

Fault Tree Analysis

Table of Contents

Introduction.....	3
Fault Tree Analysis.....	3
Basic Events.....	4
Advantages.....	4
Limitations	4
Notation.....	5
General Procedure for Fault Tree Analysis.....	6
Rules of Fault Tree Construction	7
Considerations	11
Fault Tree Evaluation	12
Boolean Algebra	12
<i>The OR Gate.....</i>	<i>12</i>
<i>The AND Gate.....</i>	<i>12</i>
Qualitative Analysis.....	14
<i>Minimal Cut Sets.....</i>	<i>14</i>
<i>Criticality.....</i>	<i>14</i>
Quantitative Analysis.....	16
<i>Common-Cause Failures.....</i>	<i>16</i>
References.	18

Introduction

- o there is a need to analyze all the possible **failure** mechanisms in complex systems (e.g. nuclear power plants)
- o also perform probabilistic analyses for the expected rate of failures
- o estimate probabilities of events that are modelled as logical combinations or logical outcomes of other random events
- o two main methods:
 - **fault tree** analysis
 - **event tree** analysis
- o **decision trees** also exist and are used in **risk** analysis (combines all feasible alternatives, possible outcomes and their probabilities, monetary consequences and utility evaluations)
- o other graphical methods include
 - reliability block diagrams
 - functional logic diagrams
 - Failure Modes and Effects Analysis (FMEA)

Fault Tree Analysis

A technique by which many events that interact to produce other events can be related using simple logical relationships.

- o a good reference is NRC "The Fault Tree Handbook"
(<http://www.nrc.gov/reading-rm/doc-collections/nuregs/staff/sr0492/sr0492.pdf>)
- o also Chapter 8 and 10 in *McCormick* (1981)
- o one of the principal methods of probabilistic safety (or risk) analysis (PRA)
- o developed by Bell Telephone Laboratories in 1962 for the U.S. Air Force Minuteman system, later adopted and extensively used by Boeing Company
- o fault tree diagrams
 - are used most often as a system-level risk assessment technique
 - can model the possible combinations of equipment failures, human errors, and external conditions that can lead to a specific type of accident
 - follow a top-down structure and represent a **graphical** model of the pathways within a system between **basic events** that can lead to a foreseeable loss event (or a failure) referred to as the **top event**
- o the contributory events and conditions are interconnected using standard logic symbols (AND, OR, etc.), also referred to as **gates**
- o events that must **coexist** to cause the top event are described using the **AND** relationship
- o alternate events that can **individually** cause the top event are described using the **OR** relationship

- 0 the occurrence of a top event may or may not lead to a serious or adverse **consequence**
- 0 the relative likelihood of a number of potential consequences will depend on the conditions or subsequent events that follow
- 0 potential consequences can be systematically identified using an **event tree**

Basic Events

An event that cannot be developed any further

- 0 all basic events are generally assumed to be **statistically independent** unless they are **common cause failures** (i.e. failures arising from a common cause or an initiating event)
- 0 basic events can be either
 - **primary** fault events, i.e. subsystem failure due to a basic mode such as a structural fault, failure to open or close, or to start or stop, or
 - **secondary** fault events, i.e. subsystem failure due to excessive operational or environmental stress resulting in the system element to be out of tolerance



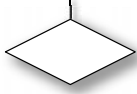
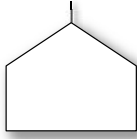
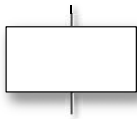
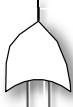
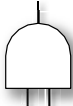
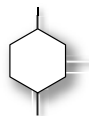


Advantages

- 0 allow the use of reliable information on component failure and other basic events to estimate the overall risk associated with new system designs for which no historical data exists
- 0 simple to understand and easy to implement
- 0 **qualitative** descriptions of potential problems and combinations of events causing specific problems of interest
- 0 **quantitative** estimates of failure frequencies and likelihoods, and relative importances of various failure sequences and contributing events
- 0 lists of recommendations for reducing risks
- 0 quantitative evaluations of recommendation effectiveness

Limitations

- 0 difficult to conceive all possible scenarios leading to the top event
- 0 construction of fault trees for large systems can be tedious
- 0 correlations between basic events (e.g. failure of components belonging to the same batch) are difficult to model and exact solutions to correlated events do not exist
- 0 subjective decisions regarding the level of detail and completeness are often necessary

Notation

Symbol	Name	Description
<u>Primary Event Symbols</u>		
	Circle	Basic Event – a basic initiating fault requiring no further development
	Oval	Conditioning Event – specific conditions or restrictions that apply to any logic gate (used with INHIBIT gate)
	Diamond	Undeveloped Event – an event that is not developed further because it is of insufficient consequence or because information is unavailable
	House	External Event – an event which is normally expected to occur (not a fault event)
<u>Intermediate Event Symbols</u>		
	Rectangle	A fault event that occurs as a result of the logical combination of other events
<u>Gate Symbols</u>		
	OR Gate	The union operation of events, i.e. the output event occurs if (at least) one or more of the inputs occur
	AND Gate	The intersection operation of events, i.e. the output event occurs if and only if all the inputs occur
	INHIBIT Gate	The output event occurs if the (single) input event occurs in the presence of an enabling condition (i.e. Conditioning Event (oval) drawn to the right of the gate)
<u>Transfer Symbols</u>		
	Triangle-in	Indicates that the tree is developed further someplace else (e.g. another page)
	Triangle-out	Indicates that this portion of the tree is a sub-tree connected to the corresponding Triangle-In (appears at the top of the tree)

General Procedure for Fault Tree Analysis

- 0 from the U.S. Coast Guard Risk-based Decision-making Guidelines, Vol. 3
- Risk Assessment Tools Reference, Chapter 9 – Fault Tree Analysis (FTA)
(<http://www.uscg.mil/hq/g-m/risk/E-Guidelines/RBDMGuide.htm>)
- Step 1.* Define the **system** of interest.
Specify and clearly define the boundaries and initial conditions of the system for which failure information is needed.
- Step 2.* Define the **top event** for the analysis.
Specify the problem of interest that the analysis will address (e.g. a specific quality problem, shutdown, safety issue, etc.).
- Step 3.* Define the **treetop** structure.
Determine the events and conditions (i.e., intermediate events) that most directly lead to the top event.
- Step 4.* Explore each branch in successive levels of **detail**.
Determine the events and conditions that most directly lead to each intermediate event. Repeat the process at each successive level of the tree until the fault tree model is complete.
- Step 5.* Solve the fault tree for the **combinations** of events contributing to the top event.
Examine the fault tree model to identify all the possible combinations of events and conditions that can cause the top event of interest. A combination of events and conditions *sufficient* and *necessary* to cause the top event is called a **minimal cut set**.
- Step 6.* Identify important **dependent** failure potentials and adjust the model appropriately (qualitative common cause failure analysis).
Study the fault tree model and the list of minimal cut sets to identify potentially important dependencies among events. Dependencies are single occurrences that may cause multiple events or conditions to occur at the same time.
- Step 7.* Perform **quantitative analysis** (if necessary).
Use statistical characterizations regarding the failure and repair of specific events and conditions in the fault tree model to predict future performance for the system.
- Step 8.* Use the results in **decision making**.
Use results of the analysis to identify the most significant vulnerabilities in the system and to make effective recommendations for reducing the risks associated with those vulnerabilities.

Rules of Fault Tree Construction

- o a fault tree should only be constructed once the functioning of the entire system is fully understood
- o objective is to identify all the component failures, or combinations thereof that could lead to the top event (Steps 2 - 4 above)
- o after *McCormick* (1981)

Rule 1. State the fault event as a fault, including the description and timing of a fault condition at some particular time.

Include

- (a) what the fault state of that system or component is,
- (b) when that system or component is in the fault state.

Test the fault event by asking

- (c) is it a fault?
- (d) is the what-and-when portion included in the fault statement?

Rule 2. There are two basic types of fault statements, state-of-system and state-of-component.

To continue the tree,

- (a) if state-of-system fault statement, use Rule 3
- (b) if state-of-component fault statement, use Rule 4

Rule 3. A **state-of-system** fault may use an AND, OR, or INHIBIT gate or no gate at all.

To determine which gate to use, the faults must be then

- (a) minimum necessary and sufficient fault events,
- (b) immediate fault events.

Rule 4. A **state-of-component** fault always uses an OR gate.

To continue, look for the primary, secondary, and command failure fault events. Then state those fault events.

- (a) **primary failure** is failure of that component within the design envelope or environment
- (b) **secondary failures** are failures of that component due to excessive environments exceeding the design environment
- (c) **command faults** are inadvertent operation of the component because of a failure of a control element

Rule 5. No gate-to-gate relationships.

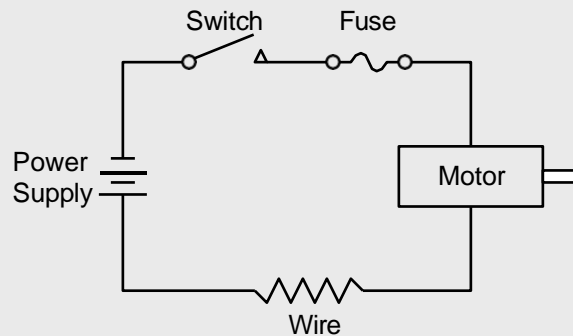
Put an event statement between any two gates.

Rule 6. Expect **no miracles**.

Those things that would normally occur as the result of a fault will occur, and only those things. Also, normal system operation may be expected to occur when faults occur.

- Rule 7.* In an OR gate, the input does not cause output.
If any input exists, the output exists. Fault events under the gate may be a restatement of the output events.
- Rule 8.* An AND gate defines a **causal** relationship.
If the input events coexist, the output is produced.
- Rule 9.* An INHIBIT gate describes a causal relationship between one fault and another, but the indicated condition must be present.
The fault is the direct and sole cause of the output when that specified condition is present. Inhibit conditions may be faults or situations, which is why AND and INHIBIT gates differ.

Example: (McCormick, 1981) Construct a fault tree for the simple electric motor circuit shown below.



Solution:

Step 1. Define the system of interest.

Need to identify

- *Intended Functions*
- *Physical Boundaries* (to avoid overlooking key elements of a system at interfaces and penalizing a system by associating other equipment with the subject of the study)
- *Analytical Boundaries* (to limit the level of analysis resolution, to explicitly exclude certain types of events and conditions, such as sabotage, from the analysis)
- *Initial Conditions*, (including equipment that is assumed to be out of service initially, which affect the combinations of additional events necessary to produce a specific system problem)

For this particular problem we have,

- | | |
|-----------------------|---|
| Intended Function | – the motor is used for some (unknown) purpose |
| Physical Boundaries | – power supply |
| Analytical Boundaries | – include all contributors in the above diagram |
| Initial Conditions | – switch closed, motor on |

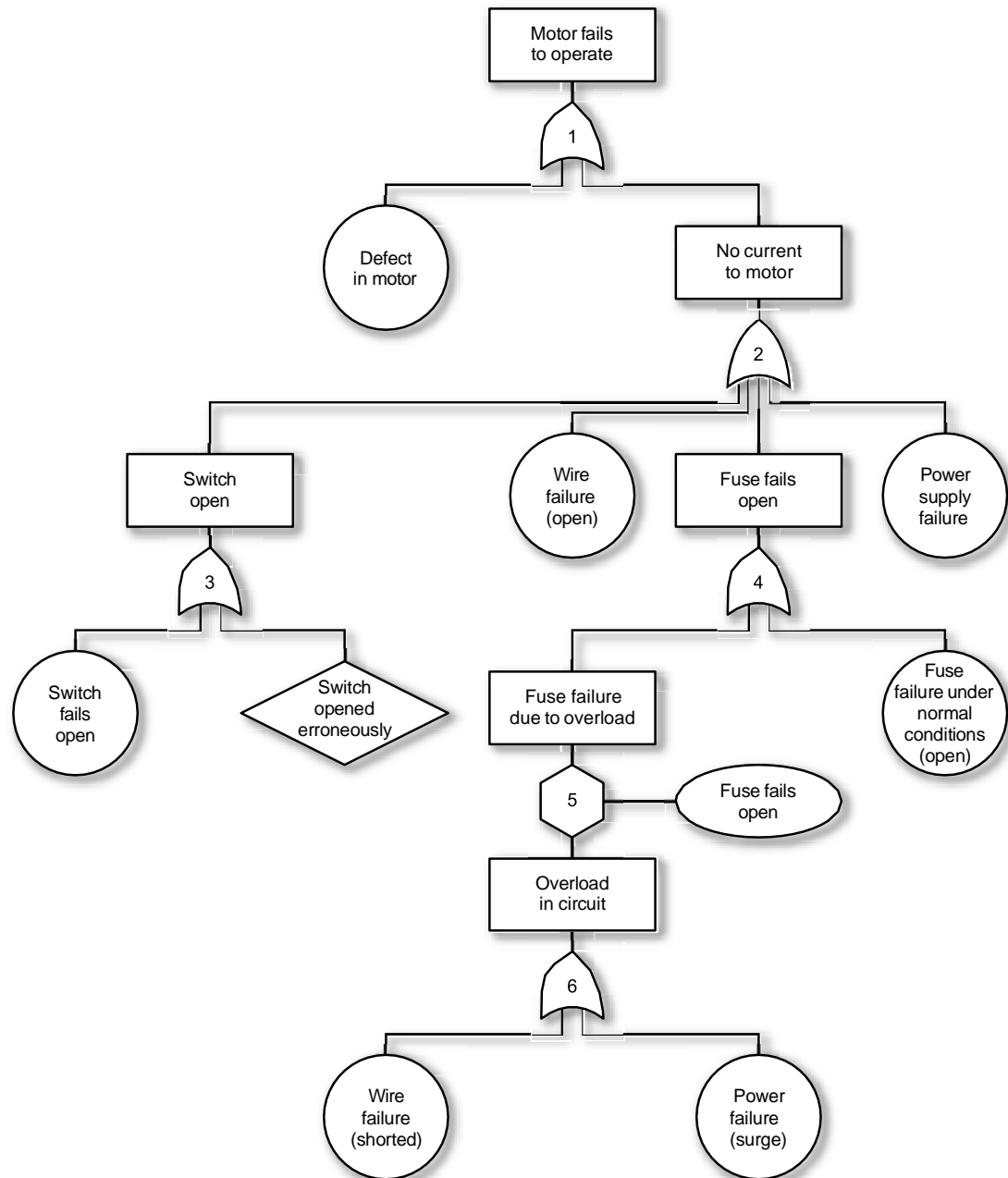
Step 2. Define the top event.

We are interested in the event that the motor fails to operate. Therefore, the top event is defined as

Motor fails to operate

Step 3. Construct the fault tree, starting from the top, i.e., define the treetop structure. Identify the main contributing events, including all events and scenarios that may cause the top event.

Step 4. Explore each branch in successive levels of detail, following the rules of fault tree construction.



Fault Tree Construction

Gate 1. One primary failure event is the failure of the motor itself (for example, due to a wiring failure within the motor or loss of lubrication to the bearings). This event is a basic event because no details of the motor are given, therefore, the event cannot be developed further. The other possibility is the event that no current is supplied to the motor.

Gate 2. The event “No current in motor” is the result of other events and is therefore developed further. The lack of current to the motor can result from a broken connection in any of the other four components in the circuit, including the failure of the wire or power supply (basic events), the switch being open, or failure of the fuse.

Gate 3. The open switch may be due to a basic failure of the switch, or the event that the switch was opened erroneously. The erroneous opening of the switch is due to human error, which could be developed further into more basic events (i.e. operator is inexperienced, under stress, etc.). However, due to insufficient information, the event is not explored further. This purposely undeveloped event is therefore denoted with the diamond symbol.

Gate 4. The fuse failure event may be caused by fuse failure under normal conditions (primary failure) or due to overload from the circuit.

Gate 5. The secondary fuse failure can occur if the fuse does not open every time an overload is present in the circuit (because all conditions of an overload do not necessarily result in sufficient overcurrent to open the fuse). This is why a conditional gate, denoted by the hexagon, is used. The condition, i.e. “Fuse fails open” is placed in the connecting oval, and the conditional gate is treated similarly to an AND gate in subsequent tree analysis.

Gate 6. The overload in the circuit may be caused either by a short or a power surge, both of which are primary (i.e. basic) events.

Considerations

- 0 construction of a fault tree is **subjective**
- 0 need to take into account
 - **Level of Detail** – the number of basic events should be defined such that the size of the tree is reasonable with respect to the scope of the analysis
 - **Probability Assignment** – need to stop development at the level where probability or failure data is available
 - **Meaningfulness** – the level of detail should be such that the basic and undeveloped events correspond to the design aspects being analyzed