investigations illustrate that application of PKI to the FDI process reduces both the expected number of sub-unit inspections and the expected number of NFF incidents per scenario. It is likely that more rigorous, formal proofs exist (over and above those of Appendixes F and G) which show that PKI (WPKI) is at least as good as the baseline method in both the path-length and NFF cases, if not better. The percentage improvements vary depending upon a number of parameters including the number of sub-units, the amount of joint class information and the ambiguity associated with a given input reading.

5.1 Summary of Weighted Posterior Knowledge Inclusion Research

- NFF incidents increase maintenance costs.
- Including posterior knowledge in FDI shows that isolating faults during the maintenance process may be optimised compared to a baseline approach.
- PKI is as good as, if not better, than the comparison approach when joint probability information is available (this reflects causality). Both simulation and theory confirm this.

- Weightings, which reflect differing maintenance costs, may be used; this is a more realistic approach and is referred to as weighted PKI or WPKI
- The original version of WPKI relies upon estimated probabilities. However, the WPKI approach is general and may use other methods of updating probabilities.
- The work has been related to the established filed of Bayesian belief networks; Indeed, a BBN may be substituted in the WPKI process.
- The current WPKI probability update method and the BBN approach represent two extremes of an information continuum, that is, empirical vs. structural (engineering knowledge.)

5.2 *Areas of Further Work*

- Implement BBN version of WPKI.
- Investigate a hybrid approach combining empirical and structural aspects.
- Investigate independence, joint information and the relationship between the amount of joint information and the potential benefits of WPKI.
- Investigate further sources of posterior knowledge. Quantisation beyond binary is a good start.
- Investigate 'batch' methods of posterior knowledge integration so as to increase the efficiency of information usage; this is in contrast to the sequential PKI applied in this work and will allow all available knowledge to be integrated in one action followed by sequential integration as more information becomes available.
- Implement prototypes in Delphi, Visual C++ or other environment.
- Investigate condition databases and the coding of engineering information.

5.2.1 Justification for Further Work

- Implementing the BBN version will allow the use of known causal relationships and prepare the way for the hybrid approach.
- A hybrid approach (empirical/structural) will allow the use of incomplete engineering knowledge to be augmented by a refined empirical approach.
- Investigating the relationship between data dependence (causality) and the WPKI approach will indicate under what conditions the approach is likely to have real cost benefits.
- Investigating further sources of PK beyond binary inspection data will enhance the maintenance process.
- Batch methods of PKI will allow maximum use of information prior to a sequential approach.
- Prototype development will push the research towards commercial reality and help to establish real needs and working assumptions.
- Finally, investigating condition database structure and the representation of engineering knowledge will facilitate the hybrid approach; such a hybrid system will allow the initial entry of engineering knowledge (causal relations) and then use empirically derived fault information to 'fill the gaps' and give revised fault

search probabilities. Bayesian belief networks require the full specification of causal connections; a hybrid approach will remove this disadvantage.

6 Matlab Prototype Documentation for Weighted Context-free Posterior Knowledge Integration (WPKI)

6.1 General Notes

This is a self-contained weighted context-free system prototype which is used to demonstrate the utility of weighted PKI. Posterior knowledge is integrated sequentially, that is, a possible reordering of posterior class probabilities occurs on each simulated LRU inspection. The code may be modified to allow the integration of posterior knowledge en-block if it is available. The core WPKI code which would be included in a maintenance system will be indicated throughout the documentation.

This version uses the posterior probability update equation (PPUE) only; notes on a possible modified version are included in the final part of the documentation.

Extensive comments will be found throughout the code modules.

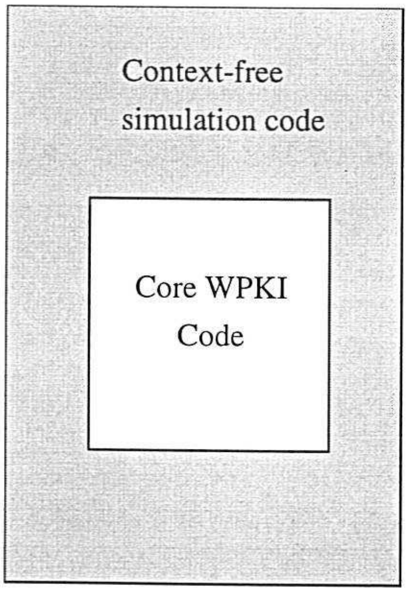


Figure 74. The Core WPKI code is the central part of the context-free simulation code.

Context-Free Algorithm:

1. Specify parameters.
2. Generate data (forward simulation) or calculate exact posteriors
3. Get data statistics.

For each reading vector

4. Do PKI/MAP
5. Calculate ensemble statistics.

6.1.1 Generate data:

In order to generate data using the forward approach (scenarios to readings), scenario priors have to be specified, thus, $P(\mathbf{S}) = \left[P(\mathbf{s}_1) \ldots P(\mathbf{s}_{2^n}) \right]$ represents the prior probabilities of those scenarios occurring.

State likelihoods also have to be specified so that states may be generated from the scenarios.

$$P(\mathbf{x}|\mathbf{S}) = \begin{bmatrix} P(\mathbf{x}_1|\mathbf{S}_1) & \ldots & P(\mathbf{x}_k|\mathbf{S}_1) & \ldots & P(\mathbf{x}_m|\mathbf{S}_1) \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ P(\mathbf{x}_1|\mathbf{S}_i) & \ldots & P(\mathbf{x}_k \mathbf{S}_i) & \ldots & P(\mathbf{x}_m|\mathbf{S}_i) \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ P(\mathbf{x}_1|\mathbf{S}_{2^n}) & \ldots & P(\mathbf{x}_k|\mathbf{S}_{2^n}) & \ldots & P(\mathbf{x}_m|\mathbf{S}_{2^n}) \end{bmatrix}$$

First, scenarios $\mathbf{S}_i$ are generated according to their priors. Then the corresponding row-vector of likelihoods is used to generate a sensor reading, $\mathbf{x}_k$.

Use Bayes' theorem to calculate the expected posteriors if data *not* going to be generated. This will give the *exact* conditional posteriors

$$P(\mathbf{S}_i|\mathbf{x}_k) = \frac{P(\mathbf{x}_k|\mathbf{S}_i)P(\mathbf{S}_i)}{P(\mathbf{x}_k)}$$

Where $P(\mathbf{x}_k) = \sum_{i=1}^{n} P(\mathbf{x}_k|\mathbf{S}_i)P(\mathbf{S}_i)$

These are used to form a 'posterior matrix' of the form

$$P(\mathbf{S}|\mathbf{x}) = \begin{bmatrix} P(\mathbf{S}_1|\mathbf{x}_1) & \dots & P(\mathbf{S}_i|\mathbf{x}_1) & \dots & P(\mathbf{S}_{2^n}|\mathbf{x}_1) \\ \vdots & \dots & \vdots & \dots & \vdots \\ P(\mathbf{S}_1|\mathbf{x}_k) & \dots & P(\mathbf{S}_i|\mathbf{x}_k) & \dots & P(\mathbf{S}_{2^n}|\mathbf{x}_k) \\ \vdots & \dots & \vdots & \dots & \vdots \\ P(\mathbf{S}_1|\mathbf{x}_m) & \dots & P(\mathbf{S}_i|\mathbf{x}_m) & \dots & P(\mathbf{S}_{2^n}|\mathbf{x}_m) \end{bmatrix}.$$

Where data is generated using a forward simulation, the data is generated from which the empirically derived posterior matrix, $\tilde{P}(\mathbf{S}|\mathbf{x})$ is computed. Compare $P(\mathbf{x}|\mathbf{S})$ with $\tilde{P}(\mathbf{S}|\mathbf{x})$ to check the simulation. PKI is then carried out using the estimated posterior matrix data.

6.1.2 Context-Free Performance indicators for both Path lengths and NFF inspections.

Protocol:

Take *each* state vector in-turn. There is a row vector of posterior probabilities, $P(\mathbf{S}|\mathbf{x}_k)$ associated with each possible input reading.

Where,

$$P(\mathbf{S}|\mathbf{x}_k) = \left[P(\mathbf{S}_1|\mathbf{x}_k) \quad \dots \quad P(\mathbf{S}_i|\mathbf{x}_k) \quad \dots \quad P(\mathbf{S}_{2^n}|\mathbf{x}_k)\right]$$

Now, *each* scenario is taken in-turn and treated as being the *actual* fault scenario (for a given state vector; both PKI and BL methods are used to identify the actual scenario.

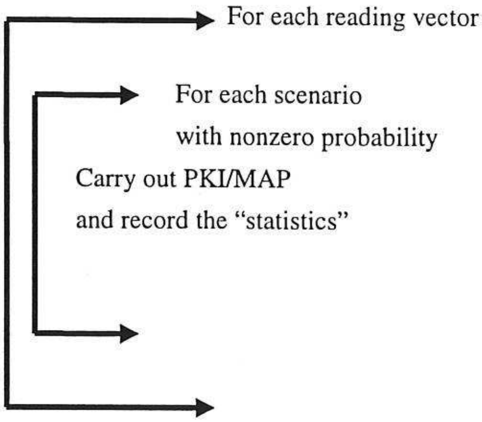


Figure 75. When the simulation data is specified, the context-free WPKI process is carried out by finding the APL for each state vector and weighting the APL by the state vector prior.

6.1.3 Detail of PKI process:

The LRUs are weighted using an integer cost weighting. The cost weighting reflects the relative cost of inspecting each LRU.

The LRU to be inspected is chosen on the basis of the posterior probabilities conditional upon the readings. The *exclusive* conditional scenario posterior probabilities,

$$P(\mathbf{S}|\mathbf{x}_k) = \left[P(\mathbf{S}_1|\mathbf{x}_k) \quad \cdots \quad P(\mathbf{S}_i|\mathbf{x}_k) \quad \cdots \quad P(\mathbf{S}_{2^n}|\mathbf{x}_k)\right]$$

are used to construct the posterior probabilities, $P(C_i|\mathbf{x}_k)$ by summing the exclusive scenario probabilities of scenarios having the ith LRU faulty.

The complement of the posteriors are calculated and then weighted before the LRU is chosen with the lowest weighted probability of not being faulty. This is equivalent to choosing the LRU with the highest posterior probability of being faulty in the unweighted case.

6.2 Context-free Simulation Code

6.2.1 Module: wpkibat.m

Top-level batch file for context-free simulations.
Calls `wtchain` and `wpkicha` routines.

Wtchain carries out the data simulation process which is used to illustrate the WPKI technique. It *does not* form part of the core WPKI code.

wpkicha carries out the WPKI process for the data and *contains* the core WPKI code embedded in a WPKI/weighted baseline comparison routine.

6.2.2 Module: wtchain.m

Second level module for generating probabilities involved in evaluating WPKI.

Calls `wtdparam` which sets up the simulation parameters.
Calls `fwdsim` which generates data
Calls getstats which gathers the statistical information on the data
Calls `bayes` which computes posterior probabilities from likelihoods
Calls `ambig` which removes scenario probabilities to reduce ambiguity

6.2.3 Module wtdparam.m

6.2.4 Module fwdsim.m

Uses forward simulation to generate data unless the theoretical values are used; in this case the actual data generation is bypassed.

Calls `genlike`
Calls `biasprior`
Calls `genlike`
Calls `genfwd`

First $P(\mathbf{S})$ is used to generate the fault scenarios. Next the likelihoods, $P(\mathbf{x}|\mathbf{S})$ are used to generate state vectors.

6.2.5 Module: genlike.m

Generates the likelihood matrix, $P(\mathbf{x}|\mathbf{S})$

6.2.6 Module: `genfwd.m`

Calls `int2bins`
Calls `vecsel`

Generates a file of simulated binary fault data of the form $(\mathbf{x}_k, \mathbf{S}_i)$ that is, an input (state) vector, points to an output (scenario) vector. This corresponds to readings gathered together with *verified* fault scenarios.

6.2.7 Module: `getstats`

Calls `bin2int`
Calls `neprecon`

This module reads in the data file, constructs a matrix of exclusive scenario relative frequencies and reconstructs the non-exclusive versions. Although the non-exclusive probabilities are not used here, they may be useful.

6.2.8 Module: `bayes.m`

Takes the likelihood matrix, $P(\mathbf{x}|\mathbf{S})$
and the scenario prior vector, $P(\mathbf{S})$
and uses Bayes' theorem to give the scenario posterior matrix $P(\mathbf{S}|\mathbf{x})$

This routine is used where the exact posteriors are computed directly from the likelihood matrix and scenario posterior matrix

6.2.9 Module: `ambig.m`

This routine removes scenario posterior probabilities associated with a given input to reduce the ambiguity of scenario choice.

6.2.10 Module: `wpkicha.m`

Second level module for evaluating WPKI given the probabilities generated or calculated previously. The WPKI cycle is carried out for each input vector

Calls `cfsmat`
Calls `cstpcmat`
Calls `inistats`
Calls `wtdcode`
Calls `pkstats`
Calls `blstats`

This is module contains core WPKI code which can be embedded in a maintenance system. The relevant code will be highlighted and discussed further in the maintenance implementation section.

One of the functions of wpkicha.m is to gather "statistics" on the performance.

Note that the conditional posterior probabilities are *exclusive*, that is, a m from n problem has been decomposed into a 1 from 2^n problem. The non-exclusive probabilities are reconstructed from the exclusive ones.

Calls `cfsmat` which creates a full fault scenario matrix for identification using the posterior probabilities. For example, the set of scenarios for three LRUs is

0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1
0	0	0

Calls `cstpcmat` which sets up a matrix containing a vector of scenario indices for each posterior probability. This enables PKI to exclude or include given scenarios depending upon the current LRU inspection. For example, if a LRU is found to be faulty (LRU bit on), then all scenarios with this LRU operating correctly are excluded.

For example, for three LRUs

4	5	6	7
2	3	6	7
1	3	5	7

where LRU one is faulty in scenarios 4,5,6 and 7 and so on. In this case, if LRU one is found not to be faulty, then scenarios 1,2,3, and 8 are the remaining possibilities. By calculating which scenarios include/exclude a given LRU, the new conditional posterior probabilities may be calculated. For example, if LRU 2 is found to be faulty, scenarios 2,3,6 and 7 are retained only.

Calls `inistats` which initialises the variables used for collecting information on the two techniques to be compared.

6.2.11 Module: `wtdcode.m`

Calls `getnff`
Calls `winsplru`

This module carries out the WPKI routines.
For a *particular set of parameters*, the exclusive scenario posteriors conditioned upon the input, $P(\mathbf{S}|\mathbf{x})$, are

scenario_posteriors =

0.0570	0.1466	0	0.0674	0	0.2594	0.1599	0.3097
0.1465	0.0254	0	0.1520	0	0.1450	0.1124	0.4186
0.2465	0	0.0866	0	0.0209	0.0878	0.0434	0.5148
0	0.0846	0.0316	0.0934	0.0789	0	0.0825	0.6289
0	0.1476	0.0420	0.1305	0.2814	0	0.0603	0.3382
0.2909	0.1822	0.2294	0.0453	0	0.1176	0.1346	0
0.0142	0.1114	0.2518	0	0.0898	0	0.1438	0.3888
0	0.2870	0.0086	0	0.1457	0.0009	0.3831	0.1748

The scenario frequency data for the

freqdat =

0.0570	0.1466	0	0.0674	0	0.2594	0.1599	0.3097

Here, the LRU fault probabilities (posteriors) conditioned upon state vector 1 are shown.
The `wtdcode` module will go through each scenario which has a non-zero conditional probability i.e. $P(\mathbf{S}_i|\mathbf{x}_k) \neq 0$ and attempt to identify that scenario using PKI.

The routine `getnff` establishes the potential no fault found incidents from the state vectors and the scenario vectors.

While there are LRU inspections, the WPKI process is carried out for a single plant scenario to be identified.

If there are LRUs to be checked, `winsplru` is called.

6.2.12 Module: winsplru.m

Calls `nofltlru`
Calls `postlru`

This module is core WPKI code which can be used to inspect single LRUs and update the posterior probabilities according to the posterior probability update equation (PPUE). Winsplru.m will be called repeatedly until the current plant scenario is identified or all LRUs have been inspected.

It requires the *exclusive* conditional scenario posterior probabilities,

$$P(\mathbf{S}|\mathbf{x}_k) = \left[P(\mathbf{S}_1|\mathbf{x}_k) \quad \cdots \quad P(\mathbf{S}_i|\mathbf{x}_k) \quad \cdots \quad P(\mathbf{S}_{2^n}|\mathbf{x}_k)\right]$$

as an input. The posterior probabilities of individual LRUs being faulty will be reconstructed from these by the PPUE.

To calculate the conditional posterior probabilities, $P(C_i|\mathbf{x}_k)$ the exclusive scenario probabilities of scenarios having the ith LRU faulty are summed.

Initially, when there is no posterior knowledge present, no scenarios are excluded and the conditional posterior probabilities are those of the preceding FDI process.

The posterior probabilities of the LRUs are calculated using the exclusive scenario probabilities. In this case posteriors =

0.4867 0.5659 0.2169 0.3097

Together with the weighting vector of [3 2 1 1] the weighted posterior complement is

wtd_post_comp =

1.5400 0.8682 0.7831 0.6903

The lowest weighted complement posterior probability (lowest weighted probability of no-fault) is used to indicate the LRU to be inspected out of the remaining (uninspected) LRUs.

```
unitindex =

     4
```

The chosen LRU is the dummy LRU (4) meaning that the system is likely to be running correctly (with these weightings!). It is less costly to check the sensors in this case.

The routine `nofltlru` is now called to remove the no fault LRU from the competition. The module `winsplru` is called again.

posteriors =

0.7050 0.8198 0.3143 0

wtd_post_comp =

0.8849 0.3604 0.6857 1.0000

unitindex =

2

The module `postlru` is then called

faultcode(unitindex)

ans =

0

newfaultprobs =

Columns 1 through 7

0.4583 0 0 0.5417 0 0 0

Column 8

0

Now only scenarios [001] and [100] are suspected.
All faults are not found (scenario is not identified) so the cycle continues with `winsplru` again.

posteriors =

0.5417 0 0.4583 0

wtd_post_comp =

1.3750 2.0000 0.5417 1.0000

unitindex =

3

The module `postlru` is then called again

faultcode(unitindex)

ans =

1

newfaultprobs =

1 0 0 0 0 0 0 0

`lruinspect=0`

All faults detected (unit 3) in scenario [0 0 1].

The weighted path length is 3.

6.2.13 Module: postlru.m

6.2.14 Module: nofltlru.m

This removes the no-fault scenario probability and renormalises the remaining probabilities.

6.3 Scaleability and Hash Coding.

One of the potential problems of the current system may appear to be the scaleability. For example, for K input vectors and N LRUs, the theoretical number of probabilities to be estimated to give the 'posterior matrix'

$$P(\mathbf{S}|\mathbf{x}) = \begin{bmatrix} P(\mathbf{S}_1|\mathbf{x}_1) & \ldots & P(\mathbf{S}_i|\mathbf{x}_1) & \ldots & P(\mathbf{S}_{2^n}|\mathbf{x}_1) \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ P(\mathbf{S}_1|\mathbf{x}_k) & \ldots & P(\mathbf{S}_i|\mathbf{x}_k) & \ldots & P(\mathbf{S}_{2^n}|\mathbf{x}_k) \\ \vdots & \ldots & \vdots & \ldots & \vdots \\ P(\mathbf{S}_1|\mathbf{x}_m) & \ldots & P(\mathbf{S}_i|\mathbf{x}_m) & \ldots & P(\mathbf{S}_{2^n}|\mathbf{x}_m) \end{bmatrix}$$

is given by $K2^N$. Furthermore, other probability vectors will be of the size 2^N. Thus, the storage requirements appear to be high even for a modest number of LRUs.

However, it is extremely unlikely that this will be the case because,

- not all state vectors will occur, and
- very few of the possible scenarios are likely for a given state vector.

The last point is guaranteed by FDI design because, if there were multiple scenarios for each state vector then the FDI process would be ambiguous and, thus, uninformative; the ideal situation is where one state vector points to a single fault. In reality, a state vector may indicate a small number of faults and requires further information to disambiguate the FDI situation.

A modification to increase scaleability of the system for commercial use is to use a hash table which codes for any of the K, 2^N or $K2^N$ quantities which are present. For example, consider a twenty LRU FDI situation which, for a given input, only has 4 scenarios, say,

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1

A scenario matrix row (for a given input vector) will contain,

[11 27 15 1]

which codes for the given scenarios. Another matrix will include a row giving the scenario probabilities. Space may be allocated as required.

6.4 Maintenance System Toolbox Implementation Guidance (PPUE version only)

Stand-alone batch file, standbat.m illustrates the active WPKI code and what is required as an interface in a maintenance system. Example: three sensor state vector, three sub-unit system

The current state vector is
`inputcode=[0 1 1]`

indicating that sub-units 2 and 3 are faulty.

In reality,

`inspect_info=[0 0 1];`

is the real situation with sub-unit 1 faulty only.

The exclusive fault probabilities are given by

newfaultprobs =

Columns 1 through 7

0.2465 0 0.0866 0 0.0209 0.0878 0.0434

Column 8

0.5148

For example, sub-unit three [0 0 1] has a probability of being faulty (exclusively) of 0.2465

The conditional posterior probabilities are then calculated as

posteriors =

0.1521 0.2178 0.3974 0.5148

For example, the *total* posterior probability of unit three being faulty is 0.3974

this comes from scenarios [0 0 1], [0 1 1], [1 0 1] and [1 1 1] which all involve sub-unit 3.

The complement is then taken, giving

post_comp =

0.8479 0.7822 0.6026 0.4852

This is because for weighted PKI, the weighted probability of no-fault is minimised which is equivalent to an unweighted maximisation of finding a fault.

When the weighting vector [3 2 1 1] is applied, the weighted posterior complement becomes
wtd_post_comp =

2.5437 1.5645 0.6026 0.4852

The lowest weighted posterior complement belongs to the *no-fault* case

unitindex =

4

This means that it is more cost-effective to check the relevant sensors because of the weighting and the higher probability of there being no fault (correct operation).

The hypothesis that there is no-fault in the system (state [0 0 0]) is then flagged so it cannot be chosen again:

lruschecked =

0 0 0 1

In this case, there is still a fault to be detected. Here, this is known beforehand for the purposes of simulation; in reality, a decision will be taken whether or not to proceed further depending upon engineering knowledge and the *current* posterior probabilities indicated by WPKI.

In corenof.m

```
numfault>0
```

indicating that the process continues until the scenario (plant state) is identified. This decision would be taken by the user in reality.

Because the maintenance process is continuing, the no-fault state probability is disregarded giving

newfaultprobs =

Columns 1 through 7

0.5081 0 0.1784 0 0.0431 0.1809 0.0895

Column 8

0

Note that these new scenario probabilities are normalised because they are exclusive. The no-fault choice dies not necessitate an LRU inspection therefore both

`wtd_pkdepth` and `wtd_NFF_inspection_count` are not incremented.

The new posterior probabilities are given by

posteriors =

0.3135 0.4488 0.8191 0

The complement is then taken, giving

post_comp =

0.6865 0.5512 0.1809 1.0000

When the weighting vector [3 2 1 1] is applied, the weighted posterior complement becomes

wtd_post_comp =

2.0596 1.1023 0.1809 1.0000

The lowest weighted posterior complement belongs to the cases where LRU 3 is faulty

unitindex =

3

This means that LRU 3 is least likely to be non-faulty according to the cost-weighted probabilities. Conversely, LRU 1 is most likely to be non-faulty and has a higher inspection cost.

So, LRU 3 is inspected

Now,

```
wtd_pkdepth =

     1
```

indicating the cost of inspection.

lruschecked =

    0   0   1   1

LRU 3 is to be inspected.

The scenario indicator vector

indicator =

    1   0   1   0   1   0   1   0

indicates that scenarios 1, 3 5, and 7 are involved ([0 0 1], [0 1 1], [1 0 1] and [1 1 1]) because they all involve LRU 3 being faulty.

If LRU 3 is inspected and found to be non-faulty, these four scenarios will be excluded as possible scenarios for the current state. If LRU 3 is found to be faulty, the current scenario (system state) will be among this group.

By including or excluding scenarios according to the state of the currently investigated LRU, the PPUE is being implemented.

{PPUE example given the current probabilities}

The new exclusive fault probabilities are

newfaultprobs =

Columns 1 through 7

0.6203 0 0.2179 0 0.0526 0 0.1092

Column 8

0

Note that these have been renormalised. A stopping criteria might be to keep a record of the original remaining probabilities and check if they drop below a given threshold. The weighting of the current sub-unit (3) is one so the current weighted inspection cost is given by the accumulator

wtd_pkdepth =

1

Now,

```
(sumffound==numfault)
```

is true because the number of faults found is equal to the number present and so `lruinspect=0;`
that is, there are no more inspections. This stopping criterion is used for simulation purposes where it is known in advance what is the actual fault scenario to be identified.
Other criteria (probability threshold or no-fault re-test) may be used. The user may set `lruinspect=0` when he or she is satisfied with the maintenance outcome.

There have been no inspections resulting in a fault not being found so,

```
wtd_NFF_inspection_count =

     0
```

The next example is for the same input code of [0 1 1] and a new target scenario to be identified given by

```
inspect_info=[0 1 0];
```

The exclusive scenario conditional probabilities given the same input are given again by
newfaultprobs =

Columns 1 through 7

0.2465 0 0.0866 0 0.0209 0.0878 0.0434

Column 8

0.5148

As before,

posteriors =

0.1521 0.2178 0.3974 0.5148

post_comp =

0.8479 0.7822 0.6026 0.4852

wtd_post_comp =

2.5437 1.5645 0.6026 0.4852

unitindex =

4

lruschecked =

0 0 0 1

and so on until unit 3 is chosen again with

wtd_post_comp =

2.0596 1.1023 0.1809 1.0000

as before.

This LRU is inspected and so the cost accumulator

wtd_pkdepth =

1

Again, the routine `postwpki.m` is entered to integrate the posterior knowledge obtained from inspecting sub-unit 3.

This time, sub-unit 3 is non-faulty and all scenarios which have sub-unit 3 faulty must be excluded from further probability operations. The indicator for scenarios involving sub-unit 3 is again given by indicator =

1 0 1 0 1 0 1 0

The scenarios are to be excluded and so the complement is used

complement =

0 1 0 1 0 1 0 1

From the previous exclusion of the no-fault (normal operation case):

lastindicator =

1 1 1 1 1 1 1 0

Combining these two vectors gives a new indicator of,

lastindicator =

0 1 0 1 0 1 0 0

which includes only three possible scenarios. Scenario 2 is amongst them.

Now, the normalised new fault probabilities are

newfaultprobs =

0 0 0 0 0 1 0 0

because scenario 2 had never occurred in conjunction with this state vector before. In other words, the FDI system has never had this set of readings for this scenario prior to this time. Scenario 4 [1 0 0] has also not given rise to a state vector of [0 1 1] before. The inspection of sub-unit 3 yielded no fault, thus there is a no fault found (NFF) incident to be recorded,

```
wtd_NFF_inspection_count =

     1
```

The weighting of sub-unit 3 is one and so the NFF inspection is unweighted.

The only remaining scenario is scenario 6 [1 1 0] where sub-units 1 and 2 are faulty together.

The new posterior probabilities

posteriors =

1 1 0 0

indicate that both sub-units are suspected as being faulty on the information gathered so far. If sub-unit 1 is checked first,—giving a NFF condition—the posterior probabilities will drop to zero but the fault will not go away. This means that sub-unit 2 has to be inspected and the scenario will be identified. This time, the maintenance cost is high because all sub-units have been inspected; the new association between state vector [0 1 1] and fault scenario [0 1 0] will be included in the FDI statistics.

Specify a new model or stored model (Interface code).
New model (starting from scratch):

Specify:

1. model name/file,
2. number of reading sensors,
3. number of sub-units,
4. LRU cost weightings

6.5 Bayesian Belief Network Code Modifications

A BBN may be used in place of the PPUE. The BBN can be represented by a link matrix and a set of conditional probability tables. It may be possible to call pre-compiled BBN routines which are available.

6.6 Further Modifications

The current implementation identifies scenarios that are specified by the system in order to test the theory and assess the utility of PKI; in reality, this information is not available and a set of 'stopping criteria' should be devised. One approach is to use a posterior probability fault threshold. This could be combined with state vector monitoring. This combination would be used to analyse the likelihood of faults. For example, if the readings are indicating that all faults have been found, the revised posterior probabilities would either tend to confirm or deny this.

7 Further Work

Knowledge of sub-system causality will vary from being totally complete to totally unknown. In-between these extremes, it will be incomplete to varying degrees.

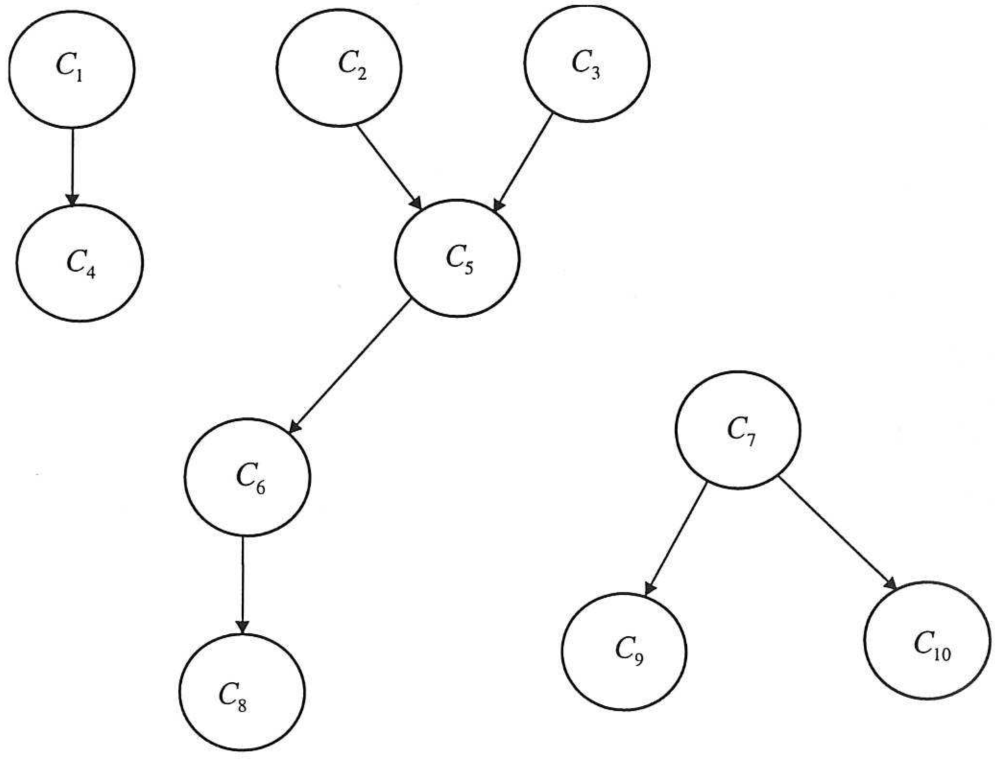


Figure 76.

The first task will be to build an incomplete system model from existing engineering knowledge (prior) in the form of a digraph or set of digraphs with quantified links. The set would consist of a set of BBN sub-nets. Sub-units upstream—in a causal sense—are the parents of nodes downstream. Known causal relationships can feed into the PPUE process. The model building must be incremental so that new prior knowledge may be added without disrupting the system. Research is required to ascertain the relationship between the empirical PPUE-based method and a *partial* BBN. In reality, information is usually incomplete which reduces the range of BBN applicability; a hybrid empirical-structural WPKI method would counter this.

8 References

Applebaum, D (1996) *Probability and Information: An Integrated Approach* CUP Cambridge

Bernardo, J. M. and Smith, A. F. M. (1994) *Bayesian Theory* John Wiley and Sons Ltd, Chichester.

Bilchev, G.; Parmee, I.(1996) Constraint handling for the fault coverage code generation problem: an inductive evolutionary approach, *Proc. International*

Conference on Evolutionary Computation -The 4th International Conference on Parallel Problem Solving from Nature. p. 880-9

Bishop, C. M. (1991) Improving the Generalisation Properties of Radial Basis Function Neural Networks, *Neural Computation* **3**, 4, 579-588

Bishop, C. M. (1993) Curvature-Driven Smoothing: A Learning Algorithm For Feedforward Networks *IEEE Transactions on Neural Networks* **4**(5) 882-884

Bishop, C. M.(1995) *Neural networks for Pattern Recognition* Oxford University Press Oxford.

Bogunovic, N. Mesic, T. (1996) Adaptive uncertainty management for a class of diagnostic expert systems. *Artificial Intelligence for Engineering Design, Analysis and Manufacturing* 10,5, 421-9

Boudoud, A. N. and Masson, M. H. (1996) The diagnosis of a technological system: on-line fuzzy clustering using a gradual confirmation of prototypes. *Proc. CESA '96 IMACS Multiconference. Computational Engineering in Systems Applications.*Vol. 1. 110-115

Broomhead, D. S. and Lowe, D. (1988) Multivariable Function Interpolation and Adaptive Networks, *Complex Systems* **2** 321-355

Dimla, D.E, Lister, P. M. and Leighton, N. J. (1997) Tool condition monitoring in metal cutting through application of MLP neural networks *IEE Colloquium on Fault Diagnosis in Process Systems (Digest No. 1997/174)* 9/1-3

Ding, Y.; Wach, D. (1994) A rule- and case-based hybrid system for rotating machinery diagnosis, *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes* - SAFEPROCESS '94. Preprints vol.2. 475-8

Dorr, R.; Kratz, F.; Ragot, J.; Loisy, F.; Germain, J.-L. (1997) *IEEE Transactions on Control Systems Technology* Vol: 5 Iss: 1 p. 42-60

Duda and Hart (1973) *Pattern Classification and Scene Analysis*

Durrett, R. (1994). *The Essentials of Probability* The Duxbury Press. Belmont California.

Eryurek, E.; Upadhyaya, B.R. (1995) An integrated fault-tolerant control and diagnostics system for nuclear power plants, *Proceedings of the Topical Meeting on Computer-Based Human Support Systems: Technology, Methods, and Future*. 267-74

Haykin, S. (1994) *Neural networks a Comprehensive Foundation* Macmillan

Halmos, P. R.(1974) *Naive Set Theory* (Springer-Verlag)

Hines, J.W.; Miller, D.W.; Hajek, B.K.(1995) Fault detection and isolation: a hybrid approach *Proceedings of the Topical Meeting on Computer-Based Human Support Systems: Technology, Methods, and Future*, 363-70

Gomm, J.B. (1994) Fault detection in a multivariable chemical process by monitoring process dynamics, *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes - SAFEPROCESS '94*. Preprints. Vol. 1, 177-82

Grimmet, G. R. and Stirzaker, D. R. (1992) *Probability and Random Processes*, Oxford Science Publications, Oxford.

Isermann, R. (1997). Supervision, Fault-Detection and Fault-Diagnosis Methods-an Introduction, *Control Engineering Practice*: **5** pp639-651

Isermann, R. (1994) Integration of fault detection and diagnosis methods. *Proc. IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes - SAFEPROCESS '94. Preprints* . 597-612 vol.2

Jensen, F. V. (1996) *An Introduction to Bayesian Networks.* UCL Press Limited London

Karsai, G.; DeCaria, F. (1997) Model-integrated on-line problem-solving in chemical engineering, *Control Engineering Practice*, Vol: 5 Iss: 1 1-9

Keravnou, E. T. and Johnson, L. (1986) *Competent Expert Systems: A Case Study in Fault Diagnosis* McGraw Hill NY

Krause, P. and Clarke, P. (1993) *Representing Uncertain Knowledge: An Artificial Intelligence Approach.* Intellect Books Oxford

Kneale, W. (1949) *Probability and Induction* Oxford at the Clarendon Press, Oxford

Korbicz, J and Kus, J (1996) Structural-parametrical identification method in fault detection and diagnosis systems. *Proc. CESA '96 IMACS Multiconference. Computational Engineering in Systems Applications*.Vol. 1. 696-700

Korbicz, J.; Kus, J. (1995) Knowledge-based fault detection system using genetic observer approach, *Proceedings of the 12th International Conference on Systems Science* 262-70 vol.3

Krishnaswami, V.; Rizzoni, G. (1994) A survey of observer based residual generation for FDI, *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes - SAFEPROCESS '94. Preprints*. 34-9 vol.1

Li, X.Q.; Wong, Y.S.; Nee, A.Y.C. (1996) Tool condition monitoring using acoustic emission sensing and an integrated multi-ART 2 neural network. *Proc. Advanced Manufacturing Processes, Systems, and Technologies (AMPST 96)* 193-200

Lianhui Chen; Ho, E. (1994) Improving fault diagnosis performance with contextual knowledge, Proc. *The Third International Conference on Automation, Robotics and Computer Vision.* vol.2,1338-42

Liu, T.I.; Singonahalli, J.H.; Iyer, N.R. (1996) Detection of roller bearing defects using expert system and fuzzy logic *Mechanical Systems and Signal Processing* Vol: 10 Iss: 5 p. 595-614

Logan, D.; Mathew, J. (1996) Using the correlation dimension for vibration fault diagnosis of rolling element bearings .I. *Basic concepts. Mechanical Systems and Signal Processing* Vol: 10 Iss: 3 p. 241-50

Ma Yizhong (1996), Diagnosis for signals in multiple correlated processes *Computers & Industrial Engineering* Vol: 31 Iss: 3-4 p. 817-20

McDonald, J. R., Burt, G. and Moyes, A. (1996) Knowledge based systems for condition monitoring, *Proc. UKACC International Conference on Control '96*, vol.2, 1424-9

MacIntyre, J.; O'Brien, J.C. (1995) Investigations into the use of wavelet transformations as input into neural networks for condition monitoring, *Artificial Neural Nets and Genetic Algorithms. Proceedings of the International Conference.* 116-19

Marriott, S, and Harrison, R. F. (1997). *The use of posterior knowledge in statistical pattern recognition with particular application to fault diagnosis.* Research Report No. 676 May 1997 The University of Sheffield, U.K.

Marriott, S, and Harrison, R. F. (1998a). *The integration of posterior knowledge into statistical pattern recognition systems with particular application to fault diagnosis, Proceedings of EIS'98* Proceedings Volume No. 3 (Artificial Intelligence) pp397-403

Marriott, S, and Harrison, R. F. (1998b) A Novel Approach to the Integration of Posterior Knowledge into Condition Monitoring Systems: Theory and Practice? *Proceedings of Control '98* UKACC Swansea, UK, Volume No. 2, pp. 1492-1497

Marriott, S, and Harrison, R. F. (1999) The Reduction of No Fault Found Incidents Through the Sequential Optimisation of Maintenance Strategies using Posterior Knowledge Integration. Proceedings of the 12th International Conference on control Systems and Computer Science (CSCS12) Bucharest, Vol II, p53-57

Melsa, J. L. and Cohn, D. L. (1978) *Decision and Estimation Theory.* McGraw-Hill

Milne, R.; Nicol, C.; Trave-Massuyes, L.; Quevedo, J. (1996) TIGER: knowledge based gas turbine condition monitoring, *AI Communications* Vol: 9 Iss: 3 p. 92-108

Moody, J. and Darken, C. J. (1989) Fast Learning in Networks of Locally-Tuned Processing Units, *Neural computation* **1** (2) 281-294

Nairac, A, Corbett-Clark, T. A., Ripley, R., Townsend, N. W., and Tarassenko, L (1997), Choosing an appropriate model for novelty detection. *Proceedings of the 1997 5th International Conference on Neural Networks*, Cambridge UK pp

Pan, M. C., Sas, P. and van Brussel, H. (1996) Non-stationary time-frequency analysis for machine condition monioring. *Proc. IEEE-SP International Symposium on Time-Frequency and Time-Scale Analysis*. 477-480.

Patel, V.C.; Kadirkamanathan, V.; Kulikov, G.G.; Arkov,V.Y.; Breikin, T.V. (1996) Gas turbine engine condition monitoring using statistical and neural network methods, *Proc. IEE Colloquium on Modelling and Signal Processing for Fault Diagnosis* (Ref. No.1996/260), 1/1-6

Patel, V. C., Kadirkamanathan, V., Thompson, H. A. and Flemming, P. J.(1996) Development of a gas turbine engine model for fault diagnosis. *Proc. IASTED International Conference. Artificial Intelligence, Expert Systems and Neural Networks*. 379-382.

Patton, R. J. and Chen, J. (1997) Observer-Based Fault Detection and Isolation: Robustness and Applications, *Control Engineering Practice*, **5** (5) 671-682

Patton, R., J., Frank, P. M. and Clark, R. N. (eds.) (1989) *Fault Diagnosis in Dynamic Systems, Theory and Application* Control Engineering Series. Prentice hall, London

Perrott, S.N.; Perryman, R.(1995) Adaptive resonance theory applied to condition monitoring of combined heat and power systems, *UPEC '95. 30th Universities Power Engineering Conference 1995. Conference Proceedings*. Vol.1 45-8

Preston, G. J. Shields, D. N. and Daley, S. (1996) Application of a robust nonlinear fault detection observer to an hydraulic system. *Proc. UKACC International Conference on Control '96*. vol. 2 1484-9.

Richard, M. D. and Lippmann, R. P. (1991) Neural Network Classifiers Estimate Bayesian *a posteriori* Probabilities. *Neural Computation*, **3** 461-483

Ruokonen, T (ed.) *IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes - SAFEPROCESS '94.*

Rodd, M. G. (ed. in chief) (1997) *Control Engineering Practice*: Special section on fault detection and diagnosis. **5** (5)

Anon. , (1986) "The Jet Engine", Rolls-Royce plc.

Tarassenko, L (1996) *Novelty Detection* Neural Computing Applications Forum digest supplement Edinburgh

Tarassenko, L (1997) *Novelty Detection*: *Key questions*, Neural Computing Applications Forum digest supplement, Bath

Trave-Massuyes, L and Milne, R (1997) Gas-turbine condition monitoring using qualitative model-based diagnosis *IEEE Expert* 12,3 22-31

Walpole, R. E. and Myers, R. H. (1989) *Probability and Statistics for Engineers and Scientists* Macmillan Publishing Company New York

Wang Xue; Yang Shuzi (1996) A parallel distributed knowledge-based system for turbine generator fault diagnosis, *Artificial Intelligence in Engineering* Vol: 10 Iss: 4 p. 335-41

Wang, X.Z.; Lu, M.L.; McGreavy, C. (1997) Learning dynamic fault models based on a fuzzy set covering method, *Computers & Chemical Engineering* Vol: 21 Iss: 6 p. 621-30

Wasserman, P. D. (1993) *Advanced Methods in Neural Computing* VNR New York

Weighell, M. Martin, E. B. and Morris, A. J. (1997) Fault diagnosis in industrial process manafacturing using MSPC. *IEE Colloquium on Fault Diagnosis in Process Systems* (Digest No. 1997/174) 4/1-3

Wilson, R. J. and Watkins, J. J. (1990) *Graphs, An Introductory Approach.* John Wiley and Sons.

Wilson, D.J.H, Irwin, G. W. and Lightbody, G. (1997) Neural networks and multivariate SPC *IEE Colloquium on Fault Diagnosis in Process Systems* (Digest No. 1997/174) 5/1-5

Yang, H.; Saif, M. (1996) Monitoring and diagnostics of a class of nonlinear systems using a nonlinear unknown input observer. *Proceedings of the 1996 IEEE International Conference on Control Applications.* 1006-11

Siyu Zhang; Ganesan, R.; Sankar, T.S. (1995) Self-organizing neural networks for automated machinery monitoring systems *Computers in Engineering*, p. 1001-9

Zhang, Q (1996) Using non-linear black-box models in fault detection. *Proc. 35th IEEE conference on Decision and Control.* Vol. 1 636-7