# TPK4120 - Lecture summary

Jørn Vatn
eMail: jorn.vatn@ntnu.no

Updated 2022-09-07

## Chapter 4 - Qualitative System Analysis

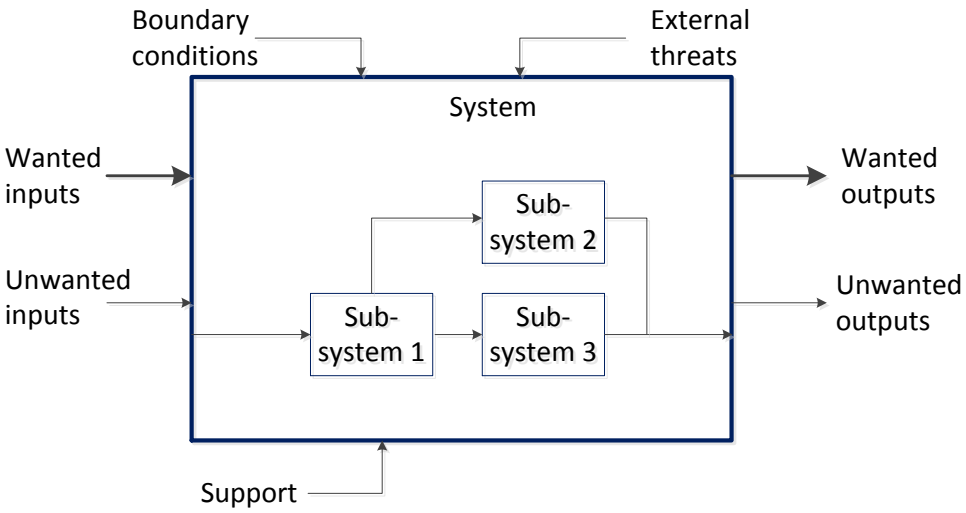
Methods to be covered in this memo are:

- FMEA/FMECA: Failure modes, effects, and (criticality) analysis
- FTA: Fault tree analysis
- ETA: Event tree analysis
- RBD: Reliability block diagram
- BBN: Bayesian Belief Networks

Through this chapter a bicycle is used as a motivating example. A drawing of a bicycle may be found here: `http://upload.wikimedia.org/wikipedia/commons/8/8a/Bicycle_diagram-en.svg`.

### System and interfaces

In the analysis of technical systems it is important to define the interfaces to humans and other framing conditions for the system to operate. The following figure shows how a system interfere with it's interfaces:




In the following some examples related to the bike are given:

- System: The bike
- Wanted input: The rider, or more specifically the force he puts on the pedals, his operation of the gears, breaks etc
- Unwanted inputs: A too aggressive biker
- Boundary conditions: Can not ride on the highway, has to follow speed limits if riding on a road
- External threats: Cars
- Support: Maintenance
- Wanted outputs: A to B
- Unwanted outputs: Accidents, delayed in going from A to B

## Functional analysis

In order to identify system and component failures, we first have to identify the functions of a system. Functional analysis basically includes:

- Identify all the functions of the system
- Identify the functions required in the various operational modes of the system (for example when parking the bike, one needs a lock)
- Provide a hierarchical decomposition of the system functions
- Describe how each function is realized

- Identify the interrelationships between the functions
- Identify interfaces with other systems and with the environment

A function is basically described by a verb and a noun, e.g., “Pump water”, “Close flow” and “Cut wood”. To help identifying functions it is valuable to categorize function, and main categories are:

- Essential functions (e.g., pump water)
- Auxiliary functions (e.g., contain water - prevent leakage out)
- Protective functions (e.g., prevent sparks from electro-motor, (Ex-protection))
- Information functions (e.g., measure and provide internal pressure, temperature)
- Interface functions (e.g., connect to in/out pipes)
- Superfluous functions (e.g., functions remaining after the system has been modified)

There are many ways to structure functions, and a popular approach is the structured analysis and design technique (SADT) covered in Chapter 2.

### Failure Modes, Effects, and Criticality Analysis

Failure Mode and Effects Analysis (FMEA) was one of the first systematic techniques for failure analysis. It was developed by reliability engineers in the late 1950’s to determine problems that could arise from malfunctions of military systems. A Failure Mode and Effects Analysis is often the first step in a systems reliability study. It involves reviewing as many components, assemblies and subsystems as possible to identify possible failure modes and the causes and effects of such failures. For each component, the failure modes and their resulting effects on the rest of the system are written onto a specific FMEA form. There are numerous variations of such forms. An example of an FMEA form is shown below.

A Failure Mode and Effects Analysis is mainly a qualitative analysis, which is usually carried out during the design stage of a system. The purpose is then to identify design areas where improvements are needed to meet the reliability requirements. The Failure Mode and Effect Analysis can be carried out either by starting at the component level and expanding upwards (the “bottom up” approach), or from the system level downwards (the “top down” approach). The component level to which the analysis should be conducted is often a problem to define. It is often necessary to make compromises since the workload could be tremendous even for a system of moderate size. It is, how¬ever, a general rule to expand the analysis down to a level at which

failure rate estimates are available or can be obtained. Most Failure Mode and Effects Analyses are carried out according to the "bottom-up" approach. One may, however, for some particular systems save a considerable amount of effort by adopting the "top down" approach. With this approach, the analysis is carried out in two or more stages. The first stage is an analysis on the functional block diagram level. The possible failure modes and failure effects of each functional block are identified based on knowledge of the block's required function, or from experience on similar equipment. One then proceeds to the next stage, where the components within each functional block are analysed. If a functional block has no failure modes which are critical, then no further analysis of that block needs to be performed. By this screening, it is possible to save time and effort. A weakness of this "top down" approach lies in the fact that it is not possible to ensure that all failure modes of a functional block have been identified.

An FMEA becomes a Failure Modes, Effects and Criticality Analysis (FMECA) if practicalities or priorities are assigned to the failure mode effects.

More detailed information on how to conduct a Failure Mode and Effects Analysis (and an FMECA) may be found in:

- MIL-STD 1629 "Procedures for performing a failure mode and effect analysis"
- IEC 60812 "Procedures for failure mode and effect analysis (FMEA)"
- SAE ARP 5580 "Recommended failure modes and effects analysis (FMEA) practices for non-automobile applications"
- SAE J1739 "Potential Failure Mode and Effects Analysis in Design (Design FMEA) and Potential Failure Mode and Effects Analysis in Manufacturing and Assembly Processes (Process FMEA) and Effects Analysis for Machinery (Machinery FMEA)"

**FMECA procedure**

1. FMECA prerequisites
2. System structure analysis
3. Failure analysis and preparation of FMECA worksheets
4. Team review
5. Corrective actions

Important aspects of FMECA prerequisites are:

1. Define the system to be analysed in terms of (a) System boundaries (which parts should be included and which should not), (b) Main system missions and functions (incl. functional requirements), and (c) Operational and environmental conditions to be considered

2. Collect available information that describes the system to be analysed; including drawings, specifications, schematics, component lists, interface information, functional descriptions, and so on

3. Collect information about previous and similar designs from internal and external sources; including FRACAS (Failure reporting, analysis, and corrective action system) interviews with design personnel, operations and maintenance personnel, component suppliers, and so on.

Various methods for the system structure analysis exist. An SADT analysis may be a good starting point.

A suitable FMECA worksheet for the analysis has to be decided. In many cases the client (customer) will have requirements to the worksheet format - for example to fit into his maintenance management system. A sample FMECA worksheet covering the most relevant columns is given below.

FMECA

System: Performed by:
Subsystem: Date:
Function: Page:

| DESCRIPTION OF UNIT | | | DESCRIPTION OF FAILURE | | | EFFECT OF FAILURE | | | FAILURE RATE | CRITICALITY | CORRECTIVE ACTION | REMARKS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IDENTI - FICATION | OPERATIONAL MODE | FUNCTION | FAILURE MODE | FAILURE MECHANISM | HOW TO DETECT | LOCAL | SYSTEM | OPERAT . STATUS | | | | |
| | | | | | | | | | | | | |

In the textbook important columns are discussed. Some comments to be highlighted:

### Operational mode

Example of operational modes are: idle, standby, and running. Operational modes for an air plane include, for example, taxi, take-off, climb, cruise, descent, approach, flare-out, and roll. Also note that operational mode at the system level is not the same as operational mode at the component level.

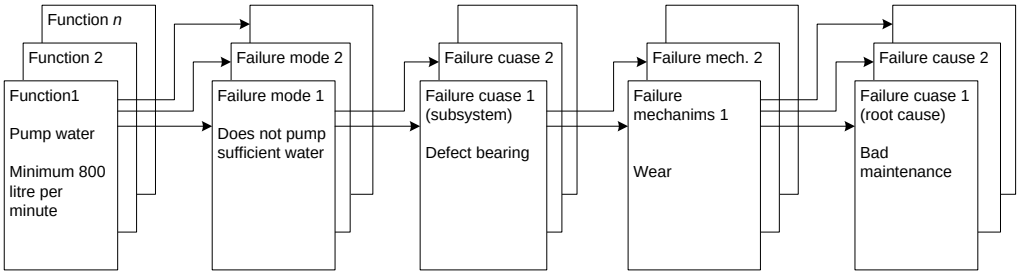
### Failure mechanisms and failure causes

*Failure mechanisms* relates to physical, chemical or other processes that deteriorates the entity, and leads to a failure The term “failure cause” is often used in two different ways:

- Proximate cause, e.g., failure on a lower level in the system hierarchy such as a defect bearing in a pump

- Root cause, for example bad maintenance, inadequate design etc.

The following figure illustrates the relation between function, failure mode, failure cause and failure mechanism:




### Hidden versus evident failures

We often distinguish between hidden and evident failures. The term "hidden" often relates to entities that is not continuously demanded. For example the SIFA valve on a train (bleed of the air pressure by activation) is a hidden function, and a failure will not be detected automatically. The term "evident" relates to entities that are continuously demanded, and a failure will most likely be detected immediately. Note that the same SIFA-valve will also have a evident function ("not bleed of air pressure under normal operation") because an unintended activation immediately will be detected (breaks are activated).

Example of FMECA:

FMECA

System: Bike
Subsystem: Traction
Function: Convert pedal force from the rider to wheel torque

Performed by: Jørn
Date: Some date
Page: 1

| DESCRIPTION OF UNIT | | | DESCRIPTION OF FAILURE | | | EFFECT OF FAILURE | | | FAILURE RATE | CRITICALITY | CORRECTIVE ACTION | REMARKS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| IDENTI - FICATION | OPERATIONAL MODE | FUNCTION | FAILURE MODE | FAILURE MECHANISM | HOW TO DETECT | LOCAL | SYSTEM | OPERAT . STATUS | | | | |
| Chain | Running | Convert torque from crank to gear | Not converting Uneven movement | Fatigue | Inspection | No gear torque | Bike is not moving | Can't reach lecture today | Low | High | Bring chain lock | |

# FAULT TREE ANALYSIS

A fault tree is a logic diagram that displays the relationships between a potential critical event (accident) in a system and the reasons for this event. The reasons may be environmental conditions, human errors, normal events (events which are expected to occur during the life span of the system) and specific component failures. A properly constructed fault tree provides a good illustration of the various combinations of failures and other events which can lead to a specified critical event. The fault tree is easy to explain to engineers without prior experience of fault tree analysis.

An advantage with a fault tree analysis is that the analyst is forced to understand the failure possibilities of the system, to a detailed level. A lot of system weaknesses may thus be revealed and corrected during the fault tree construction.

A fault tree is a *static* picture of the combinations of failures and events which can cause the TOP event to occur. Fault tree analysis is thus not a suitable technique for analysing dynamic systems, like switching systems, phased mission systems and systems subject to complex maintenance strategies.

A fault tree analysis may be qualitative, quantitative or both, depending on the objectives of the analysis. Possible results from the analysis may e.g. be:

1. A listing of the possible combinations of environmental factors, human errors, normal events and component failures that can result in a critical event in the system.

2. The probability that the critical event will occur during a specified time interval.

Figure 1 shows an example fault tree for the bike.

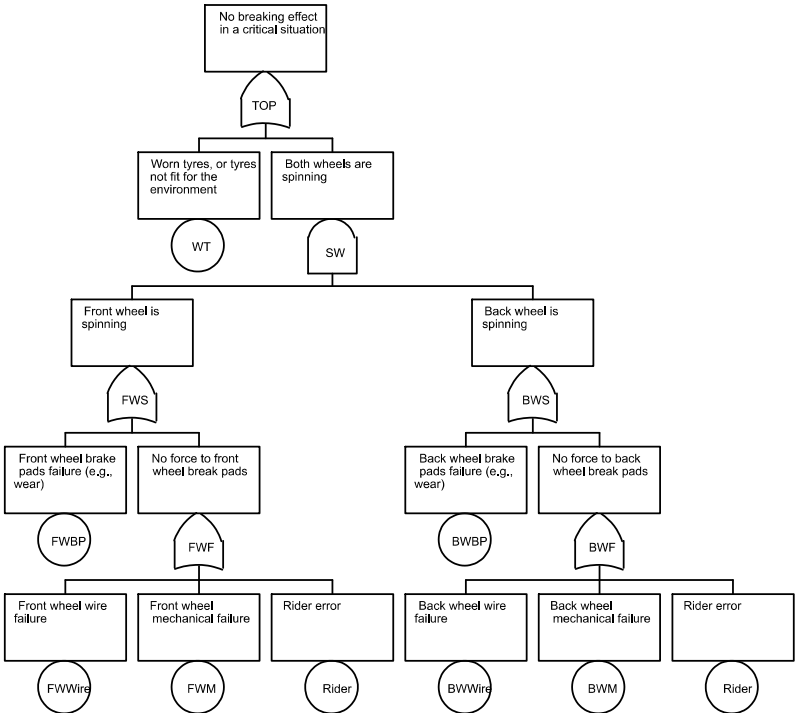



Figure 1: FTA example for a bike

The analysis of a system by the fault tree technique is normally carried out in five steps:

1. Definition of the problem and the boundary conditions.

2. Construction of the fault tree.

3. Identification of minimal cut and/or path sets.

4. Qualitative analysis of the fault tree.

5. Quantitative analysis of the fault tree (Ch 5).

In the following we will present the basic elements of standard fault tree analysis. Then we will conclude this chapter by presenting a numerical example illustrating how the technique could be utilised in relation to maintenance optimisation.

## Fault tree construction

### Fault tree diagram, symbols and logic

A fault tree is a logic diagram that displays the connections between a potential system failure (TOP event) and the reasons for this event. The reasons (Basic events) may be environmental conditions, human errors, normal events and component failures. The graphical symbols used to illustrate these connections are called "logic gates". The output from a logic gate is determined by the input events.

The graphical layout of the fault tree symbols are dependent on what standard we choose to follow.

### Definition of the Problem and the Boundary Conditions

This activity consists of:

1. Definition of the critical event (the accident) to be analysed.

2. Definition of the boundary conditions for the analysis.

The critical event (accident) to be analysed is normally called the TOP event. It is very important that the TOP event is given a clear and unambiguous definition. If not, the analysis will often be of limited value. As an example, the event description "Fire in the plant" is far too general and vague. The description of the TOP event should always answer the questions: **What, where** and **when**?

**What:** Describes what type of critical event (accident) is occurring, e.g., collision between two trains.

**Where:** Describes where the critical event occurs, e.g., on a single track section.

**When:** Describes when the critical event occurs, e.g., during normal operation.

A more precise TOP event description is thus: "Collision between two trains on a single track section during normal operation".

1. To get a consistent analysis, it is important that the *boundary conditions* for the analysis are carefully defined. By boundary conditions we mean: **The physical boundaries of the system**. What parts of the system are to be included in the analysis, and what parts are not?

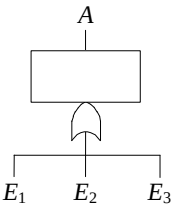
2. **The initial conditions**. What is the operational state of the system when the TOP event is occurring? Is the system running on full/reduced capacity? Which valves are open/closed, which pumps are functioning etc.?

3. **Boundary conditions with respect to external stresses**. What type of external stresses should be included in the analysis? By external stresses we here mean stresses from war, sabotage, earthquake, lightning etc.

4. **The level of resolution**. How far down in detail should we go to identify potential reasons for a failed state? Should we as an example be satisfied when we have identified the reason to be a "valve failure", or should we break it further down to failures in the valve housing, valve stem, actuator etc.? When determining the required level of resolution, we should remember that the detail in the fault tree should be comparable to the detail of the information available

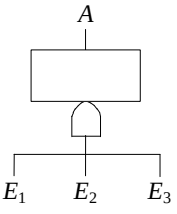
**Construction of the Fault Tree**

The fault tree construction always starts with the TOP event. We must thereafter carefully try to identify all fault events which are the immediate, necessary and sufficient causes that result in the TOP event. These causes are connected to the TOP event via a logic gate. It is important that the first level of causes under the TOP event is developed in a structured way. This first level is often referred to as the TOP structure of the fault tree. The TOP structure causes are often taken to be failures in the prime modules of the system, or in the prime functions of the system. We then proceed, level by level, until all fault events have been developed to the required level of resolution. The analysis is in other words deductive and is carried out by repeated asking "What are the reasons for...?"



OR-gate

The OR-gate indicates that the output event $A$ occurs if any of the input events $E_i$ occurs. In relation to the bike example with TOP event "No breaking effects" the two events: "No friction" and "both wheels spinning" are connected by an OR gate since any of these events will lead to the TOP event.

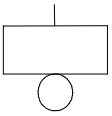


AND-gate

The AND-gate indicates that the output event $A$ occurs only when all the input events $E_i$ occurs simultaneously. In the bike example, "Front wheel is spinning" and "Rear wheel is spinning" are connected by an AND gate, since

both these event have to occur in order to full fill the requirement that both wheels are spinning.

The Basic event represents a basic equipment fault or failure that requires no further development into more basic faults or failures. An example of a basic event in the bike example is “Breakage in break wire”.



Basic event

### Identification of Minimal Cut- and Path Sets

A fault tree provides valuable information about possible combinations of fault events which can result in a critical failure (TOP event) of the system. Such a combination of fault events is called a cut set.

A ***cut set*** *in a fault tree is a set of Basic events whose (simultaneous) occurrence ensures that the TOP event occurs. A cut set is said to be* ***minimal*** *if the set cannot be reduced without loosing its status as a cut set.*

A ***path set*** *in a fault tree is a set of Basic events whose non-occurrence (simultaneously) ensures that the TOP event does not occur. A path set is said to be* ***minimal*** *if the set cannot be reduced without loosing its status as a path set.*

### Qualitative Evaluation of the Fault Tree

A qualitative evaluation of the fault tree may be carried out on the basis of the minimal cut sets. The importance of a cut set depends obviously on the number of Basic events in the cut set. The number of different Basic events in a minimal cut set is called the *order* of the cut set. A cut set of order one is usually more critical than a cut set of order two, or higher. When we have a cut set with only one Basic event, the TOP event will occur as soon as this Basic event occurs. When a cut set has two Basic events, both of these have to occur at the same time to cause the TOP event to occur.

Another important factor is the type of Basic events in a minimal cut set. We may rank the criticality of the various cut sets according to the following ranking of the Basic events:

1. Human error
2. Failure of active equipment
3. Failure of passive equipment

The ranking is based on the assumption that human errors occur more frequently than active equipment failures, and that active equipment is more failure-prone than passive equipment (an active or running pump is for example more exposed to failures than a passive standby pump).

### Identification of Minimal Cut- and Path Sets

A fault tree provides valuable information about possible combinations of fault events which can result in a critical failure (TOP event) of the system. Such a combination of fault events is called a cut set.

*A**cut set** in a fault tree is a set of Basic events whose (simultaneous) occurrence ensures that the TOP event occurs. A cut set is said to be **minimal** if the set cannot be reduced without loosing its status as a cut set.*

*A**path set** in a fault tree is a set of Basic events whose non-occurrence (simultaneously) ensures that the TOP event does not occur. A path set is said to be **minimal** if the set cannot be reduced without loosing its status as a path set.*

In practice only minimal cut sets are used for evaluation of fault trees. To find the minimal cut sets we apply the MOCUS algorithm (Method Of obtaining Cut Sets). The MOCUS algorithm essentially contains the following elements:

1. Start with the TOP event
2. As the algorithm proceeds, the result is stored in a matrix like format of rows and columns
3. AND- and OR-gates are resolved by replacing the gate with it's "children" in the fault tree diagram
4. An AND-gate means that the gate is replaced by new elements for the row(s) it is found
5. An OR-gate means that the gate is replaced by as many rows that the gate has children, where each child is inserted at the position of the OR-gate being replaced
6. When all gates are replaced, we remain with only the basic events, where each row corresponds to a cut set

Note that the the cut sets will not necessarily be be minimal. To make the cut sets minimal we have to:

1. Replace duplicates of one event with only one occurrence of that event in each row
2. If one row is a sub set of another row, then the larger of these two rows (representing non-minimal cut sets) is removed

The MOCUS algorithm is demonstrated here: `http://folk.ntnu.no/jvatn/eLearning/TPK4120/Examples/MOCUS.html` in relation to the example used in the lectures.

### $k$oo$n$ gate

The $k$oo$n$ gate is something "between" the AND and OR gate. A $k$oo$n$ gate occurs if $k$ out of the $n$ inputs occur. Note that in FTA we focus on fault states, i.e., an event occurring means a failure, hence the "voting" in FTA is different from in RBD. To clarify, the following notation is often used:

- $k$oo$n : G$ is used if we consider the functioning of components (G=Good). The system (block) functions if $k$ or more out of the $n$ components are functioning
- $k$oo$n : F$ is used if we consider the fault of components (F=Fault state). The system (gate/TOP event) occurs if $k$ or more out of the $n$ inputs are occurring (i.e., in a fault state)

Note the following relation:

$$k\text{oo}n : G = (n–k+1)\text{oo}n : F$$
$$k\text{oo}n : F = (n–k+1)\text{oo}n : G$$

Consider a system with three pumps each having 50% capacity. The system functions if at least 2 of the pumps are functioning. In an RBD we then use the $2\text{oo}3 : G$ block for this system, and for the FTA we use the $k\text{oo}n : F = n–k+1\text{oo}n : G = 3-2+1\text{oo}3 = 2\text{oo}3$ gate.

If we have 4 such pumps, the RBD representation is $2\text{oo}4 : G$, and in FTA we use the $k\text{oo}n : F = n–k+1\text{oo}n : G = 4-2+1\text{oo}4 = 3\text{oo}4$ gate meaning that 3 or more pumps must be in a fault state in order to give a system failure (TOP event).

Computerized FTA programs will offer the $k\text{oo}n : F$ as part of the drawing palette. For manual construction of a fault tree with a $k\text{oo}n : F$ gate we can use an OR-gate followed by several AND-gates. Each AND-gate is then a subset with $k$ out of the $n$ inputs. There are altogether $\binom{n}{k}$ ways we may choose $k$ inputs out of $n$ inputs, hence we will have $\binom{n}{k}$ AND-gates to put under the OR-gate.

## Qualitative Evaluation of the Fault Tree

A qualitative evaluation of the fault tree may be carried out on the basis of the minimal cut sets. The importance of a cut set depends obviously on the number of Basic events in the cut set. The number of different Basic events in a minimal cut set is called the *order* of the cut set. A cut set of order one is usually more critical than a cut set of order two, or higher. When we have a cut set with only one Basic event, the TOP event will occur as soon as this Basic event occurs. When a cut set has two Basic events, both of these have to occur at the same time to cause the TOP event to occur.

Another important factor is the type of Basic events in a minimal cut set. We may rank the criticality of the various cut sets according to the following ranking of the Basic events:

1. Human error
2. Failure of active equipment
3. Failure of passive equipment

The ranking is based on the assumption that human errors occur more frequently than active equipment failures, and that active equipment is more failure-prone than passive equipment (an active or running pump is for example more exposed to failures than a passive standby pump).

The quantitative analysis will be addressed in Chapter 6.

## Event Tree Analysis (ETA)

### Introduction

An event tree is a logical diagram which displays possible event sequences following a specified critical event in a system. An event tree analysis (ETA) is a method for systematic analysis of a system after a critical event has occurred. The result of an ETA is a list of possible event sequences that follows the initiating event. The critical, initiating event may be a technical failure or some human error. In the development of the event sequences, the effects of possible barriers and safety functions, which are designed to prevent the occurrence of the critical event or reduce the consequences of this event, are taken into account. The analysis is both qualitative and quantitative. The qualitative content is primarily a visualisation of different scenarios (the event tree) with corresponding end consequences, while the quantitative analysis gives frequencies for the different end consequences. Figure 2 shows an ETA example. The initial event could be for example SPAD = Signal passed at danger (obtained from for example an FTA), and then the various barriers are shown as $B_1$, $B_2$ etc. Each barrier has a Y=Yes output and a N=No output.

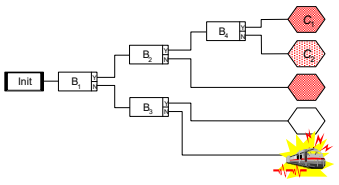



Figure 2: ETA example

## Procedure

The event tree analysis is usually carried out in six steps:

1. Identification of a relevant initiating event (which may give rise to unwanted consequences).
2. Identification of the barriers and safety functions which are designed to prevent the occurrence of the initiating event, or to reduce the con¬sequences of this event.
3. Construction of the event tree.
4. Description of the resulting event sequences.
5. Calculation of probabilities/frequencies for the identified consequences.
6. Compilation and presentation of the results from the analysis.

See the textbook for details regarding each step. As for the fault tree it is important to define an unambiguous initiating event, use the "what", "where" and "when" keywords to structure. If we consider a SPAD event, the first barrier, $B_1$, could be {Automatic train protection (ATP) OK}. When constructing the event tree the output from a barrier symbol may lead to another barrier symbol. The development is continued to the resulting consequences, illustrated by consequence symbols, $C_1$, $C_2$ etc in Figure 2. We should aim at identifying the barriers in the sequence they are expected to be activated. In this way, there will be an implicit time line from left to right. However, in some situations this is demanding because it is not always easy to say which barriers are activated first.

If we adopt the convention that the "No" branch ("barrier fails to hold") is the downhand branch from the barrier symbol. The most severe consequences will then normally be located to bottom right corner of the consequence spectrum. Note that in some presentations "Yes" is used to describe that the barrier fails. This will then give a different interpretation of the most critical events. It is important to be systematic in the use of the symbols, for examples that "Yes" always is the "best" output, and that "downhand" is used for the "No" output.

### Calculation of probabilities/frequencies for the identified end consequences

In order to carry out the quantitative analysis we need the frequency of the initiating event, and the barrier probabilities. During construction of the event tree, we enter the probability that the various barriers fails (i.e. the "NO" results). For each barrier, $i$, we need:

- $q_i$ = probability that barrier i fails (“No”) , and similarly Similarly we have:
- $p_i = 1 - q_i$ probability that barrier i functions as intended (“Yes”)

In addition to the barrier probabilities, we enter the frequency of the initiating event:

- $f$ = frequency of initiating event

When establishing the barrier probabilities and the initiating frequency it might be required to perform separate analyses, e.g., FTA. Also for the barrier probabilities we usually need separate analyses like FTA for the ATP system, failure statistics and “load/strength” methods.

To calculate the frequencies of the various consequences we may multiply the frequency of the initiating event by the barrier probabilities for each barrier along the path leading to the actual consequence . Now, consider consequence $C_j$, and assume that $S$ = is the set of barriers in the path leading to consequence $C_j$, and that represents ”success” of the barrier (Yes-terminal), and further $F$ = is the set of those barriers on the path leading to consequence $C_j$, and that represent ”the barrier fails” (No-terminal) we have that the frequency of consequence $C_j$ is given by:

$$F_j = f \prod_{i \in S} p_i \prod_{i \in F} q_i$$

This formula is only valid if the barriers are “independent”. This is not always the case, and to overcome the problem of “stochastic” dependent barriers, we should in principle specify the barrier probabilities as conditional probabilities given the course of events up to the current barrier. This is not always easy.

## Reliability Block Diagram (RDB)

Reliability block diagrams are valuable when we want to visualise the performance of a system comprised of several (binary) components.

Figures 3 and 4 shows the reliability block diagram for simple structures. The interpretation of the diagram is that the system is functioning if it is a connection between $a$ and $b$, i.e., there exists a path of functioning components from $a$ to $b$. The system is in a fault state (is not functioning) if it does not exist a path of functioning components between $a$ and $b$.

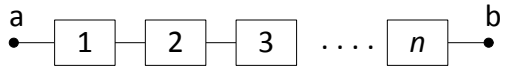



Figure 3: Reliability block diagram for a series structure

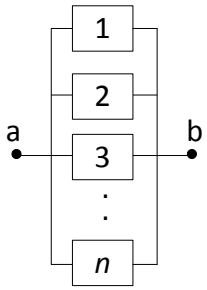



Figure 4: Reliability block diagram for a parallel structure

## Structure function

Recall that we for components have

$$x_i(t) = \begin{cases} 1 \text{ if component } i \text{ is functioning at time } t \\ 0 \text{ if component } i \text{ is in a fault state at time } t \end{cases} \tag{1}$$

For the system we now introduce

$$\phi(\mathbf{x}, t) = \begin{cases} 1 \text{ if the system is functioning at time } t \\ 0 \text{ if the system is in a fault state (not functioning) at time } t \end{cases} \tag{2}$$

$\phi$ denotes the structure function, and depends on the $x_i$s ($\mathbf{x}$ is a vector of all the $x_i$'s). $\phi(\mathbf{x}, t)$ is thus a mathematical function that uniquely determines whether the system functions or not for a given value of the $\mathbf{x}$ -vector. Note that it is not always straight forward to find a mathematical expression for $\phi(\mathbf{x}, t)$.

To simplify notation we skip the index $t$ in the following.

## The structure function for some simple structures

In the following we omit the time dependence from the notation.

For a series structure we have:

$$\phi(\mathbf{x}) = x_1 \cdot x_2 \cdot \ldots \cdot x_n = \prod_{i=1}^{n} x_i$$

For a parallel structure we have

$$\phi(\mathbf{x}) = 1 - (1 - x_1)(1 - x_2) \ldots (1 - x_n) = 1 - \prod_{i=1}^{n} (1 - x_i) = \coprod_{i=1}^{n} x_i$$

Note that we for two components in parallel may simplify:

$$\phi(x_1, x_2) = x_1 \amalg x_2 = 1 - (1 - x_1)(1 - x_2) = x_1 + x_2 - x_1 x_2$$

where the notation ⨿ ("ip") is used for the co-product. For a $k$-out-of-$n$ ($k$oo$n$ : $G$) structure we have

$$\phi(\mathbf{x}) = \begin{cases} 1 \text{ if } \sum_{i=1}^{n} x_i \geqslant k \\ 0 \text{ if } \sum_{i=1}^{n} x_i < k \end{cases}$$

A $k$-out-of-$n$ system is a system that functions if and only if at least $k$ out of the $n$ components in the system is functioning. We often write $k$ oo $n$ *to* denote a $k$-out-of-$n$ system, for example 2 oo 3.

The expression for the structure function of a $k$-out-of-$n$ structure is not attractive from a calculation point of view, i.e., we cannot multiply this expression with other parts of the structure function. We may instead represent the structure by a parallel structure where each of the parallels comprises a series structure of $k$ components. Such a parallel will then function if $k$ or more components are functioning. There are altogether $\binom{n}{k}$ ways we may choose $k$ components out of $n$ components, hence we will have $\binom{n}{k}$ branches. As an example a 2-out-of-3 system may be written as $\binom{3}{2} = 3$ parallels, i.e., {1,2},{1,3} and {2,3}.

For structures comprised of series and parallel structures we may combine the above formulas by splitting the reliability block diagram into sub-blocks, and then apply the formulas within a block, and then treat a sub-block as a block on a higher level. Figure 5 shows how we may split the reliability

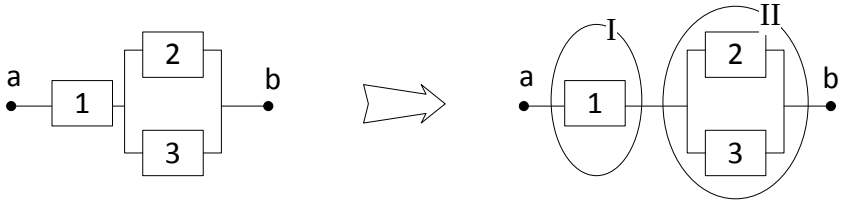



Figure 5: Splitting the reliability block diagram in sub-blocks

block diagram into sub-blocks, here I and II. We may then write $\phi(x) = \phi_{\mathrm{I}} \times \phi_{\mathrm{II}}$ because I and II are in series. Further, we have $\phi_{\mathrm{I}} = x_1$, and $\phi_{\mathrm{II}} = 1\text{-}(1\text{-}x_2)(1\text{-}x_3)$, thus we have $\phi(\mathbf{x}) = x_1(1\text{-}(1\text{-}x_2)(1\text{-}\,x_3))$.

## System structure analysis

### Coherent structures

A system of components is said to be coherent if all its components are relevant and the structure function is non-decreasing in each argument. Proofs are given in the textbook.

The results that follow are only valid for coherent structures.

**Basic results for coherent structures**

**Property 4.1**

$$\phi(\mathbf{0}) = 0 \text{ and } \phi(\mathbf{1}) = 1$$

**Property 4.2** - Any structure is "between" the series and parallel:

$$\coprod_{i=1}^{n} x_i \leq \phi(\boldsymbol{x}) \leq \coprod_{i=1}^{n} x_i$$

**Property 4.3** The effect of redundancy is higher on component level than on system level

$$\phi(\boldsymbol{x} \amalg \boldsymbol{y}) \geq \phi(\boldsymbol{x}) \amalg \phi(\boldsymbol{y})$$

We also have:

$$\phi(\boldsymbol{x} \cdot \boldsymbol{y}) \leq \phi(\boldsymbol{x}) \cdot \phi(\boldsymbol{y})$$

## Paths and Cuts

As for fault tree analysis, we may define cut- and path sets for a structure of $n$ components: $C = \{1, 2, ..., n\}$

- A cut set $K$ is a set of components in $C$ which by failing causes the system to fail. A cut set is minimal if it cannot be reduced without loosing its status as a cut set.
- A paht set $P$ is a set of components in $C$ which by functioning ensures that the system is functioning. A path set is minimal if it cannot be reduced without loosing its status as a path set.

### Structure Represented by Minimal Path Series Structures

A path, say $P_j$, may be considered as a series structure. Since any functioning path ensures the system to function, each path may be considered as one branch in a parallel structure of all the minimal paths, hence we have:

$$\phi(\boldsymbol{x}) = \coprod_{j=1}^{p} \prod_{i \in P_j} x_i$$

### Structure Represented by Minimal Cut Parallel Structures

A cut, say $K_j$, may be considered as a parallel structure. Since any failed cut ensures the system to fail, each cut may be considered as one element in a series structure of all the minimal cuts, hence we have:

$$\phi(\boldsymbol{x}) = \prod_{j=1}^{k} \coprod_{i \in K_j} x_i$$

### Pivotal Decomposition

Pivotal decomposition is often used to analyse complex structures where one or more components are causing "trouble" when we search for series and parallel structures. Typically we use this method for solving "bridge structures". Introduce the following notation:

- $\phi(1_i, \boldsymbol{x})$ = The structure function of the structure when it is given that component $i$ is in a functioning state, i.e., $x_i = 1$.
- $\phi(0_i, \boldsymbol{x})$ = The structure function of the structure when it is given that component $i$ is in a fault state, i.e., $x_i = 0$.

We then have:

$$\phi(\boldsymbol{x}) \equiv x_i \phi(1_i, \boldsymbol{x}) + (1 - x_i)\phi(0_i, \boldsymbol{x}) \text{ for all } \boldsymbol{x}$$

This result is often used when obtaining the structure function of a complex structure. The idea is to use pivotal decomposition of the component that makes the structure troublesome. Conditioning on that component is functioning, i.e., $x_i = 1$, we may rather easily obtain the structure function of the remaining structure, i.e., $\phi(1_i, \boldsymbol{x})$, and similarly if $x_i = 0$, we may rather easily obtain the structure function of the remaining structure, i.e., $\phi(0_i, \boldsymbol{x})$, and then the result for pivotal decomposition may be applied.

## Summary: Finding the structure function

The following principles may be used to find the structure function for a reliability block diagram:

- For a series structure we have $\phi(\mathbf{x}) = x_1 \cdot x_2 \cdot \ldots \cdot x_n = \prod_{i=1}^{n} x_i$.
- For a parallel structure we have $\phi(\mathbf{x}) = 1 - (1 - x_1)(1 - x_2)\ldots(1 - x_n) = 1 - \prod_{i=1}^{n}(1 - x_i) = \coprod_{i=1}^{n} x_i$.
- For a $k$-out-of-$n$ structure we may represent the structure by a parallel structure where each of the parallels comprises a series structure of $k$ components. There are altogether $\binom{n}{k}$ ways we may choose $k$ components out of $n$ components, hence we will have $\binom{n}{k}$ branches.
- For bridge structures and other structures where it is not easy to "see" series and parallel structures, we may use pivotal decomposition, i.e., $\phi(\boldsymbol{x}) \equiv x_i \phi(1_i, \boldsymbol{x}) + (1 - x_i)\phi(0_i, \boldsymbol{x})$ where we decompose around the "troublesome" component.
- If the minimal cut sets are available for a system (or a sub system), the corresponding structure function is given by: $\phi(\boldsymbol{x}) = \prod_{j=1}^{k} \coprod_{i \in K_j} x_i$, and similarly for the path sets: $\phi(\boldsymbol{x}) = \coprod_{j=1}^{p} \prod_{i \in P_j} x_i$.

- We may identify sub-blocks in the diagram (modules), where we for each module represent the state variable of the module by a structure function. i.e., a function of the elements in the module, which is also binary, hence a "formula" can replace the module as if it was a component. In principle any combination of series and parallel structure may be analysed to get the structure function. Bridge structures and $k$-out-of-$n$ structures may also be handled this way. The same principle may also be used if a sub-block is represented by its minimal cut sets or minimal path sets.

The following link shows a system we will analyse with the RBD technique both qualitatively and quantitatively: `http://folk.ntnu.no/jvatn/eLearning/TPK4120/Examples/StructureFunctionUpperBoundInclusionExclusion.docx`.

In the link below an Excel solution is provided, where also the FTA solution is included: `http://folk.ntnu.no/jvatn/eLearning/TPK4120/Excel/StructureFunctionUpperBoundInclusionExclusion.xlsx`.

### Bayesian belief network - BBN

A Bayesian belief network is a probabilistic graphical model that represents a set of variables and their conditional dependencies via a directed acyclic graph

- For example, a Bayesian network could represent the probabilistic relationships between diseases and symptoms
- Given symptoms, the network can be used to compute the probabilities of the presence of various diseases

### Working with BBN

The theory behind BBN is hard to grasp. In order to efficient perform calculation in a BBN we also need theory which is not easy to approach However, there exist computerized tools that can do the necessary manipulations:

- Hugin
- Netica
- More...

### BBN

*Definition:* Bayesian networks are directed acyclic graphs (DAG) whose nodes represent variables, and whose arcs (edges) encode conditional independences between the variables

If there is an arc from node $A$ to another node $B$, $A$ is called a *parent* of $B$, and $B$ is a *child* of $A$.

## FTA and BBN

A fault tree can be represented as a BBN, The basic events are always parent nodes. The gates are child notes, but could be parent nodes of other child nodes. Figure  shows the BBN-representation of a simple fault tree.

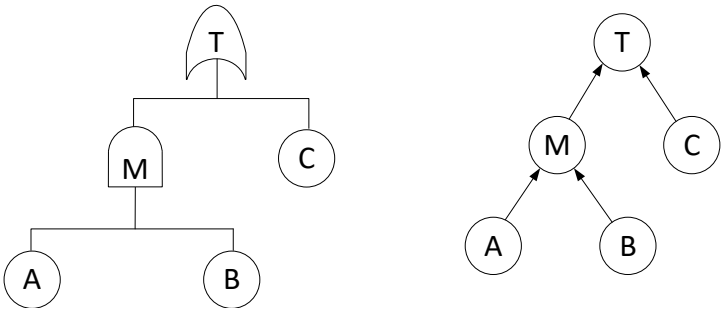



Figure 6: BBN for a simple fault tree

Note that if the gate $M$ was replaced with an OR-gate, the BBN drawing will not change. In order to describe the influence of the "parents" A and B we need a formal way to describe these influences. In general so-called conditional probability tables (CPTs) are used for this purpose. In a qualitative approach such CPTs are denoted truth tables. Tables 1 and 2 shows the truth table for the nodes M, and T respectively.

Table 1: Truth table for M

| $A$ | $B$ | $M\|A,B$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

A value 1 means that the node is true corresponding to the occurrence of the basic event or gate, whereas 0 means that the node is false.

Table 2: Truth table for T

| $M$ | $C$ | $T\|M,C$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |