Fault tree analysis for maintenance needs

To cite this article: Jari Halme and Antti Aikala 2012 J. Phys.: Conf. Ser. 364 012102

View the article online for updates and enhancements.

Related content

- Tolkku - a toolbox for decision support from condition monitoring data
  Olli Saarela, Mikko Lehtonen, Jari Halme et al.

- Using heuristic search for optimizing maintenance plans
  Teemu Mutanen

- Intelligence diagnosis method for roller bearings using features of AE signal
  J Pan, H Q Wang, F Wang et al.
Fault tree analysis for maintenance needs

Jari Halme and Antti Aikala
VTT Technical Research Centre of Finland,
P.O.Box 1000, FI-02044 VTT, Finland
E-mail: jari.halme@vtt.fi

Abstract. One of the key issues in maintenance is to allocate focus and resources to those components and subsystems which are the most unreliable and prone to failures. In industrial systems, fault tree analysis technique can be used to study the reliability of the complex systems and their substructures. In this paper a fault tree application for analyzing online the current reliability and failure probability for maintenance purposes is presented. The analysis is utilizing data connected to the fault tree root causes and events. An indication of an anomaly case, service action, cumulative loading, etc., or just time passed or service hour counter level can trigger a new calculation of current probabilities of the fault tree events and subsystem interactions. In proposed approach real time, dynamic information from several available data sources and different measurement are interconnected to each fault tree event and root cause. There is also formulated an active, constantly updated link between the fault tree events and maintenance databases for the maintenance decision support, and to keep the analysis up to date. Typically top event probability is evaluated based on updated root cause probabilities and lower level events. At the industrial plant level an identification of a failure in a component event defined within a constructed and operatively existing fault tree explicitly means that the event’s failure probability is one. By utilizing this indication, the most probable failure branches through the fault tree sub events to root causes can be identified and printed as a valid check list for maintenance purposes to focus service actions first to those fault tree branches most probable causing the failure. Respectively, during the checks, service actions, etc., components, especially those within the critical branches, detected as healthy can be updated as having zero failure probability. This information can be used to further update the fault tree and produce online a new service order list. The added value of the proposed method with respect to developed software platform functions lies in its applicability to rationalize maintenance actions and in a case of a failure allocate resources where they are assumable mostly needed.

Keywords. Maintenance, fault tree analysis, probability, failure list, online, decision support

1. Introduction
In preventive maintenance the objective is that the components are repaired or changed before their impending failures cause any unplanned loss of production. The proper moment for remediation can be decided either based on scheduled maintenance or diagnostic experience and knowledge about the status of the component. The incidence of failure itself can be stochastic. The management of
stochastic failure events presumes that the failure is modelled taking failure probabilities into account. These models can be utilised in fault tree analysis.

Fault trees were applied for the first time in 1961 for the safety analysis of the missile launch control system by Bell Telephone Laboratory [1]. Fault trees are one the most commonly used method to analyse causal relations together with the measurement and operational data, as well as use and load profiling. At the top of the fault tree structure is a selected unwanted event. It is called the top event. The fault tree structure below the top event defines all the possible series of events which can lead to the top event [2]. The structure and connections of the events are presented graphically in a shape of a tree consisting of branches. The branches and events are connected in the tree hierarchy through logical ports. In most cases single faults are linked with logical OR ports, since by using OR ports each failure event leads to the top event [3]. In cases where all events at the input must be present in order to activate the output they are connected with logical AND ports [2]. Other port types exist as well, but they are rarely used for mechanical systems and components focused on in this paper. The events associated to the roots of the tree are called root causes or root faults. Typically this is the level where the monitoring or probability based information is entered into the fault tree for quantitative or qualitative analysis. In quantitative analysis the occurrence probability of the top event is computed based on the estimated or measured occurrence probability of each root event [4]. The qualitative analysis starts from the top event and lists all the unique combinations of component failures that may result into a system failure. The qualitative results can help to identify system hazards that might lead to failure or unsafe states.

2. Fault tree construction
The input information for the fault tree analysis of a system or an object can be gained from some established failure mode and criticality analysis such as FMECA or from some other type knowledge about the system and its challenges, complexity and needs. The proper depth of the tree naturally depends on the selected system and its relevant hierarchy but also on the failure indications, measurement data, and model information available to be connected to the tree and its root nodes. The failure modes affecting to the root node events can be due to the component’s intrinsic weakness, incorrect operation conditions, wrong or missing inducement [3].

Most typically the inputs to failure trees are entered as current or expected probability values of the respective root node. The probability values can be, e.g., calculated reliability (R) values or failure probability (FP=1-R) values. There are a number of ways to model and statistically estimate the failure probabilities (or reliabilities) for the root nodes. Among these are, e.g., methods based on the exponential and Weibull distributions. Naturally, one can combine different methods in different nodes of a tree.

The simplest method (equation (1)) is based on the exponential distribution. It assumes constant failure time. It has been used, e.g., for electrical components and large, complex systems.

\[ FP_{exp} = 1 - e^{-\lambda t} \]  (1)

In equation (1) the lambda value (λ) is the inverse of statistically relevant mean time to failure (MTTF) value of the root node, t is at its simplest time or, e.g., number of operational hours of the component connected to the node event. Please note that the value of time t can be replaced by, e.g., the cumulative number of critical loadings the node has experienced so far and MFFT correspondingly with the mean cumulative number of loadings to failure, or any other relevant parameters affecting to the reliability and remaining life time of the system or its component.

In the reliability theory, the most widely used distribution type for the length of life is the Weibull distribution [5, 6]. In general, Weibull distribution has been successfully applied to provide reasonable models for lifetimes of equipment such as ball bearings, composite materials, aircraft engines [6], hot strip steel mil [7], etc. The Weibull distribution is versatile. It consists of two or three parameters,
which called shape parameter, nominal life parameter and minimum life time parameter. Equation (2) gives the two parameter version of the Weibull distribution.

\[ F_{P_{\text{weib}}} = 1 - e^{-\left(\frac{t}{\eta}\right)^{\beta}} \] (2)

Typically, first guess for the nominal life time parameter \( \eta \) in equation (2) is a value close to the MTTF value of the component represented by the node. Shape factor \( \beta \) depends on the failure occurrence type and the state of the system. If \( \beta \) is less than one, the system is considered to be at an infant mortality stage, where the failure rate first decreases as a function of time (figure 1). If \( \beta \) is equal to one, the distribution represents useful service life and, having a constant failure rate, is equivalent to the exponential distribution. Values of \( \beta \) higher than one represent wear out period where the expected failure rate value starts to increase as a function of time. Like with the exponential distribution, the time and Weibull parameter values can be replaced with more relevant parameters affecting to the reliability of the attached node. Variables focusing on the actual cause for a degradation process of a system, not just an effecting response, should be favoured, e.g., cumulative loading [8].

![Figure 1. Product life cycle shown as a so-called bathtub figure [9].](image)

Combining data from the maintenance database and real time measurements (e.g., vibration and lubrication state and quality) can also be utilized to give a rough indication about the failure probability when, e.g., excessive vibration or decreased quality of oil is detected. This is important since statistical models aren’t sufficiently accurate during the actual failure process and wear out period.

As described earlier, the root nodes are connected to the events higher in the fault tree with logical ports representing the way sub events affect to the upper ones. The failure probability at each node of the tree can be calculated with a method resembling Boolean algebra. Figure 2 depicts a causal connections with an AND port and an OR port with two input values each. The number of inputs to the port is not limited, but from the practical point of view it might not be wise to connect too many inputs to one port.
The tree is constructed by hierarchically connecting the basic elements presented in figure 2 in such a way, that the root causes are placed in the bottom, they lead through ports to consequences, which in turn are causes for the next level of consequences through the ports at that level, etc. until the top event of the tree is reached.

Calculation rules depend on the input and port type. In equations (3) and (4) are given calculation rules for the AND and OR ports when the inputs are entered as $FP$ values, and respectively in equations (5) and (6) for the $R$ values [10].

Consequence output for failure probability ($FP$) values:

\[ OUT_{\text{And}} = FP_1 \cdot FP_2 \cdot \ldots \cdot FP_n \]  
(3)

\[ OUT_{\text{Or}} = 1 - (1 - FP_1) \cdot (1 - FP_2) \cdot \ldots \cdot (1 - FP_n) \]  
(4)

And consequence output for reliability ($R$) values:

\[ OUT_{\text{And}} = 1 - (1 - R_1) \cdot (1 - R_2) \cdot \ldots \cdot (1 - R_n) \]  
(5)

\[ OUT_{\text{Or}} = R_1 \cdot R_2 \cdot \ldots \cdot R_n \]  
(6)

A fault tree can be constructed in many ways, e.g., by using some commercially available software. There are also some shareware type software available, such as OpenFTA [11], for the construction of the fault trees. One option is to use a spreadsheet program to formulate the tree structure and connections. In our application we used Excel sheets for entering the tree structures for further calculations, analysis, and other handleings of the tree. As one can expect, the spreadsheet approach has much higher flexibility for the usage and utilization, and even large structures are quite easy to outline. A simplified example of the Excel type construction of a fault tree is given in figures 3 and 4. In figure 3 is shown a simplified tree structure for a crane failure, and in figure 4 is shown how this tree can be formulated in Excel for further processing. In the simplified tree, consisting of only OR ports, each root cause forms a minimal cut set i.e. failure in any one of the root causes is capable to stop the crane.
Figure 3. Constructing a fault tree for the estimation of need for crane repair based on \( FP \) values calculated for each root cause according to their individual cumulative failure distributions at time \( t_1 \) (drawn as an example for the bearing and the hook nodes).

Fault tree names are in the first column, 'A'. Field 'type', 'name' and 'value_type' can be in any order.

Node description either with the hierarchy (shown) or without.

For root_cause nodes, the 'set_value' values are shown. Otherwise node type is shown. Note: the column for 'set_value' can be changed according to the hierarchy.

Figure 4. Constructing a fault tree for the estimation of crane repair by using a spreadsheet program. The tree consists of three sub branches, one for the motor, one for the brake and one for lifting failure.

The numerical “set_values” in figure 4 represent static failure probability starting values for the root nodes, and these can be null at the start. Naturally, root case failure probability values can be read...
from the spreadsheet program for the static, quite rarely done analysis. More dynamic analyses require that the input values are retrieved by other means. Typically the failure probability values are updated and set according to selected on-line or off-line data. The utilisation of the tree is discussed in next chapters.

3. Fault tree utilization
Fault trees are most typically calculated from the bottom to top, estimating the failure probability (or reliability) of the top event. Individual probability values can be shown for each node of the tree. These probabilities can be utilized as such for the off-line or on-line evaluation of the current failure probability value of the system and its components. For example, by analyzing earlier production schemes and seasonal variations in production, it is possible to model the fault tree input parameters as functions of time and production. Dependent on the models and their interdependencies, these can be used to estimate future parameter values and their evolution within certain limits. This gives an interesting option to use that information for the prediction of the failure probability values by entering the modelled and expected future parameter values to the roots of the failure tree and either based on the feed of individual values at fixed time points or their more versatile simulations predict the future failure probability values for the tree to find e.g. the most critical failure progression paths and nodes of the system in the future. It is also possible to test the effects of different productions etc. scenarios, if seen important.

Another way is to look at the tree from the other perspective, namely from top to bottom, and look for the most probable failure paths through the fault tree. In paper [12] different methods to rank the effect of each individual root cause’s contribution to the entire fault tree failure have been tested. Based on this analysis, Xing proposes that the method labeled as Diagnostic importance factor (DIF) is the most informative and appropriate measure for the maintenance oriented importance analysis. In this work the DIF method is selected to compute the fraction of the system failure probability that involves the failure of the selected component, i.e., \( FP_{\text{system AND component}} / FP_{\text{system}} \). In the case of an identified failure of a component or function of the tree, the analysis of the tree from top to bottom can save maintenance time and resources. Especially, in the large systems, this helps to identify the most critical sub-systems and components causing failures. The most probable failure branches from the identified failure spots through the underlying sub nodes and events to root causes can be printed as a valid priority checklist based on the evaluated criticalities and importance for the use of maintenance personnel.

During maintenance the checklist order is followed in failure identification and root cause checking, and the observed conditions of the checked components and systems can be updated online. E.g., if a component on the checklist is found working as it should, the respective fault tree node can be updated as having zero failure probability and further used online to update the fault tree and produce a new service order checklist for the identification of the real root cause of the failure. This procedure rationalizes maintenance actions and allocates resources where they are mostly needed. Naturally, the redirection of maintenance efforts can be done first at hierarchically higher levels of the tree, focusing first for the main components and systems, and then proceeding to the more detailed events and causes.

In addition to calculated node values, known usage and load profiles can be used to give additional information about the state at the nodes, and at any level of the tree. Relevant profiles can be used in parallel with the failure probability values calculated based on the root cause input values only. Also on-line or off-line process and condition monitoring measurements tacit knowledge, and, e.g., hearing and visual observations from the field can be used as an information source prominent for the evaluation of the failure probabilities at the different nodes in the tree.
4. Results and discussion

The calculations shown here are carried out with fault tree analysis tools in the Tolkku O&M toolbox for the Matlab environment. A more comprehensive look at the Tolkku toolbox and its deployment is given in *Tolkku - a toolbox for decision support from condition monitoring data* [13]. Tolkku data flow to and from a fault tree for the maintenance decision support is shown in figure 5.

![Flow of data for the fault tree usage as part of the maintenance decision support using the Tolkku toolbox.](image)

**Figure 5.** Flow of data for the fault tree usage as part of the maintenance decision support using the Tolkku toolbox.

In the following some examples are given related to the utilization of a simplified fault tree presented in figures 3 and 4. Based on the root cause failure probability values and the fault tree structure the calculation yields the failure probabilities for the crane and its sub events shown in table 1.

**Table 1.** Failure probabilities for a crane failure during normal operation at a certain time point.

<table>
<thead>
<tr>
<th>Event type</th>
<th>Event name</th>
<th>Failure probability value at time point</th>
</tr>
</thead>
<tbody>
<tr>
<td>top event</td>
<td>Crane failure</td>
<td>0.2364691</td>
</tr>
<tr>
<td>event</td>
<td>Electrical motor failure</td>
<td>0.1290777</td>
</tr>
<tr>
<td>event</td>
<td>Brake failure</td>
<td>0.1209734</td>
</tr>
<tr>
<td>event</td>
<td>Lifting failure</td>
<td>0.0026554</td>
</tr>
</tbody>
</table>

The values in table 1 are calculated from bottom to up, and these can be updated any time when the root case values change either due to encountered operational progress, due to simulations of expected operational progress in the future, or due to maintenance or repair actions. Updated failure probability values can be used for the planning of maintenance actions and their order of importance [14].

In addition to just to estimating the current failure probabilities of the machine and the related sub events, fault trees serve as a tool to look for the most probable failure causes through the tree, which is especially useful in a case of a failure. In table 2 is shown a printout of a crane service order list, where possible failure causes and their respective paths through the tree are printed in the order of...
importance based on the current status in the fault tree when the failure probability of the node representing top event (‘Crane failure’) in the fault tree is assigned value 1 (Faulty).

**Table 2.** A printout of the top 5 most possible failure causes for the observed crane failure ($FP=1$) based on the current status in the fault tree.

Check list for a crane failure during normal operation

<table>
<thead>
<tr>
<th>Priority order</th>
<th>Event</th>
<th>Crane failure</th>
<th>Electrical motor failure</th>
<th>Bearing failure</th>
<th>DIF</th>
<th>Crane failure</th>
<th>Brake failure</th>
<th>Wearing</th>
<th>0.511582</th>
<th>0.429443</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Event</td>
<td>Crane failure</td>
<td>Brake failure</td>
<td>Wearing</td>
<td>DIF</td>
<td>Crane failure</td>
<td>Brake failure</td>
<td>Leakage</td>
<td>0.511582</td>
<td>0.091145</td>
</tr>
<tr>
<td>4</td>
<td>Event</td>
<td>Crane failure</td>
<td>Electrical motor failure</td>
<td>Stator failure</td>
<td>DIF</td>
<td>Crane failure</td>
<td>Lifting failure</td>
<td>Rope failure</td>
<td>0.011229</td>
<td>0.074991</td>
</tr>
</tbody>
</table>

The list can be given to service personnel to start check first for the most prominent failures. In this example the most obvious event causing the crane failure is originating from the electrical motor caused by its bearing failure as seen from the first case printed on the check list shown in table 2. Respectively, listed as the second candidate, is a brake failure caused by wearing, etc. An alternative way, as demonstrated in [12] is just to list the DIF values for the root causes in the order of magnitude. In this simplified example case the troubleshooting is quite straightforward a task, but in larger systems tools like this can rationalize the usage of maintenance resources and reduce time for repairing.

Correspondingly, any other fault tree event related failure observation from the field can be utilized by assigning value 1 to the $FP$ of the corresponding node value and generating a new check list updated with the new information. Also, if the service personnel sent to look for the cause for the crane failure notices from the field that despite of the original list, the bearings in the electrical motor are ok and without a failure, and hear at the same time some speculation about the condition of the lifting mechanics, they can update the tree by setting the $FP$ value for the bearing failure to 0 (Healthy) and for the lifting failure to 1 (Faulty) (see table 3). Respectively, any other event or root cause detected as healthy or possibly faulty can be updated to generate a new check list for retargeting of failure search resources.
Table 3. A printout of the top 5 most possible failure causes for the observed crane failure based on the current status in the fault tree when FP value for the bearing failure is enforced to 0 (healthy) and for the lifting failure to 1 (faulty).

Check list for a crane failure during normal operation

<table>
<thead>
<tr>
<th>Priority order</th>
<th>Failure branch with events and sub</th>
<th>DIF</th>
<th>DIF</th>
<th>DIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Event: Crane failure Lifting failure Rope failure</td>
<td>1</td>
<td>1</td>
<td>0.667090</td>
</tr>
<tr>
<td>2</td>
<td>Event: Crane failure Lifting failure Rope pulley failure</td>
<td>1</td>
<td>1</td>
<td>0.266840</td>
</tr>
<tr>
<td>3</td>
<td>Event: Crane failure Brake failure Wearing</td>
<td>1</td>
<td>0.120973</td>
<td>0.101550</td>
</tr>
<tr>
<td>4</td>
<td>Event: Crane failure Lifting failure Hook failure</td>
<td>1</td>
<td>1</td>
<td>0.066709</td>
</tr>
<tr>
<td>5</td>
<td>Event: Crane failure Brake failure Leakage</td>
<td>1</td>
<td>0.120973</td>
<td>0.021553</td>
</tr>
</tbody>
</table>

In general, fault tree analysis software typically calculate the exact value for the top event and lists importance of all possible root faults in an analyzed top event failure case. For the failure case analysis, there is available both deterministic and simulation based solution. If the number of possible combinations of the tree for the failure becomes high, calculation for the deterministic, exact solution takes too much time. Then simulation (e.g. Monte Carlo) based approximation is a convenient option for those cases. For the simple example structure presented above, a deterministic solution is possible to calculate for any failure. However, the analysis software switches automatically from deterministic to simulation based solution when the number of combinations increases over a limit where the computation can cause run out of available computer memory.

Traditionally, in fault tree analysis the focus is on reliability and failure probability issues, and possible event manipulation etc. handling for the maintenance purposes is out of their area of application. As shown, in the prosed approach the fault tree event manipulation can be done in an easy way regarding any event defined in the tree for, e.g., generating service checklists, and even for the flexible manipulation based on the latest observation at the field. In addition, probability values for root causes and events can be acquired and estimated online to keep the tree information and its analysis up to date.

Current reliability and remaining lifetime should be estimated based on recent data, selected variables and statistically meaningful historical data. The effect of operation and loadings should be taken into account by using experienced loading profiles and respective parameters on-line, as well, to improve the accuracy of the fault tree calculation when possible. Failures can and should be taken into account, as seen e.g. from the case shown in table 3. Ideally, an individual maintenance strategy should be applied for every node of the tree, according its specific situation and overall needs based on criticalities and economic considerations. Naturally some components can be driven until the failure, while for others some preventive strategies should be applied. In the proposed approach, this can be taken into account by simulating the tree and its nodes with the requested models and parameters. Estimates indicating good reliability state suggest that less aggressive maintenance methods can be favoured while indications of decreased reliability faster and more responsive data and maintenance models can be selected.
5. Conclusions
Fault trees are one the most commonly used method for analyzing causal relationships between components and failures. The results of this papers showed that fault trees can be constructed with an easy and multifunctional way by utilizing spreadsheet software for the tree structure definition. The data for the fault tree can be updated online based on use and loadings of the components, and especially when a deviation in operation, wear, fault, etc. is detected. Also, the failure probabilities can be estimated from, e.g., operational hours, loadings, trends, etc. In a failure case, the most probable failure branches can be identified and printed as a prioritized check list for maintenance purposes aiming the focus first to branches representing the most probable causes of the failure. During the service, information about components detected to be healthy can be updated online for generating an updated service check list. This procedure helps to rationalize maintenance and allocate resources where they are most needed.

Acknowledgements
This paper is based on work carried out within the TOLKKU project in FIMECC EFFIMA program. The support and co-operation with Bronto Skylift, John Deere, Konecranes, Metso Minerals, and FIMA (Forum for Intelligent Machines) are gratefully acknowledged. Fruitful discussions with the TOLKKU project team and TOLKKU project manager, principal scientist Mikko Lehtonen are greatly appreciated.

References