Lecture Notes

Practical Approach to Reliability, Safety, and Active Fault-tolerance

Izadi-Zamanabadi, Roozbeh

Publication date:
2000

Document Version
Også kaldet Forlagets PDF

Roozbeh Izadi-Zamanabadi

Lecture Notes - Practical Approach to Reliability, Safety, and Active Fault-tolerance

December 11, 2000

Aalborg University

Department of Control Engineering
Fredrik Bajers Vej 7C
DK-9220 Aalborg
Denmark

Table of Contents

1. INTRODUCTION

> The fundamental objective of the combined safety and Reliability assessment is to identify critical items in the design and the choice of equipment that may jeopardize safety or availability, and thereby to provide arguments for the selection between different options for the system.

Achieving safety and reliability has been one the prime objectives for system designers while designing safety critical system for decades. With growing environmental awareness, concerns, and demands, the scope of the design of reliable (and safe) systems has been enhanced to even small components as sensors and actuators. In the past, the normal procedure to address the higher demand for reliability was to add hardware redundancy that in turn increases the production and maintenance costs. Active fault-tolerant design is an attempt to achieve higher redundancy while minimizing the costs.

In chapter 2 reliability and safety related issues are considered and described. The idea of introducing this chapter is to provide an overview of the concepts and methods used for reliability and safety assessment.

The focus in chapter 3 is on fault-tolerance concept. Type of possible faults in components and customary methods for applying redundancy is described. Finally, the chapter is wrapped up by considering and describing the main subject, which is a formal and consistent procedure to design active fault-tolerant systems.

1.1 Terminology

Definition of the used notations in this report is provided in this section.

Availability:	The probability that a system is performing satisfactorily at time t.
Fault-tolerance:	Ability to tolerate and accommodate for a fault with or without performance degradation.
Fail-operational:	The component/unit stays operational after one failure.

Fail-safe: The component/unit processes directly to a predefined safe state (actively or passively) after one (or several) fault has occurred.

Hazard: the presence of any process abnormality which represents a dangerous or potentially dangerous state. *

MTBF: Mean time between failure

MTTR: Mean time to repair

Reliability: is the characteristic of an item expressed by the probability that it will perform its required function in the specified manner over a given time period and under specified and assumed conditions.

Risk: (normally) refers to the *statistical annual frequency* of the particular hazard state. Ex. once per year, or once per 10000 years.

Risk reduction: Reduce the probability of hazardous events

Safety: is freedom from accident or losses.

Safety integrity: is the probability of a safety-related system satisfactorily performing the required safety functions under all stated conditions within a stated period of time.

Safety Integrity Level: Average probability of failure to perform its designed function on demand (**SIL**).

*Potential Hazard states, are most often identified by carrying out hazard and operational studies known as *HAZOP*s.

2. RELIABILITY AND SAFETY

2.1 Reliability

Reliability engineering is concerned primarily with failures and failure rate reduction. The reliability engineering approach to safety thus concentrate on failures as cause of accidents. Reliability engineers use a variety of techniques to minimize component failure (Leveson 1995).

Definition of reliability embraces the clear-cut criterion for failure, from which we may judge at what point the system is no longer functioning properly.

2.1.1 Basic definitions

Reliability is defined as the probability that a system survives for some specified period of time. It may be expressed in term of the random t, the *time-to-system-failure*. The probability density function (PDF), $f(t)$, has the physical meaning:

$$f(t)\Delta t = P\{t < \mathbf{t} \leq t + \Delta t\} = \left\{ \begin{array}{c} \text{Probability that failure takes place} \\ \text{at a time between } t \text{ and } \Delta t \end{array} \right\} \quad (2.1)$$

for vanishing small Δt. The cumulative distribution function (CDF), $F(t)$ has now the following meaning:

$$F(t) = P\{\mathbf{t} \leq t\} = \left\{ \begin{array}{c} \text{Probability that failure takes place} \\ \text{at a time less than or equal to } t \end{array} \right\} \quad (2.2)$$

The **reliability** is defined as:

$$R(t) = P\{\mathbf{t} > t\} = \left\{ \begin{array}{c} \text{Probability that a system operates} \\ \text{without failure for a length of time } t \end{array} \right\} \quad (2.3)$$

Since a system that does not fail for $\mathbf{t} \leq t$ must fail at some $\mathbf{t} > t$, one get:

$$R(t) = 1 - F(t) \quad (2.4)$$

or equivalently either

$$R(t) = 1 - \int_0^t f(t')dt' \tag{2.5}$$

or

$$R(t) = \int_t^\infty f(t')dt' \tag{2.6}$$

Following properties are clear:

$$R(0) = 1 \quad \text{and} \quad R(\infty) = 0 \tag{2.7}$$

One can see from equation 2.4 that reliability is the complementary cumulative distribution function (CCDF), that is, $R(t) = 1 - F(t)$. Similarly, since $F(t)$ is the probability that the system will fail before $\mathbf{t} = t$, it is often referred to as the unreliability or failure probability, i.e., $F(t) = 1 - R(t)$.

Equation 2.5 may be inverted by differentiation to give the PDF of failure times in terms of the reliability:

$$f(t) = -\frac{d}{dt}R(t) \tag{2.8}$$

One can gain insight into failure mechanisms by examining the behavior of the failure rate. The *failure rate*, $\lambda(t)$, may be defined in terms of the reliability or the PDF of the time-to-failure as follows. Let $\lambda(t)\Delta t$ be the probability that the system will fail at some time $\mathbf{t} < t + \Delta t$ given that it has not yet failed at time $\mathbf{t}$. Thus it is the conditional probability

$$\lambda(t)\Delta t = P\{\mathbf{t} < t + \Delta t | \mathbf{t} > t\}. \tag{2.9}$$

Following the definition of conditional probability we have:

$$P\{\mathbf{t} < t + \Delta t | \mathbf{t} > t\} = \frac{P\{(\mathbf{t} > t) \cap (\mathbf{t} < t + \Delta t)\}}{P\{\mathbf{t} > t\}}. \tag{2.10}$$

The numerator of equation 2.10 is just an alternative way of writing the PDF, i.e.:

$$P\{(\mathbf{t} > t) \cap (\mathbf{t} < t + \Delta t)\} \equiv P\{t < \mathbf{t} < t + \Delta t\} = f(t)\Delta t. \tag{2.11}$$

The De-numerator of equation 2.10 is just R(t) (see equation 2.3). Therefore, by combining equations, one obtains:

$$\lambda(t) = \frac{f(t)}{R(t)}. \tag{2.12}$$

This quality, the failure rate, is also referred to as the *hazard* or *mortality rate*. The most used way to express the reliability and the failure PDF is in terms of failure rate. To do this, we first eliminate $f(t)$ from Eq. 2.12 by inserting Eq. 2.8 to obtain the failure rate in terms of the reliability,

$$\lambda(t) = -\frac{1}{R(t)}\frac{d}{dt}R(t). \tag{2.13}$$

Then by multiplying both side of the equation by dt and integrating between zero and t, we obtain

$$\int_0^t \lambda(t')dt' = -\ln[R(t)] \tag{2.14}$$

since $R(0) = 1$. The desired expression for reliability can hence be derived by exponentiating the results

$$R(t) = \exp\left[-\int_0^t \lambda(t')dt'\right]. \tag{2.15}$$

The probability density function $f(t)$ is obtained by inserting Eq. 2.15 into Eq. 2.12 and solve for $f(t)$:

$$f(t) = \lambda(t)\exp\left[-\int_0^t \lambda(t')dt'\right]. \tag{2.16}$$

The most used parameter to characterize reliability is the *mean time to failure* or MTTF. It is just the expected or mean value of $E\{t\}$ of the failure at time t. Hence

$$\text{MTTF} = \int_0^\infty tf(t)dt. \tag{2.17}$$

The MTTF may be written directly in terms of the reliability by substituting Eq. 2.8 and integration by parts:

$$\text{MTTF} = -\int_0^\infty t\frac{dR}{dt}dt = -tR(t)\Big|_0^\infty + \int_0^\infty R(t)dt \tag{2.18}$$

The term $tR(t)$ vanishes at $t = 0$. Similarly, from Eq. 2.15, we see that $R(t)$ will decay exponentially, since the failure rate $\lambda(t)$ greater than zero. Thus $tR(t) \longrightarrow 0$ as $t \longrightarrow \infty$. Therefore, we have

$$\text{MTTF} = \int_0^\infty R(t)dt \tag{2.19}$$

2.1.2 Constant failure rate model and the exponential distributions

Random failures that give rise to the constant failure rate model are the most widely used basis for describing reliability phenomena. They are defined by the assumption that the rate at which the system fails is independent of its age. For continuously operating systems this implies a constant failure rate.
The constant failure rate model for continuously operating systems leads to an exponential distribution. Replacing the time dependent failure rate $\lambda(t)$ by a constant λ in Eq. 2.16 yields, for the FDP,

$$\boxed{f(t) = \lambda e^{-\lambda t}}. \tag{2.20}$$

Similarly, the CDF becomes

$$F(t) = 1 - e^{-\lambda t}, \tag{2.21}$$

and reliability may be written as (see Eq. 2.4)

$$\boxed{R(t) = e^{-\lambda t}}. \tag{2.22}$$

The MTTF and the variance of the failure times are also given in terms of λ. From Eq. 2.19 we obtain

$$\boxed{\text{MTTF} = \frac{1}{\lambda}}, \tag{2.23}$$

and the variance is found to be:

$$\sigma^2 = 1/\lambda^2. \tag{2.24}$$

2.1.3 Mean time between failure with constant failure rate

In many situations failure does not constitute the end of life. Rather, the system is immediately replaced or repaired and operation continues. In such situations a number of new pieces of information become important. We may want to know the expected number of failures over some specified period of time in order to estimate the cost of replacement parts. More important, it may be necessary to estimate the probability that more than a specific number of failures N will occur over a period of time. Such information allows us to maintain an adequate inventory of repair parts.

In modeling these situations, one restrict his/her attention to the constant failure rate approximation. In this the failure rate is often given in terms of the *mean time between failure* (MTBF), as opposed to the mean time to failure or MTTF. In fact they are both the same number if, when a system fails it is assumed to be repaired immediately to an as-good-as-new condition.

We first consider the times at which the failures take place, and therefore the number that occur within any given span of the time. Suppose that we let $\mathbf{n}$ be a discrete random variable representing the number of failures that take place between $t = 0$ and a time t. Let

$$p_n(t) = P\{\mathbf{n} = n|t\} \tag{2.25}$$

be the probability that exactly n failures have taken place before time t. Clearly, if we start counting failures at time zero, we must have

$$p_0(0) = 1, \tag{2.26}$$

$$p_n(0) = 0, \quad n = 0, 1, 2, \ldots, \infty \tag{2.27}$$

In addition, at any time

$$\sum_{n=0}^{\infty} p_n(t) = 1. \tag{2.28}$$

For small Δt, let failure $\lambda \Delta t$ be the probability that the $(n+1)$th failure will take place during the time increment between t and $t + \Delta t$, given that exactly n failures have taken place before time t. Then the probability that no failure will occur during Δt is $1 - \lambda \Delta t$. From this we see that the probability that no failures have occurred before $t + \Delta t$ my be written as

$$p_0(t + \Delta t) = (1 - \lambda \Delta t) p_0(t). \tag{2.29}$$

Then noting that

$$\frac{d}{dt} p_n(t) = \lim_{\Delta t \to 0} \frac{p_n(t + \Delta t) - p_0(t)}{\Delta t} \tag{2.30}$$

we obtain the simple differential equation

$$\frac{d}{dt} p_0(t) = -\lambda p_0(t) \tag{2.31}$$

Using the initial condition, Eq. 2.26, we find

$$p_0(t) = e^{-\lambda t}. \tag{2.32}$$

With $p_0(t)$ determined, we may now solve successively for $p_n(t), n = 1, 2, 3, \ldots$ in the following manner.We first observe that if n failures have taken place before time t, the probability that the $(n+1)$th failure will take place between t and $t + \Delta t$ is $\lambda \Delta t$. Therefore, since this transition probability is independent of the number of previous failures, we may write

$$p_n(t + \Delta t) = \lambda \Delta t) p_{n-1}(t) + (1 - \lambda \Delta t) p_n(t). \tag{2.33}$$

The last term accounts for the probability that no failure takes place during Δt. For sufficiently small Δt we can ignore the possibility of two or more failures taking place.
Using the definition of the derivative once again, we may reduce Eq. **??** to the differential equation

$$\frac{d}{dt} p_n(t) = -\lambda p_n(t) + \lambda p_{n-1}(t) \tag{2.34}$$

with the following solution

$$e^{\lambda t} p_n(t) - p_n(0) = \lambda \int_0^t p_{n-1}(t') e^{\lambda t'} dt'. \tag{2.35}$$

But since from Eq. 2.27 $p_n(0) = 0$, we have

$$p_n(t) = \lambda e^{-\lambda t} \int_0^t p_{n-1}(t') e^{\lambda t'} dt'. \tag{2.36}$$

This recursive relationship allows us to calculate the p_n successively. For p_1, insert Eq. 2.32 on the right-hand side and carry out the integral to obtain

$$p_1(t) = \lambda t e^{-\lambda t} \tag{2.37}$$

and for all $n \geq 0$ the Eq. 2.36 is satisfied by

$$p_1(t) = \frac{(\lambda t)^n}{n!} e^{-\lambda t} \tag{2.38}$$

and these quantities in turn satisfy the initial conditions given by Eqs. 2.26 and 2.27.
The probabilities $p_n(t)$ are the same as the Poisson distribution with $\mu = \lambda t$. Thus it is possible to determine the mean and the variance of the number n of events occurring over a time span t. Thus the expected number of failures during time t is

$$\mu_n \equiv E\{n\} = \lambda t, \tag{2.39}$$

and the variance of n is

$$\sigma_n^2 = \lambda t. \tag{2.40}$$

Since $p_n(t)$ are the probability mass functions of a discrete variable $\mathbf{n}$, we must have,

$$\sum_{n=0}^{\infty} p_n(t) = 1 \tag{2.41}$$

The number of failures can be related to the mean time between failures by

$$\mu_n = \frac{t}{\text{MTBF}} \tag{2.42}$$

This is the expression that relates μ_n and the MTBF assuming a constant failure rate.
In general, the MTBF may be determined from

$$\text{MTBF} = \frac{t}{n}, \tag{2.43}$$

where n, the number of failures, is large.

The probability that more than N failures have occurred is

$$P\{\mathbf{n} > N\} = \sum_{n=N+1}^{\infty} \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \tag{2.44}$$

Instead of writing this infinite series, however, we may use Eq. 2.41 to write

$$P\{\mathbf{n} > N\} = 1 - \sum_{n=0}^{N} \frac{(\lambda t)^n}{n!} e^{-\lambda t}. \tag{2.45}$$

2.2 Safety

While using various techniques are often effective in increasing reliability, they do not necessarily increase safety. Under some conditions they may even reduce safety. For example, increasing the burst-pressure to working-pressure ratio of a tank introduces new dangers of an explosion or chemical reaction in the event of a rapture. Assuming that reliability and safety are synonymous is hence true in special cases. In general, safety has a broader scope than failures, and failures may not compromise safety. There is obviously an overlap between reliability and safety (Leveson 1995) (figure 2.1); manly accidents may occur without component failure - the individual components where operating exactly as specified or intended, that is, without failure. The opposite is also true - Components may fail without a resulting accident. Accidents may be caused by equipment operation outside the parameters and time limits upon which the reliability analysis are based. Therefore, a system may have high reliability and still have accidents. Safety is an emergent property that arises at the system level when components are operating together.

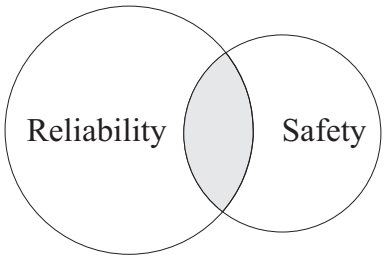

Fig. 2.1. Safety and reliability are overlapping, but are not identical.

For a safety-related system all aspects of reliability, availability, maintainability, and safety (RAMS) have to be considered as they are relevant to the responsibility of manufacturers and the acceptability of the customers. Safety and reliability are generally achieved by a combination of

- fault prevention

- fault tolerance
- fault detection and diagnosis
- autonomous supervision and protection

There are two stages for fault prevention: *fault avoidance* and *fault removal*. Fault avoidance and removal has to be accomplished during the design and test phase.

2.3 Hazard Analysis

Hazard analysis is the essential part of any effective safety program, providing visibility and coordination (Leveson 1995) (see figure 2.2). Information flows both

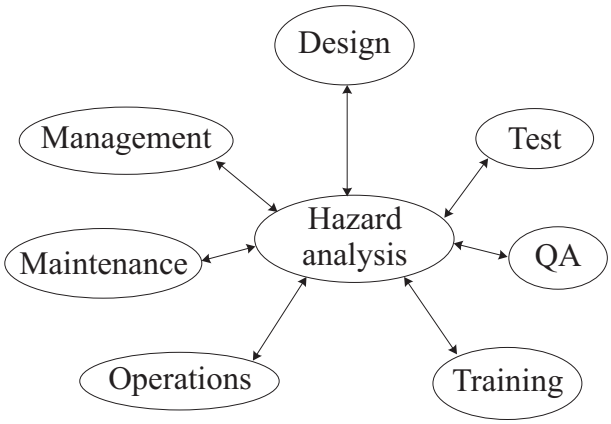

Fig. 2.2. Hazard analysis provides visibility and coordination.

outward from and back into the hazard analysis. The outward information can help designers perform trade studies and eliminate or mitigate hazards and can help quality assurance identify quality categories, acceptance tests, required inspections, and components that need special care. Hazard analysis is a necessary step before hazards can be eliminated or controlled through design or operational procedures. Performing this analysis, however, does not ensure safety.
Hazard analysis is performed continuously throughout the life of the system, with increasing depth and extent as more information is obtained about the system design. As the project progress, the use of hazard analysis will vary - such as identifying hazards, testing basic assumptions and various scenarios about the system operation, specifying operational and maintenance tasks, planning training programs, evaluation potential changes, and evaluating the assumptions and foundation of the models as the system is used and feedback is obtained.

The various hazard analysis techniques provide formalisms for systematizing knowledge, draw attention to gaps in knowledge, help prevent important consideration being missed, and aid in reasoning about systems and determining where improvements are most likely to be effective.

The way of choosing the appropriate hazard analysis process depends on the goals or purposes of the hazard analysis. The goals of safety analysis are related to three general tasks:

1. **Development**: the examination of a new system to identify and assess potential hazards and eliminate or control them.
2. **Operational management**: the examination of an existing system to identify and assess hazards in order to improve the level of safety, to formulate a safety management policy, to train personnel, and to increase motivation for efficiency and safety of operation.
3. **Certification**: the examination of a planned or existing system to demonstrate its level of safety in order to be accepted by the authorities or the public.

The first two tasks have the common goal of making the system safer (by using appropriate techniques to engineer safer systems), while the third task has the goal of convincing management or government licenser that an existing design or system is safe (which is also called **risk assessment**).

2.3.1 Steps in the Hazard analysis process

A hazard analysis consists of the following steps:

1. Definition of objectives.
2. Definition of scope.
3. Definition and description of the system, system boundaries, and information to be used in the analysis.
4. Identification of hazards.
5. Collection of data (such as historical data, related standards and codes of practice, scientific tests and experimental results).
6. Qualitative ranking of hazards based on their potential effects (immediate or protracted) and perhaps their likelihood (qualitative or quantitative).
7. Identification of caused factors.
8. Identification of preventive or corrective measures and general design criteria and controls.
9. Evaluation of preventive or corrective measures, including estimates of cost. Relative cost ranking may be adequate.
10. Verification that controls have been implemented correctly and are effective.
11. Quantification of selected, unresolved hazards, including probability of occurrence, economic impact, potential losses, and costs of preventive or corrective measures.
12. Quantification of residual risk.

13. Feedback and evaluation of operational experience.

Not all of the steps are needed to be performed for every system and for every hazard. For new systems designs, usually the first 10 steps are necessary.

2.3.2 System model

Every system analysis requires some type of model of the system, which may change from a fuzzy idea in the analysts mind to a complex and carefully specified mathematical model. A model is a representation of a system that can be manipulated in order to obtain information about the system itself. Modeling any system requires a description of the following:

- *Structure*: The interrelationship of the parts along some dimension(s), such as space, time, relative importance, and logic and decision making properties:
- *Distinguishing qualities*: a qualitative description of the particular variables and parameters the characterize the system and distinguish it from similar structure.
- *Magnitude, probability, and time*: a description of the variables and parameters with the distinguishing qualities in terms of their magnitude or frequency over time and their degree of certainty for the situations or conditions of interest.

2.3.3 Hazard level

The hazard category or level is characterized by (1) *severity* (sometimes also called *damage*) and (2) *likelihood* (or *frequency*) of occurrence. Hazard level is often specified in form of a *hazard criticality index matrix* to aid in prioritization as it is shown in figure 2.3. Hazard criticality index matrix combines hazard severity and hazard probability/frequency.

Hazard consequence classes/categories. Categorizing the hazard severity reflects worst possible consequences. Different categories are used depending on the industry or even the used system. Some examples are:

Catastrophic – Critical – Marginal – Negligible

or

Major – Medium – Minor

or simply

Category 1 – Category 2 – Category 3.

Consequence classes can vary from *Acceptable* to *Unacceptable* as it is shown in figure 2.3. Severity of hazard (for a given production plant) is considered with regard to:

- Environmental damages
- Production line damages
- Lost production

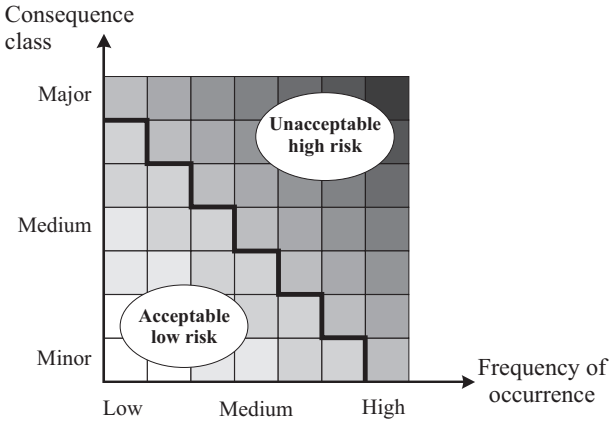

Fig. 2.3. The hazard level shown in form of hazard criticality index matrix: hazard consequence classes versus frequency of occurrence

- Lost workforce
- Lost company image
- Insurance (premium)

The basis for defining hazard consequence classes is the company policy in general. Hence, definition of hazard consequence classes are not application specific, i.e. different companies can define different consequence classes for the same case.

Hazard likelihood/frequency of occurrence. The hazard likelihood of occurrence can be specified either qualitatively or quantitatively. During the design phase of a system, due to the lack of detailed information, accurate evaluation of hazard likelihood is not possible. Qualitative evaluation of likelihood is usually the best that can be done.

Accepted frequencies of occurrence. The basis for defining *accepted frequencies of occurrence* depends on the company policy on the issue. So, the definition of accepted frequencies of occurrence is not application specific (can vary from company to company). Examples of ranking of frequencies of occurrence are:

Frequent – Probable – Occasional – Remote – Improbable - Impossible

or

High – Medium – Low

By specifying hazard consequence categories and frequencies of occurrence one can draw the hazard criticality index matrix as shown in figure 2.3.

Specifying the acceptable Hazard (risk) level is management's responsibility. This corresponds to drawing the thick line in the figure. Following issues have impact on how to specify the acceptable risk levels:

- □ National regulations
- □ Application of methods specified in standards,
- □ Following the best practice applied by major industrial companies,
- □ By drawing comparison between socially accepted risks (e.g. risk of driving by car, airplane,...),
- □ Policy to propagate the company image.

2.4 Hazard analysis models and techniques

There exists various methods for performing hazard analysis, each having different coverage and validity. Hence, several may be required during the life of the project. It is important to notice that none of these methods alone is superior to all others for every objective and even applicable to all type of systems. Furthermore, very little validation of these techniques has been done. The most used techniques, which are listed below, are described extensively in (Leveson 1995):

- ⇒ Checklists,
- ⇒ Hazard Indices,
- ⇒ Fault tree analysis (FTA),
- ⇒ Management Oversight and Risk Tree analysis (MORT),
- ⇒ Even Tree Analysis (ETA),
- ⇒ Cause-Consequence Analysis (CCa),
- ⇒ Hazard and Operability Analysis (HAZOP),
- ⇒ Interface Analysis,
- ⇒ Failure Mode and Effect Analysis (FMEA),
- ⇒ Failure Mode Effect, and Critically Analysis (FMEAC),
- ⇒ Fault Hazard Analysis (FHA),
- ⇒ State Machine Hazard Analysis,

In this report, FMEA and FMECA are described more in details, as they have been applied ((Bøgh, Izadi-Zamanabadi, and Blanke 1995)).

2.4.1 Failure Mode and Effect Analysis (FMEA)

This technique was developed by reliability engineers to permit them to predict equipment reliability. This is a hierarchical, bottom-up, inductive analysis technique, where the initiating events are failures of individual components.
The objectives of FMEA are as follows (Russomanno, Bonnell, and Bowles 1994)

- – to provide a systematic examination of potential system failures;
- – to analyze the effect of each failure mode on system operation;
- – to access the safety of various systems or components, and
- – to identify corrective actions or design modifications.

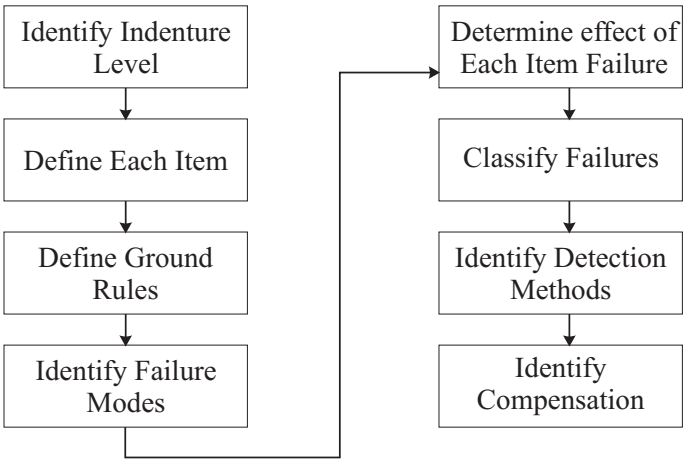

Fig. 2.4. Major logical steps in the FMEA analysis

The analysis considers the effects of a single item failure on overall system operation and spans a myriad of applications, including electrical, mechanical, hydraulic, and software domains. Before the FMEA analysis begins, a thorough understanding of the system components, their relationships, and failure modes must be determined and represented in a formalized manner. The indenture level of the analysis (i.e. the item levels that describe the relative complexity of assembly or function) must coincide with the stage of design. The indenture level progresses from the most general description; then, properties are added and the indenture level becomes more specialized. The FMEA must be conducted at multiple indenture level; hence a failure mode can be identified and traced to the most specialized assembly for ease of repair or design modification. Figure 2.4 outlines the major logical steps in the analysis.
The possible problems with the application of FMEA (also FMECA) are included in the following:

1. Engineers have insufficient time for proper analysis
2. Analysts have a poor understanding of FMEA and designers are inadequately trained in FMEA.
3. FMEA is perceived as a difficult, laborious, and mundane task.
4. FMEA knowledge tends to be restricted to small groups of specialists.
5. Computerized aids are needed to decrease the effort required to produce a FMEA.

2.4.2 Failure Mode, Effect and Critically Analysis (FMECA)

Failure Mode, Effect and Critically Analysis (FMECA) is basically a FMEA with more detailed analysis of the criticality of the failure. In this analysis criticalities or priorities are assigned to failure mode effects. Figure 2.5 shows an example of tables used in this analysis.

Failure Modes and Effects Critically Analysis

Subsystem ———— Prepared by ———— Date ————

Item	Failure Modes	Cause of Failure	Possible Effects	Prob.	Level	Possible action to Reduce Failure Rate or Effects

Fig. 2.5. A simple FMECA scheme

2.4.3 Fault Tree Analysis (FTA)

FTA is primarily a means for analyzing causes of hazards, not identifying hazards. The top event in the tree must have been foreseen and thus identified by other techniques. FTA is, hence, a top-down search method. Once the tree is constructed, it can be written as Boolean expression and simplified to show the specific combination of basic events that are necessary and sufficient to cause the hazardous event.

Fault-tree evaluation by cut sets. The direct evaluation procedures just discussed allow us to assess fault trees with relatively few branches and basic events. When larger trees are considered, both evaluation and interpretation of the results become more difficult and digital computer codes are invariably employed. Such codes are usually formulated in terms of the minimum cut-set methodology discussed in this section. There are at least two reasons for this. First, the techniques lend themselves well to the computer algorithms, and second, from them a good deal of intermediate information can be obtained concerning the combination of component failures that are pertinent to improvements in system design and operations.
The discussion that follows is conveniently divided into qualitative and quantitative analysis. In qualitative analysis information about the logical structure of the tree is used to locate weak points and evaluate and improve system design. In quantitative analysis the same objectives are taken further by studying the probabilities of component failures in relation to system design.

Qualitative analysis. In the following the idea of minimum cut-sets is introduced and then it is related to th qualitative evaluation of of fault trees. We then discuss briefly how the minimum cut sets are determined for large fault trees. Finally, we discuss their use in locating system weak points, particularly possibilities for common-mode failures.

minimum cut-set formulation. A minimum cut set is defined as the smallest combination of primary failures which, if they occur, will cause the top event to occur. It, therefore, is a combination (i.e. intersection) of primary failures sufficient to cause the top event. It is the smallest combination in that all the failures must take place for the top event to occur. If even one of the failures in the minimum cut set does not happen, the top event will not take place.

The terms minimum cut set and failure mode are sometimes used interchangeably. However, there is a subtle difference that we shall observe hereafter. In reliability calculations a failure mode is a combination of component or other failures that cause the system to fail, regardless of the consequences of the failure. A minimum cut set is usally more restrictive, for it is the minimum combination of failures that causes the top event as defined for a particular fault tree. If the top event is defined broadly as system failure, the two are indeed interchangeable. Usually, however, the top event encompasses only the particular subset of system failures that bring about a particular safety hazard.

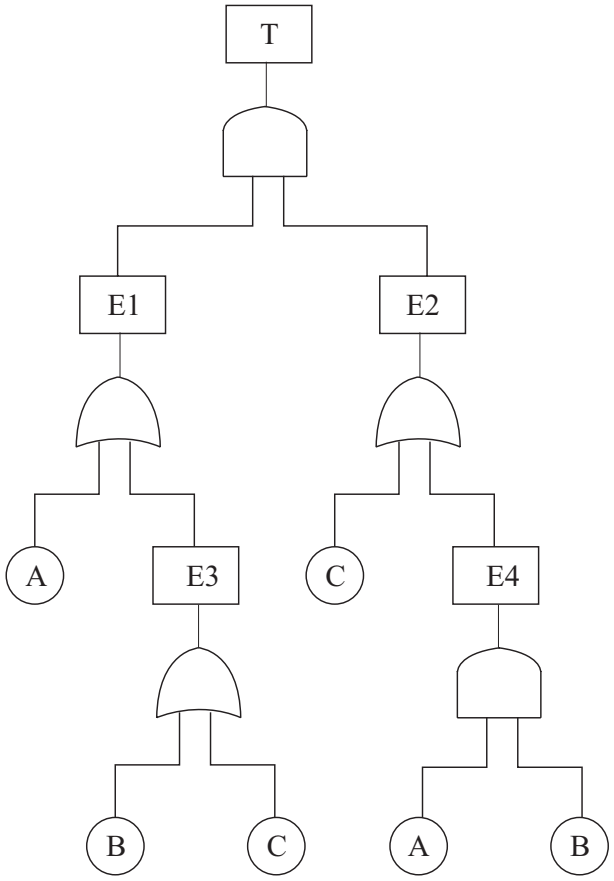

Fig. 2.6. Example of a fault tree

The origin for using the term cut set may be illustrated graphically using the reduced fault tree in Fig. 2.7. The reliability block diagram corresponding to the tree is shown in Fig. 2.6. The idea of a cut set comes originally from the use of such

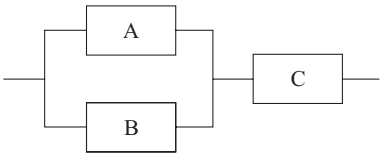

Fig. 2.7. Minimum cut-sets on a reliability block diagram.

diagrams for electric apparatus, where the signal enters at the left and leaves at the right. Thus the minimum cut set is the minimum number of components that must be cut to prevent the signal flow. There are two minimum cut sets, M_1 consisting of components A and B, and M_2, consisting of component C.

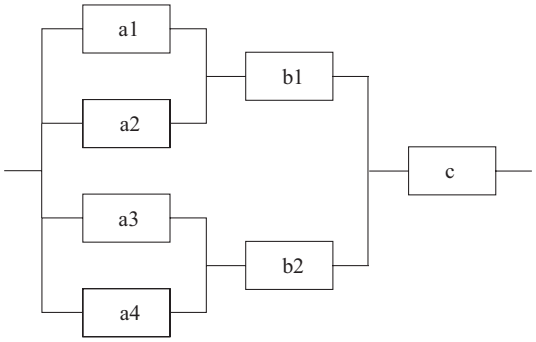

Fig. 2.8. Minimum cut-sets on a reliability block diagram of a seven component system.

Practice: Find the minimum cut sets in Fig. 2.8.

For larger systems, particularly those in which the primary failures appear more than once in the fault tree, the simple geometrical interpretation becomes problematical. However, the primary characteristics of the concept remain valid. It permits the logical structure of the fault tree to be presented in a systematic way that is amenable to interpretation in terms of the behavior of the minimum cut-sets.

Suppose that the minimum cut sets of a system can be found. The top event, system failure, may then be expressed as the union of these sets, Thus if there are N minimum cut sets,

$$T = M_1 \cup M_2 \cup \cdots \cup M_N. \tag{2.46}$$

Each minimum cut set then consists of the intersection of the minimum number of primary failures required to cause the top event. For example, the minimum cut sets for the system shown in Fig. 2.8 are

$$\begin{aligned} M_1 &= c \qquad\qquad M_3 = a1 \cap a2 \cap b2 \\ M_2 &= b1 \cap b2 \qquad M_4 = a3 \cap a4 \cap b1 \\ & \qquad M5 = a1 \cap a2 \cap a3 \cap a4. \end{aligned} \tag{2.47}$$

Before proceeding, it should be pointed out that there are other cut sets that will cause the top event, but they are not minimum cut sets. These need not be considered, however, because they do not enter the logic of the fault tree. By the rules of Boolean algebra they are absorbed into the minimum cut sets.

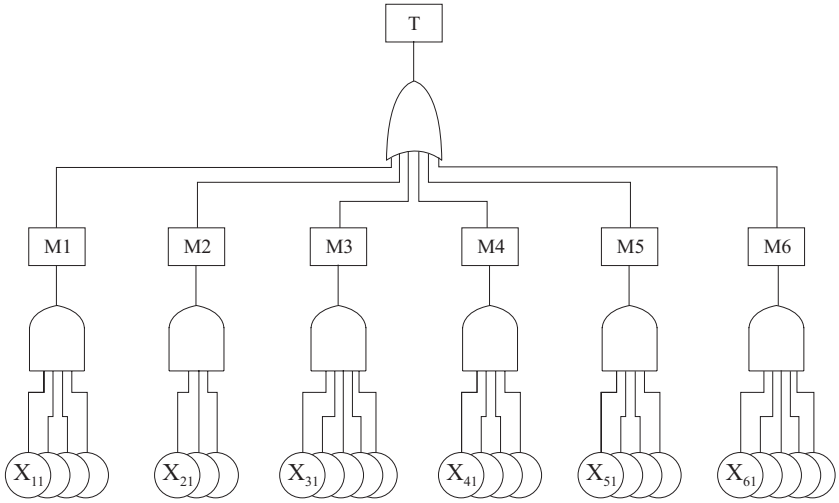

Fig. 2.9. Generalized Minimum cut-set representation of a fault tree.

Since we are able to write the top event in terms of minimum cut sets as in Eq. 2.46, we may express the fault tree in the standardized form shown in Fig. 2.9. In this X_{mn} is the nth element of the mth minimum cut set. Not that the same primary failures may often be expected to occur in more than one of the minimum cut sets. Thus the minimum cut sets are not generally independent of one another.

Cut-Set Interpretations. Knowing the minimum cut sets for a particular fault tree can provide valuable insight concerning potential weak points of complex systems, even when it is not possible to calculate the probability that either a particular cut set or the top event will occur. Three qualitative considerations, in particular, may be useful:

– the ranking of the minimal cut sets by the number of primary failures required
– the importance of particular component failures to the occurrence of the minimum cut sets, and
– the susceptibility of the particular cut sets to common-mode failures.

Minimum cut sets are normally categorized as singlets, doublets, triplets, and so on, according to the number of primary failures in the cut set. Emphasis is then put on eliminating the cut sets corresponding to small number of failures, for ordinarily these may be expected to make the largest contributions to system failure.

The common design criterion that no single component failure should cause system failure is equivalent to saying that all singlets must be removed from the fault tree for which the top event is system failure. Indeed, if component failure probabilities are small and independent, then provided that they are of same order of magnitude, doublets will occur much less frequently than singlets, triplets much less than doublets, and so on.

A second application of cut-set information is in assessing qualitatively the importance of a particular component. Suppose that we wish to evaluate the effect on the system of improving the reliability of a particular component, or conversely, to ask whether, if a particular component fails, the system-wide effect will be considerable. If the component appears in one or more of the lower order cut-sets, say singlets or doublets, its reliability is likely to have a pronounced effect. On the other hand, if it appears only in minimum cut-sets requiring several independent failures, its importance to system failure is likely to be small.

These arguments can rank minimum cut-set and component importance, assuming that the primary failures are independent. If they are not, that is, if they are susceptible to common-mode failure, the ranking of cut-set importance may be changed. If five of the failures in a minimum cut-set with six failures, for example, can occur as the result of a common cause, the probability of the cut-set's occurring is more comparable to that of a doublet.

Extensive analysis is often carried out to determine the susceptibility of minimum cut-sets to common-cause failures. In an industrial plant one cause might be fire. If the plant is divided into several fire-resistant compartments, the analysis might proceed as follows. All the primary failures of equipment located in one of the compartments that could be caused by fire are listed. Then these components would be eliminated from the minimum cut-sets (i.e. they would be assumed to fail). The resulting cut-sets would then indicate how many failures- if any- in addition to those caused by the fire, would be required for the top event to happen. Such analysis is critical for determining the layout of the plant that will best protect it from a variety of sources of damage: fire, flooding, collision, earthquake, and so on.

2.4.4 Risk

A more detailed definition of risk is given in the following (Leveson 1995):

> *Risk* is the *hazard level* combined with (1) the likelihood of the hazard leading to an accident (sometimes called *danger*) and (2) hazard exposure and duration (sometimes called *latency*).

Exposure or duration of a hazard is a component of risk: Since an accident involves a coincidence of conditions, of which the hazard is just one, the longer the hazardous state exists, the greater the chance that the other prerequisite conditions will occur.

The terms risk analysis and hazard analysis are sometimes used interchangeably, but an important distinction exists. *Hazard analysis* involves only the identification of hazards and the assessment of hazard level, while *risk analysis* adds the identification and assessment of the environmental conditions along with exposure or duration. Thus, hazard analysis is a subset of risk analysis.

Risk Reduction. Dangerous failures can undergo risk classification in order to determine their required safety integrity level and to decide whether risk reduction is necessary. To categorize risk reduction level, the term *Safety Integrity Levels* (SIL) have been defined, which allocate risk reduction factors to four predefined safety levels SIL1 – SIL4. The Safety Integrity Level is hence a measure for applying supervision and safety methods to reduce the risk. An example for the relation between safety integrity level and risk reduction factor (RRF) is illustrated in the following table:

Safety Integrity Level	Safety Availability Required	Equivalent RRF
4	$> 99.99\%$	> 10000
3	$99.9 - 99.99\%$	$1000 - 10000$
2	$99 - 99.9\%$	$100 - 1000$
1	$90 - 99\%$	$10 - 100$

Table 2.1. Safety Integrity Level vs. Risk Reduction Factor

In order to determine the safety integrity level following factors/parameters are used:

C: Consequence risk parameter
F: Frequency and exposure risk parameter
P: Possibility of avoiding risk parameter
W: Probability of unwanted occurrence

and SIL is computed as:

$$\text{SIL} = C \times F \times P \times W$$

An example of categorization of these parameters is given below:

C0: Slight damage to equipment
C1: One injury
C2: One death
C3: Several deaths

F1: Small probability of persons present in the dangerous zone
F2: High probability of persons present in the dangerous zone

P1: Good chance to avoid the hazard

P2: Highly impossible to avoid the hazard

W1: Probability of hazardous event very small (Frequency of occurrence)
W2: Probability of hazardous event small
W3: Probability of hazardous event high

Risk graph. The safety integrity level can be computed by using the risk graph as it is shown in Figure 2.10.

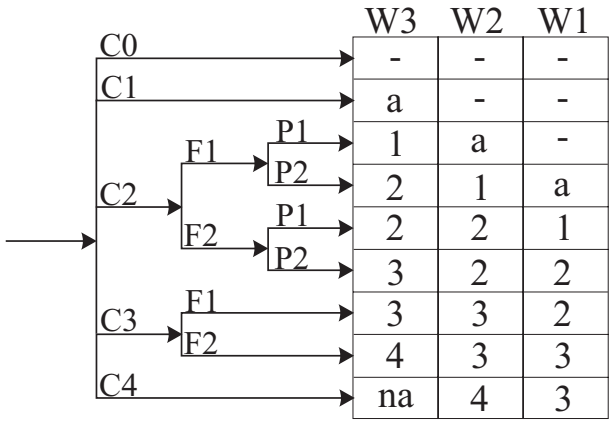

Fig. 2.10. A risk graph that is used to determine related safety integrity level. *a* : acceptable, *na*: not acceptable.

Example 1- Hazard A:.

- Hazard with probably a causality (C2),
- Large probability of persons present (F2), assume %90.
- No possibility to avoid the hazard (P2), assume %100.
- Frequency of occurrence, assume once per 10 years, (W2).
  Calculations:
  - C2*F2*P2*W2 = 1*0.9*1*0.1 = 0.09, or **9** causalities per 100 years
  - Required protection **SIL 2**.

Example 2 - Hazard B:.

- Hazard with probably several causalities, assume 5 causalities, (C3)
- Small probability of persons present (F1), assume %10.
- Frequency of occurrence, assume once per 10 years, (W2).
  Calculations:
  - C2*F1*W2 = 5*0.10*0.1 = 0.05, or **5** causalities per 100 years

– Required protection **SIL 3**.

For each SIL one should determine the required safety measures (preventive or corrective) in advance. The unavoidable failures must be covered by maintenance and on-line supervision and safety methods during operation, including protection and supervision with fault detection and diagnosis and appropriate safety actions.

2.5 Reliability and safety assessment - an overview

In the following two subsections the procedural steps used/needed to carry out reliability and safety assessments are summarized. A more detailed review can be found in (Bøgh 2000).

2.5.1 Reliability Assessment

Purpose: To analyze potential component failures and operator errors and thereby to predict the *availability* of equipments in different operational modes.

2.5.2 Steps in reliability assessment

- Qualitative reliability assessment:
  - Preliminary RAM evaluation
- Quantitative reliability assessment
  - Structural analysis
  - Operational analysis
  - Failure consequence analysis
  - Reliability calculation
  - RAM assessment

2.5.3 Safety Assessment

Purpose: To identify potential hazards and evaluate the risk of accidents associated with each hazard.

2.5.4 Steps in safety assessments

- Functional analysis
  - Functional modeling
  - Functional failure analysis
- Hazard analysis
  - Hazard identification
  - Frequency identification (risk estimation)
  - Severity identification

  - Hazard screening and risk ranking
- Risk assessment
  - Frequency analysis
  - Consequence analysis
  - Risk assessment based on the evaluated severity.

3. ACTIVE FAULT-TOLERANT SYSTEM DESIGN

3.1 Introduction

Fault-tolerant methods aim at minimizing the frequency of fault occurrence (W). The challenge that faces the engineers is to design mass-produced fault-tolerant sensors, actuators, micro-computers, and bus communication systems with hard real-time requirements with reasonable costs. In the following fault-tolerant design at component and unit level (Isermann, Schwarz, and Stölzl 2000) is discussed. The reliability degree of two most used hardware redundancies, namely cold and hot standby, is considered. In the last section of this chapter, the main issue which is a consistent and formal procedure to design an active fault tolerant (control) system is introduced.

3.2 Type of faults

The starting point for fault-tolerant design is knowledge about the possible faults that may occur in a unit/component. Type of fault characterizes the behavior for various components. They may be distinguished by their form, time and extent.
The *form* can be either systematic or random.
The *time behavior* may be described by permanent, transient, intermittent, noise, or drift (see fig. 3.1).
The *extent* of faults is either local or global and includes the size. *Electronic hard-*

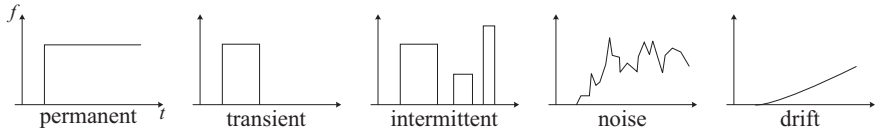

Fig. 3.1. Time behavior of fault types

ware exhibit systematic faults if they originate from mistakes that have been made during specification or design phase. During the operation, the faults in electronic hardware components are mostly random with various type of time behavior.

The faults in *software* (bugs) are usually systematic, due to wrong specifications, coding, logic, calculation overflow, etc. Generally, they are not random like faults in the hardware.

Failures in *mechanical systems* can be classified into the following failure mechanism: distortion (buckling, deformation), fatigue and fracture (cycle fatigue, thermal fatigue), wear (abrasive, adhesive, caviation), or corrosion (galvanic, chemical, biological). They may appear as drift like changes (wear, corrosion), or abruptly (distortion, fracture) at any time or after stress.

Electrical systems consist normally of large number of components with various failure modes, like short cuts, parameter changes, loose or broken connections, EMC problems, etc. Generally electrical faults appear more randomly that mechanical faults.

3.3 Hardware fault-tolerance

Fault tolerance methods use mainly redundancy, which means that in addition to the used component, additional modules/components are connected. The redundant modules are either *identical* or *diverse*. Such redundancy can be materialized for hardware, software, information processing, mechanical and electronic components such as sensors, actuators, microcomputers, buses, power supplies, etc.

There exists mainly two basic approaches for realization of fault tolerance: *Static redundancy* and *dynamic redundancy*.

In *static redundancy* scheme, the idea is to use three or more modules which have the same input signals and all are active as it is shown in figure 3.2 a). Their outputs are connected to a voter that compares these signals. The correct signals then are chosen by majority voting. For instance, when a triple modular redundant system is used, the fault in one of the modules generate wrong outputs. The faulty module can be masked by 2-out-of-3 voting. This scheme can thus handle (tolerate) one fault. Generally speaking, n (n odd) modules can tolerate $(n-1)/2$ faults.

Dynamic redundancy uses less number of modules on cost of more information processing. A minimal configuration uses 2 modules as it is illustrated in figures 3.2 b) and 3.2 c). One module is usually in operation and if it fails the standby or backup unit takes over. This requires a fault detection unit to detect the faulty situations. Simple fault detection modules use the output signal for consistency checking (range check, slew rate check, RMS check), comparison with redundant modules or use of information redundancy in computers like parity checking or watchdog timers. The task of the reconfiguration module is to switch to the standby module from the faulty one after the fault is detected.

In "*hot standby*" arrangement, shown in figure 3.2 b), both modules are operating continuously. The transfer time is short, but the price is the operational aging (wear out) of the standby module.

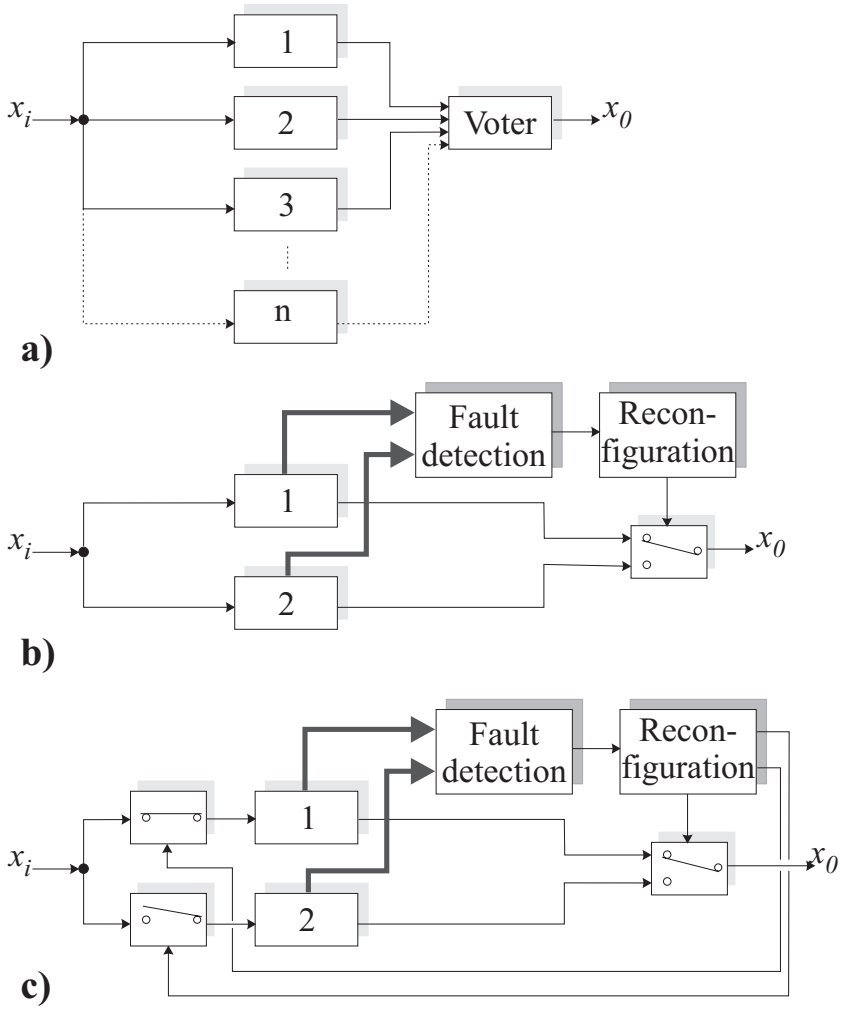

Fig. 3.2. Fault-tolerant schemes for electronic hardwares a) Static redundancy b) Dynamic redundancy (hot standby) c) Dynamic redundancy (cold standby)

In "*cold standby*" arrangement, shown in figure 3.2 c), the standby module is out of function and hence does not wear. In this arrangement two more switches at the input are needed and more transfer time is needed due to the start-up procedure. For both hot and cold standby schemes, the performance of the fault detection module is essential.

For digital computers (microcomputers) with only a requirement for fail-safe behavior, a duplex configuration like figure 3.3 can be applied. The output signals of two synchronized processors are compared in two comparators (software) which

act on two switches of one of the outputs. The purpose of this configuration is to bring the system in fail-safe state (Used in ABS braking systems).

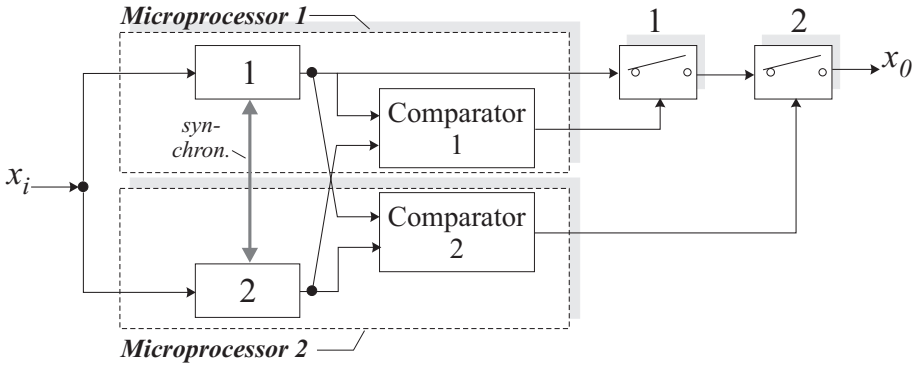

Fig. 3.3. Duplex computer system with dynamic redundancy in hot standby, fault detection with comparators, switch to possible fail-safe (not fault-tolerant)

. .

3.3.1 Mechanical and Electrical systems

For *mechanical systems* static redundancy is very often used in all kinds of homogeneous and inhomogeneous materials (e.g. metals and fibers) and special mechanical constructions like lattic structures, spoke-wheels, dual tires. Since the inputs and outputs are not signals but rather, forces, electrical current, or energy flows, the voter does not exists as it is shown in Figure 3.4. In case of failure in a component, the others automatically take over a higher force or current (called *stressful degradation*).

For *electronic systems* static redundancy is materialized in components in form of multiple wiring, multiple coil windings, multiple brushes for DC motors, or multiple contacts for potentiometers.

Mechanical and electronic systems with *dynamic redundancy*can also be built (see Figure 3.2 c)). Fault detection can be based on measured outputs. To improve the quality of fault diagnosis one should be able to measure inputs and other intermediate signals. Dynamic redundancy is normally applied for electro-mechanical systems.

3.4 Reliability considerations for standby configurations

Hot and cold standby configuration is also known as *active parallel* and *standby parallel* in reliability related literatures. To highlight the differences again, it should

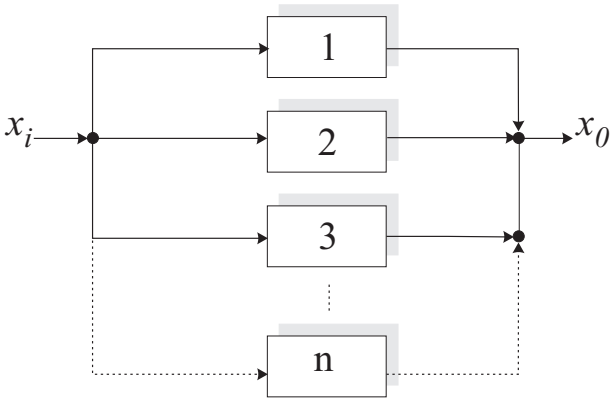

Fig. 3.4. Static redundancy for mechanical (and electrical) systems

. .

be mention that in two-unit hot standby (or active parallel) configuration both units are employed and therefore subject to failure from the onset of operation. In cold standby (or standby parallel) configuration the second unit is not brought into the operation until the first fails, and therefore can not fail until a later time. In the following the reliabilities for the idealized configurations are derived. Similar considerations also arise in treating multiple redundancy with three or more parallel units, but will not be considered here.

– *Hot standby or active parallel configuration*
The reliability $R_h(t)$ of a two-unit hot standby system is the probability that either unit 1 or unit 2 will not fail until a time greater than t. Designating random variables $\mathbf{t}_1$ and $\mathbf{t}_2$ to represent the failure times we have

$$R_h(t) = P\{\mathbf{t}_1 > t \cup \mathbf{t}_2 > t\}. \tag{3.1}$$

This yields

$$R_h(t) = P\{\mathbf{t}_1 > t\} + P\{\mathbf{t}_2 > t\} - P\{\mathbf{t}_1 > t \cap \mathbf{t}_2 > t\}. \tag{3.2}$$

Assuming that the failures in the units can occur independently from each other one can replace the last term in Eq. 3.2 by $P\{\mathbf{t}_1 > t\}P\{\mathbf{t}_2 > t\}$. Denoting the reliabilities of the units as

$$R_i(t) = P\{\mathbf{t}_i > t\}, \tag{3.3}$$

we may then write

$$\boxed{R_h(t) = R_1(t) + R_2(t) - R_1(t)R_2(t)}. \tag{3.4}$$

– *Cold standby or standby parallel configuration*
In this case the failure time $\mathbf{t}_2$ of the standby unit is dependent on the failure time $\mathbf{t}_1$ of the primary unit. Only the second unit must survive to time t for the system to survive, but with the condition that it can not fail until after the fist unit fails. Hence

$$R_c(t) = P\{\mathbf{t}_2 > t \mid \mathbf{t}_2 > \mathbf{t}_1\}. \tag{3.5}$$

There are two possibilities. Either the first unit doesn't fail, i.e. $\mathbf{t}_1 > t$, or the first unit fails and the standby unit doesn't, i.e. $\mathbf{t}_1 < t \cap \mathbf{t}_2 > t$. Since these to possibilities are mutually exclusive, we my just add the possibilities,

$$R_c(t) = P\{\mathbf{t}_1 > t\} + P\{\mathbf{t}_1 < t \cap \mathbf{t}_2 > t\}. \tag{3.6}$$

The first term is just $R_1(t)$, the reliability of the primary unit. The second term has the following interpretation. Suppose that the PDF for the primary unit is $f_1(t)$. Then the probability of unit 1 failing between t' and $t' + dt'$ is $f_1(t')dt'$. Since the standby unit is put into operation at t', the probability that it will survive to time t is $R_2(t - t')$. Thus the system reliability, given that the first failure takes place between t' and $t' + dt'$ is $R_2(t - t')f_1(t')dt'$. To obtain the second term in Eq. 3.6 we integrate primary failure time t' between zero and t:

$$P\{\mathbf{t}_1 < t \cap \mathbf{t}_2 > t\} = \int_0^t R_2(t - t')f_1(t')dt'.$$

The standby system reliability then becomes

$$R_c(t) = R_1(t) + \int_0^t R_2(t - t')f_1(t')dt', \tag{3.7}$$

or using Eq. 2.8 to express PDF in terms of reliability we obtain

$$\boxed{R_c(t) = R_1(t) + \int_0^t R_2(t - t')\frac{d}{dt'}R(t')dt'}. \tag{3.8}$$

– *Constant failure rate models*
Assume that the units are identical, each with a failure rate λ. Equation 2.22, $R(t) = \exp(-\lambda t)$, may then be inserted to obtain

$$\boxed{R_h(t) = 2e^{-\lambda t} - e^{-2\lambda t}} \tag{3.9}$$

for hot standby (active parallel), and

$$\boxed{R_c(t) = (1 + \lambda t)e^{-\lambda t}} \tag{3.10}$$

for cold standby (standby parallel).
The system failure rate can be determined for each of these cases using Eq. 2.13. For the active system we have

$$\boxed{\lambda_h(t) = -\frac{1}{R_h(t)}\frac{d}{dt}R_h(t) = \lambda\left(\frac{1-e^{-\lambda t}}{1-0.5e^{-\lambda t}}\right)}, \tag{3.11}$$

while for the cold standby system

$$\boxed{\lambda_c(t) = -\frac{1}{R_c(t)}\frac{d}{dt}R_c(t) = \lambda\left(\frac{\lambda t}{1+\lambda t}\right)}. \tag{3.12}$$

Two traditional measures are useful in assessing the increased reliability that results from redundant configurations. These are the mean-time-to-failure or MTTF and the rare event estimate for reliability at times which are small compared to the MTTF of single units. The values of the MTTF for hot and cold standby of two identical units are obtained by substituting Eqs. 3.9 and 3.10 into Eq. 2.19. We have

$$\boxed{\text{MTTF}_h = \frac{3}{2}\text{MTTF}}, \tag{3.13}$$

and

$$\boxed{\text{MTTF}_c = 2\text{MTTF}}, \tag{3.14}$$

where MTTF $= 1/\lambda$ for each of the two units. Thus, THERE IS A GREATER GAIN IN MTTF FOR THE COLD STANDBY THAN FOR THE HOT STANDBY SYSTEM.

Frequently, the reliability is of most interest for times that are small compared to the MTTF, since it is within the small-time domain where the design life of most products fall. If the single unit reliability, $R(t) = \exp(-\lambda t)$, is expanded in a power series of λt, we have

$$R(t) = 1 - \lambda t + \frac{1}{2}(\lambda t)^2 - \frac{1}{6}(\lambda t)^3 + \cdots \tag{3.15}$$

The rare event approximation has the form of one minus the leading term in λt. Thus

$$\boxed{R(t) \approx 1 - \lambda t, \qquad \lambda t \ll 1} \tag{3.16}$$

for a single unit. employing the same exponential expansion for the redundant configurations we obtain

$$\boxed{R_h(t) \approx 1 - (\lambda t)^2, \qquad \lambda t \ll 1} \tag{3.17}$$

from Eq. 3.9 and

$$\boxed{R_c(t) \approx 1 - \frac{1}{2}(\lambda t)^2, \qquad \lambda t \ll 1} \tag{3.18}$$

from Eq. 3.10. Hence, FOR SHORT TIMES THE FAILURE PROBABILITY, $1 - R$, FOR A COLD STANDBY SYSTEM IS ONLY ONE-HALF OF THAT FOR A HOT STANDBY SYSTEM.

3.5 Software fault-tolerance

A software system provides services. So, one can classify the a system's failure modes according to the impact that they have on the services it delivers. Two general domains of failure can be identified:

– Value failure - The values associated with the services is in error (constraint error - value error).
– Time failure - the services is delivered in wrong time (too early - too late - infinitely late (*omission failure*)).

These failures are depicted in figure 3.5. Combinations of value and timing failures are often called *arbitrary*.

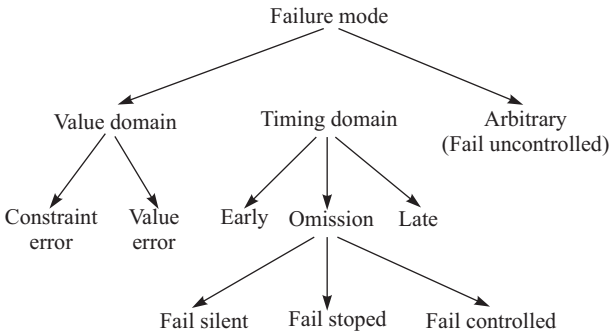

Fig. 3.5. Software failure mode classification.

As in its hardware counterpart redundancy methods also is used in software. *Static redundancy* is normally achieved through *N-version programming*, which is based on the assumptions that a program can be completely, consistently, and unambiguously specified, and that programs that have been developed independently will fail independently. N-version programming (often called *design diversity*) is defined as the independent generation of N ($N \geq 2$) functionally equivalent programs from the same initial specifications. The idea is that N individuals or groups produce the required N versions *without interaction*. The programs execute concurrently with the same inputs and their results are compared by a *driver process*. The correct result is then chosen be consensus (voting).

Dynamic redundancy in software manifest itself in form of *cold standby* scheme, where the redundant component takes over when an error has been detected. There are four steps in software dynamic redundancy:

1. *Error detection*: with following techniques:
   – *Environmental detection*: the errors are detected in the environment in which the program executes.

   - *Application detection*: These errors are detected by application itself. There exists several techniques such as: *Replication checks*, *Timing checks*, *Coding checks*, *Reversal checks*, etc.
2. *Damage confinement and assessment*: which is also called *error diagnosis*. Damage confinement is concerned with structuring the system in order to minimize the damage caused by the faulty component. It is also known as *firewalling*.
3. *Error recovery*: Aims at transforming the corrupted system into a state from which it can continue its normal operation (Degraded functionality is allowed).
4. *Fault treatment and continued service*: Is the maintenance part and aims at finding the occurred fault.

The subject of software fault-tolerance is too extensive to be handled in this note. Interested readers can refer to many available references such as (Burns and Wellings 1997), (Mullender 1993) and references therein.

3.6 Active fault-tolerant control system design

The purpose of using fault-tolerant control system is nicely formulated by Stengel ((Stengel 1993)):

> *Failure- (fault-) tolerance may be called upon to improve system reliability, maintainability and survivability. The requirements for fault-tolerance are different in these three cases. Reliability deals with the ability to complete the task satisfactorily and with a period of time over which the ability is retained. A control system that allows normal completion of tasks after component fault improves reliability. Maintainability concerns the need for repair and the ease with which repairs can be made, with no premium place on performance. Fault-tolerance could increase time between maintenance actions and allow the use of simpler repair procedures. Survivability relates to the likelihood of conducting the operation safely (without danger to the human operator of the controlled system), whether or not the task is completed. Degraded performance following a fault is permitted as long as the system is brought to an acceptable state.*

Faults that can not be avoided must be tolerated (or compensated for) in a way that the system functionality remains intact (degraded level allowed).

3.6.1 Justification for applying fault-tolerance

The main purpose of applying fault-tolerance is to achieve RAM (Reliability, Availability, and Maintainability) under a combination of the following constraints:

→ *Cost* - There is a trade-off between the cost of the preventive maintenance and the saved money from the decreased number of failures. The failure costs would need to include, of course, both those incurred in repairing or replacing the component/subsystem, and those from the loss of production during the unscheduled down-time for repair.
The trade-off decision is heavily dependent on the severity level of the failure consequences, i.e. the risk level; For an aircraft engine the potentially disastrous consequences of engine failure would eliminate repair maintenance as a primary consideration. Concern would be with how much preventive maintenance can be afforded and with the possibility of failures induced by faculty maintenance. The same argument can be used for production lines in heavy, petrochemical, chemical, etc. industries, where failure consequences can have major impact on surrounding environment and/or population.
For mass-produced components/systems such as, motors, pumps, frequency converters, PLC's, cars, etc., the added engineering costs for development and employment of active fault-tolerance will be insignificant, when is calculated per produced unit. Employing self-diagnosis possibilities on component level results in producing "intelligent" components (sensors and actuators).

→ *spatial and weight* - In many systems, specially vehicles (space, naval, aerial, or on road), the limited available space simply prevents adding additional hardware redundancy to the system. For a satellite or a spacecraft, adding additional hardware redundancy means extra payload (weight and space), which in turn adds to the manufacturing and transportation costs (if even possible).

→ *Possibility of doing test* - In many systems, for instance those used in heavy, chemical, or space industries, performing reliability test is either costly and difficult or simply impossible. Applying active fault-tolerance will greatly improve the ability to circumvent the system-down situations by detecting and handling faults or predicting the forthcoming faults.

Beyond the abovementioned arguments that are used in many industrial/production entities to justify usage of active fault-tolerance, one should mention the very important "reputation" or "image" factor; the companies that have the reputation of delivering reliable systems are the ones that are in leading market positions.

3.6.2 A procedure for designing active fault-tolerant (control) systems

In order to **design** a fault-tolerant control system one needs to apply a certain number of activities. These are depicted in Figure 3.6 and categorized in the following steps:

1. **Objectives and scope definition**: The main activities in this step include establishing:
   - the functional goal of the system (concept definition).
   - system structure (type of components and their physical placement).
   - possible damages to environment/ production/ staff (life); this includes identifying environments within which the system operates.

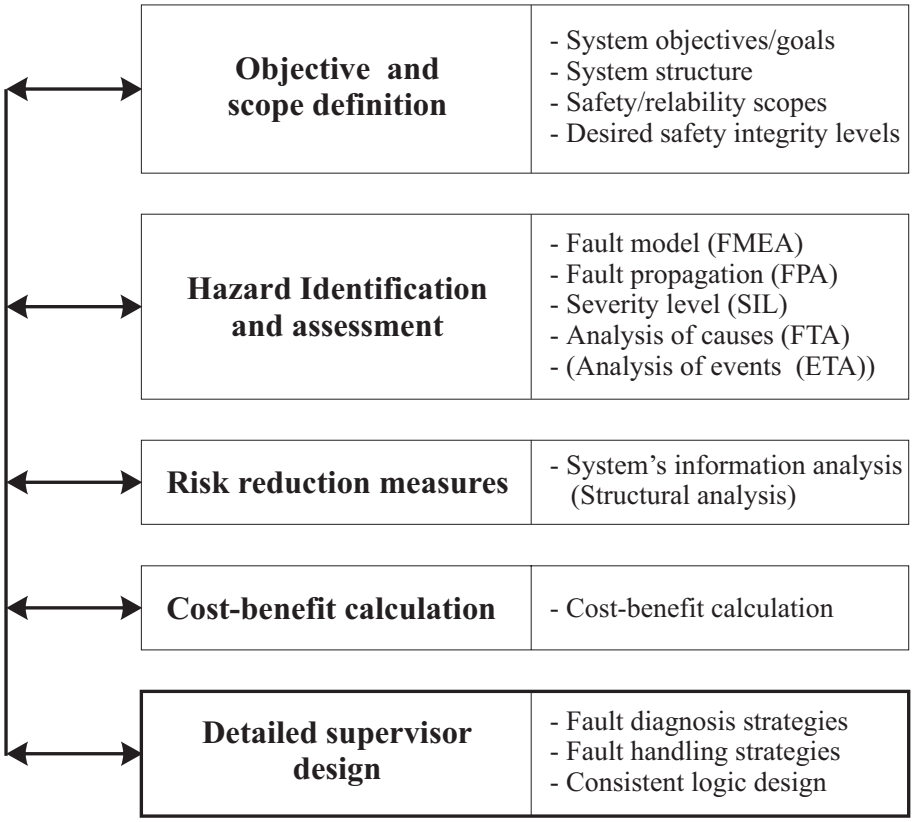

Fig. 3.6. Required steps for designing fault-tolerant control system

– acceptable level of system safety (fail safe / fail operational/ or else). Safety (and reliability) related requirements are hence specified.

2. **Hazard identification and assessment**: This step involves following activities:
   – Apply *FMEA* on the component/unit level to identify the possible faults. A fault model for each component is constructed.
   – Apply *Fault propagation analysis (FPA)* (see (Bøgh 1997)) to establish the end-effects of different components faults at the system level.
   – Apply *Fault Tree Analysis (FTA)* to identify the faulty components that cause the end-effects with high severity degree. Carrying out this activity produces a list of faults that needs to be handled.

   Notice 1. FTA and FMEA are not replacement for each other, but rather complementary steps.

   Notice 2:. One can furthermore perform Event Tree Analysis (ETA) in order to identify the outcome of the hazards.

  - Establish the severity degree of each end-effect at the system level. Since severity level has direct relation with system integrity level (SIL), one can employ *risk graph* (see section 2.3). To perform this activity one needs to use an appropriate *system model*; simulation of a fault in a system model will illustrate the end-effects at the system level and show how the system will react in the specified environment.
3. **Risk reduction measures**: This step concerns activities that are needed to be carried out in order to meet the reliability (and safety) objectives. The purpose at this step is to take the necessary steps to address the faults obtained from previous step through employment of redundancy (hardware and/or software). To minimize the number of redundant components and hence to reduce the cost, one can analyze the system to obtain knowledge about the existing inherent information within the system. This information can be used to perform fault diagnosis and fault handling. In this case a system model is needed. System model can be analyzed differently depending on the available knowledge about the system. *Structural analysis* method (Izadi-Zamanabadi 1999), (Staroswiecki and Declerck 1989) is a method that, based on the available information, provides the designer with a realistic assessment about the possibilities to detect different faults.
4. **Cost-benefit analysis** is a natural step in order to determine whether the continuation of the procedure to the next step is justifiable. Adding hardware redundancy (if possible) may prove cheaper than added cost due to engineering work related to the detailed design, implementation, and verification in the next step.
5. **Detailed design of supervision level**: When redundancy (particularly software redundancy) is applied in the system, one needs to:
   - carry out detailed analysis of the redundant information in order to develop/use appropriate fault diagnosis (i.e. fault detection and isolation) algorithms.
   - determine strategies for where and how to stop propagation of severe faults.
   - design consistent and correct logic (i.e. the intelligent part of the supervisor) to carry out decided fault handling strategies.

This is an iterative procedure; during detailed design one may discover that a severe fault can not be detected (and hence handled) with the existing system structure. there are two options for the case, either the specified reliability (and safety) requirements are relaxed or additional instrumentation are used. Choosing the second option requires going through the whole procedure again.

References

Bøgh, S. A. (1997, December). *Fault Tolerant Control Systems - a Development Method and Real-Life Case Study*. Ph. D. thesis, Aalborg University, Control Engineering Department,,, Fredrik Bajers Vej 7C, 9220 Aalborg Ø.

Bøgh, S. A. (2000). Safety and reliability assessments. LMC - safety and Reliability.

Bøgh, S. A., R. Izadi-Zamanabadi, and M. Blanke (1995, October). Onboard supervisor for the ørsted satellite attitude control system. In *Artificial Intelligence and Knowledge Based Systems for Space, 5th Workshop*, Noordwijk , Holand, pp. 137–152. The European Space Agency, Automation and Ground Facilities Division.

Burns, A. and A. Wellings (1997). *Real-Time Systems and Programming Languages - second edition*. Addison-Wesley.

Isermann, R., R. Schwarz, and S. Stölzl (2000, June). Fault-tolerant drive-by-wire systems - concepts and realizations. In A. Edelmayer and C. Bányász (Eds.), *4th IFAC on Fault Detection Supervision and Safety for Technical Processes - Safeprocess*, Volume 1, Budapest, Hungary, pp. 1–15. IFAC.

Izadi-Zamanabadi, R. (1999, September). *Fault-tolerant Supervisory Control - System Analysis and Logic Design*. Ph. D. thesis, Aalborg University, Control Engineering Department, Fredrik Bajers Vej 7C, 9220 Aalborg Ø.

Leveson, N. G. (1995). *Safeware - System safety and computers*. Addison Wesley.

Mullender, S. (Ed.) (1993). *Distributed Systems (second edition)*. Addison-Wesley.

Russomanno, D. J., R. D. Bonnell, and J. B. Bowles (1994). Viewing computer-aided failure modes and effects analysis from an artificial intelligence perspective. *Integrated Computer-Aided Engineering 1*(3), 209–228.

Staroswiecki, M. and P. Declerck (1989, July). Analytical redundancy in nonlinear interconnected systems by means of structural analysis. Volume II, Nancy, pp. 23–27. IFAC-AIPAC'89.

Stengel, R. F. (1993). Toward intelligent flight control. *IEEE Trans. on Sys., Man & Cybernetics 23*(6), 1699–1717.