Optimal design of systems with standby dependencies

This item was submitted to Loughborough University’s Institutional Repository by the/an author.


Additional Information:

• This is a conference paper

Metadata Record: https://dspace.lboro.ac.uk/2134/3687

Publisher: (c) The authors

Please cite the published version.
This item was submitted to Loughborough’s Institutional Repository by the author and is made available under the following Creative Commons Licence conditions.

For the full text of this licence, please go to:
http://creativecommons.org/licenses/by-nc-nd/2.5/
Optimal Design of Systems with Standby Dependencies

L.M.Ridley and J.D.Andrews
Department of Mathematical Sciences
Loughborough University
Loughborough
Leicestershire
LE11 3TU

Summary

The technique of Fault Tree Analysis is commonly used to assess the probability of failure of industrial systems. This method represents the failure logic of the system in an inverted tree structure and has the advantage that it provides very good documentation of the way that the system failure logic was developed. During the analysis of the fault tree the component failures or basic events are assumed to occur independently. When this condition is not satisfied, as in the case of standby systems for example, alternative approaches such as Markov methods can be used. Constructing the state transition diagram required for Markov models is not such an intuitive process for engineers as fault tree construction since state transient diagrams do not readily document the failure logic process.

This paper introduces new gates into the fault tree diagram, which enable the reliability analyst to incorporate standby dependencies. The analysis of the fault tree is then performed by identifying the sections of the fault tree which conform to the usual requirements of independence and those which do not. Using a combination of conventional fault tree analysis methods with those of Markov methods the analysis of the tree is performed by computer code in a manner which is transparent to the analyst. A similar approach has been developed for the analysis of the systems with sequential failures.

With these extended fault tree capabilities in place the above techniques have been embedded within an optimisation framework to get the best system performance for systems where failures are dependent.

1. Introduction.
An accident is the result of a coincidence of events and, or failures of individual components. Analysis of industrial systems is carried out to aid in the protection of facilities and to help reduce the risk of loss of profit, injury or death by reducing the frequency or consequence of such accidents. Analyses have been achieved over the last twenty years by using various types of mathematical models (1). The range of modelling techniques has increased over the last decade and improvements to existing techniques are constantly being developed.
The types of industrial system that are most commonly assessed are those which have a safety function. Depending on the system characteristics an appropriate mathematical model is selected such as Fault Tree Analysis (2), Markov modelling (3), Simulation (2) or Reliability Networks (3).

The fault tree represents the failure logic in a way which engineers find easy to construct and understand. The textual description provided for each gate output also provides documentation for the logic diagram. These features, together with the efficiency of the method, have resulted in the fault tree analysis technique becoming the most commonly used of the alternative analysis options.

Fault tree quantification, which is based on the Kinetic Tree Theory of Vesely (4), requires a number of assumptions regarding the system. One of these is that basic events in the tree will occur independently. Two situations frequently arise in engineering safety systems which render the Kinetic Tree Theory inappropriate: Sequential Events and Standby Systems.

Sequential events, which must occur in a specified order to cause the developed failure, can be represented on the fault tree with a PRIORITY AND gate. However, this is then conservatively treated as a normal AND gate for quantification.

Markov analysis is one means of accurately analysing sequential events. However the Markov State Transition diagram is not as easy to construct as a fault tree and is therefore not as attractive to reliability engineers. Also with the Markov modelling approach the size of the state transition diagram increases rapidly as the number of components in the analysis increases. This means that a combination of fault tree and Markov methods are required as the Markov model of the complete system would, in most cases, be too large. Dugan (5) embedded a Markov analysis option within a fault tree framework, which was transparent to the users. Thus retaining the advantages of the fault tree logic structure whilst improving the accuracy by analysing the relevant sectors of the tree with Markov models and substituting the results back into the fault tree structure.

Failure of standby components also requires a modelling technique which can handle dependencies. Failure of standby components is dependent on the failure of the primary element. This is the subject considered in this paper. A new "standby" gate type for the fault tree has been defined and a means of developing the correct Markov model to represent this situation specified. As with the sequential events results from the Markov analysis are fed back to the fault tree model for complete system quantification.

2. Definition of New Gate Types
The parts of the fault tree for which the sequence of failure events is important will be referred to as "Dynamic", the remainder of the non-sequential tree structure will be called "Static". The new gates that need to be incorporated into the traditional fault tree
structure must be able to deal with dynamic tree sections such as standby components and sequentially dependent components. For sequential sections use has been made of the PRIORITY AND gate and the analysis method proposed by Dugan (5). A new standby gate has been developed so that analysis of systems that include standby components can be achieved.

The PRIORITY AND gate shown in figure 1 will result in the output event occurrence if the input events to the gate occur in the order in which they enter the gate (from left to right).

![Figure 1. PRIORITY AND gate with 2 inputs.](image)

Therefore, for the gate shown in Figure 1, the TOP event will occur if A fails first and then B fails next. If B fails first and then A fails the system will not fail. In traditional fault tree analysis this particular gate would be quantified by replacing the PRIORITY AND gate with a normal AND gate. To achieve a more accurate result the tree can be converted into a Markov diagram and thus solved.

The Markov diagram for the PRIORITY AND gate with two input events is shown in Figure 2. Both A failed and B failed can be achieved through 2 different routes arriving at states 4 and 5. A failed first, A:F1, and B fails second, B:F2, is represented by state 4 which is the output event to the PRIORITY AND gate.
The probability, then, of system failure is the probability of residing in the state 4.

When PRIORITY AND gates are encountered with more than two inputs a general procedure for their solution would be to split the gate into pairs and solve the Markov diagram for each pair using the model in Figure 2. Thus a three input PRIORITY AND gate would be developed as shown in Figure 3 with gate G1 and then TOP solved separately. This is, however, only applicable if no events repeated elsewhere in the tree occur in the pairings.

---

**Figure 2.** The Markov Representation of two input PRIORITY AND gate. i:W indicates that component i is in a working condition as i:F indicates that component i is in a failed state. λ represents the failure rate of a component while v represents the repair rate of a component.

**Figure 3.** Fault Tree with 1 PRIORITY AND gate with 3 basic events.
3.1 Analysis Procedure.
As only a fraction of the tree is usually dynamic, transforming the entire tree into a Markov diagram in order to solve for this small dynamic part would be very inefficient and for even moderately sized systems may be impractical due to the number of states in the model. One way to overcome this is to use the principle of the combined model, solving static and dynamic parts separately.

The general procedure includes finding independent subtrees. These subtrees are then solved either by using fault tree or Markov techniques, depending on whether they contain static or dynamic components. Once the individual subtrees have been solved the subtree is replaced by a new basic event, which contains the appropriate reliability parameters gained from the solution using either fault tree or Markov analysis.

The identification method traces through the original fault tree, depth-first left most, starting at the top event. Each gate is then checked to determine whether or not it is a largest possible independent subtree, which is either dynamic or static in nature. Once these modules are identified they are solved by the appropriate technique. Their solution is integrated back into the tree to gain the solution of the entire fault tree. Several Markov and fault tree models may need to be solved to give the final result.

For example consider the fault tree in Figure 3 for a sequential system.

1) Starting at the top level consider the first gate (TOP). The inputs to TOP are G1 and C (which is a basic event and therefore does not need investigating). G1 is then checked to determine whether or not it is an independent subtree.

2) The inputs to G1 are A and B, which are basic events. Since they do not occur elsewhere in the tree G1 is an independent module.

3) It is noted that G1 is a dynamic gate and so, as it is independent and dynamic it is converted into a Markov diagram and solved. The probability and frequency of event A occurring before event B are obtained and these results are assigned to a new event, say E, which replaces G1 in the original tree.

4) The new fault tree structure with E and C inputs to a PRIORITY AND gate, TOP, is also a dynamic independent module and can be solved using Markov models.

5) The solution to TOP yields the solution to the original tree.

This simple example uses Markov analysis twice to gain the probability of the top event in Figure 3 occurring.
The most effective way to illustrate the new standby gate and its use is by means of a simple example system.

The system used to illustrate the method, shown in Figure 4, is part of a deluge system found on an offshore platform. In this particular example the pump system will fail to function given a demand for the system, if at any time less than 2 pumps are functioning. The system comprises of two electric pumps and two standby diesel pumps. Ideally the 2 electric pumps will function on demand and only in the event of either of these pumps failing will the diesel pumps be brought into operation.

![Figure 4. Deluge Pump System](image)

The Fault tree representation for this system can be seen in Figure 5. The procedure for analysing this tree follows similar lines to that for a sequential system but in this case a fault tree assessment is required in order to produce data for the Markov state transition diagram. A "Standby gate" for the top event is introduced to indicate the number of standby channels required to enable the system to function. The new standby gate indicates that at some point below the gate a standby element will be encountered, which may be dependant upon other events occurring. The standby inputs are brought in, in left
to right order. The causes of each pump stream failure can be that it fails to start, represented by the static elements of the fault tree branch, or that it fails once running given by the dynamic element. The dynamic element is labelled with a 'S' for identification. The trees need to be separated and the static section analysed to give the initial conditions for the Markov model.

To simplify the example it has been assumed that the electric and diesel power supplies will be 100% available when the system activates but can fail once running.

Figure 5. Fault Tree Representation of Deluge Pump System.
4.1 Analysis Procedure.

1) The Top gate, a new dynamic standby gate, has four inputs, G1, G2, G3 and G4, of which 2 from 4 will need to be operational for correct system functioning. Each is investigated, left-most first, to identify independent subtrees.

2) Through tracing down the left-most branch (G1) it can be identified that there exists a standby component, denoted by a 'S' sign. $PFR$ represents the failure of a pump when already running, this is clearly a dynamic component. This event is removed from the "Static" tree for processing using the Markov model.

3) The Electric Power supply (EP) and Diesel Power Supply (DP) are repeated events as they are common to more than one subtree. These events are also removed from the "Static" tree for processing using the Markov model.

4) All standby components and repeated events will be processed later, and their removal results in G1 becoming the top event of an independent subtree. This subtree is static in nature and is therefore solved using fault tree techniques to yield $P_{E1}$, the probability of Electric pump 1 failing to start.

5) The same procedure is used for analysing G2-G4 and $P_{E2}$, $P_{D1}$ and $P_{D2}$, the unavailabilities of the other pump streams, are calculated.

6) The Top gate is returned to and due to its dynamic nature needs to be solved using a Markov Approach. The Markov model starts in the initial dormant state. To gain the transition rates from the dormant state to any initial active state the probabilities $P_{E1}$, $P_{E2}$, $P_{D1}$ and $P_{D2}$ together with the deluge system demand rate $\lambda$ are used. Given a demand the system can start in a number of different initial states depending on whether any of the pump systems have failed to activate. For example, if all pumps are available, the initial state would be EP1: W, EP2: W, DP1: S, DP2: S (S=Standby, W=Works) and the transition rate from the dormant state to this state would be $\lambda (1-P_{E1})(1-P_{E2})(1-P_{D1})(1-P_{D2})$. This is state 1 on the Markov diagram illustrated in Figure 6, where the states identify the components condition in order E1, E2, D1 and D2, and $\lambda_F$, $\lambda_{P1}$, $\lambda_{P2}$, $\lambda_{P3}$ and $\lambda_{EP}$ are the failure rates of electric pump1, 2, diesel pump1, 2 and the electric power supply respectively. All possible initial states are identified by states 1-16 with transition rates from the dormant state calculated. These states together with the states that are created when components fail once running or are repaired develops the entire Markov diagram (Figure 6).

7) The repeated events are now taken into consideration, such as failure of electric and diesel power supply. For example failure of the electric power supply will cause a transition from any state with EP1/EP2 operational to an appropriate initial state using the
diesel pumps. This is also incorporated into the Markov diagram as shown by transition from state 1 to state 6 when the electric power fails.

8) The Markov diagram is now solved and the probability of the pump system failing is given by summing the probabilities of those states that have less than two pumps in operation (i.e. less than k pumps working, where standby gate needs k/n inputs to function)

5. Analysis of Design Variations.

In order to be able to determine the 'best' design, a number of different design variations need to be considered. To construct an individual fault tree for each design could be time consuming, depending on how many variations exist. The design alternatives can be specified by the designer or by embedding the analysis procedure within an optimisation framework as in (6). An advantage of retaining the fault tree structure to represent the failure logic for these dynamic elements is that design options can be represented using House events. House events are used, and are set to either TRUE or FALSE using the numerical codes 1 and 0 respectively. In this way parts of the tree can be switched on or off depending on the particular design variation. The tree is then pre-processed to remove the House events and produce a fault tree, which contains only failure events. This tree is then subject to analysis by the methodologies described above.

The algorithm below shows how the code will treat any house event when it is encountered:

1 = Delete house event input to gate. If this only leaves one input event delete AND gate; change input event to be an input to the gate at next level up.
Figure 6. Markov Representation of Dynamic Part of Deluge System.
2 = Gate output is true. Proceed up through tree structure until an AND gate is encountered. Delete this input branch up to AND gate. If this only leaves one input event delete AND gate, change input event to be an input to the gate at next level up.

3 = Gate output is false. Proceed up through tree structure until an OR gate is encountered. Delete input branch up to the OR gate. If this is the only input to the OR gate change to an input event to the gate at the next level up.

4 = Delete House Event to gate. If this only leaves one input to the OR gate change to an input event to the gate at the next level up.

Conclusion.
1) A model has been outlined that uses both fault tree and Markov analysis methods to solve a system that includes both dynamic and static components. This enables sequential failures and standby dependencies to be modelled. The starting point for the analysis is the construction of a fault tree. Dependencies are incorporated using PRIORITY AND and Standby gates.

2) The fault tree can be constructed incorporating House events to model each different design alternative. The house events are set to represent a design and the tree pre-processed prior to analysis. In this way the method can be utilised with a design optimisation procedure.

References