

This item was submitted to Loughborough's Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.



creative commons  
COMMONS DEED

**Attribution-NonCommercial-NoDerivs 2.5**

**You are free:**

- to copy, distribute, display, and perform the work

**Under the following conditions:**

 **Attribution.** You must attribute the work in the manner specified by the author or licensor.

 **Noncommercial.** You may not use this work for commercial purposes.

 **No Derivative Works.** You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of these conditions can be waived if you get permission from the copyright holder.

**Your fair use and other rights are in no way affected by the above.**

This is a human-readable summary of the [Legal Code \(the full license\)](#).

[Disclaimer](#) 

For the full text of this licence, please go to:  
<http://creativecommons.org/licenses/by-nc-nd/2.5/>

# Proceedings of the 16<sup>th</sup> Advances in Reliability Technology Symposium

## System Fault Diagnostics Using Fault Tree Analysis

E. E. Hurdle, L. M. Bartlett, J. D. Andrews

Aeronautical and Automotive Engineering, Loughborough University,  
Loughborough, UK

### Abstract

Over the last 50 years advances in technology have led to an increase in the complexity and sophistication of systems. More complex systems can be harder to maintain and the root cause of a fault more difficult to isolate.

Down-time resulting from a system failure can be dangerous or expensive depending on the type of system. In aircraft systems the ability to quickly diagnose the causes of a fault can have a significant impact on the time taken to rectify the problem and return the aircraft to service. In chemical process plants the need to diagnose causes of a safety critical failure in a system can be vital and a diagnosis may be required within minutes. Speed of fault isolation can save time, reduce costs and increase company productivity and therefore profits. System fault diagnosis is the process of identifying the cause of a malfunction by observing its effect at various test points.

Fault tree analysis (FTA) is a method that describes all possible causes of a specified system state in terms of the state of the components within the system. A system model is used to identify the states the system should be in at any point in time. This paper presents a method for diagnosing faults in systems using FTA to explain the deviations from normal operation observed in sensor outputs. The causes of a system's failure modes will be described in terms of the component states. This will be achieved with the use of coherent and non-coherent fault trees. A coherent fault tree is constructed from AND and OR logic, therefore only considers component failed states. The non-coherent method expands this allowing the use of NOT logic which implies that the existence of component failed states and working states are both taken into account. This paper illustrates the concepts of this method by applying the technique to a simplified water tank level control system.

**Keywords:** fault diagnosis, fault tree analysis

### 1. Introduction

A system can be analysed for faults in two different ways. The first is to test the system functionality at one point in time. The second continuously monitors the system and detects faults as they occur.

An example of an approach that carries out a series of tests to determine the system status at one point in time is the sequential fault diagnostic tool developed by Novak *et al* [1-5]. The approach uses information about which symptoms are exhibited when the faults are present. The sequential fault diagnostic tool determines the best sequence to conduct the test to locate the fault condition in the cheapest (or fastest)

way. A similar method by Pattipati *et al* [6] uses heuristic algorithms in order to determine the most cost-effective sequence of tests. These methods are limited to situations where only a single fault is expected to exist at any point in time; they do not take into consideration multiple component failures. Shakeri *et al* [7] extended sequential test sequencing to diagnose multiple failures in systems. However, this method takes a considerable length of time to obtain a diagnosis.

Another approach to diagnosis is the use of graphical models to describe the propagation of faults in systems. Rao [8] developed two algorithms that use the information from directed graphs of systems to diagnose single failures at one point in time. This technique was developed further by Pattipati [9] to diagnose multiple failures in systems. This graphical method however does not take into consideration faults that do not immediately affect the status of the system when they occur.

Failure Modes Effects Analysis (FMEA) is a structured, qualitative analysis of a system, subsystem or function that can be used to identify potential system failure modes, their causes and the effects on the system operation associated with the failure modes occurrence. Price [10] demonstrated the use of automated FMEA to generate reports that could be used in a diagnostic tool to diagnose multiple faults in systems at one point in time. The failures from the FMEA are only generated to a chosen likelihood of occurrence; therefore all possible outcomes of failure for a system scenario may not be obtained. Paasch and Mocko [11] use FMEA and FTA to develop a model that uses matrix manipulation for diagnosing faults in systems at one point in time. This research does not however consider multiple failures.

Papadopoulos [12] has carried out work using statecharts and fault trees to provide continuous on-line monitoring and rectification of systems. NOT logic is excluded from the fault trees; therefore only component failures are taken into account to obtain diagnosis. As a result, some faults occurring simultaneously have required conflicting remedial procedures. Yangping [13] also developed a fault tree based method that only considers component failures, which uses genetic algorithms to continuously monitor for faults in nuclear power plants. Genetic search is slow in obtaining solutions and there can be problems determining when a global rather than a local diagnosis has been obtained.

Many system failures are not usually the result of one single fault. Therefore the ability to diagnose multiple faults is vitally important. Methods of finding faults or combinations of faults as they occur are the subject of this paper. The approach is based on the fault tree method [14-16]. This is traditionally used to quantify the likelihood of a system failure. In this application the logic diagram is used to develop causes of a system symptom exhibited by a sensor reading, in terms of component conditions. In order to illustrate the features of the method described in this paper it will be applied to a simplified water tank level control system.

## 2. The Water Tank System

The water tank system is illustrated in Figure 1. It aims to maintain the level of water between two pre-determined limits. In normal operational mode water flows out of the system through valve V2. The level control system determines when water is flowing from the tank and then refills it by opening valve V1 until the desired tank level is reached. The overspill tray located beneath the tank catches water if the tank has ruptured, if water has leaked out through a crack or hole, or if the water level overflows from the top.

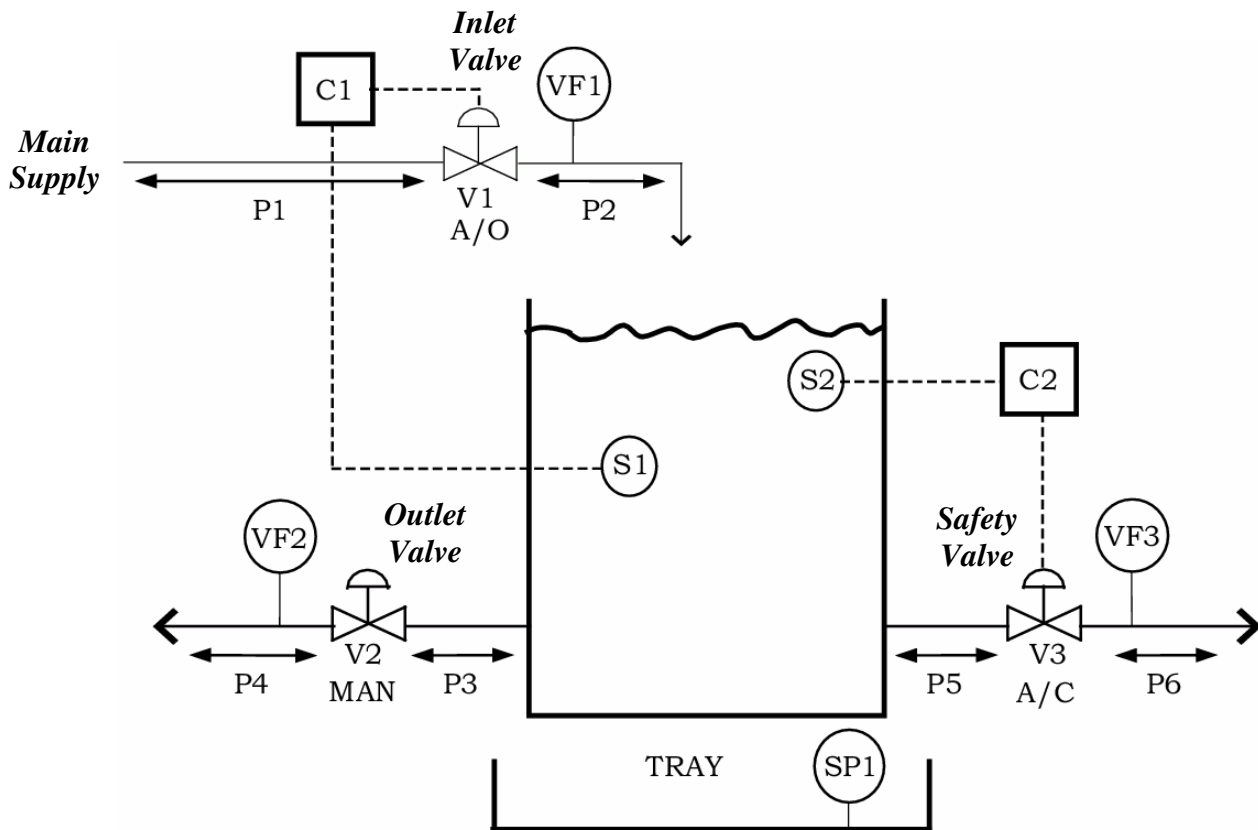


Figure 1: Water tank system

### 2.1 System Component Description

The water tank system shown in Figure 1 consists of three valves labelled V1, V2 and V3, two level sensors represented by S1 and S2, two controllers C1 and C2, and an overspill tray denoted TRAY. There are six sections of pipes identified by the labels P1 to P6.

V1 is an air-to-open (A/O) inlet valve controlled by C1. The level sensor S1 detects the height of the water in the tank. In normal operating mode if the water in the tank falls below the required level (as indicated by the sensor S1), the controller C1 would open valve V1 allowing water into the tank. Conversely, if the water in the tank rises to the required level then C1 will close V1.

V2 is a manual (MAN) valve, controlled by an operator in response to demand. Finally valve V3 is an air-to-close valve (A/C) that operates as a safety valve controlled by C2. Normally this will only become operational when a component failure occurs that causes a very high level of water in the tank. A signal from S2 would cause the controller C2 to open valve V3 to reduce the level of water in the tank.

The overspill tray, located underneath the tank collects any spillages that may occur due to a failure in the system. So, water in the overspill tray will occur if the tank has ruptured, is leaking, or if the water level overflows from the top of the tank.

## 2.2 System Operating Modes

### 2.2.1 Sensor Locations

The status of the system is determined using measurements provided by flow sensors situated next to each of the three valves in the system. The sensors are denoted by VF1, VF2 and VF3 for locations at V1, V2 and V3 respectively. These each detect the presence or absence of flow of water, which can be denoted as flow 'F', or no flow 'NF' respectively. For this demonstration study it is assumed the sensors are perfectly reliable. A fourth sensor denoted by SP1 is located in the overspill tray to indicate if any water has escaped from the tank. Its reading is interpreted as water 'W', or no water 'NW'. The sensor locations described above are called the system observation points. The observation points for V1, V2, V3 and TRAY generated 16 different scenarios that the system potentially could produce. These are listed in Table 1.

Scenario	V1	V2	V3	TRAY	Scenario	V1	V2	V3	TRAY
1	F	F	F	W	9	NF	F	F	W
2	F	F	F	NW	10	NF	F	F	NW
3	F	F	NF	W	11	NF	F	NF	W
<b>4</b>	<b>F</b>	<b>F</b>	<b>NF</b>	<b>NW</b>	12	NF	F	NF	NW
5	F	NF	F	W	13	NF	NF	F	W
6	F	NF	F	NW	14	NF	NF	F	NW
7	F	NF	NF	W	15	NF	NF	NF	W
8	F	NF	NF	NW	<b>16</b>	<b>NF</b>	<b>NF</b>	<b>NF</b>	<b>NW</b>

Table 1: System scenarios

Were this a real system additional sensors could be added in order to provide a more complete picture of its operating state. The level control sensors S1 and S2 could also be used. However for the purposes of demonstrating the fault tree based fault diagnostics technique the system sensors as described are sufficient, without any additional complexity.

The system has two operating modes; these being ‘ACTIVE’ when the operator opens valve V2, or ‘DORMANT’ when V2 is closed. In the ACTIVE operating mode water is taken out of the system through valve V2 and the tank is refilled by water coming in through valve V1 from the main water supply. Water would not exit the system through valve V3 and there would be no water in the overspill tray. The sensor readings for the system when ACTIVE should be those given in scenario 4, Table 1. In the DORMANT operating mode the system is effectively on standby with all three valves remaining closed and the overspill tray empty. This should give the sensor readings given in scenario 16, Table 1.

The sensor readings given by scenarios 4 and 16 are those which, under steady state conditions, represent the model of how the system should behave when ACTIVE and DORMANT respectively. Given that the system is in the ACTIVE or DORMANT state any sensor readings that deviate from those expected are regarded as being indicative of some fault within the system.

### 2.2.2 Possible Component Failures

In order to apply FTA to a system the faults that could occur for each of the system components need to be defined. Table 2 contains a list of possible component failures and their code.

The two operating modes are also represented in the fault trees. ACTIVE signifies the operator has attempted to open valve V2. DORMANT is used to indicate that the operator has tried to close V2. It should be noted that this is a two mode system and so only one of the variables ACTIVE or DORMANT can be true at any time.

Code	Component Failure	Code	Component Failure
$P_iB (1 \leq i \leq 6)$	- Pipe $P_i$ is Blocked	$C_iFH (1 \leq i \leq 2)$	- Controller $C_i$ Fails High
$P_iF (1 \leq i \leq 6)$	- Pipe $P_i$ is Fractured	$C_iFL (1 \leq i \leq 2)$	- Controller $C_i$ Fails Low
$V_iFC (1 \leq i \leq 3)$	- Valve $V_i$ Fails Closed	TR	- Water Tank Ruptured
$V_iFO (1 \leq i \leq 3)$	- Valve $V_i$ Fails Open	TL	- Water Tank Leaks
$S_iFH (1 \leq i \leq 2)$	- Sensor $S_i$ Fails High	NWMS	- No Water from Main
$S_iFL (1 \leq i \leq 2)$	- Sensor $S_i$ Fails Low		Supply

Table 2: Potential component failures

### 3. Sensor Deviation Models

In the application of FTA to the system fault diagnostics a series of failure logic diagrams are produced representing the causes of any sensors readings. These are developed in a fault tree in terms of the component failure conditions of the system operating state. The list of all possible sensor readings for the system is shown in Table 3.

Abbreviation	Sensor Readings	Abbreviation	Sensor Readings
FTV1	- Flow Through Valve V1	NFTV1	- No Flow Through Valve V1
FTV2	- Flow Through Valve V2	NFTV2	- No Flow Through Valve V2
FTV3	- Flow Through Valve V3	WOST	- Water in the Overspill Tray

Table 3: Sensor readings

Two other possible sensor readings for the system are ‘no flow through valve V3’ and ‘no water in the overspill tray’. However, these sensor readings would occur under normal operational conditions (i.e. without any failures occurring) in both the ACTIVE and the DORMANT operating states.

### 3.1 Fault Tree Construction

Fault trees were generated for each sensor reading listed in Table 3 using both coherent and non-coherent methods. Coherent fault trees are constructed from AND and OR logic and feature only component failure events in the causes of sensor status. The non-coherent method expands this allowing the use of the NOT operator. This implies that both the component failed and working states are taken into account when developing the causes of the sensor status.

#### 3.1.1 Coherent Fault Tree for the Sensor Reading Flow Through Valve V2

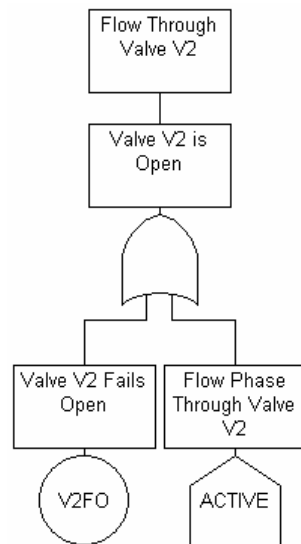


Figure 2: Coherent Fault Tree for the Sensor Reading Flow Through Valve V2

The coherent fault tree for the sensor reading ‘flow through valve V2’ is presented in Figure 2. Referring to Figure 1 ‘flow through valve V2’ can only occur because V2 is open. There are two possibilities for having V2 open, these being that either the valve has failed open, which is a basic event, or the system is in a flow phase and is therefore in the ACTIVE operating mode, represented by a house event. The two

prospective outcomes terminate the logic development – in this instance the fault tree is small.

### 3.1.2 Non-Coherent Fault Tree for the Sensor Reading Flow Through Valve V2

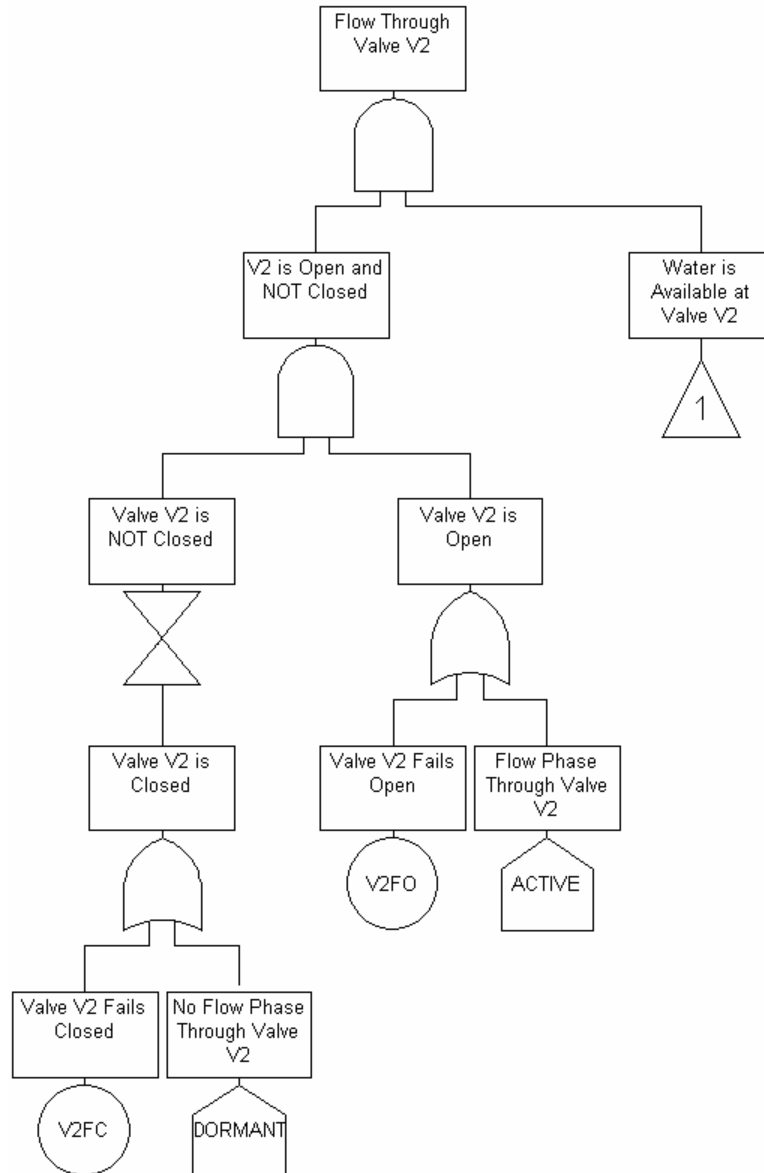


Figure 3a: Non-Coherent Fault Tree for the Sensor Reading Flow Through Valve V2



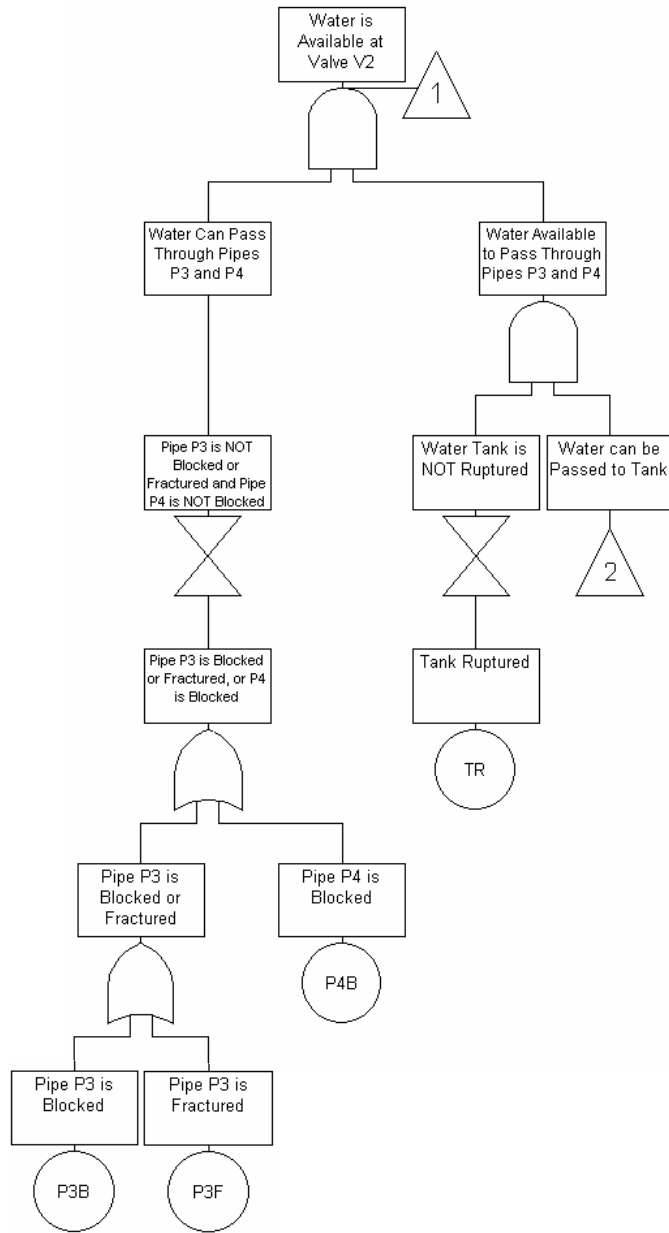


Figure 3b: Non-Coherent Fault Tree for the Sensor Reading Flow Through Valve V2 – Transfer 1

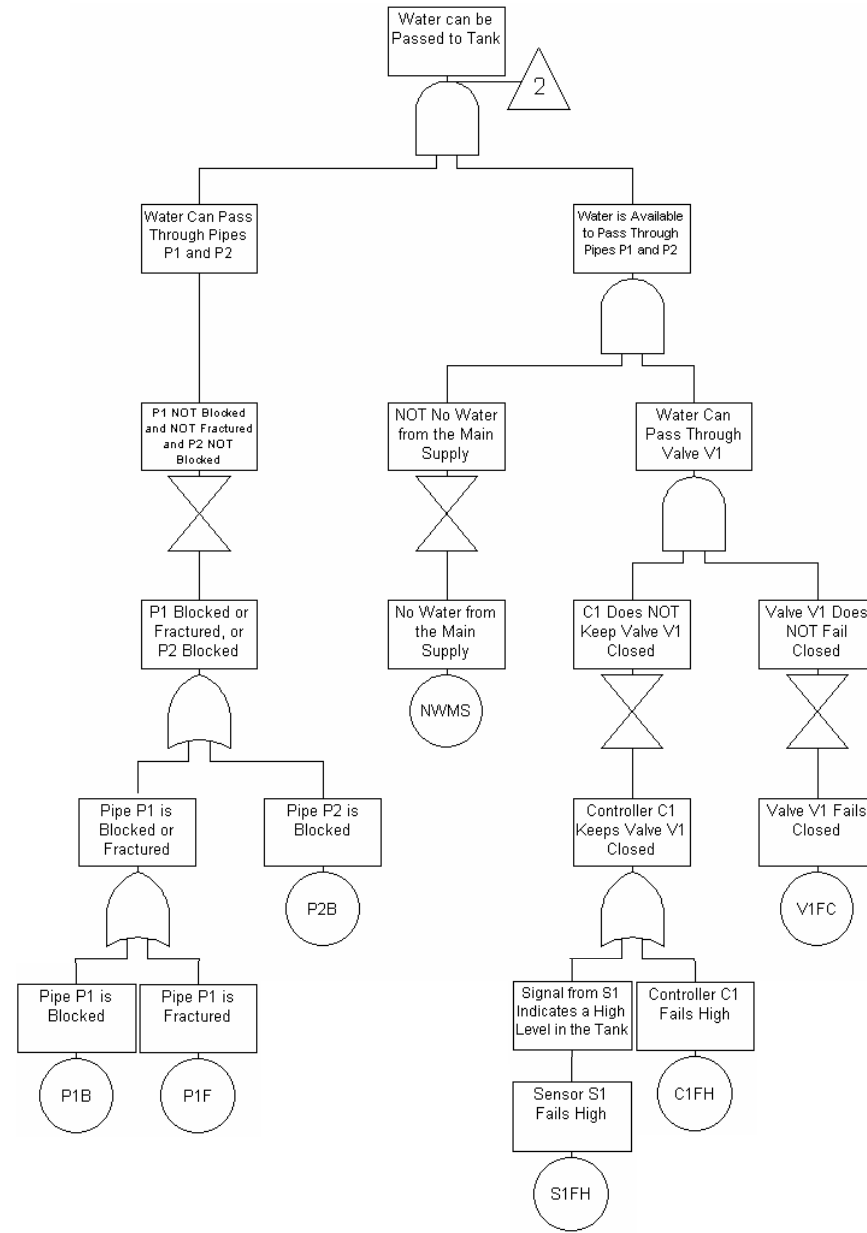


Figure 3c: Non-Coherent Fault Tree for the Sensor Reading Flow Through Valve V2 – Transfer 2

The non-coherent fault tree for the sensor reading ‘flow through valve V2’ is shown in Figures 3a to 3c. The information described by the coherent fault tree in Section 3.1.1 can now be expanded to include everything in the system that is known not to have failed. Figures 3a to 3c show that the introduction of NOT logic significantly increases the amount of information known about the system behaviour.

‘Flow through valve V2’ occurs because V2 is open and also because there is water available. As in the coherent fault tree V2 is open either because it has failed open or it is in a flow phase, therefore is in the ACTIVE operating mode. If V2 is open then it is definitely not closed. Therefore it cannot have failed closed or be in a no flow (DORMANT) phase.

As well as the valve being open water has to be available at valve V2 in order to get flow. Therefore water can pass through pipes P3 and P4, indicating that P3 is not blocked or fractured and P4 is not blocked. Water also must be available to pass from the tank to pipe P3. Therefore water is contained in the tank, indicating that it has not ruptured, and water is being supplied to the tank, indicating that pipes P1 and P2 are clear and water is coming into the tank. Therefore water is available from the main supply and water can be passed through valve V1. These non-failures together with the potential causes of water through V2 can be combined using an AND gate to form the non-coherent fault tree for the sensor reading.

### ***3.2 System Fault Detection***

The system was modelled by four different schemes, of differing complexity, in order to construct the scenarios generated from the system observation points (see Table 1). A fifth scheme that looks at switching the system operating mode is also discussed. It has been assumed throughout that the system was in a steady-state operating condition. Dynamic considerations are also discussed later (see Section 4.5).

#### **3.2.1 Scheme 1**

In this scheme coherent fault trees were used to develop causes of sensor deviations. At any given time the sensors must indicate one of the 16 scenarios listed in Table 1. If the system is ACTIVE the sensors should show the readings as in scenario 4 of Table 1. In the DORMANT state they should show the readings in scenario 16. When the actual readings do not match the pattern of the model it suggests a fault exists in at least one of the system components. To find all possible causes, a top event structure can be constructed from the information given by the system observation points. In this scheme only the observation points giving sensor readings that deviate from the normal operating mode of the system are considered. For a given set of sensor readings the causes of the scenario are developed as fault trees. These are constructed by combining the causes of sensor readings which deviate from that expected using an AND gate. This scheme is now demonstrated assuming the system is ACTIVE and the sensor readings from scenario 1 from Table 1 are obtained.

Table 4 below contains sensor readings expected when the system is in the ACTIVE operating mode (scenario 4) and the actual reading scenario 1.

Scenario	V1	V2	V3	TRAY
4 ACTIVE	F	F	NF	NW
1	F	F	<b>F</b>	<b>W</b>

Table 4: ACTIVE operating mode with expected and actual sensor readings

The table shows that the sensor reading for valve V3 is ‘flow’, when in the ACTIVE operating mode it should be ‘no flow’. Also the sensor SP1 in the overspill tray should read ‘no water’ but it is indicating that there is water present. Therefore these sensors show that in scenario 1 there are two sensors that have deviated from their expected readings in the normal operating mode.

In scheme 1 coherent fault trees for the deviated sensor readings are combined as inputs to an AND gate to form the scenario’s top event structure. This is shown in Figure 4.

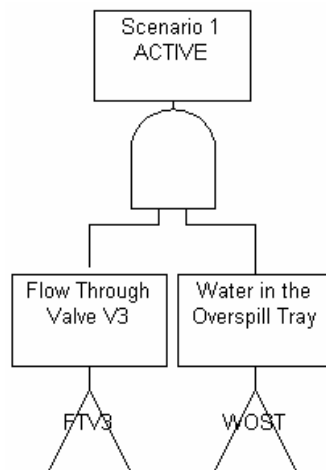


Figure 4: Top event structure for scenario 1 in the ACTIVE operating mode containing just deviated sensor readings

A qualitative analysis of this fault tree will produce its minimal cut sets. These are a list of all the combinations of component failures that would cause the sensor readings obtained. Analysis of the fault tree structure given in Figure 4 produces in total 102 potential causes. From closer inspection though only 15 of these will produce the top event structure. The remaining 87 are incorrect, the causes of this are discussed below.

The 15 valid potential causes are listed below in Table 5.

Number	Potential Causes	Number	Potential Causes
1)	TL.S2FH	9)	S1FL.TL.P4B
2)	TL.C2FH	10)	C1FL.TL.P3B
3)	TL.V3FO	11)	C1FL.TL.P3F
4)	V1FO.TL.P3B	12)	C1FL.TL.P4B
5)	V1FO.TL.P3F	13)	V1FO.V2FC.TL
6)	V1FO.TL.P4B	14)	S1FL.V2FC.TL
7)	S1FL.TL.P3B	15)	C1FL.V2FC.TL
8)	S1FL.TL.P3F		

Table 5: Potential causes of scenario 1 in the ACTIVE operating mode using scheme 1

The coherent fault trees for the sensor readings ‘flow through valve V3’ and ‘water in the overspill tray’ are both valid when analysed individually. However, one of the reasons for obtaining ‘water in the overspill tray’ is that no flow through valve V3 occurs, contradicting the sensor reading ‘flow through valve V3’. With no NOT logic present in the fault trees to indicate the component working states conflicting potential causes are not removed. For example one of the incorrect minimal cut sets is V1FO.P3B.P5B. This minimal cut set would give water in the overspill tray, however pipe P5 is blocked so water cannot get through V3, contradicting the top event structure. Therefore coherent fault trees do not contain enough information about system behaviour, implying scheme 1 is not a very reliable method.

As well as the problem of invalid potential causes the number of valid potential causes of scenario 1 listed above is quite large. Therefore it would be difficult to determine which one could be the actual cause. Scheme 1 does not take into account the sensor readings from the observation points in the scenario that are correct for the chosen operating mode. Only the first 3 potential causes actually produce scenario 1 when the sensor readings from the observation points at valves V1 and V2 are taken into account. The other potential causes are all valid for the sensor reading flow through V1, but would result in no flow through valve V2. This issue is addressed in scheme 2 where information from the sensors that are operating correctly is used to evaluate which parts of the system must function.

### 3.2.2 Scheme 2

Coherent fault trees for sensor readings are also used in scheme 2. In this case though a consistency check is introduced into the model by including all sensor readings obtained at any point in the construction of the top event for each scenario. Where sensor readings are as they should be they are included as NOT failed events in the top event structure. Writing them in their equivalent negated form now includes sensor readings not considered in scheme 1 that are true to the normal chosen

operating mode. For example, if the sensor reading for valve V1 was ‘flow’ then this would be equivalent to saying that there is ‘NOT no flow’. This therefore produces a list of working states for flow through valve V1. The fault trees for a given set of sensor readings are constructed by combining both the sensor readings that deviate from that expected, together with the expected sensor readings in their equivalent negated form using an AND gate. Scheme 2 is now demonstrated again assuming the sensor readings of scenario 1 are obtained.

Table 6 below contains the ACTIVE operating mode for the system (scenario 4) and scenario 1 with the sensor readings for valves V1 and V2 written in negated form. Fault trees for these sensor readings can now be obtained and included in the model along with those for the deviated sensor readings in order to check consistency.

Scenario	V1	V2	V3	TRAY
4 ACTIVE	F	F	NF	NW
1	NOT NF	NOT NF	<b>F</b>	<b>W</b>

Table 6: ACTIVE operating mode and scenario 1

The coherent fault trees for the deviated sensor readings are combined with the consistency checks as inputs to an AND gate, as shown in Figure 5.

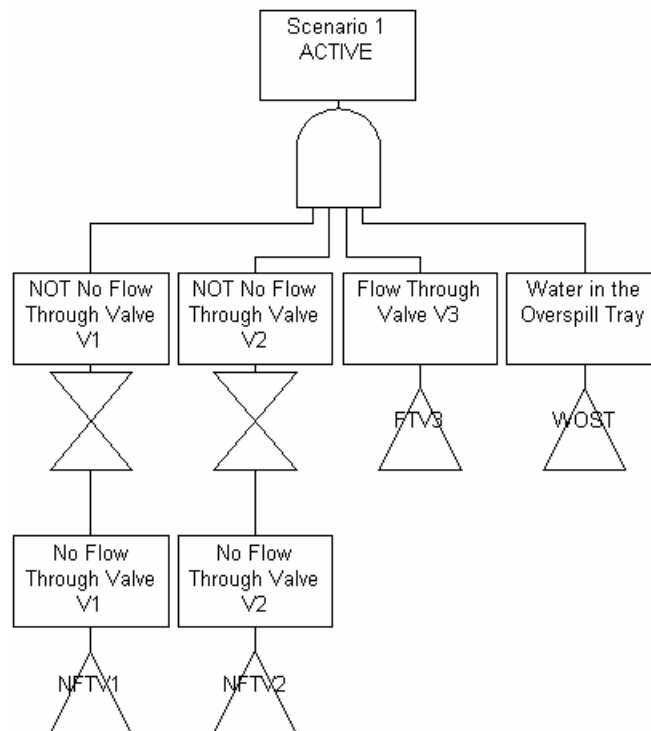


Figure 5: Top event structure for scenario 1 in the ACTIVE operating mode containing both deviated sensor readings and consistency checks

A qualitative analysis of this fault tree, which is non-coherent in structure, produces prime implicants. Prime implicants are combinations of component states (working and failed), that produce the sensor readings obtained. As only failed components

need to be considered, the working states are then removed from the prime implicants to give the coherent approximation.

The cause of NOT no flow through valve V1 is:

$$\overline{P1B.P1F.P2B.S1FH.C1FH.V1FC.V2FC.NWMS}$$

The cause of NOT no flow through valve V2 is:

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

(In the list of component states given above the bar over the event code means NOT that event).

These combined (ANDed) together with the potential causes listed in Table 5 from scheme 1 shows that scheme 2 produces 3 potential causes of scenario 1 in the ACTIVE operating mode, these are listed in Table 7.

Number	Potential Causes	Coherent Approximation
1)	$\overline{NWMS.TR.P2B.P1F.P1B.V2FC.TL.P4B.P3B.P3F.V1FC.C1FH.S1FH.S2FH}$	TL.S2FH
2)	$\overline{NWMS.TR.P2B.P1F.P1B.V2FC.TL.P4B.P3B.P3F.V1FC.C1FH.S1FH.C2FH}$	TL.C2FH
3)	$\overline{NWMS.TR.P2B.P1F.P1B.V2FC.TL.P4B.P3B.P3F.V1FC.C1FH.S1FH.V3FO}$	TL.V3FO

Table 7: Potential causes of scenario 1 in the ACTIVE operating mode using scheme 2

The potential causes listed above include the working components. Removing these produces a list of the potential causes containing only failed components, as shown in Table 7 in the Coherent Approximation column. These are the correct causes of scenario 1. This shows that for this particular scenario scheme 2 is a better model at pinpointing the actual cause compared to scheme 1. However, scheme 1 showed that coherent fault trees do not contain enough information about system behaviour to obtain a reliable list of potential causes. Scheme 2 uses the same fault trees, so invalid results from combining the deviated sensor readings may still occur. NOT logic is introduced into the fault trees for each sensor reading in scheme 3 in order to indicate which components are working.

### 3.2.3 Scheme 3

In scheme 3 system behaviour is modelled using non-coherent fault trees. As in scheme 1 the causes of the sensor readings are given in a fault tree structure obtained using only the sensor readings that deviated from the normal system operating mode. Therefore it does not include a consistency check. To illustrate scheme 3 the top event structure (assuming, as before, the sensor readings represented by scenario 1 have been obtained), can be constructed in the same way as for scheme 1 (see Figure 4), but using non-coherent fault trees for the deviated sensor readings.

Analysis of the faults, as in scheme 2, produces prime implicants. 15 potential causes of scenario 1 are produced by scheme 3. After removing the working state events these are the same as those obtained by scheme 1 which are valid for the deviated sensor readings (see Table 5). Scheme 3 therefore shows that introducing NOT logic into the fault trees indicates which components are working for each deviated sensor reading and removes any conflicting potential causes.

There is again quite a large number of alternatives and it would be difficult to determine which one could be the actual cause. In the same way as for scheme 1, scheme 3 does not take into account the sensor readings from the observation points in the scenario that are correct for the chosen operating mode. Some of the potential causes listed in the example may be invalid. This scheme has failed to differentiate between the potential causes and so scheme 4, which contains the additional complexity of non-coherent sensor deviations with consistency checks, may provide a better analysis option.

### 3.2.4 Scheme 4

Scheme 4 modelled system behaviour using non-coherent fault trees for sensor deviations. The top event structure for scenario 1 using scheme 4 is shown in Figure 5. Analysis of the faults, as in schemes 2 and 3 produces prime implicants. Removing the working component states gives the coherent approximation. There are 3 potential causes of scenario 1, listed below in Table 8.

Number	Potential Causes
1)	TL.S2FH
2)	TL.C2FH
3)	TL.V3FO

Table 8: Potential causes of scenario 1 in the ACTIVE operating mode using scheme 4

The potential causes of scenario 1 in Table 8 are the same as those obtained using scheme 2 after removing the working components (see Table 7). This shows that schemes 2 and 4 are better models than 1 and 3 at pinpointing the actual cause of scenario 1. Compared to the result obtained from scheme 3 the model in scheme 4 strengthens the belief that not enough information is being obtained by only considering sensors that have deviated from their expected readings.

The performance of the four schemes has been investigated to evaluate their potential causes of all 16 scenarios. Schemes 1 and 2 have both produced potential causes that are invalid to the top event structure. For example, for scenario 5 schemes 1 and 2 produce 132 and 111 potential causes respectively for their top event structures. Only 24 of these results for the two schemes are actually valid even though the fault trees for the sensor readings are correct when analysed individually. For certain scenarios scheme 2 has also removed potential causes that are valid to the top event structure.

This implies that there is not enough information in the coherent fault trees to explain the system behaviour. Scheme 4 produces the best performance for both the ACTIVE and the DORMANT operating modes.

### 3.2.5 Scheme 5

The two operating modes for the system are ACTIVE when the operator opens valve V2, or DORMANT when V2 is closed. Scheme 5 involves changing the operating state of the system in order to examine the system behaviour. This can be used alongside any of the other four schemes previously discussed. Schemes 2 and 4 have both narrowed the potential causes of scenario 1 in the ACTIVE operating mode down to 3 possibilities, (see Table 8). These all indicate that the tank is definitely leaking (TL) to create water in the overspill tray, but it is unclear as to whether sensor S2 has failed high (S2FH), controller C2 has failed high (C2FH) or valve V3 has failed open (V3FO) in order to cause flow through valve V3.

If the real cause were, for example, TL.S2FH – the first potential cause listed in Table 8, one way to narrow down the possibilities would be to switch the operating mode of the system, in this case from ACTIVE to DORMANT. Switching operating modes could also highlight hidden potential failures for valve V2. For example, if V2 failed open whilst the system was working as in scenario 1 in the ACTIVE operating mode then this failure would be masked. Switching to the DORMANT operating mode would reveal this failure which would otherwise have remained undetected. For scenarios where V2 has already failed in a given operating mode switching modes could conceal this failure. For example, if there is flow through valves V1 and V2, no flow through V3, and water in the overspill tray in the DORMANT operating mode and the system is switched to the ACTIVE mode then the failure at V2 would become hidden.

In the DORMANT mode a different set of sensor readings resulting from TL.S2FH would occur, which will be one of two possibilities depending on whether or not V2 was found to be working correctly. The sensors in the ACTIVE operating mode indicate that there is flow through all three valves and water in the overspill tray. When the operating mode is switched to DORMANT there should be no flow through V2. If this is the case then the new set of deviated sensor readings will be no flow through V2, flow through V1 and V3 and water in the overspill tray. If there is still flow through valve V2 after switching operating modes then this indicates that all four sensors have deviated from their normal operating modes. Treating this set of sensor values by comparison to the expected set – scenario 16 in Table 1 and applying any of the schemes 1 to 4 will produce a set of possible causes. The true cause of the fault will be in both lists. Valve V2 failing if not apparent before switching modes will be indicated by the addition of this component failure to the potential causes in the list from the mode which the system has been switched to.



## **4. Discussion**

### ***4.1 Effectiveness of the Schemes***

Schemes 1 to 5 were applied to the 16 scenarios listed in Table 1. For each scenario results for the ACTIVE and DORMANT operating modes were obtained. The introduction of a consistency check in schemes 2 and 4 has resulted in the same or a more accurate list of potential causes for each system scenario compared to schemes 1 and 3 respectively. This is because schemes 1 and 3 do not take into consideration the sensor readings that are correct for the given sensor readings. Schemes 1 and 2 have both produced potential causes that are invalid to the top event structure, even though the fault trees for the sensor readings are correct when analysed individually. For certain scenarios scheme 2 has also removed potential causes that are valid to the top event structure. This implies that coherent fault trees do not contain enough information about the system to produce an accurate list of potential causes. Schemes 3 and 4 both produce potential causes that are true for the chosen scenario. Scheme 4 uses non-coherent fault trees and consistency checks, therefore the causes obtained are more accurate than those obtained by schemes 1 to 3. Scheme 5 can be used to increase the accuracy of the list of potential causes obtained from schemes 1 to 4. Therefore scheme 4 used along with scheme 5 is the best model for obtaining potential causes of the 16 scenarios for the water tank system.

### ***4.2 Scalability and Consistency Checks***

The methods presented in this paper for diagnosing faults in systems using FTA to explain the deviations from normal values observed in the sensed variables have been illustrated using a simple example containing only a small number of components. Application to a larger system is required in order to show how effective the methods would be when used on real systems.

The computational expense from a large number of sensors is the main problem. Introducing consistency checks may be prohibitive for real time analysis of industrial systems. Scheme 4 could be heavy on computer processing power, but scheme 3 is not as reliable because it does not check consistency. However, this can be solved by taking the potential causes obtained from just the deviated sensor readings and checking these individually against those from the sensor readings which are true to the operating mode. Any potential causes which lead to a contradiction can then be removed from the list.

### ***4.3 Imperfect Sensors***

The method illustrated in this paper has been applied to an example which does not account for the possibility of there being imperfect sensors in the system. The example used demonstrates the approach taken to fault detection. It could accommodate sensor failures by incorporating these events into the fault trees for each sensor reading and the analysis performed in the same way.

#### ***4.4 Obtaining the Exact Cause of Failure***

Ideally the most efficient scheme to model the behaviour of a system would be one that could determine the exact cause of each scenario. However, in reality there may be more than one possible cause. For example, there are 3 potential causes of scenario 1 in the ACTIVE operating mode (see Table 8). These all indicate that the tank is definitely leaking (TL) to create water in the overspill tray, but it is unclear as to whether sensor S2 has failed high (S2FH), controller C2 has failed high (C2FH) or valve V3 has failed open (V3FO) in order to cause flow through valve V3.

Valve V3 can fail in 3 different ways that would still produce the same outcome for the top event structure. Switching the operating mode in this case does not reduce the number of potential causes, but could highlight if there is a problem at valve V2. Therefore, a probability for each component would have to be introduced in order to deduce which is most likely to have caused the scenario. This addition to the calculation procedure would effectively be providing measures [14] for each minimal cut set.

#### ***4.5 Dynamic Effects***

At present the water tank system is observed in a steady state. Because of this it is not possible, without taking into account transient effects, for some of the 16 scenarios to occur. For example a number of the scenarios containing a 'no flow' sensor reading for valve V1 and flow through at least one of the other two valves produced no potential causes, indicating that they could never happen. However, if dynamics were taken into consideration then depending on the state of the system these scenarios may occur, albeit for a short period of time. Therefore considering the effects of dynamics on the water tank would provide a more realistic model of the system behaviour.

#### ***4.6 Two Phased System***

The water tank system is assumed to be in one of two phases; these being the flow phase through valve V2 when in the ACTIVE operating mode and the no flow phase through V2 in the DORMANT operating mode. However, V2 could be partially open indicating an intermediate flow phase through the valve. Further analysis could therefore consider the behaviour of the system when failure modes are not clear.

### **5. Conclusions**

- Fault tree analysis can be used to identify multiple faults in a systems fault diagnostic capability.
- Non-coherent fault trees to represent the causes of sensor outputs provide more reliable results compared with those obtained using the coherent fault trees.

- The introduction of consistency checks with the information provided by all sensors is required in order to produce an accurate list of all potential causes of the current system state.
- Scheme 4, which uses non-coherent fault trees, and checks for consistency by using information from all the observation points is the most accurate of schemes 1 to 4 for modelling behaviour of the water tank system.
- Scheme 5, which switches the operating mode of the system, can be used along side scheme 4 to produce a more precise list of potential causes should the system function enable mode switching.

## Acknowledgements

The research reported in this paper has been carried out with financial support from BAE SYSTEMS.

## References

1. Zuzek A., Biasizzo A. and Novak F., "Towards a General Test Presentation in the Test Sequencing Problem", *Proceedings of the 2nd International On-Line Testing Workshop, IEEE Computer Society Press, Biarritz, France, 236-237* (1996).
2. Zuzek A., Novak F., Biasizzo A., Savnik I. And Cestnik B., "Sequential Diagnosis Tool for System Maintenance and Repair", *Electrotechnical Review*, **62**:224-231 (1995).
3. Biasizzo A., Zuzek A. and Novak F., "Sequential Diagnosis Tool", *Microprocessors and Microsystems*, **24**:191-197 (2000).
4. Novak F., Zuzek A. and Biasizzo A., "Sequential Diagnosis Tool, A Tutorial", <http://www-csd.ijs.si/applications/SDT/sdt.html>.
5. Biasizzo A., Zuzek A. and Novak F., "Sequential Diagnosis With Asymmetrical Tests", *The Computer Journal*, **41** [3] 163-170 (1998).
6. Pattipati K. R. and Alexandridis M. G., "Application of Heuristic Search and Information Theory to Sequential Fault Diagnosis", *IEEE Transactions on Systems, Man and Cybernetics*, **20** [4] 872-887 (1990).
7. Shakeri M., Raghavan V., Pattipati K. R. and Patterson-Hine A., "Sequential Testing Algorithms for Multiple Fault diagnosis", *IEEE Transactions on Systems Man and Cybernetics - Part A: Systems and Humans*, **30** [1] 1-14 (2000).
8. Rao N. S. V., "Expected-Value Analysis of Two Single Fault Diagnosis Algorithms", *IEEE Transactions on Computers*, **42** [3] 272-280 (1993).
9. Pattipati K.R., "Computationally Efficient Algorithms for Multiple Fault Diagnosis in Large Graph-Based Systems", *IEEE Transactions on SMC: Part A – Systems and Humans*, **33** [1] 73- 85 (2003).
10. Price C., "Computer-Based Diagnostic Systems", Springer-Verlag London Limited, 1999.
11. Paasch R. and G. Mocko, "Incorporating Uncertainty In Diagnostic Analysis Of Mechanical Systems", *Proceedings of the 2002 ASME Design Theory and Methodology Conference, Montreal, QB*, October 2002.

12. Papadopoulos Y., "Model-based system monitoring and diagnosis of failures using statecharts and fault trees", *International Journal of Reliability Engineering and System Safety*, **81**:325-341 (2003).
13. Yangping Z., Bingquan Z. and Dong Xin W., "Application of Genetic Algorithms to Fault Diagnosis in Nuclear Power Plants", *Reliability Engineering and System Safety*, **67**:153-160 (2000).
14. Andrews J.D. and Moss T. R., "Reliability and Risk Assessment", Second Edition, Professional Engineering Publishing Limited, London and Bury St. Edmunds, UK, 2002.
15. Lee W. S., Grosh D. L., Tillman F. A. and Lie C. H., "Fault Tree Analysis Methods and Applications – A Review", *IEEE Transactions on Reliability*, **R-34** [3] 194-203 (1985).
16. Billinton R. and Allan R. N., "Reliability Evaluation of Engineering Systems", Second Edition, Plenum Press New York, 1992.