



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



CC creative commons
COMMONS DEED

Attribution-NonCommercial-NoDerivs 2.5

You are free:

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

Under the following conditions:

BY: **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/>

Analysis of Fault Trees with Secondary Failures

John Andrews and Sarah Dunnett
Department of Systems Engineering
Loughborough University

Summary

The Fault Tree methodology is appropriate when the component level failures (basic events) occur independently. One situation where the conditions of independence are not met occurs when secondary failure events appear in the fault tree structure. Guidelines for fault tree construction, which have been utilised for many years, encourage the inclusion of secondary failures along with primary failures and command faults in the representation of the failure logic. The resulting fault tree is an accurate representation of the logic but may produce inaccurate quantitative results for the probability and frequency of system failure if methodologies are used which rely on independence.

This paper illustrates how inaccurate these quantitative results can be. Alternative approaches are developed by which fault trees of this type of structure can be analysed.

1.0 Introduction

Fault Tree Analysis is now frequently used to assess the adequacy of systems from a reliability or availability viewpoint. The technique was originally developed in the 1960's and guidelines were subsequently produced to describe how the engineering system can be modelled¹. Modelling the system results in a representation of the failure logic which can then be quantified^{2,3}. Model quantification produces: combinations of component level failures which will cause the system failure mode (minimal cut sets), system failure probability, system failure frequency and importance measures.

It is critical that the fault tree construction process is performed accurately. Following this there are many commercial software packages available to carry out the quantification. Rules cannot be determined which govern the construction of the failure logic diagram and guarantee the production of the correct fault tree for all circumstances. However, guidelines which provide a rigorous, systematic approach have been developed and are commonly applied by engineers. One such guideline which can be found in reference 1 is that state-of-component faults can be developed in the fault tree structure by an OR-gate with *primary failure*, *secondary failure* and *command faults* as inputs.

This paper shows that following this process, whilst producing correct failure logic, can lead to situations where the standard means of quantifying the top event probability will be incorrect. The error occurs as the repair of individual component failures in a minimal cut set does not rectify the system state.

Two approaches are described which can be used to overcome this difficulty. The first of these employs a Markov model to analyse the sections of the fault tree where the secondary failures are located. The second method makes use of equations pre-determined from Markov models of basic constructs which occur in sections of the fault tree containing secondary failures. Both of these methods require that the section of the fault tree to which the method is applied is independent of the remainder of the fault tree. In the first method this is the limiting factor governing the size of the fault tree section modelled using Markov.

2.0 Fault Tree Construction for state-of-component faults

A guideline proposed for the fault tree construction process¹ was to classify events to be developed in the fault tree as either *state-of component faults* or *state-of-system faults*. The distinction between the two was based on whether the event being developed could be caused by a single component failure or not. Where the event cannot be caused by a single component failure it is classified as a state-of-system fault and developed by establishing the immediate, necessary and sufficient conditions and usually brings an AND gate into the fault tree. If a single component failure can cause the event it is classed as a state-of-component fault and the fault tree is developed in terms of primary component failures, secondary component failures and command faults as shown in figure 1.

In figure 1 a *primary failure* is defined as a component failure which occurs when the component is operating in its normal expected environment. A *secondary failure* is one where the component is operating outside its intended operating environment (usually due to other failures occurring which cause an increased stress level on the component). The *command fault* traces the fault back into other parts of the system which provide an input to the component and could cause a working component to exhibit the fault being developed. For example, consider a control valve sub-system. To identify causes of the event that no fluid flows at the valve outlet, it is classified as a state-of-component fault and developed as shown in figure 1. The primary component failure is the control valve itself failing closed. A command fault is a failure of the valve control system which causes a functioning valve to close.

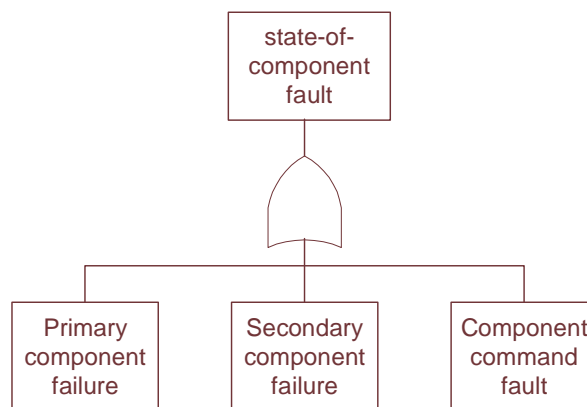


Figure 1 Fault Tree for State-of-component fault

This approach has also been incorporated into texts which cover the fault tree method in detail^{2,3}. It is an effective way of generating a fault tree with the correct failure logic and is in itself non-controversial. The potential problem comes in the later analysis stage where all basic events in the fault tree structure are assumed to occur independently. The construction of the fault tree using this guideline introduces dependencies between the repair of the basic events. A secondary failure causes the failure of another component in the system and so the rectification of the system functionality requires the repair of more components than those which combined to cause the original problem. Failure to adequately account for this, by for example assuming independence between the basic events, can introduce large errors into the numerical procedures used to calculate the top event probability and frequency.

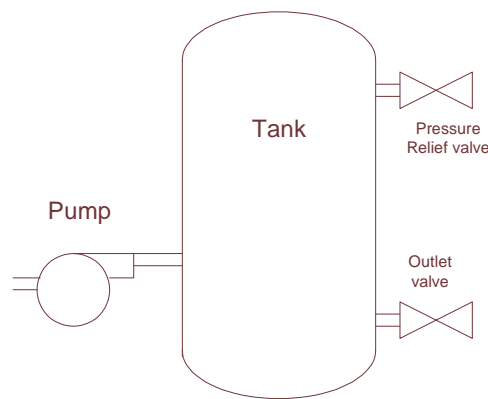


Figure 2 Simple Pressure Tank System

3.0 Pressure Tank Example

As an example, consider the simple part of a pressure tank system illustrated in figure 2. The tank is filled by activating the pump. The contents are used as required by opening and closing the outlet valve. As a safety feature, in the event of overfilling the relief valve will open to keep the pressure within acceptable bounds. It is required to predict the unavailability of the pressure tank due to its rupture. This can be classified as a state-of-component fault since failure of the tank alone can produce this event. The fault tree is then developed accordingly.

The Top Event, “Rupture of the Pressure tank”, is developed as illustrated in figure 1 and resolved into its primary and secondary causes (in this example the tank does not have a command fault). The primary failure event is that the tank fails under normal expected conditions (TANK). The secondary failure event occurs when the tank fails whilst operating outside its normal expected operating conditions and is caused by an overpressure situation. The overpressure which ruptures the tank (assuming overpressure will always have this outcome) is due to the pump control system failure which causes the pump to run for too long AND the safety feature (the pressure relief valve) fails to operate. The event ‘fault in control system fails to stop pump’ is also a state-of-component fault and therefore developed in terms of its primary failure of E and secondary failure X. The simple fault tree for this situation is represented in figure 3. The minimal cut sets for this fault tree are:

1. TANK

- 2. E. PRV
- 3. X. PRV

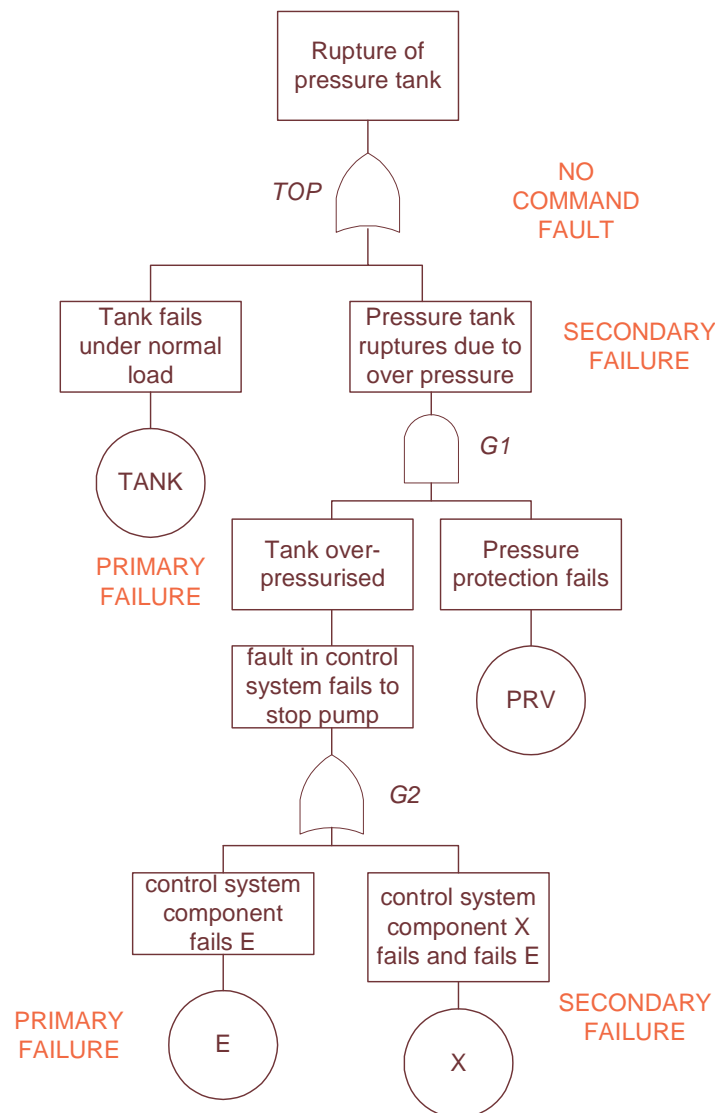


Figure 3 Pressure Tank System Fault Tree

The implication of the qualitative analysis is that the repair any of the events contained in a minimal cut set which causes the top event will result in the system failure mode no longer existing. However when one of the events is a secondary failure this is no longer true. Consider minimal cut set 2. If these two events E and PRV occur together then in addition to these two components being in the failed state, the tank will also fail. This is not an event in the minimal cut set. Considering the minimal cut set alone, if the pump or pressure relief valve are repaired it would under conditions of independence rectify the top event. However, since this failure is a secondary failure combination and results in tank rupture, the tank must also be repaired to rectify the system. Since the tank repair time is likely to be considerably longer than the two elements of the minimal cut set, failure to account for this in the analysis will result in a serious underestimation of the system unavailability.

It should also be noted that the pressure relief valve failure, PRV, is an *enabling event*⁴. ie one which permits another event to cause the top event. It is a failure of a safety device which, since it is normally inactive, will, on its own, have no effect on

the system unless the occurrence of an *initiating event*⁴ puts a demand on it to work. All other events in the fault tree are initiating events whose occurrence, unless mitigated, will cause the top event.

Basic event failure and repair data are given in table 1.

Basic Event Code	Failure rate (per hour)	Mean time to repair (hours)	Inspection interval (hours)	Type of Failure Dormant (enabler) or Revealed (initiator)
TANK	1×10^{-7}	500		R
PRV	5×10^{-4}	25	1975	D
E	2×10^{-3}	24		R
X	2×10^{-3}	24		R

Table 1 Component Failure and Repair Data

A conventional analysis of the fault tree illustrated in figure 2 with basic event data given in table 1 gives:

Top event probability 3.362×10^{-2}
 Top event frequency 1.387×10^{-3} per hour

A summary of the contribution to these results from each minimal cut set is given in table 2. The full calculations are presented in Appendix A.

Minimal Cut Set	Probability q	Frequency w
TANK	4.99975×10^{-5}	9.9995×10^{-8}
E . PRV	0.01693	7.0534×10^{-4}
X . PRV	0.01693	7.0534×10^{-4}

Table 2 Minimal Cut Set calculations to the top event

These results have been obtained assuming independence of the basic events. The correct modelling of a section of the fault tree which features secondary failures would need to be performed with a technique such as Markov methods² which can take into account the repair time dependence.

4.0 Markov Analysis

Prior to performing a Markov Analysis of the system a number of assumptions are required as to how the system will be repaired. These are as follows:

- i) If both X and E fail, PRV will be activated revealing E, if only E is repaired X will cause E to fail again hence both E and X are repaired.
- ii) If E or X fail causing PRV to activate it is assumed that there is no mechanism by which PRV can subsequently fail in an inactive (stuck) mode.

- iii) If PRV fails first and subsequently E (or X) fail then the tank will rupture and hence PRV , E (or X) and T will need to be repaired.
- iv) When the repair of more than one component is to be performed, it is assumed that they will be repaired sequentially and so the repair time for all components will be the sum of their individual repair times.

The values used for these additional repair times are given in table 2.

Repair of components	Mean time to repair (hours)	Repair rate (per hours)
	τ	ν
E+T	524	0.0019
E+T+X	548	0.0018
P+T	525	1.90476×10^{-3}
E+P+T	549	1.8215×10^{-3}
E+X	48	0.0208
E+P+X	73	0.0137
E+P	49	0.02041
E+P+X+T (all)	573	1.7452×10^{-3}

Table 2 Component combinations repair times

The Markov model for the system, which provides a direct alternative means of analysis to the fault tree in figure 3 is shown in figure 4. A list of the different states and the transitions between them is contained in Appendix B.

In this figure λ_i and ν_i are the failure and repair rates for the basic events and so $i=E, T, X$ and P . Where more than one letter appears as the subscript for the repair rate it indicates a list of components whose repair is performed sequentially. The Markov model has been constructed in two phases to model the periodic inspection process carried out on the pressure relief valve, PRV. Phase 1 is a continuous phase which operates from $t=0$ to $t=\theta$, the inspection interval for PRV. During phase 1 a failure of this component on its own will remain unrevealed as indicated by the ‘U’ defining its condition in states 5 and 8. The failure of the relief valve in phase 1 will only be revealed by a demand on it to function (failure of E or X). Transitions between states in this phase are indicated by a solid line in figure 4. Phase 2 is a discrete, instantaneous phase where the inspection takes place and reveals failures of the relief valve and transfers its status to ‘F’ (revealed failure awaiting repair) and enables the component to be repaired as indicated by states 11 and 14. Instantaneous transitions occur every θ hours and are shown as a dotted line in figure 4. Phases 1 and 2 occur cyclically until the mission time is reached.

The resulting Markov model has 15 states. Examination will indicate that a more concise model could have been developed as several states could have been merged into one. For example states 10, 13 and 15 all feature the four components in the failed state. The model has deliberately been constructed in this way to take account of the order in which the failures occur and enable the contribution each minimal cut set makes to the system failure to be identified. Where the occurrence of failures cause the secondary failure of another component, this is indicated by ‘(F)’ in the state definitions. F_2 indicates that this was the second (initiating) failure event.

For this model, states 2, 6, 7, 8, 9, 10, 12, 13, 14 and 15 are all failed states.

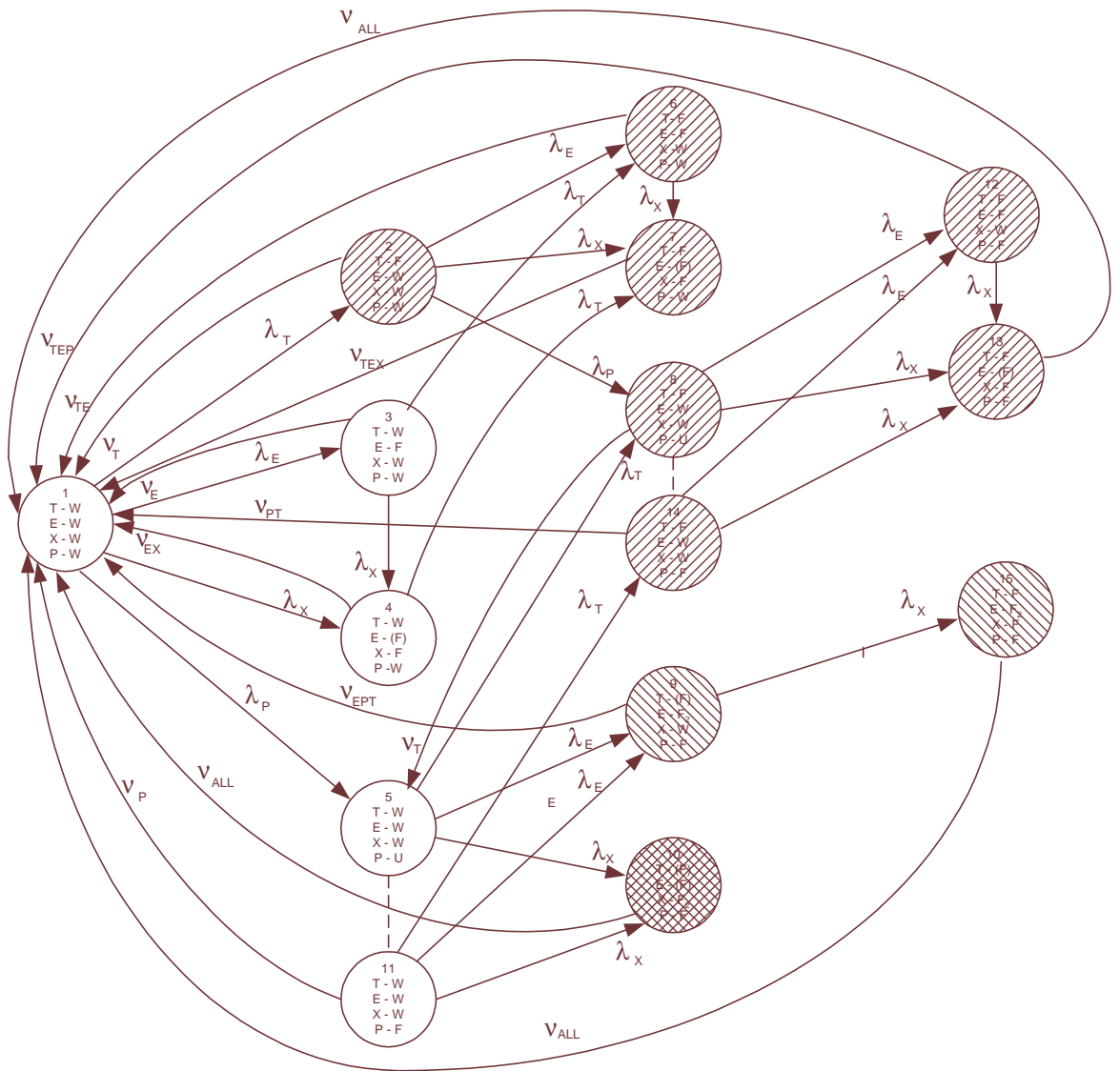


Figure 4 System Markov State Transition Diagram

The state equations for phase 1 are given by:

$$\begin{aligned}
\frac{dq_1}{dt} &= -(\lambda_E + \lambda_X + \lambda_P + \lambda_T)q_1 + v_E q_3 + v_{EX} q_4 + v_P q_{11} + v_T q_2 + \\
&\quad v_{TE} q_6 + v_{TEX} q_7 + v_{TP} q_{14} + v_{TEP} (q_9 + q_{12}) + v_{ALL} (q_{10} + q_{13} + q_{15}) \\
\frac{dq_2}{dt} &= \lambda_T q_1 - (v_T + \lambda_X + \lambda_P + \lambda_E) q_2 \\
\frac{dq_3}{dt} &= \lambda_E q_1 - (v_E + \lambda_X + \lambda_T) q_3 \\
\frac{dq_4}{dt} &= \lambda_X (q_1 + q_3) - (\lambda_T + v_{EX}) q_4 \\
\frac{dq_5}{dt} &= \lambda_P q_1 + v_T q_8 - (\lambda_E + \lambda_X + \lambda_T) q_5 \\
\frac{dq_6}{dt} &= \lambda_E q_2 + \lambda_T q_3 - (v_{TE} + \lambda_X) q_6 \\
\frac{dq_7}{dt} &= \lambda_T q_4 + \lambda_X q_2 + \lambda_X q_6 - v_{TEX} q_7 \\
\frac{dq_8}{dt} &= \lambda_P q_2 + \lambda_T q_5 - (v_T + \lambda_X + \lambda_E) q_8 \\
\frac{dq_9}{dt} &= \lambda_E (q_5 + q_{11}) - (v_{TEP} + \lambda_X) q_9 \\
\frac{dq_{10}}{dt} &= \lambda_X (q_5 + q_{11}) - v_{ALL} q_{10} \\
\frac{dq_{11}}{dt} &= -(v_P + \lambda_T + \lambda_E + \lambda_X) q_{11} \\
\frac{dq_{12}}{dt} &= \lambda_E (q_8 + q_{14}) - (v_{TEP} + \lambda_X) q_{12} \\
\frac{dq_{13}}{dt} &= \lambda_X (q_8 + q_{12} + q_{14}) - v_{ALL} q_{13} \\
\frac{dq_{14}}{dt} &= \lambda_T q_{11} - (v_{TP} + \lambda_E + \lambda_X) q_{14} \\
\frac{dq_{15}}{dt} &= \lambda_X q_9 - v_{ALL} q_{15}
\end{aligned}$$

For phase 2, at $t=n\theta$ the equations are:

$$\begin{aligned}
q_{11} &= q_{11} + q_5 \\
q_5 &= 0 \\
q_{14} &= q_{14} + q_8 \\
q_8 &= 0
\end{aligned}$$

These equations were solved for q_i , $i=1,,15$. The probability of system failure, the probability of the top event featured in the fault tree shown in figure 3, is given by:

$$Q = q_2 + q_6 + q_7 + q_8 + q_9 + q_{10} + q_{12} + q_{13} + q_{14} + q_{15}$$

and the failure intensity by

$$w = \lambda_T(q_1 + q_3 + q_4 + q_5 + q_{11}) + (q_5 + q_{11})(\lambda_E + \lambda_X).$$

Results for the failure probability and failure frequency using the Markov model are contained in table 3 and shown in figure 5 (averages taken over second and subsequent periods). A comparison of the Markov results with those produced by the fault tree method are also shown in table 3. The contributions from each of the three minimal cut sets are included.

Event	States	Markov Results				Fault Tree results	
		Average Q	Max Q	Average w (per hour)	Max w (per hour)	Q	w
System fails	Any min cut sets	0.1652	0.1794	2.95 x 10 ⁻⁴	3.27 x 10 ⁻⁴	0.03362	0.01387
Min Cut Set 1	2,6,7,8,12,13,14	4.425 x 10 ⁻⁵	4.47 x 10 ⁻⁵	8.35 x 10 ⁻⁸	8.54 x 10 ⁻⁸	4.99975 x 10 ⁻⁵	9.9995 x 10 ⁻⁸
Min Cut Set 2	9,15	0.081	0.0878	1.4725 x 10 ⁻⁴	1.63 x 10 ⁻⁴	0.01693	7.0534 x 10 ⁻⁴
Min Cut Set 3	10	0.08425	0.0915	1.4725 x 10 ⁻⁴	1.63 x 10 ⁻⁴	0.01693	7.0534 x 10 ⁻⁴

Table 3 Comparison of Markov and Fault tree results

It can be seen from the result in table 3 that there is a very large error in the fault tree results.

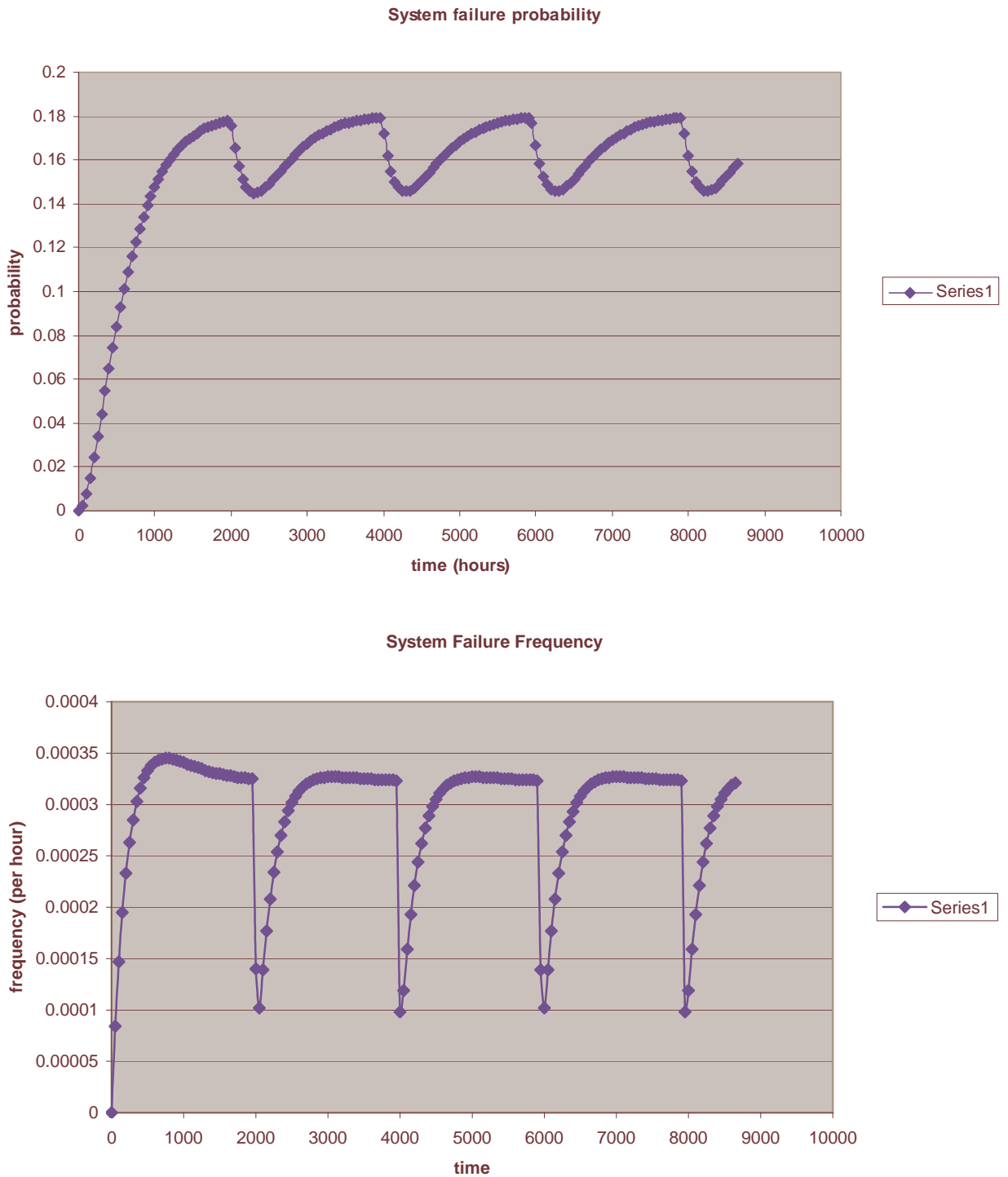


Figure 5 System failure probability and frequency from Markov model

5.0 Quantification approaches for Secondary Failures

It is clear from the example given above that there can be a significant error if fault trees which contain secondary failures are quantified by traditional techniques which assume independence. The size of the error will vary depending on the fault tree

structure, the failure and repair data used, and the number of secondary failure events it contains. Therefore no general conclusions can be made as to when the error experienced will become significant. In the example given above the fault tree gave a optimistic value for the system failure probability and a pessimistic failure frequency. The Markov method can be used to produce an accurate assessment for any system featuring secondary failures. However, the production of a Markov model for an entire system can require the solution of a large number of equations. The size of the Markov diagram can explode exponentially with the number of basic events and is inefficient for moderate to large sized systems. Therefore the generation of a large Markov model for the entire system does not provide an efficient solution to the problem. Two alternative approaches have been investigated:

- i. Fault Tree Modularization Methods
- ii. Analysis of Basic Fault Tree Structures

Each of these is described in the sections that follow.

5.1 Fault Tree Modularization Methods

In many cases it is only a small section of the system fault tree which features the secondary failures. In these circumstances it is possible to analyse that section of the fault tree alone by the more computationally intensive Markov methods. The section of the fault tree analysed in this way is then replaced by a super-event in the fault tree structure. The failure probability and failure intensity of the super-event is derived from the Markov Analysis. Analysing a section of the fault tree using the Markov approach and substituting the results back into the larger scale analysis has become a standard way of evaluating fault trees where dependencies such as those associated with standby and sequential systems are concerned^{5,6}. To perform this type of analysis in an efficient way requires that the section of the fault tree extracted for Markov analysis is itself independent of the remainder of the fault tree and has to be selected accordingly.

5.2 Basic Fault Tree Structures

A common feature of fault trees which contain secondary failures is that the secondary failure section of the fault tree is itself independent of the remainder of the fault tree. In this situation it permits the use of analytical results obtained from the assessment of Markov models representing typical features of the fault tree. This removes the need to perform the numerical analysis of a larger Markov model and makes the analysis faster to compute. Consider the example fault tree shown in figure 3, it has two basic fault tree constructs present (as do many of this type of fault tree). These are illustrated in figure 6. The fault tree section to which the method is applied must therefore be restructured in terms of the constructs illustrated. This is achieved by systematically defining complex events by pairs of basic events or other complex events occurring as inputs to the same gate type. Failure and repair parameters to the complex events are then derived as follows:

For complex events, CAND, which replace two input events X and Y into an AND gate, the failure probability, q_{CAND} , and failure frequency, w_{CAND} are given by:

$$q_{CAND} = q_X \cdot q_Y$$

$$w_{CAND} = q_X \cdot w_Y + q_Y \cdot w_X$$

For complex events, COR, which replace two input events X and Y into an OR gate, the failure probability, q_{COR} , and failure frequency, w_{COR} are given by:

$$q_{COR} = 1 - (1 - q_X) \cdot (1 - q_Y)$$

$$w_{COR} = (1 - q_X) \cdot w_Y + (1 - q_Y) \cdot w_X$$

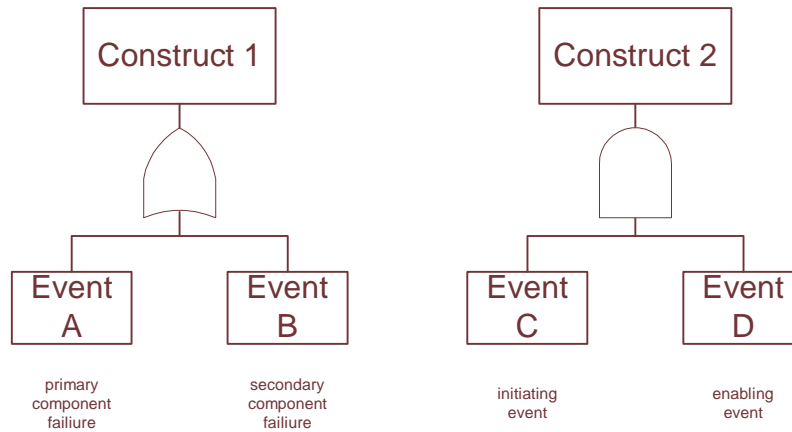


Figure 6 Basic Fault Tree constructs

Construct type 1 has two input events A and B where event A represents the primary failure of a component and event B the secondary failure. This construct type appears twice in the fault tree shown in figure 3 at the highest gate (TOP) and the lowest gate (G2). Construct type 2 appears where there is some protection (safety feature) which can mitigate the occurrence of a potential problem. The event which causes the potential problem, the initiating event, is Event C in this construct. Event D is an enabling event, failed safety system, which permits the initiating event to cause the problem. Construct 2 is illustrated in gate G1 of the fault tree. These two basic constructs have been analysed separately using Markov models to produce equations which can be used whenever they occur in a fault tree structure. Steady-state conditions are assumed to prevail at the end of each inspection cycle as seems justified by looking at the graphs shown in figure 5.

By determining probability, q_C , and failure intensity, w_C , for a construct, these data can then be used in a super-event which replaces the construct in the fault tree. By performing this type of substitution in a bottom-up manner for each gate in the fault tree a complete assessment can be accomplished for the secondary failure section.

Construct 1.

The Markov model to represent the primary and secondary failures is illustrated in Figure 7.

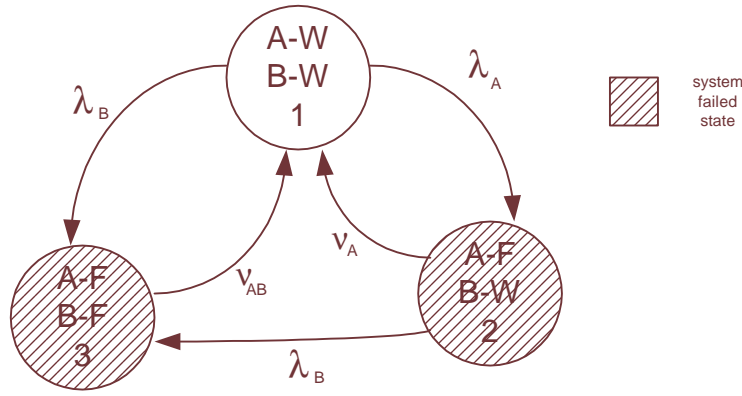


Figure 7 Markov Diagram of construct 1

λ_A , λ_B are the failure rates for components A (primary failure) and B (secondary failure) respectively, v_A , v_{AB} are the repair rates for A and A and B together respectively. States 2 and 3 are failed states for this construct. The state equations for this module are:

$$\begin{aligned}\frac{dq_1}{dt} &= -q_1(\lambda_A + \lambda_B) + q_2v_A + q_3v_{AB} \\ \frac{dq_2}{dt} &= q_1\lambda_A - q_2(v_A + \lambda_B) \\ \frac{dq_3}{dt} &= q_1\lambda_B + q_2\lambda_B - q_3v_{AB}\end{aligned}$$

where q_i is the probability of the system being in state i at time t .

In the steady-state situation solved to give q_1 , q_2 , q_3

$$\begin{aligned}q_1 &= \frac{(v_A + \lambda_B)v_{AB}}{(v_{AB} + \lambda_B)(\lambda_A + v_A + \lambda_B)} \\ q_2 &= \frac{v_{AB}\lambda_A}{(v_{AB} + \lambda_B)(\lambda_A + v_A + \lambda_B)} \\ q_3 &= \frac{\lambda_B}{v_{AB} + \lambda_B}\end{aligned}$$

The probability of the output event of this construct occurring is determined by:

$$Q_{C1} = q_2 + q_3 \quad (1)$$

The failure intensity for construct 1, w_{C1} , can be determined by:

$$w_{C1} = q_1(\lambda_A + \lambda_B) \quad (2)$$

Construct 2

This construct allows for the provision of safety features in the secondary failure section of the fault tree. The failure of the safety feature is represented by event D in figure 6. If this has happened and then the initiating event C occurs it is unable to provide mitigation for event C and the failure propagates up the fault tree structure. If the initiating event occurs prior to the enabling event the failure propagation will not result. The Markov state transition diagram for this is illustrated in figure 8.

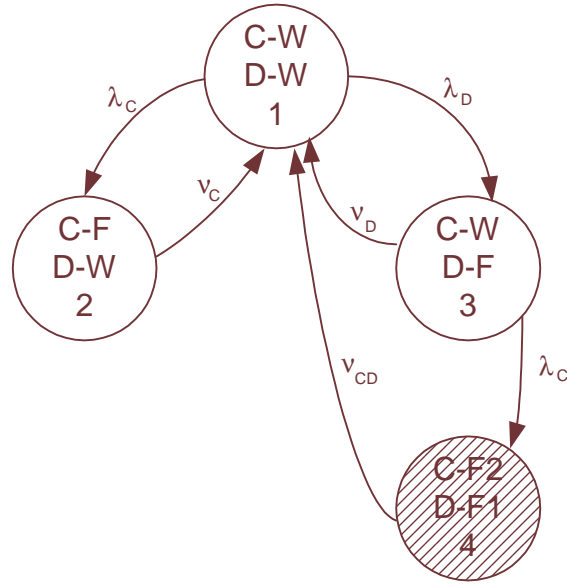


Figure 8 Markov Diagram for construct 2

ν_{CD} is the repair rate for events C and D together. In this case the only failed state is state 5. The assumption is made that once both the initiating event and the enabling event have occurred repair will be instigated which will return both to the working state.

$$\frac{dq_1}{dt} = -(\lambda_C + \lambda_D)q_1 + \nu_C q_2 + \nu_D q_3 + \nu_{CD} q_4$$

$$\frac{dq_2}{dt} = \lambda_C q_1 - \nu_C q_2$$

$$\frac{dq_3}{dt} = \lambda_D q_1 - (\nu_D + \lambda_C)q_3$$

$$\frac{dq_4}{dt} = \lambda_C q_3 - \nu_{CD} q_4$$

These equations were solved for q_i , $i=1 - 5$ for the steady-state situation to give:

$$q_1 = \frac{(\nu_D + \lambda_C)\nu_{CD}\nu_C}{\alpha}$$

$$q_2 = \frac{\nu_{CD}(\nu_D + \lambda_C)\lambda_C}{\alpha}$$

$$q_3 = \frac{\lambda_D\nu_{CD}\nu_C}{\alpha}$$

$$q_4 = \frac{\lambda_C\lambda_D\nu_C}{\alpha}$$

where:

$$\alpha = \nu_D\nu_{CD}\nu_C + \nu_{CD}\lambda_C\nu_D + \nu_{CD}\lambda_C^2 + \lambda_D\nu_{CD}\nu_C + \lambda_D\lambda_C\nu_C + \lambda_C\nu_{CD}\nu_C$$

The probability , Q_{C2} , of the super-event replacing Construct 2 is given by:

$$Q_{C2}=q_4 \quad (3)$$

The failure intensity of Construct 2, w_{C2} , is given by:

$$w_{C2}=\lambda_C q_3 \quad (4)$$

5.3 Application to the simple pressure tank system

The above constructs are applied progressively up through the pressure tank failure fault tree structure illustrated in figure 3. The lowest gate in the structure is an OR gate with events E and X as inputs. Event X is a secondary failure and hence this gate type satisfies the requirements of construct 1. Using the failure and repair data for these components and equations (1) and (2) gives:

$$Q_{G2}= 0.12767$$

$$w_{G2}= 3.4893 \times 10^{-3}.$$

Using:

$$\lambda = \frac{w}{(1-Q)} \quad (5)$$

$$\nu = \frac{w}{Q}$$

gives $\lambda_{G2}=3.999 \times 10^{-3}$, $\nu_{G2}=2.733 \times 10^{-2}$.

The next gate to consider in the fault tree, gate G1, has as inputs gate G2 and the pressure relief valve failure (PRV). The pressure relief valve is a safety feature and thus an enabling event. Gate 2 is an initiating event which causes the over-pressurisation and therefore puts a demand in the safety feature to respond. This satisfies the requirements of construct 2 where G2 is the initiating event and PRV is the enabler. Applying equations (3) and (4) to gate G1 yields:

$$Q_{G1}=0.019399$$

$$w_{G1}=3.14975 \times 10^{-4}$$

$$[\text{note } \nu_{G2P} = \frac{1}{\tau_{G2} + \tau_p} = \frac{1}{36.59 + 25} = 0.0162]$$

therefore $\lambda_{G1}=3.212 \times 10^{-4}$, $\nu_{G1}=1.6237 \times 10^{-2}$.

Finally the top gate in the fault tree is considered. Its structure is the same as gate G2 and therefore the equations developed for construct 1 can be used again, with the parameters for the tank used for event A and those for gate G1 used for event B. This gives a prediction for the top event, system failure of:

$$Q_{\text{top}}=0.15285$$

$$w_{\text{top}}=2.7219 \times 10^{-4} \text{ per hour}$$

This compares well with the average Markov values of $Q_{\text{top}}=0.1652$, $w_{\text{top}}=2.95 \times 10^{-4}$ per hour.

6.0 Conclusions

1. If secondary failures are to be modelled in a system failure probability assessment accurate results will not be obtained using the fault tree analysis

method. This is despite suggestions in the literature that the development of the fault tree should be conducted in a way which considers this type of failure for each state-of-component fault which occurs.

2. Markov methods will produce accurate results and can be applied to the smallest independent section of the fault tree which contains the secondary failure events. Results of this are obtained numerically and substituted back into the system fault tree to replace the section of the fault tree analysed.
3. If the part of the fault tree which contains the secondary failures contains events which are independent of the rest of the fault tree then constructs can be applied in a bottom-up manner to efficiently get an estimate of the top event failure parameters.

References

1. Hassl D.F., Roberts N.H., Vesley W.E. and Goldberg F.F., 'Fault Tree Handbook', US Nuclear Regulatory Commission, NUREG-0492, 1981.
2. Andrews J.D. and Moss T.R., 'Reliability and Risk Assessment', Professional Engineering Publications Ltd, 2002.
3. Henley E.J. and Kumamoto H., 'Reliability Engineering and Risk Assessment', Prentice-Hall, 1981.
4. Lambert H.E and Dunlison C., 'Interval Reliability for Initiating and Enabling events', , IEEE Transactions on Reliability, Vol 32, June 1983, pp 150-163.
5. Meshkat L., Dugan J.B. and Andrews J.D., 'Dependability Analysis of Systems with On-demand and Active Failure Modes Using Dynamic Fault Trees', IEEE Transactions on Reliability, Vol 51, No 2, June02, pp 240-252.
6. Meshkat L., Dugan J.B. and Andrews J.D., 'Maintenance Modelling for Computer Based Systems', IMechE Proceedings Part E, Journal of Process Mechanical Engineering, Vol 215, No E3, 2001, pp221-233.

Appendix A – Fault tree quantification results

The contributions (assuming basic event independence and that steady-state conditions prevail) are given by:

For minimal cut set 1 {T}

$$q_{C1} = \frac{\lambda_T}{\lambda_T + \nu_T} = 4.99975 \times 10^{-5}$$

$$w_{c1} = \lambda_T(1 - q_{c1}) = 9.9995 \times 10^{-8} / \text{hour}$$

For minimal cut set 2 {E, PRV}

For the initiating event:

$$q_E = \frac{\lambda_E}{\lambda_E + \nu_E} = 0.0458$$

$$w_E = \lambda_E(1 - q_E) = 1.9084 \times 10^{-3} / \text{hour}$$

For the enabling event:

$$q_{PRV} = \frac{\lambda_{PRV}\theta - (1 - e^{-\lambda_{PRV}\theta}) + \lambda_{PRV}\tau_{PRV}(1 - e^{-\lambda_{PRV}\theta})}{\lambda_{PRV}\theta + \lambda_{PRV}\tau_{PRV}(1 - e^{-\lambda_{PRV}\theta})} = 0.3696$$

This gives:

$$w_{C2} = w_E q_{PRV} = 7.0534 \times 10^{-4} / \text{hour}$$

$$q_{C2} = q_E q_{PRV} = 0.01693$$

For minimal cut set 3 {X, PRV}

For the initiating event:

$$q_X = \frac{\lambda_X}{\lambda_X + \nu_X} = 0.0458$$

$$w_X = \lambda_X(1 - q_X) = 1.9084 \times 10^{-3} / \text{hour}$$

For the enabling event:

$$q_{PRV} = \frac{\lambda_{PRV}\theta - (1 - e^{-\lambda_{PRV}\theta}) + \lambda_{PRV}\tau_{PRV}(1 - e^{-\lambda_{PRV}\theta})}{\lambda_{PRV}\theta + \lambda_{PRV}\tau_{PRV}(1 - e^{-\lambda_{PRV}\theta})} = 0.3696$$

This gives:

$$w_{C3} = w_X q_{PRV} = 7.0534 \times 10^{-4} / \text{hour}$$

$$q_{C3} = q_X q_{PRV} = 0.01693$$

The system parameters are then obtained from:

$$Q_{sys} = 1 - \prod_{i=1}^n (1 - q_{ci}) = 0.03362$$

$$w_{sys} = \sum_{i=1}^n \left(w_{ci} \prod_{\substack{j=1 \\ j \neq i}}^n (1 - q_{cj}) \right) = 1.3868 \times 10^{-3} / \text{hour}$$

Appendix B

Markov model states

State No.	Component T	Component E	Component X	Component PRV	System
1	W	W	W	W	W
2	F	W	W	W	F (C1)
3	W	F	W	W	W
4	W	(F)	F	W	W
5	W	W	W	U	W
6	F	F	W	W	F (C1)
7	F	(F)	F	W	F (C1)
8	F	W	W	U	F (C1)
9	(F)	F ₂	W	U	F (C2)
10	(F)	(F)	F ₂	U	F (C3)
11	W	W	W	F	W
12	F	F	W	U	F (C1)
13	F	(F)	F	U	F (C1)
14	F	W	W	F	F (C1)
15	(F)	F ₂	F	F ₁	F (C2)

Transitions between states for full fault tree.

State i	State j		State i	State j	
1	2	λ_T	7	1	v_{TEX}
1	3	λ_E	8	5	v_T
1	4	λ_X	8	12	λ_E
1	5	λ_P	8	13	λ_X
2	1	v_T	9	1	v_{EPT}
2	6	λ_E	9	15	λ_X
2	7	λ_X	10	1	v_{ALL}
2	8	λ_P	11	1	v_P
3	1	v_E	11	9	λ_E
3	4	λ_X	11	10	λ_X
3	6	λ_T	11	14	λ_T
4	1	v_{EX}	12	1	v_{TEP}
4	7	λ_T	12	13	λ_X
5	8	λ_T	13	1	v_{ALL}
5	9	λ_E	14	1	v_{TP}
5	10	v_X	14	12	λ_E
6	1	v_{TE}	14	13	λ_X
6	7	λ_X	15	1	v_{ALL}