Introduction to Fault Tree Analysis

David A. Nelson

David A. Nelson
KBRwyle
320 Corporate Way Suite 100
Orange Park, FL 32073
e-mail: david.nelson@wyle.com
SUMMARY & PURPOSE

The objective of the tutorial is to provide an introductory understanding of FTA and a few examples of application. The instructor will explain the basic rules and mathematics used in FTA and illustrate the explanations with examples.

David A. Nelson

David Nelson is an engineer and reliability practitioner. He is currently the Chief Reliability Engineer for the Sustainment Engineering Division for KBRwyle, serving the defense and aerospace industry. He is a Mechanical Engineering (Metallurgical Option) graduate of Texas A&M University with over 33 years’ experience including active service as a Naval Aviator. His expertise includes application of various reliability methods in design, supportability, test and evaluation, and risk analysis. Skill sets include Fault Tree Analysis (FTA), Life Data Analysis (Weibull), Reliability Centered Maintenance (RCM), development of diagnostic processes (specifically using Oil Analysis and Filter Debris Analysis), Root Cause Analysis, Failure Modes Effects and Criticality Analysis (FMECA) and other reliability disciplines.

Mr. Nelson is an American Society for Quality (ASQ) Certified Reliability Engineer and a Society of Maintenance and Reliability Professionals (SMRP) Certified Maintenance and Reliability Professional (CMRP). He is a member of the American Society of Mechanical Engineers (ASME), the Society of Automotive Engineers (SAE), and ASQ.

Table of Contents

1. Introduction .......................................................... 1
2. FTA Basics ............................................................ 1
3. Terminology and Symbols ......................................... 2
4. Fault Tree Construction ............................................. 3
5. The Math Behind the Analysis .................................... 4
6. Fault Tree Evaluation ................................................. 5
7. Why and When to Apply FTA ...................................... 9
8. Understanding the Rules and Limitations.............................. 9
9. Summary .................................................................. 10
10. Case Study Examples ................................................. 10
11. References ................................................................ 11
12. Tutorial Visuals.......................................................... 12
1. INTRODUCTION

Some people assume Fault Tree Analysis (FTA) is strictly a safety method. FTA is applicable to analyzing any top event defined as an event where occurrence is undesired and the failure causes are being analyzed by FTA. Understanding the combinations of faults and conditions required for a specific top event to occur is the primary objective of FTA. The overview and discussion of the methodologies of FTA provide an introductory knowledge to the attendees. The methods discussed will include the core method for graphically representing events and/or conditions that result in the specified top event and symbols used to represent the relationships between the events. The graphical representation uses a top down method based on the use of deductive logic with both qualitative and quantitative representations of combinations of faults.

The discussion will define basic concepts in two major evaluation methods of fault trees. The first method is qualitative analysis relating to a defining all the combinations of basic faults that result in the top event referred to as cut sets. Qualitative analysis typically defines more than one cut set that can result in the top event. The cut sets with the fewest faults are the minimal cut set. Cut sets and minimal cut sets are described with examples. The second method explored and discussed will be the quantitative method using Boolean algebra, probability, and failure rate evaluation. The quantitative methods include illustrations with simple examples from the wide variety of applications of FTA. FTA application includes modeling system dependability, reliability analysis, and the more commonly associated application to safety analysis. Discussion will include algorithms to identify minimum cut sets, techniques to measure event importance, and calculating the probability of occurrence for the specified top event. Fault Tree Analysis is a proven analytical tool for complex systems.

This introductory tutorial will present the concepts of fault tree, definitions of terminology, illustrations of mathematics, and discussions of application. The objective is a basic understanding of Fault Trees, the analytics used in evaluation, and some of the applications, particularly in relation to reliability and risk. The theory and subject matter will require more than just this session to gain true expertise. The discussion will also include a list of resources to gain more information and additional instruction.

2. FTA BASICS

FTA is an analysis tool “concerned with the identification and analysis of conditions and factors which cause or contribute to the occurrence of a defined undesirable event, usually one which significantly affects system performance, economy, safety or other required characteristics.” (ISO 60125) FTA is a deductive (top-down) method of analysis with an objective to identify the causes or combinations of causes that can lead to the defined top event. The term “undesirable event” refers to a “top event” in used interchangeably to describe a pivotal event defined for a given scenario. Top events are typically defined from either a safety or reliability perspective. The top event is a terminus for the logic relationships of intermediate and basic events that result in its occurrence. The deduction process is performed in steps from the top down so that the fault tree embodies all component failures that contribute to the occurrence of the top event. The result is FTA considers multiple, as well as single failure events. Consequently, it is an important tool for safety, reliability, and root cause analyses.

Central to the analysis is the graphical model showing the logical connections between events in relation to the top event. The graphical model uses three basic symbols: events, gates, and transfers. The event symbol is a text box containing a short description of the event. The first event described is the top event. Other events are logically deduced downward from the top event. There is an inherent cause and effect relationship between events. However, a fault tree should not be confused with other analysis methods, like Failure Modes Effects and Criticality Analysis (FMECA), which seek to identify all possible failures of components in a system. A fault tree does not necessarily contain all possible failure modes of the components of the system or system operation. Only those failure modes that contribute to the existence occurrence of the top event are modeled. Each top event description is specific so that only those events that result in the specific top event apply. If a new top event is described, a fault tree must be constructed that contains only events that result in the new top event. As a result, systems or subsystem may require multiple FTAs to identify and characterize the failure modes and mechanisms for all undesired events. Gate symbols define the logical requirements of the lower level events to result in the event immediately above. As the gates are completed (logic satisfied for the event above the gate to occur), the fault tree progresses through the gates culminating in the top event after the uppermost gate inputs are satisfied.

As an analysis method, FTA has its origins in the early 1960’s; Bell Laboratories developed the method for Minuteman missile systems. Other companies and organizations in aviation and defense adopted the method. Eventually the value of FTA in identifying all combinations of events that lead to an undesired top event led to adoption by the Nuclear Regulatory Commission and NASA.

FTA starts as a graphical model with logic gates defining the relationship between events. Evaluation is qualitative and quantitative with analysis involving a specific branch of mathematics called Boolean algebra, probability, and failure rates. FTA flows from the top down from a single result, the Top Event or “Undesired Event (UE).” FTA uses deductive logic, reconstructing events and relationships leading up to the top event by tracing backwards, “one-step at a time” through intermediate events to primary or basic events. The definition of “deduce”, according to Merriam Webster is “to trace the course of” or “figure out using reason or logic.” The words conclude, derive, extrapolate, and infer are all synonyms for
deduce. Deduction is essentially the Sherlock Holmes approach for logically determining what leads to the top event. For the top event to occur, other events must occur following the premises governing how those things occur. For top events resulting from component or system failure, the premises are the natural laws that govern the physics of failure and/or the design of the system that uses those same laws to produce a desired result. Human error is also typically included since system failures in particular involve some level of human interaction in operation and/or maintenance. The objective of this statement is not to proclaim the obvious but to emphasize a truth that applies to any analysis. The challenge in a successful FTA is often concerned with keeping everyone focused on interactions in the context of natural law, design, and human performance. Only by applying deduction to these laws is an accurate and useful FTA modeled. As with many reliability methods that involve investigative techniques, the interjection of implausible scenarios prefaced by a “what if” statement need to be handled with sound reasoning and judgment.

The logic flow of what needs to be present or occur (failure, normal operation, human error, environmental or any other aspect of the system) has to “go through the gates” to result in occurrence of the top event. That flow is expressed logically and mathematically allowing probabilistic prediction for system performance thereby identifying what is important individually and in combination. The logic and math identify the high-risk paths, deficiencies, common mode and common cause failures of the system. The qualitative and quantitative interpretation provides better information for better decisions. Better decisions early in design, development, and production enable higher in-service reliability and reduced risk. In-service, FTA is applied in investigative efforts to identify causality and eliminate failure modes that result in undesired events.

A basic process for FTA comprises several steps. This particular description of the process comes from the NASA Office of Safety and Mission Assurance Fault Tree Handbook with Aerospace Applications, Version 1.1 published in 2002. This handbook, was preceded by the US Nuclear Regulatory Commission (NRC) issued Fault Tree Handbook, NUREG-0492. Both of these publications are available open source. The content is thorough, authored by experts in FTA application from each agency and academia. There are common authors for each, including Dr. William Vesely, a key leader in FTA application at the NRC. In many ways, the NASA handbook is an update to NUREG-0492. Both of these references are included in the recommended list at the end of this paper. The NASA handbook defines an eight-step process.

1. Identify the objective for the FTA.
2. Define the top event of the FT.
3. Define the scope of the FTA.
4. Define the resolution of the FTA.
5. Define ground rules for the FTA.
6. Construct the FT.
7. Evaluate the FT.
8. Interpret and present the results.

As with most analysis processes, the FTA process begins with a clearly defined objective relating to a decision. The decision can relate to evaluating designs or identifying root causes. Performing the first five steps is essential to a successful FTA. Constructing a model and producing an evaluation suitable for practical interpretation to support a correct decision is difficult if not impossible without a clear objective, a well-defined top event, documented decisions relating to scope and resolution, and ground rules governing the conduct of the analysis. Steps 1 and 2 must proceed sequentially. The description of the objective should be in terminology consistent with the failure of the system under analysis. The definition of the top event will derive from the objective. Steps 3-5 can proceed concurrently. Some iteration of the ground rules will occur during later steps. Any difficulty or situations requiring decisions about scope and/or ground rules encountered during construction and evaluation requires a feedback loop to steps 3-5. The feedback and any resulting iterations in the analysis process lead to a completed FTA construction and evaluation suitable for interpretation and subsequent decision in the context of the objective.

3. TERMINOLOGY AND SYMBOLS

The graphical model uses three basic symbols: events, gates, and transfers. Events are shown by a rectangular text box containing a basic description of the event. Gates are depicted by symbols that identify the logical construct for the input to output relationship required to “go through” or “complete” the gate. Transfer symbols are used to depict internal and external relationships in the organization of the tree.

3.1 Event Symbols and Definitions

Construction of the fault tree begins with the top event and the deduction of events that result in the occurrence of the top event. The symbol for an event is a box containing a basic description of the event. Once the first event description is complete, other events are deduced downward from the top event. The deduction is a breaking down of the higher event into parts with lower events describing all the possible ways to get the event immediately above. The breaking down continues until a point is reached where basic events, which serve as the enabling inputs to the tree, are identified. Basic events are represented by circles, always at the furthest point down any route leading to the top event. Between the top event and the basic events are all the intermediate events. Intermediate events, like the top event, are represented by text boxes. The intermediate events represent all the combinations of events that are immediate, necessary, and sufficient causes for the occurrence of the intermediate (or top) events. Like the top event, intermediate events are represented by a text box containing a basic description. The concept of immediate, necessary, and sufficient is a foundational role to the FTA methodology. These concepts will be discussed in further detail in another section of the paper.
3.2 Gate Symbols and Definitions

As each event is deduced it must be connected logically to the event above it. The connection is made with gate symbols. Gate symbols define the logical requirements of the lower level events to result in the event immediately above. The fundamental logic symbols are the AND gate and the OR gate. Each of these has at least two inputs and a single output. The output for the AND gate occurs if and only if all inputs occur. The output for the OR gate occurs if and only if at least one input occurs. If more complex logical relationships are required, other logical representations can be described by combining the logic of the two basic AND and OR gates. For example, the K-out-of-N and exclusive OR logic gates can be described.

3.3 Transfer Symbols and Definitions

Transfer symbols permit management of trees as they become more complex with a large number of events and gates making them too large to display with clarity. Transfer gates allow creation of sub-trees that can be inserted into a main tree. When the transfer appended the sub-tree to a main tree it is referred to as an external transfer. Transfer symbols also provide a means to deal with managing time by eliminating the need to redraw branches of the tree that are duplicated in other locations. This type of transfer, referred to as an internal transfer, applies when the purpose for displaying the transfer is to indicate where a sub-tree is repeated internally in the fault tree. The symbol is usually a triangle. There are multiple variations of the triangle symbol in common use. In one example, the triangle is underlined or inverted to indicate a particular type of transfer. Another variation attaches the connecting line to the triangle in one place for an internal transfer and another for an external transfer. The latter is used in table of symbols.

3.4 Other Symbols and Definitions

An oval is commonly used as a condition symbol. Condition symbols are attached to the side of a logic symbol (AND or OR gate) to document priority, exclusivity, and inhibit requirements that must be satisfied for the next event up the tree to occur. The gate symbol itself may also be modified with an additional line (for the AND gate), arc (for the OR gate) or triangle (for either gate) to make a distinction that a priority or exclusivity condition applies.

In some instances an external event, not part of the system, induces a secondary failure required for the occurrence of the top event. These secondary failures are most commonly displayed as a diamond in the tree. A double diamond often depicts a special event not definable as a purely external event.

There are instances where a normal event vice a fault, failure, or induced event is a required event for the occurrence of the top or undesired event. For such instances, normal events require documentation in the FTA and consideration qualitatively and quantitatively in the evaluation. Normal events are often depicted with the top of the box formed by two lines from each edge angled upward converging to a point like a gabled or hip roof giving the symbol the appearance of a house.

FTA symbols have some variation in books and standards. There is typically consistency in the events with a rectangle depicting the top event and intermediate events, circles for basic events, and an oval for any conditions. The basic “AND” and “OR” gate symbols are also consistent. Most of the variation comes in depicting special gates with priority or exclusivity in the logic. Table 1 below shows examples of FTA symbols.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
<th>Definition and Use</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="Intermediate or Top Event" /></td>
<td>Intermediate or Top Event</td>
<td>Depict any event resulting from a basic event or combination of basic events. Depicts both intermediate events and the top event.</td>
</tr>
<tr>
<td><img src="image" alt="Basic Event" /></td>
<td>Basic Event</td>
<td>A basic event that enables the path to the top event.</td>
</tr>
<tr>
<td><img src="image" alt="AND Gate" /></td>
<td>AND Gate</td>
<td>A gate where all inputs must be satisfied for the event above to occur.</td>
</tr>
<tr>
<td><img src="image" alt="Priority AND Gate" /></td>
<td>Priority AND Gate</td>
<td>A gate where all inputs must occur in a specific order/sequence for the event above to occur.</td>
</tr>
<tr>
<td><img src="image" alt="OR Gate" /></td>
<td>OR Gate</td>
<td>A gate where only one input must be satisfied for the event above to occur.</td>
</tr>
<tr>
<td><img src="image" alt="Exclusive OR Gate" /></td>
<td>Exclusive OR Gate</td>
<td>A gate where only one specific input can result in occurrence of the event above. Other inputs do not result in the event above.</td>
</tr>
<tr>
<td><img src="image" alt="Transfer" /></td>
<td>Transfer</td>
<td>A transfer in or out of the fault tree. If the transfer is a transfer-in, a line attaching to the tree will extend from the top vertex. If it is a transfer-out, the line will intersect a side leg of the triangle at the mid-point.</td>
</tr>
<tr>
<td><img src="image" alt="Condition" /></td>
<td>Condition</td>
<td>Condition required for a gate, most often display attached to the side of the gate itself.</td>
</tr>
<tr>
<td><img src="image" alt="External Event" /></td>
<td>External Event</td>
<td>An external event or sometimes called an undeveloped event.</td>
</tr>
<tr>
<td><img src="image" alt="Normal Event" /></td>
<td>Normal Event</td>
<td>An event that is part of normal operation, a “normal” event.</td>
</tr>
</tbody>
</table>

4. FAULT TREE CONSTRUCTION

Events and gates make the logic, all “flowing down” from the top event. Event descriptions, particularly the top event, are critical elements of the fault tree methodology. Event descriptions, though simple and brief, must include not only...
undesired or failure states of the component but also a description of when the event resulting in that state occurs. The “what” and “when” criteria is essential to all event descriptions. The first event described is the top event. Other events are deduced downward from the top event. The deduction is a breaking down of the higher event into parts with lower events describing all the possible ways to get the event immediately above. Therefore, deducing events typically results in more than one lower event feeding into the event from which the deduction was derived. Figure 1 shows a simple fault tree.

![Simple Fault Tree](image)

**Figure 1: Simple Fault Tree**

The gates connecting events are input-output logic statements with the higher fault event under consideration the gate output and the more basic (“lower”) fault (or failure) events that relate to the output the gate inputs. When a fault tree is drawn, the tree is developed from the “higher” faults to the more basic faults (that is from output to inputs). In this process, certain techniques are necessary to determine which category of gate is appropriate.

A specific methodology or “rules” are applied. There are six basic rules that apply:

1. The “immediate, necessary, and sufficient” rule is a criterion to ensure that the logic is satisfied in small steps and that specific items in the tree are not overlooked by jumping to a less specific more basic cause conclusion leaving out much of the detail required to fully understand what combination of events results in the progression to the top event. It is violation of this rule that often leads to poorly constructed trees. The immediate criteria refers to what occurs immediately prior to the higher event, one step immediately prior not multiple steps, in the progression to the higher event that is the output of the gate. The “closest in space, time, and derivation” is often used to define the immediate criteria. The necessary criteria refers to the event immediately prior being required for the higher event to occur. In other words, without the event as an input the progression to the output of the higher event will not occur. The “higher” event could not result from a subset of the lower events, that is one missing but the others present in the progression.” This does not refer to OR logic gates requiring options of which only one is required, but to the cause and effect sequence leading up to the event. Essentially, necessary means you can’t skip or jump an input and get the same output. The sufficient criteria means the event immediately prior, by itself, provides all that is needed for the progression to the next event. That is “always and in all conditions” the event as an input to the logic will lead to the output of the higher event. A one small step at a time mindset is crucial to application of this rule, especially in relation to the immediate criteria.

2. The “clear statement” rule has some relation to law. For proper interpretation of statues in law, the language chosen must have a plain and straightforward meaning without vagueness or ambiguity. The event statement must be straightforward and clear, precisely describing the event. Statements must communicate definitively the “what” and “when” for each event. A common example that applies to all engineering and reliability analysis methods is describing valve failure. A valve, by definition, has positions. To state that a valve is failed without stating in what position (open or closed) violates the clear statement rule.

3. The “system or component fault” rule refers to classifying events clearly. This rule is usually approached by asking the simple question “Does this fault event consist of a component failure?” A “yes” answer classifies as a component fault, a “no” answer defaults to a system fault.

4. The “no miracles/no magic” rule refers to considering miraculous action by an operator or a coincident failure that “as if by magic” prevents the occurrence of the top event at any point in the progression. The concept that normal propagation of a fault tree occurs under normal operation of the system applies. Considering human error is appropriate and necessary, but incredible feats of human skill that prevent occurrence are not appropriate.

5. The “complete the gate” rule refers to defining all inputs to a particular gate before attempting to analytically deduce any one of them further. Without completing the gate there is no logical structure on which to logically define any lower events.

6. The “no gate-to-gate” rule means the construction of the tree must be based on the input-to-output, cause-to-effect concept. Gates define the connection between events. Gates do not define connections to other gates.

**5. THE MATH BEHIND THE ANALYSIS**

The two basic gate categories, the OR-gate and the AND-gate, relate events in exactly the same way as Boolean algebra operations. There is a one-to-one correspondence between the Boolean algebraic representation and the fault tree representation. Most people hear the term algebra and relate it to elementary algebra. Elementary algebra concerns
numerically defined variables. Boolean algebra concerns logically defined variables. Boolean algebra is a branch of mathematics based on logic that has its own set of axioms and theorems used to define and reduce Boolean expressions. The variables used in Boolean algebra only have one of two possible values, a logic “0” and a logic “1.” The variables when applied as numerical values for a “true” or “false” state for the case of FTA give mathematical representation for the state of the input from a lower event to the output of the higher event. For example, the AND logic gate has the combination of values for the inputs of (0,1) (1,0) (0,0) and (1,1). Assigning the value of “1” to the input variable when the input is true (that is fault or failure occurs), the result gives an expression where only the combination where both inputs are true the (1,1) satisfies the logic for the higher event to occur. Boolean expressions can therefore have an infinite number of variables all labelled individually to represent inputs to the mathematical expression of the fault tree. Boolean algebra is the basis for modern binary code operations in computer science. It is the basis for logic decisions in many fields. One does not have to become an expert in Boolean algebra to perform FTA.

For FTA, Boolean algebra defines the event and gate logic that forms the foundation of the qualitative and quantitative evaluation of the cut sets. The fault tree involves essentially three logical possibilities and hence two main symbols. These involve gates such that the inputs below gates represent failures. Outputs (at the top) of the gates represent a propagation of failure depending on the nature of the gate. The gate output is the “higher” fault event under consideration and the gate inputs are the more basic (“lower”) fault (or failure) events that relate to the output.

Some of the more common gate types are:
- The OR gate whereby any input causes the event to occur:
- The AND gate whereby all inputs need to occur for the output to occur;
- The voted gate, similar to the AND gate, whereby two or more inputs are needed for the output to occur.
- The exclusive OR gate whereby the output occurs if only one input event occurs
- The priority AND gate whereby the output occurs only when the inputs all occur in a specific order

For this introduction to FTA, the focus will primarily be on the logic of the AND and OR gates. The AND gate models the redundant case and is thus equivalent to the parallel block diagram. The OR gate models the series case whereby any failure causes the top event. In probability terms the AND gate involves multiplying probabilities of failure and the OR gate applies the addition rule of probability. The probability relationship is inherent in the logic governing Boolean algebraic expressions where the OR-gate represents a union of probability of occurrence for the input events and the AND-gate the intersection of the probability of occurrence for the input events. Boolean algebraic operators may look the same as standard mathematical operators common to elementary algebra. However, Boolean operators have a different context because the value set is not numerical but logical represented by (TRUE, FALSE) or (1,0) values. To clarify how the Boolean operators are applied each basic logic gate is separately discussed below. The relationship of probability in the input to output logic flow becomes readily apparent.

The tree OR-gate logic represents the union of the input events. The OR gate with two inputs, events A and B and the output event Q can be represented by its equivalent Boolean expression, \( Q = A \lor B \). Either A or B or both must occur for the output event Q to occur. Instead of the union symbol “\( \lor \)”, the equivalent “+” symbol is used in FTA. Thus, \( Q = A + B \) is the mathematical expression for an OR gate. This directly follows probability theory where success is defined by the occurrence of one of two outcomes. Under those circumstances, the probability of success is the sum of the probability of each of the two outcomes. An OR gate can have more than two inputs. If only one of many is required for success the same relationship applies. Generally, for an OR gate with “n” inputs requiring only one for the occurrence of the higher event, \( Q = A_1 + A_2 + \ldots + A_n \) will define the probability.

The AND gate can be represented by the intersection of the input events. If either A or B occur, the output event Q will not occur. Only when both A and B occur does the output event Q occur. Therefore, the Boolean equivalent of an AND gate with two inputs A and B would be \( Q = A \land B \). As with the OR logic, an equivalent “*” symbol is used in FTA. Therefore, \( Q = A \ast B \) is the mathematical expression of the AND gate. This directly follows probability theory where success is defined by the simultaneous occurrence of two independent outcomes. The probability of success is therefore the product of the probability of each of the two outcomes where occurrence of each is independent from the other. As with the OR gate, an AND gate can have more than two inputs, if all are required for success the same relationship applies with more terms. Generally, for an AND gate with n inputs requiring all for the occurrence of the higher event, \( Q = A_1 \ast A_2 \ast \ldots \ast A_n \) will define the probability.

Probability relationships in reliability engineering generally follow the same underlying logic as Boolean operations.

6. FAULT TREE EVALUATION

6.1 Qualitative Methods

The qualitative analysis is principally concerned with identifying the minimal cut sets of the top event. Minimal cut set identification is required to perform quantitative analysis, specifically calculating probability of occurrence for the top event. A cut set is a combination of events that can cause the top event. A minimal cut set is the reduction of cut sets to the smallest combination of basic events that result in the top event. Qualitative evaluation methods derive the minimal cut sets, by a variety of methods. The best method to ensure a thorough and complete evaluation is to apply an algorithm to identify the cut sets. There are multiple methods and
algorithms used, to introduce the concept of the minimal cut set; one of the more common methods will be illustrated. A simple system is represented in Figure 2. The system is a hydraulic system with redundant pumps providing pressure to a common manifold for distribution.

![Figure 2: Block Diagram of Simple Redundant Hydraulic System](image)

From the diagram in Figure 2, construction of the fault tree begins with a logical deduction from a specified top event. Fault trees typically have multiple intermediate events before a basic or primary event is defined. The example is simplified for purposes of introducing FTA concepts. It should be obvious that very similar descriptions for the intermediate events and the basic events progression under the OR gates for the “no pressure” events for the two hydraulic pumps are given. This is done only to illustrate that all branches or legs of the tree end in a basic event.

![Figure 3: Fault Tree for Simple Redundant Hydraulic System](image)

As discussed earlier, the description of the objective should be in terminology consistent with the failure of the system under analysis. The definition of the top event will derive from the objective. An example objective for Figure 2 could be to evaluate design changes to ensure hydraulic pressure from the system is always available. An example description of the top event, consistent with the failure of the system, derived from that objective is “no hydraulic pressure from the system.” The deduction down to the intermediate events and basic events at the bottom of the fault tree are illustrated in Figure 3. The gate symbols also contain the Boolean operator, the “+” or “*”, appropriate for that gate. The function of the operators will be discussed further when describing how fault trees are evaluated.

Cut sets relate the top event directly to the basic event causes.

The set of minimum cut sets for the top event represents all the ways that the basic events can cause the top event. Cut sets are not only obtainable for the top event, but for any of the intermediate events (e.g., gate events) in the fault tree.

To evaluate the fault tree qualitatively, the Method of Obtaining Cuts Sets (MOCUS) algorithm is applied. The method begins by naming all the gates and numbering all the events in the tree. Table 2 shows a list of gate names and numbered events. The list includes the description of each event and a short name. The naming of the gates and numbering of events allows representation by short names in the algorithm used for obtaining the cut sets.

![Table 2: List of Events for Fault Tree in Figure 2](image)

Applying the names and numbering systems to the fault tree results in the labeling shown in Figure 4. There are three levels of gates leading up to the top event. The primary purpose of the algorithm is to identify the elements at each level in order to uncover the number and length of the minimal cut sets.

Next, the inputs to the uppermost gate are placed in the upper left hand of a matrix. Gates are replaced with their respective inputs in one of two ways. For an AND gate, each input is listed on the same row separated by a comma. For an OR gate, each input is listed on separate rows creating a new line under each input in the same column. The substitution is repeated in a deliberate method going down one level below each gate and listing the inputs for each gate as left to right. Placing the inputs under Gate 1 in the upper left hand corner...
then expanding the matrix in steps going down each of the three levels and working from left to right to substitute the inputs under each gate completes the matrix. As the inputs are entered, annotated by either an event number or a gate number, the matrix expands to the right and down.

Event 1 is the first minimal cut set for the tree identified, it only requires one event to be “TRUE” (Boolean logic value of “1”) for the top event to occur. Event 1 cannot be reduced further; therefore, it is a minimal cut set. It is a first order cut set because it has only one event. Cut sets with more than one event are classified, as “n” order sets where “n” is the number of basic events in the set. Other sets are deduced as the matrix continues, moving right adding additional steps and expanding downward as the list of sets expands.

The presence of a basic event directly under the first gate results in one event and one gate, G2, for step 1. In step 2, the event from step 1 is carried over and gate G2 listed under step 1 is broken down into the events under it. Since G2 is an AND gate the two events are listed in the same row, separated by a comma. The listing on the same row signifies that both events are required for the tree to progress beyond G2. Events for an OR gate are listed on separate rows signifying that one or the other can result in the occurrence of the top event. The process is continued until no more gates are listed. The result is a complete list of all the cut sets. Step 5 is the list of six cut sets for the simple redundant hydraulic system. With the exception of the previously mentioned first order cut set, the cut sets all require two events to be simultaneously “TRUE” with the logic value of “1” for the top event to occur. These are all therefore second order cut sets. However, they are not all minimal cut sets. The algorithm has one more step to obtain the final list of minimal cut sets for the system. The cut set with events 2 and 3 is not a minimal cut set because it can be further reduced below the two OR gates. Once non-minimal cut sets are removed, the algorithm has obtained the final list of minimal cut sets. The minimal cut sets for the simple redundant hydraulic system are the five remaining sets that cannot be reduced further: (1), (4,6), (4,7), (5,6), and (5,7).

Minimal cut set and minimal path methods are primary approaches to qualitative analysis for fault trees. The algorithm described is one method that uses a top down substitution method. There are others that use a similar approach with a bottom up algorithm. Several sources describe the methods in detail with examples. Volume 2 of the two volume Reliability Engineering Handbook by Dimitri Kececioglu is one that the author has found particularly helpful. The NASA FTA Handbook also contains explanations of methods as well as descriptions of complimentary methods to the minimum cut set approach.

Qualitative analysis methods include simple Boolean reduction (manually performed or performed with assistance from software programs), top-down reduction like MOCUS, bottom-up reduction, use of a Binary Decision Diagram (BDD), modularization, and decomposition.

The FT can be validated for configuration accuracy by using one of the following methods:

1) Use the system schematics or functional flow diagrams to verify that minimum cut sets are indeed valid failure paths to the top event. If validating the top event directly is too difficult, e.g., the smallest order cut sets contain a large

![Figure 4: Fault Tree from Figure 3 with Gates and Events Numbered](image)
number of basic events that are difficult to check, then identify lower order faults (i.e., intermediate events) in the FT and validate the cut sets for these faults. This is where the numbering of events is particularly useful. By examining intermediate events upward at the levels defined by the gates and validating their occurrence, a large order cut set can be validated.

2) Identify the smallest success paths of the FT and use the system schematics or functional flow diagrams to validate that these are indeed success paths.

6.2 Quantitative Methods

Quantitative methods calculate the top event and contributing events probabilities. An importance measure of each basic event can also be established. Quantitative approximation methods are the most common approach to calculating top event probabilities. Importance is often expressed as a ratio of a particular cut set’s probability to the probability of occurrence for the top event.

Quantitative approximation methods involve taking the basic event level probabilities, derived from observation or evaluation of component failure rates, and applying them to the minimum cut sets. A basic premise is that the minimum cut sets represent individual root causes for occurrence of the top event. Each cut set has the appropriate probability theory calculation method applied based on the logic gates that define the minimum cut sets. For the example of the simple redundant hydraulic system previously described, the hydraulic pumps and motors are identical. Therefore, there are only three basic events for which probabilities need to be derived. They are manifold fracture, hydraulic pump failure, or motor failure. Deriving probabilities in any reliability context must of course included a specified time. For illustration, the motor in the first position is located adjacent to a furnace and is exposed to much higher temperature than the motor in the second position. Over a 1000-hour operating period, failures occur at twice the frequency for Motor 1 than for Motor 2. Table 4 shows the derived probabilities.

Table 4: Probability of Failure for Basic Events in the Simple Redundant Hydraulic System Fault Tree

<table>
<thead>
<tr>
<th>Basic Event</th>
<th>Probability of Occurrence per 1000-hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manifold fracture</td>
<td>0.000001</td>
</tr>
<tr>
<td>Hydraulic Pump (HP) failure</td>
<td>0.2</td>
</tr>
<tr>
<td>Motor failure (Motor 1)</td>
<td>0.3</td>
</tr>
<tr>
<td>Motor failure (Motor 2)</td>
<td>0.15</td>
</tr>
</tbody>
</table>

Applying these probabilities to the minimum cut sets results in the calculations in Table 5.

To calculate occurrence of the top event, probability theory must be applied. One method is to use an OR gate expansion formula for the number of minimum cut sets identified. Based on probability theory, this is an exact solution based on the premise that the cut sets are independent root causes of occurrence of the top event.

Table 5: Probability Calculation for Minimum Cut Sets of the Simple Redundant Hydraulic System

<table>
<thead>
<tr>
<th>Set ID</th>
<th>Min. Cut Set</th>
<th>Boolean Expression</th>
<th>Probability Calculation</th>
<th>Min Cut Set Prob. of Occurrence per 1000-hours</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>(1)</td>
<td>NA</td>
<td>NA</td>
<td>0.000001</td>
</tr>
<tr>
<td>B</td>
<td>(4, 6)</td>
<td>AND, intersection of input probabilities (both occur)</td>
<td>Multiplication rule</td>
<td>0.06</td>
</tr>
<tr>
<td>C</td>
<td>(4, 7)</td>
<td>AND, intersection of input probabilities (both occur)</td>
<td>Multiplication rule</td>
<td>0.06</td>
</tr>
<tr>
<td>D</td>
<td>(5, 6)</td>
<td>AND, intersection of input probabilities (both occur)</td>
<td>Multiplication rule</td>
<td>0.03</td>
</tr>
<tr>
<td>E</td>
<td>(5, 7)</td>
<td>AND, intersection of input probabilities (both occur)</td>
<td>Multiplication rule</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Depending on the number of minimum cut sets, the expansion formula for an exact calculation can get quite complicated. An expansion formula for four minimum cut sets is shown below:

\[ P = P_A + P_B + P_C + P_D - (P_{AB} + P_{AC} + P_{AD} + P_{BC} + P_{BD} + P_{CD}) + (P_{ABC} + P_{ABD} + P_{ACD} + P_{BCD}) - (P_{ABCD}) \ldots \]

The simple redundant hydraulic system example illustrated has five minimum cut sets. Each additional cut set increases the number of terms in the expansion proportionally. For large fault trees with large numbers of cut sets, the expansion is well beyond manual calculation. It is easy to see why software can be an important tool for FTA. Another method again involves an expansion with the above formula but breaks it into sections known to result in calculations of an upper and lower bound. A third method, based on an approximation of the upper bound results in the following formula for the minimum cut sets identified for the simple redundant hydraulic system:

\[ P = 1 - [(1 - P_A)(1 - P_B)(1 - P_C)(1 - P_D)(1 - P_E)] \]

From this method, \( P \) for the top event of the simple redundant hydraulic system fault tree is approximately 0.1682.

Importance measures are used to establish the significance for all the events in the fault tree in terms of their contributions to the top event probability. Both intermediate
events (gate events) as well as basic events can be prioritized according to their importance. Simply taking the ratio of the probability for each particular minimum cut set in relation to the probability of occurrence for the top event can provide insight into what components contribute the most to occurrence of the top event. Table 6 shows the results for the minimum cut sets of the simple redundant hydraulic system.

Table 6: Example Importance Measures for Minimum Cut Sets of Simple Redundant Hydraulic System

<table>
<thead>
<tr>
<th>Minimum Cut Set</th>
<th>Importance Measure (Ratio of Probabilities)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1)</td>
<td>0.000006</td>
</tr>
<tr>
<td>(4,6)</td>
<td>0.357</td>
</tr>
<tr>
<td>(4,7)</td>
<td>0.357</td>
</tr>
<tr>
<td>(5,6)</td>
<td>0.178</td>
</tr>
<tr>
<td>(5,7)</td>
<td>0.178</td>
</tr>
</tbody>
</table>

Other importance measures look at specific component contributions. One of the most common is the Fussel-Vesely method. This method involves the ratio of the probabilities of occurrence for all minimal cut sets a particular component belongs to the probability of occurrence for the top event. In the example fault tree for the simple redundant hydraulic system, the reliability of Motor 1 is half that of Motor 2. At first glance, it would appear that Motor 1 reliability would therefore be twice as important to the occurrence of the top event as Motor 2. Since each motor belongs to two cut sets, applying the Fussel-Vesely importance measurement method requires multiplying the probabilities for the two minimum cut sets and then dividing them by the probability of occurrence for the top event. The results in Table 7 show Motor 1 is actually 4X more important to occurrence of the top event than Motor 2.

Table 7: Example Fussel-Vesely Importance Measures for Motor 1 and Motor 2

<table>
<thead>
<tr>
<th>Component</th>
<th>Fussel-Vesely Importance Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Motor 1</td>
<td>0.0214</td>
</tr>
<tr>
<td>Motor 2</td>
<td>0.0053</td>
</tr>
</tbody>
</table>

Quantifying top or intermediate event probabilities and applying methods to quantify importance are essential elements of FTA evaluation. The techniques provide a better understanding quantitatively of the different contributions to occurrence of the top event under analysis.

7. WHY AND WHEN TO APPLY FTA

FTA is very useful in identifying possible system/product reliability or safety issues when designing products or systems. Specifically FTA provides a means to isolate safety-critical failures and potential paths leading to an undesired reliability or safety event. When regulatory compliance is required, FTA provides supporting evidence in conjunction with or complementing other safety analyses to assess compliance with safety or other regulatory requirements.

In addition to safety analyses, FTA provides a logical method to assist with the identification of the root cause of a failure observed during testing complimenting the Failure Reporting and Corrective Action System (FRACAS). When managing a reliability growth effort, FTA provides the ability to compare demonstrated reliability of a component to the allocated reliability of the component and study the potential engineering tradeoffs needed to reach a threshold or objective value. Evaluated either qualitatively or a quantitatively FTA is a valuable tool in decision-making, assisting in the elimination of costly design changes that do not produce a measurable impact on safety and/or reliability. In this regard, FTA supports timely mitigation during engineering and development.

When product failures occur in-service, FTA is recommended as a logical way of determining which subassemblies, components or parts might cause the output failure that was observed. FTA can help guide brainstorming what the likely root-causes may be and provide a means for managing the investigation as evidence corroborates or disputes each one of the likely root-causes.

FTA is a preferred method where there is a primary concern for safety of personnel or by-standers. It is particularly effective whenever explicit top events are readily identifiable. For example, a typical FTA for a US Army ground vehicle will have explicit undesired top events related to the mission similar to: 1) Unable to move; 2) Unable to shoot; 3) Unable to communicate. Other conditions or circumstances that result in a preference for FTA as an analysis tool include a high potential for human error, a high level of complexity and/or interconnectivity in function paths required for system success, and situations where systems are not repairable once they are placed into service and functions initiated. The latter condition is common in satellites and other space based systems.

8. UNDERSTANDING THE RULES AND LIMITATIONS OF FTA

Like any analytical method, FTA requires disciplined application within the guidelines of the method. Fault Trees, like block diagramming, are intrinsic (belonging by nature) and derivative (determined from) in relation to the design process. Fault Tree representation of events, conditions, and progression to the top event are empirical (observable and reproducible) in terms of probability and failure rate. An effort to prescribe the architecture and accompanying “design rules” for each fault tree is required. Knowledge of FTA and knowledge of the system operation must be melded together in a team environment with the appropriate buy-in and a common understanding of the objective. The terms cookie cutter or cookbook never apply to FTA.

Many of the shortcomings of any analytical method lie in the human element of the application. The required skill of the analyst in applying the method should not be
underestimated. The six basic rules described earlier must be followed with some rigor and accountability. Validation and verification in a structured review process is highly recommended.

There are some misconceptions about limitations on FTA relating to management and application of an analysis in large complex systems. The management piece is mostly the result of failing to address basic project management practices required to plan, organize, resource, and provide effective communication for the effort.

In some cases, this misconception is a consequence of failing to initially define and later refine the scope of the analysis and the ground rules during construction, evaluation, and interpretation of the tree. FTA is not limited to a specific scope based on the number of events and size of the tree. Part of that planning is including criteria for modularizing at a point when the tree is getting too large for effective analysis at a single level of detail. Ground rules and assumptions are always an evolving concept with “living” documentation.

Resourcing includes providing sufficient information and access to experts to understand the system thoroughly and make that intrinsic and derivative link between the FTA and the system design. Another resource is providing a software tool with reasonably simple user interfaces, intuitive functionality, and easy modification of the tree as the analysis progresses. Constructing a fault tree, like most analytical methods, is an iterative process. The eight-step process describe in the NASA handbook requires management and accountability throughout the process.

For complex systems, the math does get harder, but the logic in the Boolean relationships is sound. Anything that can modeled logically or mathematically can be used to evaluate FTA. Greater fidelity will require greater complexity in the calculation method. The more complex logical relationships of the priority AND gate and exclusive OR gate are prime examples of how more complex failure events can be modeled in FTA. Both provide a means to model timing of events. Multi-phasing can also be applied as well as dependency.

9. SUMMARY

FTA was developed in response to a need to understand events and combinations of events that lead to undesired top events. The genesis in Intercontinental Ballistic Missiles (ICBMs) in itself is a strong case for its usefulness. The complexity of ICBMs and the risk involved with unreliability in those systems is the design criteria under which FTA was developed. Subsequent adoption by the Nuclear Regulatory Commission, aerospace companies like Boeing, and NASA has resulted in further refinement. Design for reliability and failure analysis processes include FTA as a critical tool in reaching a successful result.

FTA is a multi-step (8 steps as described in the NASA handbook) process to identify and analyze conditions and factors which cause or contribute to the occurrence of a specific top event, typically an event with significantly degrades performance (including reliability), impacts cost, and/or poses a risk to safety. The analysis method is primarily graphical incorporating logic gates to define relationships with qualitative and quantitative evaluation. Qualitative evaluation is primarily concerned with identifying minimal cut sets, a sequence of events that are the most basic level are required for the top event to occur. Quantitative evaluation is primarily concerned with quantifying occurrence of the top event and the relative contribution of cut sets and component failures. The analysis utility is in providing an interpretation of system characteristics supporting better decisions about design, failure analysis, and mitigation strategies. That utility is especially preferred for decisions where safety, human error, high complexity, and high interconnectivity contribute to the occurrence of a specific top event. Some rules apply requiring effective project management. Limitations are most often perceived as complexity increases. In most cases, there are logical and mathematical methods that can be applied to overcome any perceived limitations. The references listed at the end of this paper provide additional information and detail of the FTA process and applicable methods.

10. ANALYSIS EXAMPLES

For each example scenario, define the top event and describe the what, when, and how of the intermediate events that are necessary and sufficient for the top event to occur. Identify the logic that connects the intermediate events and leads to the top event. Recall the six basic rules and apply them in constructing the Fault Tree. Use the MOCUS algorithm to derive the minimum cut sets.

Analysis Example Problem 1: A home break occurs while the owner is away.

Analysis Example Problem 2: Car will not start.

10.1 Industry Standards

IEC 61025-2006 Fault Tree Analysis (FTA), International Electrotechnical Commission, Geneva, Switzerland. Available at www.iec.ch

ANSI/AIAA S-102.2.18-2009, “Performance-Based Fault Tree Analysis Requirements,” American Institute of Aeronautics and Astronautics, Reston, VA. Available at www.aiaa.org


10.2 Government Agency


“Fault Tree Analysis Application Guide,” Reliability Information Analysis Center

10.3 Military Standards


11. REFERENCES


INTRODUCTION TO FAULT TREE ANALYSIS (FTA)

David A. Nelson
KBRwyle

Agenda
- Introduction
- FTA Basics
- Terminology and Symbols
- Fault Tree Construction
- The Math Behind the Analysis
- Fault Tree Evaluation
- Why and When to Apply FTA
- Understanding the Rules and Limitations of FTA
- Summary
- Analysis Examples
- References

Introduction
- Fault Tree Analysis (FTA) provides a means to understand the combinations of faults and conditions that lead to a specific undesirable “Top Event” (the event for which failure causes are being analyzed)
- Not just about safety, applies to understanding causal relationships for any event regardless of severity
  - An essential function in design for reliability
  - Part of the block diagramming methodology evaluation
  - Plays a major role “to identify and characterize failure modes and mechanisms”

Introduction
- FTA is a typical element in conceptual modeling and reliability verification processes
- When applied as an input to design decisions, has a significant impact on inherent reliability of a design
- Proven track record with origins in missile systems, nuclear energy, and space

Introduction
- The core method of FTA is graphically representing events using symbols to define the logical relationships between the events
- The graphical representation uses a top down method based on deductive logic to deduce down “one step at time” all the causal paths and events that lead to the occurrence of the top event
- Events are defined based on their relationship to the top event
  - Intermediate events, “on the way” to the top event
  - Basic events, initiate the sequence of events leading to the top event

Introduction
- The relationship is logical, using gates with inputs logically progressing “through the gates” as an output to the next event
- The gate logic has a mathematical construct based in Boolean algebra
- Therefore, the “tree” is not just a graph, it is a logic model of how combinations of faults and conditions result in occurrence of the event under analysis, the “top event”
Introduction

The fault tree, as a logic model of how combinations of faults and conditions result in occurrence of the specified top event can be evaluated in a variety of ways.

Fault Trees can be analyzed
- Qualitative analysis defines the combinations of basic events that result in occurrence of the top event, the 'cut sets'. Drills down to 'minimal cut sets', the basic events that are the maximum requirement for the top event to occur.
- Quantitative analysis uses the Boolean mathematical relationship between the events defined by the logic of the gates to quantify the occurrence of the top event and any intermediate events (events that lead up to the top event).

Fault Tree Analysis – Basics

- Top event requires specific definition of something that occurs.
- The specific definition of the top event means each fault tree is specific only to that event.
  - Alter the definition, the logic is altered, the tree is altered.
  - Define a new top event, the deductive process to develop the tree starts over in the context of the new definition or new top event.
  - Same concept for intermediate events.
  - Change something above and what is under must be reconsidered or at least reviewed to ensure the logic produces the same result.

Fault Tree Analysis Basics

- Gates define the logic for lower events to result in the events immediately above the gate.
- Evaluation begins with a qualitative look at the combinations of events that result in occurrence of the top event.
- Next the evaluation quantifies occurrence of events and the proportion of contribution for each event to the top event.
- It all gets down to the deduction from the top event “tracing the course” of what leads to its occurrence.

FTA deduction is basically like the ‘Sherlock Holmes’ approach to discovering what leads to the top event.

Introduction

FTA is applicable throughout a products life cycle
- Provides a logical, representative and quantifiable basis for decisions.
- Quantifies risk for safety.
- Evaluates design performance.
- Organizes and supports conclusions for failure investigation and root cause.
- Demonstrates regulatory compliance.
- Model human error, external events, or any other action or event that can contribute logically to occurrence of the top event.
- Use is relevant to any event progression where Boolean logic applies.

Fault Tree Analysis – Basics

- Considers multiple as well as single events.
- Graphical model uses symbols to define interaction.
  - Top and intermediate events both use rectangles with text descriptions.
  - Basic events (initiate the leg of the tree) use circles with text.
  - Gates, as previously stated, define logic using symbols with origins in electronic circuit diagrams.

Unlike Failure Modes Effects and Criticality Analysis (FMECA) not all failures of a system apply under FTA – ONLY failures, and sometimes non-failure events, THAT CONTRIBUTE to the occurrence of the top event apply.

Fault Tree Analysis Basics

- Open source government agency handbooks (Nuclear Regulatory Commission and NASA) define FTA as an eight-step process
  1. Identify the objective for the FTA.
  2. Define the top event of the Fault Tree (FT).
  3. Define the scope of the FTA.
  4. Define the resolution of the FTA.
  5. Define ground rules for the FTA.
  6. Construct the FT.
  7. Evaluate the FT.
  8. Interpret and present the results.

- FTA is a process with a managed, structured approach, and feedback.
Terminology and Symbols

- Event – failures, states, operations, errors (human too)
- Gate – logic, what is required to “go through” the gate
- Transfer – within the tree and from outside the tree

Fault Tree Construction

- Basic example of a Fault Tree
- Top Event requires at least one of two lower events to occur,
- Intermediate event “B and C Fails” requires both inputs to occur

Fault Tree Construction

- Construction not just about laying out the symbols
- Some rules apply in the methodology
- There are 8 basic rules that apply
  1. Immediate, necessary, and sufficient
  2. Clear statements
  3. System or component fault?
  4. No miracles/no magic in the logic
  5. Complete the gate before deducing down further
  6. No “gate to gate” logic
- There are other rules that are helpful including some that must be developed under a ground rules and assumptions specific to a particular FTA effort

Terminology and Symbols

- The event symbols and usage are fairly simple and intuitive
  - “Top Event”, a box/rectangle. There is only one.
  - Description of what is under analysis, the final event caused by all the others
  - Getting description right requires deliberate thought. This is the seed of the logical deduction.
  - Intermediate Events, also a box/rectangle. There are many. Good descriptions still apply.
  - Basic Events, circles. Initiate the logic flow up to the top event. Something has to be first. Deduced downward to a point where have a “basic” understanding of what initiates the path to the top event
Fault Tree Construction

- “Immediate, necessary, and sufficient” rule
  - Important criterion to ensure small steps and no jumping in the logic flow for the top event to occur
  - Immediate refers to what occurs one-step prior to the higher event,
    - Central to a “one-step at a time” progression
    - Described as “closest in space, time, and derivation”
  - Necessary means without it, the higher event does not occur... basically got to have it to get what comes next
  - Sufficient refers to nothing else needed, inputs described are enough on their own

Fault Tree Construction

- “Clear statement” rule
  - Some relation to law where legal statutes must be
    - Plain language
    - Straightforward and clear in meaning
    - No ambiguity or vagueness in the description
  - No doubt to what and when
  - Example
    - Valve failed, not a clear statement
    - Valve failed open a clear statement

Fault Tree Construction

- “System or component fault” rule
  - About classifying for greater clarity in understanding events
  - FTA looks at relationships in systems, as a result not all faults are failures but all failures are faults
    - System faults can result in occurrence of top event even when no components have failed
    - When components do fail, component faults are present resulting in system faults leading to occurrence of top event
  - Usually involves just asking the question, by default a “No” to the question of an event consisting of component failure means system fault

Fault Tree Construction

- “No miracles/no magic” rule
  - Refers to excluding any concept of an operator or any other type of interaction doing something extraordinary that prevents occurrence of the top event at any point in the progression
  - No “saves”
  - Normal operation excludes factors of extraordinary skill
  - Does not mean that human error is also excluded
    - Only exceptional performance above normal excluded
    - Human error, lack of skill less than normal performance should be considered whenever operator plays a role

Fault Tree Construction

- “Complete the gate” rule
  - Construction is one-step at a time
  - At each step, deduction must be complete satisfying all previous rules, especially “immediate, necessary, and sufficient”
  - Deducing downward from the top event, each gate and all inputs required for the gate to produce the output must be completed before deducing any lower events

Fault Tree Construction

- “No gate-to-gate” rule
  - Input/output and cause-to-effect is what brings lower events “through the gate” to higher events
  - Gates define the logic requirements connecting the events
  - Gates do not and cannot connect to other gates, gates are a language of logic
    - AND AND or OR OR not logical
    - Event A AND Event B required for Event C is logical
Math Behind the Analysis

- **Boolean Algebra**
  - Instead of numerical values for variables, variables are numerical representation of TRUE/FALSE logic
    - TRUE = 1
    - FALSE = 0
  - Boolean variables are representations of whether or not event occurs
  - Logical relationship in AND or OR gate defines how probability sets for lower events interact for top event to occur

- **AND gate and OR gate logic foundational**
  - **AND gate**
    - Models a redundant case equivalent to parallel block diagram
    - As long as at least one works, no fault progression "through the gate"
    - Both fail, fault progresses "through the gate"
  - **OR gate**
    - Models a series case and single path block diagram
    - All work, no fault progression "through the gate"
    - Any failure, even just one, and fault progresses "through the gate"

- **OR gate represents union of the input events**
  - Q represents the output for two event inputs, A being one input and B the second input
  - The Boolean expression for the output is the union of A and B or Q = A \cup B
  - Boolean algebra uses equivalent symbols, a union is "\+" to show the union of the Boolean variables, therefore Q=A+B
  - This does not mean both are required, only that the occurrence of Q can result from all the probabilities of A and B. Meaning A occurs but not B, B occurs but not A, and A and B both occurring results in Q.

- **AND gate represents the intersection of the input events**
  - Q represents the output for two event inputs, A being one input and B the second input
  - The Boolean expression for the output is the intersection of A and B or Q= A \cap B
  - Boolean algebra uses equivalent symbols, an intersection is "\*" to show the intersection of the Boolean variables, therefore Q=A*B
  - This does mean both are required, the occurrence of Q can result ONLY from the probabilities of A and B both occurring.

- **Increasing the inputs to the gates, that is more than two, results in an expansion of the Boolean equation**
  - Each additional input expands the union for OR gates and additional inputs required in the intersection. As a result
    - For the OR gate, Q=A+B+C …..
    - For the AND gate, Q=A*B*C …..
  - Probability relationships generally follow the same underlying logic as Boolean operators
Fault Tree Evaluation

- Once constructed, the events and logic leading to the occurrence of the top event are mapped
- To make decisions, an evaluation of the interactions of the logic and events is required
- Evaluation comes in two basic forms
  - Qualitative looks at composition—identifying minimal cut sets
  - Quantitative looks at proportion—identifying probability of occurrence and contribution to occurrence of the top event
- Qualitative and Quantitative analytics are complimentary in FTA, each providing information that completes each, working in concert
- Some illustration is required to demonstrate the concepts and how they relate

Fault Tree Evaluation - Qualitative

- How do I determine minimal cut sets?
- First requirement, obviously, is the graphical model of the Fault Tree itself
- To illustrate the concept of qualitative analysis, take an example of a simple system
  - Block diagram related to but not part of FTA, helpful in understanding system
  - Define a top event
  - Deduce to a basic event, a point where we understand enough to state where the progression to the top event starts
  - Apply one of the methods (there are several) to identify minimum cut sets

Fault Tree Evaluation - Qualitative

- Logical deduction from the top event results in the tree shown
- Input/output relationships for each component of the system are represented
- System architect defines which logic gates apply and where
- Cut sets and some minimal cut sets are intuitively obvious
- Keep in mind this is a simple system and simple tree

Fault Tree Evaluation - Qualitative

- Qualitative analysis is principally about identifying the minimum combinations of events that result in occurrence of the top event
- These are the “keys” to getting through the gates from the basic events, through the intermediate events, and ultimately to the top event
- The “cut set” term implies cutting a path through the gates to the top event
- A “minimal cut set” infers finding the short cut or a better description the least number of “keys” to open all the gates on the path

Diagram of a simple redundant hydraulic system
- FTA is about making decisions for an objective
- Example objective to evaluate design changes to make sure hydraulic pressure is always available
- Top event description is “No hydraulic pressure from the system”
### Fault Tree Evaluation - Qualitative

- The naming and number convention applied to the fault tree itself.
- Evaluation relates to the structure can be approached top down or bottom up also flows left or right.
- For MOCUS, the steps are top down and move left to right.
- Algorithm “backs” through the gates.

### Fault Tree Evaluation - Qualitative

- MOCUS steps down through the tree substituting input events as it backs down through the gates.
- At each step of the process, depending on whether an AND or OR gate, input events to the gates are listed as follows:
  - AND gate events listed on the same line separated by comma.
  - OR gate events listed on separate lines.
- Steps are documented in a Matrix.

### Fault Tree Evaluation - Qualitative

- Once (2,3) is removed, the list of minimum cut sets becomes: (1), (4,6), (4,7), (5,6), and (5,7).
- Note that as the reduction and substitution process is completed, there is another impact from the specific gate logic.
  - AND gates result in larger cut sets because more inputs are required 'as a set' to get the output 'through the gate' to the top event.
  - OR gates result in more cut sets because more options for inputs means more sets with combinations that get 'through the gate' to the top event.

### Fault Tree Evaluation - Qualitative

- MOCUS is one method for identifying minimal cut sets, other algorithms use similar methods focusing on minimal cut set identification.
- Binary Decision Diagrams (BDDs) are another method, different from the minimal cut set method.
  - Fault tree outcomes are binary events, event occurs or does not occur (failure or non-failure).
  - BDDs are graphical representations of the Fault Trees binary data structure.
- More info in the NASA FTA Handbook and other references on the methods.

### Fault Tree Evaluation - Quantitative

- Quantitative methods calculate the top event and intermediate event probabilities.
- Importance measures for individual events or combinations of events that comprise minimum cut sets can also be calculated.
- Since minimum cut sets are one of the entities of the quantitative analysis, qualitative analysis of minimum cut sets is required to support quantification.
Fault Tree Evaluation - Quantitative

- Quantitative methods are probability calculations
- Calculating the exact probability for occurrence of the top event, the number of terms is substantial even for a small number of minimal cut sets
- Even for smaller, simpler trees the probability math can get quite complicated and unmanageable for hand calculations
- It becomes quickly apparent where software is an invaluable enabler for quantification

<table>
<thead>
<tr>
<th>Set ID</th>
<th>Boolean Expression</th>
<th>Probabilities</th>
<th>MCO (100 hours)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td>0.40</td>
<td>0</td>
</tr>
<tr>
<td>B</td>
<td>A or B and C or D</td>
<td>0.40</td>
<td>0.50</td>
</tr>
<tr>
<td>C</td>
<td>not A or B or D</td>
<td>0.20</td>
<td>0.50</td>
</tr>
<tr>
<td>D</td>
<td>A or B or C or D</td>
<td>0.80</td>
<td>0.50</td>
</tr>
<tr>
<td>E</td>
<td>A or B or C or D</td>
<td>0.80</td>
<td>0.50</td>
</tr>
</tbody>
</table>

For organization, a table is used and set IDs assigned
- The Boolean expressions and probability are listed for illustration
- Applying the basic event probabilities shows differences resulting from the basic event probability input for the two motors being different

Fault Tree Evaluation - Quantitative

- For organization, a table is used and set IDs assigned
- The Boolean expressions and probability are listed for illustration
- Applying the basic event probabilities shows differences resulting from the basic event probability input for the two motors being different

$P = P_A + P_B + P_C - (P_A P_B + P_A P_C + P_B P_C) + (P_A P_B P_C + P_A P_C P_D)$

Fault Tree Evaluation - Quantitative

- Other approximations offer even more simplified formulas with still fewer terms
- For all approximations, complication is reduced without sacrificing any significant measure of accuracy
- One method models only the upper bound of the probability for occurrence of the top event
- For the simple redundant hydraulic system five cut sets, the approximation becomes:
  $P = 1 - (1 - P_A)(1 - P_B)(1 - P_C)(1 - P_D)(1 - P_E)$
- The approximation yields a top event probability of occurrence estimate resulting from the five cut sets of $P_{TE} = 0.1982$

Fault Tree Evaluation - Quantitative

- Once probability of the top event is determined, other quantitative relationships can be established
- One of those is the importance measure, whereby a ratio represents a measure of the portion of contribution to occurrence of the top event
- For the minimum cut sets of the simple redundant hydraulic system, the minimum cut set with the manifold has very little importance in occurrence of the top event while the two sets with motor 1 have the most

<table>
<thead>
<tr>
<th>Minimum Cut Set</th>
<th>Importance Measure (Ratio of Probabilities)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(A)</td>
<td>0.645</td>
</tr>
<tr>
<td>(B)</td>
<td>0.317</td>
</tr>
<tr>
<td>(C)</td>
<td>0.317</td>
</tr>
<tr>
<td>(D)</td>
<td>0.317</td>
</tr>
<tr>
<td>(E)</td>
<td>0.317</td>
</tr>
</tbody>
</table>
Fault Tree Evaluation - Quantitative

- Importance measures can be further reduced from minimum cut sets to the component level
- Motor 1 had a probability of failure that was twice that of motor two, intuitively one may conclude that motor 1 is twice as important as motor 2
- However, it is the interaction of motor 1 with the logic that defines occurrence of the top event that determines importance, not probability of failure or failure rate of the component
- Applying a method of measuring importance called Fussel-Vesely shows that interaction results in motor 1 being 4X more important than motor 2

Why and When to Apply FTA

- FTA identifies system/product reliability or safety issues, as a resultant the design and redesign for reliability cycle is one of the phases of the life cycle in which FTA is applied
- Another point of application is failure investigation, FTA provides a means to document evaluate, manage, and track potential root causes of failure events
- FTA is a preferred method when safety is a primary concern, especially when very specific events that result in exposure to risk can be defined
- FTA is a very useful tool for many different objectives at any point in the life cycle

Understanding the Rules and Limitations of FTA

- FTA is an analytical method, as such disciplined application and detailed knowledge of the method are required
- Like the systems FTA models, FTA itself is a designed result. There is nothing routine or cookbook about FTA
- Templating FTA is not really feasible in the same sense that FMECA and RCM can be templated
- FTA essentially begins as a clean slate with identification and description of the top event specific to system design and operation

Understanding the Rules and Limitations of FTA

- In many cases, perceived limitations may more be a result of inadequate planning, inadequate resources and/or poor execution
- Anything that follows logic, which is everything in technical fields, can be modeled with FTA
- FTA can model
  - Human error
  - Phasing
  - Maintenance influences
  - Many other aspects of system behavior that leads to occurrence of the top event

Summary

- FTA was developed in response to a need to have greater understanding of event interaction in complex systems
- Intercontinental Ballistic Missiles
- Given what was atop those missiles, the implications for risk require no further explanation
- FTA evolved through application to leading edge technologies
- FTA is a multistep process with analysis performed graphically, qualitatively, and quantitatively.
- There are some rules that apply, we covered only a few
- Further knowledge beyond this tutorial is required, more information is available in the references

Fault Tree Analysis is a valuable and essential tool in multiple applications throughout the product life cycle
References

Books:

Analysis Example Problem 1
- A home break in occurs while the owner is away

Analysis Example Problem 2
- Car will not start

References

Industry Standards
- IEC 61025-2006 Fault Tree Analysis (FTA), International Electrotechnical Commission, Geneva, Switzerland. Available at www.iec.ch
- SAE Aerospace Recommended Practice (ARP) 926B, "Fault/Failure Analysis Procedure," SAE, June 1997

References

DoD (Open Source, free)

References

Government Agency (Open source, free):
- "Fault Tree Analysis Application Guide," Reliability Information Analysis Center
Questions?