

Fault Tree and Markov Analysis Applied to Various Design Complexities

J.D. Andrews, PhD; Department of Mathematical Sciences, Loughborough University, England
Clifton A. Ericson II, The Boeing Company; Seattle, Washington

Abstract

Fault Tree Analysis (FTA) and Markov Analysis (MA) are two analysis techniques that have been around for many years. Both are well proven techniques, each having its advantages and disadvantages. One of the common questions in industry is whether or not both modeling tools are adequate for a design application, or if one tool has an advantage over the other. This is a significant question, because there are many different types of system design complexities, many different types of undesired system states, and many different size system problems, all of which have a consequential impact on the analysis tool utilized.

There are some very important questions that need to be answered in order for the systems analyst to perform a proper evaluation of a system design. For instance, can both tools handle all of the various design complexities? Do both tools provide accurate results for all cases? Do both tools handle repair? Do both tools handle catastrophic and non-catastrophic events? Are approximation results good enough? Can both tools handle large systems?

This paper describes the various types of common system design complexities and compares the FTA and MA approaches for each of these system configurations. The results of these two approaches are then compared, providing some relative conclusions.

Design Complexities

System analysis is a challenging process due to the various types of design types, complexities and factors involved in today's modern complex systems. As system design advances along with advances in technology, the corresponding complexity also advances and increases. With increased complexity, it also becomes more difficult to analyze and predict system behavior and especially system misbehavior.

Table 1 contains a list of the major factors that cause complexity in both systems design and systems analysis. Each one of these factors presents a different issue, both for designers and systems analysts trying to evaluate the design. The complexity issue of Design Type is the primary issue being addressed by this paper. Table 2 provides the relevant supporting information for the probability calculations.

Tables 3 through 8 contain an analysis of each design type. These tables describe the design type and operation, and present both the Fault Tree and Markov model for each. These tables describe the major design type categories and their corresponding philosophy of operation. Each design type is individual and unique, and each requires its own concomitant model. The analysis problem often cascades as different design types are combined together in an overall system. Some design types are easier to model than others.

Table 1 – Complexity Factors

	Complexity Factors	Consideration
1	Design Type	Affects model correctness
2	Design Size	Affects modeling tools and model correctness
3	Exposure Time	Affects model and probability calculation
4	Latency	Affects model and probability calculation
5	Repair	Affects probability calculation
6	Dependency	Affects model correctness
7	Logic Loops	Affects model correctness
8	Accuracy	Affects model capabilities (approximations)
9	Undesired States	States range from hazardous to unavailable
10	System Criticality	States range from safety critical to safe
11	Standby Redundancy	Hot, warm, cold - model correctness
12	Coverage	Affects model and probability calculation

Table 2 – Supporting Data	
Notation	Event Probability Calculation
λ_A = failure rate of A (also $\hat{\lambda}A$ in some places) ν_A = repair rate of A A_W = working \overline{A}_F = failed $\overline{\lambda}_A$ = failure rate of A in standby	$q_A = \frac{\lambda_A}{\lambda_A + \nu_A} (1 - e^{-(\lambda_A + \nu_A)T})$ $q_B = \frac{\lambda_B}{\lambda_B + \nu_B} (1 - e^{-(\lambda_B + \nu_B)T})$ <p>no repair when $\nu_A = \nu_B = 0$</p>


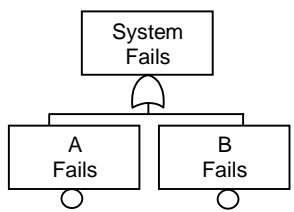
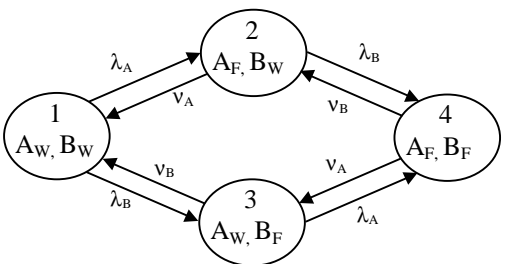
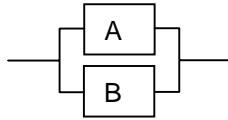
Table 3 – Series System	
<p>Design Type Description A system is comprised of two components A and B in series. System success requires that both must operate successfully at the same time. System failure occurs if either one or both fail.</p>	
	
<p>FTA Solution</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 30%;">  </div> <div style="width: 65%;"> $q_A = (\lambda_A / (\lambda_A + \nu_A)) (1 - e^{-(\lambda_A + \nu_A)T})$ $q_B = (\lambda_B / (\lambda_B + \nu_B)) (1 - e^{-(\lambda_B + \nu_B)T})$ $Q_{sys} = q_A + q_B - q_A q_B$ <p>with repair:</p> $P = P_A + P_B - P_A P_B$ $= (\lambda_A / (\lambda_A + \nu_A)) (1 - e^{-(\lambda_A + \nu_A)T}) + (\lambda_B / (\lambda_B + \nu_B)) (1 - e^{-(\lambda_B + \nu_B)T}) - [(\lambda_A / (\lambda_A + \nu_A)) (1 - e^{-(\lambda_A + \nu_A)T})][(\lambda_B / (\lambda_B + \nu_B)) (1 - e^{-(\lambda_B + \nu_B)T})]$ <p>without repair ($\nu_A = \nu_B = 0$):</p> $P = (1 - e^{-\lambda_A T}) + (1 - e^{-\lambda_B T}) - (1 - e^{-\lambda_A T})(1 - e^{-\lambda_B T})$ $= 1 - e^{-(\lambda_A + \lambda_B)T}$ </div> </div>	
<p>Markov Solution</p> <div style="display: flex; justify-content: space-between;"> <div style="width: 45%;">  </div> <div style="width: 50%;"> $\begin{aligned} dP_1 / dt &= -(\lambda_A + \lambda_B)P_1 + \nu_A P_2 + \nu_B P_3 \\ dP_2 / dt &= \lambda_A P_1 - (\lambda_A + \nu_A)P_2 + \nu_B P_4 \\ dP_3 / dt &= \lambda_B P_1 - (\lambda_A + \nu_A)P_3 + \nu_A P_4 \\ dP_4 / dt &= \lambda_B P_2 + \lambda_A P_3 - (\nu_A + \nu_B)P_4 \end{aligned}$ $P = P_2 + P_3 + P_4$ $P = 1 - e^{-(\lambda_A + \lambda_B)T} \quad \text{no repair case } (\nu_A = \nu_B = 0)$ </div> </div>	
<p>Conclusion Both methods provide the same results (i.e., the equations are identical). Derivation of the repair calculations for the MA are not shown here due to the complexity and space required.</p>	

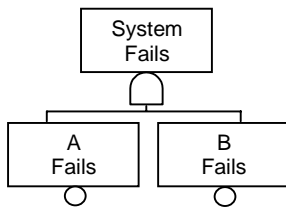
Table 4– Parallel System

Design Type Description

A system is comprised of two components A and B in parallel. System success requires that either one (or both) must operate successfully. System failure occurs only if both fail are failed at the same time.



FTA Solution



$$q_A = (\lambda_A / (\lambda_A + v_A)) (1 - e^{-(\lambda_A + v_A) T})$$

$$q_B = (\lambda_B / (\lambda_B + v_B)) (1 - e^{-(\lambda_B + v_B) T})$$

$$Q_{sys} = q_A \cdot q_B$$

with repair:

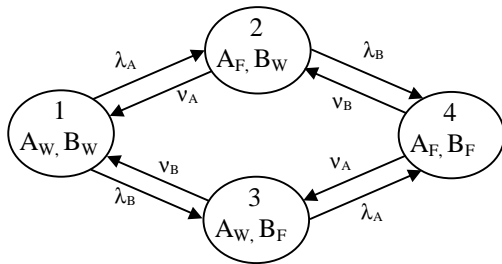
$$P = P_A \cdot P_B$$

$$= (\lambda_A / (\lambda_A + v_A)) (1 - e^{-(\lambda_A + v_A) T}) \cdot (\lambda_B / (\lambda_B + v_B)) (1 - e^{-(\lambda_B + v_B) T})$$

without repair ($v_A = v_B = 0$):

$$P = (1 - e^{-\lambda_A T}) (1 - e^{-\lambda_B T})$$

Markov Solution



$$\frac{dP_1}{dt} = -(\lambda_A + \lambda_B)P_1 + v_A P_2 + v_B P_3$$

$$\frac{dP_2}{dt} = \lambda_A P_1 - (\lambda_A + v_A)P_2 + v_B P_4$$

$$\frac{dP_3}{dt} = \lambda_B P_1 - (\lambda_A + v_A)P_3 + v_A P_4$$

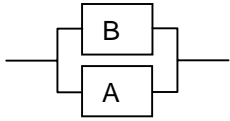
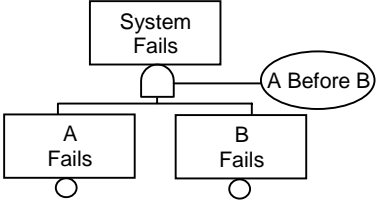
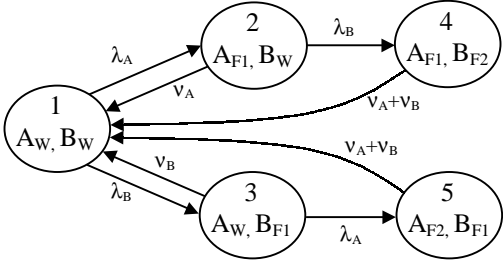
$$\frac{dP_4}{dt} = \lambda_B P_2 + \lambda_A P_3 - (v_A + v_B)P_4$$

$$P = P_4$$

$$P = (1 - e^{-\lambda_A T})(1 - e^{-\lambda_B T}) \quad \text{no repair case } (v_A = v_B = 0)$$

Conclusion

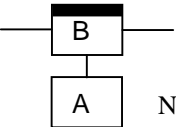
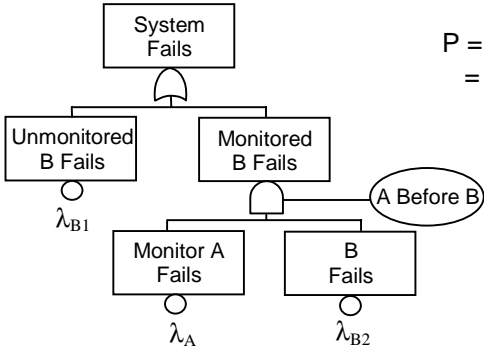
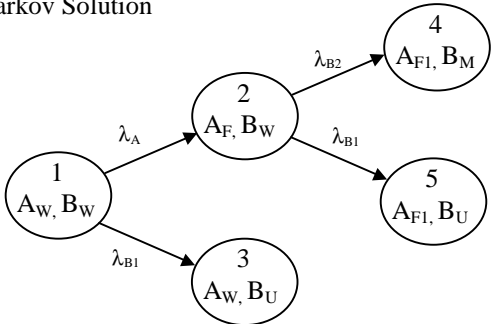
Both methods provide the same results (i.e., the equations are identical). Derivation of the repair calculations for the MA are not shown here due to the complexity and space required.

Table 5 – Sequence Parallel System	
<p>Design Type Description A system is comprised of two components A and B. System success requires that both must operate successfully at the same time. System failure occurs if both fail, but only if A fails before B.</p> 	
<p>FTA Solution</p> 	<p>$P = (P_A \cdot P_B) / N!$ General equation, where N is number of inputs and $P_A \cong P_B$.</p> <p>$P = (P_A \cdot P_B) / 2$ $= ((1 - e^{-\lambda_A T})(1 - e^{-\lambda_B T})) / 2$ no repair case ($v_A = v_B = 0$)</p>
<p>Markov Solution</p> 	<p>$P = \frac{\lambda_A(1 - e^{-\lambda_B T}) - \lambda_B(e^{-\lambda_B T} - e^{-(\lambda_A + \lambda_B) T})}{\lambda_A + \lambda_B}$</p> <p>no repair case ($v_A = v_B = 0$)</p>
<p>Conclusion Resulting equations are different, making the FT equation an approximation. Refer to Figure 1 for a graphical comparison of results.</p>	

Time (Hrs)	FTA	MA
1	5.00000 E-14	5.00000 E-14
10	4.99947 E-12	4.99998 E-12
100	4.99973 E-10	4.99980 E-10
1,000	4.99725 E-8	4.99800 E-8
10,000	4.97260 E-6	4.98006 E-6
100,000	4.73442 E-4	4.80542 E-4
1,000,000	3.00771 E-2	3.45145 E-2
10,000,000	3.16046 E-1	5.41213 E-1
100,000,000	4.99977 E-1	9.09046 E-1
1,000,000,000	4.99977 E-1	9.09091 E-1

Figure 1 – Comparison of Results for Sequence Parallel System
 (Where $\lambda_A = 1.0 \times 10^{-6}$ and $\lambda_B = 1.0 \times 10^{-7}$)

Table 6 – Full Monitor System	
<p>Design Type Description</p> <p>A system is comprised of two components, Monitor A and component B. Monitor A monitors the operation of B. If it detects any failure in B it takes corrective action. System success requires that B must operate successfully. System failure occurs if component B fails, which can only happen if Monitor A fails to detect a problem with B, and B subsequently fails. If A works it always corrects any failure in B or provides a warning</p>	
<p>FTA Solution</p> <div style="display: flex; align-items: center;"> <div style="flex: 1;"> </div> <div style="flex: 2;"> $P = (P_A \cdot P_B) / 2$ $= ((1 - e^{-\lambda_A T})(1 - e^{-\lambda_B T})) / 2 \quad \text{no repair case } (v_A = v_B = 0)$ <p>[Same as Table 5]</p> </div> </div>	
<p>Markov Solution</p> <p>[Same as Table 5]</p>	$P = \frac{\lambda_A(1 - e^{-\lambda_B T}) - \lambda_B(e^{-\lambda_B T} - e^{-(\lambda_A + \lambda_B)T})}{\lambda_A + \lambda_B}$ <p>no repair case ($v_A = v_B = 0$)</p>
<p>Conclusion</p> <p>Resulting equations are different, making the FT equation an approximation. Refer to Figure 1 for a graphical comparison of results.</p>	

Table 7 – Partial Monitor System (Coverage)	
<p>Design Type Description A system is comprised of two components, Monitor A and component B. Monitor A monitors the operation of B, however, it is only designed to monitor 80% of B. If it detects any failure in B it takes corrective action. System success requires that B must operate successfully. System failure occurs if component B fails, which can only happen if Monitor A fails to detect a problem with the monitored portion of B, or if the unmonitored portion of B fails.</p> <div style="text-align: center;">  <p>Note – the darker portion of B is not monitored (λ_{B1}).</p> </div>	
<p>FTA Solution</p> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;">  </div> <div style="flex: 1;"> $P = P_{B1} + (P_A P_{B2}) / 2$ $= (1 - e^{-\lambda_{B1}T}) + ((1 - e^{-\lambda_{AT}})(1 - e^{-\lambda_{B2}T}) / 2 - [(1 - e^{-\lambda_{B1}T})(1 - e^{-\lambda_{AT}})(1 - e^{-\lambda_{B2}T})]) / 2$ <p style="text-align: right;">no repair case ($v_A = v_B = 0$)</p> </div> </div>	
<p>Markov Solution</p> <div style="display: flex; justify-content: space-between;"> <div style="flex: 1;">  </div> <div style="flex: 1;"> $\begin{aligned} dP_1 / dt &= -(\lambda_A + \lambda_{B1})P_1 \\ dP_2 / dt &= -(\lambda_{B1} + \lambda_{B2})P_2 + \lambda_A P_1 \\ dP_3 / dt &= \lambda_{B1}P_1 \\ dP_4 / dt &= \lambda_{B2}P_2 \\ dP_5 / dt &= \lambda_{B1}P_2 \end{aligned}$ </div> </div>	
<p>Conclusion This is a coverage type problem, whereby the monitor does not provide complete coverage of the circuit being monitored. Figure 2 shows very close results between FTA and MA for this design complexity.</p>	

Time (Hrs)	FTA	MA
1	0.000010	0.000010
10	0.000100	0.000100
100	0.001000	0.001000
1,000	0.010263	0.010314
10,000	0.099664	0.103269
100,000	0.649625	0.666795
1,000,000	0.999969	0.999982

Figure 2 – Comparison of Results for Sequence Parallel System
 (Where $\lambda_A = 1.0 \times 10^{-6}$ and $\lambda_{B1} = 1.0 \times 10^{-5}$ and $\lambda_{B2} = 1.0 \times 10^{-3}$)

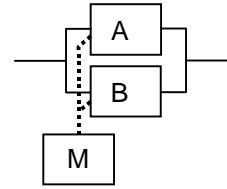
Table 8 – Standby System

Design Type Description

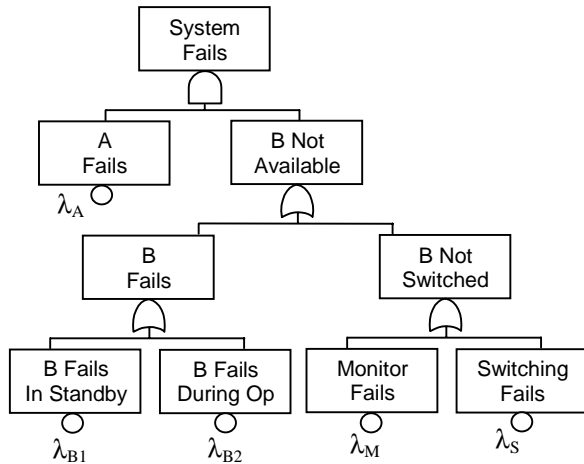
A system is comprised of two main components A and B, and a monitor. System operation starts with component A in operation and B on standby. If A fails, then B is switched on-line and it takes over. System success requires that either A or B operate successfully. System failure occurs if both components A and B fail. Note that B can be failed if switching fails to occur.

There are three classes of Standby systems:

- Hot Standby - powered during standby (uses operational λ_O)
- Warm Standby - partially powered during standby ($\lambda_w < \lambda_O$)
- Cold Standby - un-powered during standby ($\lambda_c = 0$)

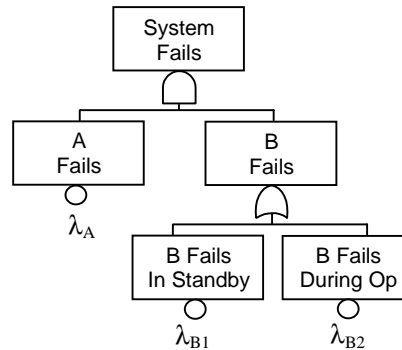


FTA Solution



$$P = P_A \cdot (P_{B1} + P_{B2} + P_M + P_S)$$

$$P = P_A P_{B1} + P_A P_{B2} + P_A P_M + P_A P_S$$

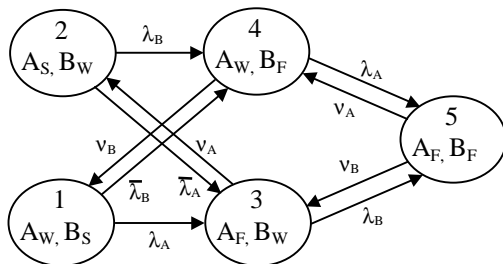


Simplified model assuming monitor is perfectly reliable.

$$P = P_A \cdot (P_{B1} + P_{B2})$$

$$P = (1 - e^{-\lambda_A T}) \cdot \left[\frac{(1 - e^{-\lambda_{B1} T}) + (1 - e^{-\lambda_{B2} T})}{(1 - e^{-\lambda_{B1} T}) + (1 - e^{-\lambda_{B2} T})} \right]$$

Markov Solution (warm standby)



Simplified model assuming monitor is perfectly reliable.

$$\begin{aligned} dP_1 / dt &= -(\lambda_A + \bar{\lambda}_B)P_1 + v_B P_4 \\ dP_2 / dt &= -(\bar{\lambda}_A + \lambda_B)P_2 + v_A P_3 \\ dP_3 / dt &= \lambda_A P_1 + \bar{\lambda}_A P_2 - (\lambda_B + v_A)P_3 + v_B P_4 \\ dP_4 / dt &= \bar{\lambda}_B P_1 + \lambda_B P_2 - (\lambda_A + v_B)P_4 + v_A P_5 \\ dP_5 / dt &= \lambda_B P_3 + \lambda_A P_4 - (v_A + v_B)P_5 \end{aligned}$$

(numerical solution required)

Conclusion

The Markov model is much more difficult when the Monitor is not 100% reliable, therefore it is modeled assuming it works perfectly for this example, but the Fault Tree is able to model the real life situation where the Monitor is fallible. Quite often the exposure time is different for operating, hot, warm and cold standby modes. When in Hot Standby $\bar{\lambda}_A = \lambda_A$ and the Fault Tree and Markov solutions are identical. When in Cold Standby $\bar{\lambda}_A = 0$. For Warm Standby and Cold Standby the Fault Tree solutions are approximations.

Example FT of System Model

The following is a larger system design that combines together several of the basic design types. This system represents a hypothetical aircraft electrical power system. The aircraft has two jet engines, each of which powers two electrical generators via bleed air from the engines. A minimum of two generators are required for aircraft electrical power. The system starts with generators G1 and G2 operating.

Should the monitors detect loss of electrical power, the computer turns on generator G3 and then G4 if necessary. A minimum of two of the three monitors is required for successful operation. Each generator also has internal fault monitoring data which it sends back to the computer, so that the computer can turn on the necessary backup generators.

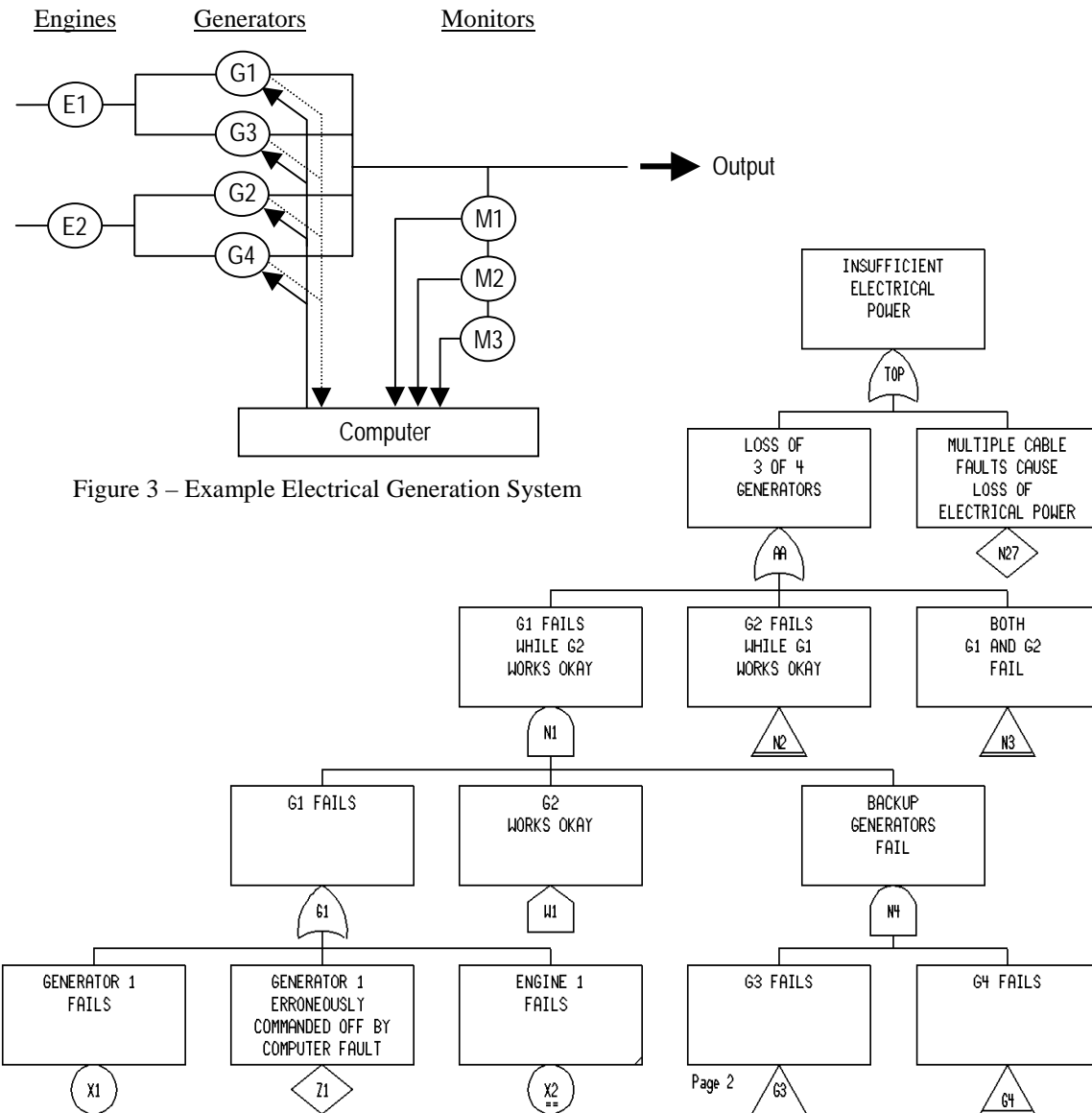


Figure 3 – Example Electrical Generation System

Figure 4a – Electrical Generation System Fault Tree

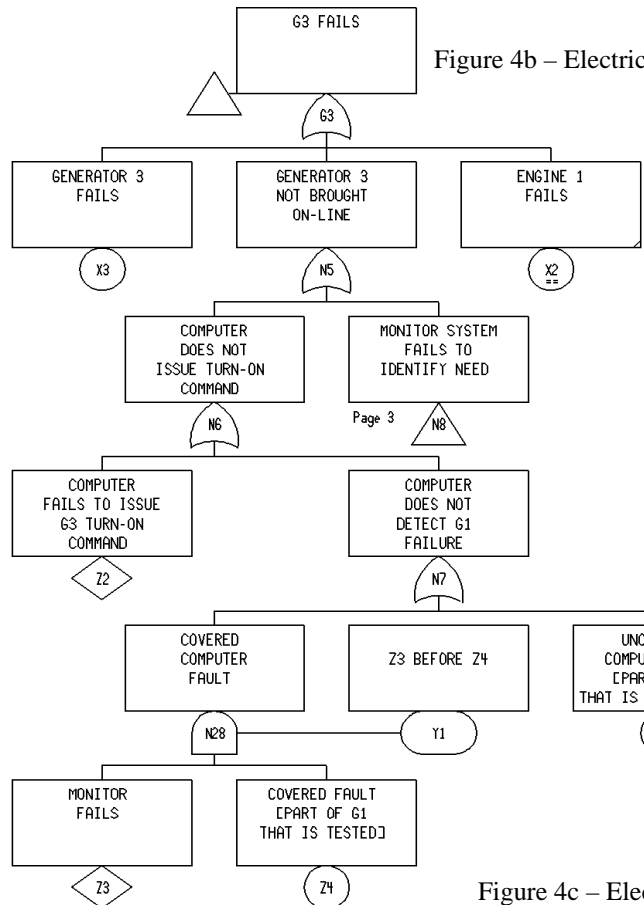


Figure 4b – Electrical Generation System Fault Tree

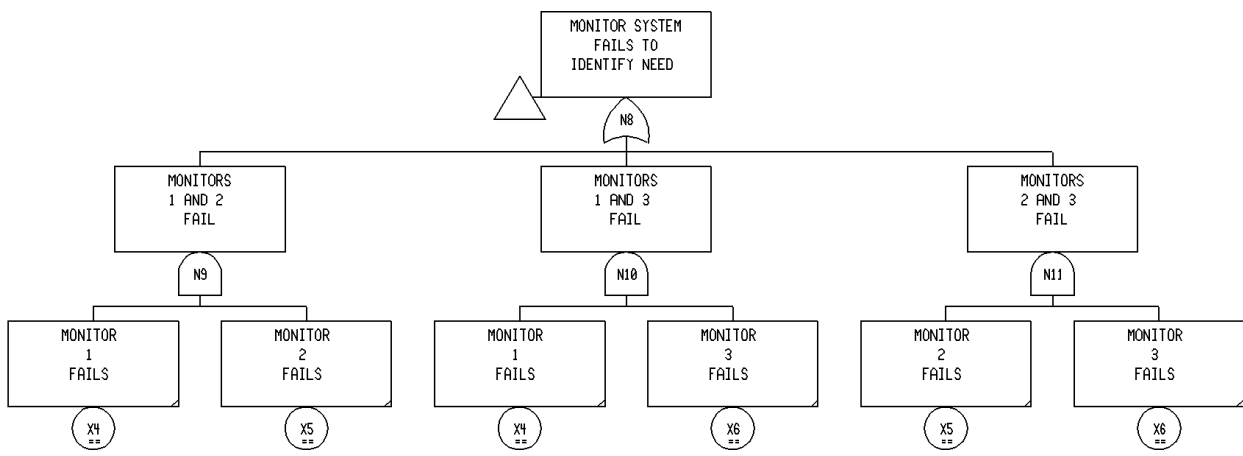


Figure 4c – Electrical Generation System Fault Tree

This example system has 10 components, most of which can be considered as 2-state (Working, Failed). The Generators G3 and G4 are 3-state (Working, Standby, Failed). This system results in a total of 2,304 states. This is without even considering other failure possibilities, such as cables, connectors, etc. Creating a Markov diagram for this many states is very difficult and very error prone, and a Markov solution could only be obtained if numerical methods were used. This problem is very easily modeled and solved using the Fault Tree, and although the

results are an approximation, the results are valuable enough because when small probabilities are involved, pin point precision becomes relative and meaningless. An answer of 1.3×10^{-9} is just as meaningful as 1.21153×10^{-9} .

Notice that Warm Standby and Cold Standby can be modeled without using special gates. This is because these modes are not dependencies, they are merely modes with different failure rates with different exposure times.

Viewpoints

One of the primary considerations in performing FTA or MA is the criticality nature of the problem. That is, is the Undesired Event a safety critical problem with catastrophic results, or is it merely a question of unavailability without any significant safety consequence.

The Reliability viewpoint :

- Non-catastrophic model
- Evaluation for system not being available when needed (Unavailability)
- Repair is often possible even when top Undesired Event (UE) occurs
- Example: Undesired State = Reactor Emergency Core Cooling System Not Available

The Safety viewpoint :

- Generally a catastrophic model
- Once the top UE occurs repair is not possible, a point of no return has been exceeded
- Example: Undesired State = Inadvertent Weapon Release

These two viewpoints have a significant impact on the analysis importance. Repair of the last failure is not possible for the catastrophic safety case. The catastrophic problem requires that all potential root causes be included in the model. FT's can model catastrophic Undesired States much more accurately than MA because FT's can easily include such events as wire shorts, EMI, RF interference, human error, etc. MA does not easily model all types of fault conditions, and when it does the model becomes intractable, or incorrect if events are left out in order to simplify the model

FTA and MA each have unique characteristics and attributes, in addition to common attributes. The following summarizes the particular attributes of each.

Markov Attributes :

- Handles event dependencies
- Handles repair
- The transition rates may not always be constant as assumed
- Assuming the systems history is not important may not always be valid
- Can only model the undesired state of being "Failed"

- Is not a root cause analysis tool
- Can easily overlook and omit causes
- Can only handle small models (about 5 states) unless numerical methods are used
- MA cannot model secondary failure causes, whereas FTA can
- Models are not documented and can be confusing when large
- Large models are difficult to validate

FTA Attributes :

- FTA is a root cause analysis tool
- Models fault combinations and relationships
- Can model more undesired states than just "Failed" state
- Can approximate event dependencies
- Can account for repair
- Can handle very, very large models
- FTA has a structured process making it difficult to leave root causes out
- Easy to modify as design changes
- Easy to validate
- Excellent for documentation
- Produces Importance measures which identify critical items

Conclusion:

The industry argument for many years has been that MA produces more accurate results than FTA, and that only MA can handle certain design complexities. The goal of this paper has been to demonstrate that this is really not the case. FTA produces the same accuracy as MA for many design complexities, for other more difficult design complexities FTA produces results very close to MA. In addition, it has been shown that it is much easier to model large and complex systems using FTA. MA breaks down in modeling large systems, and in generating models that can be easily understood. Quite often the MA models must take short cuts in order to make the model tractable, which in turn results in nothing more than an approximation.

In most real life applications the goal of trying to obtain six digit accuracy in a probability number has little relevance, when the real value involves determining the exponent magnitude. Therefore, FTA provides a better tradeoff for modeling large complex systems with little difficulty, while providing exact or very close approximations for probability estimates, with visual models that are easy to understand. FTA approximations are more than adequate, since

only one or two decimal accuracy is all that is really relevant with complex system designs.

Figures 4a, 4b and 4c contain fault trees for the example system shown in Figure 3. These fault trees demonstrate how easily FTA can model a very complex system. Attempting to model this same system by MA became too complex and intractable.

The following general conclusions have been derived from this study:

- FT's provide more versatility
 - handles larger systems
 - models any Undesired Event
 - model root causes
 - very good approximations
 - provides Importance Measures
 - models environmental effects
 - models secondary faults (shorts, EMI)
 - models human error

- models software error
- models all design complexities

- The FT model itself can identify weak system links even without resorting to probabilities.
- Quite often the MA is really only an approximation, because many contributing fault events are ignored or left out in order to simplify the model.
- When working with small numbers, approximations are generally good enough, thereby making FTA satisfactory.
- When the numbers are large, the results between MA and FTA are still very comparable, as shown in Figures 1 and 2.

Table 9 summarizes these conclusions.

Table 9 – Conclusions Summary

Consideration	FTA	MA
1) Models Undesired Events	X	Partially
2) Models Probability	X	X
3) Models Unavailability	X	X
4) Series System	X	X
5) Parallel System	X	X
6) Sequence Parallel System	Approx	X
7) Full Monitor System	Approx	X
8) Partial Monitor System	Approx	X
9) Standby Redundancy System	Approx	Difficult
10) Repair	X	X
11) Latency	X	X
12) Dependency	Approx	X
13) Large models	X	No
14) Coverage	X	X
15) Easy to follow model	X	No
16) Easy to document process	X	No

Biographies

Dr John Andrews
Department of Mathematical Sciences
Loughborough University
Loughborough, Leicestershire
LE11 3TU, England

tele: +44 (0)1509 222862
fax: +44 (0)1509 223969
email: J.D.Andrews@lboro.ac.uk

Dr Andrews is a Senior Lecturer in the Department of Mathematical Sciences at Loughborough University. He joined this department in 1989 having previously gained nine years industrial experience at British Gas and two years lecturing experience at the University of Central England.

His current research interests concern the assessment of the safety and risks of potentially hazardous industrial systems. This research has been heavily supported by funding from industry. Recent grants have been secured from Mobil North Sea Ltd, Daimler Chrysler and Rolls Royce Aero Engines.

Clifton A. Ericson II
The Boeing Company
Renton, WA 98058 USA

phone 253-657-5245
fax 253-657-2585
email clifton.a.ericson@boeing.com
cliftonericson@cs.com

Mr. Ericson works in System Safety Engineering on Boeing AWACS programs. He has 34 years experience in system safety and software design with the Boeing Company. He has been involved in all aspects of fault tree development since 1965, including analysis, computation, multi-phase simulation, plotting, documentation, training and programming. He has performed Fault Tree Analysis on Minuteman, SRAM, ALCM, Apollo, Morgantown Personal Rapid Transit, B-1, AWACS and 737/757/767 systems. He is the developer of the MPTREE, SAF and FTAB fault tree computer programs. In 1975 he helped start the software safety discipline, and has written papers on software safety and taught software safety at the University of Washington. Mr. Ericson holds a BSEE from the University of Washington and an MBA from Seattle University. He is currently Executive Vice President of the System Safety Society, and is on the technical review committee for the Journal of System Safety and the Journal of Process Mechanical Engineering (UK).