Fault Tree Analysis Approach in Reliability Assessment of Power System

Mohammad Sadegh Javadi[1], Azim Nobakht[2], Ali Meskarbashee[3]

[1]Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Fars,Iran

[2,3]Department of Electrical Engineering, Mahshahr Branch, Islamic Azad University, Mahshahr, Iran

Abstract— **This paper surveys on a reliability technique which is called Fault Tree Analysis (FTA). FTA is a top-down approach to failure analysis, starting with a potential undesirable event (accident) called a TOP event, and then determining all the ways it can happen. The analysis proceeds by determining how the TOP event can be caused by individual or combined lower level failures or events. In power system analysis this approach could maintain the static analysis of the system. The causes of the TOP event are "connected" through logic gates and modeling of the corresponding system. In this paper main features and application of this technique are discussed.**

Keywords— Fault Tree Analysis, Reliability Analysis, Boolean Algebra and Failure Analysis

I. INTRODUCTION

Fault tree analysis (FTA) is the most commonly used technique for causal analysis in risk and reliability studies [1]. Fault tree analysis is a failure analysis in which an undesired state of a system is analyzed using Boolean logic to combine a series of lower-level events [2]. This analysis method is mainly used in the field of safety engineering to quantitatively determine the probability of a safety hazard. Fault Tree Analysis (FTA) was originally developed in 1962 at Bell Laboratories by H.A Watson, under a U.S Air Force Ballistics Systems Division contract to evaluate the Minuteman I Intercontinental Ballistic Missile (ICBM) Launch Control System [3]. The use of fault trees has since gained wide-spread support and is often used as a failure analysis tool by reliability experts [4]. Following the first published use of FTA in the 1962 Minuteman I Launch Control Safety Study, Boeing and AVCO expanded use of FTA to the entire Minuteman II system in 1963-1964. FTA received extensive coverage at a 1965 System Safety Symposium in Seattle sponsored by Boeing and the University of Washington. Boeing began using FTA for civil aircraft design around 1966. In 1970, the U.S. Federal Aviation Administration (FAA) published a change to 14 CFR 25.1309 airworthiness regulations for transport aircraft in the Federal Register at 35 FR 5665 (1970-04-08). This change adopted failure probability criteria for aircraft systems and equipment and led to widespread use of FTA in civil aviation [5].

Within the nuclear power industry, the U.S. Nuclear Regulatory Commission began using probabilistic risk assessment (PRA) methods including FTA in 1975, and significantly expanded PRA research following the 1979 incident at Three Mile Island.[10] This eventually led to the 1981 publication of the NRC Fault Tree Handbook NUREG–0492, and mandatory use of PRA under the NRC's regulatory authority [6].

Fault Tree Analysis (FTA) attempts to model and analyze failure processes of engineering and biological systems. FTA is basically composed of logic diagrams that display the state of the system and is constructed using graphical design techniques. Originally, engineers were responsible for the development of Fault Tree Analysis, as a deep knowledge of the system under analysis is required [7].

Often, FTA is defined as another part, or technique, of reliability engineering. Although both model the same major aspect, they have arisen from two different perspectives. Reliability engineering was, for the most part, developed by mathematicians, while FTA, as stated above, was developed by engineers [8].

Fault Tree Analysis usually involves events from hardware wear out, material failure or malfunctions or combinations of deterministic contributions to the event stemming from assigning a hardware/system failure rate to branches or cut sets. Typically failure rates are carefully derived from substantiated historical data such as mean time between failure of the components, unit, subsystem or function. Predictor data may be assigned. Assigning a software failure rate is elusive and not possible. Since software is a vital contributor and inclusive of the system operation it is assumed the software will function normally as intended [9-10]. There is no such thing as a software fault tree unless considered in the system context. Software is an instruction set to the hardware or overall system for correct operation. Since basic

Manuscript received August 25, 2011. This work was supported in part by the Islamic Azad University, Science and Research Branch, Fars, Iran.

M. S. Javadi is with the Department of Electrical Engineering, Islamic Azad University, Science and Research Branch, Fars, Iran; e-mail: msjavadi@gmail.com).

A. nobakht is with the Department of Electrical Engineering, Islamic Azad University, Mahshahr Branch, Mahshahr, Iran; e-mail: f.nowbakht@gmail.com).

A. Meskarbashee is with the Department of Electrical Engineering, Islamic Azad University, Mahshahr Branch, Mahshahr, Iran; e-mail: Meskarbashee@gmail.com).

software events do not fail in the physical sense, attempting to predict manifestation of software faults or coding errors with any reliability or accuracy is impossible, unless assumptions are made [11]. Predicting and assigning human error rates is not the primary intent of a fault tree analysis, but may be attempted to gain some knowledge of what happens with improper human input or intervention at the wrong time [12].

FTA can be used as a valuable design tool, can identify potential accidents, and can eliminate costly design changes. It can also be used as a diagnostic tool, predicting the most likely system failure in a system breakdown. FTA is used in safety engineering and in all major fields of engineering [13].

II. THEORETICAL CONSIDERATIONS -ANALYTICAL TREES

Analytical trees are graphic representations or pictures of a project or event. They use deductive reasoning in that they start with a general top event or output event and develop down through the branches to specific input events that must occur in order for the output to be generated. Analytical trees are called trees because their structure resembles a tree, narrow at the top with a single event symbol and then branching out as the tree is developed [14].

Negative analytical trees or fault trees are excellent troubleshooting tools. They can be used to prevent or identify failures prior to their occurrence, but are more frequently used to analyze accidents or as investigative tools to pinpoint failures. When an accident or failure occurs, the root cause of the negative event can be identified [15].

Each event is analyzed by asking, “How could this happen?” In answering this question, the primary causes and how they interact to produce an undesired event are identified. This logic process continues until all potential causes have been identified.

Throughout this process, a tree diagram is used to record the events as they are identified. Tree branches stop when all events leading to the negative event are complete. Symbols are used to represent various events and describe relationships:

AND gate **-** represents a condition in which all the events shown below the gate (input gate) must be present for the event shown above the gate (output event) to occur. This means the output event will occur only if all of the input events exist simultaneously.

OR gate **-** represents a situation in which any of the events shown below the gate (input gate) will lead to the event shown above the gate (output event). The event will occur if only one or any combination of the input events exists.

There are five types of event symbols:

1. ***Rectangle*** - The rectangle is the main building block for the analytical tree. It represents the negative event and is located at the top of the tree and can be located throughout the tree to indicate other events capable of being broken down further. This is the only symbol that will have a logic gate and input events below it.
2. ***Circle*** – A circle represents a base event in the tree. These are found on the bottom tiers of the tree and require no further development or breakdown. There are no gates or events below the base event.
3. ***Diamond*** – The diamond identifies an undeveloped terminal event. Such an event is one not fully developed because of a lack of information or significance. A fault tree branch can end with a diamond. For example, most projects require personnel, procedures, and hardware. The tree developer may decide to concentrate on the personnel aspect of the procedure and not the hardware or procedural aspects. In this case the developer would use diamonds to show “procedures” and “hardware” as undeveloped terminal events.
4. ***Oval*** **–** An oval symbol represents a special situation that can only happen if certain circumstances occur. This is spelled out in the oval symbol. An example of this might be if switches must be thrown in a specific sequence before an action takes place.
5. ***Triangle*** – The triangle signifies a transfer of a fault tree branch to another location within the tree. Where a triangle connects to the tree with an arrow, everything shown below the connection point transfers to another area of the tree. This area is identified by a corresponding triangle that is connected to the tree with a vertical line. Letters, numbers or figures identify one set of transfer symbols from another. To maintain the simplicity of the analytical tree, the transfer symbol should be used sparingly.

III. FAULT TREE ANALYSIS METHODOLOGY

Events in a fault tree are associated with statistical probabilities. For example, component failures typically occur at some constant failure rate λ (a constant hazard function). In this simplest case, failure probability depends on the rate λ and the exposure time t:

$$\begin{aligned} &P = 1 - Exp(-\lambda t) \\ &P \approx \lambda t \,, \lambda t < 0.1 \end{aligned} \qquad (1)$$

A fault tree is often normalized to a given time interval, such as a flight hour or an average mission time. Event probabilities depend on the relationship of the event hazard function to this interval.

Unlike conventional logic gate diagrams in which inputs and outputs hold the binary values of TRUE (1) or FALSE (0), the gates in a fault tree output probabilities related to the set operations of Boolean logic. The probability of a gate's output event depends on the input event probabilities.

An AND gate represents a combination of independent events. That is, the probability of any input event to an AND gate is unaffected by any other input event to the same gate. In set theoretic terms, this is equivalent to the intersection of the input event sets, and the probability of the AND gate output is given by:

$$P(A \text{ and } B) = P(A \cap B) = P(A)\,P(B)$$

An OR gate, on the other hand, corresponds to set union:

$$P(A \text{ or } B) = P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

Since failure probabilities on fault trees tend to be small (less than .01), $P(A \cap B)$ usually becomes a very small error term, and the output of an OR gate may be conservatively approximated by using an assumption that the inputs are mutually exclusive events:

$$P(A \text{ or } B) \approx P(A) + P(B),\ P(A \cap B) \approx 0$$

An exclusive OR gate with two inputs represents the probability that one or the other input, but not both, occurs:

$$P(A \text{ xor } B) = P(A) + P(B) - 2P(A \cap B)$$

Again, since $P(A\cap B)$ usually becomes a very small error term, the exclusive OR gate has limited value in a fault tree.

FTA involves the following steps:

1. Define the top event.
2. Know the system.
3. Construct the tree.
4. Validate the tree.
5. Evaluate the tree.
6. Study tradeoffs.
7. Consider alternatives and recommend action.

1- ***Define the top even* -** To define the top event the type of failure to be investigated must be identified. This could be whatever the end result of an incident may have been, such as a forklift overturning. Determine all the undesired events in operating a system. Separate this list into groups having common characteristics. Several FTAs may be necessary to study a system completely. Finally, one event should be established representing all events within each group. This event becomes the undesired event to study.

2- ***Know the system* -** All available information about the system and its environment should be studied. A job analysis may prove helpful in determining the necessary information.

3- ***Construct the fault tree* -** This step is perhaps the simplest because only the few symbols are involved and the actual construction is pretty straightforward. Principles of construction. The tree must be constructed using the event symbols listed above. It should be kept simple. Maintain a logical, uniform, and consistent format from tier to tier. Use clear, concise titles when writing in the event symbols. The logic gates used should be restricted to the AND gate and or gate with constraint symbols used only when necessary. An example would be the uses of the oval constraint symbol to illustrate a necessary order of events that must happen to have an event occur. The transfer triangle should be used sparingly if at all. The more the transfer triangle is used, the more complicated the tree becomes. The purpose of the tree is to keep the procedure as simple as possible.

4- ***Validate the tree* -** This requires allowing a person knowledgeable in the process to review the tree for completeness and accuracy.

5- ***Evaluate the fault tree* -** The tree should then be scrutinized for those areas where improvements in the analysis can be made or where there may be an opportunity to utilize alternative procedures or materials to decrease the hazard.

6- ***Study tradeoffs* -** In this step, any alternative methods that are implemented should be further evaluated. This will allow evaluators to see any problems that may be related with the new procedure prior to implementation.

7- ***Consider alternatives and recommend action* -** This is the last step in the process where corrective action or alternative measures are recommended.

IV. IMPLEMENTATION OF FAULT TREE ANALYSIS

In implementing of fault tree analysis these step should be noticed:

- Identify the failure effect to be analyzed. Typically this will be a critical effect that must be eliminated or reduced. It should be a complex failure, which may be caused by combinations of other failures, rather than a low-level failure with simple causes. This may be found using other tools, such as Failure Mode and Effects Analysis.
- Write the failure effect in a box at the top-center of the diagram area. Make this a clear phrase that describes the effect as precisely as possible, describing not only what the failure is, but how it occurs. For example, 'carburetor fails when engine reaches full temperature'.
- List failures that may directly contribute to the failure described in step 2. For example, 'fuel delivery failure', 'air intake blockage', etc. When identifying ways in which an item may fail, try looking at the problem from different angles. For example:

- o Excessive stresses and strains.
- o Potential misuse and abuse.
- o Environmental extremes.
- o Natural variation in the system.
- o Failure of dependent systems.
- o Failure of related processes.

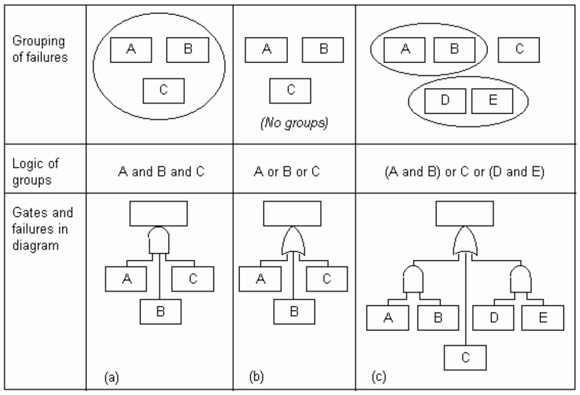

Fig. 1. Grouping failures under gates [10]

- Divide the list of failures in the list derived in step 3 into separate groups, where all members of each group must occur together for the failure in step 2 to occur. For example, 'dirt in fuel' and 'partially blocked jet'. There are three possible outcomes from this:

a) There is one group, as all failures identified in step 3 must occur together for the failure from step 2 to happen. This is an 'and' group, so draw an 'and' gate under the failure from step 2 and connect this to boxes underneath containing the failures from step 3, as in (a) in the illustration below.

b) No such groups can be found as any one failure from step 3 can result in the failure effect from step 2. This is an 'or' group, so draw an 'or' gate under the failure from step 2 and connect this to boxes underneath containing the failures from step 3, as shown in (b) in the illustration below.

c) There are several groups. This is a complex grouping, so draw each group with more than one member under an 'and' gate and connect these gates to an 'or' gate under the failure effect from step 2, as shown in (c) illustrated in Fig.1.

It may also be worth checking whether any 'and' group actually constitutes an independent failure effect. This can be shown with an additional failure box above the 'and' gate.

There may also be additional conditions for a failure or group of failures to occur. For example, environmental or procedural conditions such as 'ambient temperature >50° C' or 'engine idling'. These may be shown with an inhibit gate, as in Fig. 2.

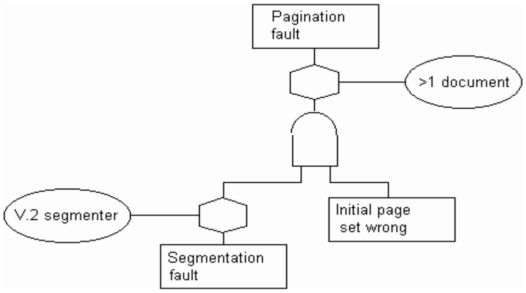

Fig. 2. Adding inhibit gate [12]

- For each failure which has no connections below it, decide whether or not to develop this further by finding other failures which may contribute to it. If the failure is not to be developed on this diagram, draw it in an appropriate box. Thus, if the failure cannot reasonably be developed further, put it in a circle; if it could be developed, but is not appropriate to do this here, and then use a diamond-shaped box. If the failure is to be developed, repeat step 3 to find contributory failures and appropriate gates.
- When the diagram is complete, examine it to draw conclusions and plan for appropriate actions. For example, acting to reduce risks such as critical failures and safety hazards [16].

V. CONCLUSION

This paper surveys on the Fault Tree Analysis in modeling of reliability assessment of an engineering system using Boolean algebra. Regardless of complexity of the modeling of large scale system, this approach can be implemented for calculation the reliability indices.

REFERENCES

[1] R.U. Nighot, Incorporation substation and switching station related outages in composite system reliability evaluation, MSc. Thesis, College of Graduate Studies and Research, University of Saskatchewan, Canada, 2003.

[2] C.C. Fong, C.H. Grigg, Bulk power system reliability performance assessment, Reliab. Eng. Syst. Saf. 46 (1994) 25–31.

[3] R. Billinton, R.N. Allan, Reliability Evaluation of Engineering Systems: Concepts and Techniques, Plenum Press, New York, 1992.

[4] M.J. Beshir, T.C. Cheng, A.S.A. Farag, Comparison of two bulk power adequacy assessment programs: TRELSS and COMREL, in: IEEE Transmission and Distribution Conf., Los Angeles, 1996.

[5] H. Haroonabadi, M.R. Haghifam, Generation reliability evaluation in power markets using Monte Carlo simulation and neural networks, in: 15th International Conf. on Intelligent Sys. Applications to Power Systems, Curitiba, 2009.

[6] C.L.T. Borges, D.M. Falcao, J.C.O. Mello, A.C.G. Melo, Composite reliability evaluation by sequential Monte Carlo simulation on parallel and distributed processing environments, IEEE Trans. Power Syst. 16 (2) (2001) 203– 209.

[7] K. Alvehag, Impact of dependencies in risk assessments of power distribution system, Licentiate Thesis, Royal Institute of Technology, Sch. Elec. Eng, Stockholm, Sweden, 2008.

[8] R. Billinton,W.Li, Reliability Assessment of Electric Power Systems using Monte Carlo Methods, Plenum Press, New York, 1994.

[9] R. Billinton, S. Kumar, Indices for use in composite generation and transmission system adequacy evaluation, Power Syst. Res. Group 12 (3) (1990) 147–155.

[10] R.N. Billinton, L. Allan, Salvaderi, Applied Reliability Assessment in Electric Power Systems, IEEE Press, New York, 1991.

[11] N. Cross, R. Herman, C.T. Gaunt, Investigating the usefulness of the beta PDF to describe parameters in reliability analyses, PMAPS (2006).

[12] W. Wangdee, R. Billinton, Reliability assessment of bulk electric systems containing large wind farms, Int. J. Elec. Power Energy Syst. (2007), doi:10.1016/j.ijepes.2007.06.028.

[13] M.H.J Bollen, Effects of adverse weather and aging on power system reliability, IEEE Trans. Ind. Appl. 37 (2) (2001) 452–457.

[14] T. Solver, Reliability in performance-based regulation, Licentiate Thesis, Royal Institute of Technology, Sch. Elec. Eng., Stockholm, Sweden, 2005.

[15] H.L. Gray, W.R. Schucany, Lower confidence limits for availability assuming lognormally distributed repair times, IEEE Trans. Reliab. R 18 (4) (1969) 157– 162.

[16] http://syque.com/quality_tools/toolbook/FTA/do.htm