Application of Bayesian Belief Networks to System Fault Diagnostics

by
Mariapia Lampis

A Doctoral Thesis Submitted in partial fulfilment of the requirements for the award of the Doctor of Philosophy of Loughborough University

April 2010

Abstract

Fault diagnostic methods aim to recognize when a fault exists on a system and to identify the failures which have caused it. The fault symptoms are obtained from readings of sensors located on the system. When the observed readings do not match those expected then a fault can exist. Using the detailed information provided by the sensors a list of the failures that are potential causes of the symptoms can be deduced. In the last decades, fault diagnostics has received growing attention due to the complexity of modern systems and the consequent need of more sophisticated techniques to identify failures when they occur. Detecting the causes of a fault quickly and efficiently means reducing the costs associated with the system unavailability and, in certain cases, avoiding the risks of unsafe operating conditions.

Bayesian Belief Networks (BBNs) are probabilistic graphical models that were developed for artificial intelligence applications but are now applied in many fields. They are ideal for modelling the causal relations between faults and symptoms used in fault diagnostic processes. The probabilities of events within the BBN can be updated following observations (evidence) about the system state.

In this thesis it is investigated how BBNs can be applied to the diagnosis of faults on a system with a model-based approach. Initially Fault Trees (FTs) are constructed to indicate how the component failures can combine to cause unexpected deviations in the variables monitored by the sensors. The FTs are then converted into BBNs and these are combined in one network that represents the system. The posterior probabilities of the component failures give a measure of which components have caused the symptoms observed. The technique is able to handle dynamics in the system introducing dynamic patterns for the sensor readings in the logic structure of the BBNs.

The method is applied to two systems: a simple water tank system and a more complex fuel rig system. The results from the two applications are validated using two simulation codes in C++ by which the system faulty states are obtained together with the failures that cause them . The accuracy of the BBN results is evaluated by comparing the actual causes found with the simulation with the potential causes obtained with the diagnostic method.

Key Words: System Fault Diagnostics, Model-based Diagnosis, Bayesian Belief Networks, Fault Tree analysis.

Acknowledgements

I would like to express my sincere gratitude to my supervisor Prof John Andrews, for his supportive role, invaluable guidance and friendship during these years.

My thanks also go to Dr Lisa Jackson, who has taken my PhD supervision over the last months, to Dr John Pearson from the System Engineering Innovation Centre for the informative meetings on the fuel rig, and to the lecturers and the research students from the Risk and Reliability research group.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

System Fault Diagnostics represents the process of the identification and isolation of underlying causal faults from a number of effects that are observed on a monitored system [1]. In recent decades, the growing complexity of modern systems has been the motivation for research to study automated diagnostic methods. This is particularly necessary for systems in which safety is important as in the airline industry, power and nuclear plants and for chemical processes.

1.1 Fault Detection and Diagnosis

Systems cannot be perfectly reliable. It can be assumed that at some point any system will encounter some sort of fault. A *Fault* is defined by Isermann [2] as a deviation that prevents the system from fulfilling a particular purpose The general supervision process is illustrated in the following phases:
- *Fault Detection*: When a fault occurs during system monitoring a fault message should be given. The presence of a fault is normally detected through the analysis of signals that come from sensors located on the system. These sensors are usually located at critical points in the system and they measure a number of variables that describe the process function. So, if the signal behaviour deviates from the expected one in the normal operating conditions, a fault is likely to have occurred.
- *Fault Diagnosis*: The cause of the fault should be diagnosed and the failure isolated. This, in principle, could be done manually, inspecting the system components in search of a failure, but it is not always possible. The system may be large, constituted by many sub-systems and components, or complex, so that the inspection is difficult and time consuming. In addition, the fault may represent a hazard and a diagnosis is required quickly. In these

cases, an automated diagnostic system is needed. This is usually a method that uses some mathematical modelling technique or logical thinking and it is able to predict or give an indication of the causes of the fault.
- *Fault Evaluation*: After the cause of the fault is localised, an assessment is carried out to understand how the failure affects the system. Faults can be classified depending on the risk they represent.
- *Decision*: At this stage, considered the type of fault encountered, a decision is made and the system can be either stopped from operating or the operating mode can be changed. In the latter case, the process can continue and repair is delayed. If operation is stopped, the last phase of the process is the:
- *Fault Elimination*: The fault is eliminated either by repairing the component that has caused it or replacing it with a working component and the system can return to its normal functioning.
The aim of a fault diagnostic system is to tell if a fault has occurred and, if this is the case, to diagnose the cause (or causes) of the malfunctioning.

1.2 Fault Diagnostic System Characteristics

The efficiency of a method is measured based on several factors. Venkatasubramanian *et al.* [3] [4] [5] and Price [6] provide two lists of characteristics that a diagnostic system should have. Both of them take into account the detection and diagnosis time. A quick response from a diagnostic system, in the case of failure, can be critical, for example, in a chemical process, where a leak can represent a risk for the people and the environment. Reducing the time taken to diagnose a fault may also reduce the costs of inoperative time for the system. For example, in the airline industry, the diagnosis of a failure in an aircraft before the flight can cause delays and cancellations.

Other factors regard the diagnostic method itself as a system, for example maintainability, efficient construction, minimum effort from the user and optimal cost/benefit ratio. Probably the most important criteria concern the accuracy of the results of the method. The diagnosis, in principle, should be able to identify exactly the failure, or multiple combinations of failures, that have caused the fault. In reality, the diagnosis will provide a set of hypotheses and the real cause should be one of these. The diagnostic method should also be able to recognise any type of failures, including unknown malfunctioning that have not been observed before, this characteristic is called novel identifiability [6]. Moreover, the method should be adaptable to changes in the system, such as changes in the environmental condi-

tions or in the process variables.
The importance of these factors will depend on the particular system application and different methods may meet a criterion better than others, therefore produce better results when applied to particular systems. In general there is not a general diagnostic system that meets all the requirements that were mentioned.

1.3 Fault Diagnostic Approaches

The type of diagnostic reasoning can be identified, depending on the approach taken to the problem, as: *case-based*, *model-based* and *rule-based*.

Case-based diagnostic systems rely on historical data known about previous actions taken for specific fault symptoms. So it uses acquired experience to solve new problems [7]. One important characteristic that a case-based reasoning system depends on is the experience gained, that is the cases available from the acquired data. Experience should be comprehensive and sufficient to cover all possible situations of failures. This can be difficult to obtain for new systems. Moreover, for systems that require safety, such as nuclear plants or aircrafts, there can be a lack of knowledge regarding some extraordinary events due to their infrequent occurrence. On the other hand, case-based reasoning can be useful when the understanding of the system is poor and knowledge of previous cases and actions taken is adequate.

Model-based diagnostics relies on a more theoretical understanding of the system, the so called *systematic knowledge* [8]. This is the knowledge typically obtained from the engineers responsible for designing and building the system and it is related to the physical functioning of the components that constitute the system. In model-based diagnostic reasoning, a theoretical model of the system is built and used to compare its actual behaviour with the expected one. If the system shows some deviation from the model, then the system may be faulty and a list of potential causes is provided analysing the actual functioning of the system and how it should work. A model created for a particular system usually cannot be used for other systems and it is also difficult to adapt to changes in the system itself. For this reason, a model-based approach is considered expensive. Model-based approaches can be divided into qualitative and quantitative models. In [3], quantitative models are considered those who rely on a deep physical understanding of the process and that describe it in terms of the quantitative input and output variables. Qualitative models, instead, describe the process in qualitative terms, considering a symbolic representation of the system variables.

Rule-based diagnostic reasoning makes use of "IF *condition* THEN *consequence*" statements that regards the fault propagation process [9]. The knowledge that is better suited for this type of diagnostic is the *expert knowledge*, which is normally acquired by performing maintenance and it is expressed in the form of cause-effect associations. A diagnostic system that is rule-based will comprise a set of rules and an inference method that derives a decision by analysing and combining such rules. This approach is considered straightforward because each rule regards a different piece of information on the system, such as the relation between symptom and fault or between component and sub-system. On the other hand, rule-based reasoning can be difficult to implement for large systems as the number of rules needed to perform fault diagnostic becomes larger.

1.4 Methods used in Fault Diagnostics

Diagnostic systems make use of several methodologies. The same method can be applied with different approaches. In this section Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA) are briefly introduced. Bayesian Networks are then discussed in more details.

1.4.1 Failure Modes and Effects Analysis

FMEA is a qualitative procedure for the analysis of the failure modes of a system and the determination of their causes and effects [6]. The analysis is carried out considering the effect of each individual potential component failure mode on the system, the component failures can be ranked according to their effects on the functioning of the system. This can be used to determine critical features in the system design or to identify the cause of a fault given the observed effect. A quantitative analysis can also be performed when reliability data are included.

The main limitations of this method are represented by the fact that generally single faults are considered in the system as engineers carry out the inspection and record the information and this can be a time consuming process. For this reason FMEA can be difficult to apply to fault diagnostics [10]. However, the fact the FMEA remains one of the most widely used approaches is motivated by its straightforward procedure. This technique is also used by experts to help identify potential failure modes and weak component or sections in the system.

Recent research has moved to the direction of automated FMEA and extending it to deal with multiple-failure situations [11]. Generating an FMEA automatically is not only faster but it also allows the analysis of multiple faults. Automated FMEA has been applied to electrical automotive systems [12] and for software FMEA [13].

1.4.2 Fault Tree Analysis

FTA is a type of analysis in which an undesired high level event, such as a malfunction in a system, is analysed using logical thinking in order to create a graphical structure that models the causes of the event. It is one of the most used tools in system safety and reliability assessment. A Fault Tree (FT) is constituted by gates and events. The top event represents a failure mode for the system or an undesired event associated with it. By means of Boolean gates (AND/OR), each event is broken down and expressed in terms of its causes that represent lower level events. The process is carried out until basic events are reached. In a system these would be the component failures. An application of FTA to diagnostics with a model-based approach can be found in the work of Hurdle et al. [14], [15]. The authors use non-coherent FTs to model the fault diagnosis of systems including dynamic behaviour. Non-coherent FTs contain NOT logic and are therefore able to deal with both failing and working components in the system. This approach has proved to be more effective than using only coherent FTs [16]. In the method, all deviating scenarios of a system are identified. These are defined as the combinations of sensor readings that result in a deviating behaviour of the system from the expected. For each sensor reading a FT is created and for each scenario, the corresponding FTs for each symptom are combined with an AND gate. Finally, the possible causes of a scenario are found by means of generating the prime implicants of the FT obtained in this way and using the importance measures. The method was first developed considering only static sensor readings. Dynamics have been included in the analysis identifying the sensor patterns for the deviating and non-deviating states of the system. The model is validated on two examples: a water tank system and a fuel rig system. The method has the advantage of developing a complete study for the faults of the system and for its deviating behaviour. Its effectiveness has also been proved. On the other hand, it involves building and analysing very large FTs for each system malfunction.

1.4.3 Bayesian Networks

Bayesian Belief Networks (BBNs) or Bayesian Networks (BNs) are probabilistic graphical models represented as Directed Acyclic Graphs (DAGs). These are applied in many fields

where reasoning under uncertainty is required. The networks are composed of nodes, representing variables of interest (e.g. the occurrence of an event or a component of a system), and links joining the nodes, representing causal relations among the variables. Nodes and links constitute the *qualitative part* of the network, i.e. its structure, while the *quantitative part* is represented by the probability associated with the variables. Each node has a finite number of exhaustive and mutually exclusive states that it can assume. Every node with direct predecessors (parent) is associated with a *Conditional Probability Table* (CPT) that contains the probability of each state of the node for any possible combination of the states of the parents. For the nodes with no parents (root nodes) the CPT specifies the probability of being in each of the states of the associated variable [17].

When the states of some of the variables in a network are known, it is possible to calculate the updated probability, given the new evidence, of the remaining unknown variables. Evaluating this probability, known as *posterior probability*, is the main task in a BN.

1.4.4 System Fault Diagnostics using Bayesian Networks

BNs are ideal for representing visually and conceptually the relations between faults and symptoms characteristic of the processes of fault detection and diagnosis [18], [19]), [20], [21]. Inference in the networks can represent the observed symptoms and posterior probabilities provide a means of identifying possible causes. Inference can actually be introduced with any reasoning associated with some given evidence, not only in symptom outputs but also with any observed facts about the variables of the system.

One of the most stressed factors in the literature about BNs is their capability to handle uncertainty. This term is associated either with the randomness of the problem they are trying to model or with the lack of knowledge on the phenomenon [22]. Model parameters can be difficult to assess and the uncertainty of their estimation propagates through the model giving a corresponding uncertainty in the output. For example, in FTA, uncertainties related to basic events failure probabilities would cause a consequent uncertainty in the top event probability. BNs are considered able to handle uncertainty as the inherent uncertainties in the system can be *absorbed* into the conditional probability tables [23]. Moreover, they are capable of *learning* missing data entries [24]. In [25] the learning problem is generally stated as follows: if U is a set of variables and D is a training set of known values D= $\{u_1, \ldots, u_n\}$, the task is to find a network B that best matches D. The question can be seen as an optimization problem but many other approaches have been studied and most of the research

is now focusing on algorithms and methods in the learning process. The authors in [26] distinguish four classes of learning: known structure and observable data, unknown structure and observable data, known structure and non observable data and unknown structure and non observable data. Learning the structure is considered generally more complicated as the number of possible candidate networks grows exponentially when the number of nodes in the networks grows. Methods have been studied to deduce both the structure and the probabilities of a partially known network, but research has focused more on the derivation of the probabilistic values once the topology is known [27].

Most of the literature on fault diagnostics using BNs has a case-based approach [28], [26], [29], [30]. Generally, the system variables are identified and the topology of the network is created. However, there is not a structured approach on how to do this process. The nodes are created considering the factors that intervene in the diagnostic reasoning and the links are introduced considering how these factors influence one another. Experts generally do this by brainstorming. Once the structure is found, the probability distribution associated to the network is obtained *training* the BN with the history data known about the previous faults. This is a type of learning. When the network is complete, it can be studied to reveal which are the causes of certain faults and it can be used for different types of analysis.

There are not many works in fault diagnostics using BNs with a model-based approach (an example is in [31]). The reason is that specifying the conditional probability tables of a BN is time consuming and requires great efforts from the experts. In most cases, the probability of the networks is generated automatically from data. Despite this tendency, [32] lists many disadvantages of automatically generated networks compared with the ones created directly from the knowledge of experts and engineers. The data sets considered for automatically generated networks is often assumed to be completely representative of the distribution of the data, i.e. *independent and identically distributed*, but this is a very restrictive assumption. The inaccuracy of the data values can also lie in the absence of some fault types. In the diagnosis of a nuclear plant, for example, data of some rare type of accident is scarce and simulations are needed but these can carry errors or distortions.

1.5 Objectives of the research

BNs are an ideal tool to model diagnostic systems. However, their use in fault diagnostics poses two main problems. One concerns the fact that there is not a structured way for building BNs as there is, for example, for FTs. The other problem has to do with the conditional probabilities of the network, which are difficult to obtain or even estimate. For these

reasons, BNs have been mostly used with case-based approaches, obtaining the probabilities by training a network structure with data on previous fault situations. These methods have some disadvantages:
- they need adequate statistical data about observed faults in the system, these data are often incomplete and inadequate;
- automatically generated BNs can be difficult to understand by the expert, especially when the structure is also partially or completely deduced from data;
- algorithms and procedures for training networks can be very complex.
This research aims at finding a way to build a model-based diagnostic system using BNs with the following characteristics:

- The method should give a general and structured procedure to build a BN for the diagnosis of a system. The BN should be obtained from the FTs of the system, assuming that the system functioning is well understood and the failure probability of the components are known.
- The diagnostic should be able to give accurate results applied to a simple water tank system. The BN analysis should be able to tell when a fault has occurred on the system and produce a ranked list of component failures that are potential causes for the system fault.
- The method should be able to handle dynamic effects on the system.
- The results of the method must be validated. A simulation could be written for this purpose to automatically generate the failures in the system and produce the symptom outputs. Introducing the outputs from the simulation into the method itself, one can check if the correct causes are identified for each fault scenario.
- The fuel rig system should be used to test the application of the method for larger and more complex systems. A simulation should again validate the results for this application.

Chapter 2

Bayesian Networks

2.1 Preliminary Notions

2.1.1 Bayesian Probability

According to Judea Pearl [33], in the Bayesian interpretation of probability

(...) *probabilities encode degrees of belief about events in the world and data are used to strengthen, update, or weaken those degrees of belief. In this formalism, degrees of belief are assigned to propositions (sentences that take on true or false values) in some language, and those degrees are combined and manipulated according to the rules of probability calculus.*

In Bayesian Probability, therefore, the mathematical theory of probability is applied to the degree to which a belief is considered *probable*. In this context, the Bayes' theorem gives a criterion for updating belief when new knowledge is introduced. This process is called *Bayesian Inference*. This subsection gives a brief summary of the most important concepts of probability theory considered under this approach.

The object of our study is a *probabilistic model* (or *probability space*), this is defined as an encoding of information that permits us to compute the probability of every well-formed sentence S [33] in accordance with the following three axioms

1. $0 \leq P(A) \leq 1$
2. $P(\text{sure proposition}) = 1$
3. $P(A \text{ or } B) = P(A)+P(B)$ where A and B are mutually exclusive propositions.

(2.1)

In the model, a set of atomic propositions, A, B, C, $\cdots$, is given and the well-formed sentences S correspond to any possible Boolean combination of the atomic propositions. The *elementary events* in the language are combinations of atomic propositions in which they or their negations appear only once. The set of elementary events correspond to the sample space in classical probability theory.

The *joint distribution function* is an application that assigns a non-negative weight to any of the elementary events such that the sum of the weights result to be 1. Any joint probability function gives a complete probabilistic model. In the case of continuous variables, the joint distribution can be given by an algebraic expression such as those describing the normal and exponential distributions. For discrete variables, there are representation methods that infer the distribution from the relationships among the variables. Graphical models are an example of these representations.

From the axioms 2.1, it follows that the probabilities associated to the propositions of the model must respect the following rules:
If B_i are a set of n exhaustive and mutually exclusive propositions, the following law holds

$$P(A) = \sum_{i}^{n} P(A \wedge B_i) = P(A \wedge B_1) + \cdots + P(A \wedge B_n), \tag{2.2}$$

where the notation $A \wedge B_i$ indicates the joint event and $P(A)$ is referred to as the *marginal probability*, that is, the result of the operation of summing up probabilities over all B_i. Equation 2.2 is know as the *law of total probability*.

The formalism $P(A \mid B)$ specifies the belief in A given the assumption that B is known. This is called the *conditional probability*. When $P(A \mid B) = P(A)$, A and B are said to be *independent*. In Bayesian probability, conditional probability is not defined in terms of joint events, $A \mid B$ is rather seen as A in the context specified by B. This approach seems to be more compatible with human reasoning. As a consequence, the joint probability is defined in term of the conditional probability by the following formula

$$P(A \wedge B) = P(A \mid B)P(B) \tag{2.3}$$

which is known as the *product rule*. A useful generalization of the product rule is the following

$$P(A) = \sum_{i} P(A \mid B_i)P(B_i) \tag{2.4}$$

where, again, B_i are exhaustive and mutually exclusive. Here the belief in A is a weighted sum over the beliefs of all possible ways in which A can occur. Assuming that equation 2.4 is applied in some larger context K, it can be written, in a more general way, as:

$$P(A \mid K) = \sum_i P(A \mid B_i, K) P(B_i \mid K). \tag{2.5}$$

Given n events, $A_1,\ A_2,\ \ldots\ ,A_n$, the probability of the joint event can be written as the product of n conditional probabilities by the following formula, called the *chain rule*

$$P(A_1 \wedge A_2 \wedge\ \cdots \wedge\ A_n) = P(A_n \mid A_{n-1},\ \cdots,\ A_2,\ A_1) \cdots\ P(A_2 \mid A_1)\ P(A_1). \tag{2.6}$$

As it has been said before, in Bayesian probability the concept of inference plays a central role. The rule of updating probabilities is given by the following

$$P(H \mid e) = \frac{P(e \mid H) P(H)}{P(e)}, \tag{2.7}$$

which gives the *posterior probability* of H given the evidence e, $P(H \mid e)$, in terms of the previous belief $P(H)$, or *prior probability*, and $P(e \mid H)$, the likelihood that e will occur when H is true. Equation 2.7 is called the *inversion formula*. In some texts, equation 2.7 is stated as *Bayes' Theorem*.

2.1.2 Graphs

A *graph* is defined as a set V of *vertices* or *nodes* together with a set E of *edges* or *links* connecting some vertices in pairs. A variable is associated to any vertex and the links represent a certain relation that correlates two variables. When an edge has a single arrowhead, it is *directed*. When all edges in a graph are directed, then the graph itself is said to be *directed*. Figure 2.1 shows an example of a directed graph with $V = \{A,\ B,\ C,\ D,\ E,\ F,\ G\}$ and $E = \{(A, C), (B, D), (C, E), (D, A), (D, E), (D, F), (E, G)\}$.

A *path* in a graph is a sequence of edges such that each of them starts with the vertex ending in the previous edge, e.g. $\{(A, C), (C, E), (E, D)\}$ in figure 2.2, where the notation (A, C) represents the edge connecting the vertex A to vertex C.

Two vertices in a graph are said to be *connected* if there exists a path between them. Although self-loop cycles are not admitted in a graph, i.e. an edge cannot connect the same

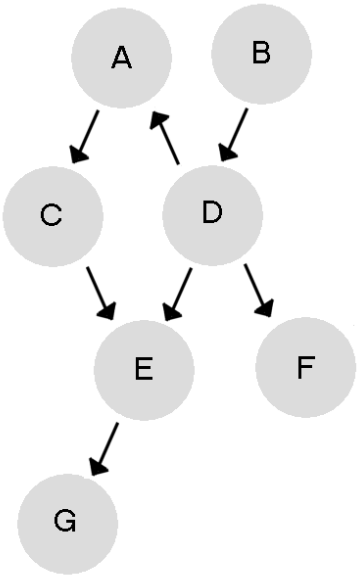


Figure 2.1 – Example of a directed graph.

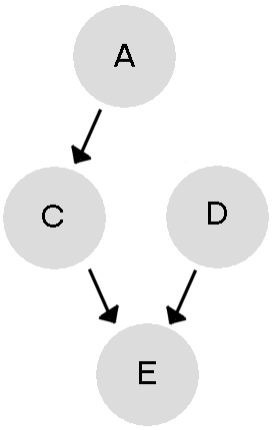


Figure 2.2 – Example of a path of the graph in figure 2.1.

vertex (e.g. (A, A)), a directed graph may have directed *cycles*, that is a path starting and ending with the same vertex. A graph is called *acyclic* if it contains no such directed cycles. When a graph is both directed and acyclic, then it is called a *Directed Acyclic Graph* (DAG). The graph in figure 2.1 is an example of DAG.

Undirected graphs, also called *Markov networks*, and DAGs have been used to facilitate the representation of probability distributions and to facilitate the process of inference [34]. Undirected graphs find their applications in the representation of spatial relations, while DAGs represent causal and temporal relations.

2.2 Definition and Properties of Bayesian Networks

2.2.1 Definition

A Bayesian Network (BN) is defined as a pair $B = ((V, E), P)$ where (V, E) is a directed acyclic graph constituted by the set of discrete variables $V = \{X_1,\ X_2,\ \dots X_n\}$ and the set of edges E, and P is the *joint distribution* associated to the variables. Each variable has a finite set of exhaustive and mutually exclusive states. The edges represent a causal relation between two nodes, in the sense that the node *parent* is a direct cause of its node *son* (figure 2.3).

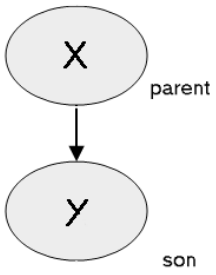


Figure 2.3 – x causes y.

The *joint distribution* $P(X_1,\ X_2,\ \dots X_n)$ over the set of variables $\{X_1,\ X_2,\ \dots X_n\}$ is defined as a table of the joint probabilities $P(X_1 \wedge\ X_2 \wedge \cdots \wedge\ X_n)$ given for all values that $\{X_1,\ X_2,\ \dots X_n\}$ can assume. In the very simple case of two binary variables X_1 and X_2, with states, respectively, $\{X_1,\ \bar{X_1}\}$ and $\{X_2,\ \bar{X_2}\}$, the joint distribution $P(X_1, X_2)$ is given by the probabilities $P(X_1 \wedge X_2)$, $P(\bar{X_1} \wedge X_2)$, $P(X_1 \wedge \bar{X_2})$ and $P(\bar{X_1} \wedge \bar{X_2})$.
The chain rule of probability (equation 2.6) applied to the variables $\{X_1,\ X_2,\ \dots X_n\}$ gives the joint distribution

$$P(X_1,\ \dots,\ X_n) = \prod_i P(X_i \mid Pa_i), \tag{2.8}$$

where Pa_i are the predecessors of X_i to which the probability of X_i is sensitive to, also called *Markovian parents* of X_i. Rigorously, $P(a_i)$ is defined as the minimal subset of $\{X_1,\ X_2,\ \dots X_n\}$ satisfying

$$P(X_i \mid Pa_i) = P(X_i \mid X_1,\ \cdots,\ X_{i-1}), \tag{2.9}$$

where $P(a_i)$ is minimal in the sense that none of its proper subsets satisfies equation 2.9. The joint distribution of a set of variables gives all information needed about the distribution.
From equation 2.8, it can be seen that the probability function in a BN is given by specifying

a set of conditional independence assumptions together with a set of *Conditional Probability Tables* (CPTs), that is specifying the prior probabilities of all root nodes and the conditional probabilities of all non-root nodes given all possible combinations of their parents. As the edges in the graph represent the conditional relations between a node and its parents, a variable is independent from its non-descendants variables.
Take the example of a BN with ten variables as in figure 2.4,

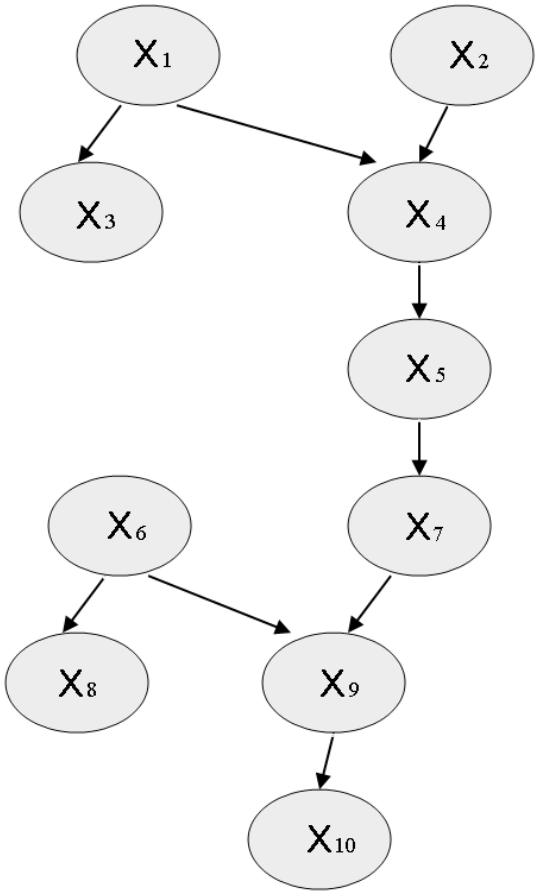


Figure 2.4 – Example of BN.

here the joint distribution is given by

$$P(X_1,\ X_2,\ \cdots,\ X_{10}) = \prod_i P(X_i \mid pa_i) =$$

$$P(X_1)P(X_2)P(X_3 \mid X_1)P(X_4 \mid X_1,\ X_2)P(X_5 \mid X_4)$$

$$P(X_7 \mid X_5)P(X_6)P(X_8 \mid X_6)P(X_9 \mid X_6,\ X_7)P(X_{10} \mid X_9).$$

The prior probability has to be specified for X_1, X_2 and X_6, which are the only nodes without

parents. For all other variables, a CPT has to be provided. If we assume that the variables are binary, for example they represent the components of a system in the states W (*working*) and F (*failing*), the CPT for a variable with n parents will have 2^n entries. For the variable X_9 in the graph in figure 2.4, for example, the entries of the CPT correspond to all possible values of the variables parents X_6 and X_7: $\{X_6 = W, X_7 = W\}$, $\{X_6 = W, X_7 = F\}$, $\{X_6 = F, X_7 = W\}$ and $\{X_6 = F, X_7 = F\}$. Therefore, the probabilities in the CPT will be 8:

1. $P(X_9 = W \mid X_6 = W \wedge\ X_7 = W)$,
2. $P(X_9 = F \mid X_6 = W \wedge\ X_7 = W)$,
3. $P(X_9 = W \mid X_6 = W \wedge\ X_7 = F)$,
4. $P(X_9 = F \mid X_6 = W \wedge\ X_7 = F)$,
5. $P(X_9 = W \mid X_6 = F \wedge\ X_7 = W)$,
6. $P(X_9 = F \mid X_6 = F \wedge\ X_7 = W)$,
7. $P(X_9 = W \mid X_6 = F \wedge\ X_7 = F)$,
8. $P(X_9 = F \mid X_6 = F \wedge\ X_7 = F)$.

The probability in the network can be updated from observation introducing *evidence* for some variables, that is, assigning fixed values to some of them. Evidence will be denoted as e, where $e = \{X_i, \cdots, X_m\}$ is a subset of variables. The task consists in calculating the posterior probability distribution on a set of variables of interest Q, given the evidence e: $P(Q \mid e)$. In the network shown in figure 2.4, we could for example obtain evidence that indicates that the component represented by X_4 is definitely *working*. This fact introduces the evidence $e = \{X_4 = W\}$. This evidence influences the causal connections between X_1 and X_5, in fact, if before the state of X_1 had an influence on X_5, now they are independent. It can be said that X_4 blocks the flow of information along the graph, this concept is known as *d-separation*. To define it rigorously, some concepts about network connection types need to be introduced. A node B in a path T is said to be *linear*, *converging* or *diverging* if it is connected to the previous and the following nodes according to figure 2.5.

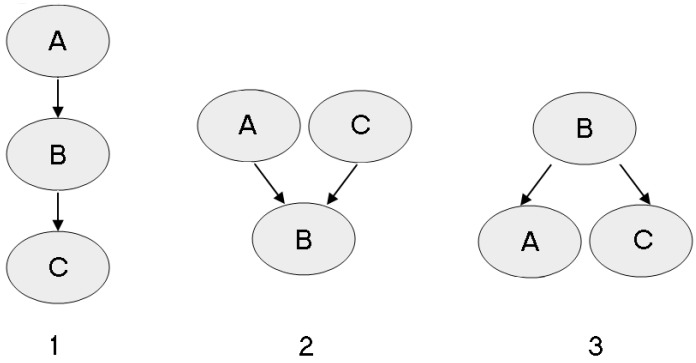


Figure 2.5 – Connection Types: 1. Linear, 2. Converging, 3. Diverging.

In [33] the three configurations are referred to respectively as *chain*, *collider* and *fork*. A path T from A to B with the evidence nodes e is said to be *d-connecting* (and the nodes A and B are *d-connected*) if every interior node I in the path is either:

- linear or diverging and not in e,
- converging, and either I or one of its descendants is a member of e.

Two nodes will be *d-separated* if there are no d-connecting paths between them. This gives another independence rule between the variables, apart from the trivial rule that a variable is independent from its non-descendants. In the previous example, node X_5 will be independent from X_3 if evidence about X_4 is given.

The basic task in BNs is the computation of $P(H \mid e)$, where e is a set of observations and H is a set of variables of interest. This is known as *inference*. The computation of the probability is conceptually easy as it comes from the inversion rule (equation 2.7) where the joint probabilities can be computed from the conditional probabilities defined in the CPTs. This will be made more clear with two examples in the following subsections.

2.2.2 Example: Fire alarm system

Let us take the example of a fire alarm system that is often considered in literature. The alarm can start caused by a fire or by a test. There is a certain probability that the alarm would fail in both cases. When the alarm goes off, an evacuation is planned to take place. Again there is a probability that the evacuation would fail to take place when the alarm goes on. The variables of a BN that can model this system are *fire*, *test*, *alarm* and *evacuation*

and they will all be binary with the states *true* (t) and *false* (f). The BN will be as in figure 2.6.

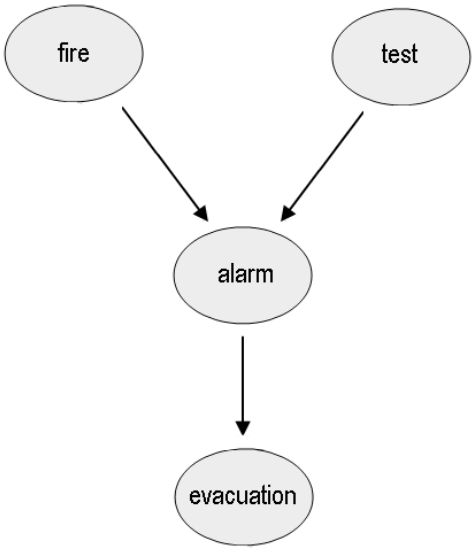


Figure 2.6 – Example of BN for the fire alarm system.

To specify the joint distribution the prior probability for the variables without parents, *fire* and *test*, are needed:

$$\begin{aligned} P(fire = \mathrm{t}) &= 0.001, \\ P(fire = \mathrm{f}) &= 0.999, \end{aligned} \tag{2.10}$$

$$\begin{aligned} P(test = \mathrm{t}) &= 0.1, \\ P(test = \mathrm{f}) &= 0.9, \end{aligned} \tag{2.11}$$

and the CPTs for the variables *alarm* and *evacuation* are shown in figures 2.7 and 2.8.

Parent Node(s)		alarm		
fire	Test	T	F	bar charts
T	T	0.9	0.1	
	F	0.8	0.2	
F	T	0.8	0.2	
	F	0.01	0.99	

Figure 2.7 – CPT for the node *alarm*.

Parent Node(s)	Evacuation		
alarm	T	F	bar charts
T	0.9	0.1	
F	0.001	0.999	

Figure 2.8 – CPT for the node *evacuation*.

The probability of *evacuation* can be calculated from the *law of probability* in equation (2.2) marginalizing all the variables except *evacuation* out of $P(evacuation, alarm, fire, test)$:

$$P(evacuation) = \sum_{alarm} \sum_{fire} \sum_{test} P(evacuation, alarm, fire, test), \tag{2.12}$$

writing explicitly the sums for the variable *test*, *fire* and *alarm* , equation (2.12) becomes

$$P(evacuation, alarm = t, fire = t, test = t) + P(evacuation, alarm = t, fire = t, test = f) +$$

$$+ P(evacuation, alarm = t, fire = f, test = t) + P(evacuation, alarm = t, fire = f, test = f) +$$

$$+ P(evacuation, alarm = f, fire = t, test = t) + P(evacuation, alarm = f, fire = t, test = f) +$$

$$+ P(evacuation, alarm = f, fire = f, test = t) + P(evacuation, alarm = f, fire = f, test = f).$$

Each of these can be obtained in terms of the conditional probabilities and the prior probabilities by means of the *chain rule* in equation 2.6, therefore the probability of *evacuation* becomes:

$$P(evacuation) \; =$$

$$= \; P(evacuation \mid alarm = t)P(alarm = t \mid \; fire = t \; \wedge \; test = t)P(fire = t)P(test = t) +$$

$$+ P(evacuation \mid alarm = t)P(alarm = t \mid \; fire = t \; \wedge \; test = f)P(fire = t)P(test = f) +$$

$$+ P(evacuation \mid alarm = t)P(alarm = t \mid \; fire = f \; \wedge \; test = t)P(fire = f)P(test = t) +$$

$$+ P(evacuation \mid alarm = t)P(alarm = t \mid \; fire = f \wedge test = f)P(fire = f)P(test = f) +$$

$$+ P(evacuation \mid alarm = f)P(alarm = f \mid \; fire = t \; \wedge \; test = t)P(fire = t)P(test = t) +$$

$$+ P(evacuation \mid alarm = f)P(alarm = f \mid \; fire = t \wedge test = f)P(fire = t)P(test = f) +$$

$$+ P(evacuation \mid alarm = f)P(alarm = f \mid \; fire = f \wedge test = t)P(fire = f)P(test = t) +$$

$$+ P(evacuation \mid alarm = f)P(alarm = f \mid \; fire = f \; \wedge \; test = f)P(fire = f)P(test = f).$$

For the state *true* of *evacuation*, from the CPTs, it follows that:

$$P(evacuation = t) \ = 0.9 \ \cdot \ 0.9 \ \cdot \ 0.001 \ \cdot \ 0.1 \ + \ 0.9 \ \cdot \ 0.8 \ \cdot \ 0.001 \ \cdot \ 0.9 \ +$$

$$+ \ 0.9 \ \cdot \ 0.8 \ \cdot \ 0.999 \ \cdot \ 0.1 \ + \ 0.9 \ \cdot \ 0.01 \ \cdot \ 0.999 \ \cdot \ 0.9 \ +$$

$$+ \ \ 0.001 \ \cdot \ 0.1 \ \cdot \ 0.001 \ \cdot \ 0.1 \ + \ 0.001 \ \cdot \ 0.2 \ \cdot \ 0.001 \ \cdot \ 0.9 \ +$$

$$+ \ 0.001 \cdot \ 0.2 \ \cdot \ 0.999 \ \cdot \ 0.1 \ + \ 0.001 \ \cdot \ 0.99 \ \cdot \ 0.999 \ \cdot \ 0.9 \ \ = 0.0816.$$

When evidence is introduced on some variables in the network, the posterior probability can be calculated employing Bayes' rule (equation 2.7). In general, for any joint distribution P, the application of the Bayes' rule leads to the formula:

$$P(H \mid E) = \frac{P(E \mid H)P(H)}{P(E)} = \frac{\sum_{S} P(H, E, S)}{\sum_{H} \sum_{S} P(H, E, S)} \tag{2.13}$$

where H is the set of variables of interest, E is the given evidence and S is the set of all variables excluding the ones in H and in E. In this case, one could be interested in knowing the probability of an evacuation when a fire is not taking place, that is when $\mathit{fire} = f$. This is expressed by the conditional probability $P(fire = f \ \mid \ evacuation = t)$ and, by equation 2.13, is given by:

$$P(fire = f \ \mid \ evacuation = t) = \frac{\sum_{alarm} \sum_{test} P(evacuation = t, \ alarm, \ fire = f, \ test)}{\sum_{fire} \sum_{alarm} \sum_{test} P(evacuation = t, \ alarm, \ fire, \ test)}. \tag{2.14}$$

The numerator in 2.14 is given by:

$$\sum_{alarm} \sum_{test} P(evac = t, \ alarm, \ fire = f, \ test) =$$

$$P(evac = t \ \wedge \ alarm = t \ \wedge \ fire = f \ \wedge \ test = t) +$$

$$+ \ P(evac = t \ \wedge \ alarm = f \ \wedge \ fire = f \ \wedge \ test = t) +$$

$$+ \ P(evac = t \ \wedge \ alarm = t \ \wedge \ fire = f \ \wedge \ test = f) +$$

$$+ \ P(evac = t \ \wedge \ alarm = f \ \wedge \ fire = f \ \wedge \ test = f) =$$

$$= P(evac = t \mid alarm = t)\, P(alarm = t \mid fire = f \wedge test = t)\, P(fire = f)\, P(test = t) +$$

$$+ P(evac = t \mid alarm = f)\, P(alarm = f \mid fire = f \wedge test = t)\, P(fire = f)\, P(test = t) +$$

$$+ P(evac = t \mid alarm = t)\, P(alarm = t \mid fire = f \wedge test = f)\, P(fire = f)\, P(test = f) +$$

$$+ P(evac = t \mid alarm = f)\, P(alarm = f \mid fire = f \wedge test = f)\, P(fire = f)\, P(test = f) =$$

$$= 0.9 \cdot 0.8 \cdot 0.999 \cdot 0.1 + 0.001 \cdot 0.2 \cdot 0.999 \cdot 0.1 + 0.9 \cdot 0.01 \cdot 0.999 \cdot 0.9 +$$

$$+ 0.001 \cdot 0.2 \cdot 0.999 \cdot 0.9 = 0.0802.$$

While the denominator in 2.14 is given by:

$$\sum_{fire} \sum_{alarm} \sum_{test} P(evacuation = t,\ alarm,\ fire,\ test) =$$

$$P(evac = t \wedge alarm = t \wedge fire = f \wedge test = t) +$$

$$+ P(evac = t \wedge alarm = t \wedge fire = f \wedge test = f) +$$

$$+ P(evac = t \wedge alarm = t \wedge fire = t \wedge test = t) +$$

$$+ P(evac = t \wedge alarm = t \wedge fire = t \wedge test = f) +$$

$$+ P(evac = t \wedge alarm = f \wedge fire = f \wedge test = t) +$$

$$+ P(evac = t \wedge alarm = f \wedge fire = f \wedge test = f) +$$

$$+ P(evac = t \wedge alarm = f \wedge fire = t \wedge test = t) +$$

$$+ P(evac = t \wedge alarm = f \wedge fire = t \wedge test = f) =$$

$$= P(evac = t \mid alarm = t)\, P(alarm = t \mid fire = f \wedge test = t)\, P(fire = f)\, P(test = t) +$$

$$+ P(evac = t \mid alarm = t)\, P(alarm = t \mid fire = f \wedge test = f)\, P(fire = f)\, P(test = f) +$$

$$+ P(evac = t \mid alarm = t)\, P(alarm = t \mid fire = t \wedge test = t)\, P(fire = t)\, P(test = t) +$$

$$+ P(evac = t \mid alarm = t)\, P(alarm = t \mid fire = t \wedge test = f)\, P(fire = t)\, P(test = f) +$$

$$+ P(evac = t \mid alarm = f)\, P(alarm = f \mid fire = f \wedge test = t)\, P(fire = f)\, P(test = t) +$$
$$+ P(evac = t \mid alarm = f)\, P(alarm = f \mid fire = f \wedge test = f)\, P(fire = f)\, P(test = f) +$$
$$+ P(evac = t \mid alarm = f)\, P(alarm = f \mid fire = t \wedge test = t)\, P(fire = t)\, P(test = t) +$$
$$+ P(evac = t \mid alarm = f)\, P(alarm = f \mid fire = t \wedge test = f)\, P(fire = t)\, P(test = f) =$$

$$= 0.9 \cdot 0.8 \cdot 0.999 \cdot 0.1 + 0.9 \cdot 0.01 \cdot 0.999 \cdot 0.9 +$$
$$+ 0.9 \cdot 0.9 \cdot 0.001 \cdot 0.1 + 0.9 \cdot 0.8 \cdot 0.001 \cdot 0.9 +$$
$$+ 0.001 \cdot 0.2 \cdot 0.999 \cdot 0.1 + 0.001 \cdot 0.9 \cdot 0.999 \cdot 0.9 +$$
$$+ 0.001 \cdot 0.1 \cdot 0.001 \cdot 0.1 + 0.001 \cdot 0.2 \cdot 0.001 \cdot 0.9 = 0.0816.$$

And therefore the result is given by:

$$P(fire = f \mid evacuation = t) = \frac{0.0802}{0.0816} = 0.9828,$$

this is the probability of an evacuation taking place with no fire.

2.2.3 Example: Happiness

We now consider an example where we try to model how the *happiness* of a individual is influenced. In a very simplified situation, we could assume that there are three factors that make an individual happy: *health*, *love* and *success*. A BN describing our *system* will have four nodes and *health*, *love* and *success* will be parents of *happiness*. We can assume all four variables will have three states: *low* (l), *medium* (m) and *high* (h). The structure of the network will be as in figure 2.9.

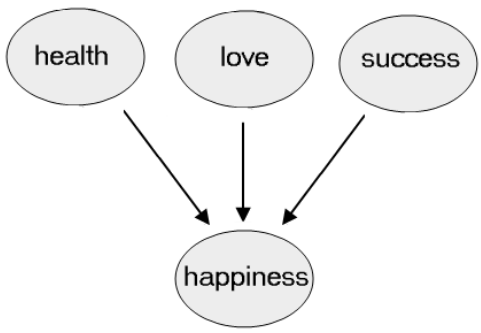


Figure 2.9 – Example of BN.

To specify the joint distribution we need to know the prior probability for the variables

without parents, *health* (HE),

$$P(HE = l) = 0.1,\ P(HE = m) = 0.3,\ P(HE = h) = 0.6, \tag{2.15}$$

$$P(LO = l) = 0.3,\ P(LO = m) = 0.4,\ P(LO = h) = 0.3, \tag{2.16}$$

$$P(SU = l) = 0.3,\ P(SU = m) = 0.4,\ P(SU = h) = 0.3, \tag{2.17}$$

the CPT for the node *Happiness* (HA) will have 3^3 entries, as it has 3 parents and 3 states. Figure 2.10 shows the CPT for *happiness*:

Parent Node(s)			HA			
HE	LO	SU	low	medium	high	bar charts
low	low	low	0.907	0.07404	0.01896	
		medium	0.866	0.10385	0.03015	
		high	0.814	0.145	0.041	
	medium	low	0.763	0.196	0.041	
		medium	0.732	0.206	0.062	
		high	0.691	0.227	0.082	
	high	low	0.649	0.238	0.113	
		medium	0.608	0.237	0.155	
		high	0.567	0.258	0.175	
medium	low	low	0.515	0.279	0.206	
		medium	0.464	0.32	0.216	
		high	0.423	0.34	0.237	
	medium	low	0.361	0.371	0.268	
		medium	0.299	0.412	0.289	
		high	0.237	0.433	0.33	
	high	low	0.175	0.454	0.371	
		medium	0.144	0.444	0.412	
		high	0.113	0.444	0.443	
high	low	low	0.124	0.38967	0.48633	
		medium	0.124	0.3285	0.5475	
		high	0.113	0.30278	0.58422	
	medium	low	0.113	0.29547	0.59153	
		medium	0.093	0.28331	0.62369	
		high	0.082	0.27174	0.64626	
	high	low	0.062	0.27187	0.66613	
		medium	0.041	0.237	0.722	
		high	0.021	0.17286	0.80614	

Figure 2.10 – CPT for the node *happiness*.

To calculate $P(HA)$, from the *law of total probability* in equation 2.2, we need to marginalise HE, LO and SU out of $P(HA, HE, LO, SU)$:

$$P(HA) = \sum_{HE} \sum_{LO} \sum_{SU} P(HA, HE, LO, SU), \tag{2.18}$$

where the probability of the joint event can be calculated in terms of the conditional probabilities:

$$P(HA = h \ \wedge \ HE = l \ \wedge \ LO = m \ \wedge \ SU = l) =$$

$$= P(HA = h \mid HE = l \ \wedge \ LO = m \ \wedge \ SU = l)P(HE = l) \ P(LO = m) \ P(SU = l)$$

which, considering equations (2.15) (2.16) (2.17) and substituting the numbers from the CPT in figure 2.10, gives

$$= 0.041 \cdot 0.1 \cdot 0.4 \cdot 0.3 = 0.0016. \tag{2.19}$$

Summing up all the possible combinations for the node *happiness*, the prior probabilities are:

$$P(HA = l) = 0.2159, \ P(HA = m) = 0.3265, \ P(HA = h) = 0.4776. \tag{2.20}$$

In terms of the posterior probabilities, for example, one could be interested in knowing the probability of *love* being in the state *low* when *happiness* is *high*. This is given by:

$$P(LO = l \mid HA = h) = \frac{\sum_{HE} \sum_{SU} P(LO = l, \ HA = h, \ HE, \ SU)}{\sum_{HE} \sum_{SU} \sum_{LO} P(LO, \ HA = h, \ HE, \ SU).} \tag{2.21}$$

The denominator in 2.21 equals $P(HA = h)$, which has already been obtained. The numerator is the marginalisation of HE and SU out of $P(LO = l, \ HA = h, \ HE, \ SU)$ and , applying the chain rule (equation 2.6), it can be obtained as:

$$\sum_{HE} \sum_{SU} P(LO = l, \ HA = h, \ HE, \ SU) =$$

$$P(HA = h \mid HE = l \ \wedge LO = l \wedge SU = l)P(HE = l)P(LO = l)P(SU = l)+$$

$$+P(HA = h \mid HE = l \ \wedge LO = l \wedge SU = m)P(HE = l)P(LO = l)P(SU = m)+$$

$$+P(HA = h \mid HE = l \ \wedge LO = l \wedge SU = h)P(HE = l)P(LO = l)P(SU = h)+$$

$$+P(HA = h \mid HE = l \ \wedge LO = l \wedge SU = l)P(HE = l)P(LO = l)P(SU = l)+$$

$$+P(HA = h \mid HE = l \ \wedge LO = m \wedge SU = l)P(HE = l)P(LO = m)P(SU = l)+$$

$$+P(HA = h \mid HE = l \ \wedge LO = h \wedge SU = l)P(HE = l)P(LO = h)P(SU = l)+$$

$$+P(HA = h \mid HE = l \ \wedge LO = l \wedge SU = l)P(HE = l)P(LO = l)P(SU = l)+$$

$$+P(HA = h \mid HE = m \ \wedge LO = l \wedge SU = l)P(HE = m)P(LO = l)P(SU = l)+$$

$$+P(HA = h \mid HE = h \ \wedge LO = l \wedge SU = l)P(HE = h)P(LO = l)P(SU = l) =$$

$$0.01896 \cdot 0.1 \cdot 0.3 \cdot 0.3 + 0.03015 \cdot 0.1 \cdot 0.3 \cdot 0.4 + 0.0041 \cdot 0.1 \cdot 0.3 \cdot 0.3 +$$

$$+0.206 \cdot 0.3 \cdot 0.3 \cdot 0.3 + 0.216 \cdot 0.3 \cdot 0.3 \cdot 0.4 + 0.237 \cdot 0.3 \cdot 0.3 \cdot 0.3 +$$

$$+0.48633 \cdot 0.6 \cdot 0.3 \cdot 0.3 + 0.5475 \cdot 0.6 \cdot 0.3 \cdot 0.4 + 0.58422 \cdot 0.6 \cdot 0.3 \cdot 0.3 = 0.1178.$$

Therefore:

$$P(LO = l \mid HA = h) = \frac{0.1178}{0.4776} = 0.2468. \tag{2.22}$$

Since the connection type of the network is converging, if evidence is not given to the node *happiness*, the remaining three variables are d-separated and, as a consequence, independent. Therefore:

$$P(HE \mid \ LO,\ SU) = P(HE),$$

$$P(LO \mid HE,\ SU) = P(LO),$$

$$P(SU \mid HE,\ LO) = P(SU).$$

On the other hand, when evidence is given to the variable *happiness*, the three parents will become d-separated and they will influence each other. For example, if the state of the *happiness* is known, and it is *high*, assuming *love* is in the state *low* makes the probability of the remaining two variables, *health* and *success*, increase. In fact, under the hypothesis of our model, knowing that the individual is happy but without love would make us infer he is healthy or successful. If instead the state of the variable *happiness* is known to be *low*, assuming the state of *love* is *high* makes the probability of *health* and *success* decrease. In fact, if $e = \{HA = h,\ LO = l\}$, for example, $P(HE = l)$ is given by:

$$P(HE = l \mid HA = h \wedge LO = l) = \frac{\sum_{SU} P(HE = l,\ HA = h,\ LO = l,\ SU)}{\sum_{SU}\sum_{HE} P(HE,\ HA = h,\ LO = l,\ SU)} = 0.0076. \tag{2.23}$$

The same calculation can be done updating the prior probability with the evidence introducing to HA, $e = \{HA = high\}$. The updated probability P^* can be calculated for HE, LO and SU as before and they will result to be:

$$\begin{aligned} P^*(HE = l) &= P(HE = l \mid HA = h) = 0.0164, \\ P^*(HE = m) &= P(HE = m \mid HA = h) = 0.1928, \\ P^*(HE = h) &= P(HE = h \mid HA = h) = 0.7908, \end{aligned} \tag{2.24}$$

$$\begin{aligned} P^*(LO = l) &= P(LO = l \mid HA = h) = 0.2468, \\ P^*(LO = m) &= P(LO = m \mid HA = h) = 0.3915, \\ P^*(LO = h) &= P(LO = h \mid HA = h) = 0.3617, \end{aligned} \tag{2.25}$$

$$\begin{aligned} P^*(SU = l) &= P(SU = l \mid HA = h) = 0.2761, \\ P^*(SU = m) &= P(SU = m \mid HA = h) = 0.3998, \\ P^*(SU = h) &= P(SU = h \mid HA = h) = 0.3241. \end{aligned} \tag{2.26}$$

Then the probability in 2.23 is given by

$$\begin{aligned} P(HE = h \mid HA = h \wedge LO = l) = P^*(HE = h \mid LO = l) = \\ = \frac{\sum_{SU} P^*(HE = l,\ LO = l,\ SU)}{\sum_{SU}\sum_{HE} P^*(HE,\ LO = l,\ SU)} = 0.0076 \end{aligned} \tag{2.27}$$

which eliminates the variable HA and gives the same result.

2.2.4 Network Simplification Methods

The *Noisy or* assumption

The number of entries in the CPTs of a BN can easily become large, it has been seen in fact from the previous example that this number grows exponentially with the number of parents and for a variable with m states and n parents it equals m^n.
Some simplification in the assumptions can help in reducing it. The so called *noisy or* assumption is one of these, it allows the number of probabilities to specify for a variable to grow linearly with the number of its parents.
Consider the configuration shown in figure 2.11

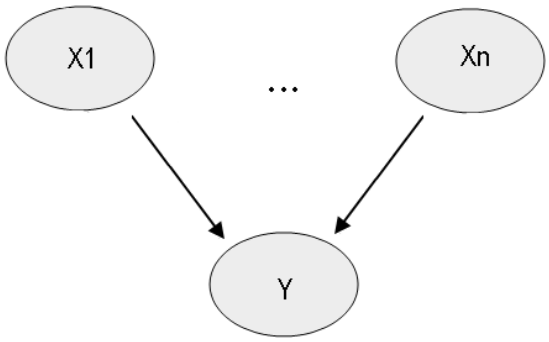


Figure 2.11 – Configuration for the *noisy or* assumption.

and let $X_1, \cdots, X_n$ be binary parents of Y with states T (*true*) and F (*false*) and assume that, for any i, $X_i = T$ causes $Y = T$ with a certain probability, unless a preventing factor, called an *inhibitor*, prevents it with probability q_i. The *noisy or* assumption assumes that all such inhibitors are independent, therefore, if

$$P(Y = F \mid X_i = T) = q_i$$

then

$$P(Y = F \mid X_1 = T, \cdots, X_n = T) = \prod_{i=1}^{n} q_i.$$

We see now the effect of this assumption in an example. In the network for the fire alarm system in figure 2.6, the node *alarm* has two parents, *fire* and *test*. Both of them, when in the state *true*, cause the alarm to be *true* with a certain probability, so there is a preventing factor that causes the state of the parents not to affect the one of the child. For example, for *fire*, this could be due to the failure of the sensor and, for *test*, to the failure of the testing system itself. In a general situation, as the two nodes parents are binary, the CPT for alarm should contain 4 probabilities (as in figure 2.7). Assuming the preventing factors *inhibitors* of the parents are independent, as in the case described, reduces the probabilities to be specified to 2, in fact, if we assume the *inhibitors* have probability q_1 and q_2:

$$P(alarm = F \mid fire = T) = q_1,$$

$$P(alarm = F \mid test = T) = q_2,$$

that is, q_1 the probability of the failure of the sensor and q_2 the probability of the failure of the test system, we will have that:

$$P(alarm = T \mid fire = T \wedge test = F) = 1 - P(alarm = F \mid fire = T \wedge test = T) = 1 - q_1 q_2.$$

Assuming that the inhibitors are independent makes it easier to calculate the combined probabilities so, in the CPTs, it will not be necessary to specify the probability of a variable for any configuration of the parents.

Divorcing

The *noisy or* assumption is a particular case of a more general method called *divorcing*. This consists in introducing a *mediating* variable that plays the role of child for a number of parents so that the configurations are partitioned into smaller sets. For example, consider the configuration in figure 2.12. This can be transformed introducing a variable Z as the child of X_1 and X_2 and parent of Y. If we assume the variables are all binary, the CPT for Y in figure 2.12 contains 2^4 probabilities. In figure 2.13 the CPT for Y contains now 2^3 probabilities and a CPT for Z has to be introduced with 2^2 entries, but, still, $2^2 + 2^3 < 2^4$.

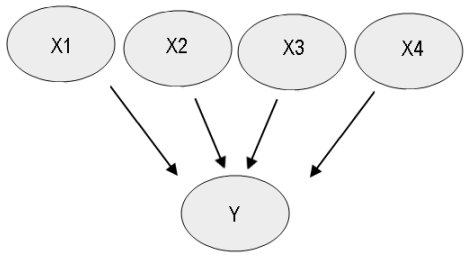


Figure 2.12 – Configuration for the *divorcing* method.

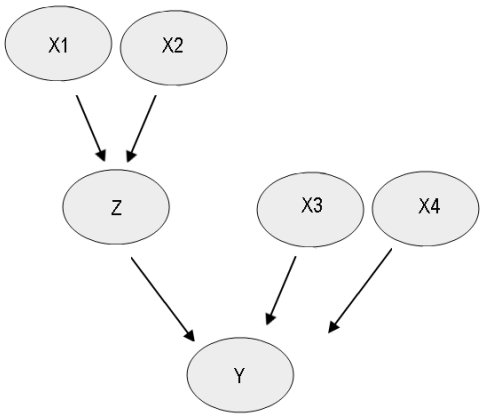


Figure 2.13 – X_1 and X_2 are *divorced* from X_3 and X_4 introducing Z.

2.2.5 Learning

The term *learning* indicates semi-automatic methods able to modify or evaluate a model by means of the experience obtained in creating it. It can be either qualitative or quantitative, depending on whether it concerns the topological structure or the probability of the network, and it is called *batch learning* if it makes use of databases of cases or *adaption* if it involves a process of consecutive modifications when new cases are acquired [17]. *Batch learning* falls under the category of case-based reasoning as it makes use of acquired data from previous experience.

Estimating the prior probabilities of a BN can be subjective and there are several methods to carry out this task. If two or more methods are given to determine the probabilities of a network whose structure is known and these give rise to different distributions, a learning process can be used to find the model that is the closest to the real process or the smallest and less expensive. In modifying a distribution during a learning process, some measurements are introduced to estimate the distance between a distribution and its approximation or modification. Among the others, the *euclidean distance* between two distributions P and P' is defined as:

$$\text{Dist } (P,\ P') = \sum_{X} \left(P(X) - P'(X)\right)^2. \tag{2.28}$$

In order to compare the structure of two networks in a process of qualitative learning we need to define the *size* of a BN. This is also a measure that can be useful when algorithms are implemented for models with a very large number of variables. Given a network B with variables V, denoting with $\text{Sp}(X)$ the number of all different configurations that the parents of X can assume, the *Size* of B is given by:

$$\text{Size } (B) = \sum_{X \in V} \text{Sp } (X). \tag{2.29}$$

The networks in figure 2.12 and 2.13 have respectively size 36 and 26. For the first one, 36 is given by $4 + 2^5$, the four probabilities of the nodes without parents plus the 2^5 probabilities in the CPT of the variable Y. For the second network, 26 is given by $4 + 2^3 + 2^4$, the four probabilities of the variables without parents, the 2^3 probabilities of the CPT of Z and the 2^4 probabilities of the CPT of Y.

Size and distance are used to define the *acceptance measure* of a modified network as:

$$\text{Acc } (P, B') = \text{ Size}(B') + k \text{ Dist}(P, P'), \tag{2.30}$$

where P' is the joint probability associated with the modified structure B', P is the one for B and k is a positive real number. In a learning process searching for a possible structure of a BN the *acceptance* has to be minimized to get a network of small size whose distribution is not too far from the original one.

2.2.6 Algorithm Methods

It has been seen how inference in a BN can be calculated by means of the Bayes' rule, however, when performing the evaluation automatically, the task consists of implementing an efficient algorithm to perform the computation for any given graphical structure. Inference algorithms are generally NP-hard (Non-deterministic Polynomial-time hard) but some methods have been found to be useful. Some of them, such as the *message passing* approach or the *join-tree* and *cut-set conditioning* methods, are briefly described by Pearl and Langseth [33] [35] and, in more details in [17].

2.2.7 Dynamic Bayesian Networks

BNs are in their first formulation a static model. But since then, there have been a number of efforts for extending the networks' definition in order to make them able to represent time, for situations where, for example, the probability of a variable changing its state depends on the time or if the parameters of the system changes with time. In literature there are several definitions of Dynamic Bayesian Networks or Temporal Bayesian Networks or, in general, BNs that incorporate temporal features. The authors give similar names to different concepts and sometimes the same name for different definitions. There are two approaches that can be distinguished: in [36], the authors classify them considering whether they represent the time as points or instances or if they divide it in time intervals. In some approaches, Dynamic networks are seen as a general case of Temporal networks where the term dynamic refers to any change the system is subjected to as the change in state [37].

In the PhD thesis [38], the author considers an approach where the events considered occur at an instant in time but the network is studied evolving at different time slice. The author then modifies the definition to allow growing complexity in the networks but they become more difficult to apply to a large scale problem.

2.3 Converting FTs into BNs

In [39] the authors make a comparison between BNs and FTA techniques for dependability problems. They show how a FT can be mapped into a BN and that any analyses performed with the FT methods by means of the minimal cut sets procedure can be carried out in a BN. Furthermore, some new analysis are permitted in a BN, such as the calculation of the posterior probability of a subset of components given the fault. Therefore any FT corresponds to a BN and any techniques applied to a FT can be performed in a BN, but the latter allows some more modelling solutions [40]. These arguments will be presented with a simple example.

2.3.1 FT Conversion Methods

The operations of the algorithm to obtain a BN from a FT are given below. It is assumed that the FT will have only AND and OR gates, the resulting BN will by binary, the variables will represent states of the components of a system and the two values they can assume will be labelled with FALSE ($\bar{V}$) for the working state, and with TRUE (V) for the failing state. However the algorithm can be generalised to any FT.

Regarding the qualitative part of the BN:

1. any basic system component of the FT corresponds to a root node in the BN;
2. any gate of the FT corresponds to a node in the BN, in particular the gate whose output is the top event in the FT will be labelled as *fault* node;
3. the nodes in the BN has to be connected as the gates in the FT.

Figure 2.14 shows how the structure of a simple FT is converted into the structure of a BN.

Regarding the probability, the quantitative part of the BN:

1. to any root node in the BN it is assigned the same prior probability of its corresponding basic event in the FT;
2. to any node in the BN corresponding to a AND gate in the FT it is associated a CPT such that the node is TRUE with probability 1 if and only if all parents are TRUE;
3. to any node in the BN corresponding to a OR gate in the FT it is associated a CPT such that the node is TRUE with probability 1 if and only if at least one of its parents is TRUE.

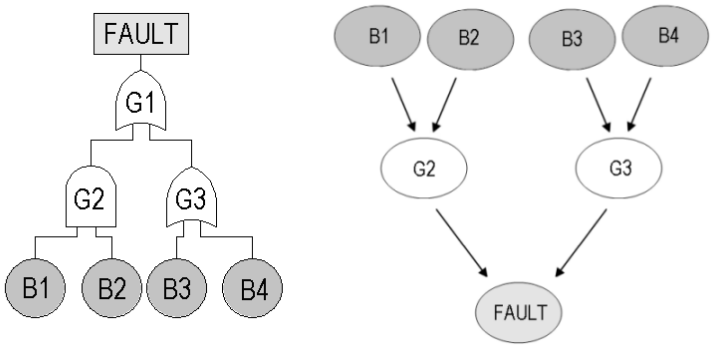


Figure 2.14 – Basic events $B1$, $B2$, $B3$ and $B4$ correspond to the homonym nodes; gates $G2$ and $G3$ correspond to nodes $G2$ and $G3$ and the top event corresponds to the node FAULT.

Parent Node(s)		G2		
B1	B2	Yes	No	bar charts
Yes	Yes	1.0	0.0	
	No	0.0	1.0	
No	Yes	0.0	1.0	
	No	0.0	1.0	

Figure 2.15 – CPT corresponding to an AND gate.

For an AND gate, such as G2 in figure 2.14, the CPT will result as in figure 2.15.
For an OR gate, such as G3 in figure 2.14, the CPT will be that in figure 2.16

Parent Node(s)		G3		
B3	B4	Yes	No	bar charts
Yes	Yes	1.0	0.0	
	No	1.0	0.0	
No	Yes	1.0	0.0	
	No	0.0	1.0	

Figure 2.16 – CPT corresponding to an OR gate.

The conversion method can be extended to FTs with other gates and the CPTs in the corresponding networks will follow the logic tables of the gates. For example, for the NOT gate in figure 2.17, the CPT will be as in figure 2.18.

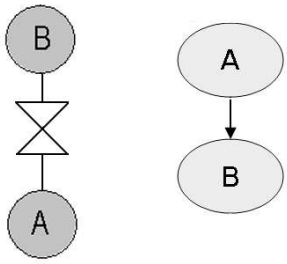


Figure 2.17 – Configuration corresponding to a NOT gate.

Parent Node(s)	B		
A	Yes	No	bar charts
Yes	0.0	1.0	
No	1.0	0.0	

Figure 2.18 – CPT corresponding to a NOT gate.

The *unavailability* of the top event in a FT corresponds to the prior probability of the node labelled as *fault* in the BN. The unavailability of a sub-system in a FT corresponds to the prior probability of the corresponding nodes in the BN. In a FT, these computations are obtained by means of the minimal cut sets, in a BN they can be obtained as $P(H \mid e)$ where H represents the *fault* (or the variables of the sub-system) and the evidence is the empty set, $e = \emptyset$. The posterior probability can be also computed in a BN and this can be considered for a single component, for a subset of components or for all components except the ones to which evidence has been assigned. When the fault is given as evidence, the posterior probability of each component gives the criticality of each of them and the posterior probability of a sub-system gives the criticality of the sub-system in causing the system failure.

2.3.2 Bayesian networks from Fault Trees with repeated events

When in a FT a basic event appears more than once, it is said that it has *repeated events*. FTs with repeated events can be mapped into BNs simply creating a single node for every basic event and linking with more than one link the nodes that correspond to the repeated events. In figure 2.19 the FT has the basic event A appearing twice as output of the gate G2 and G3.

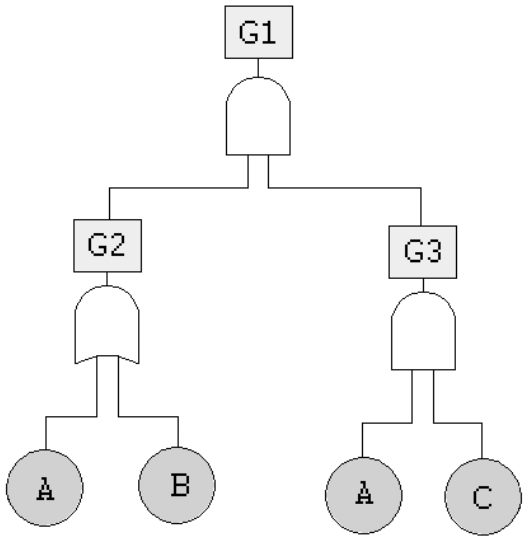


Figure 2.19 – Example of FT with repeated events.

The BN corresponding to the FT in figure 2.19 is shown in figure 2.20. Node A appears only once but is linked with two links to both node G2 and G3.

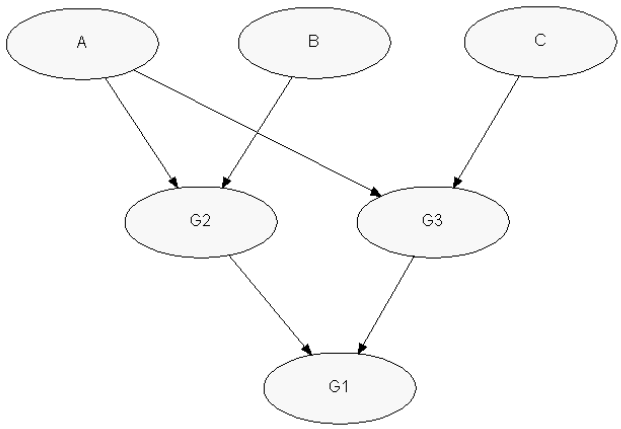


Figure 2.20 – BN corresponding to figure 2.19.

When a FT has many repeated events, its corresponding BN can assume a graphical structure where the links intersect themselves in a way that can make the visual understanding of the network more complicated. The same procedure can be applied for a FT with repeated branches. The corresponding BN will result in having nodes with more than one link as in figures 2.21 and 2.22.
The intersections of the links, especially for large networks, is a disadvantage of the graphical

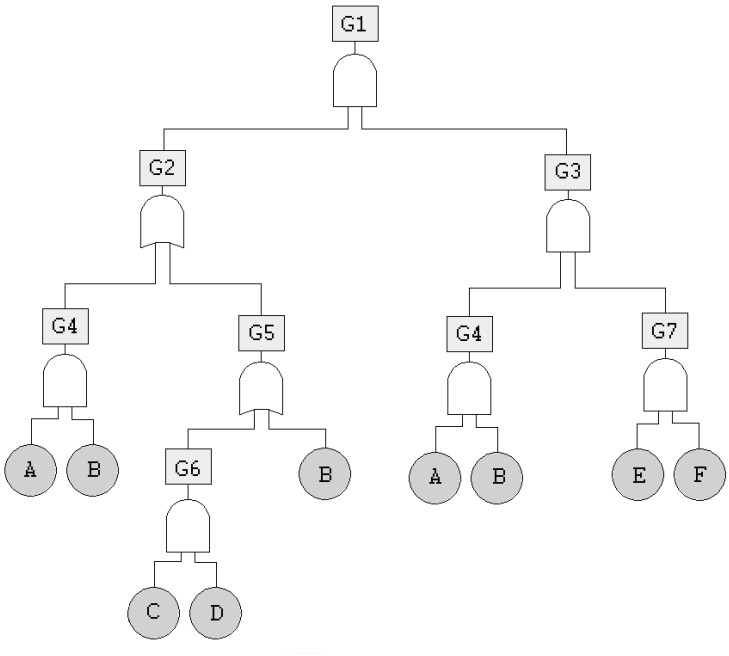


Figure 2.21 – Example of FT with repeated branches.

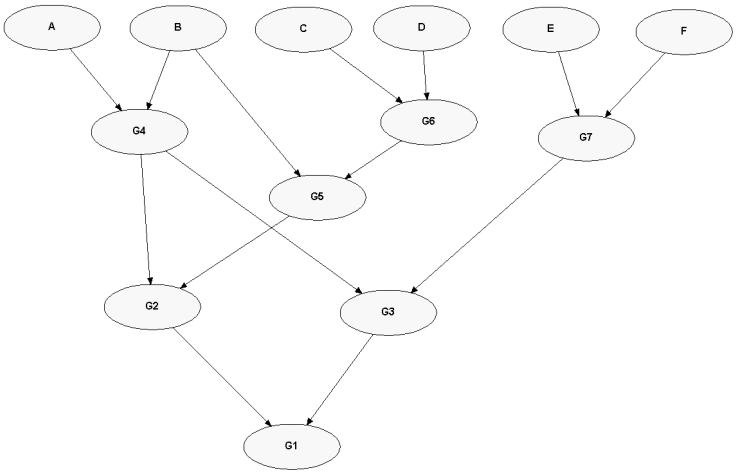


Figure 2.22 – Example of BN corresponding to the FT in figure 2.21.

representations of the BNs. In the following section, the conversion procedure from FTs to BNs is shown with the example of a pressure tank system.

2.3.3 Example: Pressure Tank System

The conversion algorithm and the probability computation will be discussed with the example of the pressure tank system, a system that discharges fluid from a reservoir into a pressure tank with a control system that regulates the operation of the pump. This example

is used only in this section for showing the conversion algorithm. Here is the description of the system [41]:

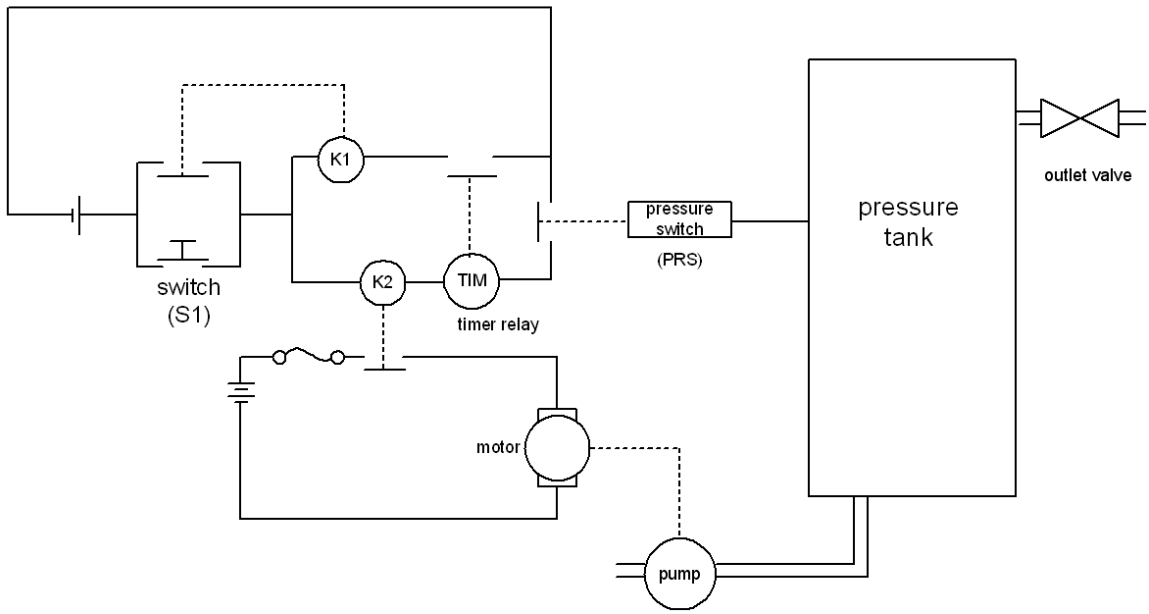


Figure 2.23 – Pressure Tank System.

The function of the control system is to regulate the operation of the pump. It is assumed that it takes 10 minutes to pressurise the tank. The pressure switch has contacts which are closed when the tank is empty. When the threshold pressure has been reached, the pressure switch contacts open, de-energising the relay K2 so that K2 contacts open, removing power from the pump motor to cease operation. The tank is fitted with an outlet valve which allows the tank contents to be used when required. When the tank is empty the pressure switch contacts close and the cycle repeats.

Initially the system is considered to be in its dormant (de-energised) mode: tank empty, switch S1 contacts open, relay K1 contacts open, timer relay (TIM) contacts closed, pressure switch contacts closed.

System operation is started by momentarily depressing switch S1. This applies power to the relay K1 closing K1 contacts so that K1 is now electrically self-latched. Switch S1 contacts open. The closure of K1 contacts allows power to the relay K2 whose contacts close and start the pump motor.

The timer relay (TIM) is provided as a safety shut-down mechanism in the event that the pressure switch contacts fail to open when the tank is full. Initially the timer contacts are closed when the power is applied to K2 and this starts a clock in the relay. When the timer contacts open this breaks the circuit to relay K1 whose contacts open removing power from K2 and stopping the pump motor. When the circuit with K2 and the timer relay is de-energised, this resets the timer relay clock to zero. When the system stops due to a safety shut-down it requires a manual restart.

The top event considered is *Pressure Tank Overfilled* and the component failure modes are

PRS Pressure switch fails to open

K2 Relay K2 contacts fail closed

K1 Relay K1 contacts fail closed

TIM Timer relay fails to time out

S1 Switch contacts fails closed

and they have the following failure rates:

PRS If the pressure switch contacts fail to open this failure will be revealed since it will result in either the pressure tank becoming over-pressurised or the timer contacts opening (which requires a manual restart). This event has a failure rate $\lambda = 1 \text{ x } 10^{-4}$ per hour. The failure event could occur anytime in the 10 minutes operational time.

K2 If K2 relay contacts fail to open then the tank will become over-pressurised and is hence a revealed failure. This failure has a rate of occurrence of $\lambda = 1 \text{ x } 10^{-2}$ per hour and could occur anytime in the 10 minutes operation time.

K1 If relay K1 contacts fail to open this failure will be unrevealed. Its rate of occurrence is $\lambda = 1 \text{ x } 10^{-3}$ per operation. These contact are inspected/tested at intervals of one year.

TIM The time contacts are a safety feature of the system. Its failure is therefore unrevealed and occurs with a rate $\lambda = 1 \text{ x } 10^{-4}$ per hour. This component is also inspected in intervals of one year.

S1 If the switch fails to open after it is initially closed, this alone will not cause any problems and hence it will not be revealed. This has a failure probability per operation of 0.01.

Figure 2.24 shows the FT relative to the system described above, its minimal cut sets are {K2}, {PRS, K1}, {PRS, TIM}, {PRS, S1}.

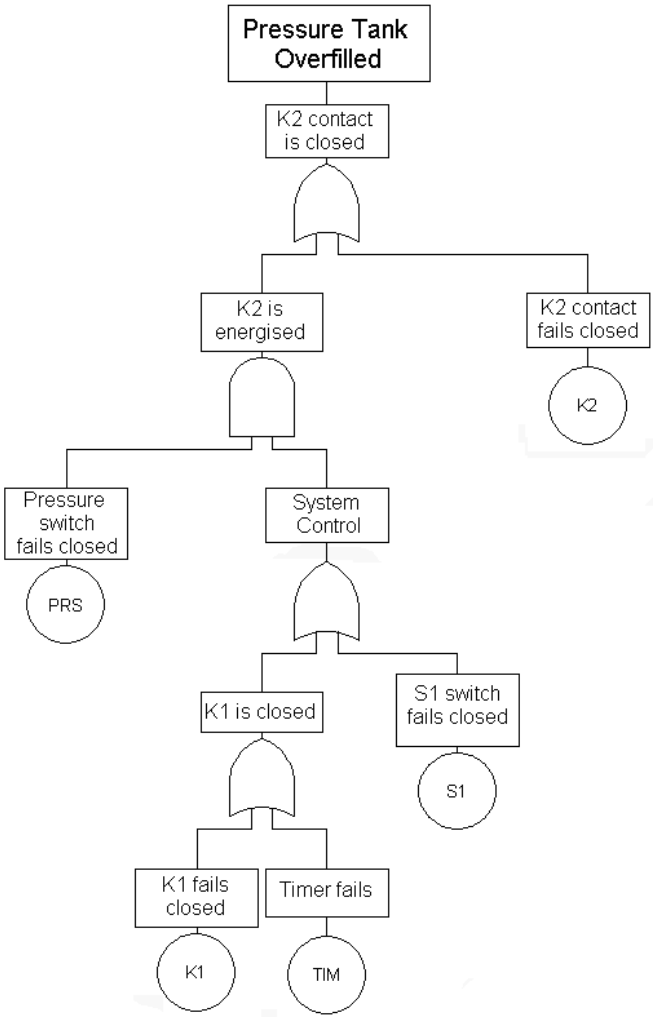


Figure 2.24 – FT for the Pressure Tank System.

Following steps $\{1-2-3\}$ in the conversion algorithm for the qualitative part of the FT, the structure of the BN will result as in figure 2.25. It can be seen that 5 root nodes have been created corresponding to the basic events S1, K1, K2, PRS and TIM. The two remaining gates *SystemControl* and $k2energised$ correspond to two nodes.
Node *SystemControl* corresponds to an OR gate, then its CPT will be as in figure 2.26 following the general rule given in figure 2.16.
Node $K2energised$ corresponds to an AND gate, so its CPT will be as in figure 2.27 following the general rule given in figure 2.15.

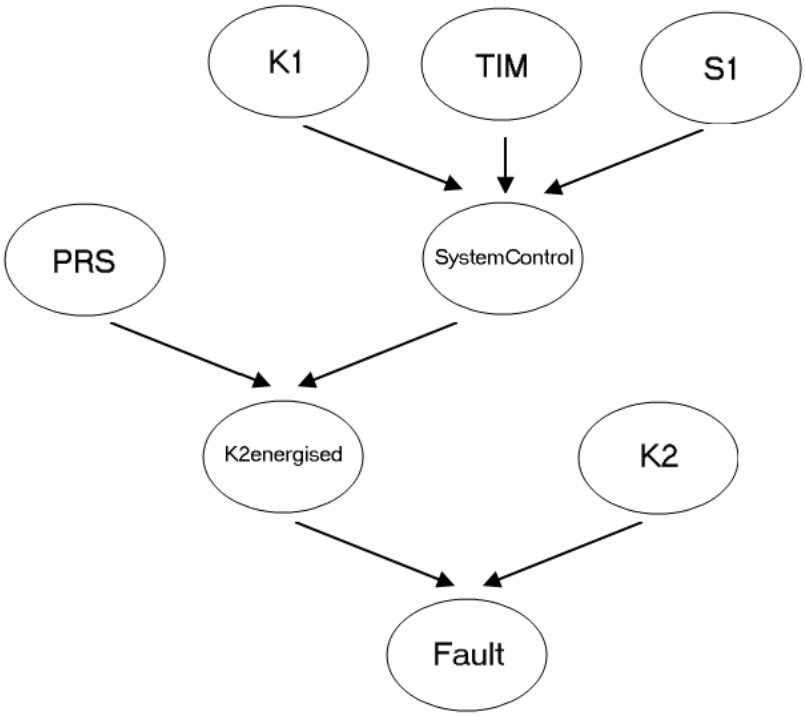


Figure 2.25 – BN for the Pressure Tank System.

Parent Node(s)			SystemControl		
K1	TIM	S1	Yes (fails)	No (works)	bar charts
Yes (fails)	Yes (fails)	Yes (fails)	1.0	0.0	
		No (works)	1.0	0.0	
	No (works)	Yes (fails)	1.0	0.0	
		No (works)	1.0	0.0	
No (works)	Yes (fails)	Yes (fails)	1.0	0.0	
		No (works)	1.0	0.0	
	No (works)	Yes (fails)	1.0	0.0	
		No (works)	0.0	1.0	

Figure 2.26 – CPT for the node *SystemControl*.

Parent Node(s)		k2energised		
PRS	SystemControl	Yes (fails)	No (works)	bar charts
Yes (fails)	Yes (fails)	1.0	0.0	
	No (works)	0.0	1.0	
No (works)	Yes (fails)	0.0	1.0	
	No (works)	0.0	1.0	

Figure 2.27 – CPT for the node *K2energised*.

Finally, the node *Fault* corresponds to an OR gate, so its CPT will be as in figure 2.28
The probability of failure P for the nodes without parents can be calculated from the failure

Parent Node(s)		Fault		
k2energised	K2	Yes (fails)	No (works)	bar charts
Yes (fails)	Yes (fails)	1.0	0.0	
	No (works)	1.0	0.0	
No (works)	Yes (fails)	1.0	0.0	
	No (works)	0.0	1.0	

Figure 2.28 – CPT for the node *Fault*.

rates, considering the operational time of 10 minutes. For the components whose failure is revealed, considering the approximation $P = \ 1 - e^{-\lambda t} \approx \lambda t$, the probabilities are:

- $P(PRS) = \frac{1}{6}$ x $10^{-4} = 0.16$ x $10^{-4} = 0.000016$,
- $P(K2) = 0.16$ x $10^{-2} = 0.001600$,
- $P(S1) = 0.010000$.

For the components with unrevealed failure K1 and TIM, inspected at intervals of one year, the following formula is used [42] :

$$P = 1 - \frac{1}{\lambda\theta}(1 - e^{-\lambda\theta}), \tag{2.31}$$

where θ represents the time between inspections in unit time. Here the unit time is the operational time of 10 minute. If the system has two operations per day, the probabilities are:

- $P(K1) = 1 - \frac{1}{10^{-3} \text{ x } 730}(1 - e^{(-10^{-3} \text{ x } 730)}) = 0.339851$,
- $P(TIM) = 1 - \frac{1}{10^{-4} \text{ x } 730}(1 - e^{(-10^{-4} \text{ x } 730)}) = 0.035628$,

where the value 730 is the number of operations in one year $\theta = 2 \cdot 365$ operations.

The structure function for the top event unreliability in the FT is

$$T = 1 - (1 - K2)(1 - PRS\ K1)(1 - PRS\ TIM)(1 - PRS\ S1),$$

an that, with the pivoting method, gives

$$= PRS[1 - (1 - K2)(1 - K1)(1 - TIM)(1 - S1)] + (1 - PRS)[1 - (1 - K2)].$$

Substituting the probabilities of the basic events gives for the top event probability:

$$Q = P(PRS)[1-(1-P(K2))(1-P(K1))(1-P(TIM))(1-P(S1))]+$$
$$+(1-P(PRS))[1-(1-P(K2))] = 0.001606.$$

The same result can be achieved for the BN calculating the prior probability of the node *fault* (denoted by F) marginalising over all the other variables:

$$P(F) =$$
$$= \sum_{k2energ} \sum_{K2} \sum_{PRS} \sum_{SysCont} \sum_{K1} \sum_{TIM} \sum_{S1} P(F, K2energ, K2, PRS, SysCont, K1, TIM, S1).$$

This can be done first calculating $P(SystemControl)$ as

$$P(SystemControl) = \sum_{K1} \sum_{TIM} \sum_{S1} P(SysCont, K1, TIM, S1),$$

then $P(K2energised)$ as

$$P(K2energised) = \sum_{PRS} \sum_{SysCont} P(PRS, SysCont, K2energ),$$

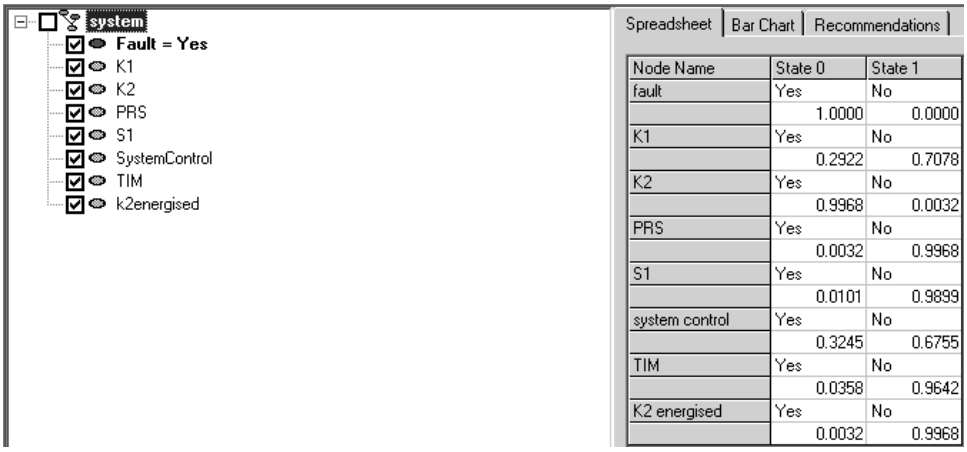
and finally

$$P(F) = \sum_{K2energ} \sum_{K2} P(F, K2energ, K2).$$

The result is shown in figure 2.29 and it has been calculated with MSBNx, a free BN Editor and Toolkit from Microsoft (see Appendix C).

Figure 2.29 – Prior Probability of the node Fault in the BN.

The FT and BN calculations lead to the same results. Assuming that the system is faulty, that is, giving evidence to the node *fault*, the posterior probabilities of the single components are calculated and shown in figure 2.30.



Node Name	State 0	State 1
fault	Yes	No
	1.0000	0.0000
K1	Yes	No
	0.2922	0.7078
K2	Yes	No
	0.9968	0.0032
PRS	Yes	No
	0.0032	0.9968
S1	Yes	No
	0.0101	0.9899
system control	Yes	No
	0.3245	0.6755
TIM	Yes	No
	0.0358	0.9642
K2 energised	Yes	No
	0.0032	0.9968

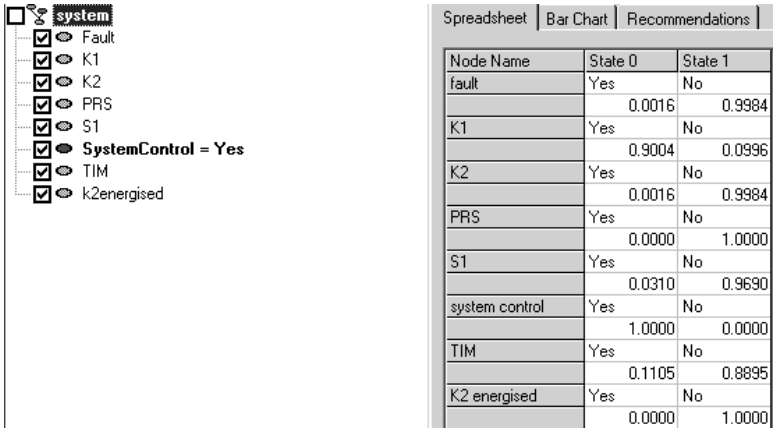
Figure 2.30 – Posterior probabilities of the components, given that the system has failed.

These are obtained by equation 2.13. For example, for node $K1$, this becomes

$$P(K1 = Yes \mid F = Yes) =$$

$$= \frac{\sum_{k2energ} \sum_{K2} \sum_{PRS} \sum_{SysCont} \sum_{TIM} \sum_{S1} P(F = Y, K2energ, K2, PRS, SysCont, K1 = Y, TIM, S1)}{\sum_{k2energ} \sum_{K2} \sum_{PRS} \sum_{SysCont} \sum_{K1} \sum_{TIM} \sum_{S1} P(F = Y, K2energ, K2, PRS, SysCont, K1, TIM, S1)}.$$

Component $K2$ appears to be the one which has the higher probability to have caused the failure of the system. Apart from the top event, other types of evidence can be introduced. An example is shown in figure 2.31, where all posterior probabilities are considered given that the control system has failed.



Node Name	State 0	State 1
fault	Yes	No
	0.0016	0.9984
K1	Yes	No
	0.9004	0.0996
K2	Yes	No
	0.0016	0.9984
PRS	Yes	No
	0.0000	1.0000
S1	Yes	No
	0.0310	0.9690
system control	Yes	No
	1.0000	0.0000
TIM	Yes	No
	0.1105	0.8895
K2 energised	Yes	No
	0.0000	1.0000

Figure 2.31 – Posterior probabilities when the evidence about the system control is given.

In this case, K1 is the component that has the highest probability to have caused the failure of the control system.

Even though it is not possible to update probability following evidence with FTA, importance measures are used in order to obtain information on the criticality of the components of the system. In the next section, the principal importance measures are defined.

2.4 Importance Measures

The *Importance Measure* is a very effective means to evaluate the role of the components in contributing to the occurrence of the fault on the system. Assigning a numerical value to each component allows them to be ranked according to their criticality with respect to the top event and it can help identifying potentially weak areas of the system. They can be *probabilistic* or *deterministic* depending on whether they involve the component's probability or not. In the following subsections three of the most important probabilistic measures will be discussed [42]. For the purpose of the BN method developed in the thesis they are not used, however it can be useful to see how these are calculated to compare the FTA and BN methods.

2.4.1 Birnbaum's Measure of Importance

Given a system with n components, a *critical state* of the system for component i is a state of the remaining $n-1$ components such that the failure of component i causes the system to pass from the working to the failing state.

Birnbaum's measure of importance of a component i, denoted by $G_i(q)$, is defined as the probability of the system to be in a critical state for component i [43]. This represents the maximum increase in risk when component i is failed compared to when component i is working. It can be calculated as the sum of the probabilities for the system of being in its critical states for component i. An expression for $G_i(\boldsymbol{q})$ can also be given in terms of conditional probability. If $Q(1_i, \boldsymbol{q})$ denotes the probability that the system fails given that component i has failed and $Q(0_i, \boldsymbol{q})$ is the probability that the system fails with component i working, then:

$$G_i(\boldsymbol{q}) = Q(1_i, \boldsymbol{q}) - Q(0_i, \boldsymbol{q}). \tag{2.32}$$

Birnbaum's importance measure is used to define other measures such as the *criticality measure* and it can be useful in calculating several system performance measures such as system failure frequency. In fact the frequency of failure of the system can be calculated in term of Birnbaum's measure and failure frequency of the components as follows:

$$w_{sys} = \sum_{i=1}^{n} G_i w_i. \tag{2.33}$$

2.4.2 Criticality Measure of Importance

Criticality measure of importance of a component i is denoted by I_i and it is defined as the probability that the system is in a critical state for component i considering the probability of failure of component i, $q_i(t)$, weighted by the system unavailability $Q_{sys}(\boldsymbol{q}(t))$ [43]:

$$I_i = \frac{G_i(\boldsymbol{q}) q_i(t)}{Q_{sys}(\boldsymbol{q}(t))}. \tag{2.34}$$

Given the occurrence of the top event, this measures determines whether the failure of the system is a result of the failure of the component. The criticality measure modifies the Birnbaum measure by considering the probability of the component failure. In this way it is possible to avoid assigning high importance measures to events that are very unlikely to occur focusing on the truly important basic events. The Criticality measure is therefore appropriate to improve system performance.

2.4.3 Fussel-Vesely Measure of Importance

Fussel-Vesely measure of importance is constructed considering minimal cut sets. It is defined as the probability of the union of the minimal cut sets that contain component i given that the system has failed:

$$F_i = \frac{P(\bigcup_{k|i\in k} C_k)}{Q_{sys}(\boldsymbol{q}(t))}. \tag{2.35}$$

The Fussel-Vesely measure determines the probability that component i has contributed to the system failure.

2.4.4 Importance Measures in Bayesian Networks

It has been seen that Birnbaum's measure can be calculated in terms of posterior probability by equation 2.32 and, as a consequence, the Criticality measure can be obtained in terms of posterior probability as well. It is therefore always possible to calculate the first two importance measures examined for any node i with states *yes* and *no* of a BN. Birnbaum measure is given by:

$$G_i = P(fault = yes \mid i = yes) - P(fault = yes \mid i = no), \tag{2.36}$$

and, as a result, the Criticality measure is given by:

$$I_i = \frac{P(fault = yes \mid i = yes) - P(fault = yes \mid i = no) \quad P(i)}{P(fault)}. \tag{2.37}$$

Regarding the Fussel-Vesely measure, this is not obtainable directly from the BN as it employs the minimal cut sets and these are not derivable from the network itself. In the next section, the example of a system with its FT model and its BN will be discussed. The importance measures of the components will be calculated from the network and they will be compared with the posterior probability.

2.5 The Firewater Deluge System

In this section it is considered a simplification of the Firewater Deluge System described in [44]. Figure 2.32 shows a diagrammatic overview of the system and the following subsection gives a description of its functioning and of the components' failure modes and failure rates.

2.5.1 Description

The function of the deluge system is to supply, on demand, water at a controlled pressure to a specific area on the platform protected by the system. As such, the Firewater Deluge System (FDS) comprises of a deluge skid, firewater pumps, associated equipment and ring mains.

The deluge valve set comprises three main elements: the main distribution line, a water closing circuit and a control air circuit. Upon receipt of a signal from the Main Fire and Gas Panel (MFGP), the solenoid valves (SV1 and SV2) are de-energised and open thus releasing air pressure from the control air circuit. The air pressure drop allows the valmatic release valve (MRM) to open, and water from the water closing circuit (WVR) runs to drain. This causes the pressure on the deluge valve diaphragm (WV) to fall. When the pressure on the diaphragm has fallen sufficiently, the firewater main pressure acting on the underside of the deluge valve overcomes the load imposed by the diaphragm, allowing the flow into the distribution pipes onto the hazard.

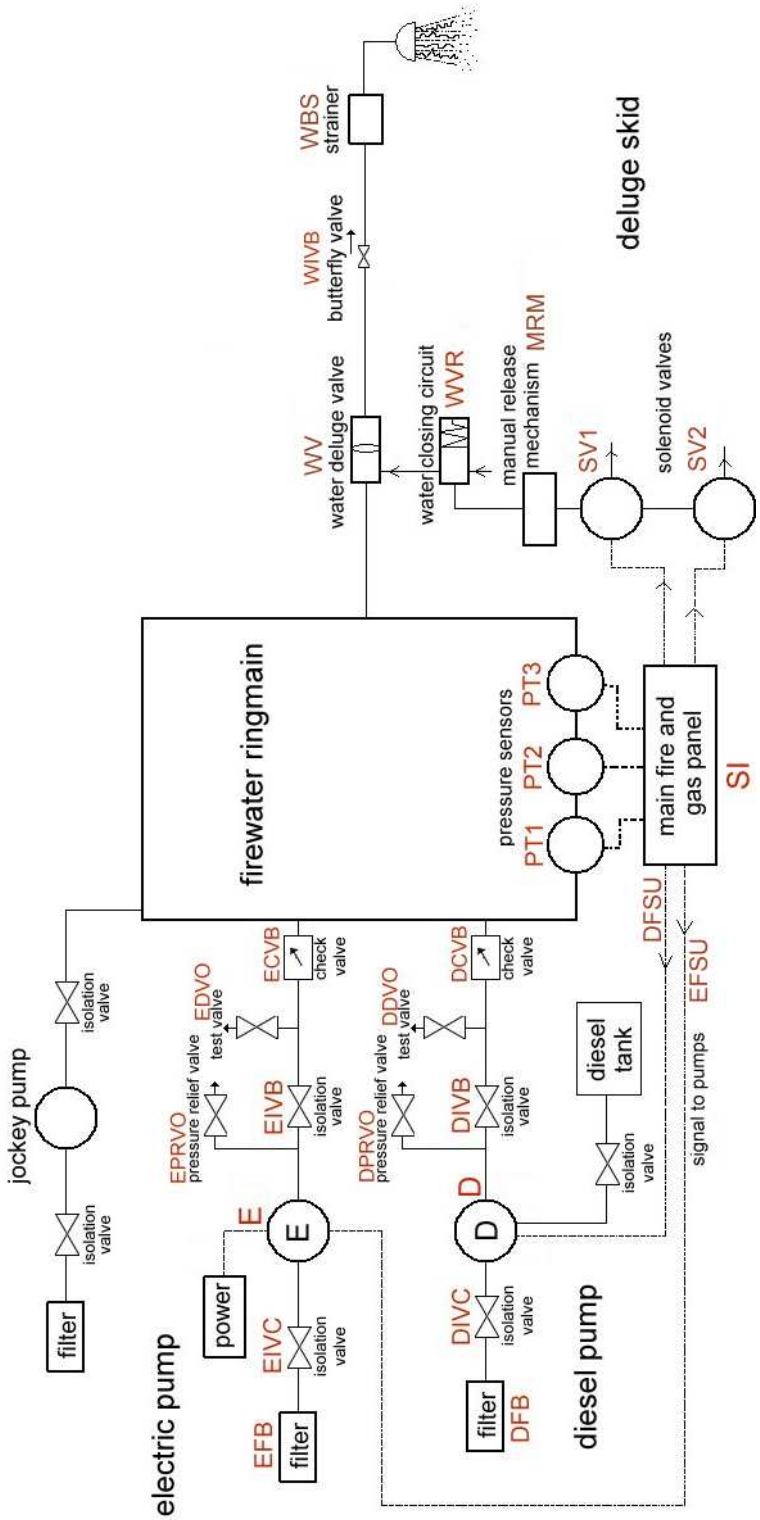


Figure 2.32 – Firewater deluge system.

The system may also be operated manually by opening the system local manual release valve (MRM) on the skid. This allows air to escape from the control air circuit and the system operates as described above.

The deluge systems are connected to a pressurised ring main network. The ring main pressure is maintained by a jockey pump drawing water from the sea. Falling pressure is detected by three pressure sensors (PT1, PT2 and PT3), which subsequently send a signal to the MFGP. When low pressure is detected by at least two of the three sensors, the MFGP activates the firewater pumps to supply water direct from the sea at sufficient pressure to meet the deluge requirements. There are two pumps, one (E) being powered from the main electric power plant and the other (D) from a diesel engine. The diesel pump is supplied by a tank. The following failures rates values can be found in [44].

EVENT DESCRIPTION and FAILURE RATES

SI Failure of MFGP to correctly select and send a close signal to the solenoid valves; $\lambda = 2\text{x } 10^{-7}$ per hr.

WBS Strainer, located upstream of the water deluge valve, blocked; $\lambda = 2.8\text{x } 10^{-5}$ per hr .

WIVB Blockage of the locked open butterfly valve located upstream of the water deluge valve; $\lambda = 1.8\text{x } 10^{-6}$ per hr.

WV Water deluge valve fails to open; $\lambda = 4\text{x } 10^{-5}$ per hr.

MRM Manual release mechanism fails to dump instrument air; $\lambda = 1\text{x } 10^{-5}$ per hr.

SV1 - SV2 Solenoid activated valves fails to dump instrument air on receipt of the signal from the MFGP; $\lambda_{\text{SV1}} = 3\text{x } 10^{-6}$ per hr, $\lambda_{\text{SV2}} = 2\text{x } 10^{-5}$ per hr.

WVR Valmatic relief valve sticks closed on activation; $\lambda = 5\text{x } 10^{-6}$ per hr.

EFB - DFB The pump, which includes seawater filter, is blocked by debris; $\lambda = 2.8\text{x } 10^{-5}$ per hr.

EIVB - DIVB Firewater pump isolation valve being blocked. The butterfly isolation valve operates on the header from pump to ring main; $\lambda = 1.8 \text{ x } 10^{-5}$ per hr.

EIVC - DIVC The firewater pump isolation valve is left closed after pump test; $Q = 0.01$.

EPRVO - DPRVO Pressure relief valve on header from pump to ring main fails open; $\lambda = 1.2\text{x } 10^{-5}$ per hr.

EDVO - DDVO Test line, used to dump flow from firewater pumps overboard during test, is left open after completion; $Q = 0.01$.

ECVB - DCVB Check valve on header between the pump and ring main blocked; $\lambda =$ 2.5x 10^{-5} per hr.

E Failure of the electric pump; $\lambda =$ 5x 10^{-4} per hr.

D Failure of the diesel pump; $\lambda =$ 5x 10^{-4} per hr.

EFSU - DFSU Failure of fire pump selector unit to initiate start of the pump on detection of failure to restore ring main pressure; $\lambda =$ 8x 10^{-6} per hr.

PT1 - PT2 - PT3 Failure of ring main low pressure sensors to indicate low ring main pressure; $\lambda_{\text{PT1}} =$ 7x 10^{-6} per hr, $\lambda_{\text{PT2}} =$ 1.4x 10^{-5} per hr, $\lambda_{\text{PT3}} =$ 2.1x 10^{-5} per hr.

The failures are supposed to be unrevealed and inspected at intervals of six months. The unavailability of the components is given by equation 2.31 considering the operational time of 12 hours:

UNAVAILABILITIES

SI $q =$ 3.599x 10^{-5}; **WBS** $q =$ 5.0231x 10^{-3} ; **WIVB** $q =$ 3.2393x 10^{-4} ;
WV $q =$ 7.16556x 10^{-3}; **MRM** $q =$ 1.79784x 10^{-3} ;
SV1 - SV2 $q_{\text{SV1}} =$ 5.398x 10^{-4}, $q_{\text{SV2}} =$ 3.59137x 10^{-3}; **WVR** $q =$ 8.9946x 10^{-4};
EFB - DFB $q =$ 5.0231x 10^{-3} ; **EIVB - DIVB** $q =$ 3.23301x 10^{-3};
EIVC - DIVC $q =$ 0.01; **EPRVO - DPRVO** $q =$ 2.15389x 10^{-3} ;
EDVO - DDVO $q =$ 0.01; **ECVB - DCVB** $q =$ 4.48653x 10^{-3}; **E** $q =$ 8.48345x 10^{-2} ;
D $q =$ 8.48345x 10^{-2}; **EFSU - DFSU** $q =$ 1.43861x 10^{-3};
PT1 - PT2 - PT3 $q_{\text{PT1}} =$ 1.25894x 10^{-3}, $q_{\text{PT2}} =$ 2.51577x 10^{-3}, $q_{\text{PT3}} =$ 3.77049x 10^{-3} .

2.5.2 Fault Tree model

Figures 2.33, 2.34 and 2.35 show the FTs for the top event *Firewater system fails to protect.*

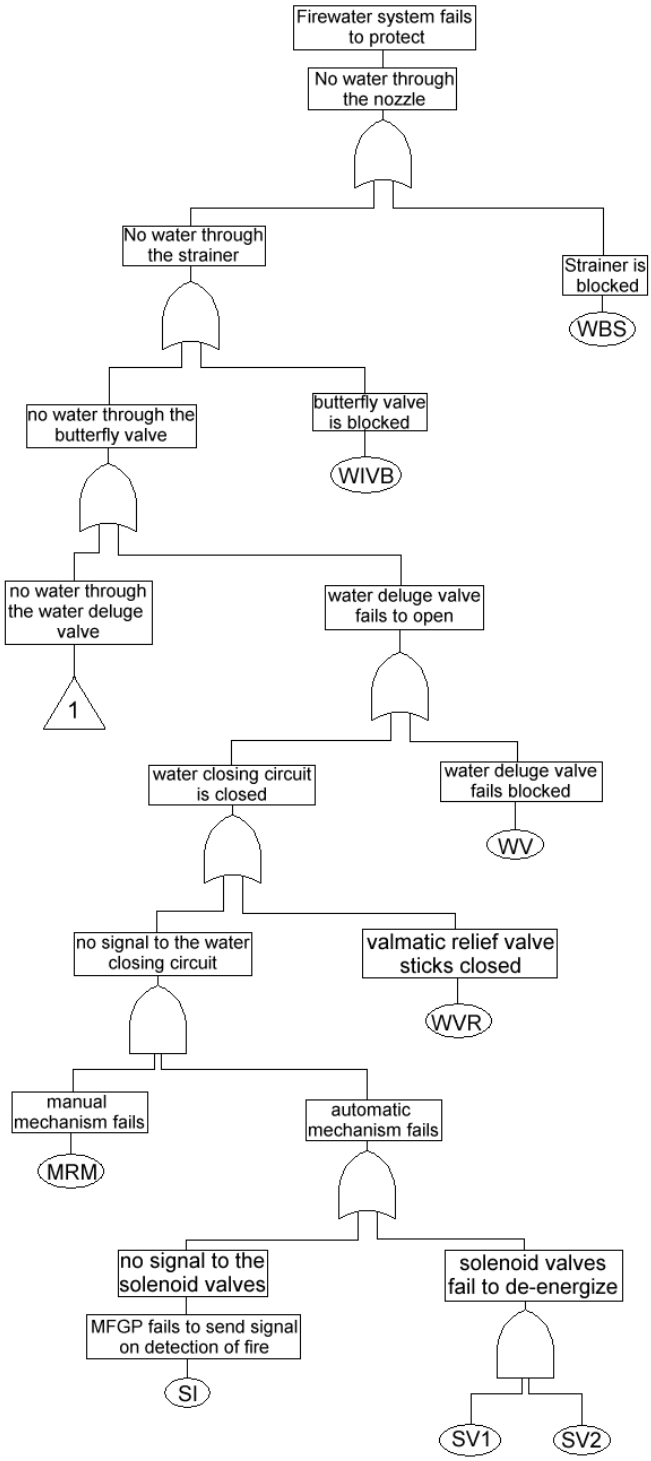


Figure 2.33 – Firewater deluge system FT (1 of 3).

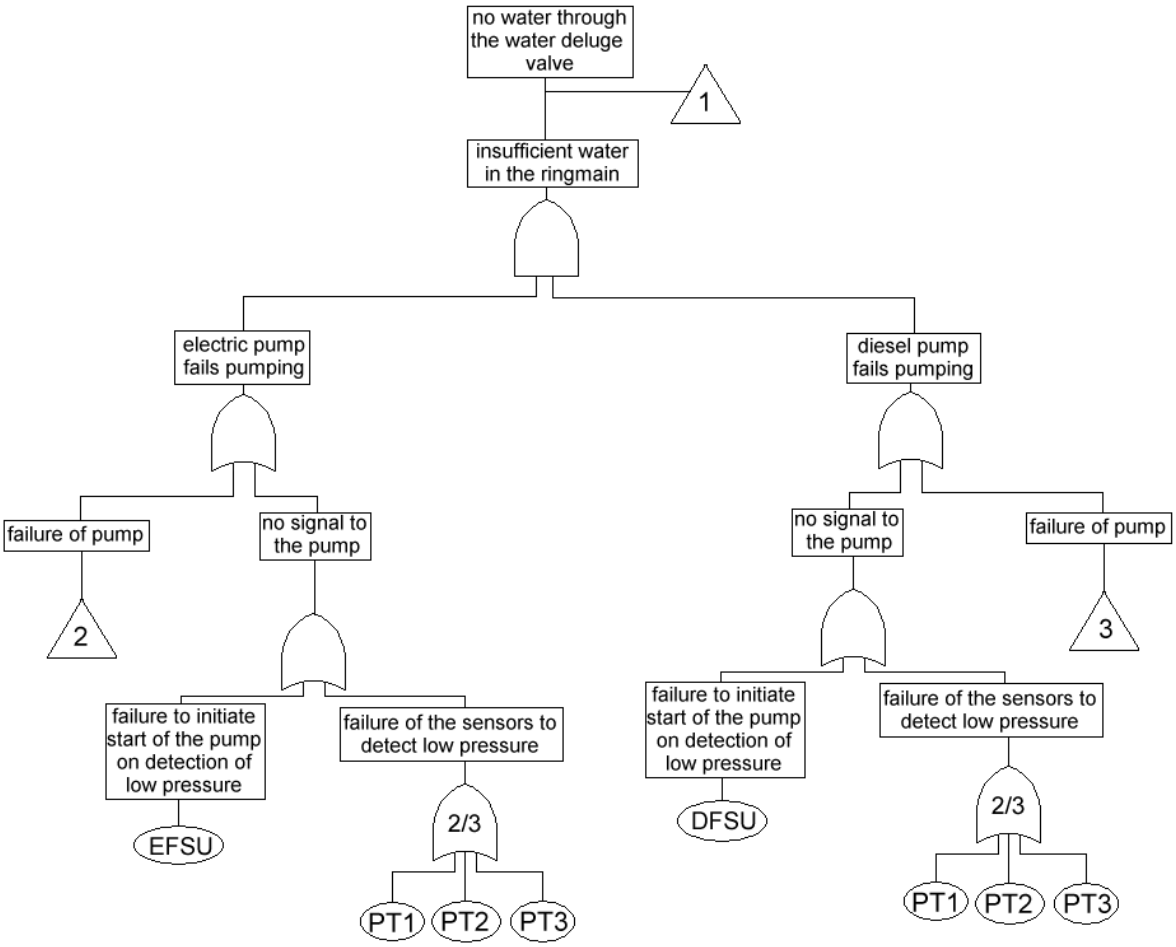


Figure 2.34 – Firewater deluge system FT (2 of 3).

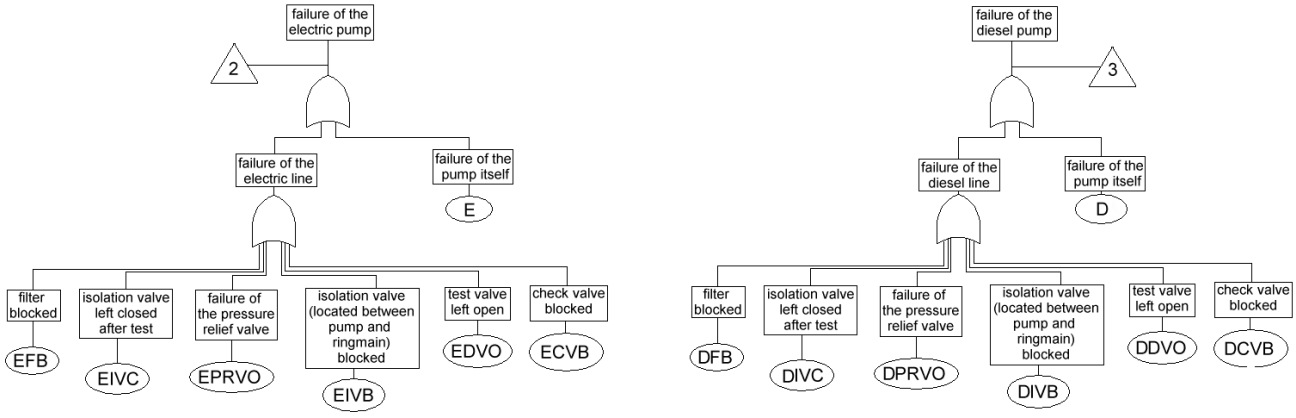


Figure 2.35 – Firewater deluge system FT (3 of 3).

MINIMAL CUT SETS

The minimal cut sets for the FT are 73, of which, one cut set of order 3, 4 cut sets of order 1 and 69 of order 2.
The top event probability has been obtained by the software *Fault Tree Plus* (see Appendix C):

$$Q = 0.027025,$$

as shown in figure 2.36.

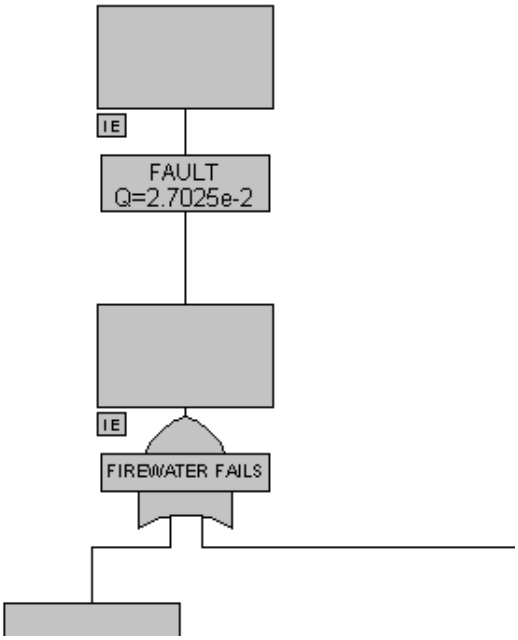


Figure 2.36 – Top event probability (*Fault Tree Plus*).

2.5.3 Bayesian model

A BN for the system has been derived as shown in figure 2.37.

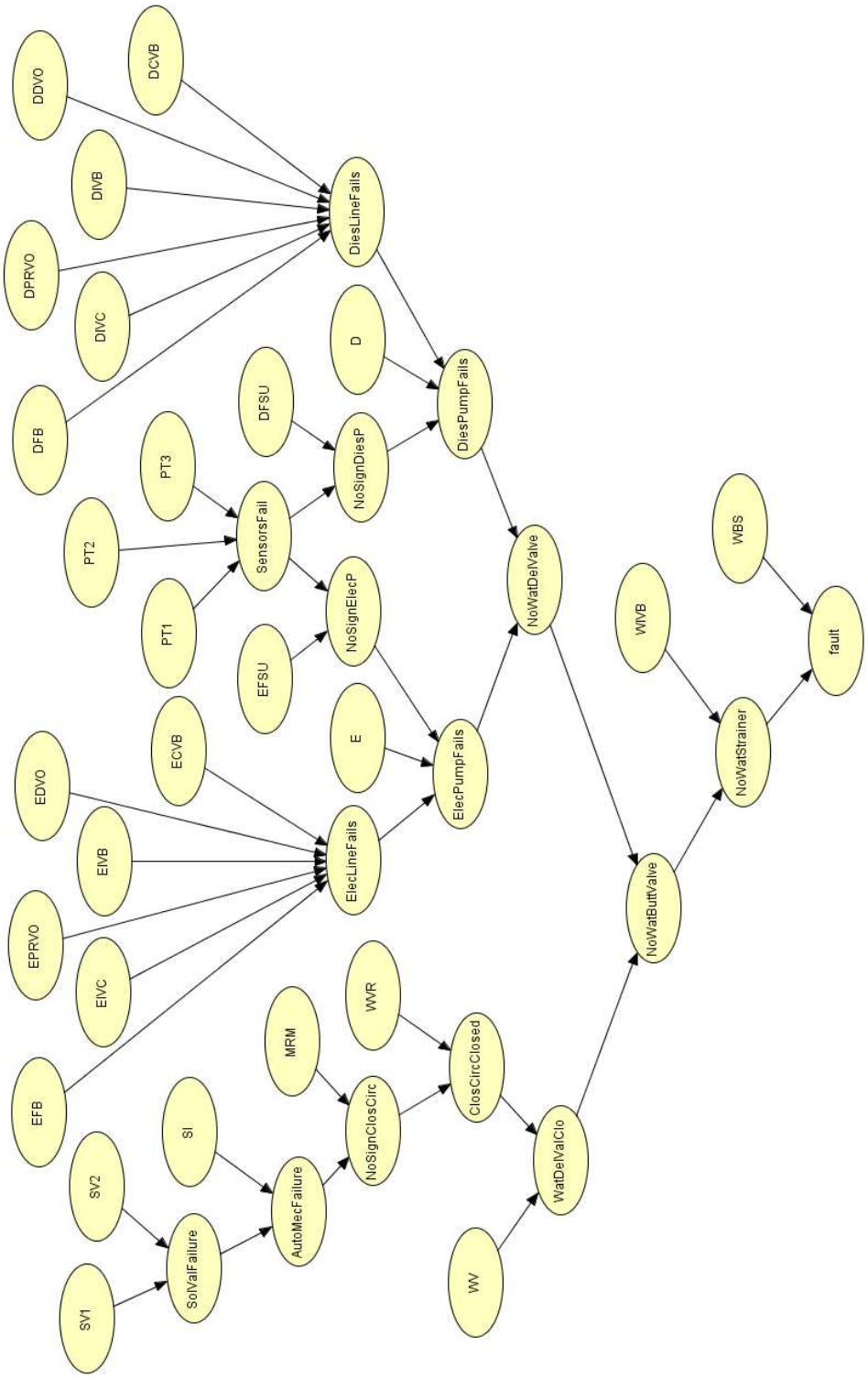


Figure 2.37 – BN for the Deluge System (*Hugin* software).

The probability of the node *fault* equals the top event probability in the FT, as shown in figure 2.38:

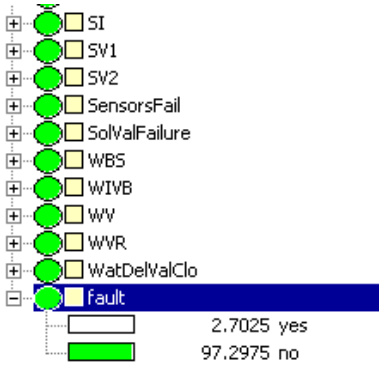


Figure 2.38 – Probability of Fault (*Hugin*).

2.5.4 Importance Measures analysis

Table 2.1 relates the posterior probability, the unavailability, the Birnbaum's measure, the Criticality measure and the failure rate of the components of the system.

Component	Posterior	Q	Birnbaum	Criticality	λ
E	0.4062448	0.0848345	0.11188045	0.351204606	0.0005
D	0.4062448	0.0848345	0.11188045	0.351204606	0.0005
WV	0.26514521	0.00716556	0.97999717	0.259841463	0.00004
WBS	0.18586826	0.0050231	0.97788696	0.181758183	0.000028
PT2	0.00296931	0.251577	0.00488436	0.045468654	0.000014
EIVC - DIVC	0.04788674	0.01	0.10342339	0.038269454	5.59295E-05
EDVO - DDVO	0.04788674	0.01	0.10342339	0.038269454	5.59295E-05
WVR	0.03328244	0.00089946	0.97385088	0.032412148	0.000005
EFB - DEF	0.02405399	0.0050231	0.10290607	0.019126976	0.000028
ECVB - DCVB	0.02148454	0.00448653	0.1028506	0.017074614	0.000025
EIVB - DIVB	0.01548183	0.00323301	0.10272126	0.012288557	0.000018
WIVB	0.01198629	0.00032393	0.97329022	0.011666136	0.0000018
EPRVO - DPRVO	0.01032864	0.00215689	0.10261048	0.008189421	0.000012
EFSU - DFSU	0.00688904	0.00143861	0.10253667	0.00545828	0.000008
PT3	0.00428012	0.00377049	0.00366666	0.000511566	0.000021
PT1	0.00154265	0.00125894	0.00609802	0.000284071	0.000007
MRM	0.00180029	0.00179784	0.0000369	2.45477E-06	0.00001
SI	0.00003832	0.00003599	0.00174925	2.32952E-06	0.0000002
SV1	0.00053993	0.0005398	0.00000628	1.25437E-07	0.000003
SV2	0.00359149	0.00359137	0.00000094	1.24917E-07	0.00002

Table 2.1 – Posterior probabilities and importance measures for the components of the firewater deluge system.

It has been observed that the criticality measure and the posterior probability follow a similar trend. This has to be expected as they both determine whether the failure of the system is a consequence of the failure of the component. Figure 2.39 compares the values of the posterior probability and criticality measure for the components of the system.
Similar results can be observed for the component measures of the pressure tank system analysed in section 2.3.3.

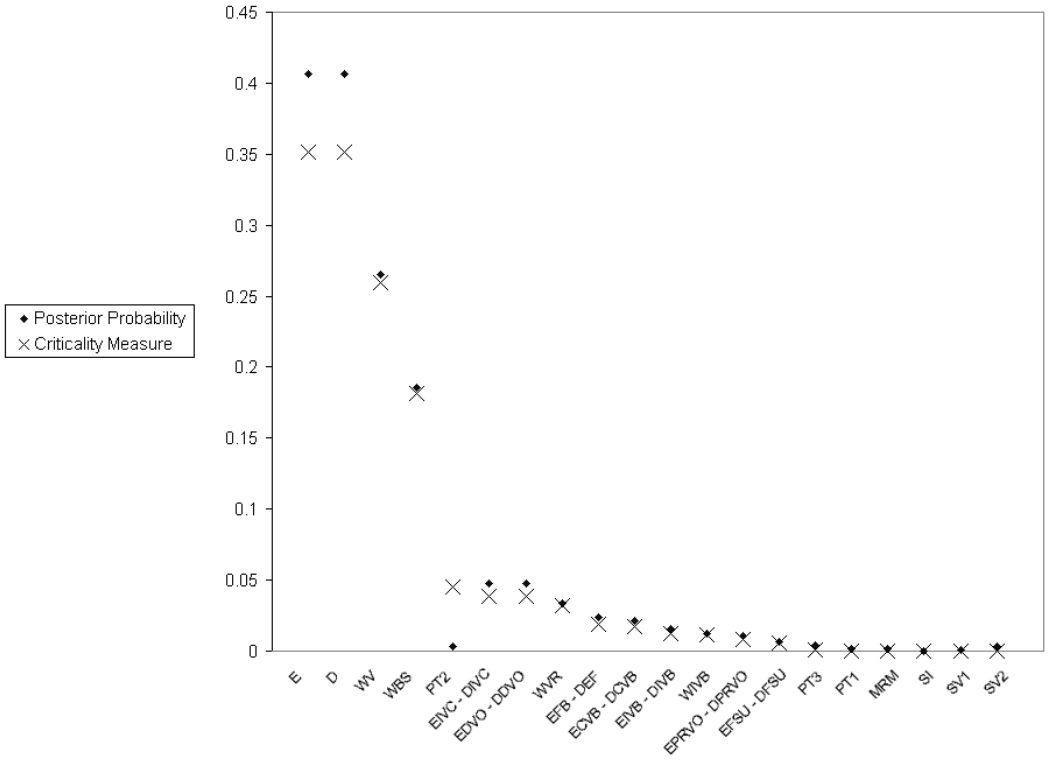


Figure 2.39 – Criticality measure and Posterior probability of the firewater deluge system components.

2.6 Summary

BNs represent an efficient modelling tool for reliability analysis as capable of performing the same analysis as the FTA. In particular it has been seen how a FT can be mapped into a BN and how this technique can provide the same information as the probability of failure of the system, the Birnbaum measure and the criticality measure. Furthermore, by BN modelling it is possible to calculate the posterior probability of the nodes of the network. This gives a measure of the criticality of the component with respect to the occurrence of the Fault node (or the top event) and it has being found to have a trend which is very similar to the criticality measure. The posterior probability, although, has the advantage to allow evidence to be introduced to more than one variable. For example, in a fault diagnostics procedure, when the system is inspected, evidence could be introduced to the components that are found in the working state. In this way, an updated probability is obtained for any configuration of the system. Furthermore, the analysis by means of BNs is immediate, while with FTA each scenario is studied by first calculating the prime implicants and then the criticality of the components. The comparison of the two techniques leads to the conclusion that they provide a similar analysis but BNs could bring more modelling solutions in processes like fault diagnostics as they are able to update the probability when evidence is introduced.

Furthermore, BNs seem to give a more concise representation, when for example, derived from FTs with repeated events. Given this, research has been directed toward the use of BNs in fault diagnostics.

Chapter 3

An Application of Bayesian Networks for System Fault Diagnostics

Introduction

FTA has been applied successfully to system fault diagnostics in a number of methods over the past years. One of these applications has been studied in [14], where different schemes are analysed for detecting faults and combinations of faults in a simple water tank level control system.

In [39], the authors make a comparison between BNs and FTA techniques for dependability problems. In particular, they show how a FT can be mapped into a BN and that any analysis performed with the FT methods by means of the minimal cut sets procedure can be carried out in a BN. We have shown the conversion procedure and an example in chapter 2.

In this chapter it is investigated how BNs can be applied to fault detection with a similar approach to the one mentioned above that made use of FTA. A method has been obtained converting the FTs into BNs. The choice to obtain the networks from the FTs is motivated by the fact that there is not a general structured way to build a BN, while FTA provides a method to create a FT to model an event in terms of its direct causes by introducing logic gates and basic events. In this chapter we show how the two methods obtained give similar results, but that using BNs has several advantages compared with FTA: the graphical representation is more concise and, for the purpose of fault detection, BNs give a straightforward approach when it comes to identifying the component failures by using posterior probability introducing evidence in the network.

The methods are demonstrated and compared using a simple water tank system. The commercial software package *Hugin Researcher* has been used for the creation and the evaluation of the networks (see Appendix C).

3.1 The Water Tank System

This section describes the operation of a simple water tank system illustrated in figure 3.1. This system will be used for the fault diagnostics in this and in the following chapter. Although the system functioning is simple, it still poses problems in terms of some dynamic aspects. The aim of the system is to maintain the level of the water in the tank between two fixed limits.

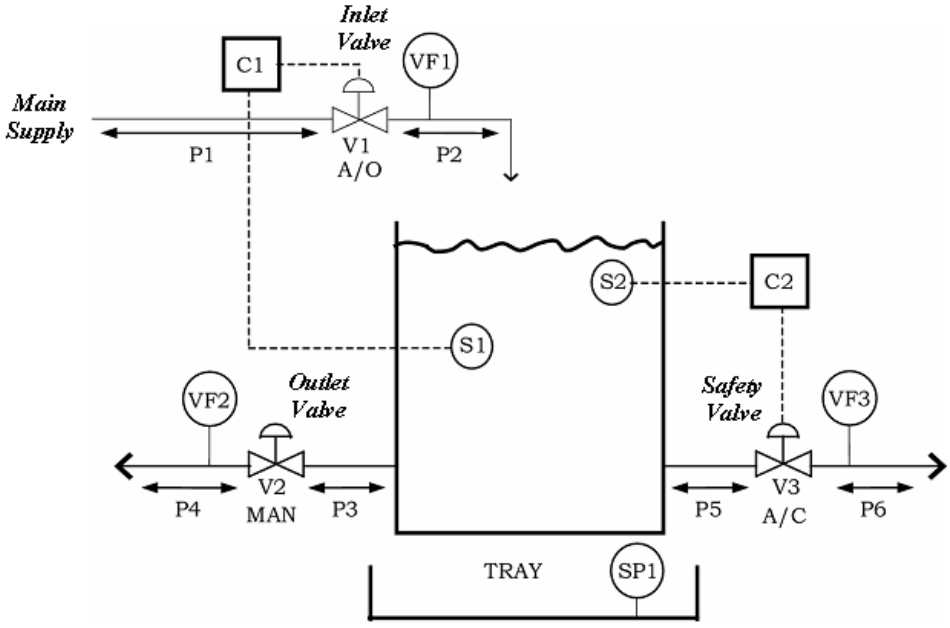


Figure 3.1 – Water Tank System

3.1.1 System Component Description

The systems consists of a number of components:
- the tank and the overspill tray denoted respectively TANK and TRAY;
The purpose of the tray is to collect any water leaking through a fracture or overflowing from the top, this is located underneath the tank.
- valves V1, V2 and V3;
Valve V1 is an air-to-open inlet valve located at the top of the tank. The valve is open or

closed depending on the level of the water detected by level switch S1. When S1 indicates that the level is below the required level, controller C1 opens V1 to allow water into the tank. When the level is above the limit, C1 closes V1 to stop the flow into the tank.
Valve V2 is a manual outlet valve and it is activated in response to demand by an operator. Valve V3 is an air-to-close valve that it is activated by controller C2 as a safety measure in case of a failure that causes the level of water to rise up to a risky level detected by the level switch S2. In this circumstance, controller C2 opens V3 to let the water flow out of the tank in order to reduce the level of the water in the tank.
- level switches S1 and S2 measure the level of the water in the tank and they are connected respectively to controllers C1 and C2.
- pipes make up sections P1 - P6.

When the system is operating normally, valve V2 is open letting water out of the tank and V1 is open to replace the water from V2. The water level is kept constant. Valve V3 should be closed as it only opens if a critical level is reached due to a failure in the system.

3.1.2 System Operating Modes and Scenarios

There are four sensors that monitor the functioning of the system. Three of them, VF1, VF2 and VF3, are located next to the valves V1, V2 and V3, respectively. These detect water flow through the valves, therefore their readings will be either *Flow* (F) or *No Flow* (NF). Another sensor, SP1, is located in the overspill tray and it detects potential presence of water leaked or overflowed from the tank. Its readings will be *Water* (W) or *No Water* (NW). All sensors are assumed to be perfectly reliable.

Since each of the four sensors has two different outcomes, there will be 16 possible different scenarios of the system resulting from any possible combination of sensor outcomes. These are listed in table 3.1.

The system has two operating modes: ACTIVE and DORMANT. In the ACTIVE mode valve V2 is open letting the water out of the system while valve V1 is open to let the water drain from the tank. Valve V3 should be closed and there should be no water in the overspill tray. The sensor readings correspond to scenario 4 in table 3.2. In the DORMANT mode the system is in standby with all valves closed and no water in the tray, as in scenario 16. When the tank is in the ACTIVE operating mode and the sensor readings differ from the ones in scenario 4 , then a deviation (fault) is expected from the normal functioning of the

Scenario	VF1	VF2	VF3	SP1
1	F	F	F	W
2	F	F	F	NW
3	F	F	NF	W
4	F	F	NF	NW
5	F	NF	F	W
6	F	NF	F	NW
7	F	NF	NF	W
8	F	NF	NF	NW
9	NF	F	F	W
10	NF	F	F	NW
11	NF	F	NF	W
12	NF	F	NF	NW
13	NF	NF	F	W
14	NF	NF	F	NW
15	NF	NF	NF	W
16	NF	NF	NF	NW

Table 3.1 – List of system scenarios.

system. The same occurs if the system is in the DORMANT mode and the sensor readings show a different behaviour from scenario 16.

3.1.3 Component Failures

The component failures are listed in table 3.2.

Component Failure	Description
P_iB $(1 \leq i \leq 6)$	Pipe P_i blocked
P_iF	Pipe P_i fractured
V_iFC $(1 \leq i \leq 3)$	Valve V_i fails closed
V_iFO	Valve V_i fails open
S_iFH $(1 \leq i \leq 2)$	Switch S_i fails high
S_iFL	Switch S_i fails low
C_iFH	Controller C_i fails high
C_iFL	Controller C_i fails low
TR	Water tank ruptured
TL	Water tank leaks
NWMS	No water from main stream

Table 3.2 – Component failures description.

The 6 sections of pipes can fail blocked or fractured; valves fail closed or open; level switchesfail high or low (sensing water level higher that it is and lower that it is) and controllers fail high or low in the same way.

It is assumed that components can only fail in one way at a time. For example TANK can fail either leaking (TL) or fractured (TR), both failures can not occur at the same time (TR.TL).

3.1.4 System Operating Assumptions

A number of assumptions are made for the system:
- when the system starts operating in the ACTIVE mode, the level of the water in the tank is adequate;
- valves V1 and V2 are assumed to have the same capability, so, if they are both open and V3 is closed, assumed that the water is at the required limit, the amount of water entering the tank equals the amount of water leaving the tank and the level remains constant;
- pipes P5 and P6 are assumed to have a larger cross sections compared to the others in order to facilitate the process when the level of water needs to be quickly reduced in the tank;
- if the tank fails ruptured, all the water will leave the tank at a faster rate than refillment is possible;
- if the tank fails leaking, the flow out through the leak is always assumed to be capable of being replaced by the flow in through valve V1.
- component failure P2F cannot be detected because a fracture in pipe section P2 would still allow water to flow into the tank.
- component failure P4F cannot be detected for a similar reason, because a fracture in pipe section P4 would still allow the water to flow through valve V2 out of the tank. Both P2F and P4F remain failures for the system but because of the location of the sensors they are not detectable.

3.2 The Non-coherent Fault Tree Method

In [14] the authors consider the water tank level control system to describe an application of FTA to fault diagnostics with a non-coherent method. Non-coherent FTs make use of NOT logic together with AND and OR logic, taking into account both component-failing and component-working states. With this approach more information about the system can be included in the analysis. The best results are shown for a scheme in which non-coherent FTs of deviating and non-deviating sensors readings are combined together in a FT with an AND gate. The term *deviating* refers to some behaviour which is not expected but it is understood. Prime implicants of the resulting FT are derived for each of the 16 possible scenarios in both ACTIVE and DORMANT modes. In the following subsection, the process of the construction of the non-coherent FTs for the water tank is explained in more detail.

3.2.1 Fault Tree construction

Non-coherent FTs are build for each of the following events:
- *No Flow through Valve V1*,
- *Flow through Valve V1*,
- *No Flow through Valve V2*,
- *Flow through Valve V2*,
- *Flow through Valve V3*,
- *No Flow through Valve V3*,
- *Water in the Overspill Tray*,
- *No Water in the Overspill Tray*.

These are all possible sensor outcomes. Some of them may represent an expected behaviour for an operating mode and a deviating behaviour for the other, for this reason, in the FT structure, the operating mode has to be specified. Starting from the top event, the immediate causes are found and connected using OR and AND gates. The basic events are the component failures of the system and the working components. To include the working components, NOT logic is used.

Figures 3.2, 3.3, 3.4 and 3.5 show the non-coherent FT for sensor reading *Flow through Valve V1*. This is shown in several parts as it would be too large for one figure. In figure 3.2, the box at the top of the tree contains the top event *Flow Through Valve V1*. This is unexpected when the system is in the DORMANT mode, while it is the required behaviour in the ACTIVE operating mode.

In order to have flow through valve V1, two conditions need to be true, water should be available to pass through and the valve should be open, therefore an AND gate is used to connect them. Each of the two events is then analysed again asking *what caused this?* until the components, in their working or failing states, are reached. For example, on the left branch of the FT in figure 3.2, the first basic event is $\overline{\text{NWMS}}$, which represents the negation of the failure NWMS, *No Water from the Main Supply*. This means that the water supply must be working.

Once the structure of the non-coherent FT is build, a qualitative analysis can be carried out obtaining its *prime implicants*. A prime implicant is defined as a minimal combination of component states (working or failing) which cause the FT top event (see [45]).

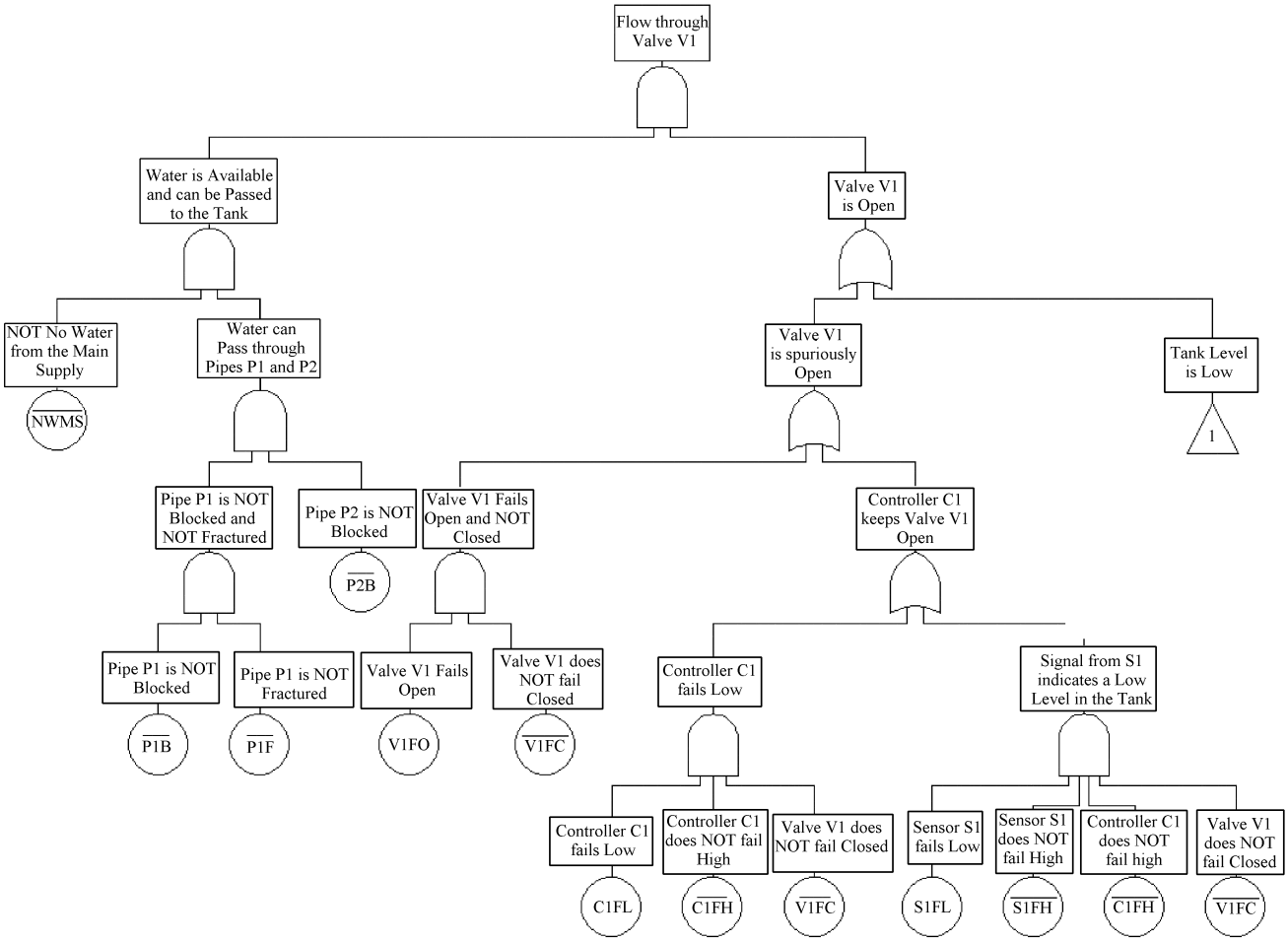


Figure 3.2 – Non-coherent FT for *Flow through Valve V1* (1 of 4).

For the FT discussed, in the ACTIVE mode, there is only one prime implicant containing only working components as flow through valve V1 is expected:

1) $\overline{\text{NWMS}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{V1FC}}.\overline{\text{C1FH}}.\overline{\text{S1FH}}.\overline{\text{V2FC}}.\overline{\text{P3B}}.\overline{\text{P3F}}.\overline{\text{P4B}}.\overline{\text{TR}}$.

For the DORMANT mode, there will be 11 prime implicants, including both working and failing components:

1) $\text{V1FO}.\overline{\text{NWMS}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{V1FC}}$,
2) $\text{C1FL}.\overline{\text{NWMS}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{V1FC}}.\overline{\text{C1FH}}$,
3) $\text{S1FL}.\overline{\text{NWMS}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{V1FC}}.\overline{\text{C1FH}}.\overline{\text{S1FH}}$,
4) $\text{TR}.\overline{\text{NWMS}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{V1FC}}.\overline{\text{C1FH}}.\overline{\text{S1FH}}.\overline{\text{TL}}$,
5) $\text{TL}.\overline{\text{NWMS}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{V1FC}}.\overline{\text{C1FH}}.\overline{\text{S1FH}}.\overline{\text{TR}}$,

6) P3F.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$.$\overline{\text{S1FH}}$.$\overline{\text{P3B}}$.$\overline{\text{TR}}$,

7) P5F.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$.$\overline{\text{S1FH}}$.$\overline{\text{P5B}}$.$\overline{\text{TR}}$,

8) V2FO.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$.$\overline{\text{S1FH}}$.$\overline{\text{V2FC}}$.$\overline{\text{P3B}}$.$\overline{\text{P3F}}$.$\overline{\text{P4B}}$.$\overline{\text{TR}}$,

9) V3FO.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$.$\overline{\text{S1FH}}$.$\overline{\text{V3FC}}$.$\overline{\text{P5B}}$.$\overline{\text{P5F}}$.$\overline{\text{P6B}}$.$\overline{\text{TR}}$,

10) C2FH.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$.$\overline{\text{S1FH}}$.$\overline{\text{V3FC}}$.$\overline{\text{C2FH}}$.$\overline{\text{P5B}}$.$\overline{\text{P5F}}$.$\overline{\text{P6B}}$.$\overline{\text{TR}}$,

11) C2FH.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$.$\overline{\text{S1FH}}$.$\overline{\text{V3FC}}$.$\overline{\text{C2FH}}$.$\overline{\text{S2FH}}$.$\overline{\text{P5B}}$.$\overline{\text{P5F}}$.$\overline{\text{P6B}}$.$\overline{\text{TR}}$.

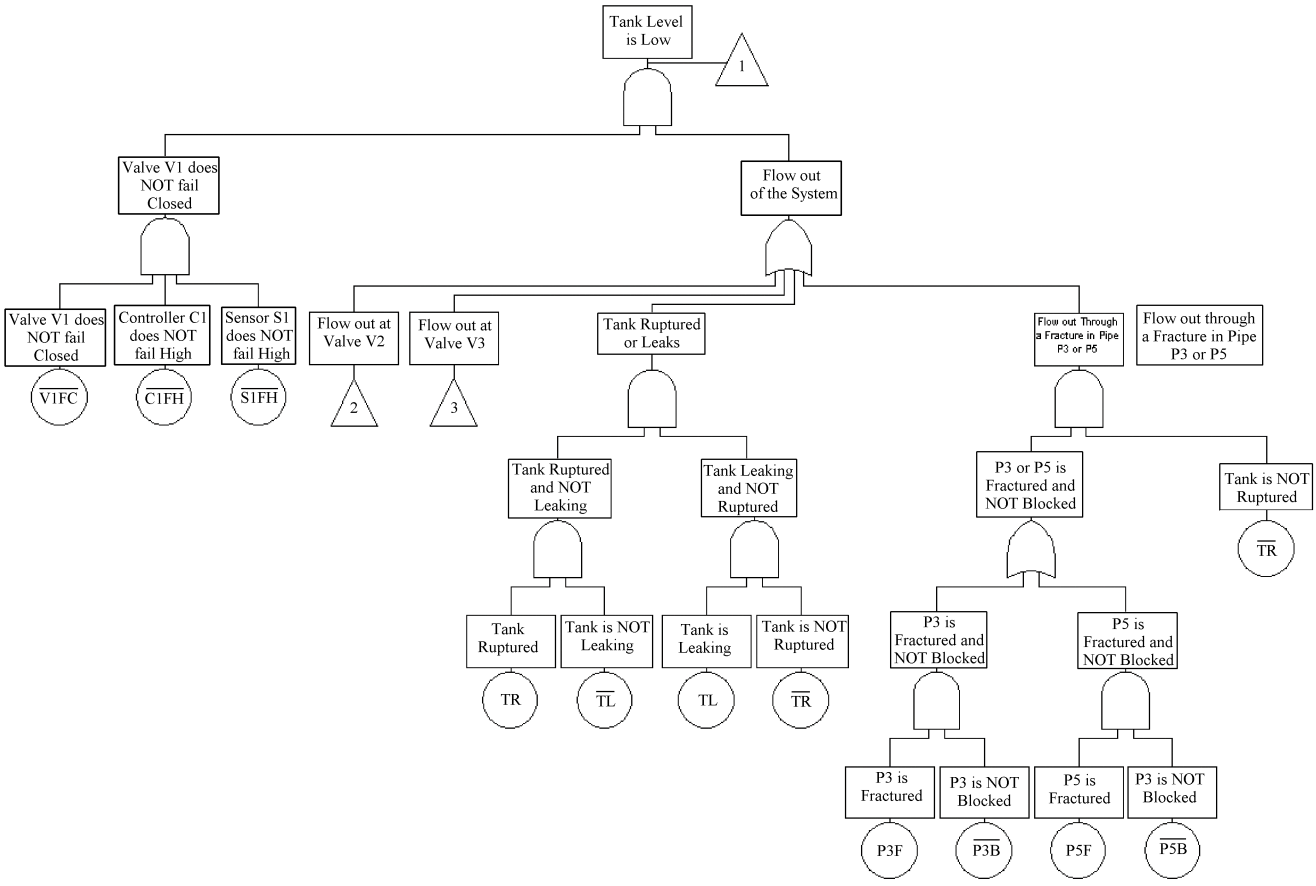


Figure 3.3 – Non-coherent FT for *Flow through Valve V1* (2 of 4).

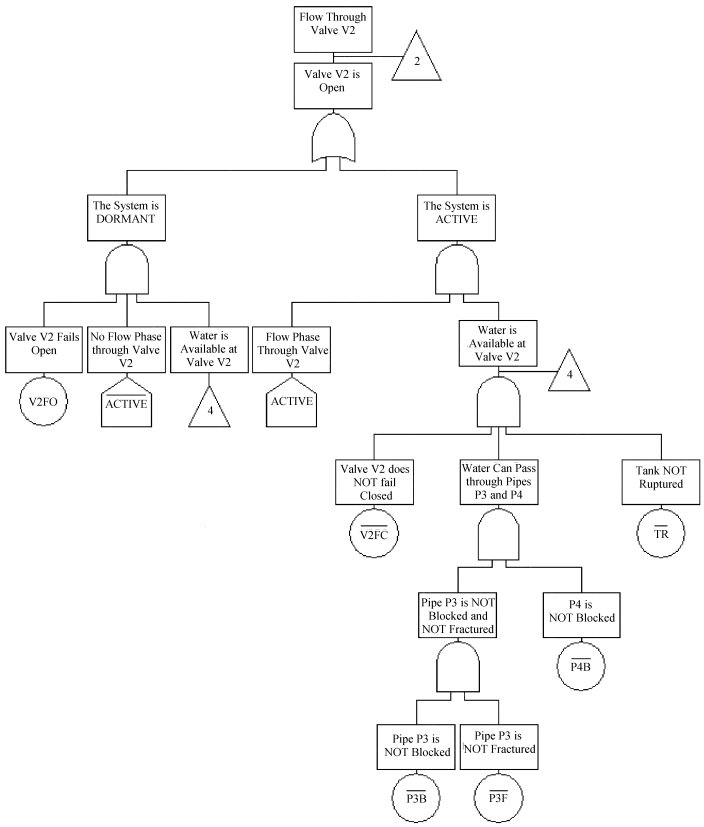


Figure 3.4 – Non-coherent FT for *Flow through Valve V1* (3 of 4).

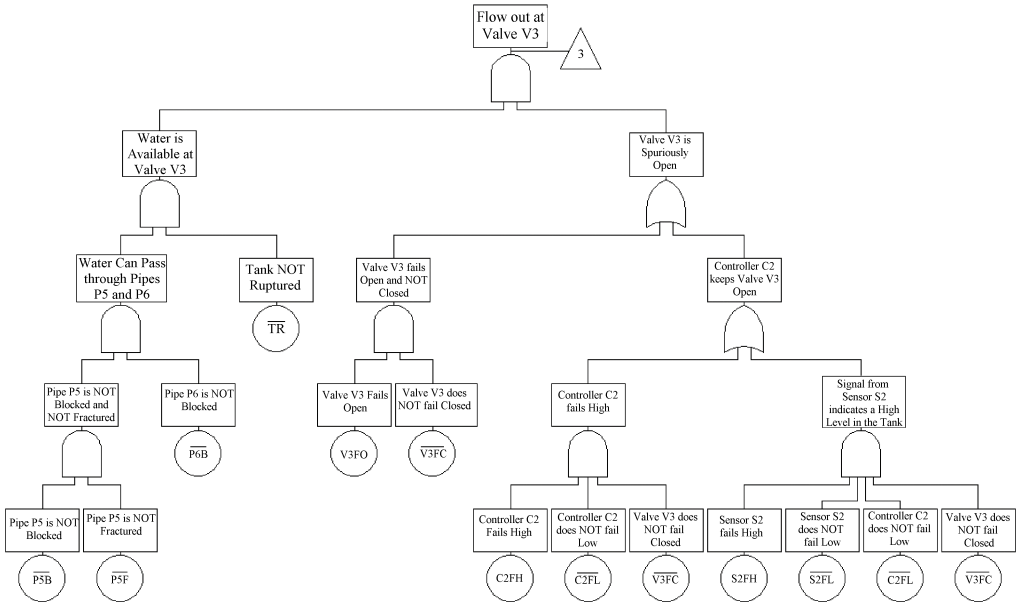


Figure 3.5 – Non-coherent FT for *Flow through Valve V1* (4 of 4).

In the left branch of the FT in figure 3.4, *house events* are used. These are events that have probability either 1 or 0, so they are used to turn on or off entire branches of the tree depending on the state of some variables. In this case, they are used to model the ACTIVE and DORMANT operating modes.
Figure 3.6 shows the non-coherent FT for the sensor reading *Flow through Valve V2* in the ACTIVE mode.

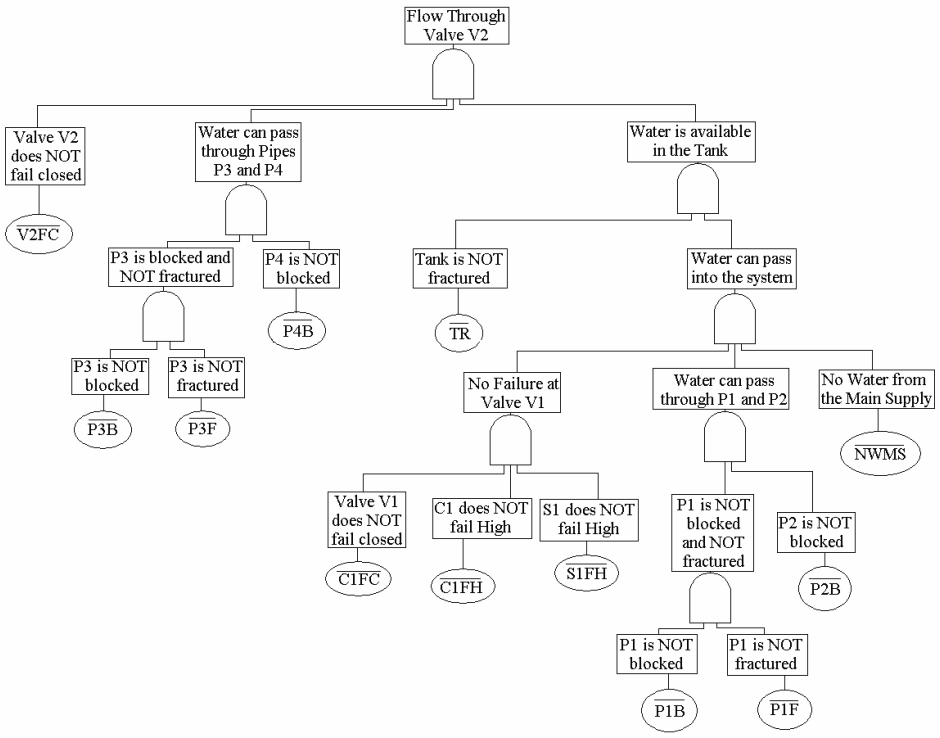


Figure 3.6 – Non-coherent FT for *Flow through Valve V2* in the ACTIVE mode.

There is only one prime implicant for the ACTIVE mode, containing only working components because flow through valve V2 is expected, and this is:

$$\overline{\text{V2FC}}.\overline{\text{P3B}}.\overline{\text{P3F}}.\overline{\text{P4B}}.\overline{\text{TR}}.\overline{\text{V1FC}}.\overline{\text{C1FH}}.\overline{\text{S1FH}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{NWMS}}.$$

In the DORMANT mode, the FT representing flow through valve V2 has a similar logic to the one in figure 3.6 and it has another prime implicant:

$$\text{V2FO}.\overline{\text{V2FC}}.\overline{\text{P3B}}.\overline{\text{P3F}}.\overline{\text{P4B}}.\overline{\text{TR}}.\overline{\text{V1FC}}.\overline{\text{C1FH}}.\overline{\text{S1FH}}.\overline{\text{P1B}}.\overline{\text{P1F}}.\overline{\text{P2B}}.\overline{\text{NWMS}}.$$

Figures 3.7, 3.8 and 3.9 show the non-coherent FT for sensor reading *Flow through Valve V3*.

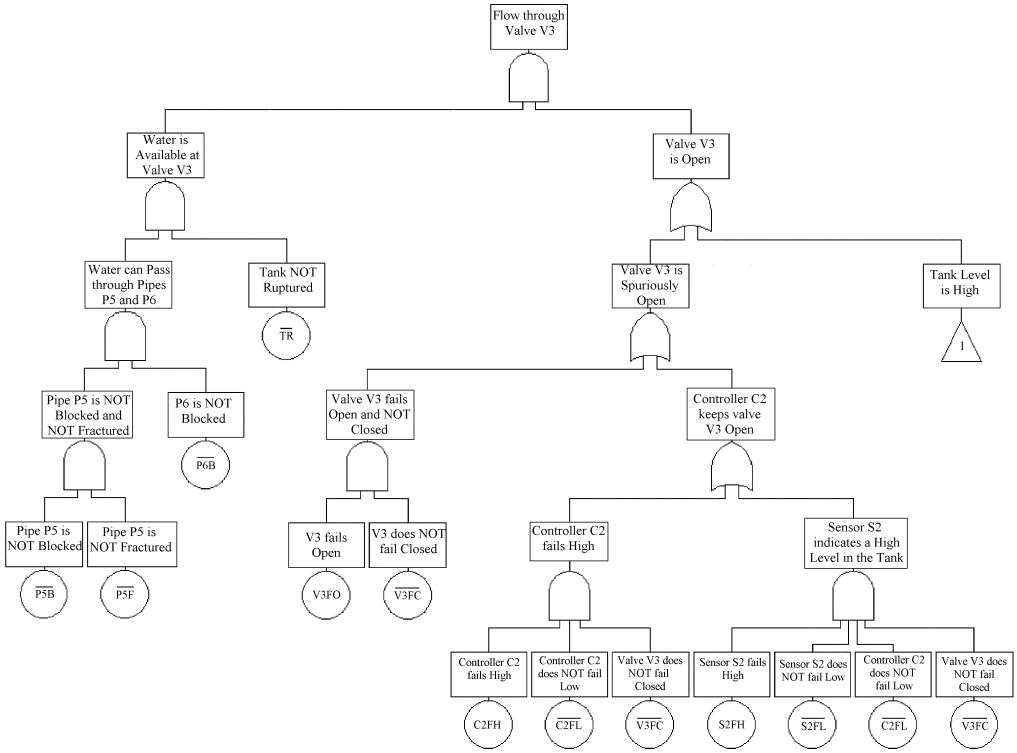


Figure 3.7 – Non-coherent FT for *Flow through Valve V3* (1 of 3).

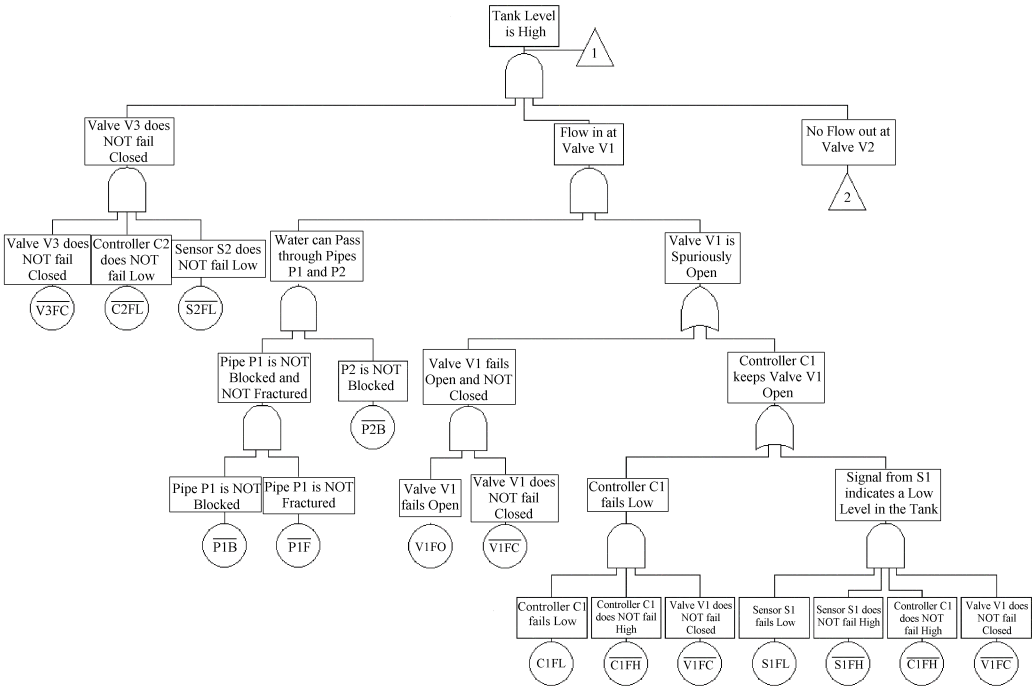


Figure 3.8 – Non-coherent FT for *Flow through Valve V3* (2 of 3).

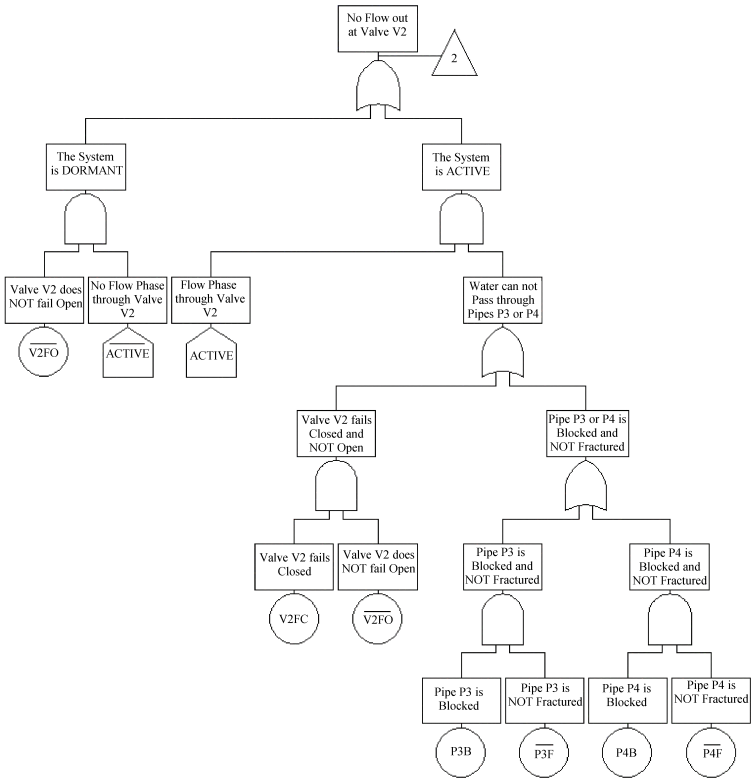


Figure 3.9 – Non-coherent FT for *Flow through Valve V3* (3 of 3).

Figures 3.10, 3.11, 3.12 and 3.13 show the non-coherent FT for sensor reading *Water in the Overspill Tray*.

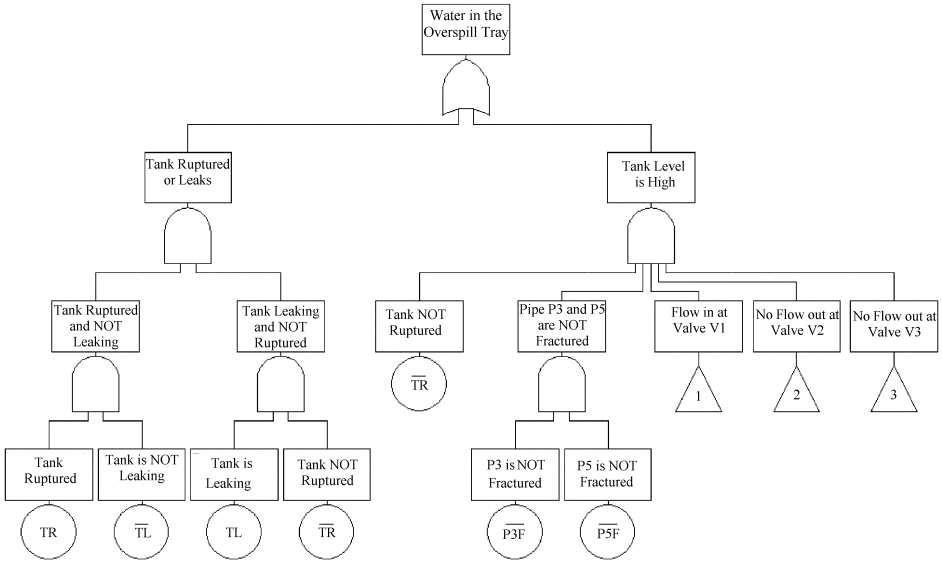


Figure 3.10 – Non-coherent FT for *Water in the Overspill Tray* (1 of 4).

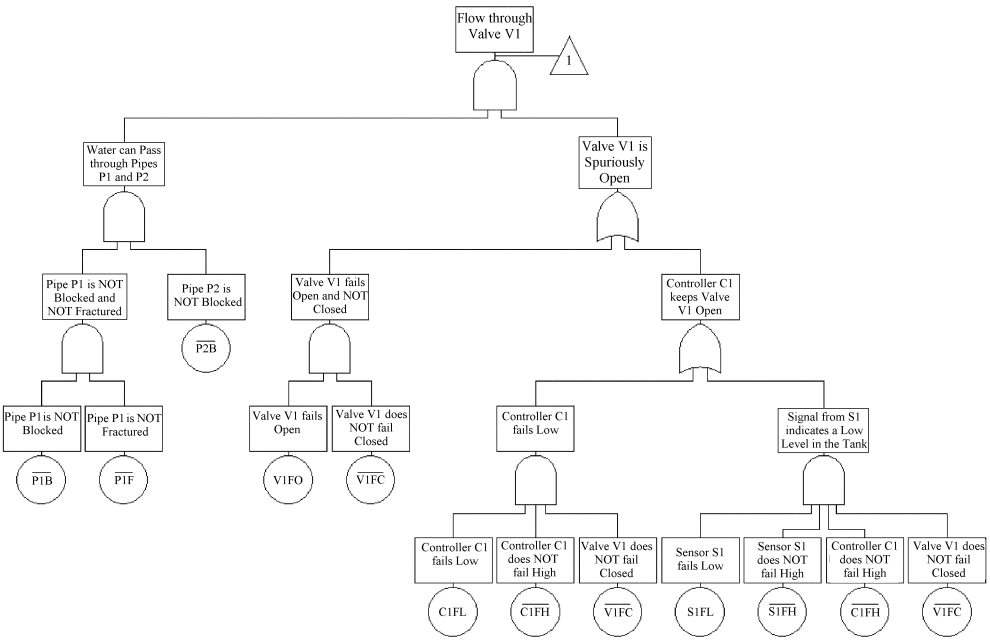


Figure 3.11 – Non-coherent FT for *Water in the Overspill Tray* (2 of 4).

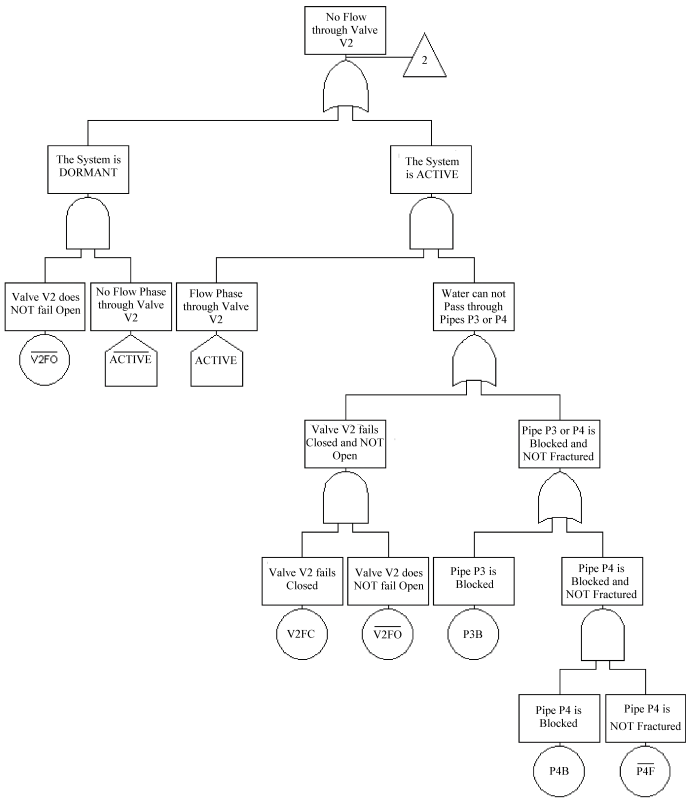


Figure 3.12 – Non-coherent FT for *Water in the Overspill Tray* (3 of 4).

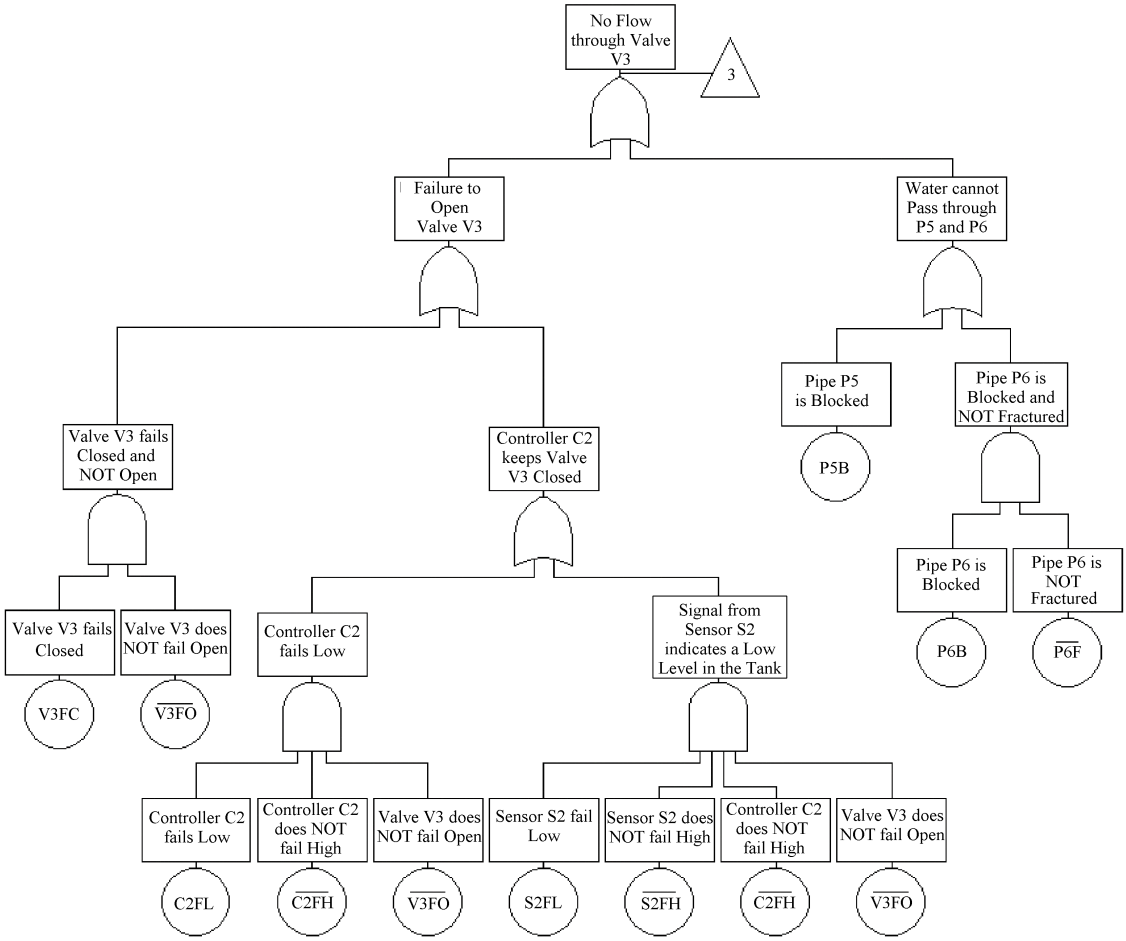


Figure 3.13 – Non-coherent FT for *Water in the Overspill Tray* (4 of 4).

It is so far been illustrated the FTs for the following events:
- *Flow through Valve V1*,
- *Flow through Valve V2*,
- *Flow through Valve V3*,
- *Water in the Overspill Tray*.

These are the only FTs that are necessary for the purpose of the BN method, for a complete list of FTs of the water tank system with their prime implicants, one can refer to [46].

Once all FTs are constructed, each deviating scenario of the system is studied creating a FT that models the scenario. This is obtained connecting with an AND gate the FTs relative to the events that represent the sensor readings of the particular scenario. This is illustrated with an example.

It is assumed that the sensor readings corresponding to scenario 1 are observed in the system when it is operating in the ACTIVE mode. Scenario 1 occurs when flow is detected through all three valves and water is present in the overspill tray (see table 3.1). The behaviour of the system corresponds to the one expected only for valve V1 ad V2. Water is unexpected through valve V3 and also in the water spill tray. Figure 3.14 shows the FT for scenario 1. This includes the deviating as well as the non-deviating sensors. For the non-deviating sensors, the negation of the deviating FT is included using a NOT gate. This approach has shown to produce better results than only including in the FT the deviating sensors [46].

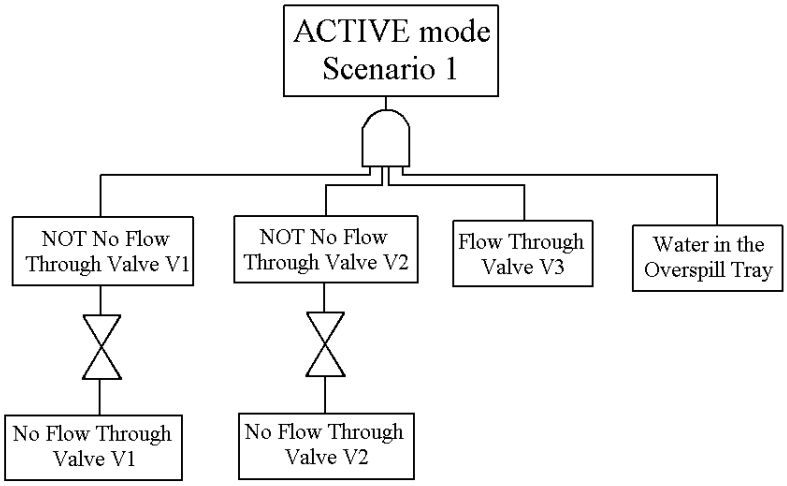


Figure 3.14 – Structure of the FT for scenario 1 in the ACTIVE mode.

Results for scenario 1, that is the component failures that may have caused the malfunction, are obtained analysing the prime implicants of the FT in figure 3.14. These include the component failures as well as the working components. If the working components are removed, we are left with the potential causes for the scenario. These are listed in table 3.3.

	Potential causes
1	TL.S2FH
2	TL.C2FH
3	TL.V3FO

Table 3.3 – Potential causes for scenario 1 in the ACTIVE mode, using the non-coherent FT method .

The results from the FT method for scenario 1 listed in table 3.3 show that there are 3 combinations of component failures that cause the scenario. A leakage in the tank must have occurred, as it appears in all 3 prime implicants. This failure will cause the unexpected behaviour of water in the Overspill Tray. This is the only possible explanation as a fracture

and an overflow are to be excluded. A fracture in the tank is not possible because this would cause the water to leave the tank immediately and flow through valve V2 and V3 would not be possible with an empty tank. An overflow from the top of the tank is not possible because flow is observed through valve V2 and V3 and it is assumed that the water entering the tank from valve V1 equals the water from valve V3, so the level could not have risen up to the top in this scenario. For the deviation of the sensor corresponding to valve V3, there are three possibilities. The valve might have failed open, the controller might have failed high, keeping the valve open, or, the switch might have failed high, measuring the level of water at the safety limit. Therefore, either S2FH, C2FH or V3FO are true.

3.3 The Fault Detection method using Bayesian Networks

Chapter 2 illustrates how a FT can be converted into a BN. This enables a method to be developed with BNs using a similar approach to the one just described. In fact, creating a BN that represents the system would enable us to find the possible causes for the scenarios by calculating the posterior probability. In this way, we would have the advantage of avoiding building and evaluating large FTs for each scenario.

A model for the fault detection of the water tank system has been created using BNs with the following steps:

Conversion: the four non-coherent FTs relative to the sensor readings:

Flow through Valve V1,
Flow through Valve V2,
Flow through Valve V3,
Water in the Overspill Tray,

are converted to four BNs (see figures 3.15, 3.16, 3.17 and 3.18);

Connection: the BNs are connected together to form an *Object Orientated Bayesian Network* (OOBN), which is a class of distinct BNs that are connected to each other in a unique BN that represents the system and that incorporates all scenarios (see figure 3.21);

Evaluation: evidence is given to the fault nodes of the BNs representing the sensor readings in accordance to the 16 possible scenarios in table 3.1. The component failures whose posterior probability have increased with respect to the prior probability are the potential causes for the scenario. A list of the potential causes is produced for each scenario of the ACTIVE mode (see tables 3.6 and 3.7).

3.3.1 Converting the FTs into BNs

In this subsection the detailed networks for the ACTIVE operating mode are shown. The FTs have been converted using the method described in chapter 2. All event nodes have states *yes* and *no*. The failure probability for all component failures are, for simplicity, specified as $q = 0.001667$. It was chosen to give the same figure to all components in order to compare the results with the ones from the FTA.

Figure 3.15 shows the BN obtained converting the FT for *Flow through Valve V1*.

Figure 3.15 – BN for *Flow through Valve V1* in the ACTIVE mode.

When evidence *yes* is given to the fault node, *Flow through Valve V1*, a number of root nodes (components) will have posterior probability that equals 1. These are failures and non-failures that have definitely occurred when flow is observed. Some others will have increased or decreased their prior probabilities according to their contribution to the sensor outcome.

From the network itself, it is not possible to find the exact combinations of failures that cause the sensor outcome, but only the values of each posterior probability. Nevertheless, we know that when a node has posterior probability 1, then the relative component failure will be contained in all prime implicants.

When evidence *yes* is given to node *Flow through Valve V1*, the following non-failing components will have posterior probability 1:

NOT NWMS, NOT P1B, NOT P1F, NOT P2B, NOT V1FC.

As a consequence, the corresponding component failures, NWMS, P1B, P1F, P2B and V1FC, will have posterior probability 0. These represent the component failures that must not have occurred in order to have a correct behaviour through valve V1. Water should be available from the main stream, pipe section P1 should not have failed blocked or fractured and valve V1 should not have failed closed. Other components, such as the controller C1 or the level switch S1 should not have failed. In fact, for example, if controller C1 fails high, it would cause the closure of the valve and consequent no flow. But, the reason NOT C1FH does not have posterior probability 1 is that, even if the controller has failed high, the failure of the valve in the open position would still cause flow through. Therefore, NOT C1FH has increased its posterior probability but it is not a certain event.

If evidence *no* is given to the fault node in the BN, that is, if an unexpected behaviour is observed in valve V1, the posterior probability would reveal the potential causes of the fault. The component failures in table 3.4 will have increased posterior probability. The first 5 causes are the most probable. They are in fact direct causes of the the fault, that is, each of them, alone, would cause the fault in any situation. Causes number 6 and 7 are direct causes but their occurrence alone would not result with the fault in the case of another failure, that is V1FO for C1FH and both V1FO and C1FH for S1FH. This means that the switch failure causes flow unless the valve fails open or the controller fails high.

	Pot. causes	Probab. (in %)
1)	NWMS	10.12
2)	P1B	10.12
3)	P1F	10.12
4)	P2B	10.12
5)	V1FC	10.12
6)	C1FH	10.11
7)	S1FH	10.09
8)	P3B	9.97
9)	P4B	9.96
10)	V2FC	9.96

Table 3.4 – Potential causes for *No Flow through Valve V1* from the posterior probabilities in the BN.

Causes number 8, 9 and 10 are indirect causes. If, for example, pipe section P3 fails blocked, this would cause the level of the water in the tank to rise and, as a consequence, valve V1 would close and flow through the valve would stop. The prime implicants obtained from the FT *No Flow through Valve V1* are indicated in table 3.5.

	Prime Implicants
1)	NWMS
2)	P1B
3)	P1F
4)	P2B
5)	V1FC
6)	C1FH.$\overline{\text{V1FO}}$
7)	S1FH.$\overline{\text{V1FO}}$.$\overline{\text{C1FL}}$
8)	P3B.$\overline{\text{V1FO}}$.$\overline{\text{C1FL}}$.$\overline{\text{S1FL}}$
9)	P4B.$\overline{\text{V1FO}}$.$\overline{\text{C1FL}}$.$\overline{\text{S1FL}}$
10)	V2FC.$\overline{\text{V1FO}}$.$\overline{\text{C1FL}}$.$\overline{\text{S1FL}}$

Table 3.5 – Prime Implicants obtained from the FT *No Flow through Valve V1*.

The tables show consistent results. The prime implicants show the exact combinations of components, failing and working, that lead to the malfunction. The posterior probability of the BN show the probability, given the fault, that the component failure has contributed to the fault. Figure 3.16 shows the BN obtained converting the FT for *Flow through Valve V2*.

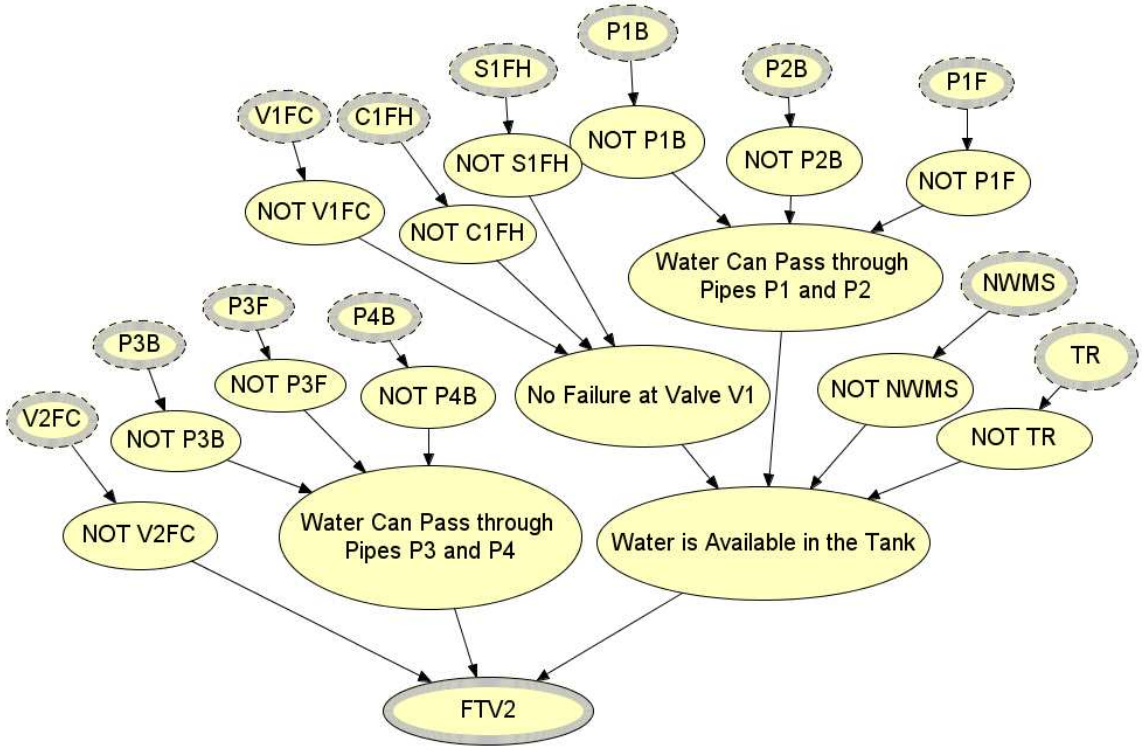


Figure 3.16 – BN for *Flow through Valve V2* in ACTIVE mode.

Comparing the BN in figure 3.16 with the corresponding FT in figure 3.6, one can see that in the BN structure the root nodes representing component failures have been connected to the working states of the components. This is done because it will facilitate the connection of the networks at the next stage.
After giving evidence *yes* to *Flow through Valve V2*, nodes:

$$\overline{\text{V2FC}},\ \overline{\text{P3B}}, \overline{\text{P3F}}, \overline{\text{P4B}}, \overline{\text{TR}},\ \overline{\text{V1FC}},\ \overline{\text{C1FH}},\ \overline{\text{S1FH}},\ \overline{\text{P1B}},\ \overline{\text{P1F}},\ \overline{\text{P2B}},\ \overline{\text{NWMS}}$$

will have posterior probability 1. These correspond to the components non-failures of the prime implicant found for FT *Flow through Valve V2*. Figure 3.17 shows the BN obtained converting the FT for *Flow through Valve V3*.

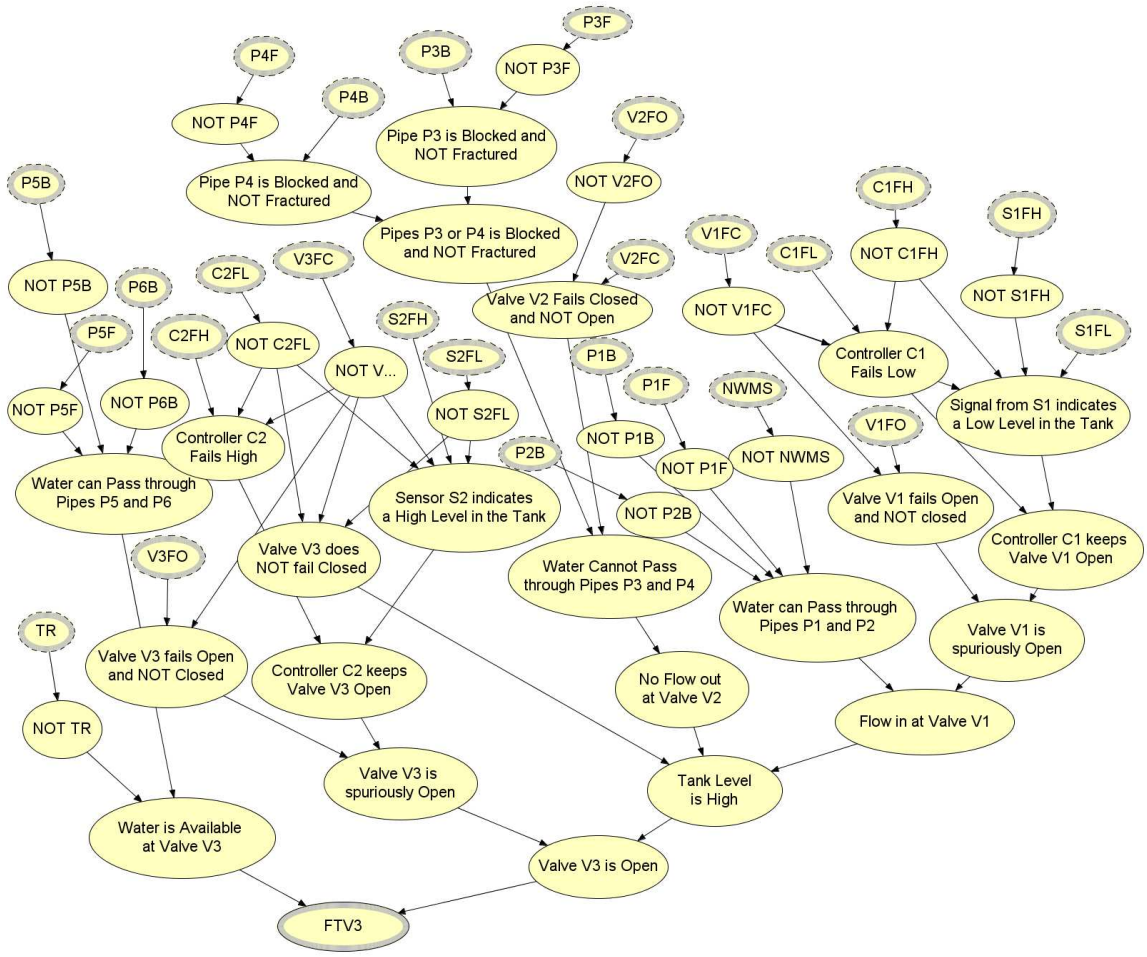


Figure 3.17 – BN for *Flow through Valve V3* in ACTIVE mode.

Finally, figure 3.18 shows the BN obtained converting the FT for *Water in the Overspill Tray*.

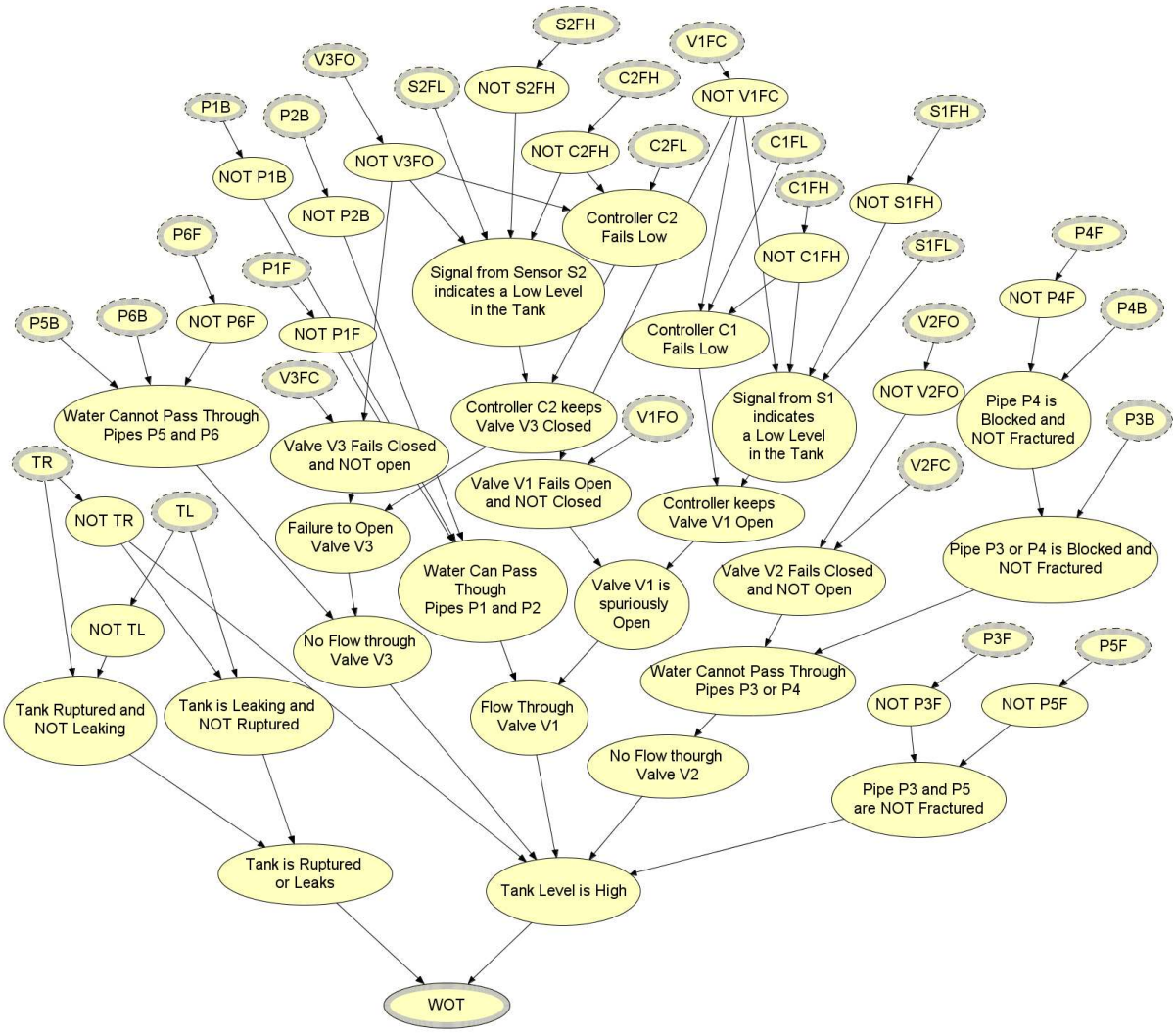


Figure 3.18 – BN for *Water in the Overspill Tray* in ACTIVE mode.

Since the same failure probability has been given to all component failures, even if it is not possible to derive the minimal cut sets, the posterior probability can give a measure in which each node contributes to the sensor outcome. This can be compared with the number of prime implicants a component fault belongs to. When a node has posterior probability 1, the correspondent component's fault will belong to all prime implicants.

When the method is applied in a real world situation, the components' prior probabilities will not be the same for all components. Furthermore, the posterior probability takes into account the timing factor. For example, the posterior probability of node NOT P3B when

evidence *yes* is given to *Flow through Valve V1* is not exactly 1, because pipe P3 blocked is not a direct cause for flow through valve V1. However, the posterior probability is very close to 1 because pipe P3 blocked would obstruct the flow out of the tank and this would make the level of water increase. In this case level switch S1 would detect the level of water above the limit and controller C1 would close the valve. Still, in case the switch or the the controller failed low, or the valve itself failed open, there would be flow through V1.

In the next subsection, the networks relative to the sensor readings will be connected together to analyse different system level scenarios. The advantage of this is that only one network is constructed and all scenarios will be produced giving evidence to the sensor nodes. We have chosen to distinguish the BNs relative to the ACTIVE operating mode and to the DORMANT mode. In principle, the two modes could be described in one network adding a node that has probability 1 or 0 that models house events in the FT.

3.3.2 Connecting the networks

Once the BNs relating to the sensor outputs are created, these are connected together to form a unique BN, able to model all system scenarios. In order to visualise the whole network easily, *input* and *output* nodes have been used in the networks from the previous section. These are nodes that are visible from outside so they can be seen and used in networks contained in other files (this is a feature in version 6.8 of *Hugin Researcher*). Using input and output nodes we are able to create a class collection of networks, called object orientated Bayesian Network (OOBN) [47].

The BNs shown in the previous section will be *subnetworks* of a *master* network representing the system. The subnetworks and the master network will form the OOBN. This approach was found after trying to connect the networks together in one large BN. The resulting system BN contains 240 nodes and dealing with it proved difficult for two reasons. Firstly, the visualisation of a large BN on a computer screen is troublesome and, secondly, locating the nodes from a large list can take a long time. Furthermore, when it comes to the process of evaluating the network, it is not necessary to display all the information about the system, only the nodes representing the component failures and the sensor outputs are needed. Using OOBNs has allowed the model to improve both graphically and in terms of probability evaluation. The graphical view of the separate subnetworks is more concise and clear. The probability evaluation is also facilitated as one can consider the whole system BN and the section separately.

Input nodes are displayed as in figure 3.19. They are fictitious nodes for the network they belong to and they are identical to the nodes they are linked to. So they are identical to their nodes parents. For our purpose, the root nodes representing component faults of the system contained in the networks are input nodes.

Figure 3.19 – Input node.

Output nodes are displayed as in figure 3.20. They are nodes that can be used as parents of nodes belonging to other networks. In the water tank system networks, fault nodes representing sensor readings are output nodes.

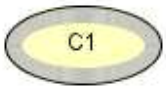


Figure 3.20 – Output node.

It is now possible to connect the four networks creating a new BN that contains the component fault nodes as parents of the fictitious component faults nodes. The four output nodes will be parents of the node *level*, which will be the fault node of the new network. Figure 3.21 shows the system BN. The 28 nodes at the top, represent the component failures of the system. They link toward four square boxes that represent the BNs modelling the sensor outputs. In figure 3.22, one of the squared boxes is expanded to show the INPUT and OUTPUT nodes that belong to the BN relative to valve V2.
In this way, it is possible to visualise how the root nodes in the BN link to the input nodes in the subnetworks and the output nodes in the subnetworks link to the node called *Level*. This is a node with 4 states: *Low*, *Normal*, *High* and *Very high*. It is assumed that the level of water in the tank is predictable in the different scenarios. If the level of water in the tank could be measurable, evidence could be given to the node level as well and this would add extra information. In our particular case, this extra node does not actually give any advantage.

3.3.3 Evaluating the BN

It is now possible to give evidence to the nodes representing the sensor outputs obtaining the posterior probabilities of the component failures. These probabilities provide a measure for the components for having caused the fault when a deviation from the normal functioning is

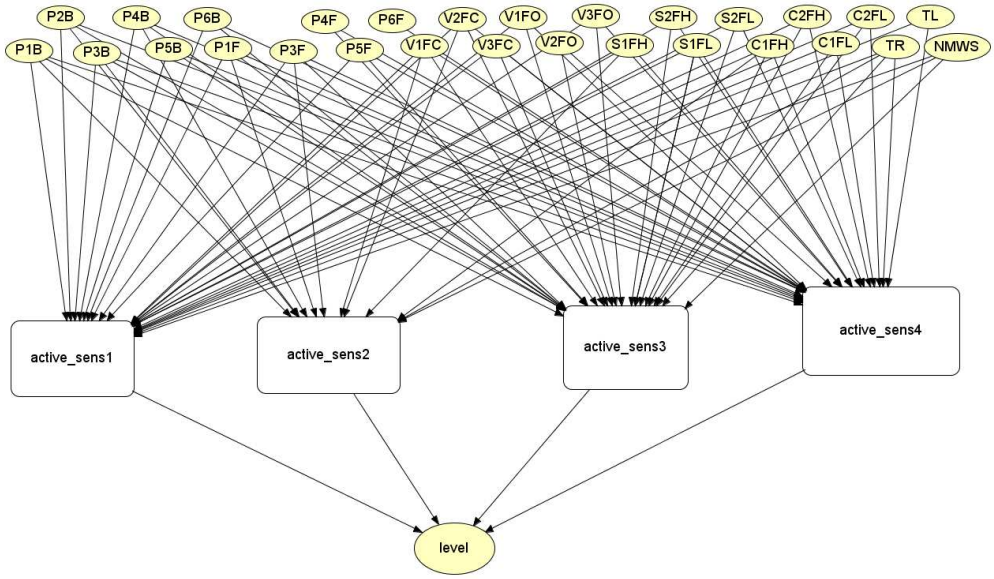


Figure 3.21 – System BN in ACTIVE mode.

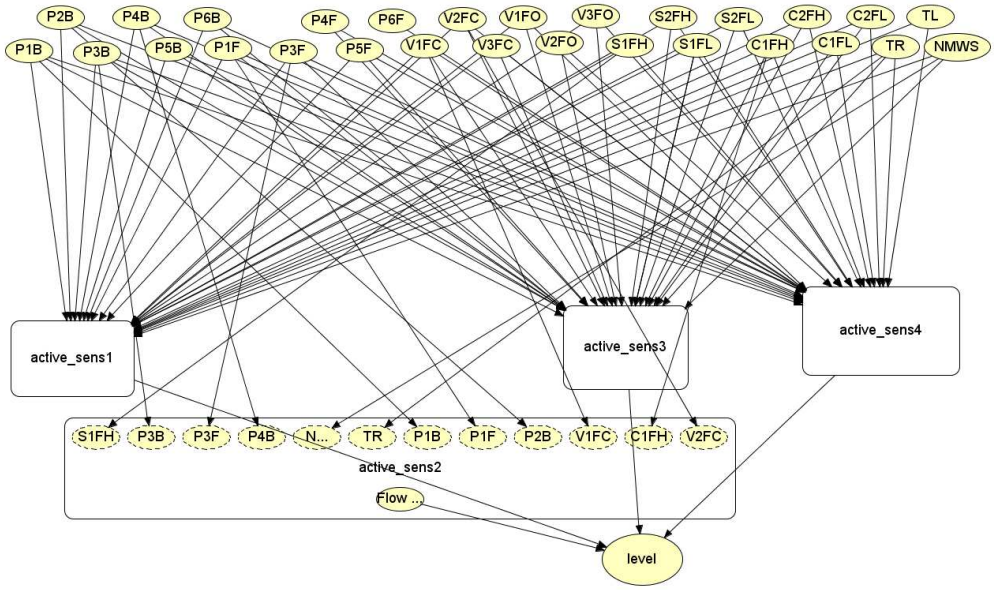


Figure 3.22 – System BN in ACTIVE mode.

observed by the sensors. So we can obtain a list of potential causes for any scenario. Figure 3.23 shows how the posterior probabilities are displayed by the *Hugin* program interface. For scenario 1, for which the sensor outputs are as in table 3.1, evidence *yes* is given to all four fault nodes of the BNs relative to the sensors. These nodes can be accessed from the window in figure 3.23 in the software with no need to open the other BNs. The posterior probabilities of the component faults are also displayed. Once this BN is created, the evaluation is a matter of seconds. The evaluation consists of giving evidence to four nodes and a list of potential

causes is obtained.

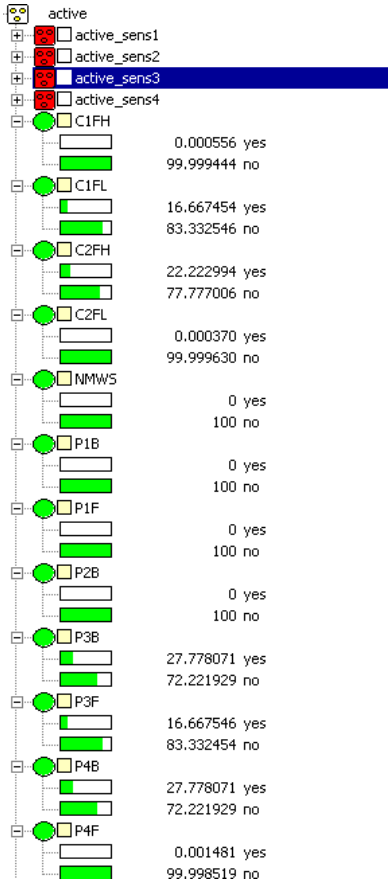


Figure 3.23 – Posterior probabilities in the software *Hugin*.

In the next subsection, the causes are listed for the ACTIVE operating mode for all possible scenarios.

3.3.4 Results

Results are shown in details for the ACTIVE mode. The posterior probabilities of the component failures that represent potential causes are given for all observable scenarios in tables 3.6 and 3.7.

When the posterior probability equals the prior probability, it can be assumed that the component failure has not influenced the sensor outcomes. In other words, the evidence introduced by the sensor observations does not change the probability of the component failure. In the same way, when two events A and B are independent, the conditional probability equals the probability: $P(A \text{ given } B) = P(A)$. Therefore a component failure is considered a potential cause for a scenario if its posterior probability, which is the conditional probability

given the sensor observations, is greater than the prior probability.

Scenario	Symptoms				Possible Causes		
	VF1	VF2	VF3	SP1		Component	Probability (%)
1	Flow	Flow	Flow	Water	1	TL	100
					2	V3FO	33.334444
					3	C2FH	33.333889
					4	S2FH	33.333333
2	Flow	Flow	Flow	No Water	1	V3FO	33.334444
					2	C2FH	33.333889
					3	S2FH	33.333333
3	Flow	Flow	No Flow	Water	1	TL	100
ACTIVE 4	Flow	Flow	No Flow	No Water		No Causes	
5	Flow	No Flow	Flow	Water	1	TL	100
					2	P3B	27.778302
					3	P4B	27.778302
					4	V2FC	27.778302
					5	V3FO	22.223272
					6	C2FH	22.222901
					7	S2FH	22.222531
					8	P3F	16.667685
					9	V1FO	16.667593
					10	C1FL	16.667315
					11	S1FL	16.66431
6	Flow	No Flow	Flow	No Water	1	P3B	27.778302
					2	P4B	27.778302
					3	V2FC	27.778302
					4	V3FO	22.223272
					5	C2FH	22.222901
					6	S2FH	22.222531
					7	P3F	16.667685
					8	V1FO	16.667593
					9	C1FL	16.667315
					10	S1FL	16.66431
7	Flow	No Flow	No Flow	Water	1	TR	99.993334
					2	TL	0.006666
					3	S1FL	0.005416
					4	S2FL	0.005416
					5	V3FC	0.005416
					6	C2FL	0.00516
					7	V1FO	0.005
					8	C1FL	0.004583
					9	P3B	0.003333
					10	P3F	0.003333
					11	P4B	0.003333
					12	V2FC	0.003333
8	Flow	No Flow	No Flow	No Water	1	P3F	99.990001
					2	P5F	0.006666
					3	P4B	0.005
					4	V2FC	0.005
					5	P3B	0.003333

Table 3.6 – Possible causes for scenario 1-8 when the system is operating in the ACTIVE mode.

Results for scenario 1 are shown in table 3.6. In this scenario the identified potential causes for the deviating functioning of the system are 4: TL, V3FO, C2FH and S2FH, with probabilities, respectively, 1, 0.3333, 0.3333 and 0.3333. So, the tank must have a leak and valve V3 may have failed open, the controller and the sensor may have failed high with similar probability. These are the correct potential causes. In fact, water is detected in the overspill

tray but it cannot come from a rupture in the tank, because the tank contains water which is flowing out of valve V2. Therefore a leak is causing the presence of water in the overspill tray. There must also be a failure that is causing valve V3 to stay open, so either V3FO, C2FH or S2FH must have occurred. For this scenario, with the FTA diagnostic method, 3 minimal cut sets were calculated (see table 3.3), these are TL.V3FO, TL.C2FH and TL.S2FH. Results from the BN and from the FTA show the same behaviour.

Scenario	Symptoms				Possible Causes		
	VF1	VF2	VF3	SP1		Component	Probability (%)
15	No Flow	No Flow	No Flow	Water	1	TR	66.665417
					2	TL	33.334583
					3	NMWS	16.667014
					4	P1B	16.667014
					5	P1F	16.667014
					6	P2B	16.667014
					7	V1FC	8.334826
					8	C1FH	8.334687
					9	S1FH	8.334549
					10	P3B	0.002222
					11	P3F	0.002222
					12	P4B	0.002222
					13	V2FC	0.002222
16	No Flow	No Flow	No Flow	No Water	1	V2FC	14.2885544
					2	NMWS	14.286735
					3	P1B	14.286735
					4	P1F	14.286735
					5	P2B	14.286735
					6	P3B	14.285782
					7	P4B	14.285544
					8	P3F	0.002143
					9	C1FH	0.001905
					10	S1FH	0.001905
					11	V1FC	0.001905

Table 3.7 – Possible causes for scenario 15 and 16 when the system is operating in the ACTIVE mode.

For scenario 4 there are no causes, this is because the sensor readings for this scenario are the expected ones and the system is working normally. Scenarios 9 to 14 do not provide any results. This is because the evidence from sensor VF1 and VF2 are in contradiction. That is *No Flow through Valve V1* and *Flow through Valve V2* cannot occur at the same time in the BN. The FTs built for the system and, consequently, the BNs do not take into account the time factor. They analyse the system in steady state time. Therefore, assuming that there is no water through valve V1 and water is still leaving the tank through valve V2 implies that, after a period of time, the tank would be empty, so water through valve V2 would be impossible to observe. The method has the limitation of not considering dynamics in the system.

Another limitation of the diagnosis is the fact that component failures P4F, P6F and P2F are always undetected. However this is due to the fact that the sensors are located corresponding to the valve, therefore, in order to detect these failures, another 3 sensors would

be necessary. However, this does not represent a problem with the method, but it shows the need for more sensors for an effective fault diagnostic method. Apart for these disadvantages, the BN method is able to detect all the potential causes in the scenarios and it does not show any failures that are not actual causes.

Compared with the FTA technique, BNs do not find the exact combinations of failures that can cause a fault as they are not able to produce minimal cut sets or prime implicants. However, in case of a fault, it can be more useful to have a list of potential causes that is ranked starting from the most probable. BNs would also allow, during the inspection, to further update the probability when evidence is observed on the states of the system components. Therefore, for example, if the first component in the list is actually found working, evidence *no* can be given and a new list can be produced.

3.4 Summary

A method for the fault detection and diagnosis of a system has been introduced using BNs with this procedure: first, non-coherent FTs are built to model the sensor outputs. FTs are then converted into BNs and these are connected together. The posterior probability is used to list the potential causes for the deviating scenarios of the system. The method is demonstrated with the example of a water tank system. BNs provide similar results as the FTA but it has many advantages:

- they are easier to deal with since the scenarios are studied in only one network that is analysed giving evidence to its nodes,
- they are more concise in their graphical representations,
- the diagnosis time results faster as there is no need to deduce the prime implicants,
- not only do they identify the possible failures, but they also provide a quantification of the probability associated with a failure. This could also be done with FTA using importance measures but it involves further calculations.
- they allow the introduction of extra evidence about the system. For example, in a maintenance procedure, when a component is found faulty or working, the probability can be updated giving evidence to the node representing the component.

The main disadvantage of the method is that it does not take into account dynamics in the system. Therefore, some scenarios cannot be studied as they are not observable in steady state time. This can be improved introducing sensor readings that measure the flow rate of the water through the valves, rather than just *Flow* or *No Flow*.

Chapter 4

Introducing Dynamics in the Fault Diagnostic method

Introduction

In the previous chapter, a fault diagnostic method has been presented based on BN and FTA techniques. The advantages of both techniques are exploited by combining the use of the two methods together and results show the effectiveness of this approach by application to an example of a water tank system. One of the issues not resolved by this method, when applied to some types of system, is the consideration of dynamic effects.

Systems are often dynamic. This means that their variables and parameters change continuously with time. However, considering dynamic factors in the system when performing fault diagnostics adds difficulties to the task. In the previous analysis, in order to simplify the diagnosis, the sensors were considered to provide very simple information describing, for example, the flow through a valve in only two ways: *Flow* or *No Flow*. In real systems, sensors are generally capable of providing much more information, such as how the measured variable varies over time. So, in the case of a flow sensor, the reading could be the flow rate over time rather than simply a static reading: *Flow*. This simplification in the previous method produces limitations on the analysis since some scenarios are not observable without considering the time trends of the sensor observations.

In [46] the problem is approached introducing dynamic sensor patterns in the diagnosis. In this chapter, we try to follow a similar method using BNs. Since the introduction of more sensor information increases the size of the BNs, the issue of reducing the size of the BNs and optimizing the use of the nodes are confronted.

4.1 Diagnostic method phases

The aim is to produce a general method that is applicable for the diagnosis of many systems and that is also able to deal with some dynamic aspects. Therefore, the procedure is first described in a general way and it is then demonstrated again making use of the water tank system example.

The diagnostic model is built in two stages: the "system modelling and preparation stage" and the "FTs and BNs development stage". In the first phase, the system is analysed in a structured manner in order to identify its sections and, from the ways that the sections behave, all possible system scenarios. In the second phase, first, non-coherent FTs are built representing the deviating states of the sections, then, these FTs are converted into BNs that are used to create a unique network to represent the system. The main difference of this approach with respect to the one in the previous chapter is that the FTs and BNs are built modelling the functioning of the sections in the system, rather than the sensor outcomes. This allows the introduction of more sensor readings that take into account the dynamic variables in the system. The following two sections relate to the two stages of the method.

4.2 System Modelling and Preparation stage

All available information should be collected about the system and its components. Depending on the tasks of different groups of components or sub-systems, the system is divided into sections.

Each section consists of a number of components that are connected together in the system and accomplish a particular task in the process. In every section, there should be at least one sensor monitoring the state of a significant variable for the section. The state of this variable gives an idea of how well the section is performing.

By examining the effects that component failures produce on the functioning of the sections, all possible states are identified for each section. These will include the working state and a number of failing states that depend on the complexity of the section and on the different ways a section can fail to carry out the task for which it is designed.

Once the states are identified for each section, the behaviour of the variables monitored by the sensors of the sections are observed in the different section states. This should lead to the identification of all possible patterns that a section sensor follows. At times, a single

state can give rise to several sensor patterns.

Finally, the combinations of patterns for the sensor readings are used to list the system scenarios. A system scenario is a set of patterns that is observable on the system when a combination of section states (functioning and failed) occurs. Not all possible combinations of patterns in the sections are always achievable in a system.

The modelling and preparation stage described above can therefore be divided in four sub-stages:
- system division into sections,
- identification of section states,
- identification of sensor patterns,
- identification of system scenarios.
This procedure is carried out on the example of the water tank system in the following subsection. The system assumptions are the same as in chapter 3 with the only difference that now sensors S1 and S2 are considered components as well as level sensors.

4.2.1 System division into Sections

The water tank system is divided into 4 sections, indicated in figure 4.1 with the dashed-lined boxes. Each section contains a sensor. Sections 1, 2 and 3 correspond to the flow sensors VF1, VF2 and VF3 respectively. Section 4 contains the tray with sensor SP1.

Section 1 is comprised of valve V1, pipe sections P1 and P2, the controller C1 and sensor S1. The task performed by the section is to introduce water to the tank when the level of water is below the requested limit and to stop the water flow when this limit is reached. Sensor VF1 monitors the flow rate through valve V1.

Section 2 includes valve V2 and pipe sections P3 and P4. As the valve is operated manually, the section should simply allow water to flow out of the tank or stop the water flow when requested. Sensor VF2 measures the flow rate through the valve.

In section 3 there is valve V3, pipe sections P5 and P6, controller C2 and sensor S2. The section task is to allow the water to quickly leave the tank in case of a failure that causes the water level to rise above a safety limit. The sensor in this section, VF2, monitors the flow rate through valve V3.

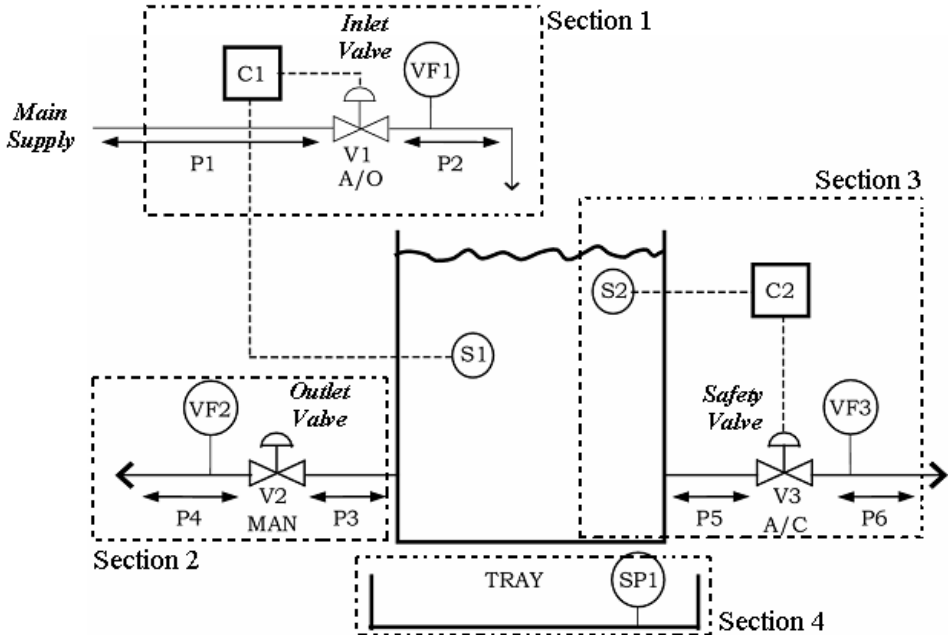


Figure 4.1 – Division into sections in the water tank system.

Section 4 includes the overspill tray and the tank. The overspill tray's task is to collect any spillage from a leak, a rupture or an overflow from the tank. The sensor, SP1, is located in the tray and detects the level of water in the tray. The rate of change of the level can also be measured.

An extra section can also be included by considering the water level in the tank and associating to it the sensors S1/S2 that measure the level of water in the tank. This cannot be considered an actual section as the state of it depends on the state of the whole system. However it will be added as section 5 of the system as it gives extra information on the level of water in the tank and it can validate the states of the other sections.

4.2.2 Identification of Section States

From the description of the sections above, it can be seen that the functioning of sections 1, 2 and 3 depend on the state of the flow through their valves. Section 1 is in its working state if valve V1 is able to open and close according to the level of water in the tank. We deduce that section 1 is failing if it does not perform this task, and this can happen in two ways: water flows into the tank when the level has already reached the requested limit or there is no flow into the tank when water is needed. Therefore, there are three possible states for section 1:

- *Working* (W): valve V1 is able to open and close according to the level of water in the tank.
- *High Flow* (HF): valve V1 is always open allowing water into the tank when this is not needed, that is, when the required level is reached.
- *Low Flow* (LF): valve V1 is always closed and it does not allow water to enter the tank when the level is below the requested limit.

Section 2 and section 3 are similar to section 1, therefore they have 3 possible states: *Working* (W), *High Flow* (HF), and *Low Flow* (LF). The description of the states is the same as for section 1 with the difference that valves V2 and V3 are considered for section 2 and 3 respectively. Note that for section 2, in the ACTIVE mode, failures causing HF are hidden as valve V2 should always stay open. So, for the analysis of the ACTIVE mode, the section can be effectively only in states W or LF. For the same reason, in the DORMANT mode, section 2 has only two states: W and HF.

Regarding section 4 (tank/tray), this is not a process section and is only there to provide information on the status of other elements of the system. The only failing component relevant to the section is the tank (T). The associated sensor, SP1, monitors the level of water in the tray. Therefore, there are 2 states for section 4:

- *Working* (W): the tank is working (it does not fail leaking or ruptured),
- *Failing* (F): the sensor detects water in the tray caused by a rupture or a leak from the tank (overflowing does not imply that the section is failing).

Note that the section is considered in its working state if the presence of water in the tray is caused by an overfilling. This is because only the failures of the components contained in a section can cause the section itself to be in a failing state. If the section has a different behaviour from the one expected and this is a consequence of the malfunctioning of other parts of the system, the section is still considered to be working. In summary, regarding the ACTIVE mode, the sections of the system can be in the following states:

- Section 1: *Working* (W), *High Flow* (HF), *Low Flow* (LF)
- Section 2: *Working* (W), *Low Flow* (LF)
- Section 3: *Working*(W), *High Flow* (HF), *Low Flow* (LF)
- Section 4: *Working*(W), *Failing* (F).

Section 5 has 2 states: *Working* (W), if the level of water in the tank is kept constant,

or *Failing* (F), if a failure in the system causes the level of water to decrease, increase or oscillate. However, the state of section 5 is a consequence of the states of the components of the other 4 sections.

4.2.3 Identification of Sensor Patterns

In the normal operating mode, flow rate is constant through valves V1 and V2 and no flow is observed through V3. When there is a failure in the system, different flow rate measurements can occur. For example, if valve V1 fails closed, there will be no flow through valve V1 so that the level of water in the tank will decrease and the flow rate through valve V2 will also decrease as a consequence of the decreasing pressure from the water in the tank.

Assuming the level of water in the tank is initially set to normal and single or multiple failures occur before the system is started, the following patterns are observable for each sensor reading:

Sensor VF1 - Measuring flow through valve V1 in section 1
Possible patterns:
1. *Constant Flow*
2. *No Flow*
3. *From Constant Flow to No Flow*
4. *Oscillating Flow*

Figure 4.2 represents plots for all possible patterns observed for the sensor VF1 in section 1. These were identified considering the failure states of the section and looking at how these effect the flow trends.

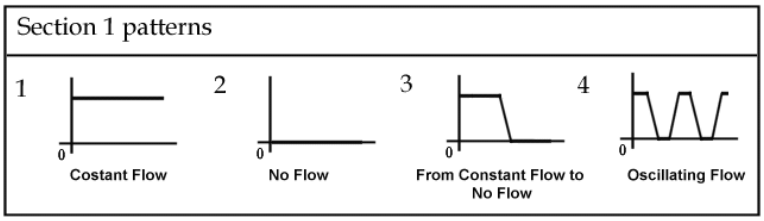


Figure 4.2 – Patterns describing possible flow rate through V1.

As the changes in the flow in section 1 only depend on the valve opening and closing, flow is either constant or zero. In reality, in the act of closing a valve, the flow would not go from a non zero value to zero instantly. The plots reflects this behaviour.

Sensor VF2 - Measuring flow through valve V2 in section 2. Possible patterns:
1. *Flow* (either decreasing or constant)
2. *No Flow*

Both a decreasing and a constant flow through valve V2 in the ACTIVE mode reveal that the section is failing. For this reason, all non zero flow readings can be grouped in a unique pattern.

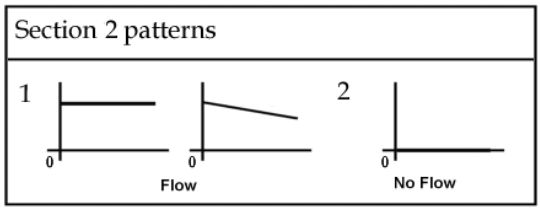


Figure 4.3 – Patterns describing possible flow rate through V2.

Sensor VF3 - Measuring flow through valve V3 in section 3. Possible patterns:
1. *Decreasing Flow*
2. *No Flow*
3. *Oscillating Flow*

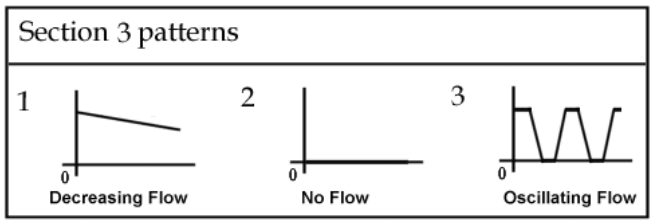


Figure 4.4 – Patterns describing possible flow rate through V3.

The oscillating flow pattern in section 3 occurs when water leaves the system only through valve V3, as a consequence of a failure in section 2. As the level reaches the safety limit the valve is first open and successively closed as the level drops to an acceptable limit. This happens as the cross section of the pipes in section 3 are assumed to be larger that the other pipes.

Sensor SP1 - Measuring the level of water in the tray in section 4 Possible patterns:
1. *No Water in the Tray*
2. *Increasing Water in the Tray*

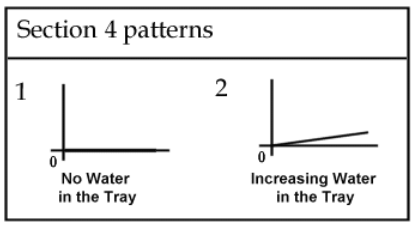


Figure 4.5 – Patterns describing possible level of water in the tray in section 4.

Since it is assumed that all the failures have occurred before the system is started and that components cannot be repaired during monitoring, the only possible failing pattern for the water in the tray is increasing. In case of a leak, the level would increase slowly while a rupture would cause a sudden rise.

Sensor S1/S2 - Measuring the level of water in the tank in section 5. Possible Patterns:

1. *Constant Level*
2. *Decreasing Level*
3. *Oscillating Level*
4. *Increasing Level* (to the Safety Limit)
5. *Increasing level over the Safety Limit* (overspill of the tank)

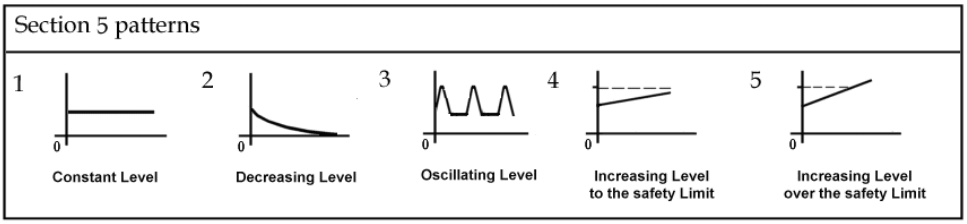


Figure 4.6 – Patterns describing possible level of water in the tank in section 5.

Pattern 3 identifies the behaviour of the level of water in the tank that are caused by either an oscillating pattern in section 1 or in section 3.

4.2.4 Identification of System Scenarios

A scenario is defined as a combination of sensor patterns that is observable in the system as a consequence of one of more component failures. In general this stage could give some problems. In fact, being able to list all possible scenarios can be difficult when the systems are large, complex and with many sensors. For such systems, the method can still be used without this step, that is without knowing exactly the complete list of possible scenarios. There are two ways the diagnostic method can be used: on-line and off-line. The type of analysis we are describing here would be done off-line, before the system is operated, to study the potential causes of all system scenarios. If the diagnostic method is used on a system on-line, while it is operating, it is not necessary to know all possible scenarios. Simply, when a scenario is observed on the system, the causes are deduced from the BNs. The reason it is useful to know all system scenarios in this context is because the method can be validated, checking if the BNs are able to find the actual causes for each of the system scenario.

The total combinations of patterns for the water tank system are 240. But only 24 of them can actually occur in the system, so they represent system scenarios. This is because in the water tank system the behaviour of one section influences the others, therefore the sensor patterns are associated. For example if the patterns in VF1 and VF3 are oscillating then, as a consequence, the level pattern will be oscillating as well.

The scenarios are listed in figure 4.1. This list is obtained manually in the case of the water tank and it is validated by a simulation code in C++ that is described later in section 4.5. For the moment, we can assume that the scenarios are found manually observing the system operation and from the experts knowledge.

	Sensors				
Scenario	**VF1**	**VF2**	**VF3**	**SP1**	**S1 / S2**
1	Constant Flow	Flow	Decreasing Flow	No Water	Decreasing Level
2	Constant Flow	Flow	Decreasing Flow	Increasing Water	Decreasing Level
3	Constant Flow	Flow	No Flow	No Water	Constant Level
4	Constant Flow	Flow	No Flow	No Water	Decreasing Level
5	Constant Flow	Flow	No Flow	Increasing Water	Decreasing Level
6	Constant Flow	No Flow	Decreasing Flow	No Water	Decreasing Level
7	Constant Flow	No Flow	Decreasing Flow	Increasing Water	Decreasing Level
8	Constant Flow	No Flow	No Flow	No Water	Constant Level
9	Constant Flow	No Flow	No Flow	No Water	Decreasing Level
10	Constant Flow	No Flow	No Flow	Increasing Water	Decreasing Level
11	Constant Flow	No Flow	No Flow	Increasing Water	Increasing Over Safety
12	Constant Flow	No Flow	Oscillating Flow	No Water	Oscillating Level
13	Constant Flow	No Flow	Oscillating Flow	Increasing Water	Oscillating Level
14	No Flow	Flow	Decreasing Flow	No Water	Decreasing Level
15	No Flow	Flow	Decreasing Flow	Increasing Water	Decreasing Level
16	No Flow	Flow	No Flow	No Water	Decreasing Level
17	No Flow	Flow	No Flow	Increasing Water	Decreasing Level
18	No Flow	No Flow	Decreasing Flow	No Water	Decreasing Level
19	No Flow	No Flow	No Flow	No Water	Constant Level
20	No Flow	No Flow	No Flow	No Water	Decreasing Level
21	No Flow	No Flow	No Flow	Increasing Water	Decreasing Level
22	From Flow to No Flow	No Flow	No Flow	No Water	Increasing Level
23	Oscillating Flow	No Flow	No Flow	Increasing Water	Oscillating Level
24	No Flow	No Flow	Decreasing Flow	Increasing Water	Decreasing Level

Table 4.1 – System Scenarios.

Scenario 1 occurs when section 1 is *Working* or failing with *High Flow*, section 2 is definitely *Working*, section 3 is failing with *High Flow* and the over-spill tray is *Working*. Considering the sensor patterns and the level in the tank, for all scenarios one can understand the section states and, consequently, the behaviour of the system.

At this point, the system modelling and preparation stage is over and FTs and BNs can be built.

4.3 Fault Tree and Bayesian Networks Development stage

Non-coherent FTs are built for each failing section state. This is more convenient than building FTs that model the sensor outcomes because the same outcome can occur in different situations. While the state identifies the actual way a section is failing. FTs for sections working state, that is success trees, are not necessary as these will be modelled using posterior probability in the BNs.

The FTs are converted into BNs with two differences with respect to the procedure used in the previous chapter: nodes representing the failures of a component are grouped in a unique node and the BNs relative to different failing states of a section are grouped in one network. We obtain in this way a BN for each system section, that will be called *section BNs*. The root nodes of the section BN are the components of the section and the fault nodes are the failing states. A section BN may have more than one fault node as a section of the system can fail in different ways.

Once the system BNs are built, they are connected to form a BN that models the entire system, called the *system BN*. Section BNs and the system BN together form a class of BNs, that is an Object Orientated Bayesian Network (OOBN). As described in the previous chapter, input and output nodes are used to connect the networks, but some extra nodes are introduced to represent the section states and the sensor patterns. Each sensor is modelled by a node whose states are the possible patterns of the sensor, as the patterns of any sensor are exhaustive and mutually exclusive. The section nodes are connected to other nodes that represent section states. They can be connected together as the analysis of the sensor patterns reveal the states of the section. The section nodes are finally connected to the fault nodes of the section BNs that are output nodes.

In summary, the FTs and BNs development stage is achieved in 3 sub-stages:
- Building Non-coherent FTs
- Converting FTs into BNs (building the sections BNs)
- Connecting the section BNs (building the system BN).

In the following subsections this procedure is applied to the water tank system. The analysis is considered only for the ACTIVE operating mode as for the DORMANT mode it is very similar.

4.3.1 Building Non-coherent Fault Trees

The deviating states of the sections of the water tank system, among sections 1 to 4, are 6:

Section 1: *High Flow* (HF) and *Low Flow* (LF)
Section 2: *Low Flow* (LF)
Section 3: *High Flow* (HF) and *Low Flow* (LF)
Section 4: *Failing* (F)

A non-coherent FT is built for each of these. FTs for section 5 are not considered because this is not an actual section as the states of it are a consequence of the component failures of the whole system. The sensor of section 5 is still observed to validate the section states and it will be introduced in the BN structure.

Figure 4.7 shows the non-coherent FT for state *High Flow* in section 1.

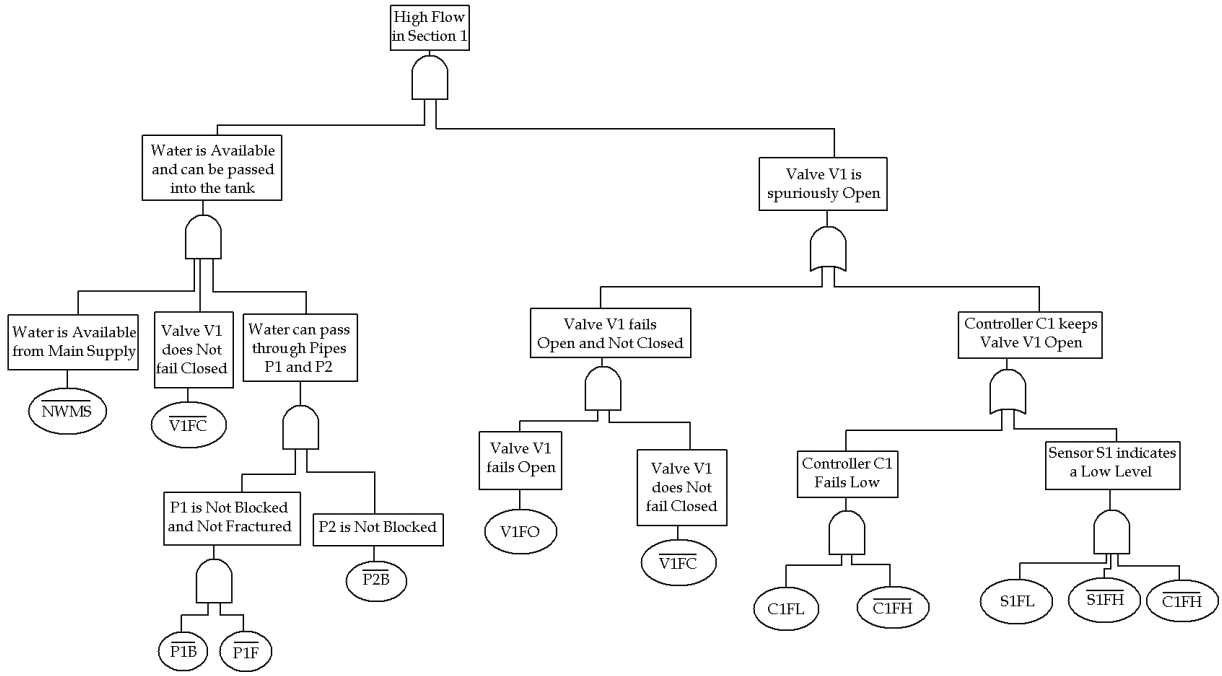


Figure 4.7 – Non-coherent FT for *High Flow* in section 1.

The event *High Flow* occurs if water flow passes through valve V1 when it is not expected. If this FT is compared with the one in chapter 3 in figures 3.2, 3.3, 3.4 and 3.5 that models *Flow through Valve V1*, it can be seen how considering dynamics has made the FTs simpler. This is because before the event *Flow through Valve V1* was considered in all possible con-

texts, while now *High Flow* identifies a specific situation.

There are 3 prime implicants for the FT in in figure 4.7, the possible combinations of working and failing components in the system that cause *High Flow*:

1) V1FO.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$
2) C1FL.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$
3) S1FL.$\overline{\text{NWMS}}$.$\overline{\text{P1B}}$.$\overline{\text{P1F}}$.$\overline{\text{P2B}}$.$\overline{\text{V1FC}}$.$\overline{\text{C1FH}}$.$\overline{\text{S1FH}}$.

Deducing the prime implicants from the FTs can be used as a way of checking their accuracy but it is not necessary for the purpose of the method itself.

Figure 4.8 shows the non-coherent FT for state *Low Flow* in section 1.

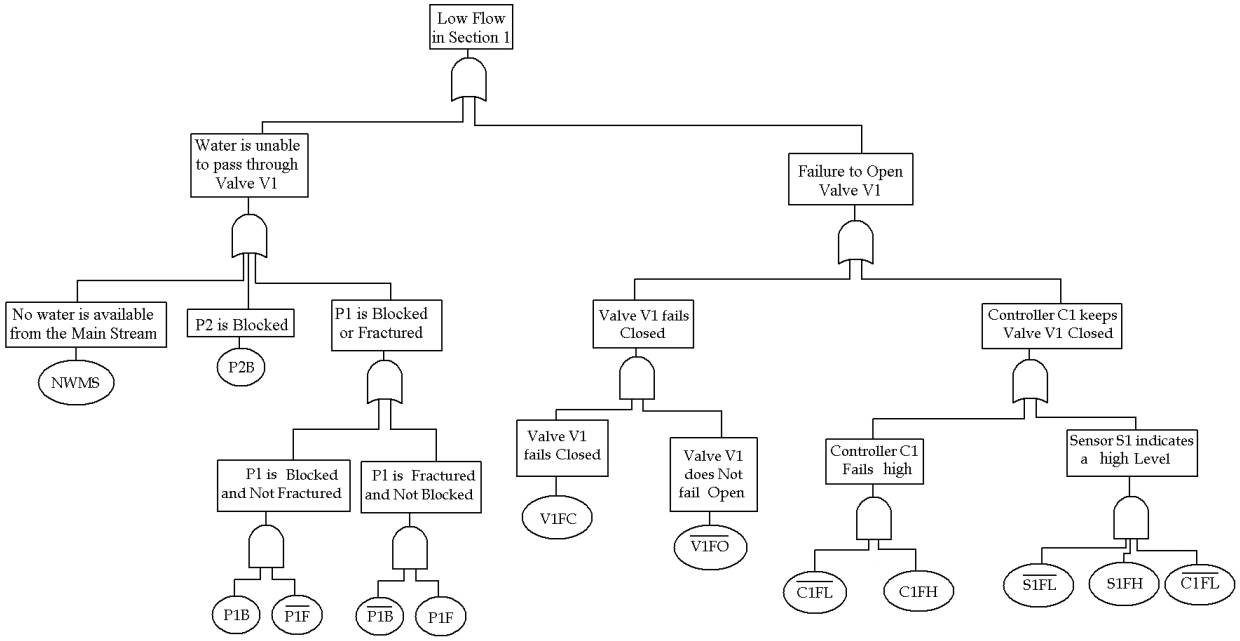


Figure 4.8 – Non-coherent FT for *Low Flow* in section 1.

The FTs relative to the other deviating section states are built with a similar procedure. In the next subsection, it is explained how they are converted into BNs.

4.3.2 Converting Fault Trees into Bayesian Networks

The algorithm to convert a FT into a BN is given in chapter 2 and it is now used with a modification made for the nodes that correspond to the basic events. The basic events in the FTs represent the component failures of the system. Some components can have several

failure modes and it is assumed that they cannot occur at the same time. For example, valve V1 can fail open or closed, but if it has failed open it cannot fail closed. It is equivalent to say that the failure modes of a component are mutually exclusive. If we represent a component as a node in a BN, the failure modes could be the states of the node. Including the working state of a component, the states would also be exhaustive. For example, V1 can be working, failed closed or failed open. One has to be true and V1 cannot be found in another state. The states of a node in a BN are by definition exhaustive and mutually exclusive so it becomes natural to make this change in the conversion. Introducing the components as root nodes gives also two advantages:
- it reduces considerably the number of root nodes, as before if a component had 3 component failures, 3 nodes were needed to represent it while one node with 4 states is sufficient now.
- the non-coherent logic is made simpler. The states of a node are mutually exclusive by definition, therefore when a component failure appears in a BN there is no need to specify that the other component failures from the same component have not occurred. This had to be done in the non-coherent FT structure, as can be seen from the FT in figure 4.8 for example for the event *P1 is Blocked or Fractured*. Introducing the components as root nodes, P1 failed blocked automatically excludes that fails fractured and also with the events on the other way around.

As an example, component V1 is represented by a node called V1 with 3 states: *Working*, *Fails Closed* and *Fails Open*, with conditional probability table as in table 4.2.

	V1
Working	0.99996666
Fails Closed	0.00001667
Fails Open	0.00001667

Table 4.2 – CPT of node V1.

The FTs relating to the same section are converted and combined together in one unique BN. Therefore, each BN will have as many fault nodes as the deviating states of the section. Section 1 has 2 deviating states, *high flow* and *low flow*, so section 1 BN has two fault nodes as in figure 4.9.
Similar BNs are created for the other 3 sections and they can be found in appendix A.

4.3.3 Connecting the Bayesian Networks

Figure 4.10 shows the system BN obtained connecting the section BNs. On the top of the graphical representation are the root nodes representing the components of the entire sys-

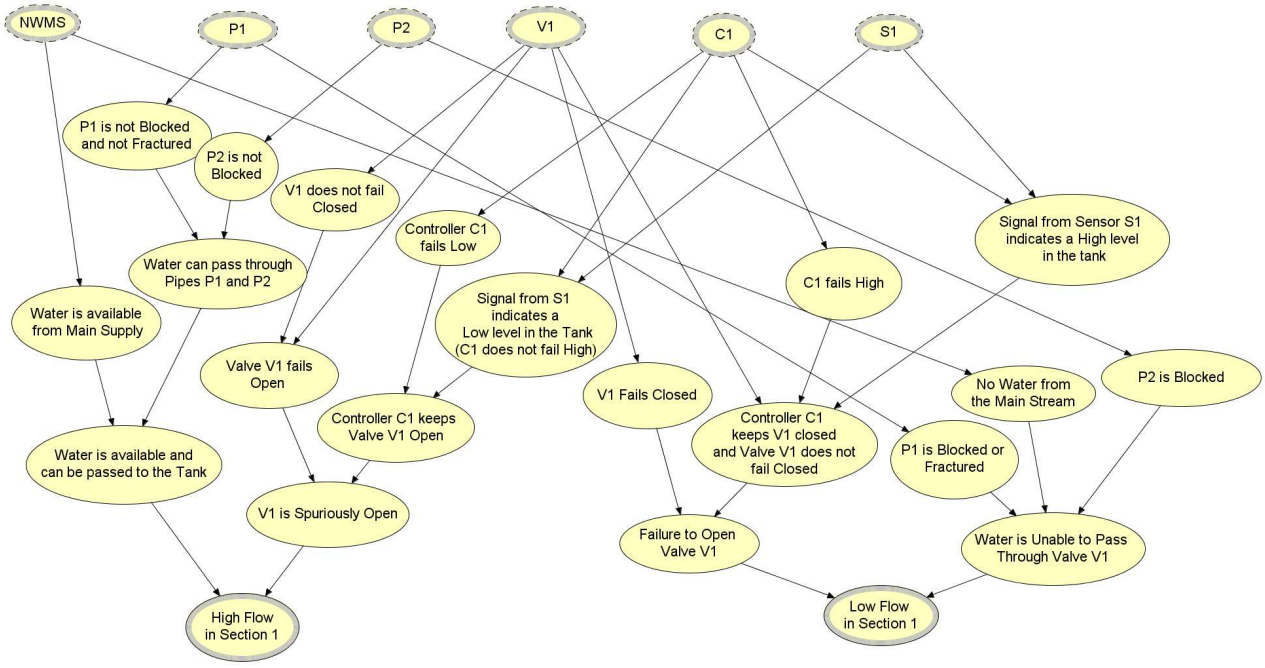


Figure 4.9 – Section 1 BN.

tem. They link to the input nodes contained in the section BNs, represented in the figure with rectangular boxes called SEC1, SEC2, SEC3 and SEC4. The output nodes belonging to the section BNs link to some nodes that are their exact image, called *Sec1HighFlow*, *Sec1LowFlow*, *Sec2LowFlow*, *Sec3HighFlow*, *Sec3LowFlow* and *Sec4Failing*. They represent the deviating section states and they assume the same probability as the output nodes contained in the BNs. These nodes then link to 4 nodes that represent the sections, called *Section 1*, *Section 2*, *Section 3* and *Section 4*. The states of these 4 nodes are the deviating states of the sections plus the working state. For example, node Section 1 has 3 states: W, HF and LF, corresponding to the working state and the 2 deviating states, *High Flow* and *Low Flow*. The conditional probabilities are as in figure 4.11.

As it is impossible that *High Flow* and *Low Flow* occur at the same time for section 1, the probability for the entry of the table corresponding to both *High Flow* and *Low Flow* in state *yes* are given as 50% to HF and 50% to LF but this is a random choice as any other probability would not make any difference.

Sections 2, 3 and 4 have similar nodes. The section nodes link to 5 nodes that represent the sensor patterns called *Patterns 1*, *Patterns 2*, *Patterns 3*, *Patterns 4* and *Patterns 5*, corresponding to sensors VF1, VF2, VF3, SP1 and level sensor S1/S2 respectively. The patterns observed in the sensors are seen as a consequence of the states of the sections plus

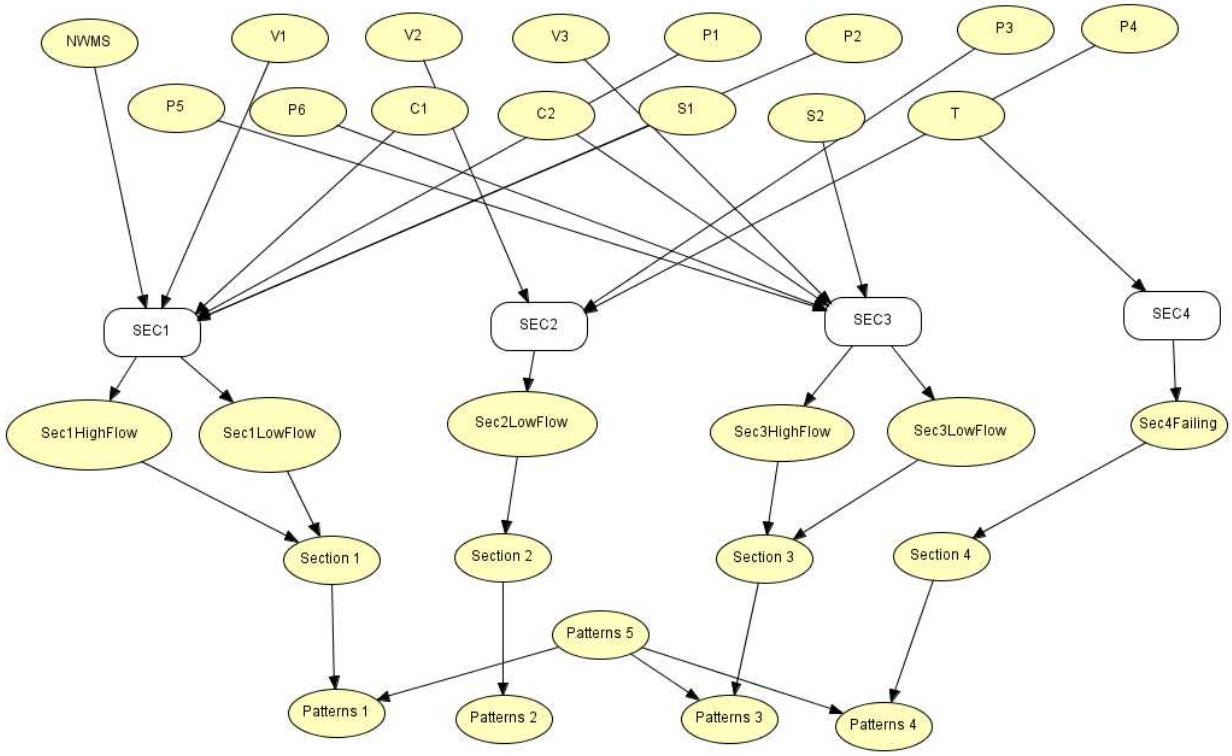


Figure 4.10 – System BN.

Section 1

Sec1LowFlow	yes		no	
Sec1HighFlow	yes	no	yes	no
W	0	0	0	1
HF	0.5	0	1	0
LF	0.5	1	0	0

Figure 4.11 – Conditional Probability Table for node *Section 1* in figure 4.10.

the pattern observed for the water level. For example, *Patterns 1* depends on the state of *Section 1* and on the pattern observed in the level. The states of node *Patterns 1* are the possible observable patterns for sensor VF1, that is, *Flow*, *No Flow*, *From Flow to No Flow* and *Oscillating Flow*. The CPT for *Patterns 1* is represented in figure 4.12 and it shows all possible situations for the behaviour of the sensor pattern. For example, if *Section 1* is in the *Working* state and the level observed in the tank is constant then the observed pattern in *Section 1* will be *Constant Flow*, therefore, in the first entry of the CPT, state *Constant* has probability 1 and all the other states have probability 0.

If the level in the tank is *Increasing* and section 1 has not failed, that is, it is in the working

Patterns 1

Patterns 5	Constant			Decreasing			Increasing			Oscillating			Increasing to full		
Section 1	W	HF	LF	W	HF	LF	W	HF	LF	W	HF	LF	W	HF	LF
Constant	1	1	0	1	1	0	0	1	0	0	1	0	0	1	0
No Flow	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1
From Flow t...	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0
Oscillating Fl...	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

Figure 4.12 – Conditional Probability Table for node *Patterns 1* in figure 4.10.

state, then the observed pattern for section 1 is *From Flow to No Flow*, this means that the valve V1 will close and flow goes from constant to no flow. Instead, if the level is increasing but the section has failed with HF, the pattern in the section will remain constant. All possible combinations of states are included in the CPT. The nodes corresponding to other pattens have similar CPTs. Node *Patterns 2* only depends on the state of *Section 2*, therefore it only has one parent and it does not depend on the trend observed in the tank level.

The structure and probability of the BN is obtained in the way described. The next section now illustrates how such BN can be used for the diagnosis of the system faults.

4.4 Employment of the Diagnostic Method

The method described can be used to derive the list of possible causes for each scenario of the system. A scenario is a combination of sensor patterns that are observable in the system. To derive the component failures that are most likely to have caused a scenario, evidence is given to the following nodes: *Patterns 1*, *Patterns 2*, *Patterns 3*, *Patterns 4* and *Patterns 5*. The posterior probability is considered for the failure modes of the components. These are ranked starting from the most probable to obtain a list of potential causes for the scenario.

Scenario 1 is now considered as an example. The sensor patterns in scenario 1 are:

Constant Flow in VF1(Patterns 1)
Flow in VF2 (Patterns 2)
Decreasing Flow in VF3 (Patterns 3)
No Water in SP1 (Patterns 4)
Decreasing Level in S1/S2 (Patterns 5)

Evidence is introduced to the corresponding states of the nodes *Patterns 1*, *Patterns 2*, *Patterns 3*, *Patterns 4* and *Patterns 5*, as it can be seen in figure 4.13.

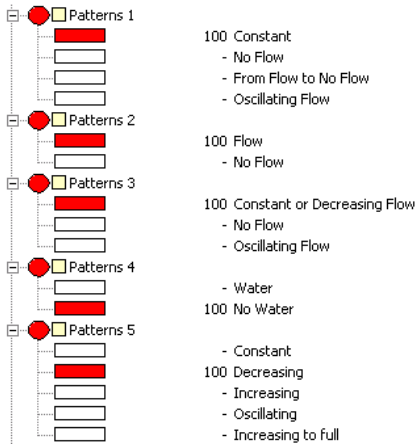


Figure 4.13 – Probability of the nodes Patterns 1-5 when evidence corresponding to scenario 1 is introduced in the BN.

The updated probability is propagated back through the nodes of the network to the root nodes. This is the probability of the component failures to have contributed to scenario 1. Figure 4.14 shows these probabilities displayed in the software *Hugin* for node C2. Scenario 1 occurs if a failure in section 3 causes valve V3 to open when this is not necessary, in other words, section 3 is *failing high*. Section 1 is either *working* or *failing high*, as flow is requested in the tank due to the decreasing level. Section 2 and section 4 are working. As a consequence of the flow through valve V3, the level in the tank is decreasing. It is clear that a failure must have occurred in section 3. One of these is represented by the controller C2 failing high, that is, causing the valve to open when this is not necessary. The BN correctly identifies this failure as the posterior probability of C2 in the state *fails high* is approximately 33.3337% and therefore it has increased with respect to its prior probability, which was 0.001667%.

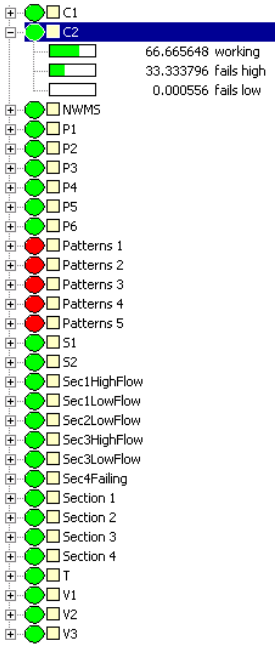
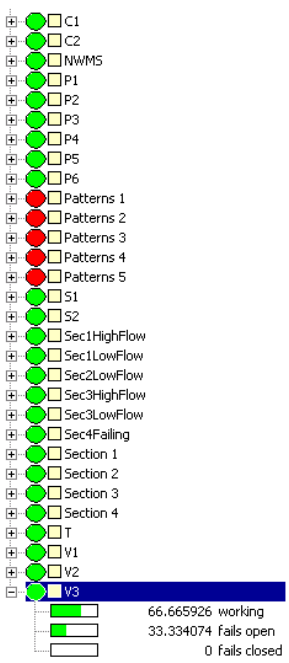
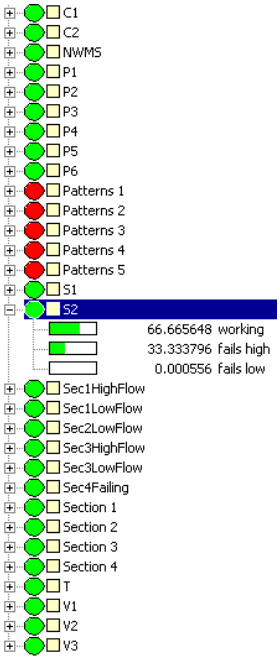


Figure 4.14 – Posterior probability of node C2 in scenario 1.

Components V3 and S2 have also increased their posterior probability (displayed in figures 4.15(a) and 4.15(b)) with respect to their prior probability, which was 0.001667%.



(a) Posterior probability of node V3 in scenario 1.



(b) Posterior probability of node S2 in scenario 1.

Figure 4.15

It is interesting to note that state *fails closed* of node V3 has posterior probability 0 as this is an impossible event for scenario 1 where flow is observed through the valve. State *fails low* of sensor S2 has probability 0.0005% . This is decreased with respect to the prior probability but it is not 0 as it is still possible that sensor S2 fails low in this scenario in the event that V3 fails open or C2 fails high.

In summary, the list of potential causes for scenario 1 are:

V2 *fails open* with probability 33.334%
C2 *fails high* with probability 33.3337%
S2 *fails high* with probability 33.3337% .

These are the actual causes for the scenario. Therefore, the method is able to identify correctly the failures in scenario 1.

Tables 4.3, 4.4 and 4.5 display the results obtained for scenarios 1-24. Although the method identifies correctly the vast majority of the causes among all scenarios, there is a limitation due to the fact that for some scenarios the diagnostic system does not detect the fractures in the pipe sections located next to the sensors. This is the case of scenario 4, which occurs when pipe P5 fails fractured, and it is discussed in the following section.

Scenario	Symptoms Patterns					Possible Causes		
	VF1	VF2	VF3	SP1	S1/S2		Component	Probability (%)
1	Constant Flow	Flow	Decreasing Flow	No Water	Decreasing Level	1	V3FO	33.334
						2	C2FH	33.3337
						3	S2FH	33.3337
2	Constant Flow	Flow	Decreasing Flow	Water	Decreasing Level	1	TR	50
						2	TL	50
						3	V3FO	33.334
						4	C2FH	33.3337
						5	S2FH	33.3337
Expected 3	Constant Flow	Flow	No Flow	No Water	Constant Level		No Causes	
4	Constant Flow	Flow	No Flow	No Water	Decreasing Level		No Causes	
5	Constant Flow	Flow	No Flow	Water	Decreasing Level	1	TR	50
						2	TL	50
6	Constant Flow	No Flow	Decreasing Flow	No Water	Decreasing Level	1	V3FO	33.334
						2	C2FH	33.3337
						3	S2FH	33.3337
						4	P3B	25.0005
						5	P3F	25.0005
						6	P4B	25.0005
						7	V2FC	25.0005
7	Constant Flow	No Flow	Decreasing Flow	Water	Decreasing Level	1	TR	50
						2	TL	50
						3	V3FO	33.334
						4	C2FH	33.3337
						5	S2FH	33.3337
						6	P3B	25.0005
						7	P3F	25.0005
						8	P4B	25.0005
						9	V2FC	25.0005
8	Constant Flow	No Flow	No Flow	No Water	Constant Level		No Causes	
						1	P3B	25.0005
						2	P3F	25.0005
						3	P4B	25.0005
						4	V2FC	25.0005
9	Constant Flow	No Flow	No Flow	No Water	Decreasing Level		No Causes	
						1	P3B	25.0005
						2	P3F	25.0005
						3	P4B	25.0005
						4	V2FC	25.0005
10	Constant Flow	No Flow	No Flow	Water	Decreasing Level	1	TR	50
						2	TL	50
						3	P3B	25.0005
						4	P3F	25.0005
						5	P4B	25.0005
						6	V2FC	25.0005
11	Constant Flow	No Flow	No Flow	Water	Increasing Over Safety	1	C1FL	33.3339
						2	V1FO	33.3339
						3	S1FL	33.3337
						4	P3B	25.0005
						5	P3F	25.0005
						6	P4B	25.0005
						2	V2FC	25.0005
						3	P5B	16.6674
						4	P5F	16.6674
						5	P6B	16.6674
						6	V3FC	16.6674
						7	C2FL	16.6671
						8	S2FL	16.667
12	Constant Flow	No Flow	Oscillating Flow	No Water	Oscillating Level	1	C1FL	33.3339
						2	V1FO	33.3339
						3	S1FL	33.3337
						4	P3B	25.0005
						5	P3F	25.0005
						6	P4B	25.0005
						7	V2FC	25.0005

Table 4.3 – Results for scenarios 1-12 with the BN method.

Scenario	Symptoms Patterns					Possible Causes		
	VF1	VF2	VF3	SP1	S1/S2		Component	Probability (%)
13	Constant Flow	No Flow	Oscillating Flow	Water	Oscillating Level	1	TR	50
						2	TL	50
						3	C1FL	33.3339
						4	V1FO	33.3339
						5	S1FL	33.3337
						6	P3B	25.0005
						7	P3F	25.0005
						8	P4B	25.0005
						9	V2FC	25.0005
14	No Flow	Flow	Decreasing Flow	No Water	Decreasing Level	1	V3FO	33.334
						2	C2FH	33.3337
						3	S2FH	33.3337
						4	NWMS	15.3853
						5	P1B	15.3853
						6	P1F	15.3853
						7	P2B	15.3853
						8	C1FH	14.2863
						9	S1FH	14.2862
						10	V1FC	14.2862
15	No Flow	Flow	Decreasing Flow	Water	Decreasing Level	1	TR	50
						2	TL	50
						3	V3FO	33.334
						4	C2FH	33.3337
						5	S2FH	33.3337
						6	NWMS	15.3853
						7	P1B	15.3853
						8	P1F	15.3853
						9	P2B	15.3853
						10	C1FH	14.2863
						11	S1FH	14.2862
						12	V1FC	14.2862
16	No Flow	Flow	No Flow	No Water	Decreasing Level	1	NWMS	15.3853
						2	P1B	15.3853
						3	P1F	15.3853
						4	P2B	15.3853
						5	C1FH	14.2863
						6	S1FH	14.2862
						7	V1FC	14.2862
17	No Flow	Flow	No Flow	Water	Decreasing Level	1	TR	50
						2	TL	50
						3	NWMS	15.3853
						4	P1B	15.3853
						5	P1F	15.3853
						6	P2B	15.3853
						7	C1FH	14.2863
						8	S1FH	14.2862
						9	V1FC	14.2862
18	No Flow	No Flow	Decreasing Flow	No Water	Decreasing Level	1	V3FO	33.334
						2	C2FH	33.3337
						3	S2FH	33.3337
						4	P3B	25.0005
						5	P3F	25.0005
						6	P4B	25.0005
						7	V2FC	25.0005
						8	NWMS	15.3853
						9	P1B	15.3853
						10	P1F	15.3853
						11	P2B	15.3853
						12	C1FH	14.2863
						13	S1FH	14.2862
						14	V1FC	14.2862

Table 4.4 – Results for scenarios 13-18 with the BN method.

Scenario	Symptoms Patterns					Possible Causes		
	VF1	VF2	VF3	SP1	S1/S2		Component	Probability (%)
19	No Flow	No Flow	No Flow	No Water	Constant Level	1	P3B	25.0005
						2	P3F	25.0005
						3	P4B	25.0005
						4	V2FC	25.0005
						5	NWMS	15.3853
						6	P1B	15.3853
						7	P1F	15.3853
						8	P2B	15.3853
						9	C1FH	15.3851
						10	V1FC	15.3851
						11	S1FH	7.6931
20	No Flow	No Flow	No Flow	No Water	Decreasing Level	1	P3B	25.0005
						2	P3F	25.0005
						3	P4B	25.0005
						4	V2FC	25.0005
						5	NWMS	15.3853
						6	P1B	15.3853
						7	P1F	15.3853
						8	P2B	15.3853
						9	C1FH	14.2863
						10	S1FH	14.2862
						11	V1FC	14.2862
21	No Flow	No Flow	No Flow	Water	Decreasing Level	1	TR	50
						2	TL	50
						3	P3B	25.0005
						4	P3F	25.0005
						5	P4B	25.0005
						6	V2FC	25.0005
						7	NWMS	15.3853
						8	P1B	15.3853
						9	P1F	15.3853
						10	P2B	15.3853
						11	C1FH	14.2863
						12	S1FH	14.2862
						13	V1FC	14.2862
22	From Flow to No Flow	No Flow	No Flow	No Water	Increasing Level	1	P3B	25.0005
						2	P3F	25.0005
						3	P4B	25.0005
						4	V2FC	25.0005
23	Oscillating Flow	No Flow	No Flow	Water	Oscillating Level	1	TR	50
						2	TL	50
						3	P3B	25.0005
						4	P3F	25.0005
						5	P4B	25.0005
						6	V2FC	25.0005
24	No Flow	No Flow	Decreasing Flow	Water	Decreasing Flow	1	TR	50
						2	TL	50
						3	V3FO	33.334
						4	C2FH	33.3337
						5	S2FH	33.3337
						6	P3B	25.0005
						7	P3F	25.0005
						8	P4B	25.0005
						9	V2FC	25.0005
						10	NWMS	15.3853
						11	P1B	15.3853
						12	P1F	15.3853
						13	P2B	15.3853
						14	C1FH	14.2863
						15	S1FH	14.2862
						16	V1FC	14.2862

Table 4.5 – Results for scenarios 19-24 with the BN method.

4.4.1 The case of Scenario 4

From the table in figure 4.3 it can be seen that the BN method is not able to identify any failure to cause scenario 4. The only difference this scenario shows with respect to scenario 3, the expected behaviour, is the level trend in the tank. The flow rate through the three valves and the water level in the tray are the ones expected. The BN method assumes that the level in the tank is a consequence of the other 4 patterns, therefore it does not find any failure. This assumption is always true with only two exceptions, when pipe sections P3 or P5 fail fractured and when the tank fails leaking or fractured. In some scenarios, these failures can be hidden. In the case of scenario 4 for example, although the sensor patterns are correct, the level in the tank is decreasing because there is loss of water through the pipe. But this loss of water is undetected because the sensor is located after the pipe in correspondence of the valve. The BN method does not use the information about the level in the tank. The fact that the level of water is decreasing and that no flow is detected through valve V3 brings to the conclusion that section 3 is working, while it is actually failing.

4.4.2 Modification of system BN

Some modification to the system BN are made in order to include some undetected failures in a few scenarios. This can be done by linking the node patterns 5 with the nodes relative to the section states plus nodes P3, P5 and T. Node *Patterns 5* was a root node before while now it has 7 parents. Its CPT can be determined because knowing the states of the other 4 sections, plus the state of P3, P5 and T, the tank level pattern can be determined exactly. For example, if section 1 works, section 2 works, section 3 fails low and P3, P5 and T work, then the level in the tank will be constant. If, instead, section 1 works, section 2 works, section 3 fails low and P3 and T work while P5 fails fractured, then the level in the tank will be decreasing.

Figure 4.16 shows the system BN with the addition of the links from T, P3, P5 and the section nodes to the node *Patterns 5*. An extra node called *Scenarios* has also been added at the bottom of the graphical representation. This does not change the nature of the way the diagnosis works but it can be useful for giving evidence in a faster way. As each scenario is a combination of sensor patterns, instead of giving evidence to each of the 5 nodes representing the patterns, node *Scenarios* has 24 states representing the scenarios and it is linked to the 5 nodes. When evidence is given to one of its states, the nodes representing the patterns automatically assume the correct evidence for that particular scenario. So the probability is calculated giving evidence only to a node.

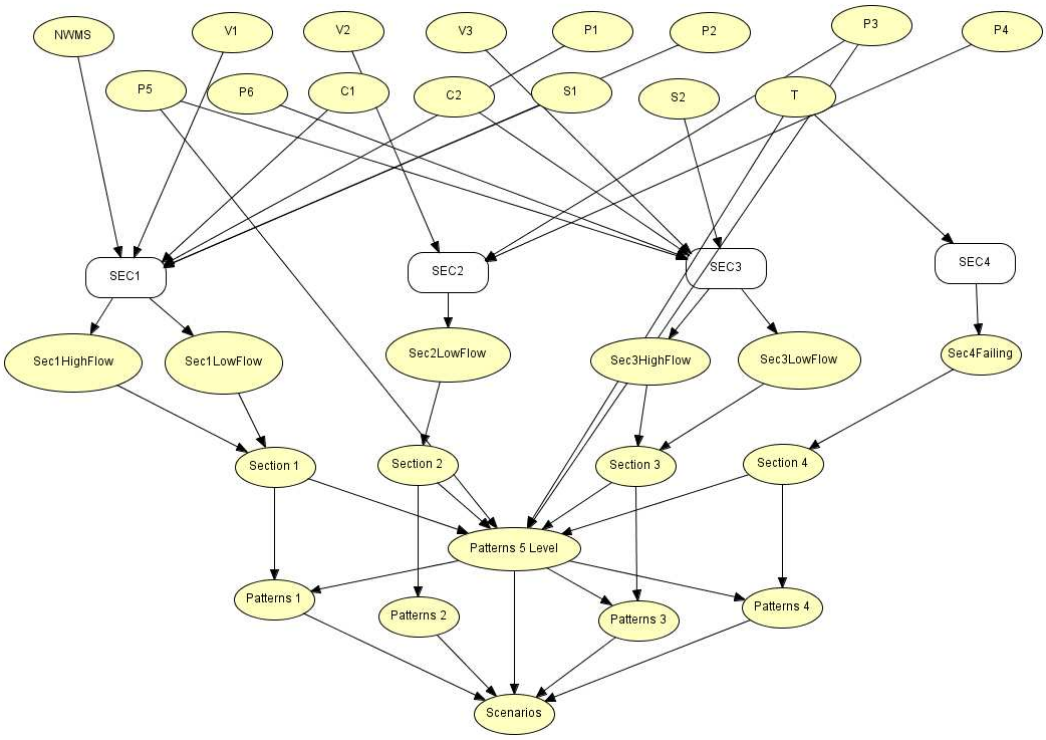


Figure 4.16 – System BN with the modification of the node scenarios and the links to node *Patterns 5*.

The BN is now able to recognise when a fracture occurs on P3 or P5 depending on the level of the tank decreasing when it is not expected. For example, for scenario 4, which is caused by a fracture on pipe P5, the only node with increased posterior probability is P5 with state *fractured* at 100% and the other states at 0 % . This can be seen on figure 4.17 in which node *Scenarios* has been given evidence to its state *4* representing scenario 4. The posterior probability of state *fractured* of node P5, as a consequence, is 100 % .

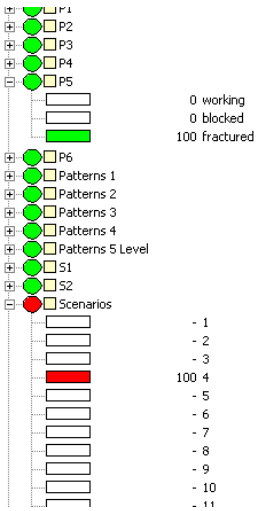


Figure 4.17 – Posterior probability of node P5 for scenario 4 with the modified system BN.

The results of the BN method for all scenarios with the modification introduced by linking *Patterns 5 Level* are discussed in section 4.6 and the results of the modified BN are compared with the ones in the tables in figure 4.3, 4.4 and 4.5. In order to be able to validate the results, a simulation code has been written in C++. Using the simulation of the system, it is possible to assess the effectiveness of the diagnosis for all scenarios. In the next session the simulation code is presented.

4.5 Validation of the Diagnosis with the Simulation of the system

The results of the BN method are validated by a simulation code in the programming language C++. The code is written for two purposes, the first is to deduce all possible scenarios for the system and the other is to obtain the list of the actual component failures that lead to each scenario. Comparing the causes that are found by the BN method and the actual failures from the simulation gives a measure of the effectiveness of the method.

4.5.1 Simulating a Fault in the System

The code is structured in different parts. A function of the code is used to simulate the functioning of the system for a period of time when a number of failures are induced on the water tank. This function is called *Tank Functioning*. The function has three inputs:

- the structural parameters of the tank, such as the section of the pipes and the dimensions of the tank;
- the initial conditions, such as the initial volume of water in the tank;

- the failures that are present in the system (any number of failures is allowed).

After a period of time, decided by the user, the function returns two outputs:

- the water level in the tank;
- the 5 sensor patterns observed on the system.

It is assumed that the failures have occurred before the system is started, this assumption is motivated by the fact that the BN system was designed with the same assumption.

At each time step, the function calculates the water that enters the tank and the water that leaves the tank. The amount of water entering the tank depends only on section 1. If the valve is open and there are no failures such as blockages in the pipes, the flow rate through valve V1 is constant. The amount of water leaving the tank depends on the states of sections 2 and 3 and it also depends on the tank itself when a fracture or a leak are present. The flow rate through valves V2 and V3 depends on the level of water, since a larger volume of water produces a higher pressure and, as a consequence, it causes the flow rate to increase. At each time step, the amount of water that has entered the tank and the water that has left is calculated and the water level is updated. In a text file, the values of the flow rates for sensors VF1, VF2 and VF3 are stored together with the level of water in the tray and the level of water in the tank. Figure 4.18 shows an example of such an output file. The values at the top represent the structural parameter of the tank, these are read from an input file where the user can indicate the inputs for the code. In this example, two failures are supposed to have occurred in the system. For each time step, the tank and tray level and the flow rates through the valves are written in the file.

```
Value of tank_height: 2.000000
Value of tank_radius: 1.000000
Value of At: 3.141593
Value of a1: 0.030000
Value of a2: 0.001000
Value of a3: 0.002000
Value of a4: 0.000100
Value of u: 0.186900

Total volume: 6.283000
Required volume: 5.026400
Safety volume: 5.968850

Failures:

Valve V3 fails Open
No Water in the Main Stream


Initial volume = 5.000000 m^3
Initial level = 79.579819 % of the tank

Tank Level:    Tray Level:     Flow through Valve V1:  Flow through Valve V2:  Flow through Valve V3:
79.577148 %    0.000000 %      0.000000 litres/sec     0.005591 litres/sec     0.011183 litres/sec
79.574478 %    0.000000 %      0.000000 litres/sec     0.005591 litres/sec     0.011183 litres/sec
79.571800 %    0.000000 %      0.000000 litres/sec     0.005591 litres/sec     0.011182 litres/sec
79.569130 %    0.000000 %      0.000000 litres/sec     0.005591 litres/sec     0.011182 litres/sec
79.566460 %    0.000000 %      0.000000 litres/sec     0.005591 litres/sec     0.011182 litres/sec
79.563789 %    0.000000 %      0.000000 litres/sec     0.005591 litres/sec     0.011182 litres/sec
79.561119 %    0.000000 %      0.000000 litres/sec     0.005591 litres/sec     0.011182 litres/sec
```

Figure 4.18 – Example of an Output File generated by the simulation code (1).

At the end of the observation the sensors patterns are calculated based on these values, the output file will end with a few lines as in figure 4.19.

```
19.658083 %    0.000000 %      0.000000 litres/sec     0.002779 litres/sec     0.005558 litres/sec
19.656757 %    0.000000 %      0.000000 litres/sec     0.002779 litres/sec     0.005558 litres/sec
19.655430 %    0.000000 %      0.000000 litres/sec     0.002779 litres/sec     0.005558 litres/sec
19.654104 %    0.000000 %      0.000000 litres/sec     0.002779 litres/sec     0.005558 litres/sec
19.652779 %    0.000000 %      0.000000 litres/sec     0.002779 litres/sec     0.005557 litres/sec
19.651451 %    0.000000 %      0.000000 litres/sec     0.002779 litres/sec     0.005557 litres/sec
19.650126 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005557 litres/sec
19.648800 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005557 litres/sec
19.647472 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005557 litres/sec
19.646147 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005556 litres/sec
19.644821 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005556 litres/sec
19.643494 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005556 litres/sec
19.642168 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005556 litres/sec
19.640842 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005556 litres/sec
19.639515 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005555 litres/sec
19.638187 %    0.000000 %      0.000000 litres/sec     0.002778 litres/sec     0.005555 litres/sec

Sensors outcomes after 5 minutes:

No Flow through Valve V1

Flow through Valve V2

Decreasing Flow through Valve V3

No Water in the Overspill Tray

Decreasing Level in the Tank
```

Figure 4.19 – Example of an Output File generated by the simulation code (2).

A simple pattern recognition algorithm is performed to identify the patterns from the list of values generated. The possible patterns are known for each sensor, so the procedure is better

defined as a pattern matching problem. Since every sensor has different potential patterns, the algorithm is different for every sensor. For example, the patterns for VF2, measuring flow rate through valve V2, are either *Flow* or *No Flow* as all the possible flow patterns that are non zero are grouped together. Therefore, the pattern recognition for sensor VF2 is very simple as it consists in checking for any non zero values of the flow through valve V2. If such a value exists, then the pattern is *Flow* otherwise it is *No Flow*. As failures can only occur before the system is started, it is impossible that patterns change during observation.

Regarding sensor VF1, measuring the flow rate through valve V1, there are 4 possible patterns:
1. *Constant Flow*
2. *No Flow*
3. *From Constant Flow to No Flow*
4. *Oscillating Flow*.

In order to match the behaviour on the system with one of these, the algorithm calculates for each time step the difference of the flow rate value with the previous one. If these differences remain zero, then patterns 1 or 2 are selected, if one of these is negative, then pattern 3 is considered and if they change sign, then the oscillating pattern is detected. In the example shown, the values of the first sensor (in the third column of figures 4.18 and 4.19) do not change and it is sufficient to check one of them to determine that the pattern is *No Flow*. A similar procedure is used for all patterns.

It has to be said that this simple patterns matching algorithm could not work on real sensors, as these are generally affected by noise and a small positive or negative change in the flow rate could be caused by a random variation error in the readings. However, for the purpose of our research, the intention is to use this only for the values generated by the simulation.

Depending on the system parameters, the observation time to be able to correctly observe the patterns can change. For example, if the level in the tank is increasing due to a failure, but the system is not observed for a sufficient interval of time, the level in the tank may not reach the limit where valve V1 is set to close. For the structural parameters in the example, the observation time is set to 5 minutes as this is sufficient to correctly identify the faulty behaviours. By changing the physical structure of the tank, as the pipes sections or the tank volume, it is possible to change the minimum necessary observation time and adapt it to the requirements of a particular system.

4.5.2 Automatic generation of the Faults in the System

The computer program is also capable to work in loop, iterating the function *Tank Functioning*, automatically generating the failures on the system. For example an output file can be created listing the scenarios that are produced by the single failures. One component failure at the time is induced in the system, the tank level is reset each time to the required level and the system is started and run for 5 minutes, or any given time. The sensor patterns caused by each failure are then stored on a file.

A similar procedure is used to create a simulation of the system that aims to detect all possible scenarios. This is achieved by inducing all possible single failures, then all combinations of 2 and 3 failures. The scenarios that are found for the first time are numbered and when the same scenario is observed again a counter variable is incremented. In this way, the code is able to count the scenarios that are caused by combinations of up to 3 failures and for each scenario the number of times that it occurs is memorised together with the component failures that have caused it. There are 240 combinations of sensor patterns, given by 4 (number of patterns for VF1) times 2 (number of patterns for VF2) times 3 (number of patterns for VF3) times 2 (number of patterns for SP1) times 5 (number of patterns for S1/S2). Among these, only 23 of them are observed by the simulation. These correspond to the first 23 that are listed in table 4.1 and that were found manually. The 24th scenario is not detected by the simulation because it is caused only by combinations of 4 or more failures.

For each of the 23 scenarios in the simulation, the occurrence of each component failure is also counted. For example, scenario 1 is generated 118 times. This means that 118 combinations of failures have led to the sensor readings corresponding to scenario 1. Of these 118 combinations, C2FH, controller C2 failing high, belongs to 47 of them. It may be said that C2FH contributes in the 39.83 % of cases to cause scenario 1. In the following, this is referred as the occurrence of a component failure for a particular scenario. In table 4.6 the failure occurrences for scenario 1 are listed in a descending order.

The first 3 failures in the table represent real causes for the scenario, the others are failures that may have occurred in that scenario but they are not the responsible for this deviating behaviour. For example, V1FO may always occur when flow is observed through the valve but, as flow is expected, this is a hidden failure for the scenario as it will be revealed when no flow will be required at some point. Since the deviation of scenario 1 is not caused by section 1, V1FO is not considered an actual cause. It is necessary, for each scenario, to manually identify the actual causes and to discard the hidden ones. This is not a difficult

Failure	Occurrence
V3FO	56
C2FH	47
S2FH	40
V1FO	30
C1FL	27
P2F	24
P4F	24
P6F	24
V2FO	24
S1FL	24
S2FL	16
C2FL	9
S1FH	6
C1FH	3
P1B	0
P1F	0
P2B	0
P3B	0
P3F	0
P4B	0
P5B	0
P5F	0
P6B	0
V1FC	0
V2FC	0
V3FC	0
TR	0
TL	0
NWMS	0

Table 4.6 – Occurrence of the Component Failures for Scenario 1.

task as the actual causes are always ranked as first and when a hidden cause appears on the list, the following ones are never actual causes. The reason this happens in the simulation is that some scenarios are caused only by a single failure, as in the case of scenario 1. Since all combinations of 3 failures are induced, when a real failure occurs together with 2 of the hidden failures for the scenario, this leads to that scenario. However, since the hidden failures occur only to "complete" the set of failures, their occurrences will never be greater than a real cause. In the next subsection, all the results of the simulation code are presented.

4.5.3 Results of the Simulation Code

Tables 4.7 and 4.8 list the results for the 23 scenarios identified by the simulation code. For each scenario, the occurrence of the actual causes identified by the simulation are given. These represent the component failures that are responsible for the deviating behaviour of each scenario. In the next section, these results will be used to validate the diagnostic

method.

Scenario	Simulation Results		
	Component	Occurrence	Occurrence in %
1	V3FO	56	47.46%
	C2FH	47	39.83%
	S2FH	40	33.90%
2	TR	27	50.00%
	TL	27	50.00%
	V3FO	22	40.74%
	C2FH	20	37.04%
	S2FH	18	33.33%
Expected 3			
4	P5F	90	100.00%
5	TR	91	50.00%
	TL	91	50.00%
6	V3FO	47	40.17%
	C2FH	43	36.75%
	S2FH	39	33.33%
	P3B	33	28.21%
	P3F	33	28.21%
	P4B	33	28.21%
	V2FC	33	28.21%
7	TR	12	50.00%
	TL	12	50.00%
	V3FO	8	33.33%
	S2FH	8	33.33%
	C2FH	8	33.33%
	P3B	6	25.00%
	P3F	6	25.00%
	P4B	6	25.00%
	V2FC	6	25.00%
8	P3F	100	100.00%
9	P5F	59	100.00%
	P3B	16	27.12%
	P3F	16	27.12%
	P4B	16	27.12%
	V2FC	16	27.12%
10	TR	55	75.34%
	P3F	30	41.10%
	TL	18	24.66%
	P4B	17	23.29%
	V2FC	17	23.29%
	P3B	16	21.92%
11	P3B	15	33.33%
	P4B	15	33.33%
	V2FC	15	33.33%
	V1FO	15	33.33%
	S1FL	15	33.33%
	C1FL	15	33.33%
	P5B	9	20.00%
	P6B	9	20.00%
	V3FC	9	20.00%
	S2FL	9	20.00%
	C2FL	9	20.00%
12	V1FO	25	43.86%
	P3B	24	42.11%
	C1FL	22	38.60%
	P4B	21	36.84%
	V2FC	21	36.84%
	S1FL	19	33.33%

Scenario	Simulation Results		
	Component	Occurrence	Occurrence in %
13	TL	9	100.00%
	P3B	3	33.33%
	P4B	3	33.33%
	V2FC	3	33.33%
	V1FO	3	33.33%
	S1FL	3	33.33%
	C1FL	3	33.33%
14	V3FO	90	39.47%
	C2FH	83	36.40%
	S2FH	76	33.33%
	NWMS	45	19.74%
	P1B	42	18.42%
	P1F	42	18.42%
	P2B	42	18.42%
	V1FC	42	18.42%
	C1FH	39	17.11%
	S1FH	36	15.79%
15	TL	21	61.76%
	V3FO	14	41.18%
	C2FH	13	38.24%
	TR	13	38.24%
	S2FH	7	20.59%
	P1B	5	14.71%
	P1F	5	14.71%
	P2B	5	14.71%
	V1FC	5	14.71%
	S1FH	5	14.71%
	NWMS	5	14.71%
	C1FH	4	11.76%
	P3F	100	100.00%
16	NWMS	176	21.97%
	P1B	159	19.85%
	P1F	159	19.85%
	P2B	159	19.85%
	V1FC	159	19.85%
	C1FH	147	18.35%
	S1FH	137	17.10%
17	TR	112	51.85%
	TL	104	48.15%
	NWMS	39	18.06%
	P1B	37	17.13%
	P1F	37	17.13%
	P2B	37	17.13%
	V1FC	37	17.13%
	C1FH	36	16.67%
	S1FH	33	15.28%
18	V3FO	28	33.33%
	S2FH	28	33.33%
	C2FH	28	33.33%
	P3B	21	25.00%
	P3F	21	25.00%
	P4B	21	25.00%
	V2FC	21	25.00%
	P1B	12	14.29%
	P1F	12	14.29%
	P2B	12	14.29%
	V1FC	12	14.29%
	S1FH	12	14.29%
	C1FH	12	14.29%
	NWMS	12	14.29%

Table 4.7 – Simulation Results for Scenarios 1-18.

Scenario	Simulation Results		
	Component	Occurrence	Occurrence in %
19	P3B	111	37.25%
	P4B	104	34.90%
	V2FC	104	34.90%
	NWMS	55	18.46%
	P1B	52	17.45%
	P1F	52	17.45%
	P2B	52	17.45%
	V1FC	52	17.45%
	C1FH	49	16.44%
	S1FH	46	15.44%
20	P3F	118	84.89%
	P5F	28	20.14%
	NWMS	24	17.27%
	P1B	23	16.55%
	P1F	23	16.55%
	P2B	23	16.55%
	V1FC	23	16.55%
	C1FH	22	15.83%
	S1FH	21	15.11%
	P4B	14	10.07%
	V2FC	14	10.07%
21	TR	28	50.00%
	TL	28	50.00%
	P3B	14	25.00%
	P3F	14	25.00%
	P4B	14	25.00%
	V2FC	14	25.00%
	P1B	8	14.29%
	P1F	8	14.29%
	P2B	8	14.29%
	V1FC	8	14.29%
	S1FH	8	14.29%
	C1FH	8	14.29%
	NWMS	8	14.29%
22	P3B	61	43.57%
	P4B	52	37.14%
	V2FC	52	37.14%
23	TL	28	100.00%
	P3B	11	39.29%
	P4B	10	35.71%
	V2FC	10	35.71%

Table 4.8 – Simulation Results for Scenarios 19-23.

4.6 Results and Discussions

The results obtained by the diagnostic method using BNs are assessed by comparing them with the results obtained by the simulation code. The results obtained using the modified BN in figure 4.10 and 4.16 are also compared. In the following, the method developed using the BN in figure 4.10 will be referred to as method I, while the improved BN in figure 4.16 will be referred to as method II. This was the one obtained linking the node relative to the level of water in the tank.

Method I, compared with the simulation results, is able to identify the majority of the poten-

tial causes for they faulty scenarios of the system. The results for method I are summarised in the table in figure 4.9. Of the 172 failures among all scenarios, it detects 168 of them, therefore it correctly finds 97.67 % of the failures. Although, it gives two main problems:
- for two scenarios that are caused by a single failure, scenario 4 and scenario 8, the method does not find any cause;
- among all scenarios, the method indicates 9 component failures that are not actual causes for the system state.

Summary				
Scenario	Actual Causes	Causes Identified	Causes Not Found	Extra Causes
1	3	3		
2	5	5		
3	0	0		
4	1	0	1	
5	2	2		
6	7	7		
7	9	9		
8	1	0	1	
9	5	4	1	
10	6	6		
11	11	13		2
12	6	7		1
13	7	9		2
14	10	10		
15	12	12		
16	7	7		
17	9	9		
18	14	14		
19	10	11		1
20	11	10	1	
21	13	13		
22	3	4		1
23	4	6		2
24	16	16		
TOT	172	177	4	9

Table 4.9 – Summary of results for method I.

The limitations of method I have been completely solved by method II. This is in fact able to find all causes of the scenarios and it does not indicate any extra causes that are not among the ones found by the simulation. The diagnosis can be considered 100% effective compared with the simulation that takes into account the behaviour of the system when up to 3 failures occur.

The two methods and the simulation results are summarised and compared with all details of failures and probabilities in the tables in figures 4.10, 4.11 and 4.12.

Scenario	Possible Causes Method I			Possible Causes Method II			Simulation Results			
		Component	Probability (%)		Component	Probability (%)		Component	Occurrence	Occurrence in %
1	1	V3FO	33.334	1	V3FO	33.334	1	V3FO	56	47.46%
	2	C2FH	33.3337	2	C2FH	33.3337	2	C2FH	47	39.83%
	3	S2FH	33.3337	3	S2FH	33.3337	3	S2FH	40	33.90%
2	1	TR	50	1	TR	50	1	TR	27	50.00%
	2	TL	50	2	TL	50	2	TL	27	50.00%
	3	V3FO	33.334	3	V3FO	33.334	3	V3FO	22	40.74%
	4	C2FH	33.3337	4	C2FH	33.3337	4	C2FH	20	37.04%
	5	S2FH	33.3337	5	S2FH	33.3337	5	S2FH	18	33.33%
Expected 3		No Causes			No Causes			No Causes		
4		No Causes		1	P5F	100	1	P5F	90	100.00%
5	1	TR	50	1	TR	50	1	TR	91	50.00%
	2	TL	50	2	TL	50	2	TL	91	50.00%
6	1	V3FO	33.334	1	V3FO	33.334	1	V3FO	47	40.17%
	2	C2FH	33.3337	2	C2FH	33.3337	2	C2FH	43	36.75%
	3	S2FH	33.3337	3	S2FH	33.3337	3	S2FH	39	33.33%
	4	P3B	25.0005	4	P3B	25.0005	4	P3B	33	28.21%
	5	P3F	25.0005	5	P3F	25.0005	5	P3F	33	28.21%
	6	P4B	25.0005	6	P4B	25.0005	6	P4B	33	28.21%
	7	V2FC	25.0005	7	V2FC	25.0005	7	V2FC	33	28.21%
7	1	TR	50	1	TR	50	1	TR	12	50.00%
	2	TL	50	2	TL	50	2	TL	12	50.00%
	3	V3FO	33.334	3	V3FO	33.334	3	V3FO	8	33.33%
	4	C2FH	33.3337	4	C2FH	33.3337	4	S2FH	8	33.33%
	5	S2FH	33.3337	5	S2FH	33.3337	5	C2FH	8	33.33%
	6	P3B	25.0005	6	P3B	25.0005	6	P3B	6	25.00%
	7	P3F	25.0005	7	P3F	25.0005	7	P3F	6	25.00%
	8	P4B	25.0005	8	P4B	25.0005	8	P4B	6	25.00%
	9	V2FC	25.0005	9	V2FC	25.0005	9	V2FC	6	25.00%
8		No Causes		1	P3F	100	1	P3F	100	100.00%
9	1	P3B	25.0005	1	P5F	100	1	P5F	59	100.00%
	2	P3F	25.0005	2	P3B	25.0005	2	P3B	16	27.12%
	3	P4B	25.0005	3	P3F	25.0005	3	P3F	16	27.12%
	4	V2FC	25.0005	4	P4B	25.0005	4	P4B	16	27.12%
				5	V2FC	25.0005	5	V2FC	16	27.12%
10	1	P3B	25.0005	1	TR	79.9988	1	TR	55	75.34%
	2	P3F	25.0005	2	P3F	40.0002	2	P3F	30	41.10%
	3	P4B	25.0005	3	TL	20.0011	3	TL	18	24.66%
	4	V2FC	25.0005	4	P4B	20.0008	4	P4B	17	23.29%
				5	V2FC	20.0007	5	V2FC	17	23.29%
				6	P3B	20.0004	6	P3B	16	21.92%
11	1	C1FL	33.3339	1	P3B	33.3342	1	P3B	15	33.33%
	2	V1FO	33.3339	2	C1FL	33.3339	2	P4B	15	33.33%
	3	S1FL	33.3337	3	V1FO	33.3339	3	V2FC	15	33.33%
	4	P3B	25.0005	4	P4B	33.3337	4	V1FO	15	33.33%
	5	P3F	25.0005	5	S1FL	33.3337	5	S1FL	15	33.33%
	6	P4B	25.0005	6	V2FC	33.3336	6	C1FL	15	33.33%
	7	V2FC	25.0005	7	P5B	20.001	7	P5B	9	20.00%
	8	P5B	16.6674	8	P6B	20.0007	8	P6B	9	20.00%
	9	P5F	16.6674	9	V3FC	20.0007	9	V3FC	9	20.00%
	10	P6B	16.6674	10	C2FL	20.0003	10	S2FL	9	20.00%
	11	V3FC	16.6674	11	S2FL	20.0003	11	C2FL	9	20.00%
	12	C2FL	16.6671							
	13	S2FL	16.667							
12	1	C1FL	33.3339	1	P3B	33.3342	1	V1FO	25	43.86%
	2	V1FO	33.3339	2	C1FL	33.3339	2	P3B	24	42.11%
	3	S1FL	33.3337	3	V1FO	33.3339	3	C1FL	22	38.60%
	4	P3B	25.0005	4	P4B	33.3337	4	P4B	21	36.84%
	5	P3F	25.0005	5	S1FL	33.3337	5	V2FC	21	36.84%
	6	P4B	25.0005	6	V2FC	33.3336	6	S1FL	19	33.33%
	7	V2FC	25.0005							

Table 4.10 – Summary of the results of the two methods and the simulation code for scenarios 1-12.

Scenario		Possible Causes Method I			Possible Causes Method II			Simulation Results		
		Component	Probability (%)		Component	Probability (%)		Component	Occurrence	Occurrence in %
13	1	TR	50	1	TL	100	1	TL	9	100.00%
	2	TL	50	2	P3B	33.3342	2	P3B	3	33.33%
	3	C1FL	33.3339	3	C1FL	33.3339	3	P4B	3	33.33%
	4	V1FO	33.3339	4	V1FO	33.3339	4	V2FC	3	33.33%
	5	S1FL	33.3337	5	P4B	33.3337	5	V1FO	3	33.33%
	6	P3B	25.0005	6	S1FL	33.3337	6	S1FL	3	33.33%
	7	P3F	25.0005	7	V2FC	33.3336	7	C1FL	3	33.33%
	8	P4B	25.0005							
	9	V2FC	25.0005							
14	1	V3FO	33.334	1	V3FO	33.334	1	V3FO	90	39.47%
	2	C2FH	33.3337	2	C2FH	33.3337	2	C2FH	83	36.40%
	3	S2FH	33.3337	3	S2FH	33.3337	3	S2FH	76	33.33%
	4	NWMS	15.3853	4	NWMS	14.2864	4	NWMS	45	19.74%
	5	P1B	15.3853	5	P1B	14.2864	5	P1B	42	18.42%
	6	P1F	15.3853	6	P1F	14.2864	6	P1F	42	18.42%
	7	P2B	15.3853	7	P2B	14.2864	7	P2B	42	18.42%
	8	C1FH	14.2863	8	C1FH	14.2863	8	V1FC	42	18.42%
	9	S1FH	14.2862	9	S1FH	14.2862	9	C1FH	39	17.11%
	10	V1FC	14.2862	10	V1FC	14.2862	10	S1FH	36	15.79%
15	1	TR	50	1	TR	50	1	TL	21	61.76%
	2	TL	50	2	TL	50	2	V3FO	14	41.18%
	3	V3FO	33.334	3	V3FO	33.334	3	C2FH	13	38.24%
	4	C2FH	33.3337	4	C2FH	33.3337	4	TR	13	38.24%
	5	S2FH	33.3337	5	S2FH	33.3337	5	S2FH	7	20.59%
	6	NWMS	15.3853	6	NWMS	14.2864	6	P1B	5	14.71%
	7	P1B	15.3853	7	P1B	14.2864	7	P1F	5	14.71%
	8	P1F	15.3853	8	P1F	14.2864	8	P2B	5	14.71%
	9	P2B	15.3853	9	P2B	14.2864	9	V1FC	5	14.71%
	10	C1FH	14.2863	10	C1FH	14.2863	10	S1FH	5	14.71%
	11	S1FH	14.2862	11	S1FH	14.2862	11	NWMS	5	14.71%
	12	V1FC	14.2862	12	V1FC	14.2862	12	C1FH	4	11.76%
16	1	NWMS	15.3853	1	NWMS	15.3853	1	NWMS	176	21.97%
	2	P1B	15.3853	2	P1B	15.3853	2	P1B	159	19.85%
	3	P1F	15.3853	3	P1F	15.3853	3	P1F	159	19.85%
	4	P2B	15.3853	4	P2B	15.3853	4	P2B	159	19.85%
	5	C1FH	14.2863	5	C1FH	14.2863	5	V1FC	159	19.85%
	6	S1FH	14.2862	6	S1FH	14.2862	6	C1FH	147	18.35%
	7	V1FC	14.2862	7	V1FC	14.2862	7	S1FH	137	17.10%
17	1	TR	50	1	TR	50	1	TR	112	51.85%
	2	TL	50	2	TL	50	2	TL	104	48.15%
	3	NWMS	15.3853	3	NWMS	14.2864	3	NWMS	39	18.06%
	4	P1B	15.3853	4	P1B	14.2864	4	P1B	37	17.13%
	5	P1F	15.3853	5	P1F	14.2864	5	P1F	37	17.13%
	6	P2B	15.3853	6	P2B	14.2864	6	P2B	37	17.13%
	7	C1FH	14.2863	7	C1FH	14.2863	7	V1FC	37	17.13%
	8	S1FH	14.2862	8	S1FH	14.2862	8	C1FH	36	16.67%
	9	V1FC	14.2862	9	V1FC	14.2862	9	S1FH	33	15.28%
18	1	V3FO	33.334	1	V3FO	33.334	1	V3FO	28	33.33%
	2	C2FH	33.3337	2	C2FH	33.3337	2	S2FH	28	33.33%
	3	S2FH	33.3337	3	S2FH	33.3337	3	C2FH	28	33.33%
	4	P3B	25.0005	4	P3B	25.0005	4	P3B	21	25.00%
	5	P3F	25.0005	5	P3F	25.0005	5	P3F	21	25.00%
	6	P4B	25.0005	6	P4B	25.0005	6	P4B	21	25.00%
	7	V2FC	25.0005	7	V2FC	25.0005	7	V2FC	21	25.00%
	8	NWMS	15.3853	8	NWMS	14.2864	8	P1B	12	14.29%
	9	P1B	15.3853	9	P1B	14.2864	9	P1F	12	14.29%
	10	P1F	15.3853	10	P1F	14.2864	10	P2B	12	14.29%
	11	P2B	15.3853	11	P2B	14.2864	11	V1FC	12	14.29%
	12	C1FH	14.2863	12	C1FH	14.2863	12	S1FH	12	14.29%
	13	S1FH	14.2862	13	S1FH	14.2862	13	C1FH	12	14.29%
	14	V1FC	14.2862	14	V1FC	14.2862	14	NWMS	12	14.29%

Table 4.11 – Summary of the results of the two methods and the simulation code for scenarios 13-18.

Scenario		Possible Causes Method I			Possible Causes Method II			Simulation Results		
		Component	Probability (%)		Component	Probability (%)		Component	Occurrence	Occurrence in %
19	1	P3B	25.0005	1	P3B	33.3342	1	P3B	111	37.25%
	2	P3F	25.0005	2	P4B	33.3342	2	P4B	104	34.90%
	3	P4B	25.0005	3	V2FC	33.3336	3	V2FC	104	34.90%
	4	V2FC	25.0005	4	C1FH	14.2863	4	NWMS	55	18.46%
	5	NWMS	15.3853	5	NWMS	14.2863	5	P1B	52	17.45%
	6	P1B	15.3853	6	P1B	14.2863	6	P1F	52	17.45%
	7	P1F	15.3853	7	P1F	14.2863	7	P2B	52	17.45%
	8	P2B	15.3853	8	P2B	14.2863	8	V1FC	52	17.45%
	9	C1FH	15.3851	9	S1FH	14.2862	9	C1FH	49	16.44%
	10	V1FC	15.3851	10	V1FC	14.2862	10	S1FH	46	15.44%
	11	S1FH	7.6931							
20	1	P3B	25.0005	1	P3F	99.9949	1	P3F	118	84.89%
	2	P3F	25.0005	2	NWMS	14.2864	2	P5F	28	20.14%
	3	P4B	25.0005	3	P1B	14.2864	3	NWMS	24	17.27%
	4	V2FC	25.0005	4	P1F	14.2864	4	P1B	23	16.55%
	5	NWMS	15.3853	5	P2B	14.2864	5	P1F	23	16.55%
	6	P1B	15.3853	6	C1FH	14.2863	6	P2B	23	16.55%
	7	P1F	15.3853	7	S1FH	14.2862	7	V1FC	23	16.55%
	8	P2B	15.3853	8	V1FC	14.2862	8	C1FH	22	15.83%
	9	C1FH	14.2863	9	P5F	0.0066	9	S1FH	21	15.11%
	10	S1FH	14.2862	10	P4B	0.0033	10	P4B	14	10.07%
	11	V1FC	14.2862	11	V2FC	0.0033	11	V2FC	14	10.07%
21	1	TR	50	1	TR	50	1	TR	28	50.00%
	2	TL	50	2	TL	50	2	TL	28	50.00%
	3	P3B	25.0005	3	P3B	25.0005	3	P3B	14	25.00%
	4	P3F	25.0005	4	P3F	25.0005	4	P3F	14	25.00%
	5	P4B	25.0005	5	P4B	25.0005	5	P4B	14	25.00%
	6	V2FC	25.0005	6	V2FC	25.0005	6	V2FC	14	25.00%
	7	NWMS	15.3853	7	NWMS	14.2864	7	P1B	8	14.29%
	8	P1B	15.3853	8	P1B	14.2864	8	P1F	8	14.29%
	9	P1F	15.3853	9	P1F	14.2864	9	P2B	8	14.29%
	10	P2B	15.3853	10	P2B	14.2864	10	V1FC	8	14.29%
	11	C1FH	14.2863	11	C1FH	14.2863	11	S1FH	8	14.29%
	12	S1FH	14.2862	12	S1FH	14.2862	12	C1FH	8	14.29%
	13	V1FC	14.2862	13	V1FC	14.2862	13	NWMS	8	14.29%
22	1	P3B	25.0005	1	P3B	33.3342	1	P3B	61	43.57%
	2	P3F	25.0005	2	P4B	33.3342	2	P4B	52	37.14%
	3	P4B	25.0005	3	V2FC	33.3336	3	V2FC	52	37.14%
	4	V2FC	25.0005							
23	1	TR	50	1	TL	100	1	TL	28	100.00%
	2	TL	50	2	P3B	33.3342	2	P3B	11	39.29%
	3	P3B	25.0005	3	P4B	33.3337	3	P4B	10	35.71%
	4	P3F	25.0005	4	V2FC	33.3336	4	V2FC	10	35.71%
	5	P4B	25.0005							
	6	V2FC	25.0005							

Table 4.12 – Summary of the results of the two methods and the simulation code for scenarios 19-23.

The occurrence of the components in the simulation code gives a measure of the probability for a failure to have occurred in a particular scenario. This measure is very similar to the posterior probability and this is proved by the fact that the ranking of the components in the BN methods and in the simulation results are similar. It can be concluded that the BN method is able to find the potential causes of a fault in the system and the posterior probability gives an efficient way to produce the most probable causes among the potential ones.

4.7 Summary

This chapter has described a general method for the application of BNs to the fault diagnostic of a dynamic system. The method is applied to the example of the water tank system. By introducing the sensor patterns, the analysis is now able to study the scenarios that were

not observable in the method previously shown. The effectiveness of the diagnosis is validated by a simulation code that generates the scenarios and the actual causes of the system. With respect to FTA, the BN diagnosis is faster and more concise and all the advantages that were pointed out in the previous chapter are confirmed.

Two methods were developed. In method I the root nodes representing the component failures link to the nodes that represent the patterns of 4 of the 5 sections of the system. Method I is then improved introducing the links to patterns 5 in the system BN, that is, considering the tank level as a consequence of the failures in the system. This modification makes the BN able to find all the actual causes of all scenarios and it does not indicate any false cause. The method can be considered, for this example, 100 % accurate.

Even though the simulation code that validates the method only considers up to 3 failures in the system, this is not restrictive as only one scenario was left out of the simulation. The actual failures of this scenario were found manually and then compared with the method.

A limitation of the method is represented by the assumption that the failures have occurred before the system is started and this can be restrictive to those situation where a failure occurs after some time with respect to other failures. However, this problem is beyond the scope of the research reported in this thesis.

The method should be used for a larger and more complex system to prove its applicability.

Chapter 5

Application of the Diagnostic Method to the Fuel Rig System

Introduction

In this chapter, the diagnostic method described in chapter 4 is applied to a fuel rig system. As the fuel rig is considerably larger than the water tank system, the method is modified and adapted to deal with the increased complexity of the diagnosis. This application should prove that the method can be applied to systems that are more complex with respect to the water tank system.

5.1 Fuel Rig system Description

The fuel rig system is based on a real aircraft fuel system, the Advance Diagnostic Test-bed (ADT) is located at the System Engineering Innovation Centre (SEIC) in Loughborough (figure 5.1).
The facility consists of a number of tanks, pumps, valves, flow meters, level sensors, and other components which form the mechanical and electrical parts of a modern aircraft fuel system ([48], [49]). The instrumentation can be connected and used with different configurations and it is possible to induce some types of faults into the rig. For the purpose of our research a particular configuration has been chosen in which the system comprises three fuel tanks: a wing, a collector and a main tank. Fuel is fed from both the main and the wing tank into the collector tank, and, from the collector tank, it is pumped into the engine. Figure 5.2 shows a schematic of the system. Each tank has two fuel stream lines, with two pumps. Only one line is used at the time and the other represents a back up in the case of a failure.
Fuel is fed into the engine only from the collector tank. When the level in the collector tank

Figure 5.1 – Photo of the Advanced Diagnostic Test-bed (ADT).

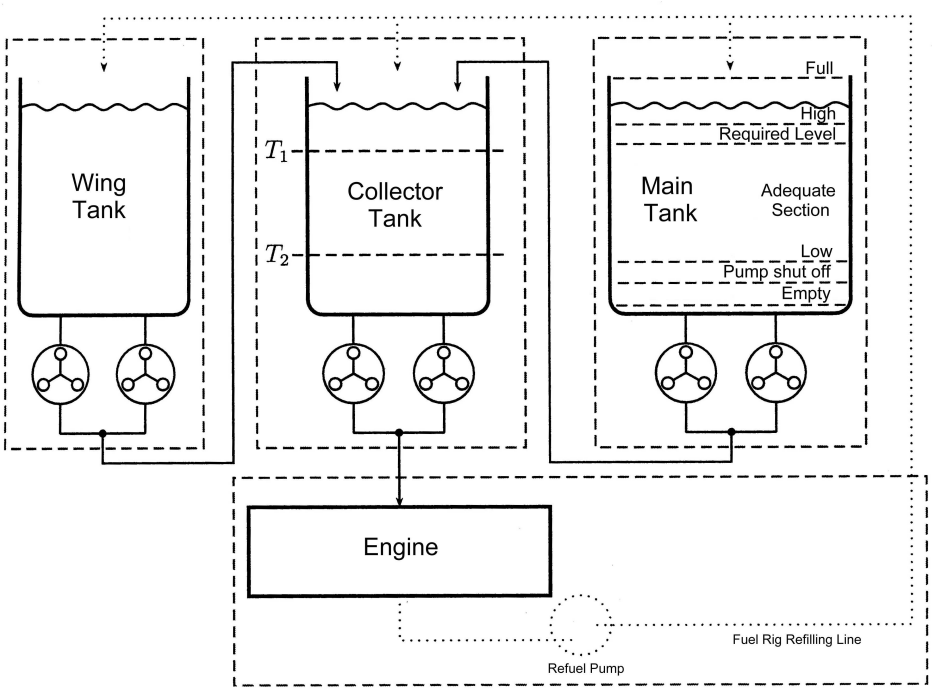


Figure 5.2 – Schematic of the fuel rig configuration.

drops below threshold T_1, fuel is transfered from the wing tank into the collector. When the wing tank is empty or if the level in the collector tank reaches the limit T_2, fuel is pumped from the main tank into the collector tank. Since it is assumed that the flow rate of fuel pumped into the collector tank equals the fuel out of the tank, the level in the collector tank in the normal operation should never increase. The engine is represented by a larger tank located underneath the three tanks and it is possible to refill the tanks from the engine through a refilling line.

5.1.1 Components Description

The main tank is represented in figure 5.3. The two fuel stream lines are labelled as L1 and L2. The following components are contained in line L1: a pump (PP0110), a powered isolation valve (IVP0110), a controller (CT0110), a back pressure valve (BP0110) and 6 sections of pipe work (P0101-P0106).

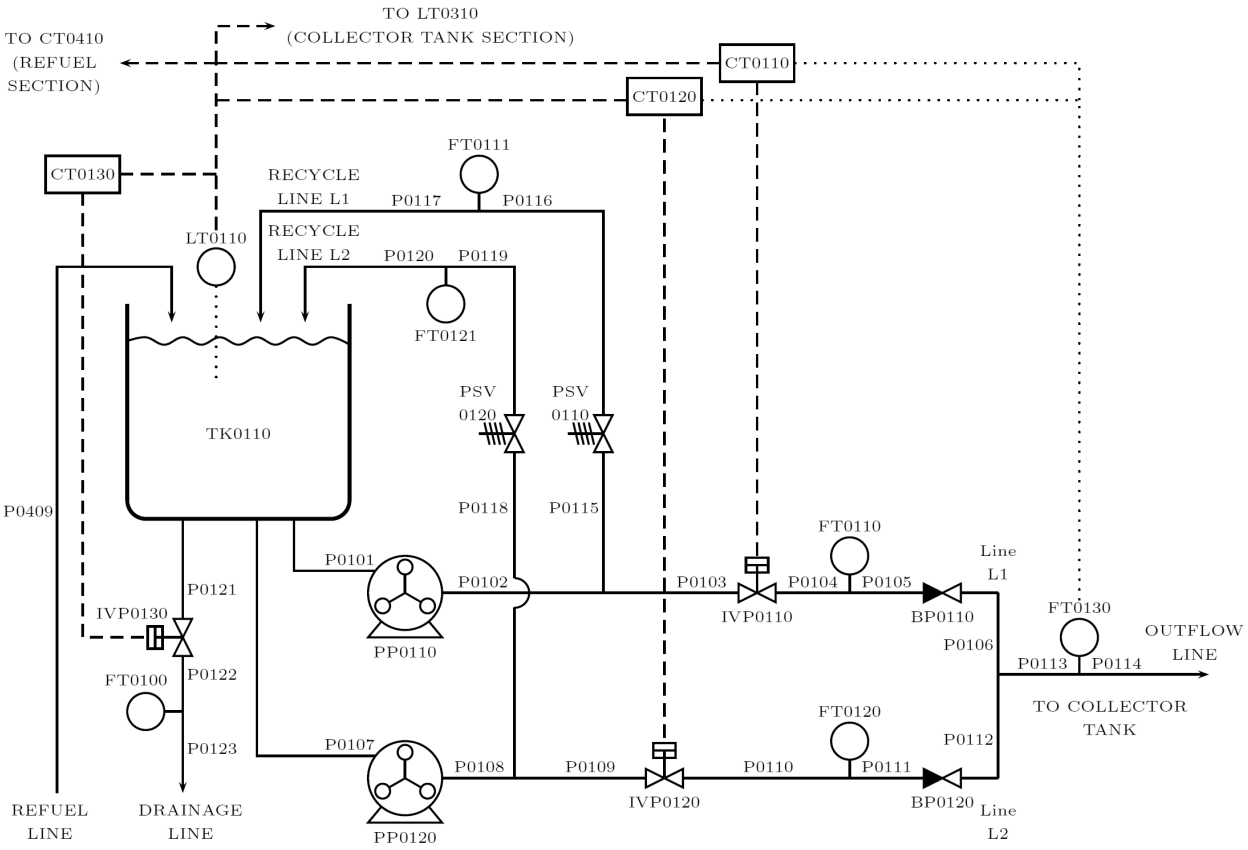


Figure 5.3 – Main Tank of the Fuel Rig System.

Line L2, which is used as a back up line, has the same components, labelled as: PP0120, IVP0120, CT0120, BP0120 and P0107-P0112. The two lines join together into the main outflow line of the tank, which contains two components: pipe sections P0113 and P0114. When the engine is started, both pumps in the lines are activated but, for the line on standby, the powered isolation valve stays closed so that fuel is re-circulated back into the tank via a recycle line. There are two recycle lines, labelled as recycle lines L1 and L2. Recycle line L1 contains the following components: a pressure relief valve (PSV0110) and 3 sections of pipe work (P0115-P0117). Similarly, the components of the recycle line L2 are PSV0120 and P0118-P0123. A drainage line is also located at the bottom of the tank and it is used to simulate dumping of fuel from the aircraft, its components are: a power isolation valve

(IVP0130), a controller (CT0130) and 3 pipe sections (P0121-P0123). Each tank has 7 sensors, one of which is a level transmitter that measures the level of fuel in the tanks. The others are flow sensors that measure the flow rate at different points in the pipes. In the main tank, LT0110 is the level sensor, FT0110 and FT0120 are located in lines L1 and L2 respectively, FT0111 and FT0121 are in the recycle lines, FT0130 measures the flow rate in the outflow line and, finally, FT0100 is at the drainage line.

The Wing and Collector Tanks have similar components to the ones described for the Main Tank. The numbers used to identify the components for each tank are as follows:
Main Tank 01**,
Wing Tank 02**,
Collector Tank 03**.
For example, the power isolation valves that belongs to line L1 and line L2 of the wing tank are labelled as IVP0210 and IVP0220 respectively.

Several failure modes are considered for each component of the system. For example, pipe work sections have 4 failure modes: *Blocked*, *Fractured*, *Partially Blocked* and *Leaking*. A blockage in a pipe is defined as a failure that would cause fuel flow to stop completely, a fracture would cause all the fuel to be lost from the pipe, while a leak represents a partial loss. A partial blockage would allow only half of the fuel to pass through the pipe. Table 5.1 lists all component failures of the system. Considered that each tank has 35 components, that the pipes, the valves and the pumps have 3 or 4 failure modes while the controllers have 2 failure modes, there are in total 396 component failures in the system.

Component	Failure Modes
Pumps PP****	1 Fails On - 2 Fails Shut Off - 3 Fails Mechanically - 4 Fails Leaking
Valves IVP****	1 Fails Open - 2 Fails Blocked - 3 Fails Partially Blocked - 4 Fails Leaking
Valves PSV****	1 Fails Open - 2 Fails Blocked - 3 Fails Partially Blocked - 4 Fails Leaking
Valves BP****	1 Fails Blocked - 2 Fails Partially Blocked - 3 Fails Leaking
Controllers CT****	1 Fails True - 2 Fails False
Pipes P****	1 Fails Blocked - 2 Fails Fractured - 3 Fails Partially Blocked - 4 Fails Leaking

Table 5.1 – Component failure modes.

5.1.2 System Operating Modes

The system has two operating modes: ACTIVE and DORMANT. In the DORMANT mode, there is no fuel transfer and the pumps are shut down. In the ACTIVE mode, all pumps in the tanks are activated and fuel is fed from the collector tank into the engine. The transfer

of fuel from the main and the wing tanks depends on the level in the collector tank.

The level thresholds are: *Empty* (E), *Pump Shut Off* (PSO), *Low* (L), *Adequate Section* (AS), *Required Level* (RL), *High Level* (HL), and *Full* (F) (as indicated in figure 5.2). In the collector tank, the adequate section has two additional thresholds indicated as T_1 and T_2. Their use is to specify whether fuel is pumped into the collector from the main or the wing tank. If the level of fuel in the collector tank is above T_1, then the flow from the wing tank into the collector tank is stopped. If the level drops below threshold T_1, then fuel is pumped into the collector tank in order to keep the tank replenished. Fuel from the main tank is allowed into the collector tank if the level drops below threshold T_2. In the case the level in all tanks drops below the *Pump Shut Off* limits, then the pumps are shut off to prevent damage.

If dumping of fuel is simulated on the rig, the collector tank is drained to *Low*, while both the main and the wing tank are drained to PSO. The amount of fuel left in the collector tank is used for the landing.

5.2 Fuel Rig Diagnostic System

In chapter 4, a method is described for the diagnostics of dynamic systems. The following is a brief summary of the procedure:
First, the system is divided into sections, each section has the capability to effect a system process variable and contains a sensor that monitors the trends of the variable of interest. The possible trends of the monitored variables are studied and they are correlated to the states of the section. In this way, specific patterns are identified for each possible section failed state. Non-coherent FTs are then built to represent the causality relations between the failed states of the sections and the component failures. The FTs are converted into BNs and these are connected together in a network that represents the system and by which all system scenarios can be analysed. The trends observed in the sensors are also included in the structure of the BN so that evidence can be introduced to the networks when a particular sensor observation is made. The posterior probability is calculated for the component failure events in all scenarios and the list of the component failures whose posterior probability has increased with respect to their prior probability is derived. This gives the list of potential causes for all system scenarios. The method is implemented in two stages: the system modelling and preparation stage and the BN development stage. In the first stage all information known about the system is collected and the system sections, states and scenarios are identified. FTs and BNs are then built to model the fault causality

of the system. After this, the system is ready to perform the diagnostics. The details of the application of the method to the fuel rig system is described in the next two sections. Some adaptations are necessary, such as the division into sub-systems due to the size of the problem or the use of identical BNs for redundant parts of the system.

5.3 System Modelling and Preparation stage

For this stage, the following tasks are performed:
1. the system is divided into sub-systems and, in turn, the sub-systems are divided into sections,
2. the section states are identified,
3. all possible sensor patterns for each section variable are identified,
4. the system scenarios are listed.

5.3.1 System division into sub-systems and sections

The fuel rig system comprises 105 components, 18 flow sensors and 3 level sensors. Due to its size, it was necessary to divide it into sub-systems. The diagnostics is carried out by first identifying the faulty sub-systems and then focusing on them individually to detect the causes. It is natural to divide it in 3 sub-systems: the main tank, the wing tank and the collector tank. There could be an extra sub-system, consisting of the refuelling part, that was not considered in this analysis. Each tank is then analysed individually, so that the division into sections, the identification of the section states and patterns is performed three times. As the tanks have a very similar structure, we will focus here in details only on the analysis of the main tank. The diagnostics should also take into account the operating modes of the system, as different sensor readings are expected in different modes. We will only consider the ACTIVE mode.

The main tank is divided into 6 sections, relative to the 6 flow transmitters located in the pipes: fuel line L1, fuel line L2, recycle line L1, recycle line L2, the drainage line and the outflow line. An extra section, the tank itself, can be considered in correspondence to the level sensor, but the state of the level in the tank depends on the states of the other sections, as the level in the tank is a consequence of the fuel flow in the pipes.

5.3.2 Identification of section states

Once the system is divided into sections, accounting for the sensor outcomes and the component failures, one should be able to identify for each section a number of possible states.

These will include the working state and each failed state.

In the main tank, for each section, four possible states are identified: *Working* (W), *High Flow* (HF), *Low Flow* (LF) and *Partial Flow* (PF). *High Flow* occurs if unexpected full flow is observed in a line when no flow was expected. Low flow occurs when, vice versa, no flow is observed and full flow was expected. Finally, partial flow is the state corresponding to a situation when a partial blockage or a leak in a pipe work or in a valve causes unexpected partial flow through the line. For the purpose of building the BNs, partial flow is studied by considering it separately for the situation when flow or no flow is expected. Therefore there will be two separate FTs, one whose top event is *Partial Flow* when flow is expected and one for *Partial Flow* when no flow is expected.

5.3.3 Identification of sensor patterns

For each section state, the sensor outcomes should show a number of possible dynamic behaviours for the monitored variable. The readings can be grouped together to form possible trends for the variable. For each section, all possible variable patterns are identified.

In the main tank, the possible patterns are identified for each of the 6 sensors. Figure 5.4 shows the possible flow patterns for sensor FT0110 in the fuel line L1.

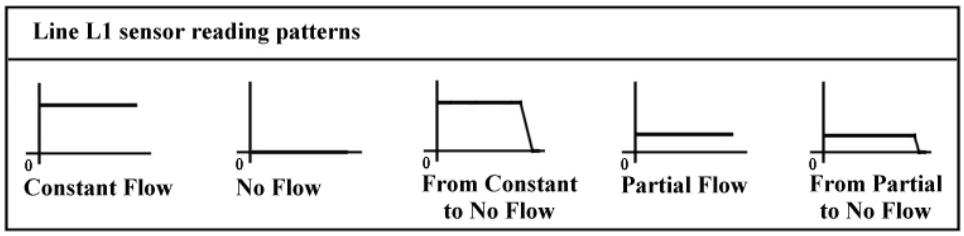


Figure 5.4 – Reading patterns for sensor FT0110 in the main tank.

Whether a pattern is expected or not depends on the particular phase the system is in. The system phase depends on the level of fuel in the collector tank. To facilitate the analysis and to be able to tell which pattern is expected, it in necessary to divide the ACTIVE operating mode into phases. For the entire system, 6 phases are identified. In phase 1 the level in the collector is between threshold T_1 and the required level (RL), so fuel is fed into the engine from the collector tank and there is no flow into the collector tank from either the main or the wing tank. The system is in phase 2 when the level in the collector tank is at T_1, here the fuel starts to be pumped from the wing tank into the collector tank and from the collector into the engine. This phase is terminated when the level in the wing tank reaches the PSO

limit and the system goes into phase 3, in which the level in the collector tank is between thresholds T_1 and T_2. At this point, again there is no transfer of fuel from the main and wing tanks into the collector tank. Fuel is pumped from the main into the collector tank when the level in the collector tank reached threshold T_2 in phase 4. Phase 5 starts when the level in the main tank reaches limit PSO, at this stage, the collector tank is emptied to the PSO limit. The final phase, phase 6, is reached when all tank levels are at PSO limit and no fuel is transferred into the engine. Table 5.2 shows the phases of the system in the active mode, with the level in the collector tank indicated as L_c.

Phase	Level interval
1	$T_1 < L_c < RL$
2	$L_c = T_1$
3	$T_2 < L_c < T_1$
4	$L_c = T_2$
5	$PSO < L_c < T_2$
6	$L_c = PSO$

Table 5.2 – Phases of the ACTIVE operating mode.

Figure 5.5 shows the expected sensor readings of the main tank sub-system for phase 4 of the ACTIVE operating mode.

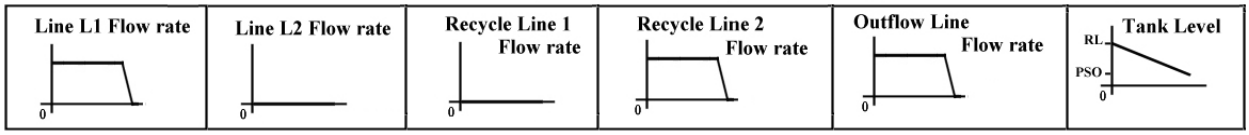


Figure 5.5 – Expected sensor patterns for the sensors in the main tank when the system is in phase 4 of the ACTIVE operating mode.

In phase 4, the fuel is transferred from the main tank into the collector tank and from the collector into the engine. As a consequence, the level of fuel in the collector tank remains constant and it decreases in the main tank from threshold TL to PSO. Fuel is expected to pass along line L1 and it should be recirculated back through the recycle line L2. Therefore no flow is expected in the recycle line L1 and in the line L2. The flow in the outflow line is the same as the one from line L1. The tank level decreases from TL to PSO and, when it reaches PSO, the pumps are stopped and flow in line L1, in the outflow line and in the recycle line L2 goes from flow to no flow.

The flow patterns among all scenarios in all phases for the flow sensors can be distinguished in 3 types, *Flow*, *No Flow* and *Partial Flow*. This simplification can be done if one does not consider the phase transitions, that is the patterns that are observable when the system goes from one phase to the other. For example, if the system goes from phase 4 to phase

5, the expected pattern for the flow rate in line L1, sensor FT0110, should be *From Flow to No Flow*. If the phases are distinguished, then the expected pattern in phase 4 results *Flow* while the expected pattern in phase 5 is *No Flow*. With this assumption, the possible patterns for sensor FT0110 are shown in figure 5.6.

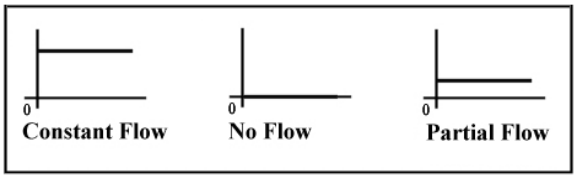


Figure 5.6 – Possible patterns for the sensor in line L1 excluding the ones at the stage transitions.

The same patterns can be considered with this assumption for all the other sensors in the main tank and in the other sub-systems.

5.3.4 System scenarios

A system scenario is defined as a combination of sensor patterns that can be observed on the system. Considering that each of the 6 sections in the main tank can follow three patterns, there are 729 possible combinations of patterns that can generate possible scenarios for the main tank. This number becomes even larger if all three sub-systems are considered. However, using BNs it is not necessary to identify which of the patterns combinations are the actual scenarios for the system as the diagnosis can be performed on line when a scenario is observed. Moreover, knowing the scenarios is not required for building the BNs. In previous FT based methods, listing and studying the possible scenarios was necessary for the calculation of the prime implicants.

Obtaining the scenarios can be useful for the validation of the method, since the actual and the potential causes identified can be compared for each scenario and the accuracy of the method can be assessed. This will be done later on in the chapter using a simulation code for the system, in a similar way as was performed for the water tank system.

5.4 Bayesian Networks Development

Four tasks are performed at this point:
1. building non-coherent FTs for the failing states of the sections,
2. converting the FTs into BNs,
3. connecting the BNs to form a unique BN for each sub-system,

4. creating a BN that models the entire system and that is used to understand which sub-system is likely to be faulty.

5.4.1 Non-coherent Fault Trees construction

Non-coherent FTs are built for all deviating states of the sections. Non-coherent logic includes both failed and working states of the components so that NOT gates are included in the FTs. In the main tank, for example, section 1 has 3 deviating states: *High Flow*, *Low Flow* and *Partial Flow*. Because the causality of the failure modes depends on the working phases of the system, the FTs are differentiated for the different phases. For example, for line L1 and its state *Partial Flow*, two FTs are built, one corresponds to phase 4, when line L1 is expected to provide fuel, and the other is for all other phases, in which there should not be fuel transfer through line L1. In total there are 4 non-coherent FTs built for line L1 with the following top events:

- *High flow* (when no flow is expected),
- *Low flow* (when flow is expected),
- *Partial Flow* when flow is expected,
- *Partial Flow* when no flow is expected.

The FTs construction follows the same process used for the water tank system. However the FTs are much larger for two reasons. One is the increased number of components and the other is the number of failure modes considered for each component.

5.4.2 Conversion of the FTs into BNs

The FTs are converted into BNs. All FTs representing the causes of the deviating states of the same section are converted in a unique network. In a similar way as for the water tank system, the top event is converted into a node representing the deviating section states, while the basic events are root nodes in the networks. The basic events in the FT are component failures. As each component can have many failure modes, the number of nodes and links in the BN can be minimised by creating for each component a node whose states are the working plus the failed states.

For the main tank, a BN is created for each section in all system phases. The choice of different BNs depending on the phase of the system is motivated by the fact that it is not necessary to include the BN for which evidence will not be given. For example, if the system is in phase 4, flow is expected at the sensor FT0110 in section 1, therefore the BNs that

will be considered are the ones that relate to the FTs whose top events are *Low Flow* and *Partial Flow* when flow is expected. It would still be possible to create BNs that include all possible working modes, but they are larger and they carry no extra information for a particular system phase.

The section relating to the drainage line remains unchanged for all phases of the ACTIVE mode, as no flow is always expected. In this case, two FTs are included in the network: *High Flow* and *Partial Flow*. The top events are converted into the fault nodes (at the top in figure 5.7), and the components of the section are the root nodes at the bottom. There are only 5 components in the drainage line: 2 sections of pipework, a valve and a controller. Figure 5.7 shows the structure of the BN.

The next figure, figure 5.8, shows the BN for fuel line L1 in phase 4 of the ACTIVE mode. Two FTs are included in this BN and it is already quite large. If all deviating states would be included, representing all phases of the system, four FTs would be converted into the BN and this would double its size. Figures 5.9 and 5.10 represent separately the two subnetworks that form figure 5.8. These are included for a better reading.

The choice of representing the networks now *upside down* with respect to the previous ones is simply motivated by the fact that having the fault nodes at the top and the component failures at the bottom with the links pointing upward can make the conversion and the comparison with the original FTs easier. However this graphical scheme does not make any difference to the logical meaning and the probability structure of the BNs.

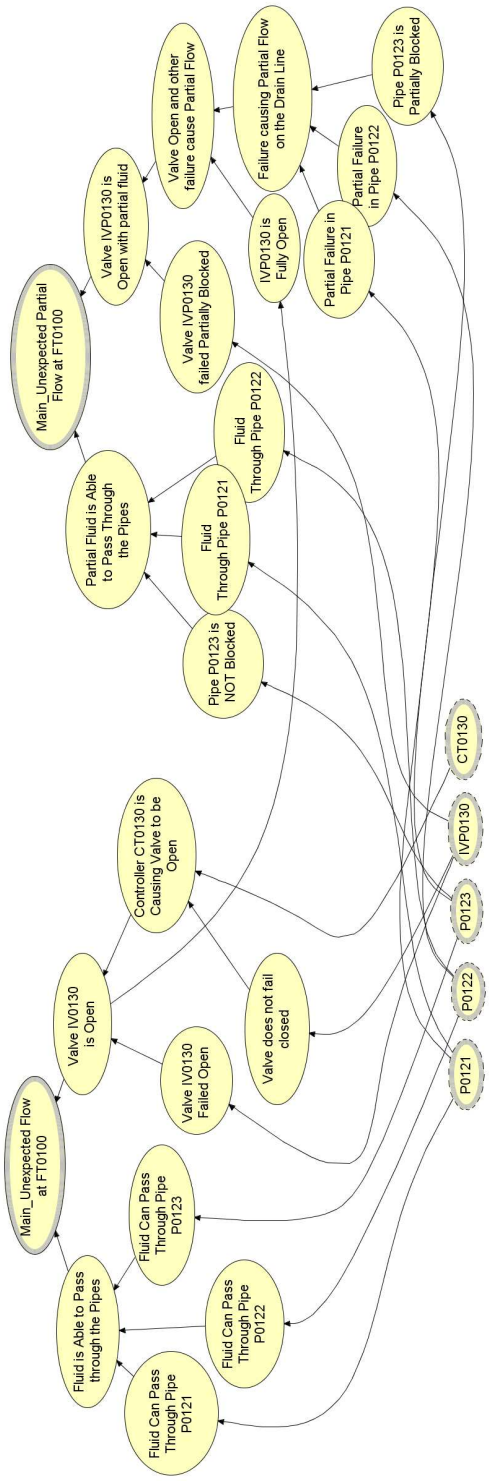


Figure 5.7 – BN for the drainage line section in the main tank.

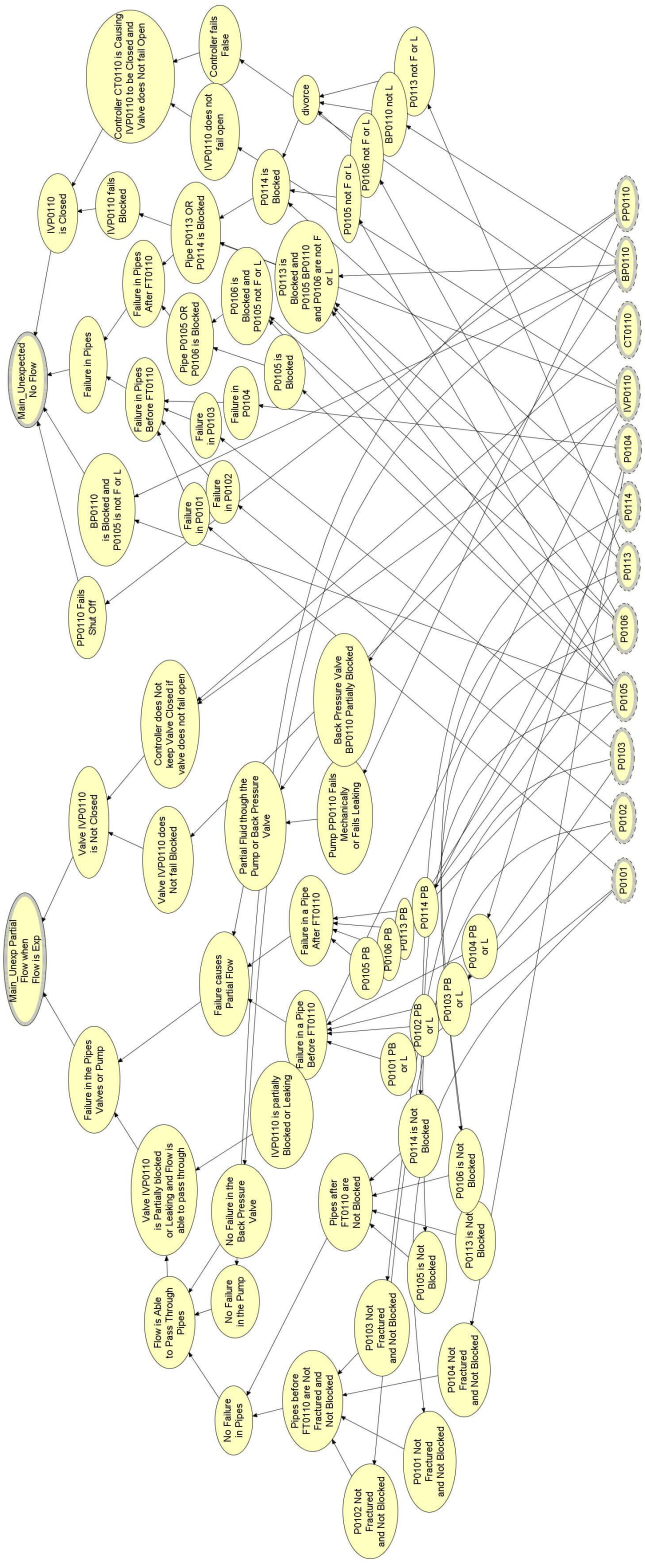


Figure 5.8 – BN for line L1 section in the main tank for phase 4.

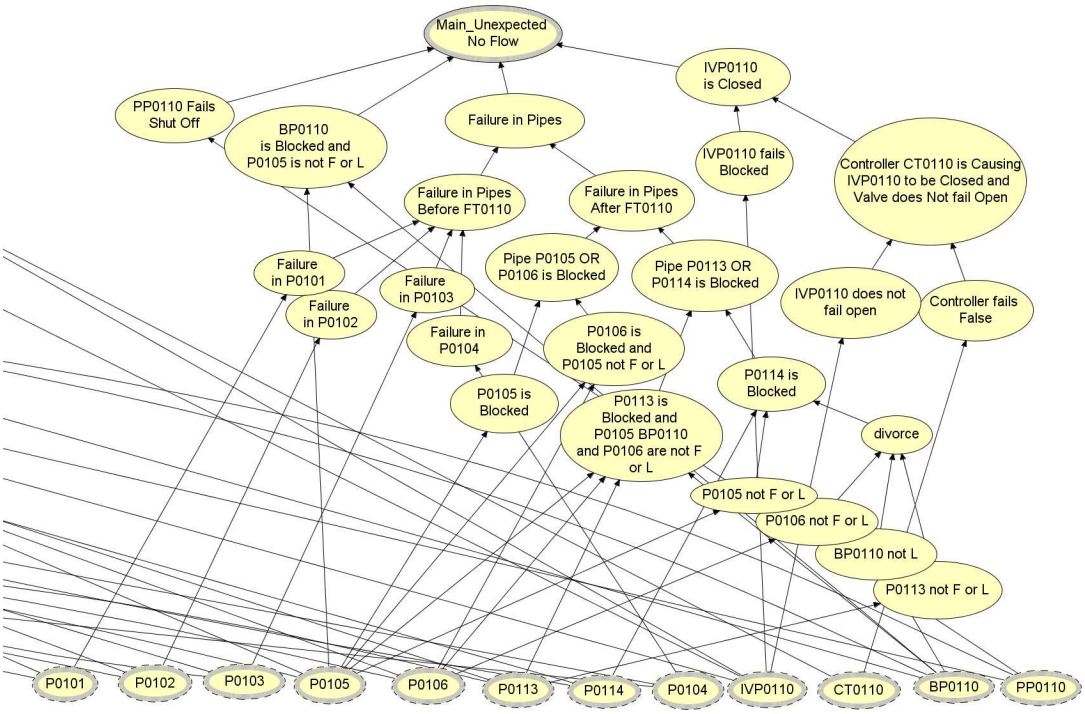


Figure 5.9 – BN for the event *No Flow* in line L1 of the main tank in phase 4.

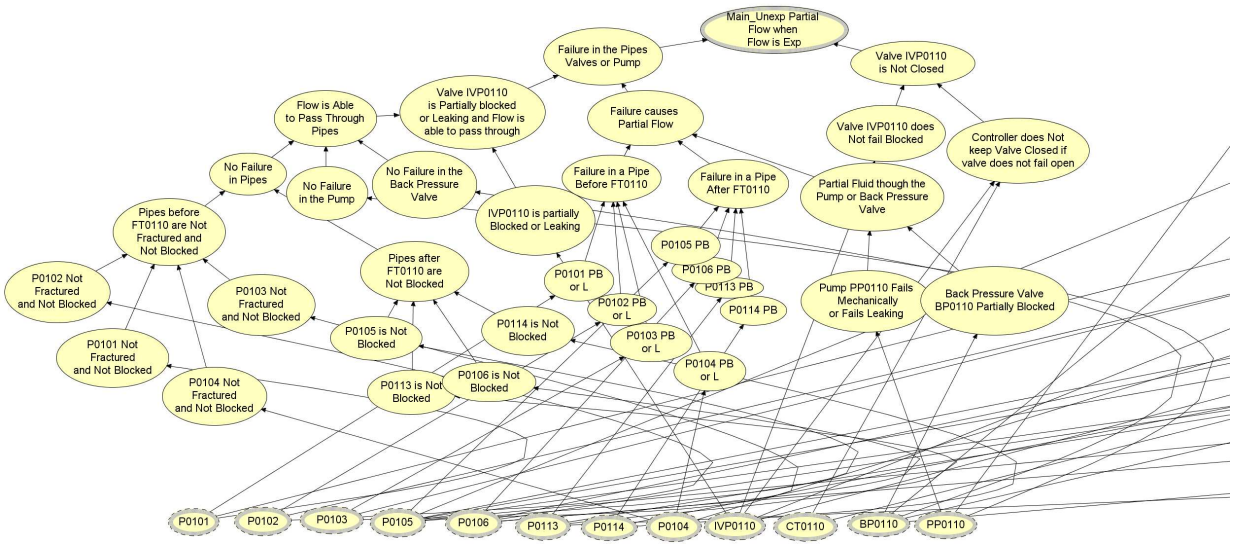


Figure 5.10 – BN for the event *Partial Flow* in line L1 of the main tank in phase 4.

Appendix B contains all the BNs for the main tank.

Redundant Sections

When building the FTs and BNs for the fuel rig system, it was clear that some of their logic structure, which models how the component failures cause the symptoms, are repeated many times. For example, line L1 and line L2 are very similar, as well as recycle line L1 and recycle line L2. This is true for the sections that provide the system with redundancy, but it can be observed for the entire sub-systems. BNs allow these similarities to be used in order to avoid a repetition of the same logic. It is the same principle as for the repeated basic events in the FTs that appear only once in the BN structure.

For example, the BNs relating to the recycle line L2 has the same logic structure as the BN for recycle line L1 with the replacement of the following components:

- P0107 instead of P0101,
- P0108 instead of P0102,
- P0118 instead of P0115,
- P0119 instead of P0116,
- P0120 instead of P0117,
- PP0120 instead of PP0110,
- PSV0110 instead of PSV0120.

The sensor FT0121 in the recycle line L2 has the same role as sensor FT0111 in recycle line L1. As all BNs representing the sections belong to a same class collection, a BN for recycle line L2 can be created including the BN for recycle line L1 as in figure 5.11. The fault nodes at the bottom representing the deviating sensor states of the recycle line L2 are linked to the fault nodes in the external BN for the recycle line L1. Similarly the input nodes representing the components in the recycle line L2 link to the components belonging to recycle line L1. In this way, the input nodes assume the same probability as for the nodes in the external BN. This procedure can be done for the sections of the system or for entire sub-systems. For example, the main tank in phase 4 has the same expected behaviour as the wing tank in phase 2. The BN created for the main tank sub-system for phase 4 will be identical to the BN for the wing tank sub-system for phase 2 with the replacement of the correct components and the sensors labels. This will be shown in more details when all BNs relating to the sections are connected together.

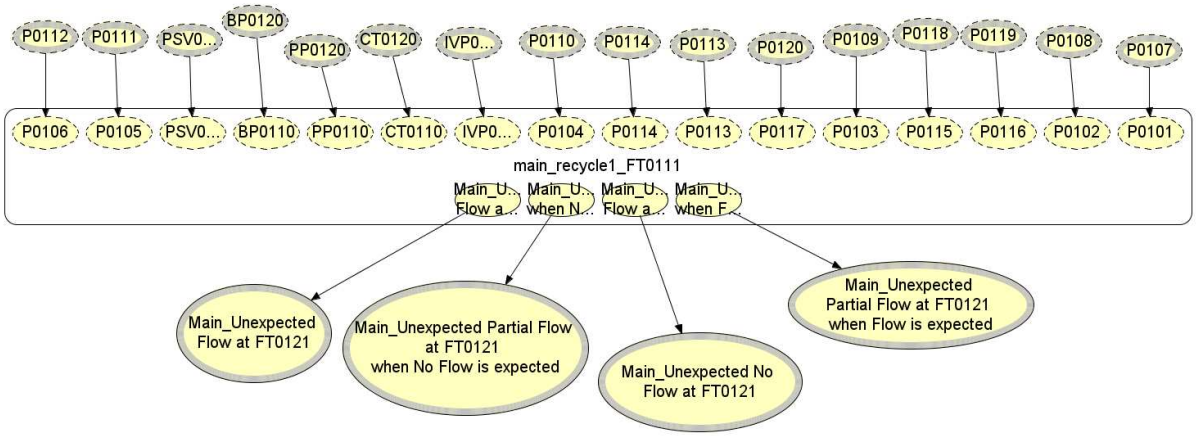


Figure 5.11 – BN for recycle line L2 created using the BN for recycle line L1.

5.4.3 Connecting the BNs

As for the water tank system, the BNs relating to the separate sections are connected together creating a class collection for each sub-system. A BN *master* represents the entire main tank. The BNs relating to the sections of the main tank are visible from the network master. This is represented in figure 5.12.

Figure 5.13 visualises how the input nodes of the system BN link to the external BN and the output nodes in the section BNs link to the patterns nodes at the bottom of the figure. It is also possible to see how two patterns nodes, representing line L1 and line L2 link outside the network to the section BN representing the outflow line. This was done to reduce the size of the BN for the outflow line as its patterns depend on the patterns in line L1 and L2. Therefore, instead of repeating the BN structure for the deviating patterns in the two fuel lines, two input nodes are created representing the behaviour in line L1 and line L2. The size of the BN for the outflow line becomes in this way much smaller.

The 6 nodes at the bottom of the graphical representation of the BN represent the patterns of the sensor readings. Each of them has 3 states: *Flow*, *No Flow* and *Partial Flow*. Evidence is given to them when the sensor readings are observed and the posterior probability is calculated to identify the potential causes. A BN is obtained in this way for every sub-system in each of the operating phases. Connecting all of them to form a BN for the entire system makes the problem size very large. Instead, a BN can be created for the system to understand which sub-system is faulty and then the faulty sub-system can be studied separately to find the potential causes.

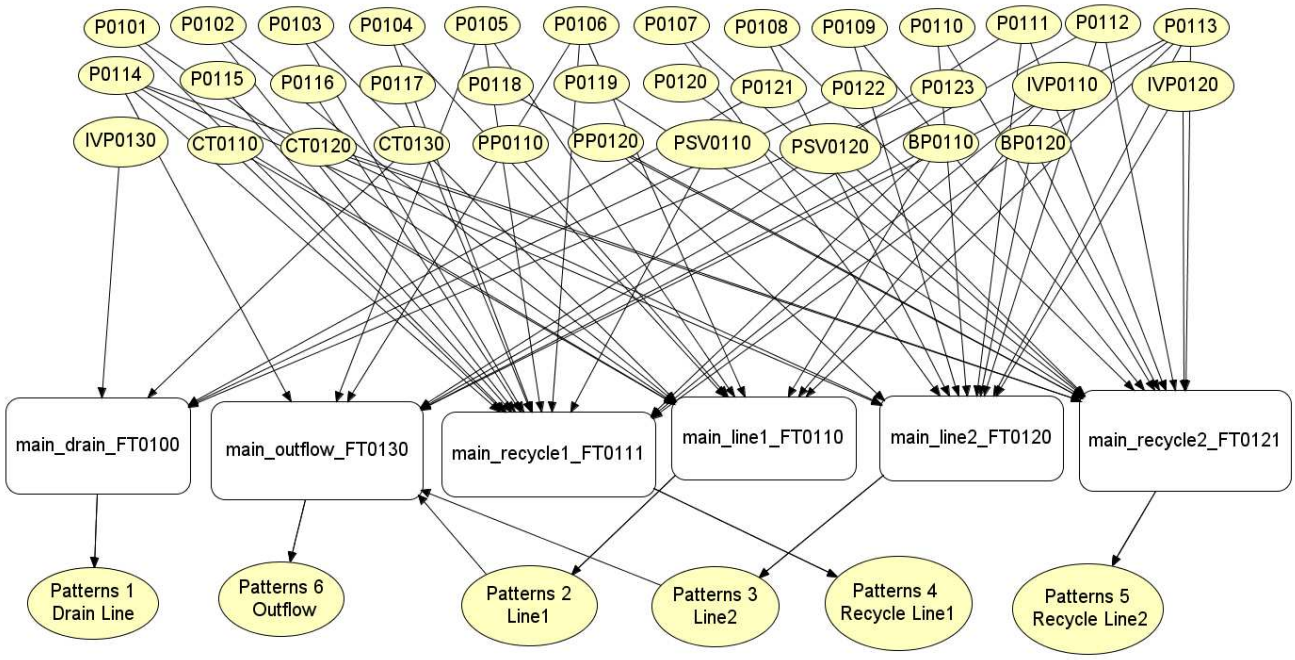


Figure 5.12 – BN for the main tank in phase 4.

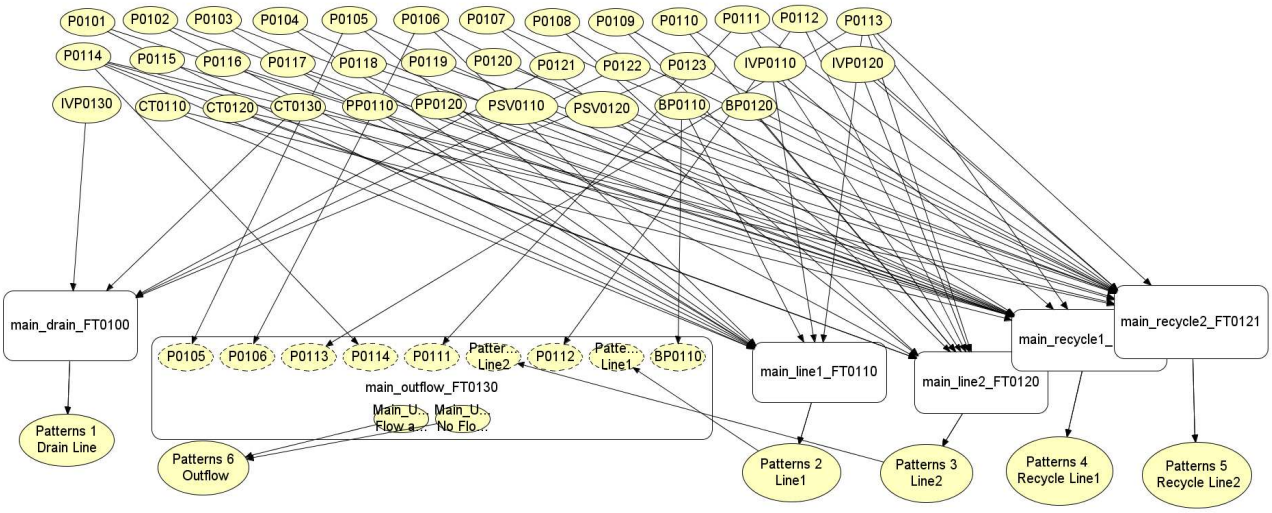


Figure 5.13 – BN for the main tank in phase 4 with the visualization of the BN for the outflow line.

A BN is created for each sub-system. In many cases, sub-systems can be very similar. For the fuel rig system, the main tank and the wing tank have the same components. The functioning of the main tank in phase 4 of the ACTIVE mode is the same as for for wing tank in phase 2. When creating the BN for the wing tank, one can use the BN for the main tank substituting the component nodes and the sensor nodes. This can be done by simply creating a BN that links to the main tank BN. The root nodes for the components in the wing tank link to the input nodes of the main tank and the fault nodes representing the sensor readings in the main tank link to the fault nodes in the wing tank. Figure 5.14 represents the BN for the wing tank created with this procedure.

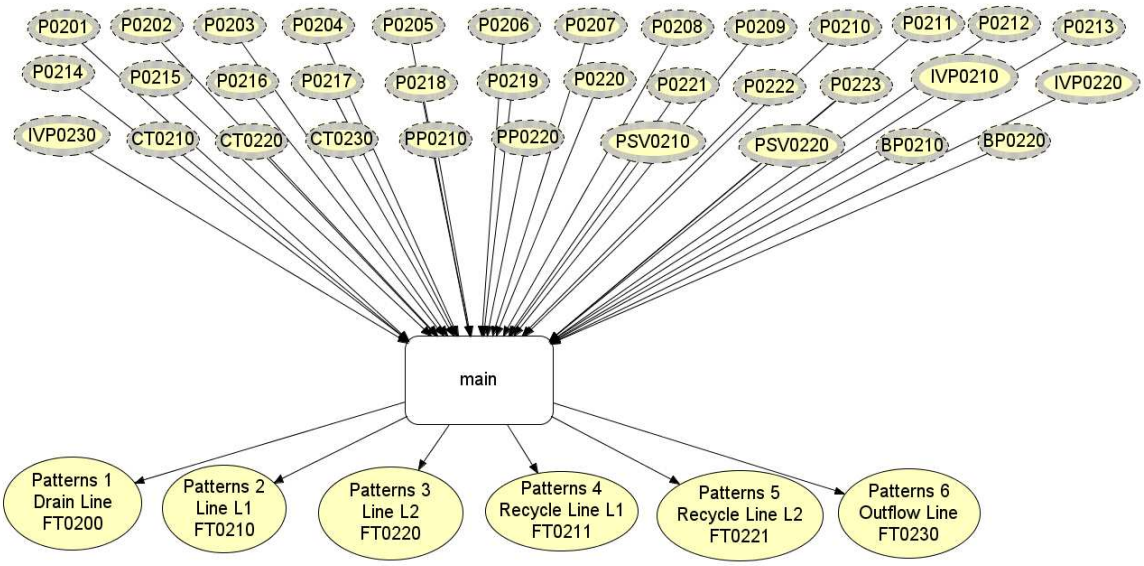


Figure 5.14 – BN for the wing tank created using the BN for the main tank.

The fact that entire sub-systems have similar functioning and the same types of components is something that is common to many types of system. This characteristic of the method applied to the fuel rig can therefore be generalised.

5.4.4 Creating the system BN

Once the BNs are created for each of the sub-systems, a BN is constructed to model the system, this will be referred to as the system BN. For each sub-system one or more sensors are selected. These sensors should give an idea of the state of the sub-system, that is, they should be able to tell if a sub-system is working or failing. The number of sensors that can identify the state of a sub-system is generally smaller than the total number of sensors, which are located at different points for detecting the state but also to identify the faulty components. The system BN is created with two types of nodes, the nodes that represent the

sub-systems and the nodes that represent the sensor patterns. The first link to the second as the states of the sub-systems influence the sensor patterns.

Figure 5.15 represents the BN that models the fuel rig system. This is built to identify which sub-system is more likely to be faulty. It is not necessary to observe all sensors in the system to do so, the sensors that are used are only 6: the flow sensors in the three outflow lines and the three level sensors in the tanks.

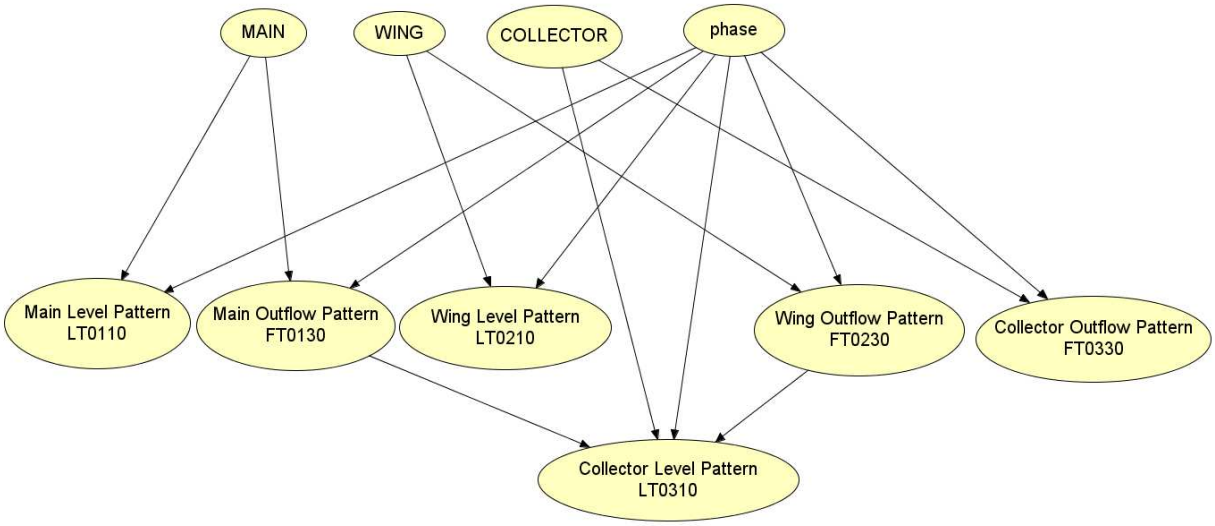


Figure 5.15 – BN for the entire fuel rig system.

Three of the four root nodes at the top of the network are labelled as: MAIN, WING and COLLECTOR. These nodes have 2 states: *working* and *faulty*. The extra root node, *phase*, is a node with 6 states and it represents the phases of the system. The 6 nodes at the bottom represent the sensors mentioned above and they are all, except the one for LT0310, connected to the node phase and to their corresponding sub-system nodes. For example, the node representing the level in the main tank has two node parents, MAIN and phase, and it has 3 states: *Constant*, *Increasing* and *Decreasing*. The state of the level in the main tank depends on the state of the tank itself and it also depends on the phase the system is in. If, for example, the main tank is working and the system is in phase 4, then the level in the main tank is decreasing. The node for the level in the collector tank, labelled as *Collector Level Pattern LT0310*, has two extra parents which are the nodes representing the flow sensors in the outflow lines of the main and wing tank. This is because the behaviour of the level in the tank depends on the state of the tank itself, on the phase of the system and on the flow that enters the tank from the other two tanks. If for example, the main tank and the wing tank are faulty and flow enters the collector tank from both of them, the level in the collector tank will be increasing in all phases of the ACTIVE operating mode. This

BN can be used to determine if a fault has occurred in the system and which sub-system, or sub-systems, have caused it. This is done giving evidence to the 6 sensor nodes plus the node *phase* and calculating the posterior probability of the root nodes representing the sub-systems. If the posterior probability of the *faulty* state of one these nodes is greater than 50 %, then the sub-system BN for that particular phase can be analysed. The reason the posterior probability should be greater than 50 % is purely because the prior probability for these nodes is set as 50% for each of their states.

5.4.5 Diagnostic System application

The diagnostic system is applied in two steps. First, the BN that represents the entire system is used to determine whether the system is faulty or not and which of the sub-systems could be the cause of the fault. Then the sub-systems that were identified in this way are considered in turn by observing the patters from their sensors and calculating the posterior probability for their component failures. The diagnosis application is shown with an example. When the sensors that are monitored on the entire system show a deviation from the expected behaviour, the system is found faulty. We consider a particular situation where the sensors have the states represented in figure 5.16.

These probabilities correspond to the situation when in phase 2 of the active mode, the fuel should be passed from the wing tank into the collector tank. Node *phase* has evidence 100% to its state *2*. Because of the assumption that the amount of fuel entering the collector tank equals the fuel leaving the collector tank, the level of fuel should stay constant. From the observation of sensors outputs, there is no flow through the outflow line of the wing tank (evidence 100 % to the state *No Flow* of node *Wing Outflow Pattern FT0230*) and the level in the wing tank is decreasing (evidence 100% to the state *Decreasing* of node *Wing Level Pattern LT0210*). In the main tank, there is no flow in the outflow line and the level is constant as expected (evidence 100 % to state *Constant* of node *Main Level Pattern LT0110* and to state *No Flow* of node *Main Outflow Pattern FT0130*). Finally, for the collector tank, *Flow* is observed in the outflow line and the level is decreasing (evidence 100 % to the state *Decreasing* of node *Collector Level Pattern LT0310* and to the state *Flow* of node *Collector Outflow Patter FT0330*).

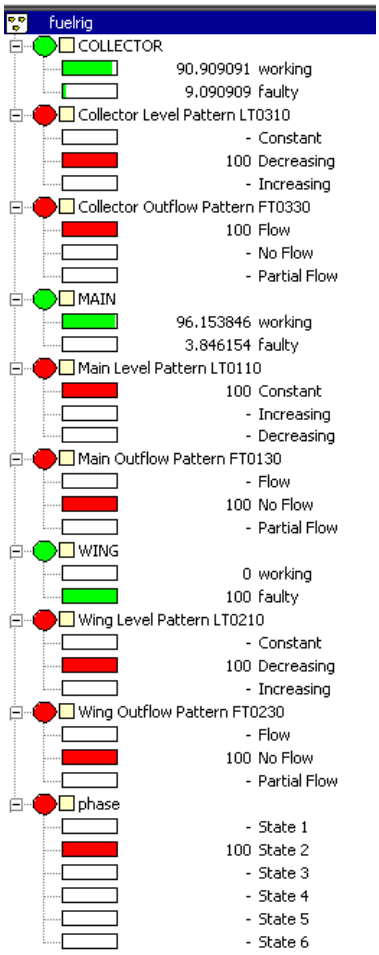


Figure 5.16 – Evidence and Posterior probabilities in the system BN.

As a consequence, the COLLECTOR, MAIN and WING nodes have posterior probabilities that show that the collector and the main tank are working while there is a failure in the wing tank. Node COLLECTOR has posterior probability 90.90% for its state *working*, MAIN has posterior probability 96.15% for its steate *working*, while node WING is 100% *faulty*. At this point, the BN for the wing tank sub-system is considered and the readings from all the sensors in the wing tank are observed, evidence is given to the corresponding nodes and the posterior probability is obtained for the components of the wing tank.

The component failures that have increased their posterior probability with respect to the given prior probability are the potential causes of the fault in the sub-system and, consequently, in the system. In our example, we assume that the patterns observed in the sensors in the wing tank are as in figure 5.17.

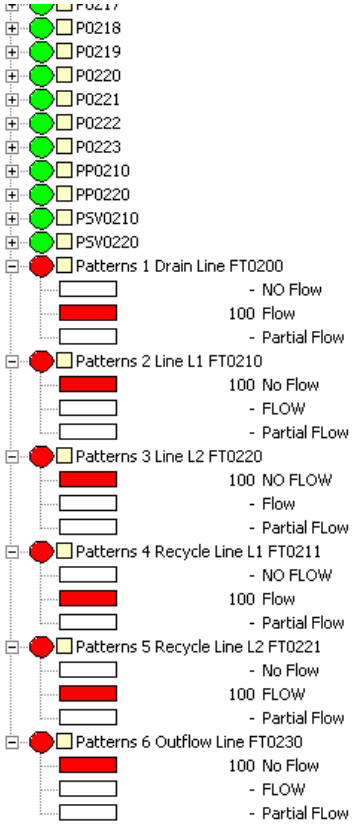


Figure 5.17 – Sensor evidence in the wing tank.

The sensor patterns are unexpected in line L1, where there is no flow, while full flow is detected through the recycle line L1. Line L2 and recycle line L2 show the expected patterns. There is no flow through the outflow line and unexpected flow through the drain line.

Given the evidence in the BN, the posterior probabilities that have increased with respect to the prior probability are shown in figure 5.18. The results in the figure are summarised and ranked in table 5.3. Here, the component failures that are most likely to have caused the deviation from the expected behaviour in the wing tank are listed. It can be seen that a failure must have occurred in the drain line, either the valve has failed open or the controller has failed true (that is, keeping the valve open). Another failure is also present in line 1, this must be a blockage in the pipes or in the valve but it cannot be a fracture or a leak, because full flow is observed in the recycle line. The BN is also able to understand that the blockage cannot have occurred before P0203 as this would also have caused no flow in the recycle line.

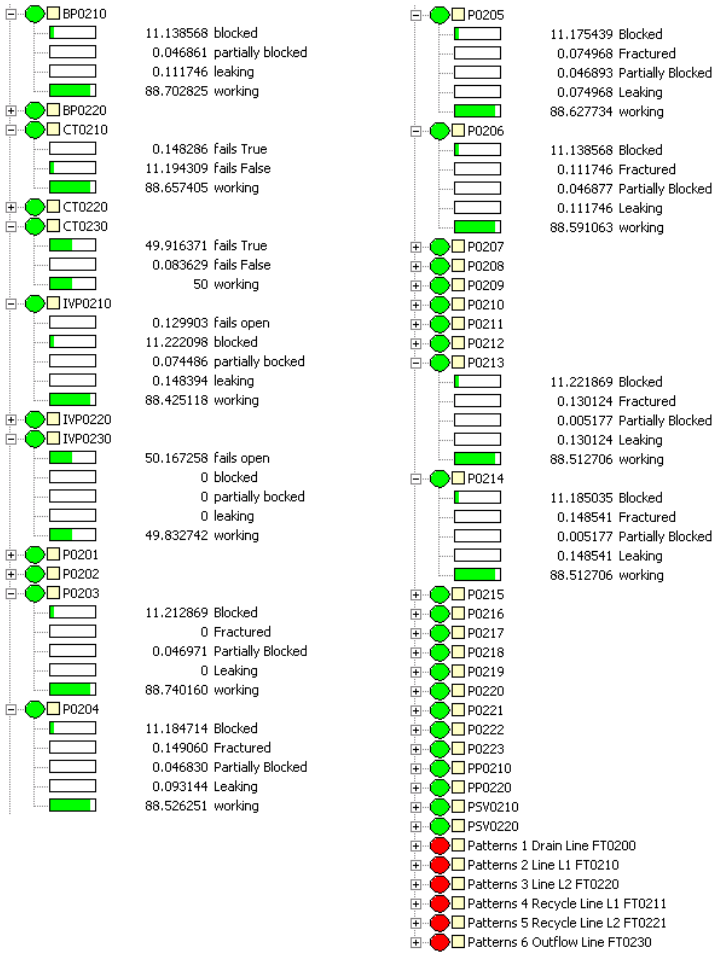


Figure 5.18 – Potential causes for a fault in the wing tank when the sensor are as in figure 5.17.

Component Failure	Probability (%)
IVP0230 fails open	50.1672
CT0230 fails true	49.9163
IVP0210 fails blocked	11.2220
P0213 fails blocked	11.2218
P0203 fails blocked	11.2128
CT0210 fails false	11.1943
P0214 fails blocked	11.1850
P0204 fails blocked	11.1847
P0205 fails blocked	11.1754
BP0210 fails blocked	11.1385
P0206 fails blocked	11.1385

Table 5.3 – Ranked list of the potential causes found in figure 5.18.

If two or more sub-systems are found faulty, then all the appropriate BNs should be studied to obtain the component failures. For this particular example, the BN method is able to identify all the potential causes of the scenario. However, in order to evaluate the method, a

system simulation is needed and results should be found and compared with the simulation. Results from these two tasks are provided in the next two sections.

5.5 System Simulation

The fuel rig system is quite a large system to be simulated with all its sub-systems and with all the possible failures. As the behaviour of the sub-systems are very similar and they have the same types of components, in order to validate the method, it is sufficient to simulate the main tank and it can be assumed that the collector tank and the wing tank give very similar results.

A simulation code was written in C++ whose aims are to deduce all possible scenarios that can occur for the 6 sensor patterns in the main tank when up to 3 failures are present in the sub-system and to identify, for each of these scenarios, the failures that are most likely to have caused the fault. The first task, deducing the scenarios, is something that would be very long to do by hand as, considering that the 6 sensors have all 3 possible patterns, there are 729 possible pattern combinations. To identify the system scenarios manually, it would be necessary to check each of them. Regarding the second task, identifying the potential causes for each scenario, this is exactly what the BN method does and, therefore, the simulation provides a means to validate the results.

The simulation of the main tank is more difficult compared with the simulation of the water tank system in chapter 4. This is because the sub-system is constituted of more components but also, and more importantly, because several failure modes are considered. In particular the partial failure modes can be difficult to model. A partial failure is defined as a failure on the system that may record as partial flow by the flow sensor. For example, a partial blockage in pipe P0104 causes partial flow through the line but it would also cause partial flow through the recycle line as part of the water would push the valve PSV110 and it would manage to pass through sensor FT0111.

The structure of the code is similar to the simulation performed for the water tank system in chapter 4. There is a basic function that has as inputs the structural parameters of the tank, the initial conditions and the failures and it gives as outputs the patterns observed in the variables measured by the sensors. This function can be iterated in a loop inducing all possible combinations of failures, up to three. Grouping together the results that the failures produce, the scenarios can be listed and the component failures that lead to each scenario are observed. In the next subsection, it is explained how the code is able to predict

the sensor patterns with the example of the line L1 in the main tank.

5.5.1 Simulating line L1

The simulation code is divided into modules. Each of them models the behaviour of a section in the main tank. They are considered separately but they influence each other, for example, the sensor pattern in the outflow line depends on the behaviour of the flow in lines L1 and L2 and flow through the outflow line cannot be observed if there is no flow through both lines. It is shown here in detail how the sensor pattern in line L1 can be predicted when any number of failures are considered in the line. We will refer to figure 5.3.

The inputs given to the code are the level of water in the tank, the system phase and the failures assumed to have occurred. The output is, in this case, the pattern observed for the flow measured at sensor FT0110. It is assumed that the system is in phase 4, therefore fuel is supposed to flow from the main tank into the collector tank. The fuel should pass through line L1 while line L2 is kept on stand-by as a back up. In the code, the line is divided into parts, each part corresponds to a component in the line. The components considered in the line are P0101, PP0110, P0102, P0103, IVP0110, P0104, P0105, BP0110, P0106, P0113 and P0114. For each component, two variables are defined, one is used to memorise the fuel flow that passes through the component, and the other the fuel that is eventually lost in correspondence to the component. These variables assume values from 0 to 1, from 0 flow to full flow, and they are called *fuel through* and *fuel out*.

If the tank is not empty and P0101 is working, *fuel through* at P0101 is 1 while *fuel out* at P0101 is 0. If, instead, P0101 has failed leaking, only partial flow will manage to pass through the pipe and a part of it will leave through the leak, therefore *fuel through* at P0101 will be 0.75 and *fuel out* is 0.25. If P0101 has failed blocked, both *fuel through* and *fuel out* are 0. If at any point in the line *fuel through* becomes 0, then this is transmitted to all following parts of the line. If *fuel through* is 0, then *fuel out* will also be 0 for the following sections, in this way, if a fracture occurs after a blockage, there is no fuel loss through the fracture as no fuel has reached the point where the fracture has occurred due to the blockage. It is assumed that a fracture causes all fuel to be lost through it, while a leak causes a quarter of the fuel to be lost. When a blockage occurs in the line after the sensor, not only does it cause *fuel through* to be 0 corresponding to the faulty component and for the following components, but it also gives a 0 value to the component in the line that precedes it, unless a fracture or a leak are causing loss of fuel. After considering all components in the line and establishing all the values of *fuel through* and *fuel out*, the sensor pattern at FT0110 simply

depends on the value of *fuel through* at P0104, which is the component located immediately before the sensor. The logic reasoning in the line L1 is shown schematically step by step:

It is assumed that *fuel through*, representing the amount of fuel that is able to pass through the line, has a value for each component of the line, so, for example, the value of it at component P0104 is indicated as *fuel through (P0104)*. The fuel that is lost through a fracture or a leak is represented by *fuel out*. The values of these variables are determined step by step starting from the first component, P0101, and ending with P0114.

At P0101

If the tank is empty or the level is below the SO limit, then $fuel\ through = 0$, $fuel\ out = 0$ for all components in the line and, as a consequence, FT0110 has pattern *No Flow*.

If P0101 is blocked, then $fuel\ through\ (P0101) = 0$, $fuel\ out\ (P0101) = 0$.

If P0101 is fractured, then $fuel\ through\ (P0101) = 0$, $fuel\ out\ (P0101) = 1$.

If P0101 is leaking, then $fuel\ through\ (P0101) = 0.75$, $fuel\ out\ (P0101) = 0.25$.

If P0101 is partially blocked, then $fuel\ through\ (P0101) = 0.5$, $fuel\ out\ (P0101) = 0$.

At PP0110

Initially, the value of *fuel through* at PP0110 is set as the previous value, that is, $fuel\ through\ (PP0110) = fuel\ through\ (P0101)$. In this way, if there was a blockage or a fracture in P0101, no fuel would reach PP0110. Then the possible failure modes of pump PP0110 are considered.

If PP0110 fails shut off, then $fuel\ through\ (PP0110) = 0$, $fuel\ out\ (PP0110) = 0$.

If PP0110 fails mechanically, $fuel\ through\ (PP0110) = 0.5 * fuel\ through\ (P0101)$, $fuel\ out\ (PP0110) = 0$. A mechanical failure is assumed to cause partial flow through the pump, similar to a partial blockage in a pipe section. Only half flow is pumped in the line.

If PP0110 fails leaking, then $fuel\ through\ (PP0110) = 0.75 * fuel\ through\ (P0101)$,

$fuel\ out\ (P0101) = 0.25\ *\ fuel\ through\ (P0101)$. This means that 25 % of the fuel that was transferred from P0101 to PP0110 is lost through the leak while 75 % of the fuel that was transferred from P0101 to PP0110 manages to be passed to the next component.

At P0102

Initially, $fuel\ through\ (P0102) = fuel\ through\ (PP0110)$. Then the possible failures modes of P0102 are considered.

If P0102 is blocked, then $fuel\ through\ (P0102) = 0$, $fuel\ out\ (P0102) = 0$.

If P0102 is fractured, then $fuel\ through\ (P0102) = 0$ and
$fuel\ out\ (P0102) = fuel\ though\ (PP0110)$. This means that all the fuel that has passed from PP0110 to P0102 is lost.

If P0102 is leaking, then $fuel\ through\ (P0102) = 0.75\ *\ fuel\ through\ (PP0110)$,
$fuel\ out\ (P0101) = 0.25\ *\ fuel\ through\ (PP0110)$.

If P0102 is partially blocked, then $fuel\ through\ (P0102) = 0.5\ *\ fuel\ through\ (PP0110)$, $fuel\ out\ (P0102) = 0$. In this case, a partial blockage causes all the flow to pass through the line.

For the remaining components, the reasoning is similar with the difference that the components that are located after the sensor influence the values of *fuel through* for the components that are located before. For example, if P0113 is blocked, unless one of the pipe sections that are located between the sensor and P0113 have failed leaking or fractured, then *fuel through* at P0104 is set to 0. When P0114 is reached, the sensor pattern can be determined as follows.

FT0110

If $fuel\ through\ (P0104) = 1$, then the pattern at FT0110 is full flow. This occurs when there are no blockages in the line and no leaks or fractures.

If $fuel\ through\ (P0104) = 0$, then the pattern at FT0110 is no flow. This occurs when there is a blockage in a pipe section, a valve has failed blocked, the pump has failed shut off or there is a fracture in a pipe section that precedes the sensor.

If $0 < fuel\ through\ (P0104) < 1$, then the pattern at FT0110 is partial flow. This is caused by a leak or a partial blockage.

LT0110

The pattern of the level sensor in the main tank can be determined considering again the values of *fuel through* but also the values of *fuel out* are needed. This is because now if *fuel through* is 0 at P0114, this does not imply necessarily that fuel is not leaving the tank. For example, if there is a fracture at P0101, this causes no flow at FT0110, but the level in the tank is decreasing anyway. The pattern in LT0110 can be either constant or decreasing. The fuel level cannot increase, as for the collector tank, because fuel does not enter the tank. The level is calculated considering the initial level and subtracting at each time step the amount of fuel that leaves the tank from the failed components (leaks and fractures) and the fuel that leaves the tank from the outflow line.

5.5.2 Generating the failures

The main use of the simulation code is to automatically generate all possible combinations of failures, up to 3, in the main tank sub-system and deducing the scenarios. A scenario for the main tank is a combination of sensor patterns that are observed when one or multiple failures occur. The code induces the failures, it runs the system for a period of time and it memorises the sensor patterns obtained. The scenarios that are obtained in this way are counted and the component failures that cause each of them are also recorded. For each possible scenario, it is recorded how many times it occurs and which component failures lead to it.

The scenarios are identified using ternary numbers. Each sensor can assume 3 patterns: *No Flow*, *Flow*, *Partial Flow*. The digits 0, 1 and 2 are associated to them as follows: 0 *No Flow*, 1 *Flow* and 2 *Partial Flow*. Giving this order to the flow sensors in the main tank: FT0100, FT0111, FT0121, FT0120, FT0121 and FT0130, a patten combination is identified with a ternary number. For example, the ternary number 210100 identifies the following combination of patterns: *Partial Flow* in sensor FT0100 (2), *Flow* in sensor FT0111 (1), *No Flow* in sensor FT0121 (0), *Flow* in sensor FT0120 (1) and *No Flow* in sensors FT0121 and FT0131. Not all combinations of patterns are possible on the system, for example, if there is *No Flow* through both the fuel lines, then the same pattern must occur in the outflow line. The simulation code creates a text file where all scenarios are listed. Of the 729 possible combinations of patterns, 195 are the ones that can be observed when up to 3 failures can occur on the system.

5.5.3 Simulation results

The simulation code creates a text file with a summary of the results for all scenarios. For each scenario that results from some combination of failures, the following information are recorded: the sensor patterns, the scenario number obtained converting the ternary number given by the sensor patterns into decimal, the occurrence of the scenario (that is the number of times it is generated by the code) and the occurrence of each of the component failures for the scenario (that is the number of times a failure appears in any combination). These results are shown for the first scenario identified by the simulation in table 5.4. This occurs when *No Flow* is observed in all sensors except in the recycle line 2 where there is full flow through the line. The scenario number is 3 as the ternary number 000010 corresponds to the decimal 3. The number of failure combinations in this scenario are 67932. The component failures are listed and ranked based on the number of times they appear in a combination. For example, P0101 blocked appears in 9234 of the 67932 combinations, that is, it causes the scenario in the 13.59 % of cases.

In table 5.4 the component failures are displayed in two columns only to fit the table in one page. In total, there are 108 component failures that can occur in this scenario. Some of them, which appear first in the ranked list, are the potential causes for the scenario. These are the components whose failures explain the deviating behaviour of the system, the others are hidden failures. As for the water tank system, in the fuel rig system the potential causes are ranked first in the list because of the nature of the simulation. When a scenario can be caused by less than 3 failures, as all combinations of 1, 2 and 3 failures are automatically generated, some combinations must contain 1 or more failures that are not actual causes and they are used by the code to complete the set. Therefore, when a hidden failure is identified, all the following component failures are hidden as well. In the example, the first 22 component failures represent potential causes for the scenario. The 23rd component failure is CT0130 fails true, this is a failure that occurs in the controller located in the drainage line that keeps the valve closed. In all phases of the ACTIVE operating mode the valve in the drainage line is required to be closed, hence this failure is not an actual cause for the scenario and it will remain hidden until, in another operating mode, the drainage line will be used to simulate the dumping of fuel.

The simulation code produces 195 tables such as the one for scenario 3 in table 5.4. Comparing the results of the simulation with the posterior probability in the system BN, one can assess the accuracy of the diagnostic method for all scenarios that are caused by up to 3 failures. Table 5.5 shows a summary of the scenarios obtained by the simulation where the

sensor in the drainage line has the pattern *No Flow*. The scenarios can be divided in three groups, depending on the pattern in the drainage line, whether it is *No Flow*, *Full Flow* and *Partial Flow*. The drainage line does not have any component in common with the other sections of the main tank. The results for the scenarios in which the sensor in the drainage line, FT0130, shows *No Flow* are the ones where no failures occurred in the drainage line. This is because *No Flow* is the expected pattern.

	Scenario	3					
	Occurrence	67932					
	Drainage Line	0	No Flow				
	Line 1	0	No FLow				
	Line 2	0	No FLow				
	Recycle Line 1	0	No FLow				
	Recycle Line 2	1	Flow				
	Outflow Line	0	No Flow				
	Component Failure	**Occurrence**	**Occ. (%)**		**Component Failure**	**Occurrence**	**Occ. (%)**
1	P0101 blocked	9234	13.59%	66	CT0110 true	956	1.41%
2	P0101 fractured	9234	13.59%	67	P0105 part blocked	955	1.41%
3	P0102 blocked	9234	13.59%	68	P0106 part blocked	955	1.41%
4	P0102 fractured	9234	13.59%	69	P0106 leaking	955	1.41%
5	P0103 fractured	9234	13.59%	70	P0113 part blocked	955	1.41%
6	PP0110 off	9234	13.59%	71	P0114 fractured	955	1.41%
7	P0104 fractured	8637	12.71%	72	P0114 leaking	955	1.41%
8	P0115 blocked	1997	2.94%	73	BP0110 part blocked	955	1.41%
9	P0115 fractured	1997	2.94%	74	P0117 fractured	952	1.40%
10	P0116 blocked	1997	2.94%	75	P0117 part blocked	952	1.40%
11	P0116 fractured	1997	2.94%	76	P0117 leaking	952	1.40%
12	P0117 blocked	1997	2.94%	77	PSV0110 open	952	1.40%
13	PSV0110 blocked	1997	2.94%	78	PSV0110 part blocked	952	1.40%
14	P0105 blocked	1650	2.43%	79	PSV0110 leaking	952	1.40%
15	BP0110 blocked	1638	2.41%	80	P0113 fractured	949	1.40%
16	P0113 blocked	1633	2.40%	81	P0113 leaking	949	1.40%
17	P0106 blocked	1632	2.40%	82	P0114 part blocked	949	1.40%
18	P0114 blocked	1627	2.40%	83	P0106 fractured	943	1.39%
19	P0104 blocked	1534	2.26%	84	P0115 part blocked	943	1.39%
20	IVP0110 blocked	1534	2.26%	85	P0115 leaking	943	1.39%
21	CT0110 false	1528	2.25%	86	P0116 part blocked	943	1.39%
22	P0103 blocked	1412	2.08%	87	P0116 leaking	943	1.39%
23	CT0130 true	1045	1.54%	88	BP0110 leaking	937	1.38%
24	P0109 blocked	1003	1.48%	89	P0105 fractured	931	1.37%
25	P0110 blocked	982	1.45%	90	P0105 leaking	931	1.37%
26	PP0111 blocked	982	1.45%	91	PP0110 on	834	1.23%
27	P0121 blocked	982	1.45%	92	PP0110 leaking	834	1.23%
28	P0121 fractured	982	1.45%	93	PP0110 fails mechanic	834	1.23%
29	P0122 blocked	982	1.45%	94	IVP0110 part blocked	834	1.23%
30	P0122 fractured	982	1.45%	95	P0104 part blocked	833	1.23%
31	P01032 blocked	982	1.45%	96	P0104 leaking	833	1.23%
32	PP0120 on	982	1.45%	97	P0103 leaking	831	1.22%
33	IVP0120 blocked	982	1.45%	98	P0103 part blocked	712	1.05%
34	IVP0130 blocked	982	1.45%	99	P0101 part blocked	705	1.04%
35	PSV0120 open	982	1.45%	100	P0101 leaking	705	1.04%
36	P0109 part blocked	975	1.44%	101	P0102 part blocked	705	1.04%
37	IVP0130 leaking	975	1.44%	102	P0102 leaking	705	1.04%
38	PSV0120 part blocked	975	1.44%	103	IVP0120 part blocked	47	0.07%
39	P0120 fractured	968	1.42%	104	IVP0130 open	35	0.05%
40	P0120 part blocked	968	1.42%	105	IVP0130 part blocked	35	0.05%
41	P0120 leaking	968	1.42%	106	IVP0120 open	21	0.03%
42	CT0120 falsetemp	968	1.42%	107	CT0120 false	21	0.03%
43	CT0130 false	968	1.42%	108	IVP0120 leaking	7	0.01%
44	P0110 fractured	961	1.41%	109	P0107 blocked	0	0.00%
45	P0110 part blocked	961	1.41%	110	P0107 fractured	0	0.00%
46	P0110 leaking	961	1.41%	111	P0107 part blocked	0	0.00%
47	P0111 fractured	961	1.41%	112	P0107 leaking	0	0.00%
48	P0111 part blocked	961	1.41%	113	P0108 blocked	0	0.00%
49	P0111 leaking	961	1.41%	114	P0108 fractured	0	0.00%
50	P0112 blocked	961	1.41%	115	P0108 part blocked	0	0.00%
51	P0112 fractured	961	1.41%	116	P0108 leaking	0	0.00%
52	P0112 part blocked	961	1.41%	117	P0109 fractured	0	0.00%
53	P0112 leaking	961	1.41%	118	P0109 leaking	0	0.00%
54	P0121 part blocked	961	1.41%	119	P0118 blocked	0	0.00%
55	P0121 leaking	961	1.41%	120	P0118 fractured	0	0.00%
56	P0122 part blocked	961	1.41%	121	P0118 part blocked	0	0.00%
57	P0122 leaking	961	1.41%	122	P0118 leaking	0	0.00%
58	P0123 fractured	961	1.41%	123	P0119 blocked	0	0.00%
59	P0123 part blocked	961	1.41%	124	P0119 fractured	0	0.00%
60	P0123 leaking	961	1.41%	125	P0119 part blocked	0	0.00%
61	BP0120 blocked	961	1.41%	126	P0119 leaking	0	0.00%
62	BP0120 part blocked	961	1.41%	127	P0120 blocked	0	0.00%
63	BP0120 leaking	961	1.41%	128	PP0120 off	0	0.00%
64	IVP0110 leaking	958	1.41%	129	PP0120 leaking	0	0.00%
65	IVP0110 open	957	1.41%	130	PP0120 fails mechanic	0	0.00%
				131	PSV0120 blocked	0	0.00%
				132	PSV0120 leaking	0	0.00%

Table 5.4 – Results from the simulation code for scenario 3.

Scenario	Number of Actual Causes
3	22
6	38
9	21
12	9
15	24
18	20
21	17
24	20
27	18
28	11
29	10
36	8
37	9
38	9
46	30
47	10
62	10
63	12
65	15
71	10
80	28
81	15
82	12
83	16
84	3
85	Expected
86	4
87	15
88	14
89	21
108	7
109	2
110	6
135	22
136	1
137	8
142	4
143	6
162	32
164	22
165	14
167	9
168	25
170	27
180	5
182	17
183	5
185	6
186	18
188	7
189	15
190	18
191	11
207	4
208	8
209	11
216	16
218	11
223	13
224	9
234	4
235	13
236	7
241	9
242	11

Table 5.5 – Scenarios and number of potential causes identified by the simulation code for the scenarios in which the pattern is the drainage line is *No Flow*.

Scenario 85 occurs when the patterns in the main tank are as follows: *No Flow* in the drainage line, *Flow* in line L1, *No Flow* in line L2, *No Flow* in the recycle line L1, *Flow* in the recycle line L2 and *Flow* in the outflow line. These are the expected sensor patterns for the phase 4 of the active operating mode.

Changing the pattern in the drainage line to *Flow*, the scenarios obtained by the simulation have exactly the same causes as the ones for *No Flow*, with the addition of two extra causes, which are the valve in the drainage line failing open and the controller CT0130 failing true. A similar thing can be said for the scenarios in which the pattern in the drainage line is *Partial Flow*, the scenarios are the same with the extra failures that cause partial flow in the drainage line, which are 7 and they are the partial blockages and leaks of the pipe and valve in the line. There are 65 scenarios that occur when up to 3 failures are induced in the system and in which the first sensor reading is *No Flow*. The other 130 scenarios are the ones obtained adding 242 to the scenario number and that are produced by the extra 2 causes in the drainage line (second column in the table in table 5.6) and the ones obtained adding 484 to the scenario number and that are produced by the extra 7 causes in the drainage line (third column in the table 5.6).

Table 5.6 lists the scenarios and the number of actual causes for each scenario. Given the scenario number, it is possible to obtain the sensor patterns that identify the scenarios as these are given by the digits of the ternary number equivalent to the decimal number that labels the scenario. For example, let us consider scenario 3, the equivalent ternary is 10. Writing 10 in 6 digits gives 000010, that is scenario 3 is the one for which the patterns for the sensors are 0, 0, 0, 0, 1 and 0, that is, *No Flow* in the drainage line, in line 1 and 2, in the recycle line L1 and in the outflow line, and *Flow* in the recycle line L2.

Scenario	Number of Actual Causes	Scenario	Number of Actual Causes	Scenario	Number of Actual Causes
3	22	245	24	487	29
6	38	248	40	490	45
9	21	251	23	493	28
12	9	254	11	496	16
15	24	257	26	499	31
18	20	260	22	502	27
21	17	263	19	505	24
24	20	266	22	508	27
27	18	269	20	511	25
28	11	270	13	512	18
29	10	271	12	513	17
36	8	278	10	520	15
37	9	279	11	521	16
38	9	280	11	522	16
46	30	288	32	530	37
47	10	289	12	531	17
62	10	304	12	546	17
63	12	305	14	547	19
65	15	307	17	549	22
71	10	313	12	555	17
80	28	322	30	564	35
81	15	323	17	565	22
82	12	324	14	566	19
83	16	325	18	567	23
84	3	326	5	568	10
85	Expected	327	2	569	7
86	4	328	6	570	11
87	15	329	17	571	22
88	14	330	16	572	21
89	21	331	23	573	28
108	7	350	9	592	14
109	2	351	4	593	9
110	6	352	8	594	13
135	22	377	24	619	29
136	1	378	3	620	8
137	8	379	10	621	15
142	4	384	6	626	11
143	6	385	8	627	13
162	32	404	34	646	39
164	22	406	24	648	29
165	14	407	16	649	21
167	9	409	11	651	16
168	25	410	27	652	32
170	27	412	29	654	34
180	5	422	7	664	12
182	17	424	19	666	24
183	5	425	7	667	12
185	6	427	8	669	13
186	18	428	20	670	25
188	7	430	9	672	14
189	15	431	17	673	22
190	18	432	20	674	25
191	11	433	13	675	18
207	4	449	6	691	11
208	8	450	10	692	15
209	11	451	13	693	18
216	16	458	18	700	23
218	11	460	13	702	18
223	13	465	15	707	20
224	9	466	11	708	16
234	4	476	6	718	11
235	13	477	15	719	20
236	7	478	9	720	14
241	9	483	11	725	16
242	11	484	13	726	18

Table 5.6 – Scenarios and number of potential causes identified by the simulation code.

The scenarios that are not found by the simulation code are the scenarios that are caused

by more than 3 failures and the scenarios that cannot occur for any combination of failures. The results of the diagnostic method are validated using the simulation results in the next section.

5.6 Results

The BN method is validated using the simulation results. For each scenario, the posterior probability of the nodes that represent the actual causes identified in the code is checked. This was done manually. Although there are 195 scenarios, the section regarding the drainage line can be considered independently as its components are not common to any other section. The deviating behaviour represented by the unexpected patterns in the drainage line are correctly identified by the BN diagnosis. This is quite a simple problem as there are only 4 components that belong to the drainage line. Excluding the drainage line simplifies the problem of checking the results from 195 scenarios to 65.

The process of comparing the simulation results with the BN method is explained with the example of scenario 3. In scenario 3, the sensor patterns are the ones shown in figure 5.4. Evidence is given to the nodes representing the sensor patterns of the BN that models the main tank. The posterior probability of all root nodes, the component failures, is deduced. In table 5.7, the posterior probabilities of the component failures is added in a column next to the simulation occurrence given by the simulation. The actual causes of scenario 3 are the first 22 in the list. It can be seen that the only component failures that have increased the posterior probability with respect to their prior probability are the first 22 components. It can also be seen that these are the only ones whose posterior probability has increased. The diagnostic method, for this scenario, has given accurate results. Comparing the two columns of results in the table, the occurrence in percentage given by the simulation and the posterior probabilities, it can also be seen how the ranking of the two results is very similar.

The diagnosis is checked for each of the other 64 scenarios, considering if the actual causes found by the code are among the potential causes identified by the BN method. A complete list of the number of actual causes and causes identified is given for all scenarios in table 5.8. Calculating that the total number of actual causes is 2787 and that the causes identified by the method are 2724, the diagnosis correctly identifies 97.73 % of the failures.

	Scenario	3							
	Occurrence	67932							
	Drainage Line	0	No Flow						
	Line 1	0	No FLow						
	Line 2	0	No FLow						
	Recycle Line 1	0	No FLow						
	Recycle Line 2	1	Flow						
	Outflow Line	0	No Flow						
	Component Failu	**Occurrence**	**Occ. (%)**	**Posterior Prob.**		**Component Failur**	**Occurrence**	**Occ. (%)**	**Posterior Prob.**
1	P0101 blocked	9234	13.59%	14.0291%	66	CT0110 true	956	1.41%	0.1665%
2	P0101 fractured	9234	13.59%	14.2352%	67	P0105 part blocked	955	1.41%	0.0613%
3	P0102 blocked	9234	13.59%	14.2352%	68	P0106 part blocked	955	1.41%	0.0613%
4	P0102 fractured	9234	13.59%	14.2352%	69	P0106 leaking	955	1.41%	0.1661%
5	P0103 fractured	9234	13.59%	14.2352%	70	P0113 part blocked	955	1.41%	0.0016%
6	PP0110 off	9234	13.59%	13.9219%	71	P0114 fractured	955	1.41%	0.1668%
7	P0104 fractured	8637	12.71%	13.9795%	72	P0114 leaking	955	1.41%	0.1668%
8	P0115 blocked	1997	2.94%	0.3754%	73	BP0110 part blocke	955	1.41%	0.0613%
9	P0115 fractured	1997	2.94%	0.3754%	74	P0117 fractured	952	1.40%	0.1663%
10	P0116 blocked	1997	2.94%	0.3754%	75	P0117 part blocked	952	1.40%	0.1664%
11	P0116 fractured	1997	2.94%	0.3754%	76	P0117 leaking	952	1.40%	0.1663%
12	P0117 blocked	1997	2.94%	0.3754%	77	PSV0110 open	952	1.40%	0.1663%
13	PSV0110 blocked	1997	2.94%	0.3754%	78	PSV0110 part block	952	1.40%	0.1664%
14	P0105 blocked	1650	2.43%	0.2707%	79	PSV0110 leaking	952	1.40%	0.1664%
15	BP0110 blocked	1638	2.41%	0.2702%	80	P0113 fractured	949	1.40%	0.1665%
16	P0113 blocked	1633	2.40%	0.2729%	81	P0113 leaking	949	1.40%	0.1665%
17	P0106 blocked	1632	2.40%	0.2702%	82	P0114 part blocked	949	1.40%	0.0008%
18	P0114 blocked	1627	2.40%	0.2723%	83	P0106 fractured	943	1.39%	0.1661%
19	P0104 blocked	1534	2.26%	0.2709%	84	P0115 part blocked	943	1.39%	0.1661%
20	IVP0110 blocked	1534	2.26%	0.2846%	85	P0115 leaking	943	1.39%	0.1661%
21	CT0110 false	1528	2.25%	0.2844%	86	P0116 part blocked	943	1.39%	0.1661%
22	P0103 blocked	1412	2.08%	0.2361%	87	P0116 leaking	943	1.39%	0.1661%
23	CT0130 true	1045	1.54%	0.0019%	88	BP0110 leaking	937	1.38%	0.1661%
24	P0109 blocked	1003	1.48%	0.1695%	89	P0105 fractured	931	1.37%	0.1655%
25	P0110 blocked	982	1.45%	0.1689%	90	P0105 leaking	931	1.37%	0.1655%
26	P0111 blocked	982	1.45%	0.1689%	91	PP0110 on	834	1.23%	0.0728%
27	P0121 blocked	982	1.45%	0.1675%	92	PP0110 leaking	834	1.23%	0.0728%
28	P0121 fractured	982	1.45%	0.1675%	93	PP0110 fails mecha	834	1.23%	0.0728%
29	P0122 blocked	982	1.45%	0.1675%	94	IVP0110 part blocke	834	1.23%	0.1207%
30	P0122 fractured	982	1.45%	0.1675%	95	P0104 part blocked	833	1.23%	0.0612%
31	P01032 blocked	982	1.45%	0.1675%	96	P0104 leaking	833	1.23%	0.0728%
32	PP0120 on	982	1.45%	0.1672%	97	P0103 leaking	831	1.22%	0.1862%
33	IVP0120 blocked	982	1.45%	0.1695%	98	P0103 part blocked	712	1.05%	0.0494%
34	IVP0130 blocked	982	1.45%	0.1675%	99	P0101 part blocked	705	1.04%	0.0607%
35	PSV0120 open	982	1.45%	0.1675%	100	P0101 leaking	705	1.04%	0.0607%
36	P0109 part blocked	975	1.44%	0.0023%	101	P0102 part blocked	705	1.04%	0.0607%
37	IVP0130 leaking	975	1.44%	0.1675%	102	P0102 leaking	705	1.04%	0.0607%
38	PSV0120 part blocl	975	1.44%	0.0000%	103	IVP0120 part blocke	47	0.07%	0.0020%
39	P0120 fractured	968	1.42%	0.1672%	104	IVP0130 open	35	0.05%	0.0013%
40	P0120 part blocked	968	1.42%	0.0000%	105	IVP0130 part blocke	35	0.05%	0.0013%
41	P0120 leaking	968	1.42%	0.1672%	106	IVP0120 open	21	0.03%	0.0000%
42	CT0120 falsetemp	968	1.42%	0.0002%	107	CT0120 false	21	0.03%	0.1686%
43	CT0130 false	968	1.42%	0.1669%	108	IVP0120 leaking	7	0.01%	0.0025%
44	P0110 fractured	961	1.41%	0.1672%	109	P0107 blocked	0	0.00%	0.0000%
45	P0110 part blocked	961	1.41%	0.0023%	110	P0107 fractured	0	0.00%	0.0000%
46	P0110 leaking	961	1.41%	0.1669%	111	P0107 part blocked	0	0.00%	0.0000%
47	P0111 fractured	961	1.41%	0.1670%	112	P0107 leaking	0	0.00%	0.0000%
48	P0111 part blocked	961	1.41%	0.0023%	113	P0108 blocked	0	0.00%	0.0000%
49	P0111 leaking	961	1.41%	0.1669%	114	P0108 fractured	0	0.00%	0.0000%
50	P0112 blocked	961	1.41%	0.1689%	115	P0108 part blocked	0	0.00%	0.0000%
51	P0112 fractured	961	1.41%	0.1669%	116	P0108 leaking	0	0.00%	0.0000%
52	P0112 part blocked	961	1.41%	0.0023%	117	P0109 fractured	0	0.00%	0.0000%
53	P0112 leaking	961	1.41%	0.1669%	118	P0109 leaking	0	0.00%	0.0000%
54	P0121 part blocked	961	1.41%	0.1666%	119	P0118 blocked	0	0.00%	0.0000%
55	P0121 leaking	961	1.41%	0.1666%	120	P0118 fractured	0	0.00%	0.0000%
56	P0122 part blocked	961	1.41%	0.1666%	121	P0118 part blocked	0	0.00%	0.0000%
57	P0122 leaking	961	1.41%	0.1666%	122	P0118 leaking	0	0.00%	0.0000%
58	P0123 fractured	961	1.41%	0.1666%	123	P0119 blocked	0	0.00%	0.0000%
59	P0123 part blocked	961	1.41%	0.1666%	124	P0119 fractured	0	0.00%	0.0000%
60	P0123 leaking	961	1.41%	0.1666%	125	P0119 part blocked	0	0.00%	0.0000%
61	BP0120 blocked	961	1.41%	0.1672%	126	P0119 leaking	0	0.00%	0.0000%
62	BP0120 part blocke	961	1.41%	0.0025%	127	P0120 blocked	0	0.00%	0.0000%
63	BP0120 leaking	961	1.41%	0.1669%	128	PP0120 off	0	0.00%	0.0000%
64	IVP0110 leaking	958	1.41%	0.1665%	129	PP0120 leaking	0	0.00%	0.0000%
65	IVP0110 open	957	1.41%	0.1663%	130	PP0120 fails mecha	0	0.00%	0.0000%
					131	PSV0120 blocked	0	0.00%	0.0000%
					132	PSV0120 leaking	0	0.00%	0.0000%

Table 5.7 – Compared results of simulation (occurrence of component failures) and BN method (posterior probability) for scenario 3.

Scenario	Number of Actual Causes	Number of Causes Identified	Scenario	Number of Actual Causes	Number of Causes Identified	Scenario	Number of Actual Causes	Number of Causes Identified
3	22	22	245	24	24	487	29	29
6	38	37	248	40	39	490	45	44
9	21	21	251	23	23	493	28	28
12	9	9	254	11	11	496	16	16
15	24	24	257	26	26	499	31	31
18	20	20	260	22	22	502	27	27
21	17	17	263	19	19	505	24	24
24	20	19	266	22	21	508	27	26
27	18	18	269	20	20	511	25	25
28	11	11	270	13	13	512	18	18
29	10	10	271	12	12	513	17	17
36	8	8	278	10	10	520	15	15
37	9	9	279	11	11	521	16	16
38	9	9	280	11	11	522	16	16
46	30	28	288	32	30	530	37	35
47	10	10	289	12	12	531	17	17
62	10	10	304	12	12	546	17	17
63	12	12	305	14	14	547	19	19
65	15	15	307	17	17	549	22	22
71	10	10	313	12	12	555	17	17
80	28	24	322	30	26	564	35	31
81	15	15	323	17	17	565	22	22
82	12	12	324	14	14	566	19	19
83	16	16	325	18	18	567	23	23
84	3	3	326	5	5	568	10	10
85	Expected		327	2	2	569	7	7
86	4	4	328	6	6	570	11	11
87	15	15	329	17	17	571	22	22
88	14	14	330	16	16	572	21	21
89	21	21	331	23	23	573	28	28
108	7	7	350	9	9	592	14	14
109	2	2	351	4	4	593	9	9
110	6	6	352	8	8	594	13	13
135	22	22	377	24	24	619	29	29
136	1	1	378	3	3	620	8	8
137	8	1	379	10	3	621	15	8
142	4	4	384	6	6	626	11	11
143	6	3	385	8	5	627	13	10
162	32	32	404	34	34	646	39	39
164	22	22	406	24	24	648	29	29
165	14	14	407	16	16	649	21	21
167	9	9	409	11	11	651	16	16
168	25	25	410	27	27	652	32	32
170	27	27	412	29	29	654	34	34
180	5	5	422	7	7	664	12	12
182	17	17	424	19	19	666	24	24
183	5	5	425	7	7	667	12	12
185	6	6	427	8	8	669	13	13
186	18	18	428	20	20	670	25	25
188	7	7	430	9	9	672	14	14
189	15	15	431	17	17	673	22	22
190	18	18	432	20	20	674	25	25
191	11	11	433	13	13	675	18	18
207	4	4	449	6	6	691	11	11
208	8	8	450	10	10	692	15	15
209	11	8	451	13	10	693	18	15
216	16	13	458	18	15	700	23	20
218	11	11	460	13	13	702	18	18
223	13	13	465	15	15	707	20	20
224	9	7	466	11	9	708	16	14
234	4	4	476	6	6	718	11	11
235	13	13	477	15	15	719	20	20
236	7	7	478	9	9	720	14	14
241	9	9	483	11	11	725	16	16
242	11	11	484	13	13	726	18	18

Table 5.8 – Summary of results comparison between the simulation and the diagnosis.

Among the scenarios there is one in particular where the BN method fails to correctly identify the causes of the fault, it is scenario 137. This can give a typical example of why the BN diagnosis can go wrong. Scenario 137 occurs when the sensor patterns are as follows (ternary number 012002):

Drainage line: *No Flow*
Line L1: *Flow*
Line L2: *Partial Flow*
Recycle line L1: *No Flow*
Recycle line L2: *No Flow*
Outflow line: *Partial Flow*.

The simulation generates this scenario with the component failure occurrences shown in table 5.9.

Scenario	Occurrence		Drainage Line	Line L1	Line L2	Recycle Line 1	Recycle Line 2	Outflow Line
137	25		0	1	2	0	0	2
IVP0120 part blocked	25	100.00%						
P0105 fractured	7	28.00%						
P0106 fractured	7	28.00%						
P0113 leaking	5	20.00%						
P0111 fractured	4	16.00%						
P0112 fractured	4	16.00%						
P0105 leaking	2	8.00%						
P0106 leaking	2	8.00%						

Table 5.9 – Results from the simulation code for scenario 137.

The table shows only the actual causes of the scenario. From the sensor readings, it seems that line L1 is working as flow is correctly flowing through the line and no flow is sensed on the recycle line. Partial flow occurs on the outflow line. There is also an unexpected behaviour in the line L2 as no flow is expected while partial flow is measured. The scenario occurs when there is a partial blockage in the valve in line L2, IVP0120. The cause of partial flow in the outflow line is either due to line L1 or line L2. If the fault is in line L1, there is a fracture in pipe P0105 or P0106 and partial flow comes from line L2. If, instead, there is a fracture in line L2, in either P0111 or P0112, there should be a leak in either P0105, P0106 or P0113 that is causing partial flow in the outflow line. In the BN, the cause of the partial flow in the outflow line is justified with the partial flow through line L2. Therefore, only the failure in the valve IVP0120 is detected. This error comes from the fact that fractures in the pipes are failure modes that are difficult to detect. The BN could be modified ad hoc to be able to understand this particular scenario, however it is difficult to obtain a diagnostic method applied to a system such as the fuel rig that is 100 % accurate.

5.7 Discussions and Conclusions

The diagnostic method developed using BNs has been applied to the fuel rig system. A validation achieved by comparing the results with a simulation program in C++ shows that considering the scenarios generated by up to 3 failures the diagnosis is able to find 97.73 % of the failures in the main tank sub-system. The simulation was carried out for a particular phase of the active operating mode of the main tank, but this can be extended to the whole system because of the similarities of the sub-systems and phases. The method has been modified and adapted with respect to that reported in the previous chapter. The division into sub-systems allows the networks to be kept relatively small even if the system is much larger than the water tank. The BN also allows the use of similar sub-systems or redundant parts in order to use the same nodes without the need of repeating entire networks that are logically equivalent. The BNs are also diversified depending on the phase the system is in, that is, depending on the sensor behaviour that is expected. This helps in reducing the complexity of the BNs.

The application to the fuel rig system has proved that using BNs with this diagnostic approach is effective. Calculating the posterior probability and evaluating the scenarios is simpler than calculating the prime implicants and the importance measures in FTA. Moreover, using BN is possible to produce a list of hidden failures for a scenario. These are the component failures that show the same symptoms if they are failing or working for a particular operating mode. For example, the blockage in a valve when this is supposed to be closed.

The main disadvantage of the method is the fact that the sensors are assumed to be 100% reliable. This can represent a restrictive assumption. An approach that could be taken to adjust the method including sensor failures is to modify the CPTs of the nodes representing the sensor patterns making them probabilistic rather than deterministic. If the failure probability of a sensor is known, this could be treated as a component and the probability of failure is introduced in the pattern nodes. Introducing sensor failures would make the diagnosis less accurate as a deviating pattern could be caused by a fault in the system or a fault in the sensor. However, when an impossible scenario is observed, then a failure in one of the sensors must have occurred and the BN could be used to determine which sensor is most likely to have failed.

5.8 Summary

The diagnostic method has been applied to the fuel rig system. This has required some adaptations and modifications in order to reduce the complexity of the networks.

The diagnostic procedure is applied in two stages. First a BN representing the system is used to detect which of the sub-systems is potentially faulty and then the BN modelling the sub-systems are evaluated to identify the component failures that have caused the fault.

The evaluation of a network consists of introducing the evidence provided by the sensors and calculating the posterior probability of the nodes. Once the BNs are developed and created, this is a fast procedure compared with the calculation of the prime implicants in FTA.

A simulation program in C++ is used to generate the faulty scenarios of the systems and the failures that cause them. Comparing the results from the simulation enables the method to be validated and the diagnosis has proved to be accurate.

Chapter 6

Summary, Conclusions and Further Work

6.1 Introduction

In this thesis a fault diagnostic method has been developed using BNs. The method uses FTs for the purpose of building the networks. In this way, the advantages of both techniques are employed: systematic construction with FTs and the ability to introduce evidence in the evaluation with BNs. The aim of the research was to establish if BNs could be used effectively for system diagnostics using a model-based approach. BNs have shown with this application they are ideal in the field of fault diagnostics when used with a model-based approach. The diagnostics methodology was applied to two systems, the water tank system and the fuel rig system, and the results were validated with two simulation codes in C++. This chapter summarises the achievements of the research and suggests further work that can be continued in the same direction.

6.2 Summary

6.2.1 Achievements

The main achievement of the research was the creation of a general approach that uses BNs for a model-based fault diagnostic system. Literature on BNs is dominated by case-based diagnostic reasoning, where the probabilities of the networks are obtained by training them using historical data known about previous faults. The method developed gives a general, straightforward and structured way to build a class of BNs and to evaluate them. The aim of the strategy is detecting when the system is faulty and diagnosing the cause or causes.

The diagnostic system is built in two stages: the modelling and preparation stage and the FTs and BNs development. The modelling and preparation stage and the development of the FTs follows the approach taken from a previously developed diagnostic system that uses FTA. FTs are then converted into BNs and these are manipulated to form a BN that models the entire process. When the observed sensor readings deviate from those expected, the BN posterior probability is calculated and a list of potential causes is obtained. Introducing BNs has several advantages with respect to the FTs that can be grouped in two aspects: the graphical representation and the probability evaluation. The graphical representation of the BNs is more concise because repeated events can be avoided and because BNs allow more modelling solutions such as the introduction of nodes with several states that represent the system components. Regarding the evaluation, that is the calculation of the causes of a fault, with the FTA approach, a different FT is built for every faulty scenario of the system. The prime implicants are then calculated for this FTs and finally the importance measured are considered. Using BNs enables to evaluate the same model for every scenario of the system by simply introducing evidence on the nodes that represent the symptoms. The posterior probability gives a measure of the likelihood for a component failure to be the cause of a fault. The calculation of the updated probability is almost instant with the software used.

6.2.2 Characteristics of the method

One of the aims of the research was to be able to model dynamic behaviours in the system. This was done introducing sensor patterns in the analysis and creating nodes in the BNs whose states represent the sensor patterns. The patterns symbolises dynamic trends that are observable in the sensors, therefore they show how a monitored variable changes over time. This is an important dynamic aspect in the system because considering *static* sensor readings can give a limitation to the number of scenarios and causes that can be analysed.

Another relevant aspect of the diagnosis is the fact that is a general approach. In principle, any system for which FTs can be built can be diagnosed using this strategy. The way the component failures cause a system fault must be understood and the failure probabilities of the components must be available.

6.2.3 The water tank system

The diagnostic method is applied to the water tank system with two approaches. First, the system is studied in steady state and the FTs and BNs are built without considering the dynamics of the system. This approach has shown to provide accurate results however it

was unable to take into account all faulty scenarios. For this reason, the method has been improved introducing dynamic sensor patterns. The scale of the problem was relatively small and a more complex system had to be studied to prove the method applicability.

6.2.4 The fuel rig system

The fuel rig system is considerably larger than the water tank system, in terms of number of components and failure modes considered. The method was improved considering a division into sub-systems and modifying it introducing a two stage "modularization" strategy. In the first stage, a BN is evaluated to understand which sub-system contains the cause of a fault while in the second stage, the faulty sub-system or sub-systems are considered one at the time to identify the component failures that represent potential causes.

6.2.5 Validation of the method

In order to validate the results given by the diagnostic system, one should know all the ways a system can fail, that is, the scenarios, and, for each of them, the component failures that cause them. In this way, for each scenario, one can check if the method correctly identifies the actual causes. For both the water tank system and the fuel rig system, a simulation code in C++ was implemented. These codes produce the system scenarios when up to 3 failures occur and they also create the list of potential causes for each scenario. Comparing the results from the simulation and the ones from the diagnostic methods allows the accuracy of the method to be assessed.

6.3 Conclusions

- The method gives a general and structured approach for building BNs for the fault diagnostics of a system. It is applicable to a wide range of systems: the ones for which the component failure probabilities are known and the sensor trends and the causality of the system failure modes are understood. The method is less suitable for those systems whose systematic knowledge is poor and for which, instead, data are available from previous fault situations. The networks are obtained from the conversion of FTs with some modifications such as the introduction of a structure logic that models the sensor readings of the system and that allows all scenarios to be studied with the same BNs. Therefore, the introduction of BNs has simplified the FTA diagnostic approach.
- The diagnostic has proved to give accurate results when applied to a simple water tank system. The BN analysis is able to tell when a fault occurs on the system and it produces a ranked list of component failures that are potential causes for the system fault. Once

the BNs are built for a system, the diagnosis can be performed almost instantly as the posterior probability calculations are fast due to the deterministic nature of the BNs.

- The method is able to handle dynamic effects on the system as the patterns of the sensor readings are considered in the BN structure.
- The results of the method are validated with a simulation program that automatically generates the failures in the system and produces the symptom outputs. Comparing the results produced by the simulation and by the BN allow the accuracy of the fault diagnostics to be assessed.
- The fuel rig system is finally used to test the application of the method for larger and more complex systems. A simulation programme again validates the results for this application.
- To refer back to the parameters that a diagnostic system should have, mentioned in section 1.2 of chapter 1, the method developed has an efficient construction, as it is structured, it has an optimal cost/benefit ratio as it is software based, it is not difficult to maintain and it gives a quick response when sensors evidence is introduced. The strategy is also able to detect failures that have not been encountered before, as it does not rely on previous experience. Once the BN for the system is built, however, changes in the system, in the environmental conditions or in the process variables may be difficult to incorporate in an existing network, therefore the method may not be easily adaptable to substantial changes.

6.4 Further Work

The research leaves some points that could be further investigated. One regards the system sensors in two aspects: their unreliability and their location. Other aspects concern the application of the method, which could to be extended to an entire system such as an aircraft or to systems of different nature from the ones considered.

- In the approach taken in this thesis it is assumed that the sensors are perfectly reliable. However, sensors are components and, as such, they may fail in a number of different failure modes. In order to introduce unreliable sensors, the CPTs of the nodes representing the sensors could be modified substituting 1s with the failure probability of the sensor. In this way, the logic of the BN would represent the event "the component failures cause the sensor to show this patterns unless with a certain probability the sensor has failed" rather than "if certain component failures occur then, with probability 1,

the sensor shows this behaviour". In the thesis this modification was not included in order to assess the results from the method with the simulation codes.

- The other aspect that regards the sensors is their location when considering their whole life costs. Sensors are an important part of any system, the more the sensors the more accurate a diagnosis will be. However they are also generally expensive and their number is kept to an optimal minimum. Some more work could be done to understand, given a limited number of sensors available, which is their optimal localisation on the system or which number of sensors is the best considering a cost/effect ratio.
- Further work could be done applying the diagnostic method developed for the fuel rig on the real system in real time. Some failures can be induced in the laboratory rig such as blockages in the pipe sections, pump failures or pipe leakage. A code could be developed to directly accept the sensor information from the system for a period of time after a failure has been induced and run automatically the BN with the sensor evidence. The code should also include a pattern recognition routine that converts the observed sensor trends into an identified pattern.
- The method has shown to be applicable to systems of the size of the fuel rig. However the modularisation has shown that by dividing a system in sub-systems, the analysis can be carried out in two stages. This approach could also be taken on very large systems, such as aircrafts, iterating the division until a sub-system of a manageable size is reached (for example, considering 6 sensors as in the case of the sub-systems of the fuel rig). This could be proved on a specific application.
- The BN strategy was applied on two systems that are of different size but they are similar in the component types and structure. In order to prove its adaptability and generality, a study could be carried out on systems of different nature and scope.

References

[1] Y. Papadopoulos and J. McDermid, "Automated safety monitoring: A review and classification of methods," *International Journal of COMADEM*, vol. 4, no. 4, pp. 14–32, 2001.

[2] R. Isermann, "Process fault detection based on modeling and estimation methods–A survey," *Automatica*, vol. 20, no. 4, pp. 387–404, 1984.

[3] V. Venkatasubramanian, R. Rengaswamy, K. Yin, and S. Kavuri, "A review of process fault detection and diagnosis Part I: Quantitative model-based methods," *Computers and chemical engineering*, vol. 27, no. 3, pp. 293–311, 2003.

[4] V. Venkatasubramanian, R. Rengaswamy, and S. Kavuri, "A review of process fault detection and diagnosis Part II: Qualitative models and search strategies," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 313–326, 2003.

[5] V. Venkatasubramanian, R. Rengaswamy, S. Kavuri, and K. Yin, "A review of process fault detection and diagnosis Part III: Process history based methods," *Computers and Chemical Engineering*, vol. 27, no. 3, pp. 327–346, 2003.

[6] C. Price, *Computer-based diagnostic systems*. Springer Verlag, 1999.

[7] J. Kolodner, "An introduction to case-based reasoning," *Artificial Intelligence Review*, vol. 6, no. 1, pp. 3–34, 1992.

[8] O. Kipersztok and G. Dildy, "Evidence-based Bayesian networks approach to airplane maintenance," in *Neural Networks, 2002. IJCNN'02. Proceedings of the 2002 International Joint Conference on*, vol. 3, 2002.

[9] T. McGinnity, W. Fenton, and L. Maguire, "Fault diagnosis of electronic systems using intelligent techniques: A review," *IEEE TRANS SYST MAN CYBERN PT C APPL REV*, vol. 31, no. 3, pp. 269–281, 2001.

[10] C. Price and N. Taylor, "Multiple fault diagnosis from FMEA," in *PROCEEDINGS OF THE NATIONAL CONFERENCE ON ARTIFICIAL INTELLIGENCE*, pp. 1052–1057, Citeseer, 1997.

[11] C. Price and N. Taylor, "Automated multiple failure FMEA," *Reliability engineering and system safety*, vol. 76, no. 1, pp. 1–10, 2002.

[12] C. Price, "AutoSteve: automated electrical design analysis," in *ECAI*, pp. 721–725, Citeseer, 2000.

[13] C. Price and N. Snooke, "An Automated Software FMEA," in *Proceedings of the International System Safety Regional Conference, Singapore*, 2008.

[14] E. Hurdle, L. Bartlett, and J. Andrews, "System fault diagnostics using fault tree analysis," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 221, no. 1, pp. 43–55, 2008.

[15] E. Hurdle, L. Bartlett, and J. Andrews, "Fault tree based fault diagnostics for dynamic systems," *Risk, reliability and societal safety: proceedings of the European safety and reliability conference: risk, reliability and societal safety*, 2007.

[16] J. Andrews, "The use of not logic in fault tree analysis," *Quality and Reliability Engineering International*, vol. 17, no. 3, pp. 143–150, 2001.

[17] F. Jensen and T. Nielsen, *Bayesian networks and decision graphs*. ASA, 2001.

[18] M. Lampis and J. Andrews, "Bayesian belief networks for system fault diagnostics," *Quality and Reliability Engineering International*, 2008.

[19] H. Kirsch and K. Kroschel, "Applying Bayesian networks to fault diagnosis," in *Control Applications, 1994., Proceedings of the Third IEEE Conference on*, pp. 895–900, 1994.

[20] H. Limin, Z. Yongli, L. Ran, and Z. Liguo, "Novel method for power system fault diagnosis based on Bayesian networks," in *Power System Technology, 2004. PowerCon 2004. 2004 International Conference on*, vol. 1, 2004.

[21] C. Shi, R. Zhang, and G. Yang, "Fault Diagnosis of AUV Based on Bayesian Networks," *Computer and Computational Sciences, 2006. IMSCCS'06. First International Multi-Symposiums on*, vol. 2, 2006.

[22] E. Castillo, J. Sarabia, C. Solares, and P. Gomez, "Uncertainty analyses in fault trees and Bayesian networks using FORM/SORM methods," *Reliability Engineering and System Safety*, vol. 65, no. 1, pp. 29–40, 1999.

[23] T. Mast, A. Reed, S. Yurkovich, M. Ashby, and S. Adibhatla, "Bayesian belief networks for fault identification in aircraft gas turbine engines," in *Control Applications, 1999. Proceedings of the 1999 IEEE International Conference on*, vol. 1, 1999.

[24] D. Heckerman *et al.*, "A tutorial on learning with Bayesian networks," *NATO ASI SERIES D BEHAVIOURAL AND SOCIAL SCIENCES*, vol. 89, pp. 301–354, 1998.

[25] J. Matsuura and T. Yoneyama, "Learning Bayesian networks for fault detection," in *Machine Learning for Signal Processing, 2004. Proceedings of the 2004 14th IEEE Signal Processing Society Workshop*, pp. 133–142, 2004.

[26] F. Sahin, M. Yavuz, Z. Arnavut, and Ö. Uluyol, "Fault diagnosis for airplane engines using Bayesian networks and distributed particle swarm optimization," *Parallel Computing*, vol. 33, no. 2, pp. 124–143, 2007.

[27] K. Przytula and D. Thompson, "Construction of Bayesian networks for diagnostics," in *Proceedings of 2000 IEEE Aerospace Conference*, vol. 24, 2000.

[28] G. Nunnari, F. Cannavò, and R. Vrânceanu, "Bayesian Networks Approach for a Fault Detection and Isolation Case Study," *Applied Soft Computing Technologies: The Challenge of Complexity*, pp. 173–183, 2006.

[29] J. Cheng, D. Bell, and W. Liu, "An algorithm for Bayesian belief network construction from data," in *proceedings of AI & STAT97*, pp. 83–90, Citeseer, 1997.

[30] Z. Yongli, H. Limin, and L. Jinling, "Bayesian networks-based approach for power systems fault diagnosis," *IEEE Transactions on Power Delivery*, vol. 21, no. 2, pp. 634–639, 2006.

[31] C. Rojas-Guzman and M. Kramer, "Comparison of belief networks and rule-based expert systems for fault diagnosis of chemical processes," *Engineering Applications of Artificial Intelligence*, vol. 6, pp. 191–191, 1993.

[32] N. Santoso, C. Darken, G. Povh, and J. Erdmann, "Nuclear plant fault diagnosis using probabilistic reasoning," in *IEEE Power Engineering Society Summer Meeting, 1999*, vol. 2, 1999.

[33] J. Pearl, *Causality: models, reasoning, and inference*. Cambridge university press, 2000.

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

[35] H. Langseth, "Bayesian networks with applications in reliability analysis," tech. rep., Technical Report PhD Thesis, Dept. of Mathematical Sciences, Norwegian University of Science and Technology, 2002, 2002.

[36] F. Palmer, R. Sterritt, C. Shapcott, E. Curran, and K. Adamson, "Exploring Dynamic Belief Network Visualisation," in *Conference on Artificial Intelligence and Soft Computing–ASC 2000*, 2000.

[37] R. Sterritt, A. Marshall, C. Shapcott, and S. McClean, "Exploring dynamic Bayesian belief networks for intelligent fault management systems," in *2000 IEEE International Conference on Systems, Man, and Cybernetics*, vol. 5, 2000.

[38] K. Murphy, *Dynamic Bayesian networks: representation, inference and learning*. PhD thesis, UNIVERSITY OF CALIFORNIA, 2002.

[39] A. Bobbio, L. Portinale, M. Minichino, and E. Ciancamerla, "Comparing fault trees and Bayesian networks for dependability analysis," *Lecture Notes in Computer Science*, pp. 310–322, 1999.

[40] A. Bobbio, L. Portinale, M. Minichino, and E. Ciancamerla, "Improving the analysis of dependable systems by mapping fault trees into Bayesian networks," *Reliability Engineering & System Safety*, vol. 71, no. 3, pp. 249–260, 2001.

[41] W. Vesely, F. Goldberg, N. Roberts, and D. Haasl, "Fault Tree Handbook," 1981.

[42] J. Andrews and T. Moss, *Reliability and risk assessment*. Professional Engineering Pub London, 2002.

[43] J. Andrews, "Birnbaum and criticality measures of component contribution to the failure of phased missions," *Reliability Engineering & System Safety*, vol. 93, no. 12, pp. 1861–1866, 2008.

[44] J. Andrews and L. Bartlett, "Genetic algorithm optimization of a firewater deluge system," *Quality and reliability engineering international*, vol. 19, no. 1, 2003.

[45] J. Andrews, "The use of not logic in fault tree analysis," *Quality and Reliability Engineering International*, vol. 17, no. 3, pp. 143–150, 2001.

[46] E. Hurdle, *System Fault Diagnosis Using Fault Tree Analysis*. PhD thesis, Loughborough University, 2008.

[47] G. Weidl, A. Madsen, and S. Israelson, "Applications of object-oriented Bayesian networks for condition monitoring, root cause analysis and decision support on operation of complex continuous processes," *Computers & chemical engineering*, vol. 29, no. 9, pp. 1996–2009, 2005.

[48] P. Bennett, J. Pearson, A. Martin, R. Dixon, M. Walsh, M. Khella, and R. Goodall, "Application of diagnostic techniques to an experimental aircraft fuel rig," *6th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes (SAFEPROCESS) 2006*, 2006.

[49] P. Bennett, R. Dixon, and J. Pearson, "Comparing residual evaluation methods for leak detection on an aircraft fuel system test-rig," *Systems Science*, vol. 34, no. 2, pp. 63–74, 2008.

[50] C. Kadie, D. Hovel, and E. Horvitz, "MSBNx: A component-centric toolkit for modeling and inference with Bayesian networks," *Microsoft Research, Richmond, WA, Technical Report MSR-TR-2001-67*, vol. 28, 2001.

[51] A. Madsen, M. Lang, U. Kjærulff, and F. Jensen, "The Hugin tool for learning Bayesian networks," *Symbolic and Quantitative Approaches to Reasoning with Uncertainty*, pp. 594–605, 2003.

Appendix A

Bayesian Networks for the Water tank system

In this appendix the BNs for section 2, 3 and 4 of the water tank system are shown. Section 1 BN is represented in figure 4.9 of chapter 4.

A.1 Bayesian Network for Section 2

Section 2 of the water tank system comprehends two FTs for the event *Flow* and *Low Flow*. The two output nodes in figure A.1 represent the top events, while the input nodes at the top are the components of the section.

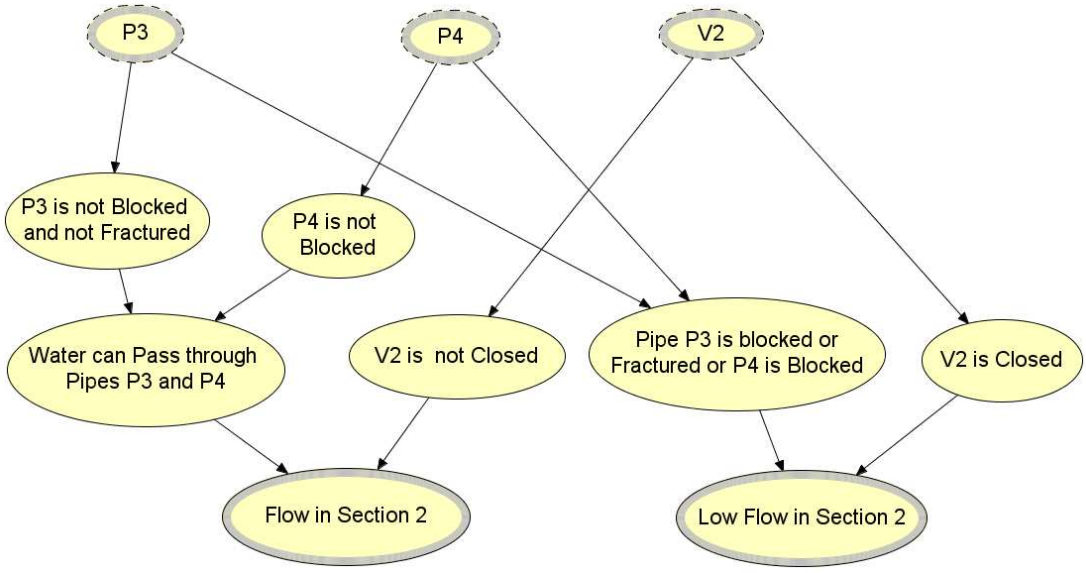


Figure A.1 – BN for section 2 of the water tank system.

A.2 Bayesian Network for Section 3

The BN for section 3 is represented in figure A.2. This is constituted by two BNs modelling *High Flow* and *Low Flow* in the section.

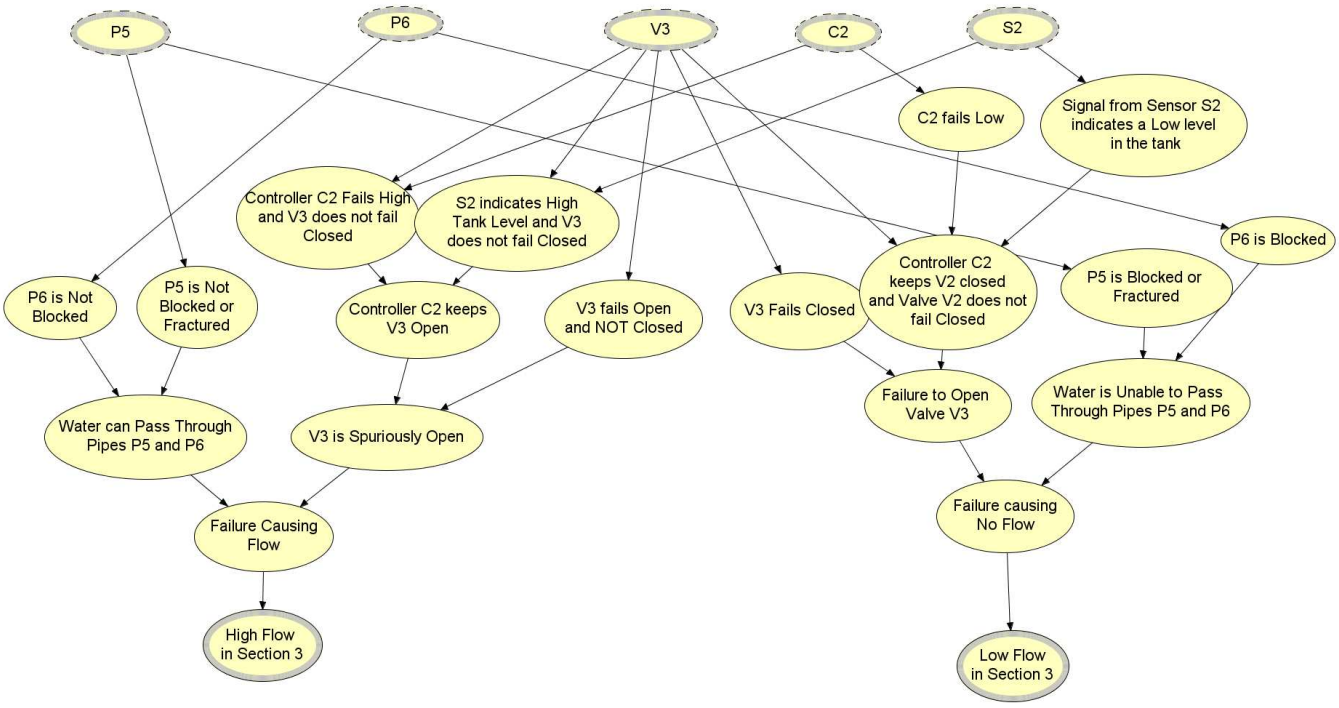


Figure A.2 – BN for section 3 of the water tank system.

A.3 Bayesian Network for Section 4

The BN for the tray/tank section is very simple as the section is its failing state when there is water in the overspill tray that is caused by a rupture or a leak in the tank.

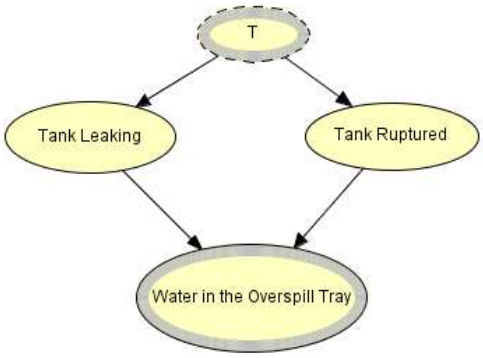


Figure A.3 – BN for section 4 of the water tank system.

Appendix B

Bayesian Networks for the Fuel Rig system

In this appendix the BNs for the main tank sub-systems are shown for phase 4 of the ACTIVE operating mode. The BNs for the drainage line and for line L1 are represented in figures 5.7 and 5.8 of chapter 5. In the following, the BNs for line L2, for the recycle lines and for the outflow line are displayed.

The BN for line L2 is shown in figure B.1. As No flow is expected in line L2 when the system is in phase 4, the unexpected readings are *Flow* and *Partial Flow*. The BN comprehends two subnetworks for the two unexpected behaviours. These are displayed in other two figures to be better readable, figure B.2 and B.3.

The BN for the recycle line L1 is shown in figure B.4. In phase 4, *No Flow* is expected through the sensor in the recycle line L1, therefore the unexpected patterns are *Flow* and *Partial Flow*, these are represented in figure B.5 and B.6 respectively.

Recycle line L2 BN is represented in figure B.7. As the expected behaviour through its sensor is *Flow*, the BN includes the FTs whose top events are *No Flow* and *Partial Flow*. The separate BNs for these events are represents in figures B.8 and B.9.

The sensor located in the outflow line should detect full flow through when the system is in phase 4. Therefore, the unexpected patterns for the outflow line are *Partial Flow* and *No Flow*. The BN in figure B.10 shows the entire BN while figures B.11 and B.12 represent the two subnetworks.

B.1 Bayesian Networks for Line 2

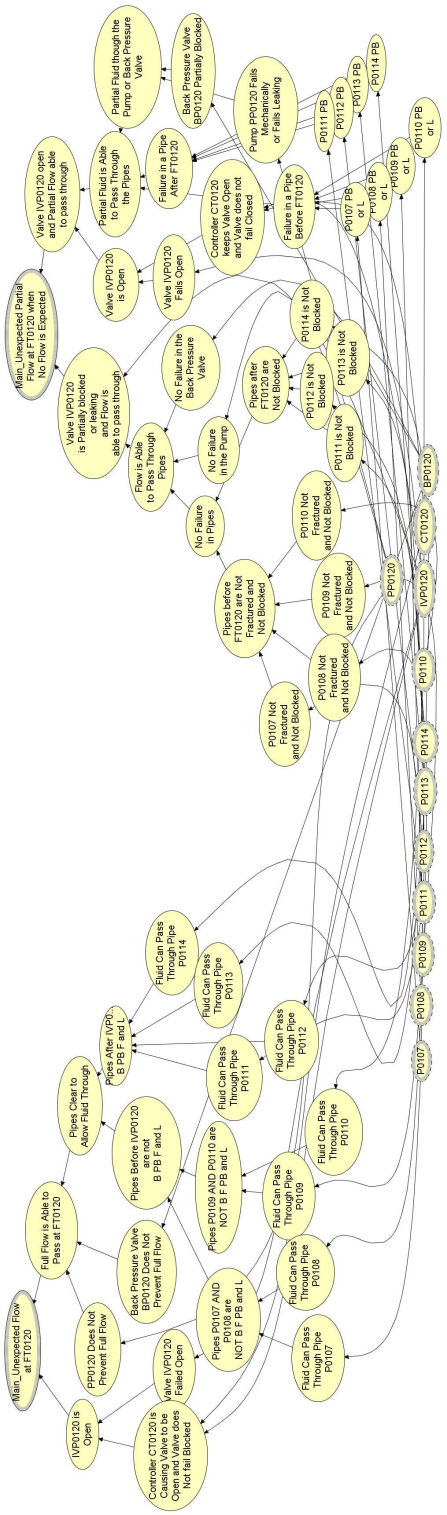


Figure B.1 – BN for line L2 section of the main tank for phase 4.

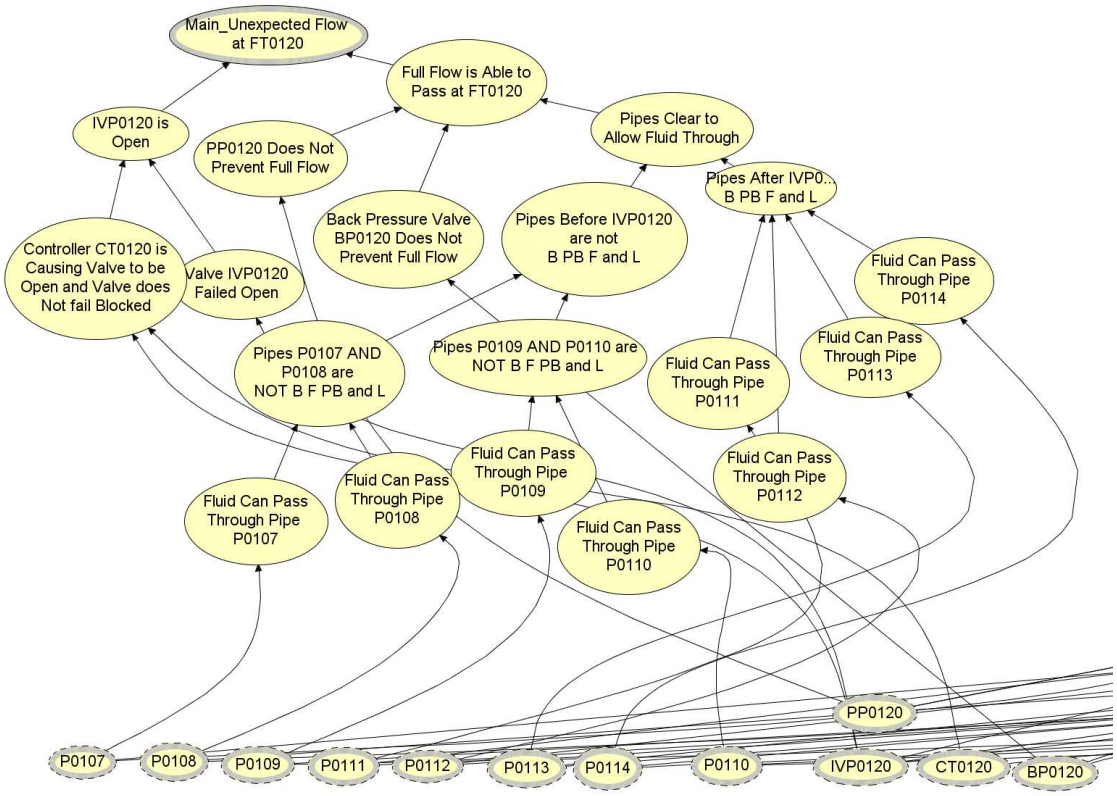


Figure B.2 – BN for *Flow* in line L1 section of the main tank for phase 4.

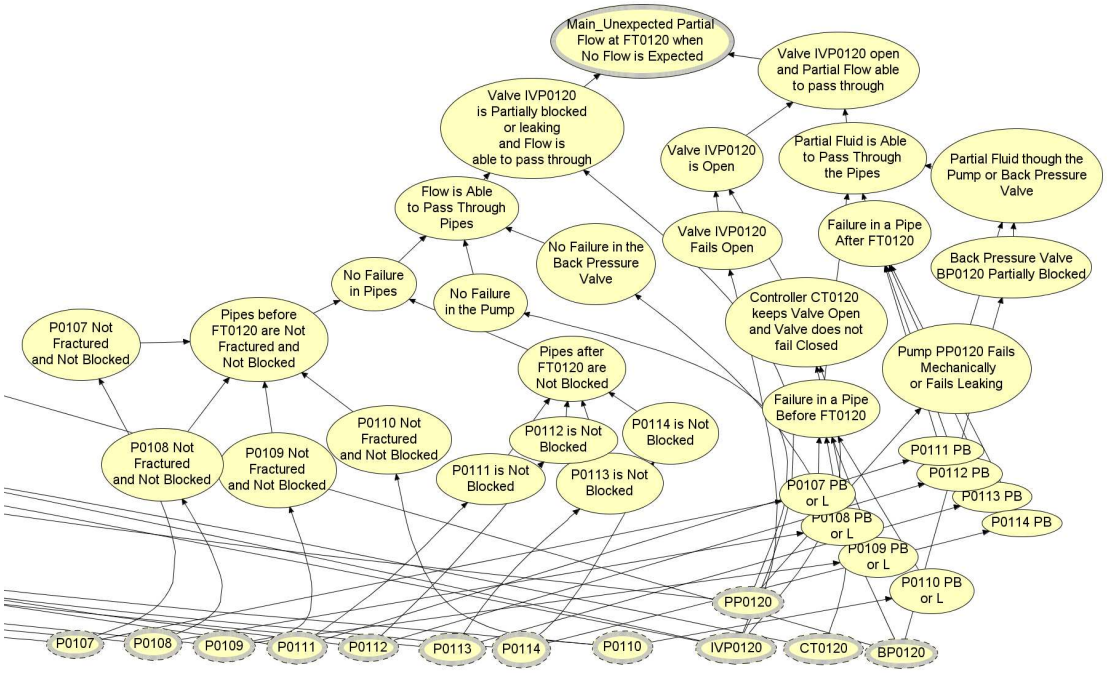


Figure B.3 – BN for *Partial Flow* in line L1 section of the main tank for phase 4.

B.2 Bayesian Networks for the Recycle line L1

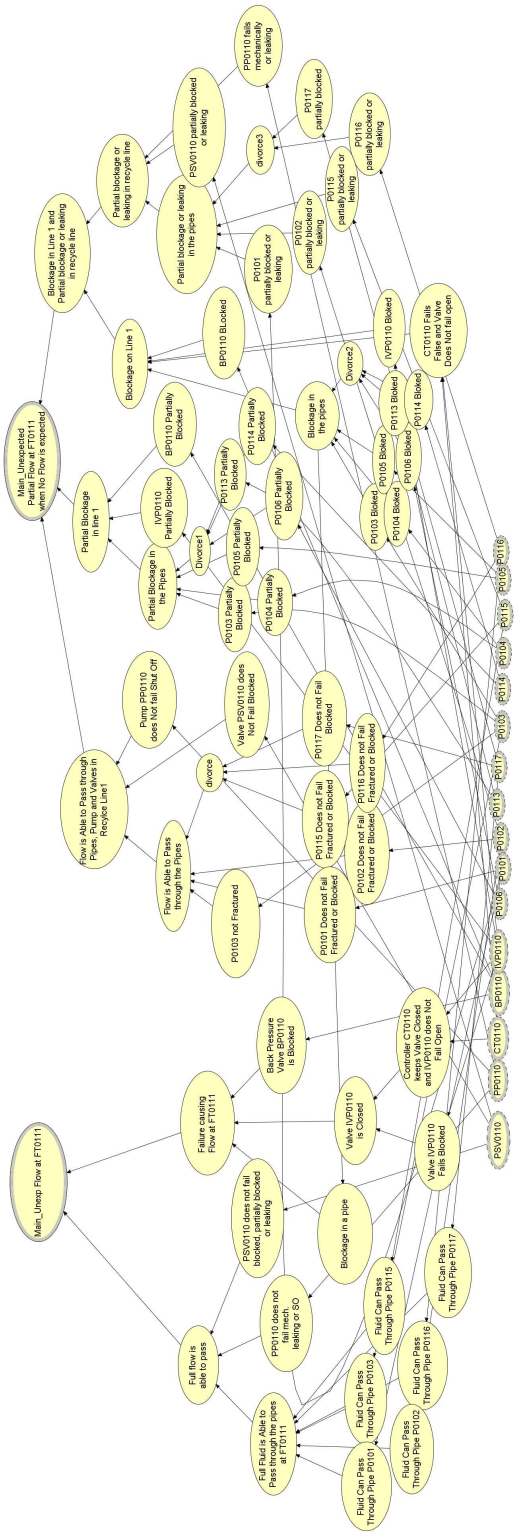


Figure B.4 – BN for recycle line L1 section of the main tank for phase 4.

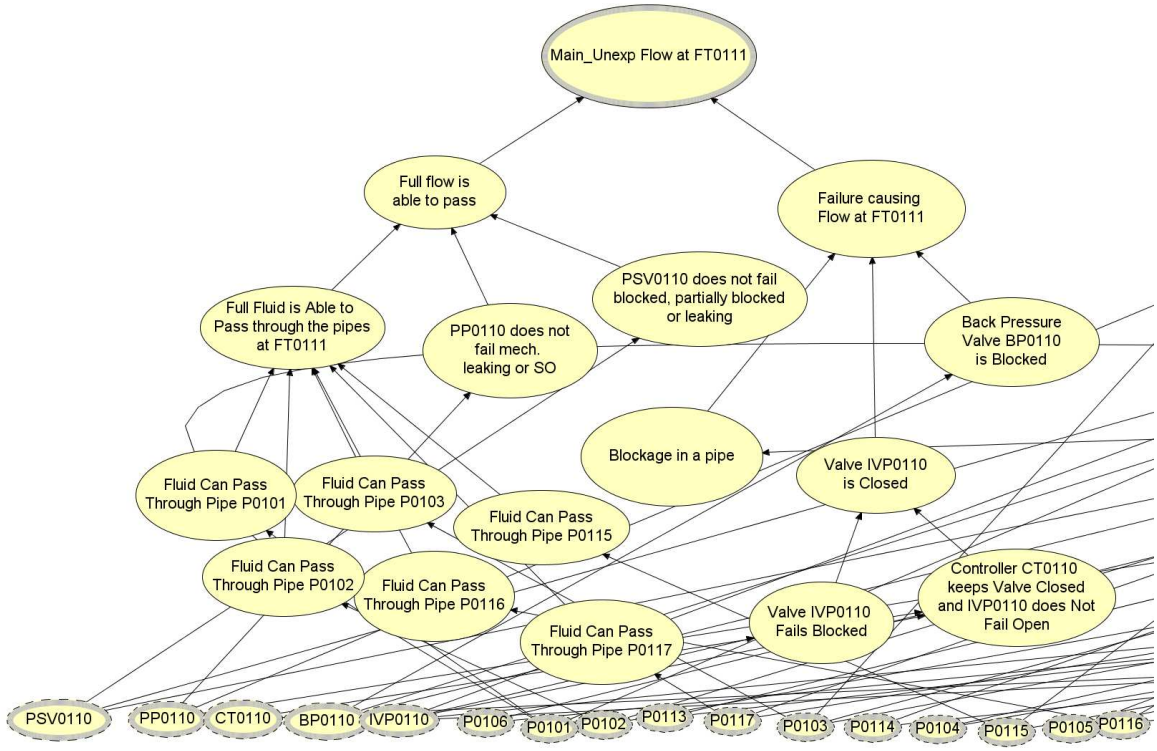


Figure B.5 – BN for *Flow* in the recycle line L1 section of the main tank for phase 4.

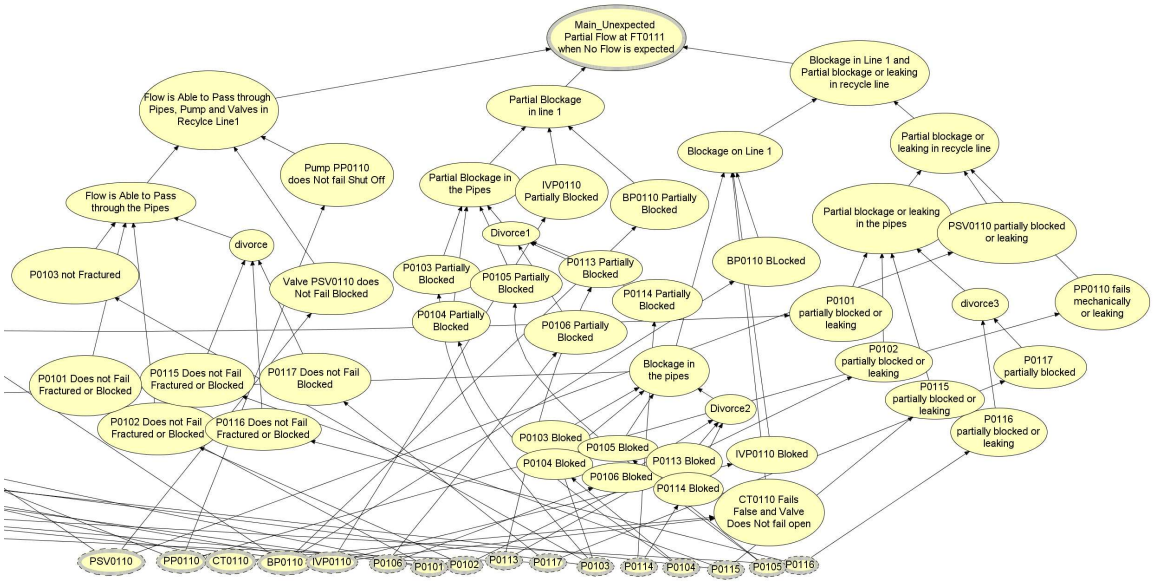


Figure B.6 – BN for *Partial Flow* in the recycle line L1 section of the main tank for phase 4.

B.3 Bayesian Networks for the Recycle line L2

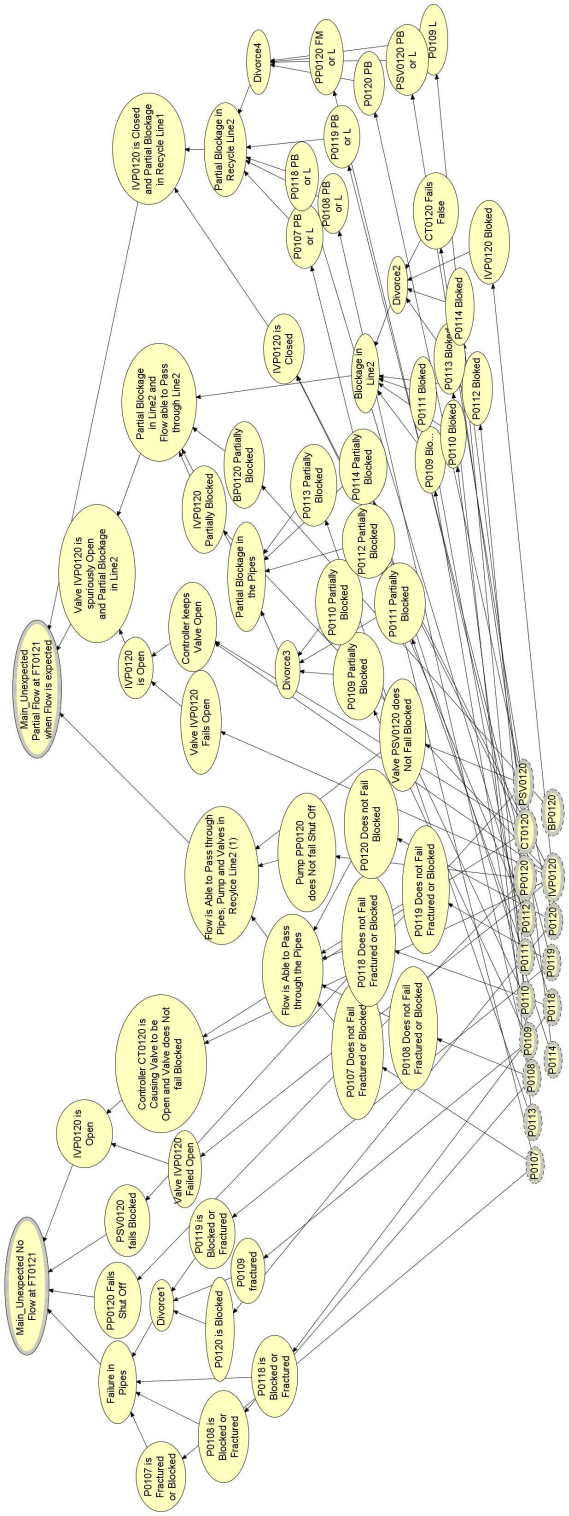


Figure B.7 – BN for recycle line L2 section of the main tank for phase 4.

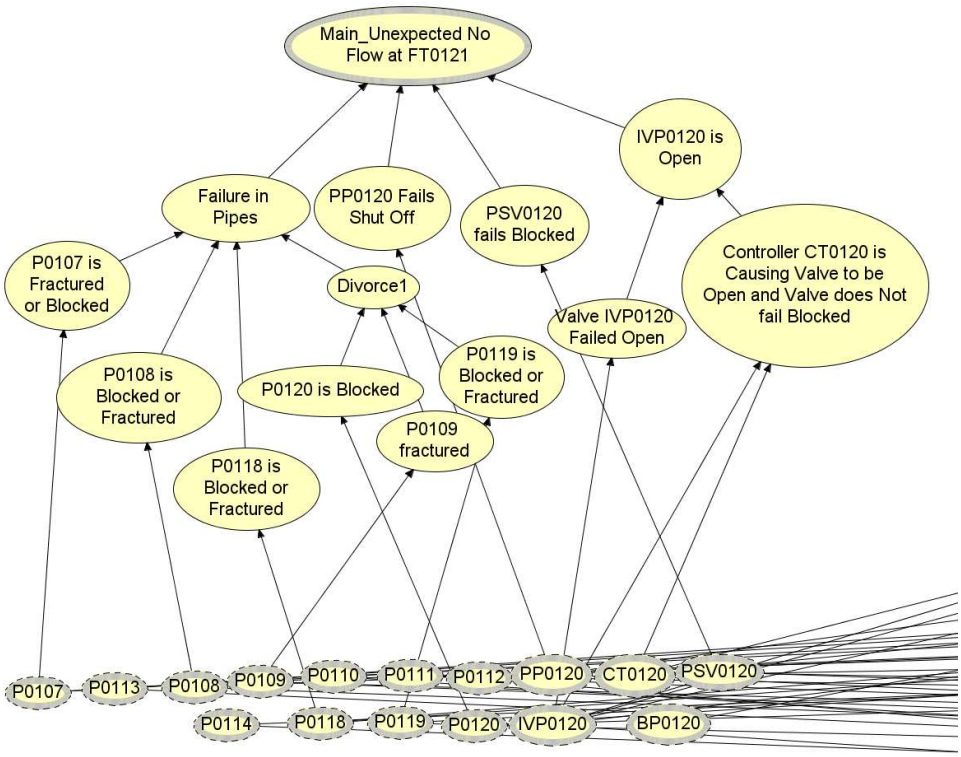


Figure B.8 – BN for *No Flow* in the recycle line L2 section of the main tank for phase 4.

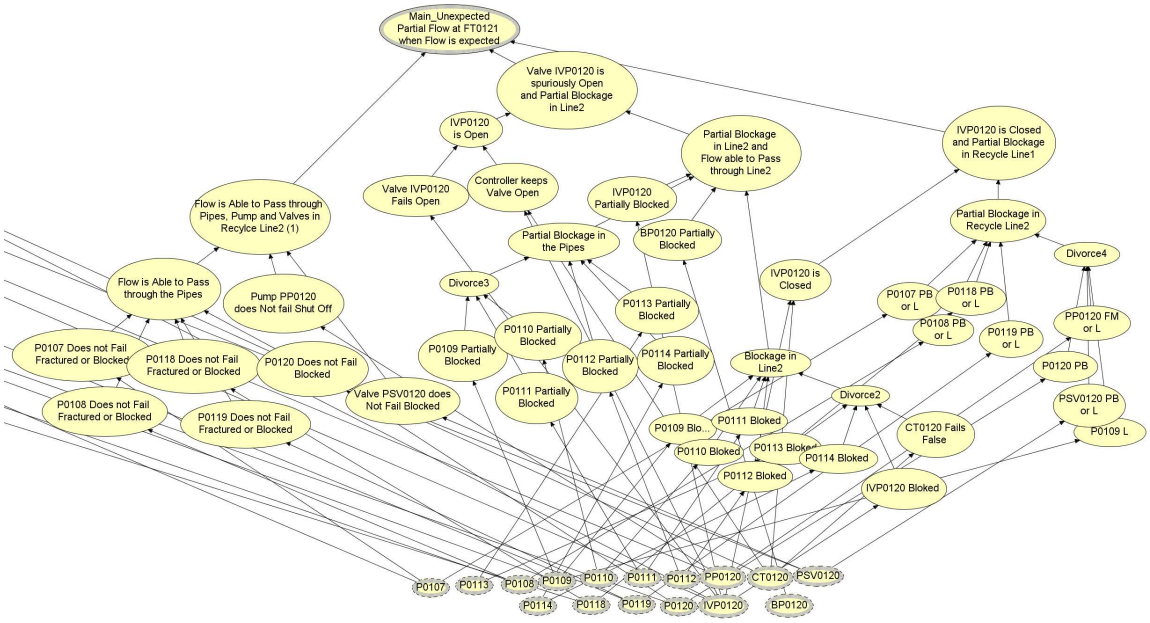


Figure B.9 – BN for *Partial Flow* in the recycle line L2 section of the main tank for phase 4.

B.4 Bayesian Networks for the Outflow line

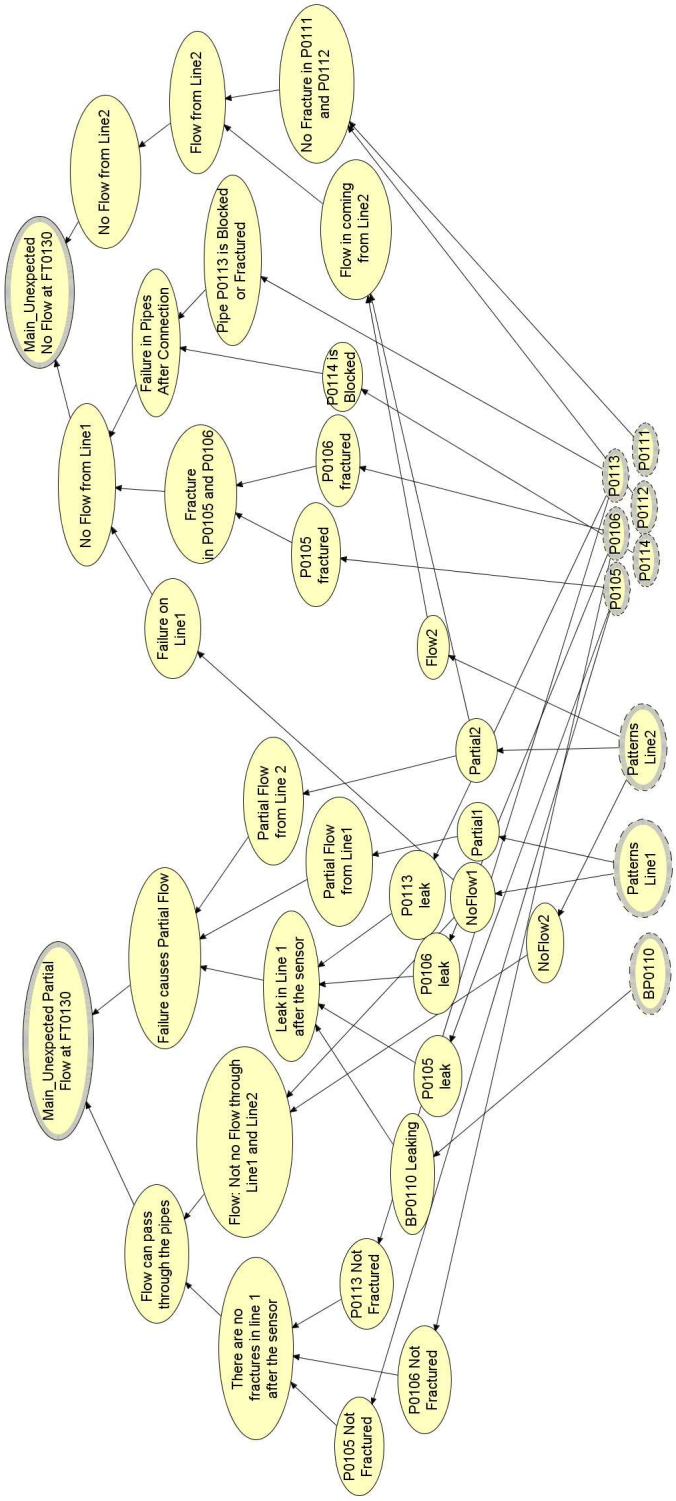


Figure B.10 – BN for line L2 section of the main tank for phase 4.

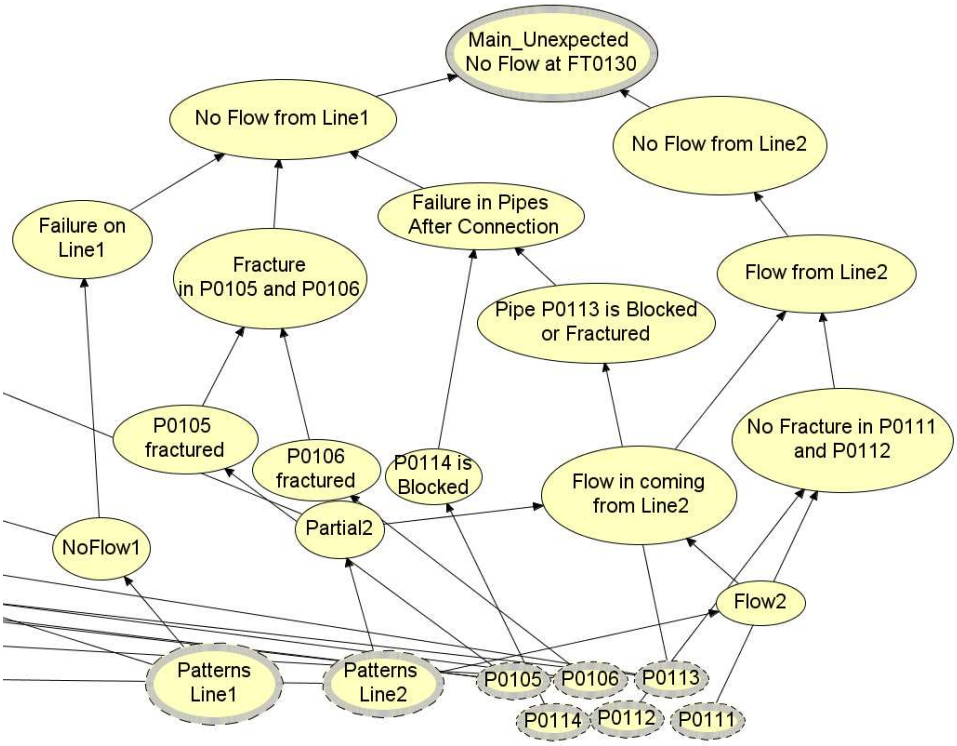


Figure B.11 – BN for *No Flow* in the outflow line section of the main tank for phase 4.

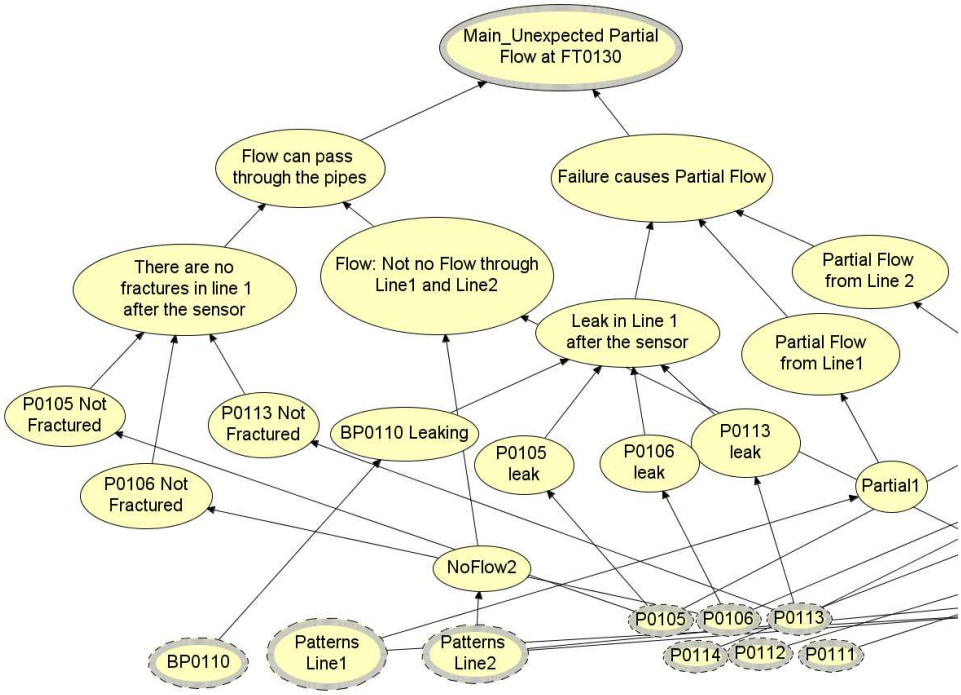


Figure B.12 – BN for *Partial Flow* in the outflow line section of the main tank for phase 4.

Appendix C

Fault Trees and Bayesian Networks Softwares

In this appendix the softwares used for the thesis work are presented and briefly described. Fault Trees have been built and evaluated using *Fault Tree Plus* (or *FaultTree+*), while two softwares were used for Bayesian Networks: *MSBNx* and *Hugin Researcher*. The following sections explain the features and characteristics of each of them.

C.1 Fault Tree Plus

Fault Tree Plus is a widely used software for the construction and evaluation of FTs as well as Event Tree and for Markov analysis. In the thesis work, it has been used to produce the graphical representation of the FTs and to calculate the unreliability of the top events as well as the importance measures of the components.

C.2 MSBNx

MSBNx is a free application for the creation and evaluation of BNs [50]. It was used in the initial phases of the research. The figures and the simple calculations of the example in the first two chapter are performed with this software. Among the free softwares available, MSBNs has been found to be straightforward in the graphical interface and in the probabilities features. However, it has many limitations in particular in terms of the size of the networks that can be created. For this reason, Hugin was chosen for the development of the BNs of the water tank system and the fuel rig system.

C.3 Hugin Researcher

Hugin produces several commercial softwares for general decision support tools using statistical models [51]. *Hugin Researcher* aims at academics and it allows advanced applications of BNs. The graphical interface permits the implementation of Object Oriented Bayesian Networks, this was particularly useful for the thesis as it enables to handle large size BNs. The software also incorporates automated learning from databases.