Reliability optimization of hardware components and system´s topology during early design phase

Von der Fakultät für Ingenieurwissenschaften, Abteilung Maschinenbau

der

Universität Duisburg-Essen

zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften (Dr.-Ing.)

genehmigte Dissertation

von

Amir Kazeminia

aus

Teheran

Gutachter: Univ.-Prof. Dr. Dirk Söffker
Univ.-Prof. Dr. Arno Meyna

Tag der mündlichen Prüfung: 3. Juli 2013

Danksagung

Die vorliegende Arbeit enstand im Rahmen eines Forschungsprojektes durch die Zusammenarbeit zwischen der Firma TRW Automotive, Lucas Varity GmbH und dem Lehrstuhl Steuerung, Regelung und Systemdynamik der Universität Duisburg-Essen.

Mein ganz besonderer Dank gilt meinem Doktorvater Herrn Univ.-Prof. Dr.-Ing. Dirk Söffker für seine hervorragende Betreuung und die zahlreichen Anregungen während der Entstehung dieser Arbeit. An dieser Stelle möchte ich mich nochmals bei ihm für das wiederholt in mich gesetzte Vertrauen bedanken.

Ebenfalls bedanke ich mich bei Herrn Univ.-Prof. Dr.-Ing. Arno Meyna für die sorgfältige Begutachtung der Arbeit.

Herrn Dr.-Ing. Rüdiger Eick danke ich nicht nur für die fachliche und didaktische Unterstützung, sondern auch für den menschlichen Beistand, gerade in der finalen Phase dieser Arbeit.

Bei meinen Kollegen am Lehrstuhl Steuerung, Regelung und Systemdynamik und in der Firma TRW Automotive, Lucas Varity GmbH möchte ich mich für die kollegiale Zusammenarbeit bedanken. Ich erinnere mich gern an die freundschaftliche Atmosphäre am Lehrstuhl Steuerung, Regelung und Systemdynamik.

An dieser Stelle bedanke ich mich bei Herrn Dipl.-Ing. Erwin Michels, Herrn Dr.-Ing Dirk Kesselgruber, Herrn Prof. Dr.-Ing. Marco Junglas, Herrn Dr.-Ing. Simon Schilling, Herrn Dr.-Ing. Kai Höfig und Herrn Dipl.-Ing. Oliver Sacher für die großartige Hilfe und die fachlichen Diskussionen.

Zu guter letzt gilt mein besonderer Dank meiner Familie und Julia, die mir immer Stärke und den nötigen Rückhalt gegeben haben.

Koblenz, im Juli 2013

Abstract

To master the complexity in modern vehicle, Original Equipment Manufactures (OEM) attempt to integrate as many functions as possible into the given Electronic Control Unit (ECU), sensors, and actuators without degrading the safety and comfort functionalities. Furthermore scalability, versatility, and performance of products are key to success of electronic development in new modern vehicles. Various functional and nonfunctional requirements obviously shall be fulfilled during development of such complex systems.

Choosing of hardware design structure and determination of hardware characteristics are the initial steps during early design phase. The conventional methods for selection of hardware components and topologies are mostly functional-driven. Conventional approaches are largely lacking in versatility and scalability.

Due to innovative and complex trend of mechatronic product development, new approaches for hardware decision must be available which support the designers in case of changing (growing) customer demands.

One of most important customer requirement for a complex system is reliability. The need for more reliable system design drives up the cost of design and influences the other system characteristics such as weight, power consumption, size, etc. These design goals like reliability, cost potentially impose conflicting requirements on the technical and economic performance of a system design.

Hence, visualization and evaluating of the conflicting design preferences and early choosing optimal design are one of the most critical issues during design stage. Many multi-objective optimization approaches have been proposed to tackle this challenge. This dissertation proposes an efficient reliability optimization framework which aids the designers to determine the optimal hardware topology with optimal set of components under known technical and financial restrictions.

The proposed reliability optimization framework allows describing the hardware structure of a complex system by a System Reliability Matrix (SRM) and the failure rate vector of involving hardware components. The reliability characteristics of components and the redundancy policy can be varied automatically via the SRM and its corresponding failure rate vector in order to determine optimal solutions. The proposed methodology ultimately addresses the most efficient system architecture (topology) and ascertain the unknown reliability characteristics of hardware components under consideration of financial and technical constraints.

It is to be noted that the numerical deterministic search methods and genetic algorithms are applied to optimize the defined objective function under multiple constraints (reliability, cost, weight, size, etc.) and to determine the reliability characteristics of components. A general enumerative algorithm generates all design architectures (topologies) and filters the feasible design architectures (topologies) based on given constraints like budget and etc.

Contents

List of Figures

List of Tables

Abbreviations

ECU	Electronic Control Unit
OEM	Original Equipment Manufacture
AUTOSAR	AUTomotive Open System ARchitecture
SRM	System Reliability Matrix
RBD	Reliability Block Diagram
BOM	Bill of Material
COTS	Commercial-Off-The-Shelf
SIL	Safety Integrity Level
SPFM	Single point fault metric
LPFM	Latent point fault metric
ASIL	Automotive Safety Integrity Level
CAN	Controller Area Network
LIN	Local Interconnect Network
ESC	Electronic Stability Control
RAMS	Reliability, Availability, Maintainability; Safety
PCB	Printed Circuit Board
FIT	Failures In Time
EA	Evolutionary Algorithm
ABS	Anti-lock Braking System
EMB	Electromechanical Braking
EBD	Electronic Brake Distribution
ESP	Electronic Stability Program
ACC	Adaptive Cruise Control
TCS	Traction Control System
MOEA	Multi-Objective Evolutionary Algorithms
CRA	Component Reliability Allocation
TRA	Topology Reliability Allocation
FTA	Fault Tree Analysis

Nomenclature

Latin Letters

a_i	Fixed cost parameter
b_i	Fixed cost parameter
$C(\lambda_i)$	Cost function of component i
C_{max}	Maximum costs
C_{System}	Total costs of a system
d_i	Upper limit for linear constraints
DC	Diagnostic coverage of safety mechanism
e_i	Upper limit for non-linear constraints
$f(t)$	Failure density function
$f(x)$	Sub-objective function
$g(\mathbf{x})$	Linear constraints
$\mathbf{F}(\mathbf{x})$	Vector of objective functions
J	Total objective function
$\mathbf{J}(\mathbf{x})$	Inequality constraints
$\mathbf{K}(\mathbf{x})$	Equiality constraints
L	Lagrangian for a certain numerical problem
$LPFM$	Latent point fault metric
n	Number of components
$q(\mathbf{x})$	Non-linear constraints
$r_i(t)$	Component reliability used in subsystem i at time t
$R(t)$	General reliability function at time t
R_{br}	Relability of compontents in a bridge structure
$R_{\text{k out of n}}$	Relability of k-out-of-n structure
R_{max}	Maximum relability level of specific component available in the market
R_{min}	Minimum relability level of specific component available in the market
R_{P}	Relability of compontents in a parallel structure
R_{required}	Minimal acceptable relability level of a system structure
R_{S}	Relability of compontents in a series structure
R_{system}	Relability level of a system

R_{voter}	Relability of decision making device
S_i	Size of component i
$S_{\max}$	Maximum size
$SPFM$	Single point fault metric
t	Time point
w_i	Weighting factor for a sub-objective function
W_i	Weight for component i
$W_{\max}$	Maximum weight
$\mathbf{x}$	Vector containing the design variables of the optimization problem
z	Numerical cost function

Greek Letters

α	Weighting factor for the reliabilty term of the total objective function
β	Weighting factor for the cost term of the total objective function
ε_i	Reliability improvement for component i
λ	Component failure rate
$\lambda(t)$	Hazard rate function at time t
λ_{MPF}	Component failure rate associated with multiple point faults
λ_{RF}	Component failure rate associated with residual faults
λ_{S}	Component failure rate associated with safe faults
λ_{SPF}	Component failure rate associated with single point faults
$\rho(u)$	Reliability of detection/switching for the subsystem u
θ	Lagrangian multiplier

1. Introduction

1.1. Application of complex mechatronic systems in vehicles and future of Drive By Wire systems

In recent years, the demand of high ergonomic comfort requirements, making safety functions as reliable as possible, reducing weight, consideration of restricted available space for new innovative systems in the vehicle, and affordability of a vehicle have changed the visions of international automakers and the consumer buying behavior dramatically. The above expectations have led to staggering functional advancement of mechatronic system within vehicles in the past few years.

Therefore a vehicle which encompasses more than between 70 up to 100 ECUs, about 4 kilometers of insulated wiring, a high number of sensors, different electromechanical actuators and hundreds of thousand of lines of software code can not be categorized as a traditional mechanical system as shown in the figure 1.1. Subsequently the required methods for requirement management, architecture evaluation, system development, type of production, product services shall be matched to new advancement. In the conjunction with these facts, OEMs and their supplier are faced with

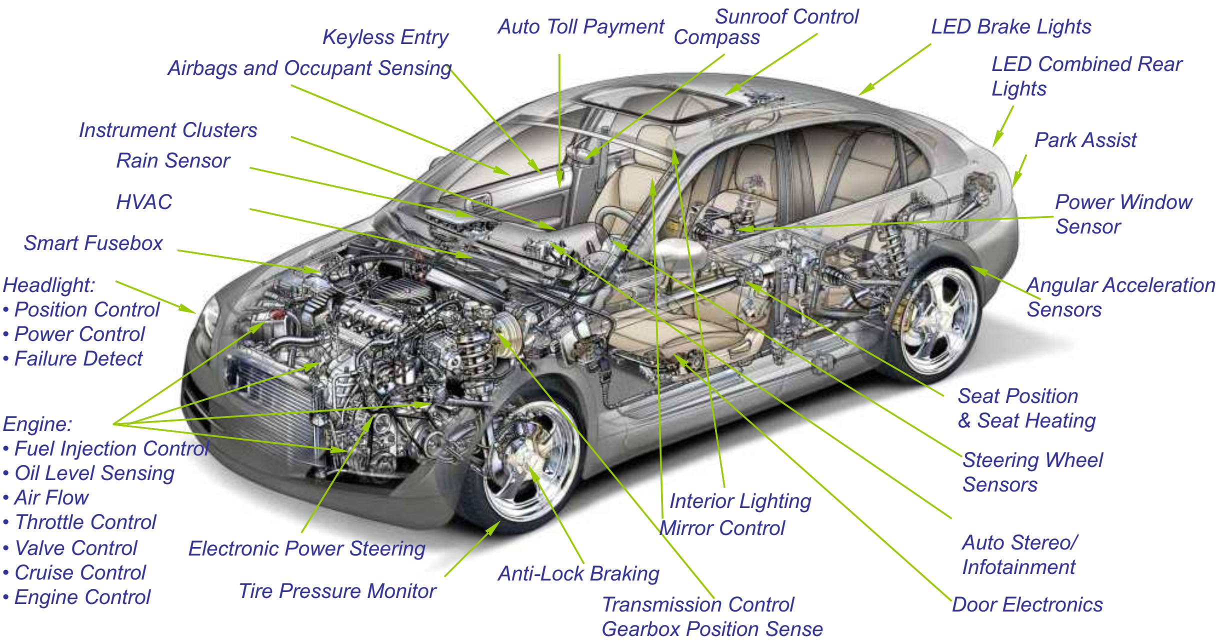


Figure 1.1.: Typical deployment of mechatronic systems in current vehicles [BOS]

different challenges and motivations as illustrated in the figure 1.2. According to OEM and supplier statements, more than 90% of all innovations in vehicle will be associated with electronics and mechatronics in the coming years [Gei06]. New ad-

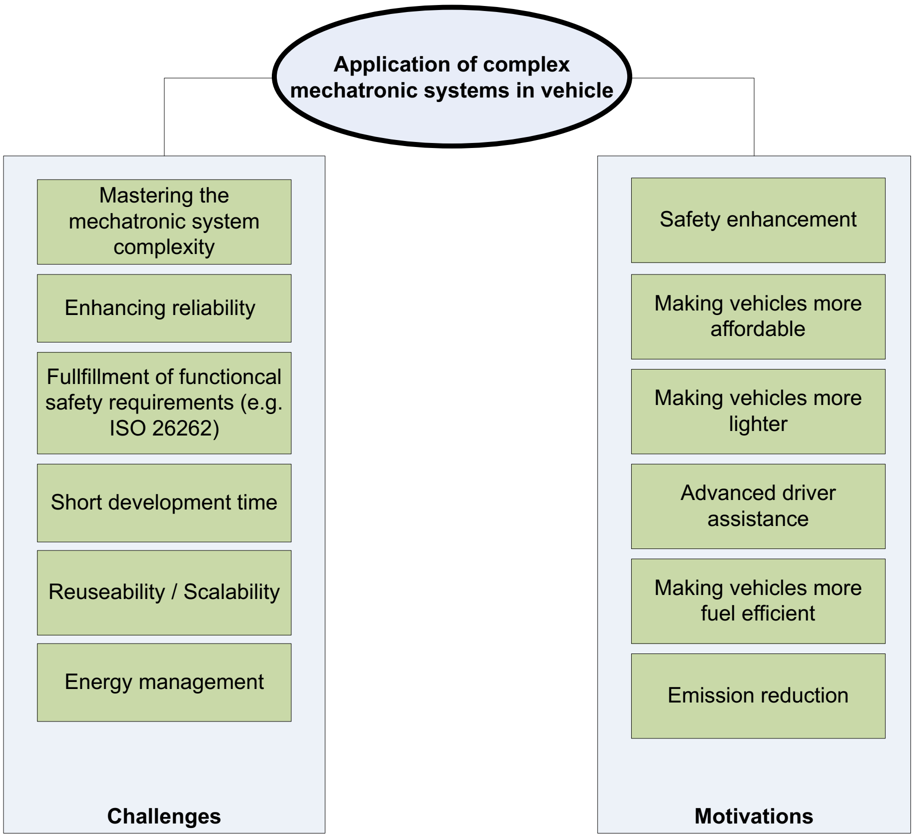


Figure 1.2.: Deployment of mechatronic systems

vancements in the field of automotive electronics partially liberate the driver from routine tasks, in order to enhance vehicular safety and performance. Most major manufacturers recently introduce concepts and prototypes for drive-by-wire systems which change the driver's role fundamentally. There are currently many arguments available which indicate application of drive-by-wire systems is indispensable in near future. The most important arguments favoring deployment of drive-by Wire systems can be formulated as follow:

– reducing overall mass of vehicle

– keeping emissions and fuel consumption as low as possible

– enhancement of safety and comfort

– providing more place available inside of vehicle (e.g. increasing engine room)

– raising the performance and agility of main vehicle tasks

– avoiding time-consuming mechanically and hydraulic laborious services [LH02] [The02].

Deployment of X-by-wire systems without having mechanical backup (brake-by-wire, steer-by-wire) require to design highly reliable and fault-tolerant electronic systems. Indeed, the real challenge is to industrialize a fault-tolerant reliable system at a reasonable cost for mass production [BFM+03]. Besides the drive-by-wire systems typically are safety related systems. Hence the RAMS analysis during early design phase are relevant in order to convince customers and lawmakers [ISS02] [GSS98].

It is indeed vital to note that although x-by-wire systems are well-established in the aerospace, railway industries, drive-by-wire systems have been slow in order to establish themselves in the vehicle world. One of the key obstacles is related to the fact that development and production costs play much greater role in the automotive industry in comparison to other industry.

Regardless of the mentioned benefits and limitations of drive-by-wire systems, there has been a rapid expansion in the complexity of mechatronic safety systems in recent years. Traditionally, the safety systems are divided into two categories namely, active safety systems and passive safety systems. Passive and active safety systems typically are separated. Passive safety features like airbag, safety belt buckle switch

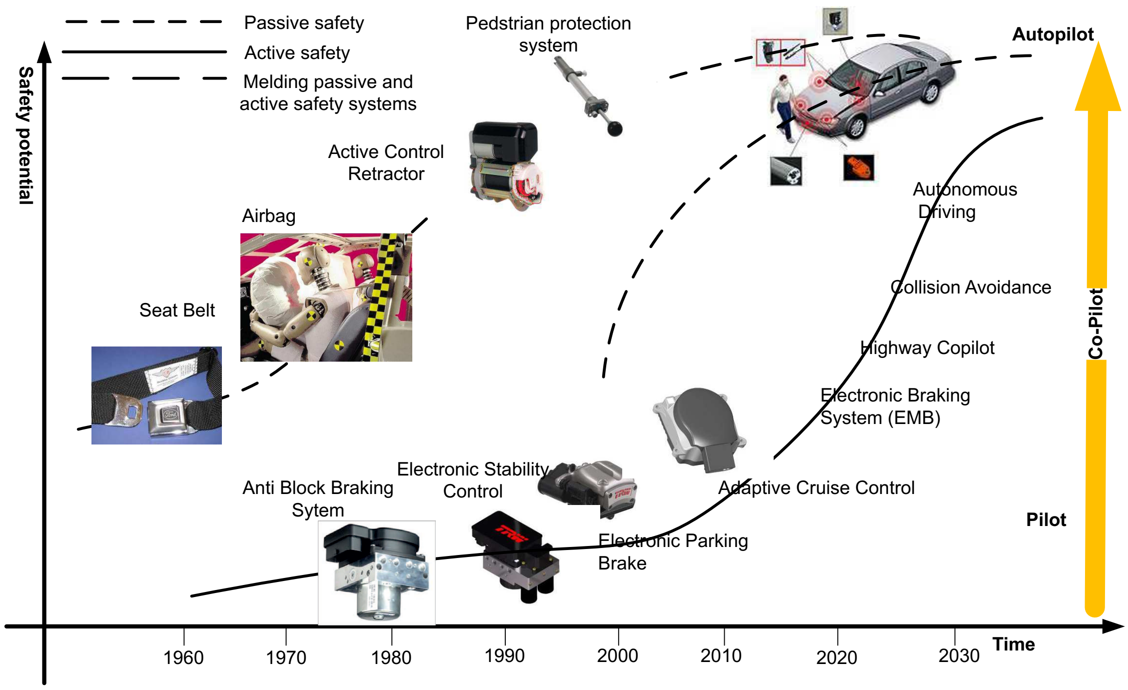


Figure 1.3.: Safety ECU development introduced in [LH02]

etc. minimize the impact of hazardous accidents on the occupants and pedestrians. Whilst active safety features offer functions like electric park brake, stability control system, adaptive cruise control and lane departure warning, and tire-pressure monitoring, which proactively support drivers to avoid an accident [TRW09]. For instance one of most established active safety features of such system is realized by combining multiple source of sensor information with Electronic Stability Control ESC controller. The system analyzes the data from the wheel speed sensors, the yaw rate, lateral acceleration sensor, the wheel hand angle sensor permanently. Simultaneously, ESC compares the driver's input with the vehicle's actual behavior. This evaluation process can provide the vehicle the possibility to apply the brakes and adjust the the engine torque autonomously once the car understeers or oversteers.

The use of these safety features (passive and active) are proving to be indispensable to improve safety in the vehicle. Nevertheless downside of using passive and active safety features are that the number of ECUs and sensors in the vehicle grows rapidly. Most suppliers currently provide these functions as a hardware device which is one of the main reasons why the number of ECUs and sensors has been increased rapidly within the last years. This trend leads to increased level of complexity and high cost [Che08].

Due to this fact, the automakers are attempting to manage the functions and to combine active and passive safety systems together in order to minimize the number of sensors, ECUs, and actuators to master complexity, reduce cost, and restrict likelihood of unintended events as shown in the figure 1.3. In other words, a function shall not necessarily be carried into execution in a dedicated ECU but a bunch of functions can be integrated in a central ECU. For facilitating this trend, following domains need to considered for future automotive development [AUT12]:

- optimal selection of hardware electronic topologies and robust component characteristics,
- separation between hardware level and software level, and
- creation of open standards and interfaces in order to provide the possibility for integration of different safety function from various suppliers on multipurpose ECUs [SMWM09].

1.2. Reliability challenges in current vehicle development-some examples

Development and integration of these functions are accompanied with the fact that all of these ECUs and sensors are interlinked by complex communication bus system like FlexRay, CAN, LIN, Ethernet, etc. The drawback of this trend is that the possible failure of these ECUs, sensors, mechanical, or hydraulic parts influences sometimes the reliability of whole interlinked systems and different comfort/safety functions negatively. The growing complexity and interoperability of ECUs, sensors and actuators are triggering an urgent need for high reliability systems design. This can be observed in the callback rate in the recent years as illustrated in the figure 1.4. For example Toyota globally recalled 12 millions of vehicles due to safety concerns in

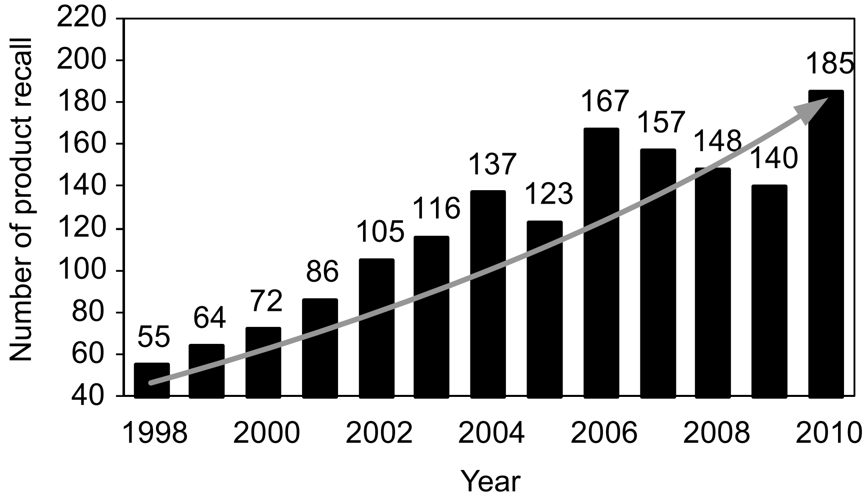


Figure 1.4.: Recent vehicle callbacks [Kra10]

2010 and 2011 (particularly more than 110,000 hybrid vehicles over concerns about a problem with the power supply circuit) [BBC]. Such excessive recalls and warranty-related issues can demolish profits and ruin the image of vehicle manufacturers and their suppliers. In other words, the final customers are not concerned if new safety features and comfort functions are implemented in simple mechanical/hydraulic way or in complex mechatronic way.

The customers require an innovative system with complex safety features at least as reliable as conventional product at same or lower price. The above customer expectation drives the manufacturers to analyze the different mechatronic product aspects (performance, safety, reliability, comfort, price, power consumption etc.) during early design time. The experience shows that the success or breakdown of mechatronic product is highly dependent to how comprehensively all important functional and nonfunctional aspects have been considered during early stage of design.

On the other side, competitive pressures for new product are dramatically high and drive the need for innovation and shorter time-to-market. For example, due to competitive pressures, vehicle manufacturers currently are economically not allowed to invest more than 2 up to 3 years for the development of safety critical systems like safety critical and driving assistance systems. The shortening of development times forces the suppliers to reduce their product design time which generate great conflict at times. Figure 1.5 predicts microcircuit reliability trends in case satisfying these requirements.

During recent development of the safety critical electronic like microcontrollers, the number of transistors inside of such components has been almost doubled every two years. This development has been taken place through decreasing in transistor sizes which provides swifter, smaller integrated circuit with high reduced power

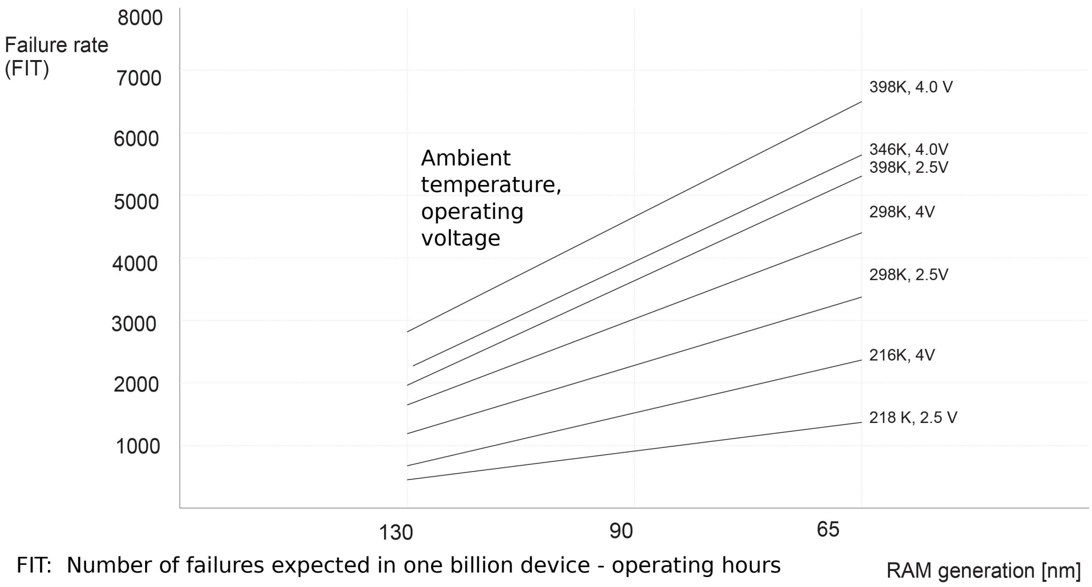


Figure 1.5.: Memory structure and its impact on reliability [GW10]

dissipation [WCH11]. Vehicle industry currently forces the semiconductor suppliers to push this trend and change the memory technology on the microcontroller. The notion behind this pressure is that the new comfort and safety-critical functions necessitate more performance and much more memory for calculating the complex algorithms. Semiconductor failure mechanisms (bit flipping, stuck at failures, etc.) lead to higher failure rates, shorter device lifetimes and unanticipated early device wear out [WB11].

Failure rate of such components is increasing with feature size scaling, temperature, and operating voltage as illustrated in the figure 1.5. Impact of radiation is not investigated in the figure 1.5. The reliability of such components falls off dramatically (fall below 10 years) once memory sizes are designed below 100nm [Bec11]. Such negative impacts on the reliability influences the use of microcircuits in automotive application that requires longer service time.

1.3. State of the art in mechatronic system development

Generally there exists different definitions about mechatronic. One of most accepted definitions is "synergetic integration of mechanical engineering with electronic and intelligent computer control in the design and manufacturing of industrial products and processes" [HTF96].

The multiple disciplines involved with mechatronics are shown in figure 1.6. In order to address design challenges during mechatronic system development, a comprehensive analysis over all mechatronic system disciplines is mandatory [The04]. During design of mechatronic system, it is typically recommended that designers shall use an interdisciplinary development approach to develop a complex mechatronic system.

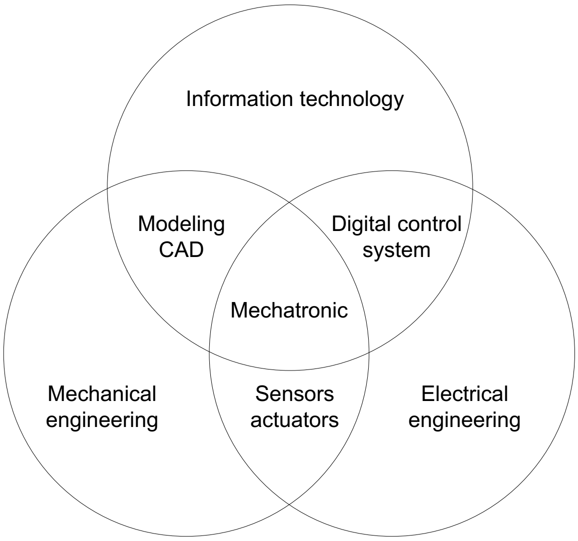


Figure 1.6.: Mechatronic system domains from [The04]

In this regard, the VDI 2206 recommends strongly using V-Model in order to justify that the requirements of the developed product are verified and validated during implementation as depicted in figure 1.7. In other words, it needs to be ensured that all specified requirements are implemented at the most possible detail and simultaneously the verification of requirements using tests, simulation, etc. follows bottom up Principe [The04].

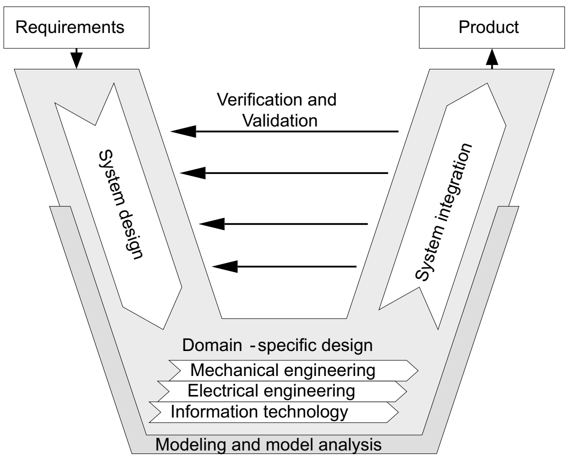


Figure 1.7.: Recommended V-model approach for complex mechatronic system [The04]

The VDI 2206 doesn't recommend applying the conventional design process which deals with the different engineering domains independently from each other. A real mechatronics approach requires that an optimal choice must be made in the different domains simultaneously. Ultimately a design requires optimization of the system as a whole. Traditionally there are five key stages to develop a complex mechatronic product as smartly as possible:

1. understanding the customer's functional and nonfunctional requirements and their translations into a technical specification
2. selection an initial topology of system which suits the requirements in the derived specification
3. component selection based on both the requirements in the derived technical specification from all domain (hardware and software) and the financial constraints
4. improvement and optimization of the last two steps and definition and implementation of software mechanisms
5. test and validation.

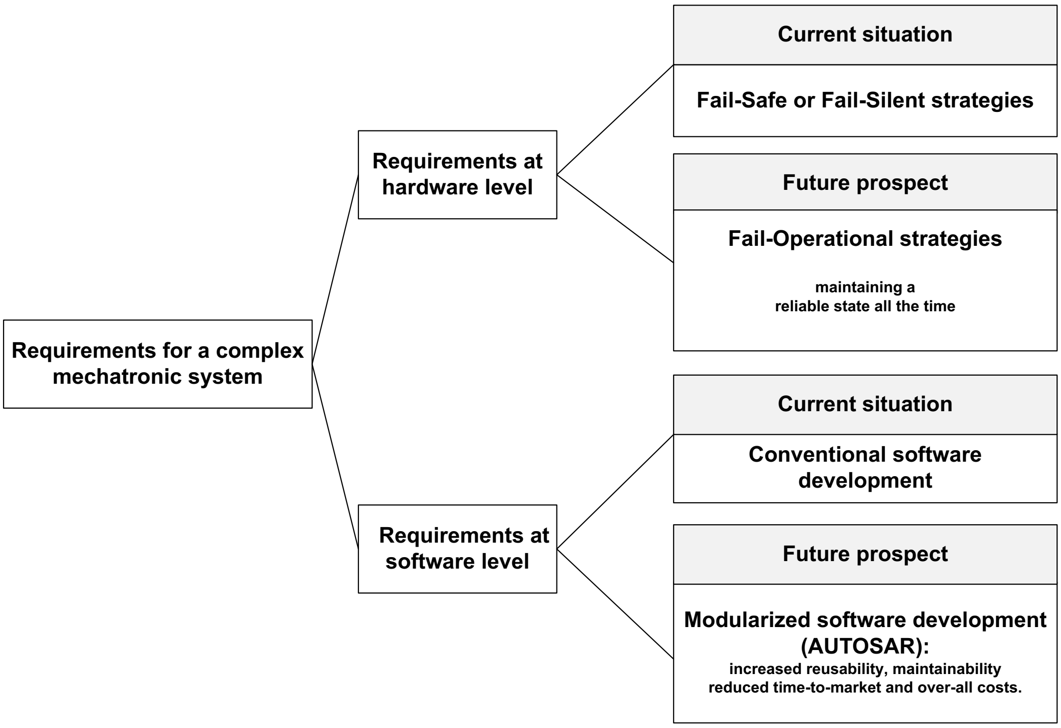


Figure 1.8.: Decomposition of system requirements into software and hardware requirements - current and future

Such a system engineering approach helps the designers to deal with the complexity of system design. The notion behind this approach is that a system is required to be decomposed to software and hardware levels in order to separate the design concerns with different causes. This splitting the system requirements into two manageable chunks allows the designers to narrow down the concrete requirements at both levels (software and hardware) as illustrated in figure 1.8. The modular-based approaches for hardware and software designs are indispensable.

For software concerns, there are historically different attempts to reduce the likelihood of software failures by using V-Model and process improvement.

It is proven that development of new system based on the existed software code in the field instead of using new software from scratch improves the reliability of system and reduces the software development cost tremendously.

Due to software development cost and pricey software tests, manufacturers attempt to use the principle of *Modularization of software* in order to meet reliability requirements and shorten the development time. Indeed the main precondition of reusing software code is that the hardware of system is separated from application software. It is pivotal to note that the most important factor for software reuse is good software architecture/partitioning and optimal hardware topology.

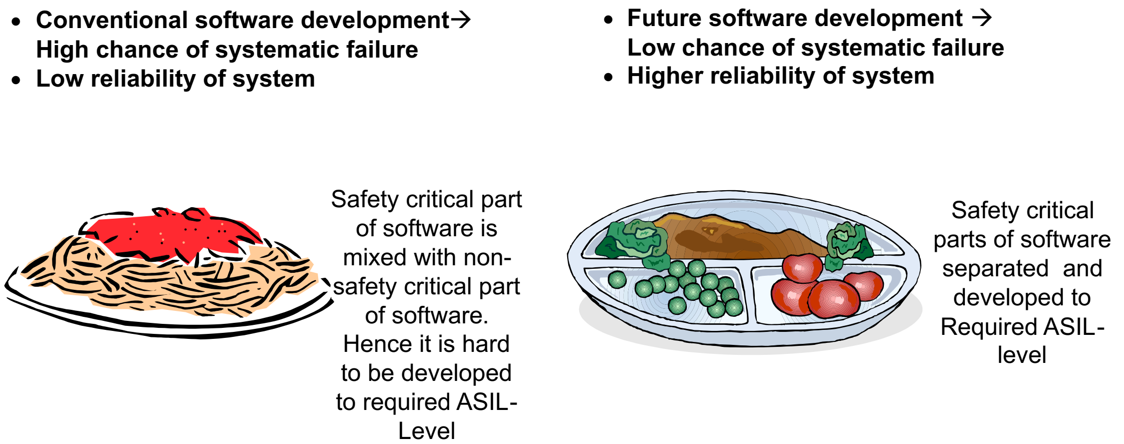


Figure 1.9.: Advantages of modularized software development against current software development

AUTOSAR (AUTomotive Open System ARchitecture) offers an open and standardized automotive software architecture which aids the developer to master the complexity and improve the quality of software in an efficient way [AUT12].

The main objectives of AUTOSAR are

1. to increase the quality of automotive software, its maintainability,
2. to support usage of COTS components across product lines, and
3. and finally to optimize costs and time to market [AUT12].

AUTOSAR facilitates the re-usability of software and bug fixing of software over service life of the vehicle [The04]. The standardization of interferences also offers the possibility to make a difference between safety critical software modules and non-safety critical part of software as described in the figure 1.9.

The partitioning of software modules also makes a smart validation plan for designers possible. In other words, the software designers can focus more on the development quality of those parts of software which are relevant to safety and thus could jeopardize the reliability of system. In addition to these benefits , it drastically reduces the unnecessary expense of additional tests.

Dealing with software concerns is out of the scope of this dissertation. This work merely concentrates on the hardware concerns in regard to reliability optimization. Generally definition and implementation of software application will be nailed down after the hardware strategy is clear [HKK04]. For example, amount of memory, type of processing unit, type of power supply and other hardware specifications ascertain which volume of software code can be proceed and handled properly.

1.4. Motivation of dissertation

Typically the hardware design team investigates primarily the system functional requirements before any other types of requirements in order to build a trustworthy system and fulfill the customer's expectations. The investigation usually leads to the determination of a system hardware schematic and a list of required hardware components in the form of a BOM. A system schematic characterizes the topology of system while a BOM specifies what types of components meet the functional and financial requirements.

Due to the complexity of new products, there is generally less attention bestowed on some none-functional requirements like system reliability, functional safety issues, and their quality technical constraints during early design phase.

Therefore, defining precise product none-functional requirements like reliability requirement (e.g. fault tolerant or fault operational behavior of product) and implementation of required countermeasures become an afterthought. This leads to phenomena which many none functional requirements just can be investigated grossly after the design architecture is wrapped up.

However the non-functional quality attributes such as safety, reliability, performance, maintainability are at least as pivotal as the functional requirements for decision on hardware architecture [HKK04]. Ignoring the fulfillment of such requirements during early design phase may result in high warranty cost. Consequently it leads to loss market share. That is why the VDI 2206 recommends applying V-model approach for various attributes of a complex systems concurrently as illustrated in the figure 1.10.

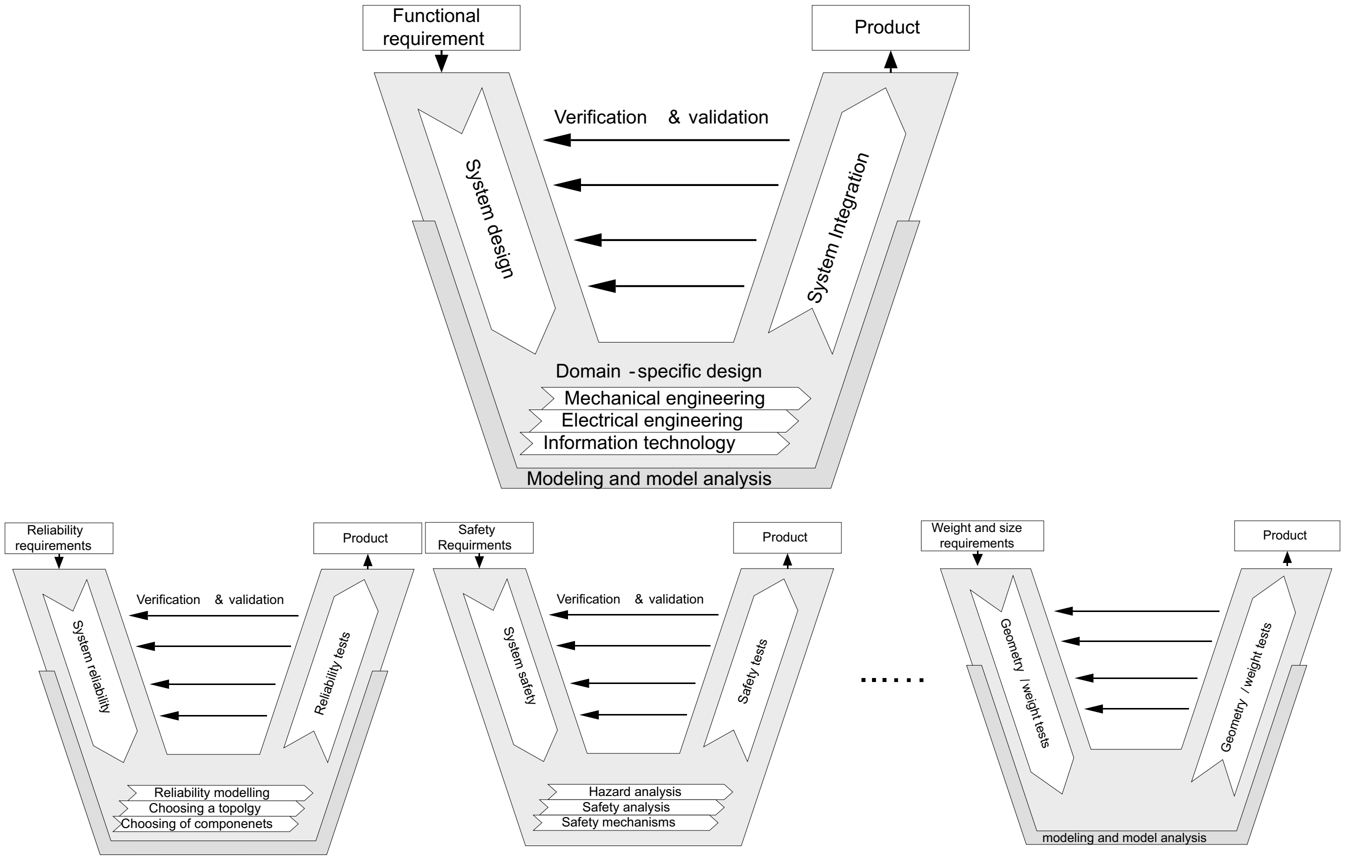


Figure 1.10.: Functional and non-functional requirements based development

Hence the suppliers need not only to provide evidence to demonstrate how a design performs its intended function, as specified by the functional requirements. It is also required to bring forward probative evidence showing the fulfillment of system non-functional requirements like safety, reliability, size, and weight requirements under given budget restrictions as illustrated in the in the figure 1.11.

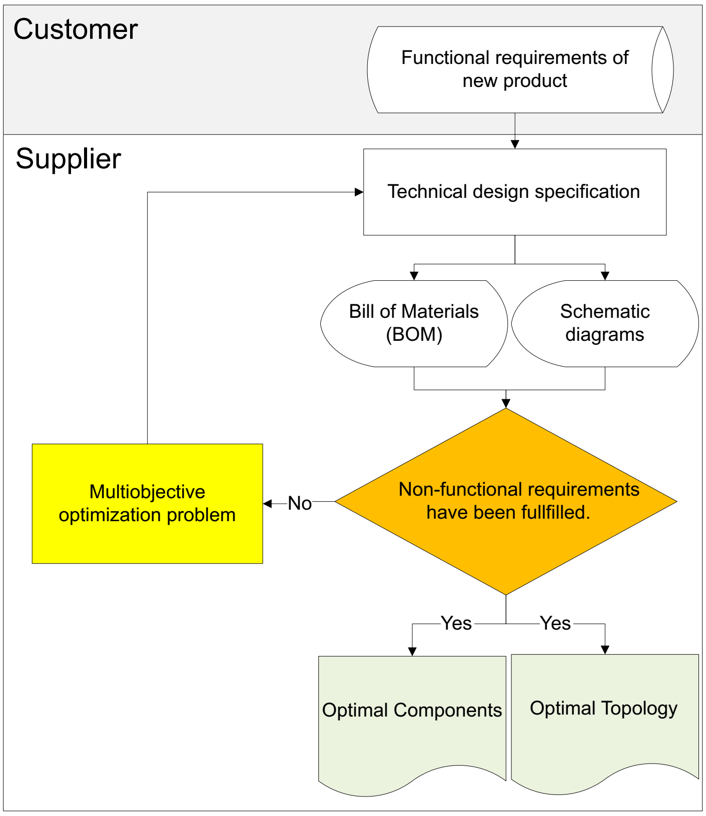


Figure 1.11.: Consideration of functional and non-functional requirements during proposed design optimization

One of the most essential nonfunctional requirements which is defined in the customer specification directly or indirectly is the reliability requirement. The reliability requirements specify the minimum level of performance expected over a given time span by the user. In the automotive domain, the OEMs usually express their reliability requirements in following different ways:

- the required operation time in years (e.g.15 years)
- the required range in kilometers or miles (e.g. 300000 Km)
- the required ignition driving cycle (e.g. 54000)

From reliability technical point of view, there are generally two conceptual approaches to increase system reliability at the hardware level:

– perfection (fault avoidance)

– and redundancy (fault coverage) [FGF80].

The first approach strives for preventing malfunctions of components by over-design, 100% inspection, etc. This approach is not always economically and technically feasible. On the other hand, redundancy seeks to change the topology by adding parallel components during development which increase life time of whole system. By using a multi-objective optimization, it can be determined which components shall be over-designed and where a topology change (a new redundancy strategy) is required.

The suppliers are required to scrutinize all requirements (including reliability requirements) and make sure that the expressed reliability requirements can be translated into detailed design specification. In the current design methods, designers are browsing through catalog of off the shelf items in the market and attempt to map the features and constraints of these components to mentioned requirements. Iteratively the set of components in different topologies are clustered in order to find more plausible solutions sets. In the end, a combination of components based on a feasible topology can be validated during test, in order to ensure that all functional and non functional requirements are fulfilled. Manually choosing the right compo-

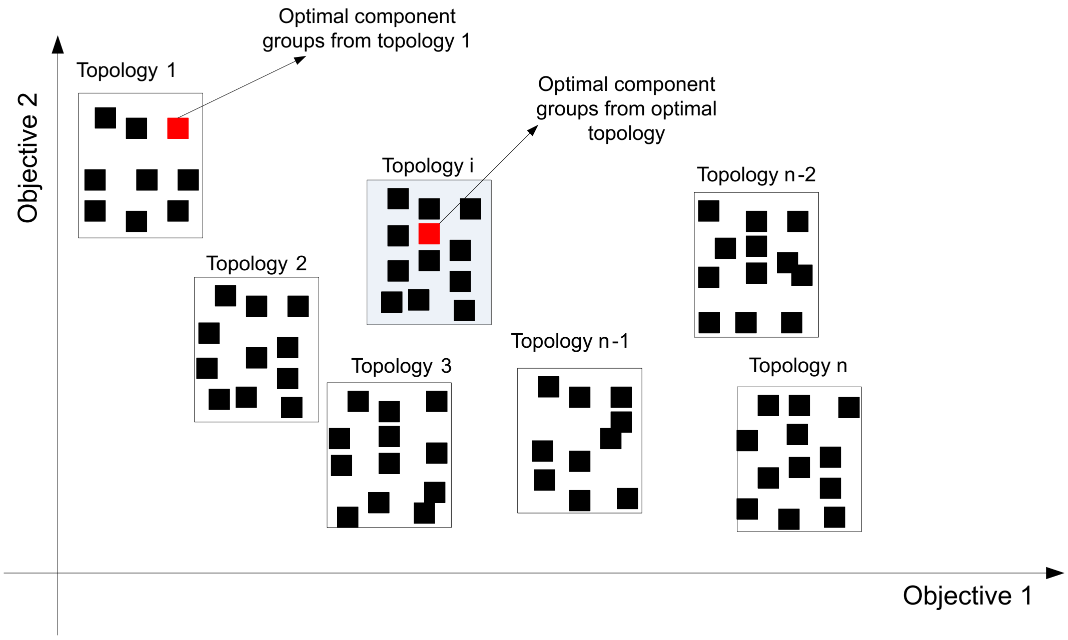


Figure 1.12.: Optimal topology and optimal component groups

nents and selecting an optimal topology is costly in terms of time and resources during short time development. In case of innovative complex system, many design hardware alternatives are supposed to be taken into consideration which their manual evaluations are almost impossible. Besides the design team come up against challenge of achieving different design objectives which some of them are in conflict with each other. Under this circumstance, the designers must make tradeoff and optimize conflicting characteristics against each other [CS96b].

For instance how much is longer life time of a specific component worth in terms of higher manufacturing costs or in terms of heaviness of components which take grand part of limited space in the vehicle. As in the figure 1.12 is illustrated, there exists different hardware topologies which they have several component group choices. One or maybe more than one these component groups can be considered as the optimal of the optimal topology (topology highlighted in blue) which fulfills the defined conflicting objectives (objective 1 and objective 2). The searching for best reliable design takes place at the topology level (schematic) and at the component level (BOMs).

To optimize hardware design and to provide an adequate reliability level of architecture under given financial, technical, environmental constraints, application of a consistent platform is indispensable. The high number of design alternatives and complexity of technical constraints makes the manual decision taking impossible.

Therefore this dissertation proposes an efficient optimization framework which aids the designers to determine the optimal hardware topology with optimal set of components under given restrictions.

The proposed approach assumes that functional requirements based on customer needs are specified already. Based on the functional requirements, the requirements on hardware can be determined roughly. In other words, the an initial design with initial component groups are available within the scope of a prototype. The elements of the proposed framework is shown in the figure 1.13.

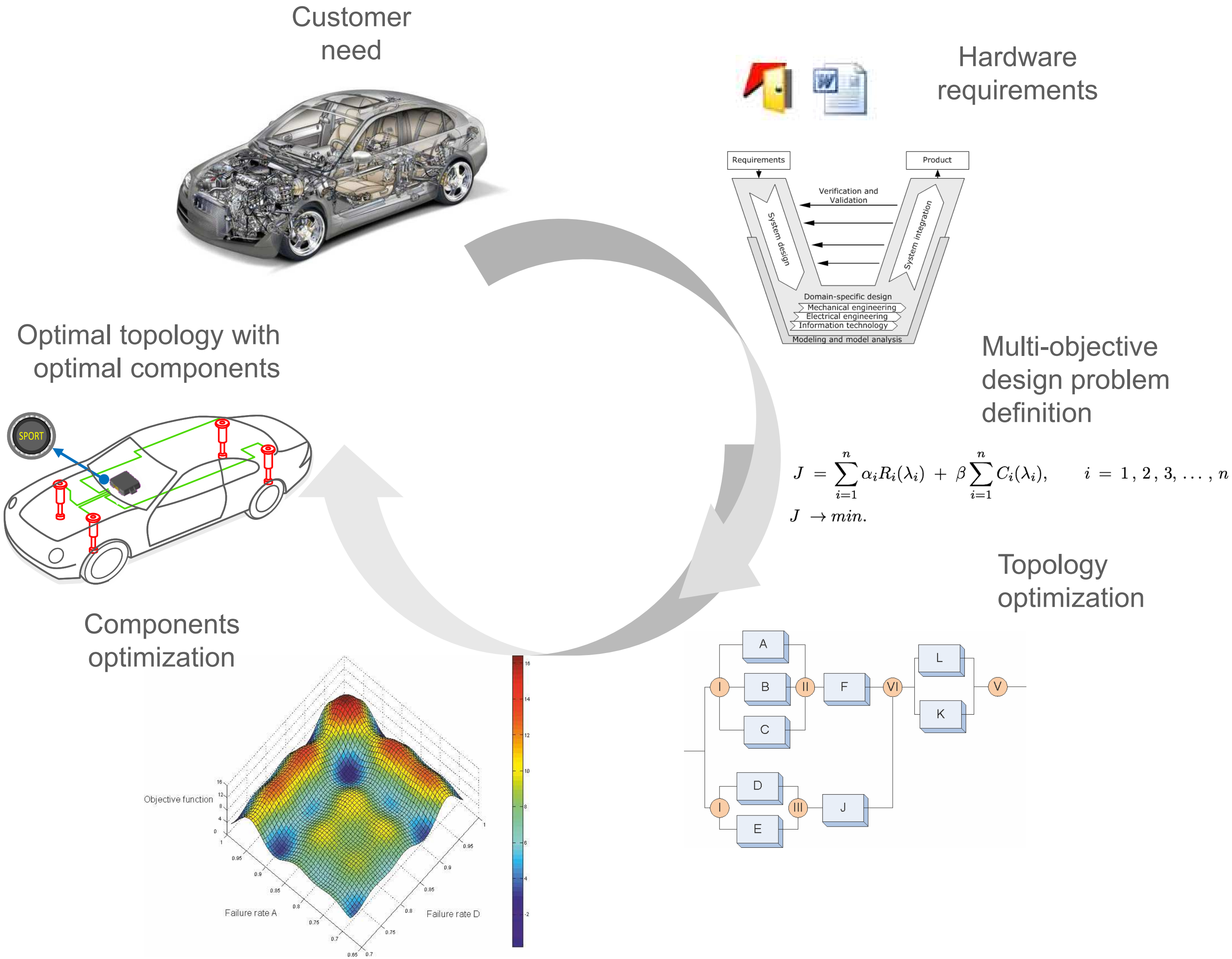


Figure 1.13.: Proposed framework for obtaining optimal topology with optimal components

1.5. Objectives and scope of dissertation

This dissertation is directed toward the development of a methodology to support the designers for topology and component selection during early phase of design. This methodology covers following concrete objectives:

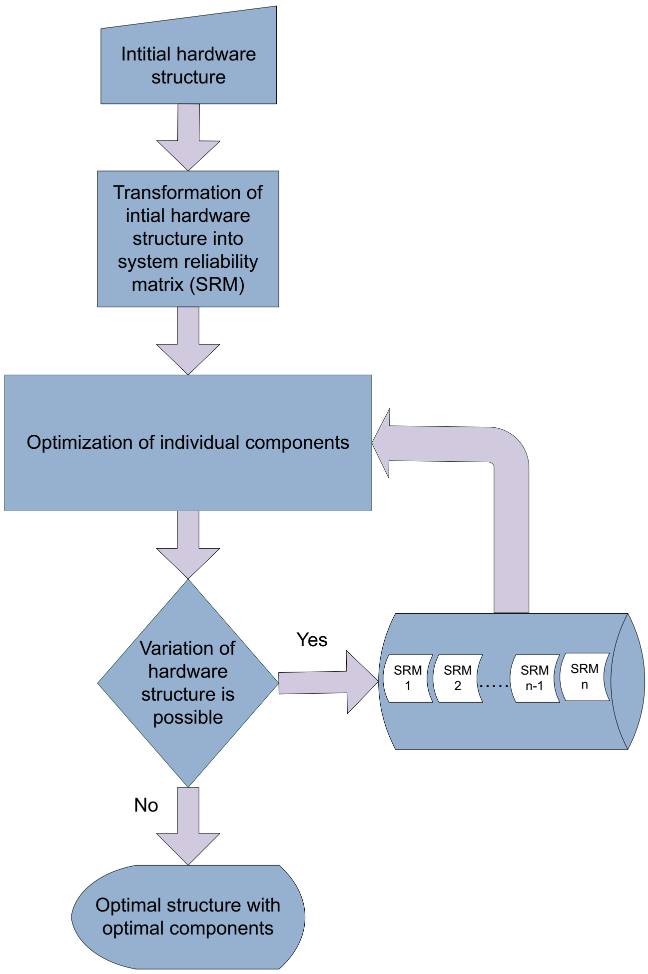


Figure 1.14.: The developed concept for determination of optimal hardware topology with optimal feasible hardware components [KJS10]

i) development of system reliability matrix (SRM) for automatic reliability calculation of each arbitrary hardware structure
ii) visualization method to demonstrate the conflict objectives (reliability, cost, weight, size) subject to linear and nonlinear constrains
iii) using deterministic and stochastic techniques for optimization of component characteristics regarding reliability, cost, size, weight, and technical requirements during the design phase

iv) Development of an approach to search for optimal feasible architectures.

The all four steps are integrated into a comprehensive framework [KJS10] as shown in the in the figure 1.14.

1.6. Dissertation outline

Chapter 2 presents the general basics which are used for this contribution. Chapter 3 deals with literature review of multi-objective problems, multi-objective methods, comparison of optimization methods, and reliability optimization methods which are applied in the past years. In this chapter, an overview about existed reliability optimization literatures has been given. Chapter 4 introduces the framework for meeting the challenges like reliability, cost, size, and weight and etc during early design phase. Chapter 4 specifies the detailed description of above four steps. Chapter 5 applies the developed methodology on two practical examples. The first optimization example is aimed at specifying the most reliable and affordable topology with optimal component properties for a steer-by-wire prototype. The second example analyzes and optimizes a typical safety critical mechatronic system which are using fusion data. Chapter 6 closes this contribution with detailed summary and an outlook.

2. Definitions and basics

2.1. Definitions

Reliability
is mostly defined as the ability of an item to perform a required function under defined conditions during a specified period of time [MMS07] [Dhi99] [MP10]. Basically the observed time to failure, is a random variable T which represents the lifetime of an item. The probability distribution function of T is given by

$$F(t) = P(T \leq t), \qquad 0 < t. \tag{2.1}$$

$F(t)$ is defined as unreliability at time t. It represents the probability of failure in the interval [0, t]. The reliability function is given by

$$R(t) = P(T \leq t) = 1 - F(t). \tag{2.2}$$

The reliability function is the probability of no failures in the interval [0, t].

Failure density function $f(t)$
is expressed mathematically by

$$f(t) = -\frac{\mathrm{d}R(t)}{\mathrm{d}t}, \tag{2.3}$$

where $R(t)$ represents the item reliability at time t and $f(t)$ is the failure density function [Dhi99].

Hazard rate function $\lambda(t)$
is expressed by

$$\lambda(t) = \frac{f(t)}{R(t)}, \tag{2.4}$$

where $\lambda(t)$ denotes the item time dependent failure rate [Dhi99].

The time dependent failure rate associated with hazard rate function and item reliability can be yielded by substituting equation 2.3 into equation 2.4. Consequently the failure rate function can be phrased by

$$\lambda(t) = -\frac{1}{R(t)} \cdot \frac{\mathrm{d}R(t)}{\mathrm{d}t}. \tag{2.5}$$

General Reliability Function $R(t)$
can be derived by rephrasing the equation 2.5 into the following form

$$R(t) = e^{-\int_0^t \lambda(\tau)\mathrm{d}\tau}. \tag{2.6}$$

Basically the general reliability function can be deployed to obtain reliability of an item when its times to failure follow known statistical distribution, for example, exponential, Rayleigh, Weibull, gamma, and etc. Generally, non-constant failure

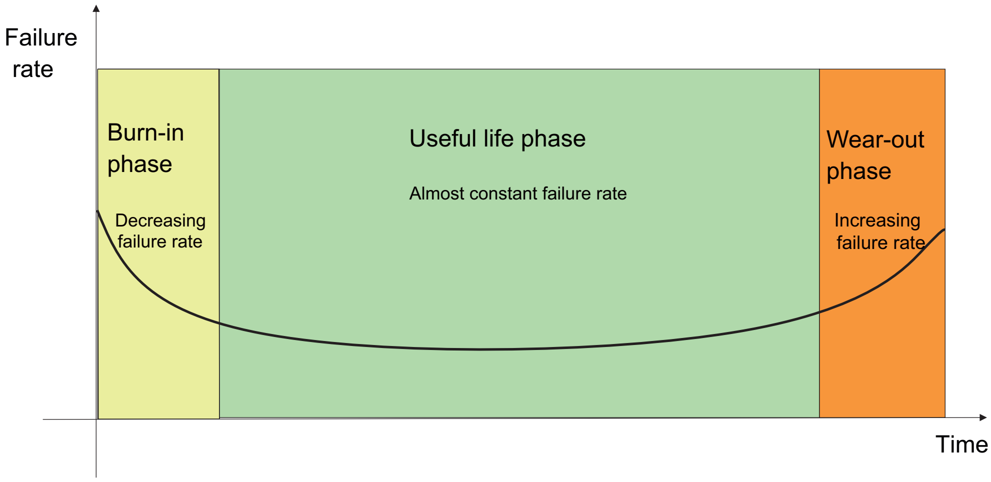


Figure 2.1.: Bathtube curve

rate curve ("bathtube" curve) can be illustrated in the figure 2.1.

Failure rate of each component is approximately dividable in three phases [Ams77]:

- infant mortality region (i.e. first months of product operation)
- useful region
- terminal mortality region (i.e. at the end of product operation).

This curve is the result of three types of failures:

- quality,
- stress-related, and
- wear out [IAU76].

Infant mortality region (early failure period) and terminal mortality region (wear-out period) are basically due to design errors or manufacturing process errors.

In the majority of publication including this dissertation, the failure rates of electronic components are assumed to be a constant value [Sta09]. In order to constitute this assumption, following care shall be taken that:

– the wear-out period for hardware component is far removed from the usage time interval and

– the manufacturing process and test strategy shall ensure that failures during early failure period are insignificant [SFP+12] [MP10].

The aim of this dissertation is to focus on the intrinsic failure rate. The failure rate for electronic components is expressed in units of FIT, where 1 FIT $= 1 \cdot 10^9/\mathrm{h}$ (1 failure per 10^9 component hours) [SFP+12].

Designers mostly are using different approaches illustrated in the figure 2.2 in order to estimate the failure rate of hardware components.

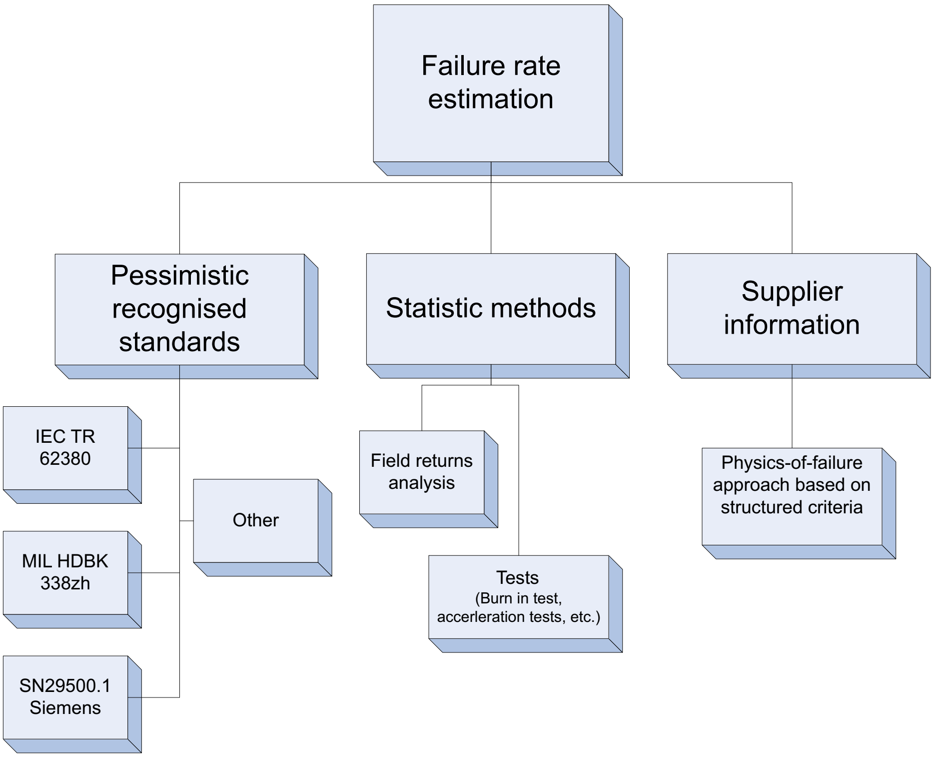


Figure 2.2.: Failure rate estimation methods

The most practical and recognized approach of failure rate estimation appears to be pessimistic standard like (IEC or MIL standards etc.). The main reason behind this trend typically lies in high effort of validation and test program. Reliability prediction models in the standards offer mathematical description in order to illustrate the external influence on failure rates of hardware components. The failure rate depends on several factors such as

– mission profile (i.e. temperature profile and field use condition) and

– stress environment (i.e. electrical voltage, current overloads, humidity, mechanical shock, and vibrations [CM07] [SFP+12].

The external stresses and factors can be categorized in following ways [CM07] [Int04]:

1. **Temperature factor**: It is widely accepted that temperature has a moderate to strong impact on component reliability [Int04]. The negative effect of high ambient temperature on life time is significant for some families (e.g. active components on the PCB like aluminum capacitors with non-solid electrolyte). Figure 2.3 shows how the temperature could damage the electronic components on the PCB.

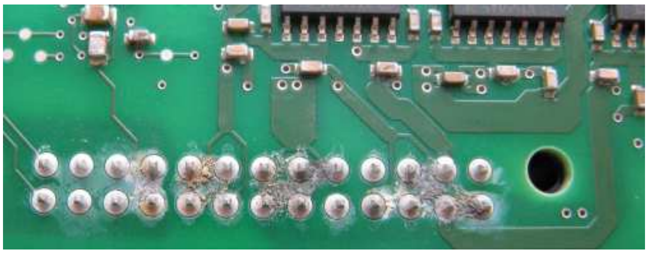


Figure 2.3.: Temperature impact on a electronic component (photo taken at the TRW laboratory)

   Indeed there exists components that are limited to be used in the hotter environment. For example most passive components like ceramic chip capacitors are deployed in an environment temperature of less than 125 °C [SFP+12].

   The mission profile defines local ambient temperature in time and temperature cycle parameters due to power dissipation as described in the IEC TR 62380. This standard applies two mission profiles in the vehicle context namely [Int04] [SFP+12]:

   – motor control and

   – passenger compartment.

   For instance failure rates of a ceramic capacitor on the ECU located close to engine bay is estimated with 2.79 FIT. while the failure rate of same electronic component on the ECU located close to the passenger compartment is estimated with 1.78 FIT [SFP+12]. Hence decision on components takes influence where the system shall be installed in the vehicle.

2. **Electrical stress due to power supply**: Failure rate of components also heavily depends on electrical environment conditions (over-voltage and current overloads). The example shown in the figure 2.4 illustrates how the over-voltage generate high temperature on the PCB and destroys the components.

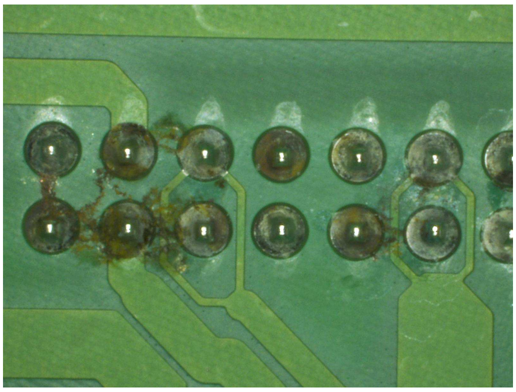


Figure 2.4.: Over-voltage impact on a electronic component (photo taken at the TRW laboratory)

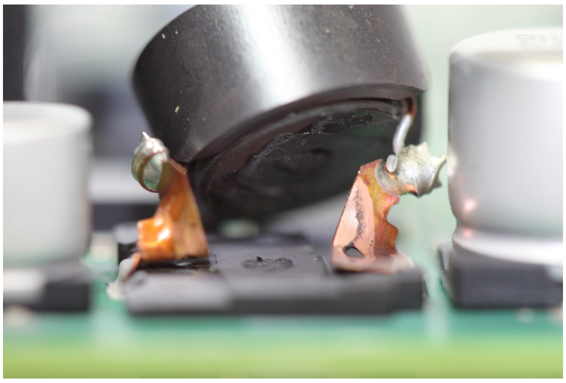


Figure 2.5.: Vibration impact on a electronic component (photo taken at the TRW laboratory)

3. **Environmental conditions**: Parameters like vibration, noise, dust, pressure, relative humidity and shock can influence the failure rate value as illustrated in the figure 2.5. The figure 2.5 shows how the vibration detaches the components from PCB.
4. **Manufacturing quality**: Manufacturing technology has an impact on the failure rates of hardware components

Due to above stress factors, the failure rates of components in a similar class for a specific application locate in an scalar value interval. The exact estimated value of failure rate of components depends strongly on environmental conditions and manufacturing technology. During proposed optimization process, the exact failure rates of components can be determined. The selected components by proposed dissertation approach typically impose constraints on the design, environmental and manufacturing assumptions [CM07].

2.2. Reliability block diagram

To evaluate the reliability of a system, different prediction methods can be used:

- Reliability block diagram (RBD),
- Fault tree analysis,
- Markov analysis,
- Monte Carlo simulation, and
- etc.

In this dissertation, RBDs are used for modeling and calculation of the overall system reliability. RBD represents the connections between system components from a reliability perspective. The purpose of the RBDs is to show the association of required hardware components to system operational success [MP10] [OHTS11].

It must be noted that RBD basically displays the connections between the hardware elements from reliability point of view and does not show the process flow. The RBDs generally have elements connected in parallel, series, bridge, and k-of-n structure. It is assumed that the failure of any component is independent of the failure or success of other involved components. RBD is a block diagram of a system showing all essential functions required for system operation [Sta09].

In the typical RBD, if all elements are necessary for faultless operation of the system, all blocks are strung together. This type of reliability relationship is a series relationship. The structure of components are said to be parallel if the functional path can be split into two individual structures. In other words, either individual structure is adequate to execute the function. For example, if two motors are running in parallel to control and power a group of valves where either motor fulfills the power requirement independently. Then is it assumed that the functional condition of one motor has no impact on the healthy condition of other motor. The system can keep running if one of the motors fails [OHTS11] [Sta09].

Constructing a right RBD from typical technical schematic is one of most important activities which mostly is underestimated in the literature [Kon07]. Since a translation of the schematic into RBD without technical expertise is not possible. In order to make the case two important challenges are addressed as follows:

Challenge 1:
The first chalenge is approached with the aid of simple example given in the following figure 2.6. The schematic can not be converted automatically into RBD. Intuitively the right RBD in the figure 2.6 (inductor and capacitor in parralel) can be chosen. While the right RBD does not show the real technical connection from reliability point of view. That is the reason why it is highly recommended to construct the RBD with help a of experienced designers [Kon07].

One of typical faulty scenario is illustrated in figure 2.6.

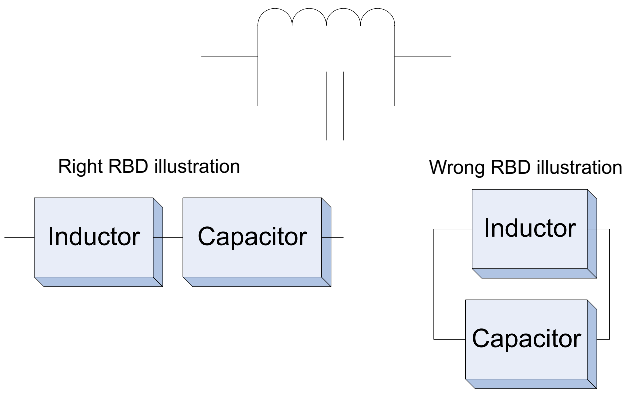


Figure 2.6.: Converting of a schemetic into RBD - Example from [Kon07]

Challenge 2:
Complex systems typically have different functions which require constructing different RBDs at time. The example in the figure 2.7 illustrates a typical automotive mechatronic system which performs two different functions. The figure 2.8 shows the components which are involved for first function execution. The involved components are highlighted by red in the figure 2.8. The corresponding RBD can be determined by using expert knowledge as shown in the figure 2.9. The figure 2.10 displays the components which provide the second function execution. The involved components are highlighted by red. The corresponding RBD can be determined by using expertise of expert as depicted in the figure 2.11. Now, the challenge is to decide which RBD is demonstrating the reliability of this complex system.

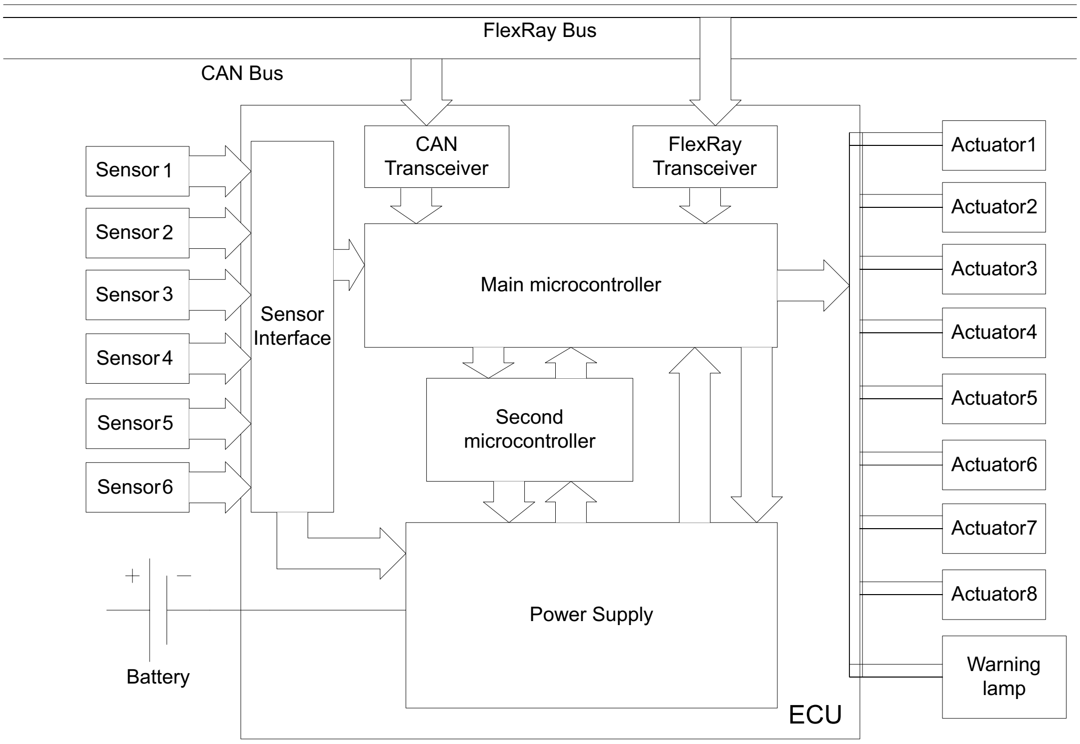


Figure 2.7.: Typical automotive mechatronic system - Example

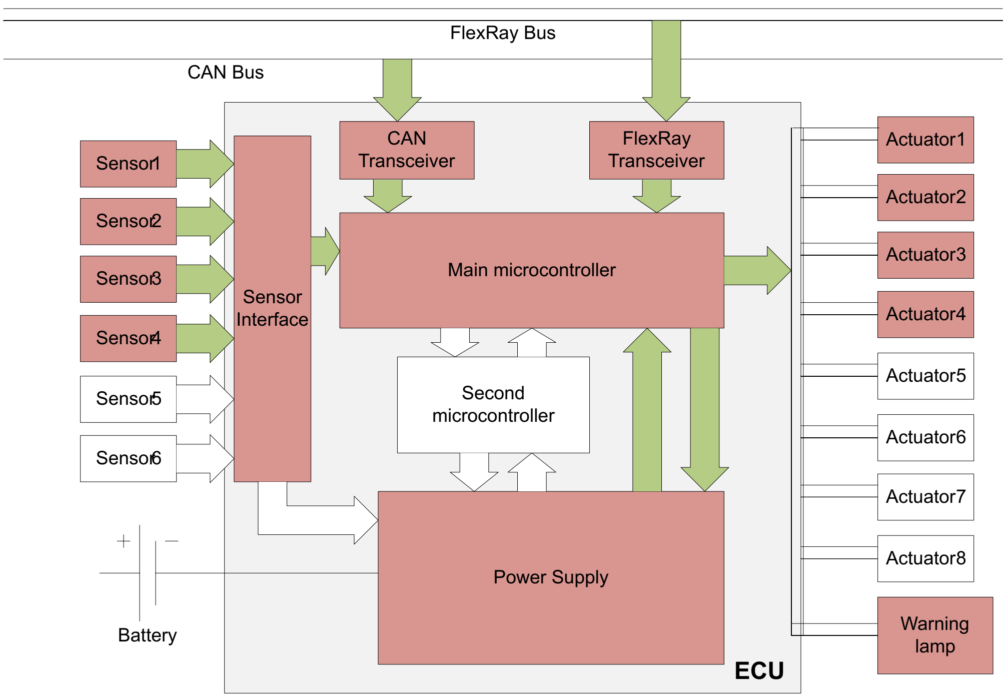


Figure 2.8.: The involved components for performing function 1

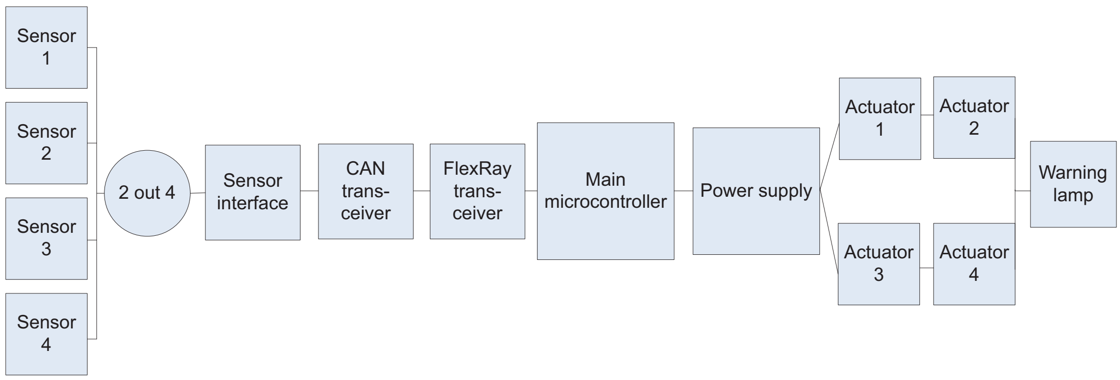


Figure 2.9.: RBD for function 1

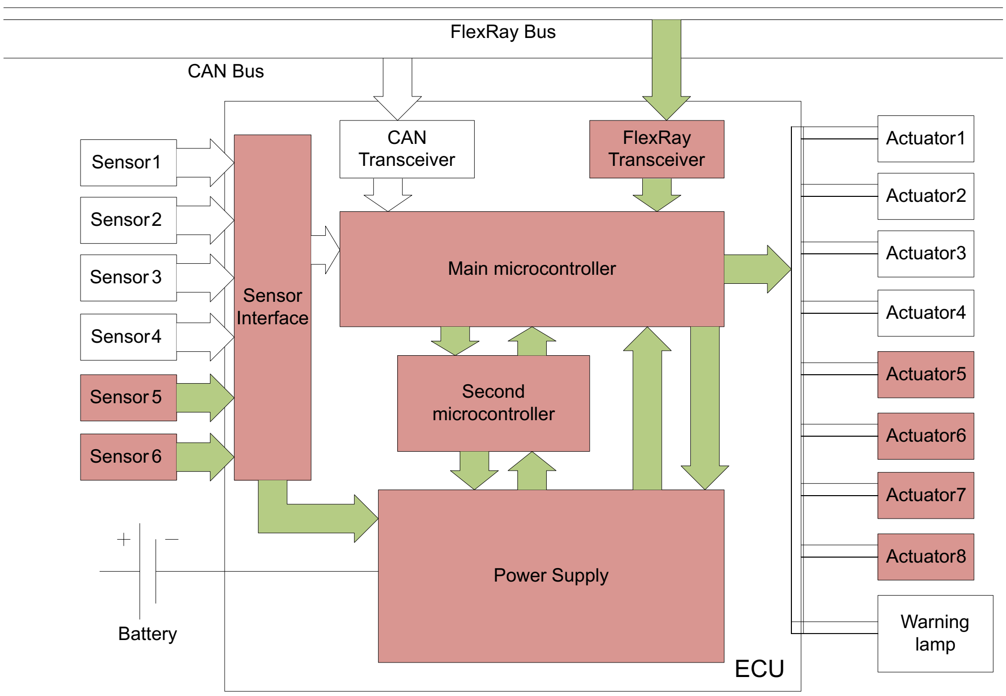


Figure 2.10.: The involved components for performing function 2

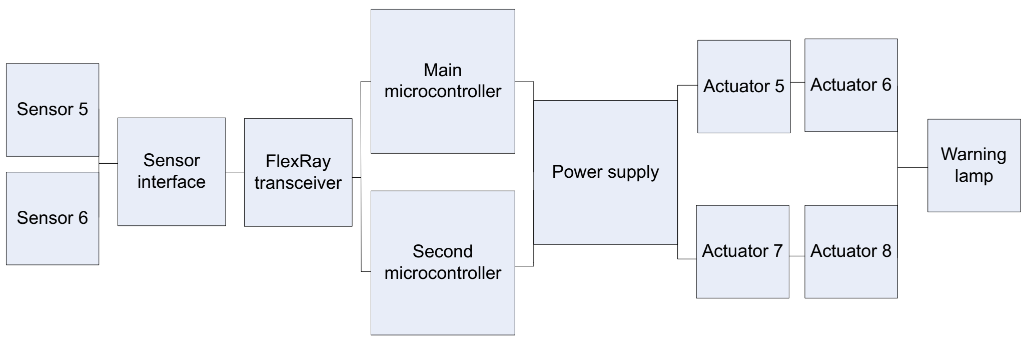


Figure 2.11.: RBD for function 2

In this dissertation, the different RBDs can be considered during design optimization. In order to model a system using RBD, most important structures are divided to following elementary structures:

Series structure

The simplest structure in reliability mathematical modeling is the series structures as depicted in the figure 2.12. The reliability of series structure can be calculated

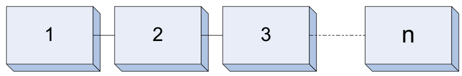


Figure 2.12.: Components in a series structure

by

$$R_{\mathrm{S}} = \prod_{i=1}^{n} R_i, \qquad i = 1, 2, \ldots, n. \tag{2.7}$$

Parallel structure

The other elementary structure in reliability modeling is the parallel structures, as illustrated in the figure 2.13. The reliability of parallel structure is given by

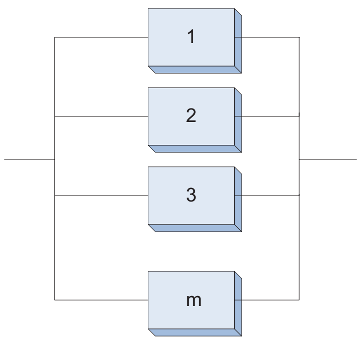


Figure 2.13.: Components in parallel

$$R_{\mathrm{P}} = 1 - \prod_{i=1}^{m} (1 - R_i), \qquad i = 1, 2, \ldots, m. \tag{2.8}$$

k-out-of-n system structure

In the automotive domain, deploying of k-out-of-n system structure is an attractive type of redundancy in fault tolerant x-by-wire systems. This type of system is called voting redundancy. As illustrated in the figure 2.14, the voter structure has n parallel outputs which are linked through a decision-making device that delivers the required

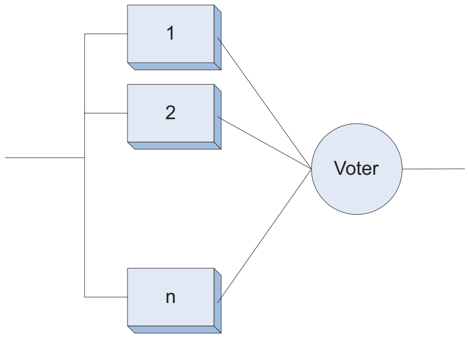


Figure 2.14.: Components in a voter structure

system function as long as at least k of n parallel outputs are in agreement. If the reliability values of the n individual components are same $R(t)$, then the reliability of the whole structure results to

$$R_{\text{k out of n}}(t) = R_{\text{voter}} \sum_{i=k}^{n} \binom{n}{i} R^i(t)(1 - R(t))^{n-k}, \tag{2.9}$$

where R_{voter} denotes the reliabilty of decision making device [Dhi99].

For cold-standby, redundancy detection and switching mechanism are required to sense the presence of a failed component and to activate a standby component. The reliability of a series and of a parallel system with cold-standby redundancy and imperfect switching is given by

$$R(t) = \prod_{i=1}^{n} \left(r_i(t) + \sum_{x=1}^{n_i - 1} \int_{u=0}^{t} \rho^x(u) r_i(t-u) f_i^x(u) \mathrm{d}u \right), \tag{2.10}$$

where $r_i(t)$ denotes the components reliability used in subsystem i at time t; the pfd for the x-th failure arrival for the subsystem u is expressed by $f_i^x(u)$ i.e., the sum of x ii-d components failure time, and $\rho^x(u)$ represents the reliability of detection/switching [Coi01]. For the simple case of two components in standby redundancy, assuming components identical failure rates with exponential time-to-fail distribution, the reliability of the system is

$$R(t) = e^{-\lambda t}(1 + p\lambda t). \tag{2.11}$$

Bridge structure

In some applications, the components may be connected in bridge structure as illustrated in the figure 2.15. The reliability calculation of a typical bridge structure is

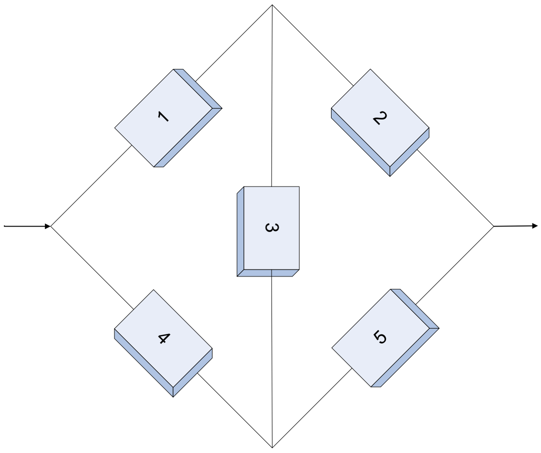


Figure 2.15.: Reliability bridge structure

given by

$$
\begin{aligned}
R_{\text{br}} =& 2\prod_{i=1}^{5} R_i + \prod_{i=2}^{4} R_i + R_1 R_3 R_5 + R_1 R_4 + R_2 R_5 \\
& - \prod_{i=2}^{5} R_i - \prod_{i=1}^{4} R_i - R_5 \prod_{i=1}^{3} R_i - R_1 \prod_{i=1}^{5} R_i - R_1 R_2 R_4 R_5,
\end{aligned}
\tag{2.12}
$$

where R_{br} denotes the bridge network reliability and R_i describes the i-th unit reliability for $i = 1, 2, 3, \ldots, 5$.

For simplicity bridge structure is transformed into series and parallel structure using delta star technique in the given dissertation. Transforming a bridge structure into series and parallel structure using delta star method often leads to a small error in the final system reliability calculation. The generated small error can be ignored for engineering application purposes [Dhi99].

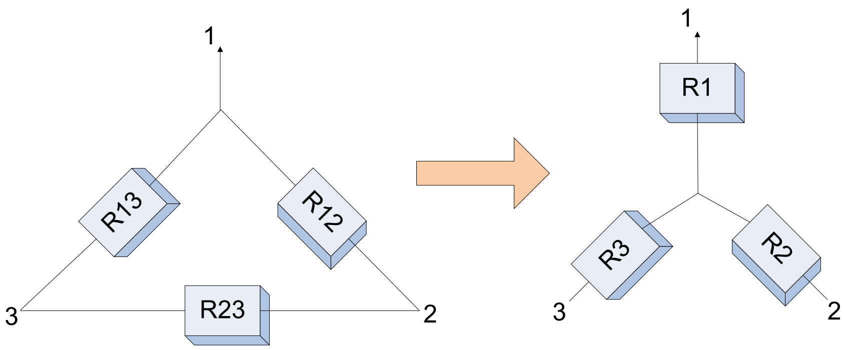


Figure 2.16.: Delta star technique for solving bridge structure

As presented in the figure 2.16, the equivalent reliability equations for the network reliability between notes 1, 2; 2, 3; and 1, 3 can be phrased into the following expressions

$$R_1R_2 = 1-(1-R_{12})(1-R_{13}R_{23}) \tag{2.13}$$
$$R_2R_3 = 1-(1-R_{23})(1-R_{12}R_{13}) \tag{2.14}$$
$$R_1R_3 = 1-(1-R_{13})(1-R_{12}R_{23}). \tag{2.15}$$

respectively [Dhi99].

Solving equations 2.13 to 2.15, results into the expression

$$R_1 = \sqrt{\frac{AC}{B}}, \tag{2.16}$$

where

$$A = 1-(1-R_{12})(1-R_{13}R_{23})$$
$$B = 1-(1-R_{23})(1-R_{12}R_{13})$$
$$C = 1-(1-R_{13})(1-R_{12}R_{23})$$
$$R_2 = \sqrt{\frac{AB}{C}}$$
$$R_3 = \sqrt{\frac{BC}{A}}$$

[Dhi99].

2.3. Functional Safety - ISO 26262

In addition to reliability concerns, the functional safety standards like ISO 26262 [Int11] require probabilistic evaluation of each mechanism used to reduce risk in a safety related system. The failure rates λ of each safety critical components can be

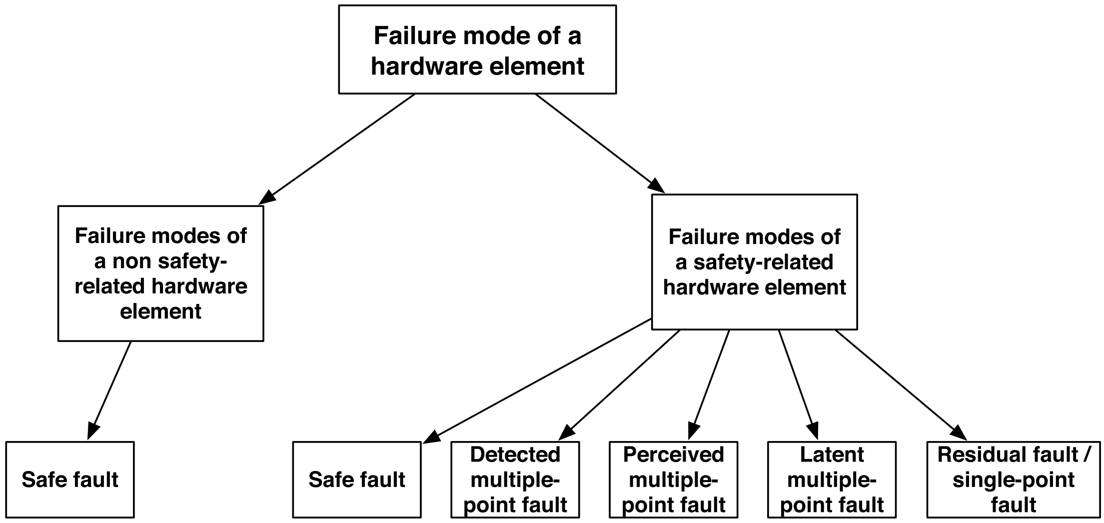


Figure 2.17.: Failure mode classification of a hardware element [Int11]

expressed as follows

$$\lambda = \lambda_{\text{SPF}} + \lambda_{\text{RF}} + \lambda_{\text{MPF}} + \lambda_{\text{S}}, \tag{2.17}$$

where λ_{SPF} and λ_{RF} represent the failure rate associated with single point faults and the failure rate connected to residual faults respectively. Note that λ_{MPF} and λ_{S} denote multiple point faults and failure rates associated with safe faults [Int11].

The failure rates regarding residual faults can be expressed as follows

$$\lambda_{\text{RF, estimated}} = \lambda \cdot \left(1 - \frac{DC_{\text{with respect to residual faults}}}{100}\right), \tag{2.18}$$

where $DC_{\text{with respect to residual faults}}$ stands for the diagnostic coverage of safety mechanism which prevents single point faults of hardware components [Int11].

The failure rates connected to latent faults is given using the diagnostic coverage of safety mechanism that prevents latent faults

$$\lambda_{\text{MPF, estimated}} = \lambda \cdot \left(1 - \frac{DC_{\text{with respect to latent faults}}}{100}\right), \tag{2.19}$$

where $DC_{\text{with respect to residual faults}}$ indicates the diagnostic coverage of safety mechanism which prevents multiple point faults of hardware components [Int11].

According to ISO 26262 [Int11] there are two significant metrics from hardware point of view which provide an evidence if the safety goals are achieved. Investigation of these metrics is illustrated in the figure 2.18.

The single point fault metric (SPFM)

$$\text{SPFM} = 1 - \frac{\sum\limits_{\text{safety critical HW components}} (\lambda_{\text{SPF}} + \lambda_{\text{RF}})}{\sum\limits_{\text{safety critical HW components}} \lambda} \tag{2.20}$$

illustrates the robustness of safety critical hardware components against single point and residual faults. High single point fault metric indicates the portion of single point faults and residual fault are high in the hardware components.

The latent point fault metric (LPFM)

$$\text{LPFM} = 1 - \frac{\sum\limits_{\text{safety critical HW components}} (\lambda_{\text{MPF}})}{\sum\limits_{\text{safety critical HW components}} (\lambda - \lambda_{\text{SPF}} - \lambda_{\text{RF}})} \tag{2.21}$$

shows the robustness of the hardware components against multiple point faults.

The required single point fault metric and latent point fault metric values are given in the table 2.1. One of potential sources for defining a quantitative target for

	ASIL B	**ASIL C**	**ASIL D**
Single point fault metric	$\geq 90\,\%$	$\geq 97\,\%$	$\geq 99\,\%$
Latent fault metric	$\geq 60\,\%$	$\geq 80\,\%$	$\geq 90\,\%$

Table 2.1.: Target values for single point fault metric and latent point fault metric according to ISO 26262 [Int11]

random hardware failure evaluation is given in the table 2.2 [Int11].

ASIL	**Random hardware failure target values [1/h]**
D	$< 10^{-8}$
C	$< 10^{-7}$
B	$< 10^{-7}$

Table 2.2.: Possible source for the derivation of the random hardware failure target values from ISO 26262 [Int11]

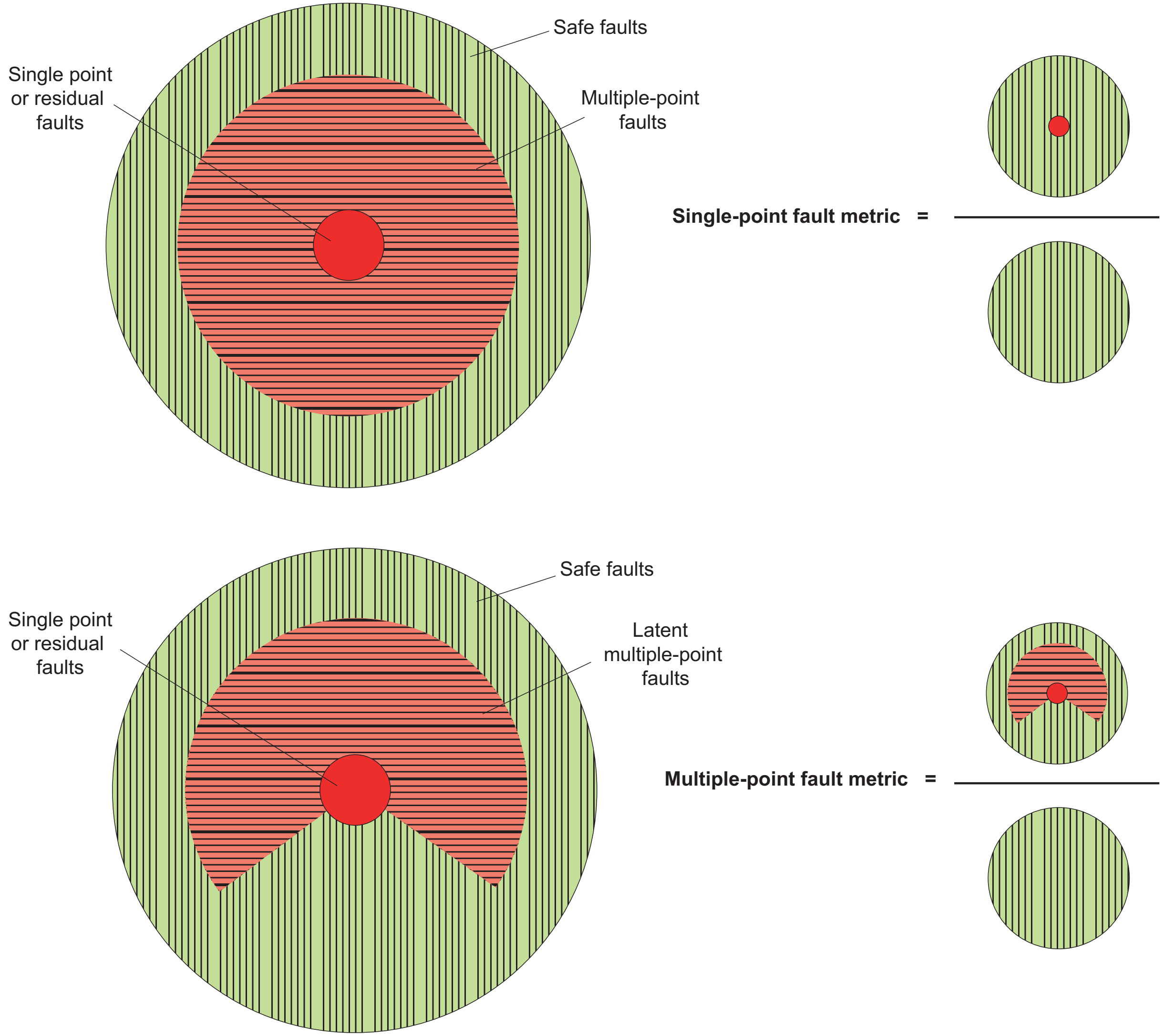


Figure 2.18.: Single and multiple point metric according to ISO 26262 [Int11]

2.4. Reliability Cost Relationship (RCR)

Reliability aspect of product certainly can grantee or destroy the short- and long-term success of a product due following reasons:

1. Designing too high reliable product could cause the product to be not affordable for customer and consequently market share will be lost.
2. Manufacturing a product with low reliability presumably will lead to high warranty and undesirable repair costs and therefore market share will be lost.

In order to enhance the reliability of a product, increasing the cost of the design and/or production are typically the negative consequences. The overall costs are calculated generally by integrating the total costs of the products through its lifetime. This includes guarantee and repair costs for defective products, costs initiated by loss of image against customers due to defective products, loss of future sales, etc. By raising product reliability, the initial product costs may increase, but the support costs decrease at times. Calculating the optimum reliability for such a product determines an optimum minimal total product cost. Such a scenario is demonstrated in the figure 2.19. The total product cost is the sum of the production and design costs. In many cases, it can be observed that the optimum reliability stands at the point which meets the minimum total cost over the entire lifetime of the product as illustrated in the figure 2.19. The conflict between the required reliability level of

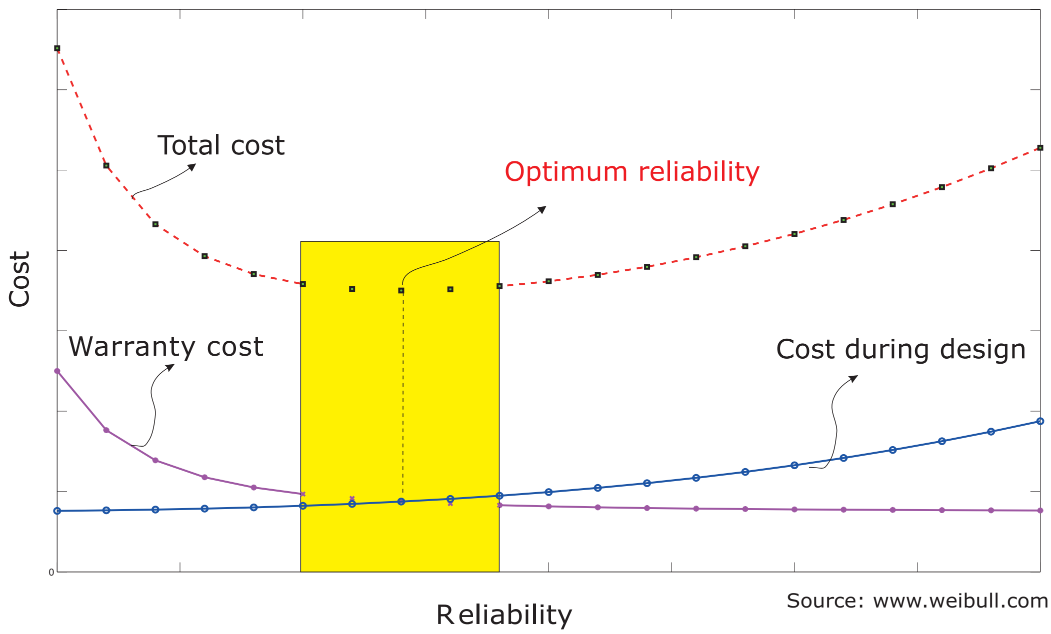


Figure 2.19.: Reliability Cost Relationship [Rel]

system and cost mostly occurs during design and maintenance phases. Reliability design should be ideally performed in a least-cost way. Some of the components with low reliability and critical allocation may demand special attention to improve the overall reliability level. There exists two methods to display this conflict mathematically and express the relationship between the cost of individual components and reliability levels of them.

1. One of the most useful approach for cost formulation based on the failure rate of each single component. For any single component, the cost function is defined by

$$C(\lambda_i) = \frac{a_i}{b_i - \frac{1}{\lambda_i}}, \tag{2.22}$$

where a_i and b_i are fixed cost parameters and $1/\lambda_i$ denotes the value of expected life-time [Met00]. As illustrated in the figure 2.20, cost function holds two asymptotes. The vertical asymptote indicates technical specification regarding reliability. The horizontal asymptote shows the minimal costs.

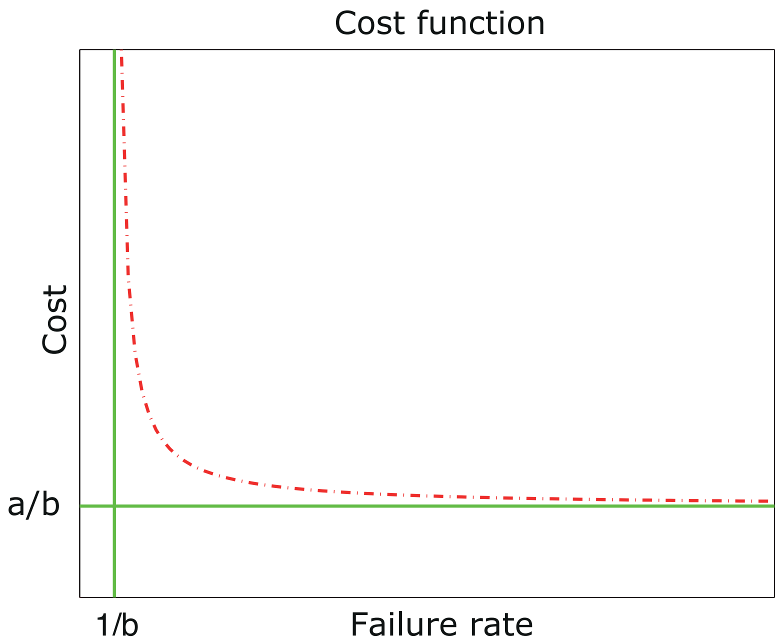


Figure 2.20.: Cost function of a single component

2. The relationship between cost and reliability level can be given empirically. In the majority of cases, this type of methods assumes that the relationship can be specified based on the given data of history or data for identical components. Nevertheless, in order to find the cost of each component, a general formulation [Met00] is proposed as

$$C_i(R_i;\, R_{i,\min};\, R_{i,\max};\, f_i) = e^{\left[(1-f_i)\frac{R_i - R_{i,\,\min}}{R_{i,\,\max} - R_i}\right]}, \tag{2.23}$$

where $R_{i,\min}$ and $R_{i,\max}$ represent the lower and upper reliability bound of each single component. The feasibility is described by f_i if the reliability of component rises in value. The feasibility parameter is subject to design complexity, technological constraints, and management priorities. The effect of feasibility value satisfies the following rule: the lower the feasibility value, the more quickly the cost function increase.

In this dissertation, all to be optimized parameters (cost, size, weight, etc.) are defined in association with failure rates like reliability. Reducing the number of optimization parameters and illustration of the relationship between the objective functions and failure rates of each hardware components are the most important benefits of this step. Typically the discrete data for component properties exists from suppliers (e.g. cost and failure rate of component, mission profile, etc.). The relationship between financial and technical aspects and failure rate of components can be formulated over a continuous function.

Failure rate 10^{-6} [1/h]	**Cost** [-]
0.3	98
0.4	290
0.5	266
0.6	380
0.65	280
0.7	450
0.75	303
0.8	400
0.9	350
0.95	390
0.99	450

Table 2.3.: Cost and failure rates of a set of hardware components

There are generally two typical methods for curve fitting. The first one is to use first polynomials or splines and the other one is to use simple polynomials which will be fitted by the least squared error method. The both methods are used to approximate the system behavior and to display an interactive graph. Relating to the topic of the thesis one simple example of both curve fitting variants is given in figure 2.21. In this case the data about the cost of specific components is associated with its failure rates. The green curve represents the least error squared method, the red one a spline interpolation. In contrast to the least squared method the fitting curve using first polynomials or splines crosses all given data points.

Figure 2.21.: Different ways of failure rates and cost interpolation

2.5. Algorithm for nonlinear constrained optimization problem

In the reliability design, there are different kinds of optimization problems with inequality and equality nonlinear constraints. There are different methods for solving such problems with equality und inequality nonlinear constraints. In this dissertation, the method of Lagrange multipliers is used to find the optimal solutions.

The reliability-cost optimization problem can be defined by

$$\text{min. } z = \sum_{i=1}^{n} C_i \varepsilon_i^2 \tag{2.24}$$

$$\text{subject to} \quad \prod_{i=1}^{n} (R_i + \varepsilon_i) \geq R_{\text{required}} \quad \text{and} \tag{2.25}$$

$$R_{\min} < R_i + \varepsilon_i \leq R_{\max} < 1, \tag{2.26}$$

where ε_i is the reliability improvement for component i [BS79] [MP06] [Han92]. Cost function is quadratic function because reliability growth drives up cost at an increasing rate. The Lagrangian for problem results to

$$L(\varepsilon_i; \theta) = \sum_{i=1}^{n} C_i \varepsilon_i^2 - \theta \cdot \left[\prod_{i=1}^{n} (R_i + \varepsilon_i) - R_{\text{required}} \right], \tag{2.27}$$

where θ is the Lagrangian multiplier. The key notion behind the Lagrangian multiplier method is to take the constraints into consideration by augmenting the objective function with a weighted sum of the constraint functions. After taking the partial derivatives of L and two independent variables, the following equation for optimization are given as

$$\frac{\delta L(\varepsilon_i;\, \theta)}{\delta \varepsilon_i} = 2C_i\varepsilon_i - \theta \cdot \left[\prod_{j=1}^{i-1} (R_j + \varepsilon_j) \cdot \prod_{j=i+1}^{n} (R_j + \varepsilon_j) \right] = 0 \quad \text{and} \tag{2.28}$$

$$\frac{\delta L(\varepsilon_i;\, \theta)}{\delta \theta} = \prod_{i=1}^{n} (R_i + \varepsilon_i) - R_{\text{required}} = 0. \tag{2.29}$$

After multiplying the term $(R_i + \varepsilon_i)$ to equation 2.28, the equation can be rearranged to

$$2C_i\varepsilon_i(R_i + \varepsilon_i) = \theta \cdot R_{\text{required}}. \tag{2.30}$$

By restructuring the above term, the result is given by

$$2C_i\varepsilon_i^2 + 2C_i\varepsilon_i R_i - \theta \cdot R_{\text{required}} = 0. \tag{2.31}$$

The reliability improvement is given as

$$\varepsilon_i = \frac{-C_i R_i + \sqrt{C_i^2 R_i^2 + 2C_i \theta R_{\text{required}}}}{2C_i}. \tag{2.32}$$

The above expression provides the possiblity for a sensitivity analysis. For example, it can be determined in a objective way what is the cost consequence of reliablity improvement [SBMR+12] [KLXZ87]. With above procedure, other possible technical constraints (like weight, volume, etc.) can be also integrated into the objective function.

3. Literature review

3.1. State of the art in optimization area

For realizing a certain engineering example product designers must satisfy different kind of criterias. These are especially to

- minimize the product cost and energy consumption,
- maximize the reliability,
- enhance the comfort, and
- increase the functional safety.

The process of maximizing or minimizing of criteria are defined as optimization process. Depending on type of product design development, various factors like type of material, design structure, applied technology, outside temperature (mission profile), given pressure, etc. can take influence on fulfillment of criteria which are called design decision variables. In order to specify how well conflicting criteria are fulfilled, a set of objective functions are determined. The to be considered technical and financial limitations are recognized as constraints.

Optimization of conflicting objective functions under given constraints often means achieving the optimum for one objective or and best possible compromises among other objectives. In the literature, such problems are recognized as multi-objective optimization problem. Basically the challenge of design optimization is divided into two following parts:

1. formulation of design optimization problem
2. choosing an optimization techniques.

3.1.1. Formulation of design optimization problem

In general the multi-objective problem is expressed by

$$\min_{\mathbf{x}}. \quad \text{or} \quad \max_{\mathbf{x}}. \quad \mathbf{F}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) & f_2(\mathbf{x}) & \dots & f_i(\mathbf{x}) & \dots & f_k(\mathbf{x}) \end{bmatrix}^T \tag{3.1}$$

subject to

$$\begin{aligned} &J_j(\mathbf{x}) \leq 0, \quad j = 1, 2, \dots, m, \\ &K_l(\mathbf{x}) = 0, \quad l = 1, 2, \dots, n, \end{aligned}$$

where k denotes the number of objective functions, m represents the number of inequality constraints, and n provides the number of equality constraints. f_i denotes

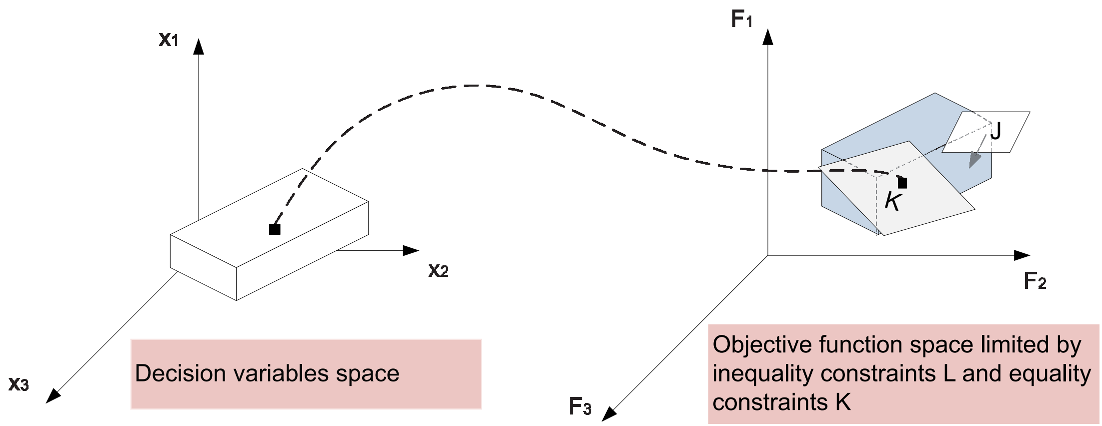


Figure 3.1.: Impact of equality and inequality constraints on objective functions

the i^{th} objective function, and $\mathbf{x} = \begin{bmatrix} x_1 & \dots & x_n \end{bmatrix}$ denotes the design variables which are subject to change during optimization process [KPTH06] and [Bes06]. In the many engineering application including reliability optimization cases, each objective function is supposed to be scaled to limited range due to value ranges of various functions. For instance, reliability values are basically moving between 0 and 1 while cost or weight lies in much greater scalar value intervals.

The design objectives are normalized to have the same order of magnitude. To do the normalization, it is subject to the condition that for each objective the designer provides a target value (an estimate of a desired target or good design) and a bad value (an estimate of an undesired design). The most practical scaling methods have been introduced (see table 3.1) by [MA04].

Scaling function	**Range**	**Reference**
$f_i^{\text{scaled}} = \frac{f_i(\mathbf{x})}{\lvert f_i^{\max} \rvert}$	$[0 \ \dots \ 1]$	[PSQX01]
$f_i^{\text{scaled}} = \frac{f_i(\mathbf{x}) - f_i^{\text{ideal}}}{f_i^{\text{ideal}}}$	$[0 \ \dots \ \infty]$	[Osy84], [HY81], [Sal79]
$f_i^{\text{scaled}} = \frac{f_i(\mathbf{x}) - f_i^{\text{ideal}}}{f_i^{\max} - f_i^{\text{ideal}}}$	$[0 \ \dots \ 1]$	[Kos84], [KS87], [RR89]

Table 3.1.: Overview of scaling functions in the recent literature; summarized in [MA04].

In the literature, there exists different approaches which have been applied for handling multi-objective optimization problems [FGF80]:

1. Scalarization approaches: For many engineering design problems, translation the conflicting objectives into a single scalar problems is the most practical approach. The precondition of using this approach is that the weights between objectives are known. One of major Challenges is due to the right choice of

the weights to nail down the decision-maker's preferences. The mechanism of scalarization is a key issue behind many different methods for multi-objective optimization issued in the literature [MA04] [FF95].

2. Goal programming: In many real engineering problems, it is reasonable to optimize one of the objective functions and set scalar goals and relative weights for each of the objective functions [JT10]. In dealing with goal programming problems, the multiple goal problems can be converted into a single objective which a weighted sum of deviations from goals shall be minimized [Ign78] [Rom91].

3. Pareto optimality approaches: For those multi-objective problem which the preferences between different objectives are not given, application of Pareto analysis is an appropriate approach [KL01] [VSVL08].

All three methods have advantageous and disadvantageous. Application of the scalarazation method and goal programming or combination of both of them (e.g. weighted sum of objective functions, weighted min-max formulation, etc.) is very popular due to its trivial implementation. Nonetheless, each objective function is supposed to be scaled due to various function value ranges. For instance, reliability values are basically moving between 0 and 1 while cost lies in much greater scalar value intervals. Scaling process requires a high expertise about the to be optimized system. Therefore using such methods during new innovative system development is not very simple [MA04].

Sometimes it is difficult for a single objective to sufficiently portray a real design problem [ZT99]. Due to this fact, analyzing a multi-objective system design problem deserves much more attention at times. For these multi-objective problems which the preferences between different objectives are not given, application of Pareto analysis is an appropriate approach. Pareto optimality approaches are preferred as well if designers are not able to assign weighting factor to each objective functions [ZT99]. However, as soon as the number of objective functions is growing, determination of the entire Pareto optimal set is practically impossible due to its size [KCS06]. The disadvantage of Pareto approaches is that designers are faced with many general solutions which decision making and analyzing them are very time consuming [And01], [Deb01].

3.1.2. Optimization techniques

In multi-objective optimization literature, the term ”optimize” is associated with finding a solution which offers the design team the acceptable and feasible values of all the conflicting objective functions. In general the optimization process splits up in the two parts of

1. evaluation of design initial ideas based on available technical and financial objectives and constraints and
2. generation of new feasible designs [And01].

In the literature, various approaches have been applied for solving multi-objective optimization problems. These approaches can be grouped into categorie of

1. enumerative,
2. deterministic,
3. and stochastic methods [Lue90] and [BSS93].

A complete literature review regarding nonlinear multi-objective optimization classification is available in [VV99] and [CVVL02]. Each of above methods offers some advantages and is subject to some limitations. Typically, the most time consuming but effective and robust optimization approach is to enumerate all feasible possible solutions. However precondition of using such methods is discretization of continues search space. Granularity of discretization process influences the chance of finding global optimum massively. This means the grosser is the granularity level of discretized search space, the higher is the likelihood of wrong global optimum determination. In practice, application of enumeration methods is merely reasonable if the number of variables is low and the amount of discretized search space is small.

Deterministic methods typically search for global optimum values within the optimization space, based on the function gradient information. Such methods use derivative information to find the next iteration [BSS93]. Such gradient-based methods use first derivatives (gradients) or second derivatives (Hessians), in order to search for right direction. The sequential quadratic programming (SQP), the augmented lagrangian method, and the nonlinear interior methods belong to this category. The success of deterministic methods application is dependent on the characteristic of objective function and its boundary condition [VV99].

They only yield good outcomes with functions which are continuous, convex, and unimodal [BSS93]. For instance, if the variables are related to each other in a nonlinear way or the problem is non-convex, the deterministic algorithms are not typically reliable. Indeed; deterministic optimization using gradient descent methods would converge much faster than stochastic optimization methods such as genetic algorithms (GA) or enumerative methods. Ultimately the stochastic algorithms (non-derivative) are most reliable methods because the stochastic search algorithms need only the objective function value and boundary condition value. The drawback of such methods is the random search nature of method and there is obviously no 100% guaranty for random finding global optimum. To be noted, that the stochastic algorithms is only able to estimate the optimum solution. For engineering problem, Evolutionary algorithms (EAs) are one category of non-derivative methods which are very popular in the reliability optimization methods. The benefit of using Evolutionary algorithms (EAs) in compare to gradient methods has been studied in many applications by Beasley [LNF11]. EAs need some knowledge about the multi-objective problem being optimized, and EAs s are trivial to implement, robust, and the probability of finding global optimum is higher than other available methods. EAs techniques are based upon the imitation of natural selection and biological evolution.

3.1.3. Some significant articles on the topic reliability optimization

Optimal design problems are generally formulated to maximize an appropriate system performance measure under resource constraints. Design optimization is basically associated with the performance characteristics of engineering systems. Measures of system performance for a mission under various conditions can be normally gauged as followed:

1. reliability,
2. availability,
3. mean time to failure, and
4. percentile life [KW07].

Reliability has been mostly employed as a system performance measure for non-repairable systems [KW07]. Availability could be deployed as the percentage of time the system really functions for a repairable system and the performance measure for renewable multi state system. Percentile life and mean time to failure is basically applicable if the system mission time is not exactly forcastable, as in most practical cases. As explained in the last chapter, the reliability of system can be improved by employing of redundancy at different levels of system or overdesigning the critical components. Both techniques increase presumably the reliability at the expense of some critical re-sources [KPTH01]. Regarding system reliability design optimization, three kinds of improvement possibilities have been basically addressed in the literature:

1. Component Reliability Allocation (CRA). The reliability of the components or subsystems is subject to change in order to find an optimum. Depending on given application, CRA could be performed continuously or discretely. Discrete allocation is related to the selection of reliability values from off-the-shelf. While component reliability could be determined in a continuous search space by using gradient or non-gradient methods.
2. Topology Reliability Allocation (TRA). Reliability of system structure is subject to change by selecting right redundancy levels for the subsystems or components.

3. Combination of above methods (CRA and TRA).

Other technical and financial considerations like low cost, high safety, low weight, small size, and low environmental pollution are contemporaneously required to taken into account. These aspects are often in conflict with one another. These conflicting objectives and corresponding constraints pose limitation on the amount of redundancy level and the kind of component overdesigning. There exists many researches about the reliability optimization of multi-objective problems. A comprehensive survey about multi-objective problems has been introduced in [KPTH06]. A discussion on solving approaches regarding optimal reliability allocation is presented in the literature survey [KPTH06], [KPTH01] and [KW07].

The most applicable formulation of reliability optimization is given as

$$\text{max. } R_{\text{system}} = f(\mathbf{x}) \quad \text{or} \quad \text{min. } C_{\text{system}} = f(\mathbf{x})$$

or defined by

$$\begin{aligned} & g(\mathbf{x}) \leq d_i; && \text{for } i = 1, \ldots, m && \text{Linear constraints} \\ & q(\mathbf{x}) \leq e_i; && \text{for } i = 1, \ldots, n && \text{Non-linear constraints} \\ & R_{\text{system}} \geq R_{\text{required}}. \end{aligned}$$

Such problem formulations offer typical traditional reliability redundancy allocation problem with either reliability or cost as the objective function. Additionally various linear or nonlinear constraints could be involved simultaneously. Variety of reliability experts has applied this formulation in their investigations for instance [FHL68], [CS96b], [MS91], [ADS03], [CL00], and [Coi03]. A comprehensive survey on this problem formulation is given [KW07]. One of the first significant efforts to use deterministic methods has been made by [ML73]. A generalized Newton's method has been applied to determine the optimal solution for a cost reliability problem.

It must be noted that initial guess and the achieved optimum were not far away from each other inside of feasible region. Another interesting approach has been published in [FHL68] which employed dynamic programming for system reliability allocation. Aim of the paper [FHL68] was to select components and redundancy-levels based on the formulated objective function which is optimized under given system-level constraints on reliability, cost, and weight. It also has been analyzed how the value of Lagrange multiplier could influence the achieved optimum.

Subsequently the same example in [FHL68] has been analyzed using genetic algorithms by [CS96b]. In compare to up that date literature, this approach solved the difficulties in regard to implicit restriction concerning the type of redundancy allowed, the number of available component choices, and whether mixing of components is allowed. However these approaches were only deployable in the case of a system has independent components and the system has n stages in series. Additionally, each stage has several identical components in parallel to provide the redundancy. Up to that date, the traditional reliability redundancy allocation problem focused on the active redundancy.

There are some other efforts to extend the conventional reliability optimization in order to cold redundancy and k out of n system [CL00], [Coi01]. The choice of redundancy strategy (active, cold) has been included as an additional decision variable like reliability, cost, and weight [Coi03]. There are other contributions to consider the effect of imperfect fault coverage due to switching implementation, error handling, and common cause failures issue [ADM99], [APD04].

Nevertheless there exist different challenges about the level of uncertainty due to reliability models and accuracy of reliability characteristics. In order to address and solve this problem, an optimization method has been developed which includes maximizing system reliability estimation and minimizing the variance simultaneously [CJW04].

All above reliability design problems have been basically formulated to maximize or minimize a single objective function (mostly system reliability or time to failure) under performance limitation and geometrical resource constraints. In contrast to above approaches, there also exist applications which a single objective function under consideration of linear and non-linear constraints could not picture the reality on the ground. Due to this fact, the multi-objective reliability attracted many reliability experts. There are hundreds of publications on the topic reliability multi-objective optimization. The most important survey and tutorial for reliability multi-objective are explained in [Deb01] and [KCS06]. Majority of these applications fall into discrete optimization problems. Following table gives an overview about most important contribution in the reliability optimization area.

Literature	Objective functions(s)	Reliability assumption	Solution
[FHL68]	System reliability	Deterministic	Unique solution
[CS96b]	System reliability	Genetic algorithms	Unique solution
[Coi03]	System reliability with the choice of redundancy strategy	Deterministic	Unique solution
[TEC08]	Multi-state system availability	Genetic algorithms	Unique solution and Pareto optimal set
[MS91]	System reliability associated with other system objectives	Deterministic	Pareto optimal set
[CP96]	Lower bound of system MTBF	Genetic algorithm	Unique solution
[Coe08]	System reliability	Differential evolution	Unique solution
[Hwa79]	System reliability	Generalized Lagrangian-function and reduced-gradient methods	Unique solution
[Kuo87]	System reliability	Lagrange-multiplier and branch-and-bound technique	Unique solution
[CS96a]	System reliability	Adaptive penalty method	Unique solution
[RMCK04]	System reliability	Max–min approach	Unique solution

Table 3.2.: Literature review of reliability optimization methods introduced partially by [CJW04]

4. Developed approach for reliability optimization of complex systems

Basically the complex mechatronic systems contain largely available off the shelf components with known properties (like failure rates, cost, weight, etc.). As findings from reliability literature review [Coi03] [CJW04] [FHL68] [CL00] suggest one of main challenges during mechatronic system development is to choose the hardware elements and determine the redundancy level in order to meet maximum reliability level under consideration of technical constraints at the affordable cost level.

Due to this fact, many design alternatives are supposed to be taken into consideration during the design phase. For example, a series-parallel system with 8 subsystems and 10 element choices has been analyzed by [Che08]. There exists more the 10^{30} solutions to consider. This dissertation proposes a new approach which deals with following topics:

- development of a calculating system reliability approach using a system reliability matrix (SRM),
- preparation of a suitable problem formulation,
- demonstrating of conflict among reliability value of a system, total cost of a system, and other technical constraints such size and weight, and
- constructing an optimization process which simultaneously modify the system topology (level of redundancy) and determine component properties using numerical methods during the design phase.

The steps taken to address an optimal structure with optimal components are shown as shown in the figure 4.1. In the first step, the initial hardware structure is transformed into a SRM. The main assignment of this matrix is to calculate reliability automatically once the reliability characteristic of the given topology gets manipulated during optimization process.

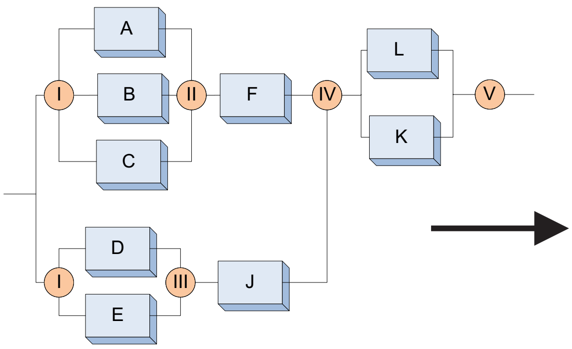
In the next step, an objective function is designed as a measure describing the performance of considered hardware topology in an objective way with respect to the given weighting factors. Numerical search methods (deterministic and genetic algorithms) are employed in order to find best component properties under given component constraints.

In the end, the structure will be varied in an appropriate way, in order to improve the reliability level of overall system under consideration of restricted budget, maximum size, maximum weight, or etc. The above procedure should be iterated as often as the desired topologies with feasible components are detected.

Figure 4.1.: Detailed steps in the proposed optimization concept based on introduced approach in [KJS10]

4.1. System Reliability Matrix

SRM introduced in [KJS10] involves the construction of a matrix denoting the functional connections of the subsystems of the system. The major objective of establishing such a matrix is the creation of a base for automatic calculation of reliability values in case of arbitrarily defined but given topologies. In order to calculate the reliability value of a topology in each optimization iteration, the application of an approach that lends itself easily to automation seems to be indispensable. The SRM approach is explicated using the example illustrated in figure 4.2. SRM is constructed by different nodes and elements (as hardware components). The heading row of the



	A	B	C	D	E	J	F	L	K
I	1	1	1	1	1				
II	2	2	2				1		
III				2	2	1			
IV						2	2	1	1
V								2	2

Figure 4.2.: Reliability block diagram and its SRM - Example

matrix in the figure 4.2 denotes the considered hardware components. The first column demonstrates the nodes of the considered system. Basically the number of components between two nodes represents the level of redundancy. Position of each hardware component can be recognized by the numbers 1 and 2. Starting point and ending point of each component are given by 1 and 2 respectively. For instance, component A is connecting nodes I and II. Parallel hardware components can be recognized using observed same starting- and ending nodes as illustrated in the figure 4.2.

For example, components A, B, and C share same starting and ending nodes (I, II). Also components D and E have same starting and ending nodes. Ultimately the components L and K show the same configuration. In order to calculate the reliability, the developed algorithm begins with the detection of parallel components and putting all components in parallel together. Subsequently the algorithm merges the parallel components and calculate reliability of merged components by using formula 2.8. In the following step, the algorithm continues with detection of the components in series and putting the components in series together. Afterward the algorithm creates one component out of all components in series and calculates the reliability of new created component by using formula 2.7.

These two steps iterate as long as just one component and two nodes exist as shown in the figure 4.3. This approach also is applicable for bridge structure. Indeed the assumption of this dissertation is that the bridge structure have been transformed into a series and parallel structure using delta star method before optimization process starts. This step has been integrated during matrix reading in the developed tool. In order to identify the components in parallel in the SRM, algorithm explores

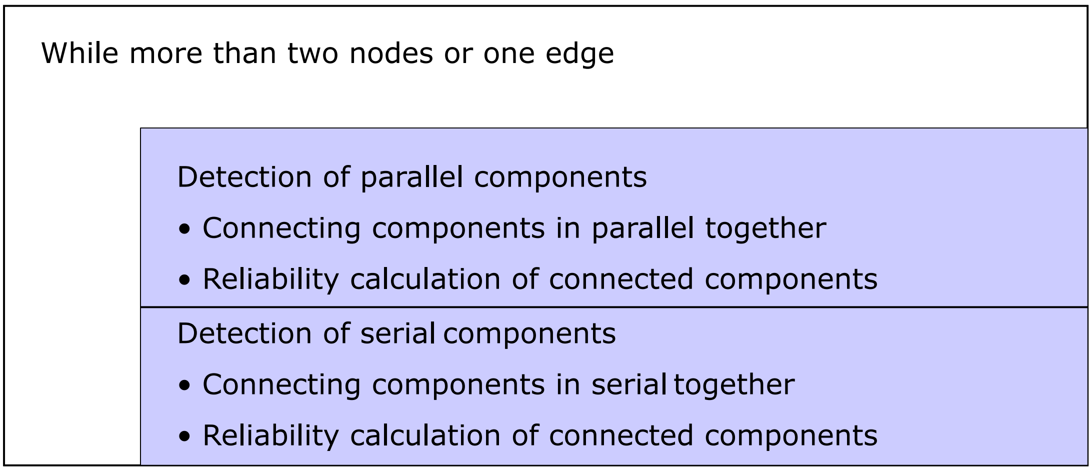
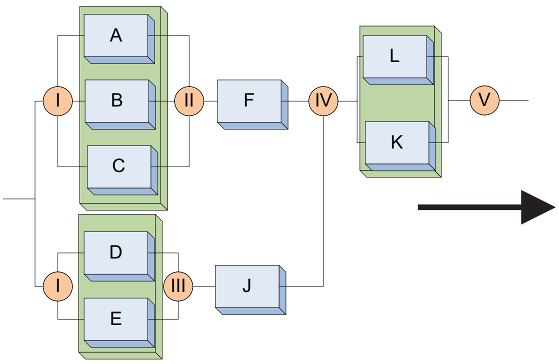


Figure 4.3.: Description of SRM approach by using Nassi Shneiderman diagram based on [KJS10]

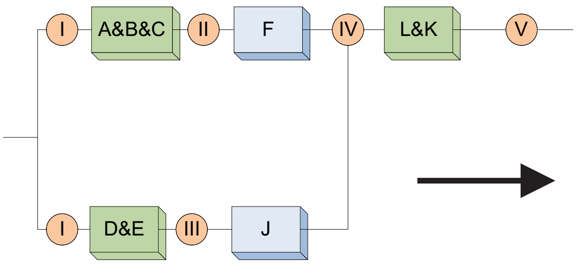
through the columns and searches for identical columns as illustrated in the figure 4.4. After finding the parallel components, the components will be combined and the



	A	B	C	D	E	J	F	L	K
I	1	1	1	1	1				
II	2	2	2				1		
III				2	2	1			
IV						2	2	1	1
V								2	2

Figure 4.4.: Detection of parallel components – step 1 – based on [KJS10]

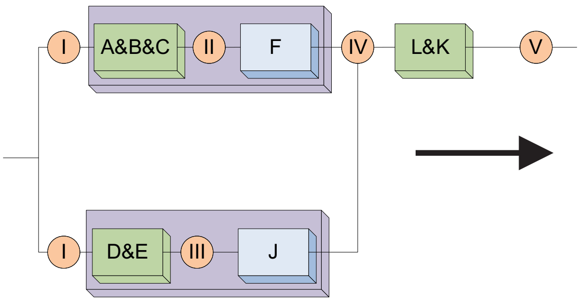
reliability of united components is calculated using formula 2.8 as shown in the figure 4.5.



	A&B&C	D&E	J	F	L&K
I	1	1			
II	2			1	
III		2	1		
IV			2	2	1
V					2

Figure 4.5.: Combining of parallel components – step 2 – based on [KJS10]

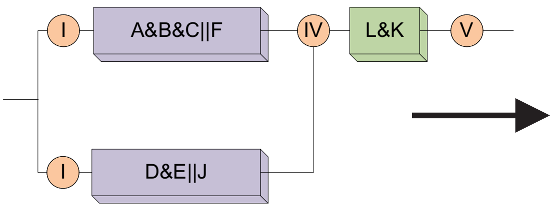
After detecting all parallel components, hardware components in series can be recognizable in the SRM if a hardware component labeled 1 follows a component labeled 2 on a same row as illustrated in the figure 4.6. It should be noted if two different



	A&B&C	D&E	J	F	L&K
I	1	1			
II	2			1	
III		2	1		
IV			2	2	1
V					2

Figure 4.6.: Detection of components in series – step 3 — based on [KJS10]

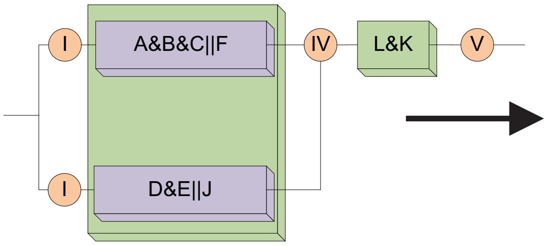
2s follow a 1 like node IV, this rule does not come into effect (this case is occurred in the forth row as illustrated figure 4.6 . After detection the components in series



	A&B&C||F	D&E||J	L&K
I	1	1	
IV	2	2	1
V			2

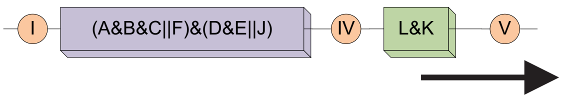
Figure 4.7.: Combining of components in series – step 4 – based on [KJS10]

using above rule, the components will be combined and the reliability of united components is calculated using formula 2.7 as shown in the figure 4.7. In the next steps, the leftover components will be checked if they are in parallel. In the case of parallelism, the components will be combined and the reliability of parallel component can be calculated by formula as addressed in the figure 4.8 and figure 4.9. In the next steps, the algorithm continues to detect the components in series and combine them as shown in figure 4.10 and the figure 4.11. These two steps iterate as long as just one component and two stations are left over.



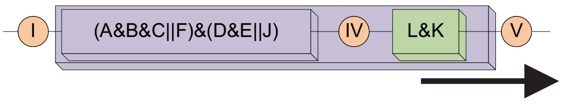
	A&B&C||F	D&E||J	L&K
I	1	1	
IV	2	2	1
V			2

Figure 4.8.: Detection of parallel components – step 5 – based on [KJS10]



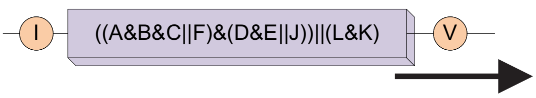
	(A&B&C||F) &(D&E||J)	L&K
I	1	
IV	2	1
V		2

Figure 4.9.: Combining of parallel components – step 6 – based on [KJS10]



	(A&B&C||F) &(D&E||J)	L&K
I	1	
IV	2	1
V		2

Figure 4.10.: Detection of components in series – step 7 – based on [KJS10]



	(A&B&C||F) &(D&E||J) ||(L&K)
I	1
V	2

Figure 4.11.: Combining of components in series – step 8 – based on [KJS10]

4.2. Formulation of the objective function and numerical methods to find best objective function

The component properties like reliability, cost, weight, size, and safety metrics locate generally in the scalar intervals. For example if design team searches for a pressure sensor, a variety of pressure sensors with different accuracy, reliability, weight, size, and etc. can be found on the market.

Typically the choosing the best hardware components concerning reliability and accuracy could result in a system which is not affordable or does not meet size or weight requirements or consume much more electricity as specified at the system design description. Due to large number of components varieties with different properties, application of a design optimization process is essential in order to strike a compromise among different conflicting properties of components.

In this dissertation, component properties are assumed to be moved between continuous scalar ranges. For instance the reliability of pressure sensor can be varied between eighty and nighty percent. The most premium sensor demonstrates 90 percent and the most low-priced sensor provides 80 percent reliability in a given life product interval. The reliability of all other sensors are placed in between of these two limits. Other components properties exhibit a similar intervals which can be explored within a optimization process at the component level in order to figure out the feasible and optimal components properties.

For the optimization process, one of the most important steps is to define an adequate objective function and constraints functions which represent the reality of conflicting behavior among different optimization variables. As illustrated in the chapter of literature review, there exist different approaches for objective function formulation.

The most practical approach basically involves specifying the relative importance values for each objective and aggregating the objectives into an overall objective function. The classical approach to define an optimization problem is to assign a weight w_i to each normalized sub-objective function f_i so that the problem is converted to a single objective problem as expressed as follows:

$$\begin{aligned} &\text{min. } J = \sum_{i=1}^{n} w_i f_i(x) = w_1 f_1(x) + \cdots + w_n f_n(x), \\ &\sum_{i=1}^{n} w_i = 1. \end{aligned} \tag{4.1}$$

During the optimization process at component properties level, there are two topics which affect the optimization results rigorously:

– problem formulation strategy which can be partitioned in the following subjects:

- scaling of objective functions f_i and assigning weighting factors w_i,
- demonstration the conflict between reliability and other technical and financial parameters,
- determining internal relation between optimization parameter like cost and reliability and effect of constraint types on optimization outcome,

– the impact of the type of optimization methods and type of constraints on the optimization results.

In the next sections, the both topics are subject to investigation using some case studies.

4.2.1. Investigation of problem formulation strategy

One of the pivotal issues during reliability optimization is associated with showing the conflict between reliability and other technical and financial parameters, scaling strategy, and ascertaining of weighting factors. In order to address the mentioned challenges, two examples are studied.

Example 1
During the optimization process at component properties level, reliability improvement of components frequently results in violating the vital system characteristics. In order to make the case, following example is utilized to:

– demonstrate the influence of reliability improvement on other technical and financial parameters,

– reveal a scaling procedure, and

– study of weighting factors values on optimization outcome.

The RBD of given example is illustrated in the figure 4.12. The properties of the components A and D are subject to investigation. It is assumed that the properties of other components are fixed.

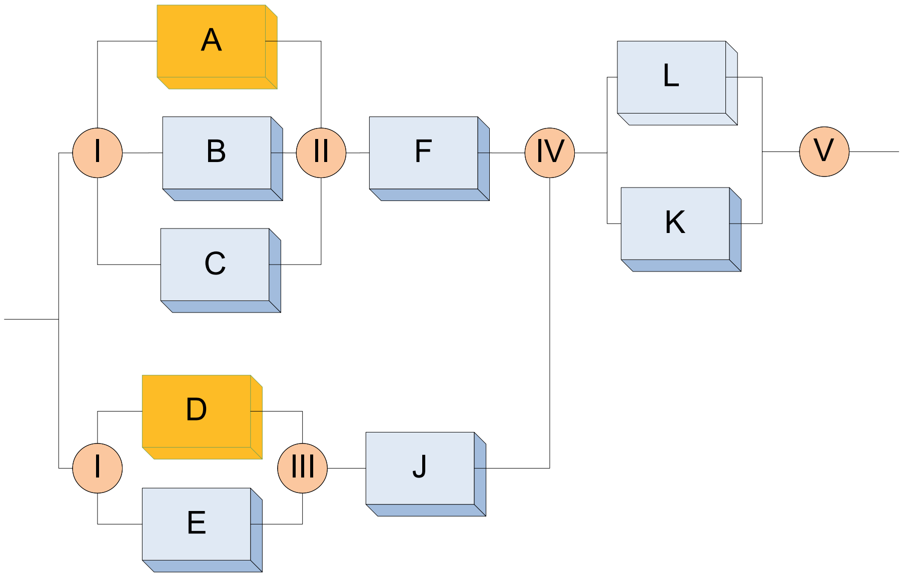


Figure 4.12.: RBD example with two variables

The reliability optimization problem under consideration of various parameters can be expressed as follows:

$$\begin{aligned}
J =& \beta \cdot (\sum_{i}^{n} g(C_i(\lambda_i)) + f_1(\varepsilon_i))+ \\
& \gamma \cdot (\sum_{i}^{n} d(S_i(\lambda_i)) + f_2(\varepsilon_i))+ \\
& \alpha \cdot (\sum_{i}^{n} e(W_i(\lambda_i)) + f_3(\varepsilon_i)), \qquad i = 1,\, 2,\, \dots,\, n \\
J \rightarrow & \text{ min.} \\
n: & \qquad \text{number of components} \\
\alpha + & \beta + \gamma = 1 \\
\alpha,\ & \beta\text{, and } \gamma: \qquad \text{weighting factors} \\
\text{subject to } & \prod_{i=1}^{n} (R_i + \varepsilon_i) \geq R_{\text{required}} \quad \text{and} \\
R_{\min} <& R_i + \varepsilon_i \leq R_{\max} < 1
\end{aligned} \tag{4.2}$$

where J and ε_i represent the objective function and the reliability improvement in the each components respectively. Cost, weight, and size of each components are denoted as follows: $C_i(\lambda_i)$, $W_i(\lambda_i)$, $S_i(\lambda_i)$. All three functions are phrased in association with reliability parameters. Besides the weighting factors β, γ, α are mapped to cost, weight, and size sub-functions in order to display the importance of each subfunction. In order to normalize each subfunction, following scaled functions are introduced:

$$\begin{cases} f(R_i(\lambda_i)) = \frac{R_i(\lambda_i)-R_{\min}(\lambda_i)}{R_{\max}(\lambda_i)-R_{\min}(\lambda_i)} \\ g(C_i(\lambda_i)) = \frac{C_i(\lambda_i)-C_{\min}(\lambda_i)}{C_{\max}(\lambda_i)-C_{\min}(\lambda_i)} \\ d(S_i(\lambda_i)) = \frac{S_i(\lambda_i)-S_{\min}(\lambda_i)}{S_{\max}(\lambda_i)-S_{\min}(\lambda_i)} \\ e(W_i(\lambda_i)) = \frac{W_i(\lambda_i)-W_{\min}(\lambda_i)}{W_{\max}(\lambda_i)-W_{\min}(\lambda_i)}. \end{cases} \tag{4.3}$$

where the scaling function of each parameter (reliability, cost, size and weight) are given as follows: $f(R_i(\lambda_i))$ $g(C_i(\lambda_i))$ $d(S_i(\lambda_i))$ $e(W_i(\lambda_i))$. The scaling factors are defined in way which all subfunction are located in a fixed limited range namely $[0\,;1]$.

The reliability improvement rate impacts the cost, size, weight consultation of each system as illustrated in the figure 4.13. This depiction shows many local minima for a system with two variables. Accordingly to this fact, a manually making decision about an optimal and feasible consultation for a complex system with high number of variables is either impossible or it must be put a lot of effort due to different local and global optima. This fact delivers a strong argument why a numerical optimization method is necessary to apply for component properties determination.

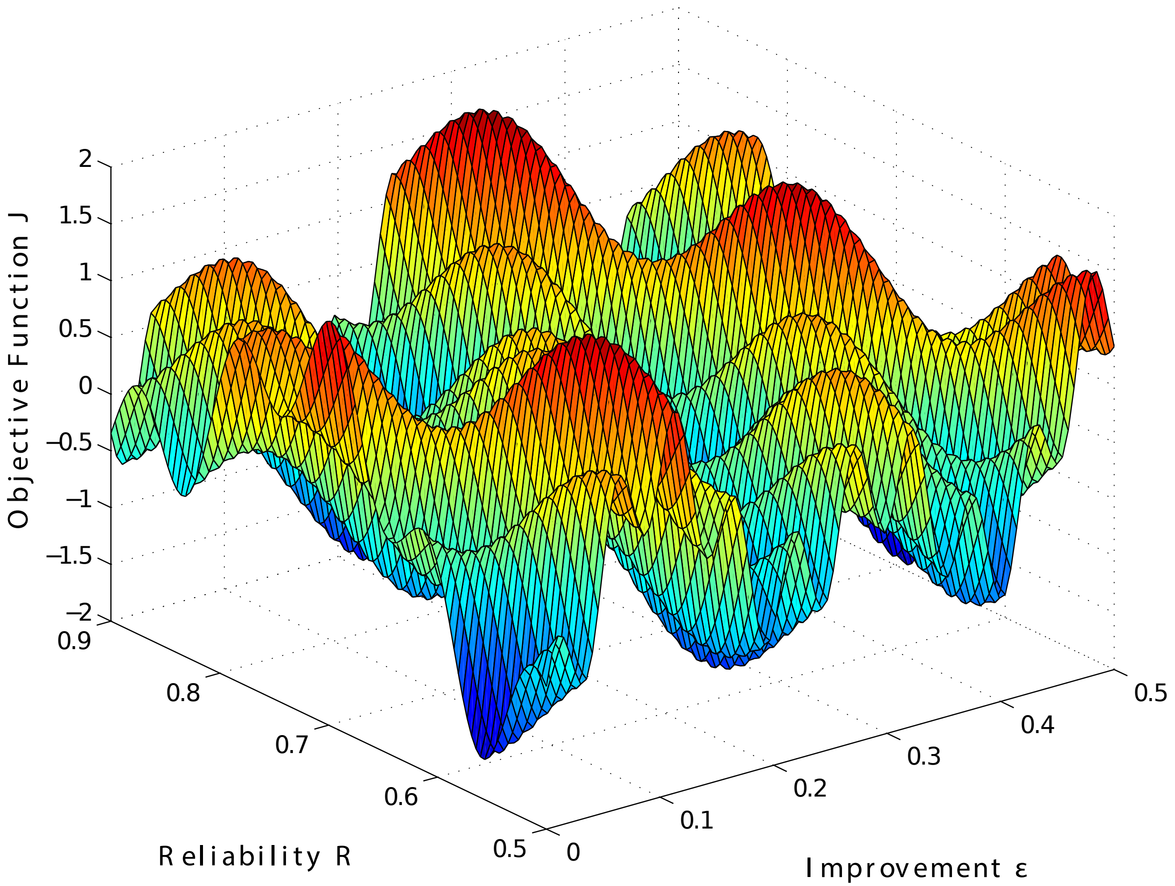


Figure 4.13.: Impact of reliability improvement on other system parameters

Example 2

The second example is related to same structure illustrated in the figure 4.12. This example is aimed to show:

– the impact of weighting factors values on optimization outcome and

– study of cost and failure rate association.

The general objective function for defined example is expressed mathematically as follows:

$$\begin{aligned}
&J = \alpha \cdot \frac{1}{(1 - f(R_i(\lambda_i)))} + \beta \cdot \sum_i^n g(C_i(\lambda_i)) \\
&J \rightarrow \text{min.} \\
&n: \qquad\qquad \text{number of components} \\
&\alpha \,\text{and}\, \beta: \qquad \text{weighting factors} \\
&\begin{cases}
R \;\geq\; R_{\text{required}} \\
7 \cdot 10^{-7} \;\leq\; \lambda_A \;\leq\; 10^{-6} \\
6.5 \cdot 10^{-7} \;\leq\; \lambda_D \;\leq\; 10^{-6} \\
f(R_i(\lambda_i)) = \frac{R_i(\lambda_i) - R_{\min}(\lambda_i)}{R_{\max}(\lambda_i) - R_{\min}(\lambda_i)} \\
g(C_i(\lambda_i)) = \frac{C_i(\lambda_i) - C_{\min}(\lambda_i)}{C_{\max}(\lambda_i) - C_{\min}(\lambda_i)}.
\end{cases}
\end{aligned} \tag{4.4}$$

It is subject to the condition that exponential time-to-failure reflects the behavior of the component appropriately, so the failure rates are assumed to be constant. The objective function is modeled as the sum of two sub functions with individual weighting. The first sub function deals with the system reliability $\frac{1}{(1-f(R_i(\lambda_i)))}$. The second part involves cost objective in association with component failure rates $g(C_i(\lambda_i))$ as described in the past sub section. The above problem formulation can be readily extended to accommodate other objectives. The design of the objective function and the choice of the corresponding weighting factors are necessary steps within the numerical optimization process. The weighting factors α_i and β can be determined using relation between reliability and cost at the system level. The design team shall determine which parameter is more relevant to design. As discussed in chapter 2, it is recommended to define all sub objective functions (cost, size, weight, etc.) in association with failure rates like reliability sub function. Reducing the number of optimization parameters is the notion behind of step. The coefficients of failure rate and cost functions are determined by least squares polynomial fitting method.

The relation between cost and reliability for component A and D is given by

$$C_{\text{A}}(\lambda_i) = \sigma \cdot \lambda_i^4 + \Gamma \lambda_i^2 + \mu \cdot \lambda_i^3 \tag{4.5}$$

$$C_{\text{D}}(\lambda_i) = \sigma \cdot \lambda_i^3 + \Gamma \lambda_i. \tag{4.6}$$

In order to show the impact of weighting factors on the optimal component characteristics, the behavior of objective function is depicted by two different weighting factors. The first weighting factors are phrased as follows:

$$\alpha = \frac{f(R_i(\lambda_i)}{g(C_i(\lambda_i))} \tag{4.7}$$

$$\beta = 1 - \frac{f(R_i(\lambda_i)}{g(C_i(\lambda_i))}. \tag{4.8}$$

The second weighting factors are expressed as follows:

$$\alpha = \frac{1 - \frac{f(R_i(\lambda_i)}{g(C_i(\lambda_i))}}{1 + \frac{f(R_i(\lambda_i)}{g(C_i(\lambda_i))}}. \tag{4.9}$$

$$\beta = 1 - \frac{1 - \frac{f(R_i(\lambda_i)}{g(C_i(\lambda_i))}}{1 + \frac{f(R_i(\lambda_i)}{g(C_i(\lambda_i))}}. \tag{4.10}$$

The objective function behavior in case of using first weighting factors is illustrated in the figure 4.15. While the objective function behavior by using second weighting factors is plotted in the figure 4.14. The difference between figure 4.14 and figure 4.15 give credence to the notion, that the effect of the weighting factors on the objective functions measure is formidable. Determination of weighting factors obviously takes influence on optimum of objective functions [Mut12]. Same effort to show the impact of weighting factors on reliability optimization is given in [Mut12]. The trend of both figures also represents diverse local and global optima which are not trivial to be explored and found in manual way. Consequently design team feels impelled to use deterministic, stochastic or enumerable methods. The next subsection copes with usage of optimization methods in order to find the globally optimal solution.

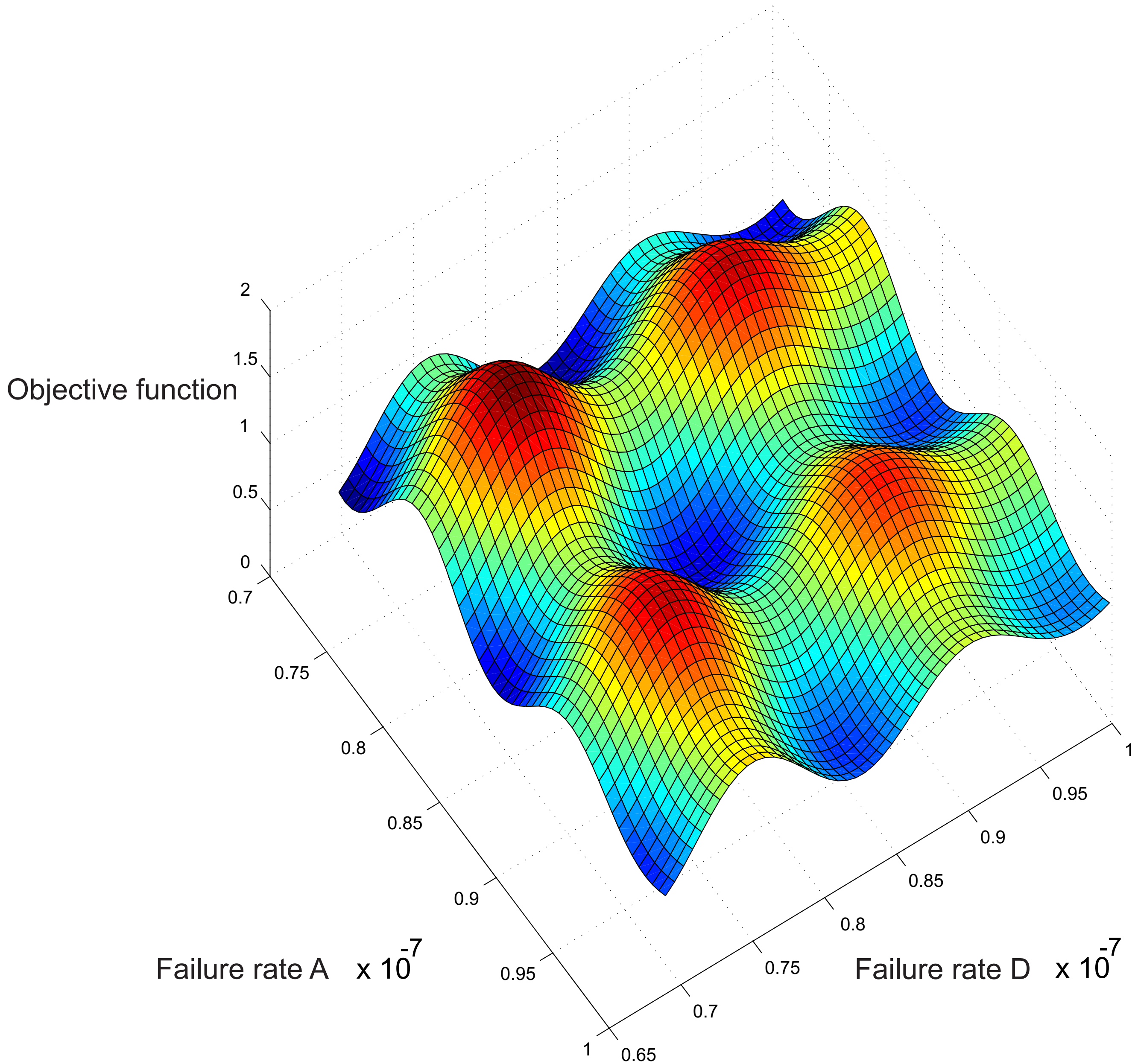


Figure 4.14.: Impact of weighting factors on optimization process based on [KJS10]

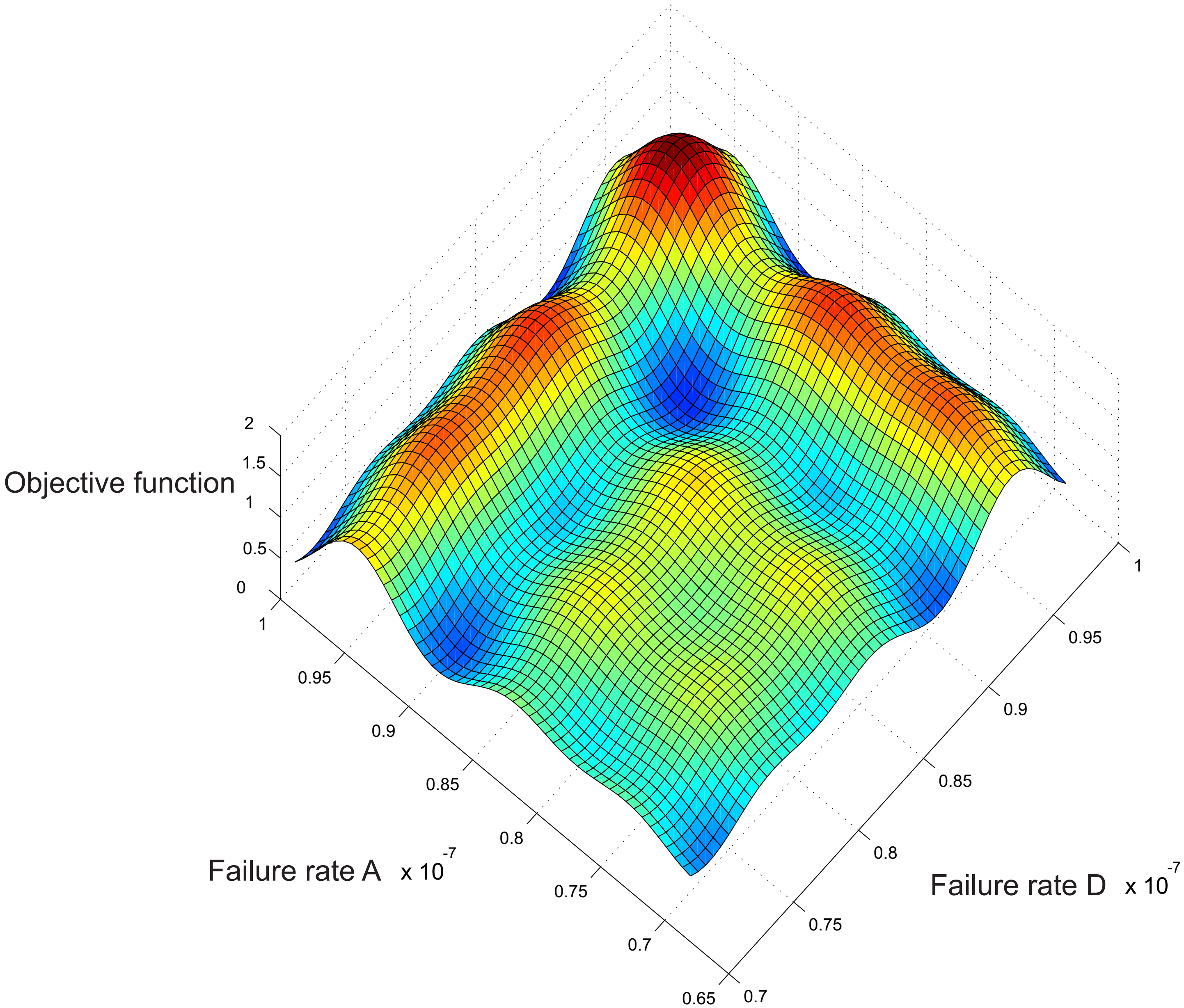


Figure 4.15.: Impact of weighting factors on optimization process based on [KJS10]

4.2.2. Determining of feasible and optimal design at the component properties level

Once the topology of system is fixed and the component properties of each specific component are subject to alter, there exists different methods to solve such problem. As discussed in the previous example, the practical optimization problems have multiple local minima. Therefore, the manual decision-making for finding the optimal solutions is not utilizable in most instances. In this dissertation, the numerical deterministic algorithms and genetic algorithms have been used to find the optimal value of the given objective function at the component level. In order to illustrate the application of numerical approach, two case studies are explored. In the first example, the optimization problem has linear constraints and is solved by using genetic algorithms. The second example includes nonlinear constraints. The second optimization problem is solved by application of a numerical deterministic algorithm.

Example 1

In the given example, the properties of three component are subject to change namely components A, D, L. The objective function expressed as follows:

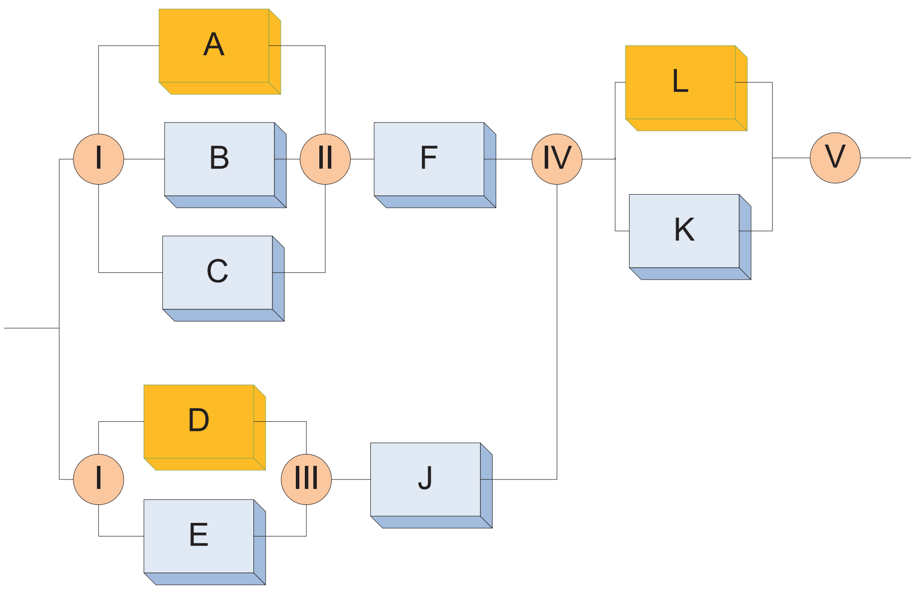


Figure 4.16.: RBD example with three variables

$$
\begin{aligned}
& J = \alpha \cdot \frac{1}{(1 - f(R_i(\lambda_i)))} + \beta \cdot \sum_i^n g(C_i(\lambda_i)) \\
& J \rightarrow \text{min.} \\
& n: \qquad\qquad\qquad \text{number of components} \\
& \alpha \,\text{and}\, \beta: \qquad\qquad \text{weighting factors} \\
& \begin{cases} \sum_i^n C_i \ \leq \ 100, \\ R \ \geq \ 0.8 \quad \text{and} \\ C_i(\lambda_i) = \sigma \cdot \lambda_i^4 + \Gamma \lambda_i^2 + \mu \cdot \lambda_i^3 \end{cases}
\end{aligned}
\tag{4.11}
$$

where the weighting factors are given based on same principle as described in formula 4.7 and 4.8. In this example, there exists linear constraints.

Range	λ_{A}	λ_{L}	λ_{D}
Lower bound	$4.3 \cdot 10^{-5}$	$5 \cdot 10^{-5}$	$4.2 \cdot 10^{-5}$
Upper bound	$9.5 \cdot 10^{-5}$	$9.9 \cdot 10^{-5}$	$8.5 \cdot 10^{-5}$

Table 4.1.: Component failure rate ranges

The failure rate intervals of to be investigated components are illustrated in the table 4.1 . The behavior of objective function in association with failure rate of components A and D is displayed in the figure 4.17. Multiple local minima can be observed in the figure 4.17 also pictures . In order to find the feasible optimum, genetic algorithms have been used to solve this optimization problem. The genetic algorithms deploy the principles of biological evolution, iteratively changing a group of individual points using rules modeled on gene combinations in biological reproduction. Due to its random nature, the genetic algorithm improves the chances of finding a global solution. It does not require the functions to be differentiable or continuous like numerical deterministic algorithms.

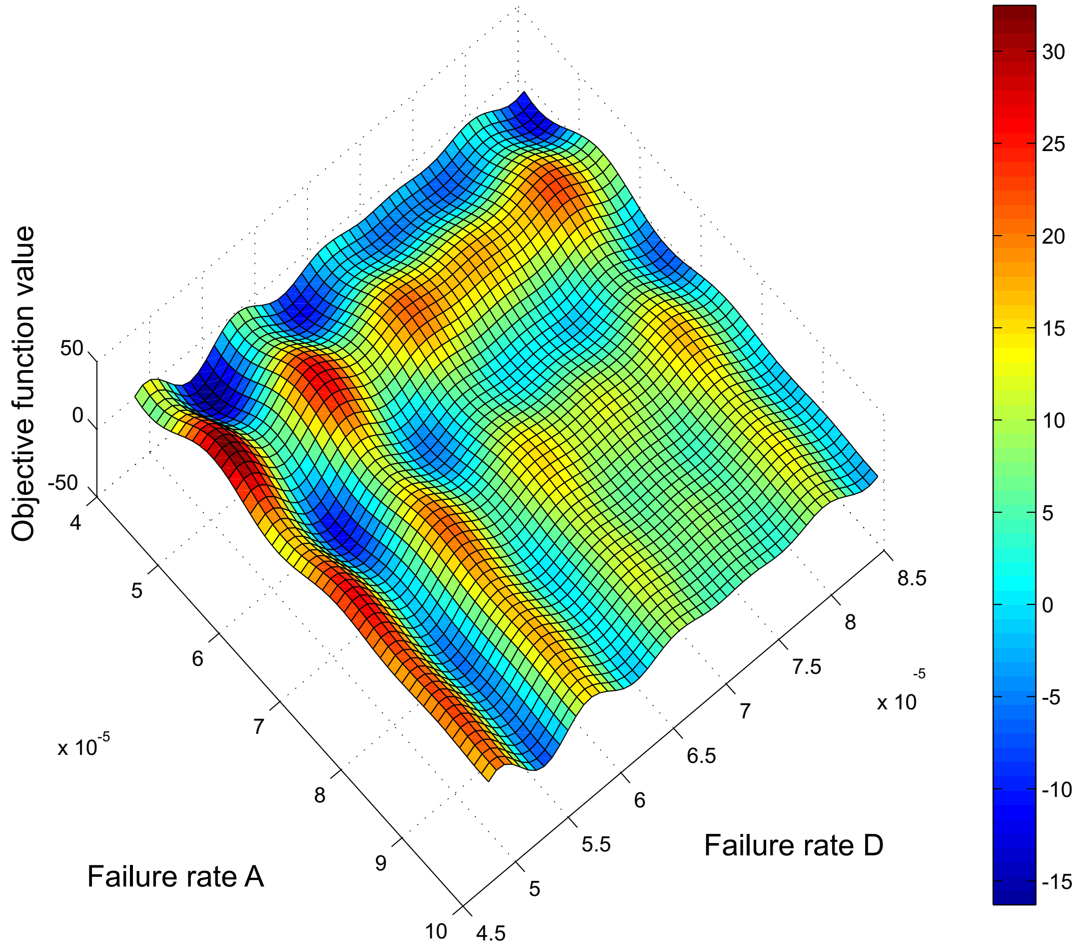


Figure 4.17.: Illustration of failure rates relationship (A and D) on objective function based on [KJS10]

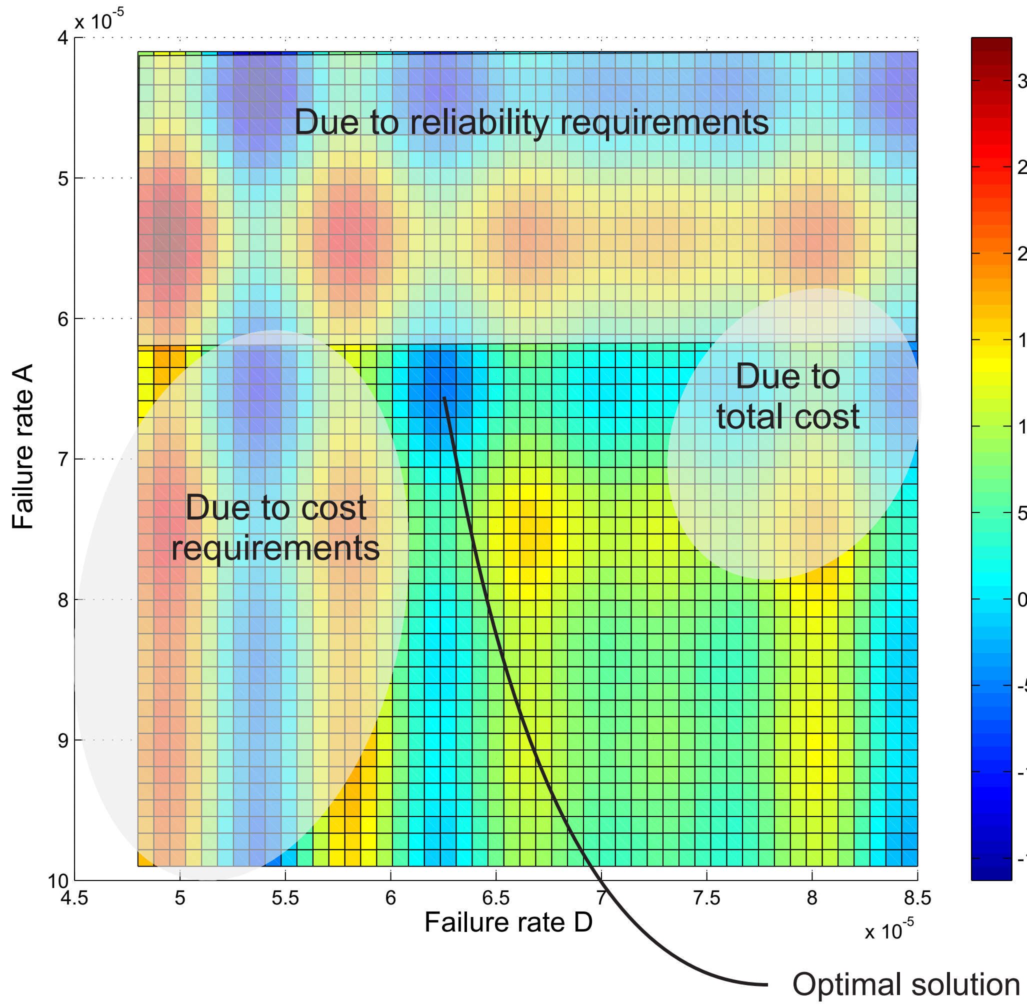


Figure 4.18.: Optimal failure rates under given reliability and cost restrictions based on [KJS10]

The genetic algorithm calculates well approximating solutions to all types of problems (convex and non-convex) because genetic algorithms do not make any assumption about the shape of the objective function.

Failure rate	**Value**
$\lambda_{\mathrm{A,\,optimal}}$	$6.4 \cdot 10^{-5}$
$C_{\mathrm{A,\,optimal}}$	25
$\lambda_{\mathrm{D,\,optimal}}$	$6.3 \cdot 10^{-5}$
$C_{\mathrm{D,\,optimal}}$	42
$\lambda_{\mathrm{L,\,optimal}}$	$8.2 \cdot 10^{-5}$
$C_{\mathrm{L,\,optimal}}$	32

Table 4.2.: Optimal failure rates

The drawback of using genetic algorithms is certainly that the searching operation consumes a lot of time. Nevertheless with new advancement in processing computation area, this challenge can be ignored in many applications. By using genetic algorithms, the feasible search space can bordered easily due to random nature operations. In order to show that the genetic algorithms find the real optimal solution

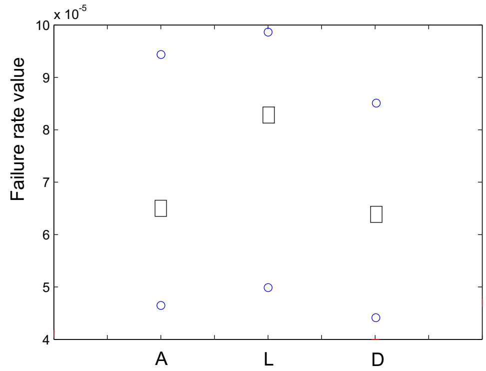


Figure 4.19.: Optimal failure rate of components A, D, and L

in right way, the parts of search space which can not be considered due to cost and reliability constraints during optimization process are highlighted in the figure 4.18. The optimal solution of objective function with given constraints using the genetic algorithms is given in the table 4.2 and in the figure 4.19. It can be observed that the determined optimal solution, as shown in the figure 4.18, is located in the allowed search space .

Example 2

The second example is related to same problem with nonlinear constraints. The optimization problem is represented in the following way:

$$
\begin{aligned}
&J = \alpha \cdot \frac{1}{(1 - f(R_i(\lambda_i)))} + \beta \cdot \sum_i^n g(C_i(\lambda_i)) \\
&J \to \min. \\
&n: \qquad \text{number of components} \\
&\alpha \,\text{and}\, \beta: \qquad \text{weighting factors} \\
&\begin{cases}
\lambda_1 \cdot \lambda_2 \geq 7 \cdot 10^{-12} \\
\lambda - 10^6 \cdot \lambda_1 \cdot \lambda_2 - \lambda_1 \geq 1.5\ 10^{-6} \\
10^{-7} \leq \lambda_1 \leq 10^{-6} \\
10^{-7} \leq \lambda_2 \leq 10^{-6} \\
1.5 \leq C_1 \leq 2.5 \\
1.5 \leq C_2 \leq 2.5.
\end{cases}
\end{aligned}
\tag{4.12}
$$

For solving this example, the numerical deterministic approach has been used by minimizing the scalar function of multiple variables, within a region specified by linear or nonlinear constraints and bounds. The search space of the objective function is demonstrated in the figure 4.20 (top left). After considering the first nonlinear constraint, a part of search space is eliminated as illustrated on top right side of the illustration. The third figure (left down) indicates the search space in the presence of second constraint. Ultimately the last figure (right down) show the reduced search region in the presence of both nonlinear constraints.

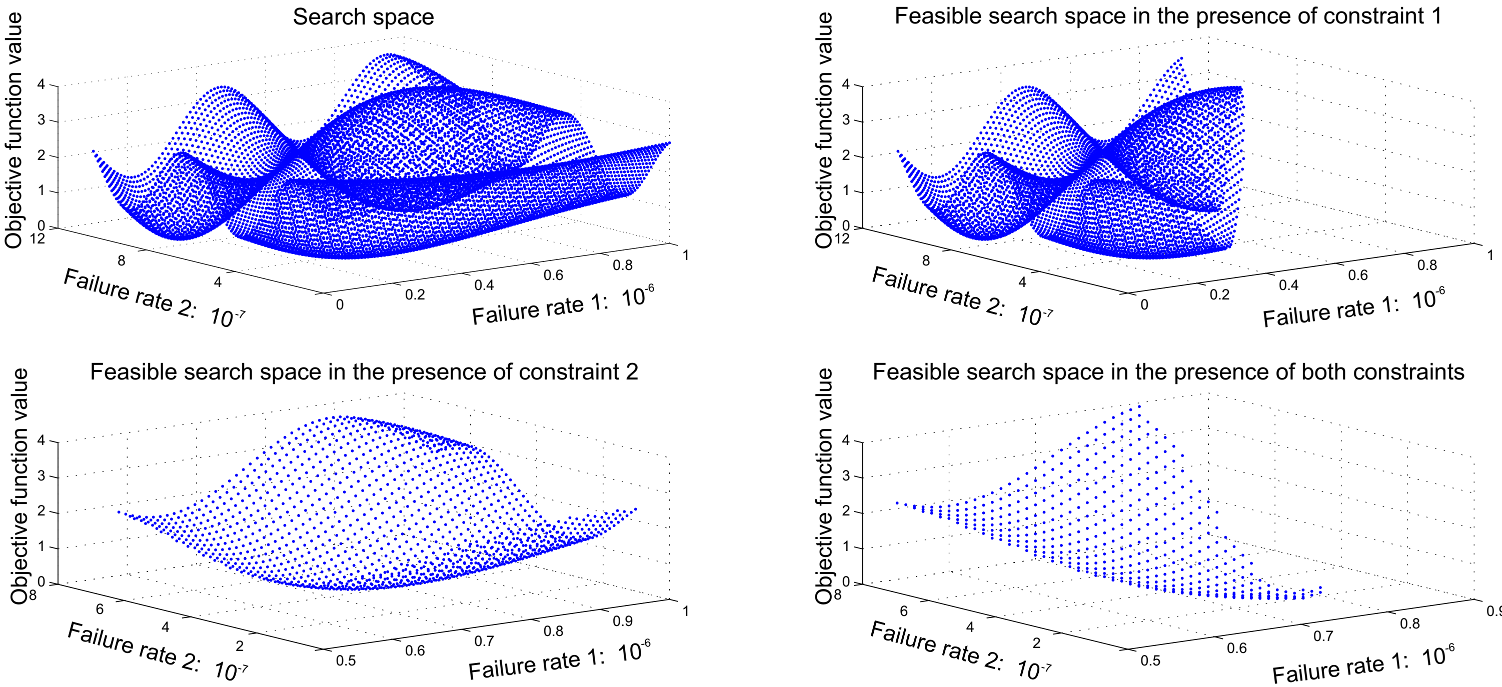


Figure 4.20.: Impact of constraints on search space during optimization based on [KJS10]

To determine the optimal failure rates of these components, the interior-point method [FGW02] is applied. The notion behind of this decision is that this kind algorithm can handle optimization problems with nonlinear constraints very efficiently and does not need initial solution in the feasible region. In many applications, initial design solution is not known to design team. In such cases, it is advisable to apply numerical deterministic methods. The solution out of each iteration can be shown with red circles in the figure 4.21. The blue search space demonstrates the feasible region. After 21 iterations an optimal solution has been achieved.

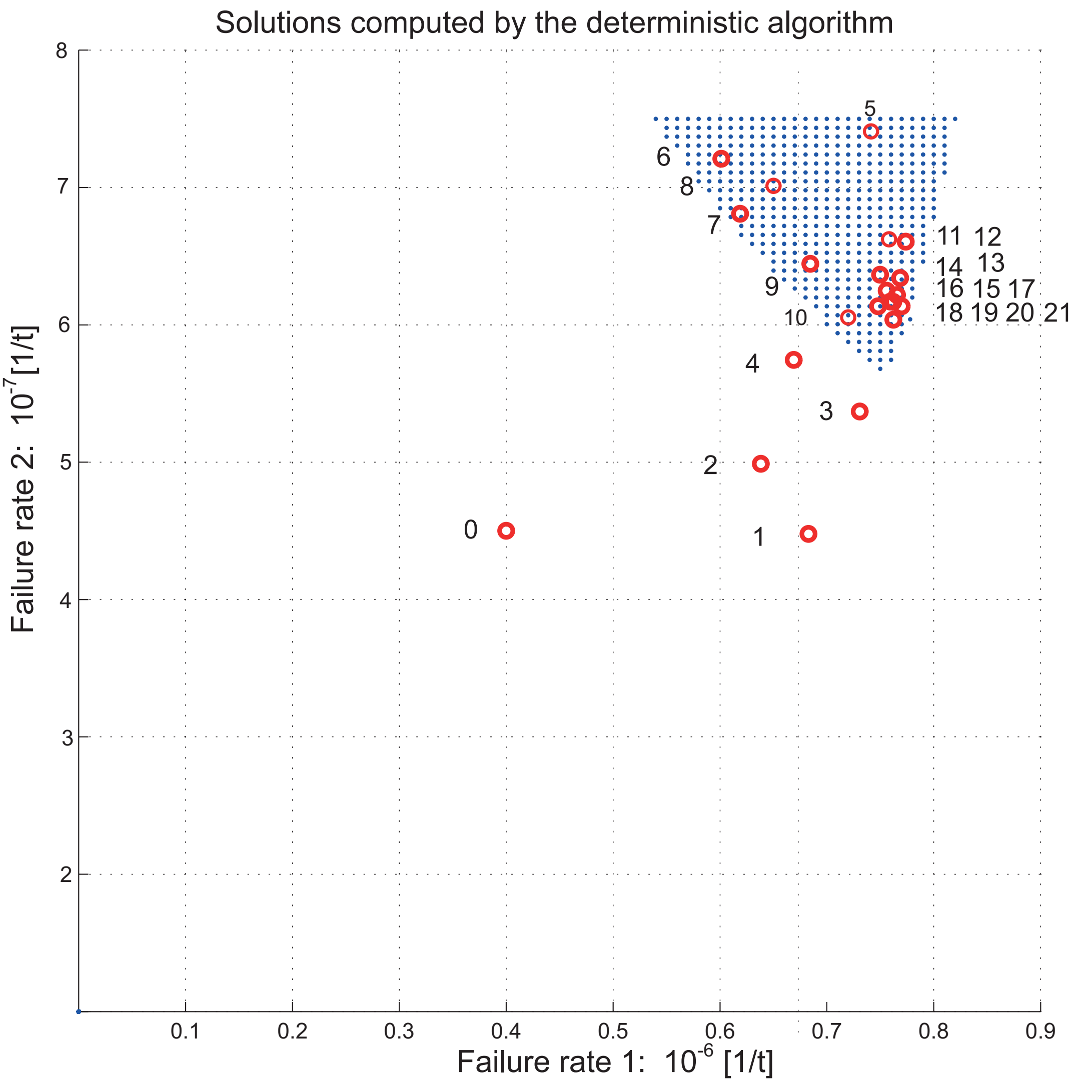


Figure 4.21.: Trend of finding the optimal failure rates based on [KJS10]

4.3. Enumerative strategy for varying system topologies

In the past step, the numerical deterministic method and genetic algorithms are applied to find the optimal failure rates of hardware components. Indeed it is also required to consider that the hardware architecture (topology) is in reality not given and is subject to investigation during early design stage.

In order to choose the best hardware architectures for hardware component optimization, it is mandatory to check all possible variant of system topologies during preliminary design phase. Basically, a real system contains certain number of parts which have diverse off-the-shelf component choices to select at the market. Selection of off-the-shelf components and possible redundancy strategies is a challenging combinatorial optimization.

In the meantime, this challenge is broadened in the vehicle industry as well because mechatronic systems in vehicles are assembled using off-the-shelf components with known characteristics (reliability, cost, size, weight). In this work, the investigation of topology is based on two steps as shown in the figure 4.23:

i) Generation of all possible architectures

ii) Choosing most plausible architecture.

In this part, in order to determine optimal feasible architecture, it is assumed that a database for empirical failure rates of diverse hardware components exists. In this context, the search enumerative algorithm begins to generate all possible architectures using SRM manipulation. Note that it is assumed which an initial topology is existed based on the available hardware components and desired functional requirements. In order to generate the new topologies automatically, the initial SRM can be changed systematically. In this dissertation, it is proposed to manipulate the SRM Iteratively in order to create new topologies and the algorithm subsequently chooses which architectures (topologies) can be optimal and technically feasible. The manipulation of SRM can be realized as follows:

1. elimination of first column of current SRM,
2. adding two or more than two same column at the end current SRM,
3. updating the component properties (initial failure rate, tolerance failure rate range, initial cost, tolerance cost range, and etc.),
4. checking if the global constraints are violated by changing the SRM and updating the component properties
5. in case of violation, coming back to old SRM and trying step i and ii for the next column and other,
6. in case of not violation, providing the current SRM and its component properties to numerical optimization process at component level.

These five steps iterate as long as the latest column of initial SRM are replaced by new columns. The first two important steps are illustrated in the figure 4.22.

	A	B	C	D	E	J	F	L	K
I	1	1	1	1	1				
II	2	2	2				1		
III				2	2	1			
IV						2	2	1	1
V								2	2

D1	D2
1	1
2	2

D1	D2	D3
1	1	1
2	2	2

....

Figure 4.22.: Generating new topology by manipulation of SRM

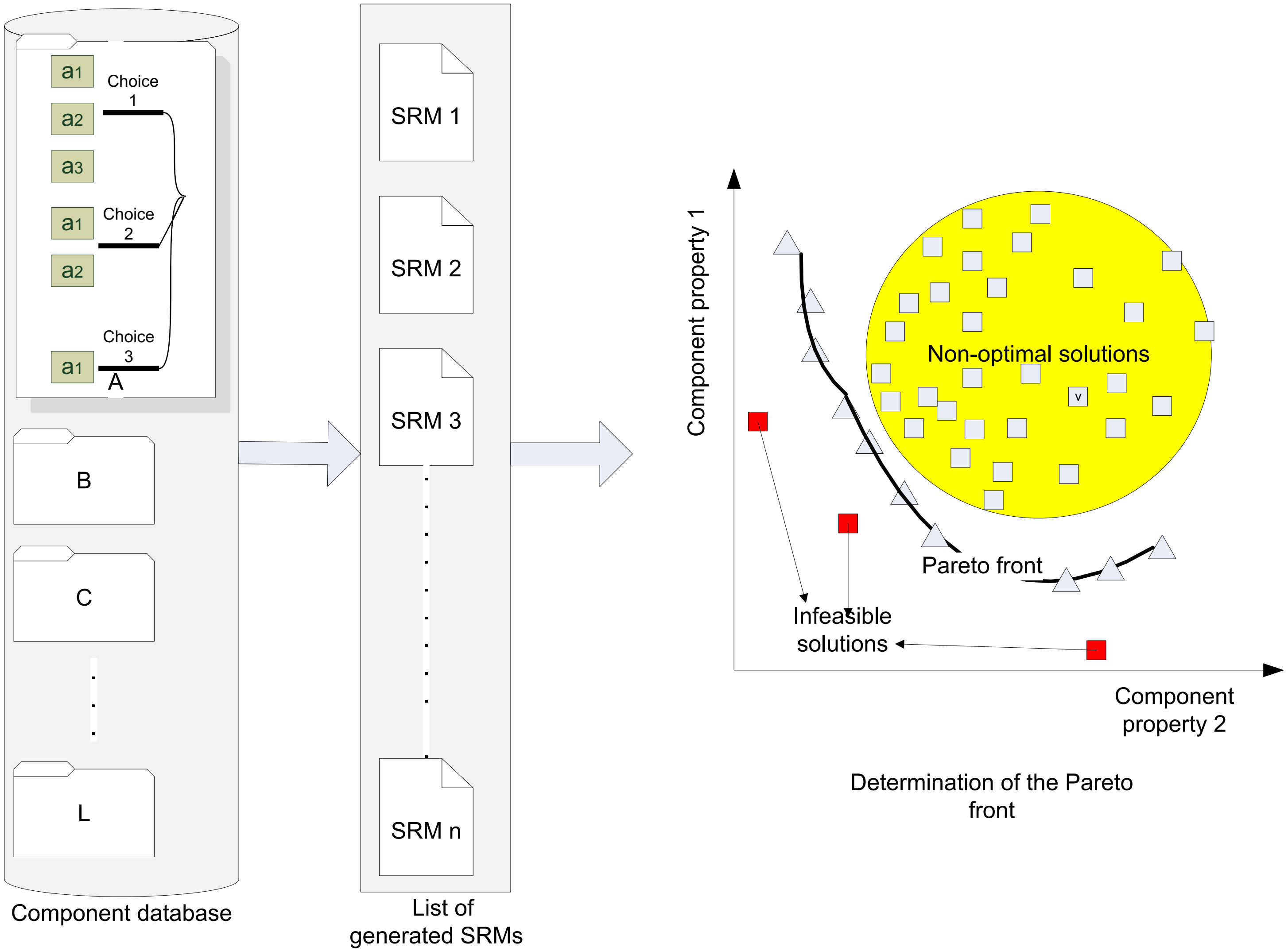


Figure 4.23.: Generation and choosing of new topology

5. Practical applications

As in the past chapters discussed, recent advancements in the automotive industry illustrate a clear trend for integrating an increasing number of safety-related electronic systems such as driver assistance and x-by-wire systems that are replacing traditional mechanical control systems.

Actuating and monitoring of the throttle, brakes, and steering in an electronic way improve safety in the vehicle. However, the elimination of conventional mechanical link between the driver and the vehicle raises some new questions regarding reliability, cost of such systems. Steer-by-wire and brake-by-wire systems are major part of fully integrated vehicle stability control systems which offer an optimal facility for collision avoidance systems and potential design even for autonomous driving. The current challenge for using such systems is the choosing of a fault tolerant electrical architecture (hardware and software) with internal redundancies.

In this chapter, two current innovative applications are investigated. In the first example, analysis of a typical steer-by-wire system is given regarding hardware reliability, safety, and cost. The second application deals with reliability analysis of EMB systems under given constraints like cost, size, and weight. The proposed platform, as shown in the figure 4.1, is deployed to select an optimal feasible topology with optimal component properties.

5.1. Steer-by-wire example

Power steering systems in current vehicle possess a mechanical connection between the vehicle's front wheels and the steering wheel. Once the driving assistance system (no matter if it is realized in an electric or hydraulic way) fails, the mechanical backup is still in place and operational. However, the heavy mechanical link provokes a major drawbacks like noise, vibration, harshness (NVH), and negative safety impact in case of frontal accident [Den04].

In the meantime, there are different concepts which eliminates the mechanical backup between a vehicle's front wheels and its driver. This means that turning the steering wheel does not directly generate steering motion at the front wheels. In case of fully steering by wire system, hand-wheel angle sensor continuously sends a signal to the ECU which controls and instructs the motors to turn the wheels [Yih05]. The figure 5.1 shows the development trend of steering system development in the past years. Simultaneously the designers are analyzing the future of steer-by-wire system in combination of other drive-by-wire systems in the vehicle.

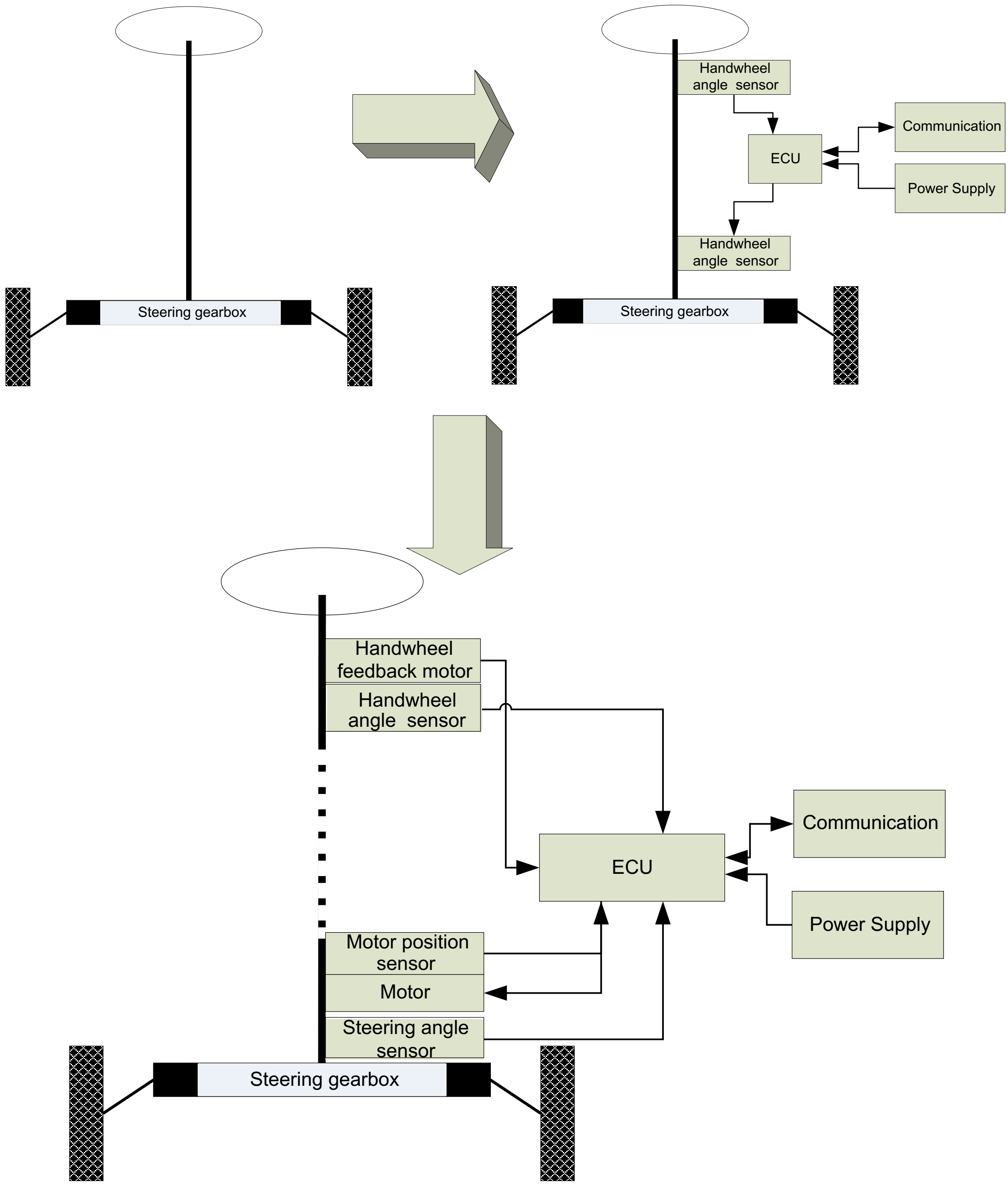


Figure 5.1.: Automotive steering system development in the recent years

The major incentive of steer-by-wire technology is to replace the mechanical components (steering shaft, column, gear reduction mechanism, etc.) with a mechatronic system (couple of sensor,. Fully replacing conventional steering system with steer-by-wire system offers various benefits, namely [Yih05]:

- eliminating of steering column facilitates the car interior design.
- eliminating of steering shaft, column and gear reduction mechanism provides much better space available in motor compartment and reduction of injuries in accidents
- reducing the total weight of the car which leads to lower energy consumption [Yih05].

Representative number	**Component**
1	Handwheel angle sensor
2	ECU
3	Communication
4	Motor
5	Steering angle sensor
6	Motor position sensor
7	Handwheel feedback motor
8	Power supply

Table 5.1.: Representative numbers of the components in the given example

In the figure 5.2, a simple RBD for a typical steer-by-wire system is introduced. While the table 5.1 demonstrates the minimal components which are required to realize a steer-by-wire system. This topology and its given components are acting as an initial topology and initial components during using of introduced reliability optimization platform.

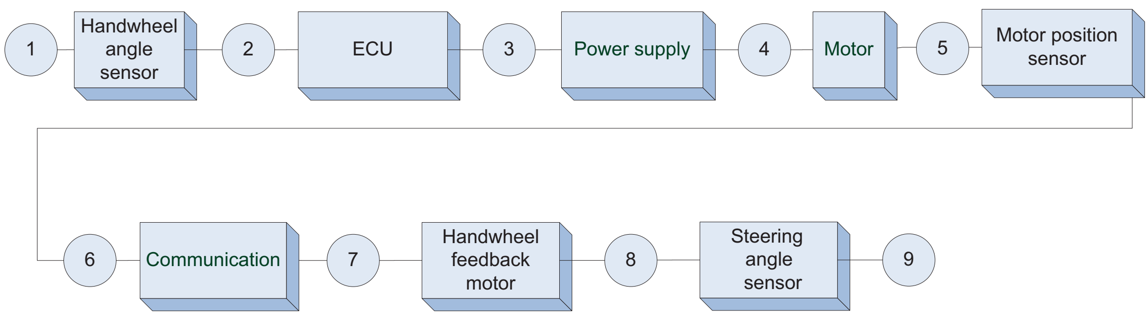


Figure 5.2.: Typical RBD for a steer-by-wire example

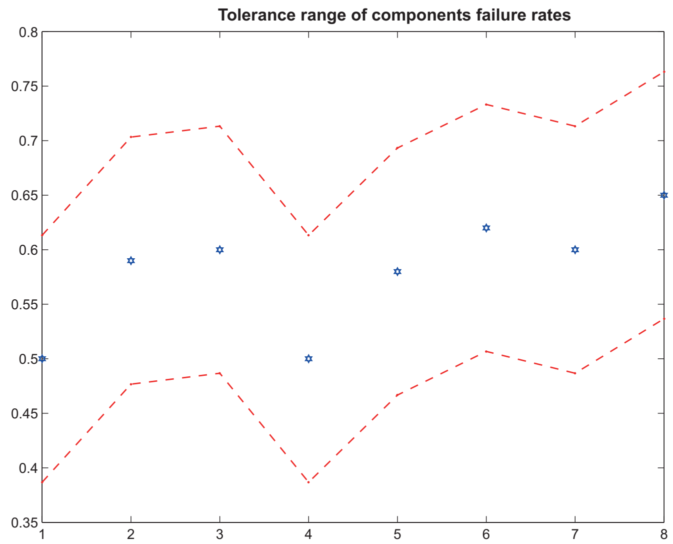


Figure 5.3.: Initial components failure rates in 10^{-6} [1/h]

Initial failure rates and failure rate interval of to be analyzed hardware components for a fictive steer-by-wire system are plotted in the figure 5.3.

The optimization problem is expressed by

$$\begin{aligned} & J = \alpha \cdot \frac{1}{(1 - f(R_i(\lambda_i)))} + \beta \cdot \sum_i^n g(C_i(\lambda_i)) \\ & J \rightarrow \min. \\ & n: \qquad \text{number of components} \\ & \alpha \text{ and } \beta: \qquad \text{weighting factors} \\ & \begin{cases} \sum_i^n C_i \leq C_{\max} = 400, \\ R \geq 0.8 \\ f(R_i(\lambda_i)) = \frac{R_i(\lambda_i) - R_{\min}(\lambda_i)}{R_{\max}(\lambda_i) - R_{\min}(\lambda_i)} \\ g(C_i(\lambda_i)) = \frac{C_i(\lambda_i) - C_{\min}(\lambda_i)}{C_{\max}(\lambda_i) - C_{\min}(\lambda_i)} \end{cases} \end{aligned} \tag{5.1}$$

where $R_i(\lambda_i)$ is calculated using SRM methods as soon as topology and type of components are changing. The following weighting factors are applied for this example

$$\alpha = \frac{1 - \frac{f(R_i(\lambda_i))}{g(C_i(\lambda_i))}}{1 + \frac{f(R_i(\lambda_i))}{g(C_i(\lambda_i))}}, \tag{5.2}$$

$$\beta = 1 - \frac{1 - \frac{f(R_i(\lambda_i))}{g(C_i(\lambda_i))}}{1 + \frac{f(R_i(\lambda_i))}{g(C_i(\lambda_i))}}. \tag{5.3}$$

Components	Choice 1 $[10^{-6}]$	Choice 2 $[10^{-6}]$	Choice 3 $[10^{-6}]$	Choice 4 $[10^{-6}]$	Choice 5 $[10^{-6}]$
Handwheel angle sensor	[0.5 ; 0.7]	[0.3 ; 0.5]	[0.45 ; 0.6]	[0.2 ; 0.4]	[0.25 ; 0.35]
ECU	[0.3 ; 0.6]	[0.5 ; 0.8]	[0.4 ; 0.75]	[0.2 ; 0.35]	[0.6 ; 0.9]
Communication	[0.5 ; 0.75]	[0.75 ; 0.98]	[0.35 ; 0.69]	[0.45 ; 0.7]	[0.2 ; 0.4]
Motor	[0.45 ; 0.7]	[0.2 ; 0.5]	[0.3 ; 0.6]	[0.5 ; 0.9]	[0.35 ; 0.7]
Steering angle sensor	[0.45 ; 0.75]	[0.5 ; 0.7]	[0.25 ; 0.5]	[0.35 ; 0.65]	[0.5 ; 0.8]
Motor position sensor	[0.5 ; 0.9]	[0.5 ; 0.9]	[0.5 ; 0.9]	[0.5 ; 0.9]	[0.5 ; 0.9]
Handwheel feedback motor	[0.3 ; 0.6]	[0.4 ; 0.8]	[0.5 ; 0.9]	[0.35 ; 0.65]	[0.2 ; 0.5]
Power supply	[0.1 ; 0.3]	[0.2 ; 0.5]	[0.3 ; 0.6]	[0.25 ; 0.55]	[0.15 ; 0.4]

Table 5.2.: Failure rate interval of various choices

The table 5.2 illustrates the feasible intervals of the each different subsystem choices. Under consideration of cost limit (400 units) and minimal reliability requirement constraints, the assignment of developed reliability optimization platform is to find the optimal topology (redundancy level) and search for best component failure rates in each optimal topology. The cost and failure rate relation of all components is expressed by

$$\begin{aligned} C_(\lambda_i) =& 8032 \cdot \lambda_i^7 - 4215 \cdot \lambda_i^6 + 65000 \cdot \lambda_i^5 - 48739 \cdot \lambda_i^4 + 17363 \cdot \lambda_i^3 \\ & - 1500 \cdot \lambda_i^2 + 30 \cdot \lambda_i. \end{aligned} \tag{5.4}$$

Note that it must be considered in the real applications which each components have different cost and failure rate relationship. The above relationship is constructed by using a polynomial fitting method.

By using proposed framework, eight optimal topologies are classified as optimal which are depicted in the figure 5.4. The dashed line in red indicate the interval which the failure rates of components are located. The blue points show the optimal failure rate under cost consideration in the figure 5.4. This application is required to perform ASIL(D) functions. Therefore the quantitative values from ISO 26262 shall be kept based on the target values in the table 2.2 [Int11]. Designers are required to check which topology is applicable in reality due to other constraints like software architecture, size, energy consumption, weight, and etc. After several reviews, one or two topologies can be considered as feasible during design. In the given example, the topology 4 seems to be a realistic topology which can be implemented for large-scale production. The iteration steps to find the optimal components for topology 4 are plotted in the figure 5.5.

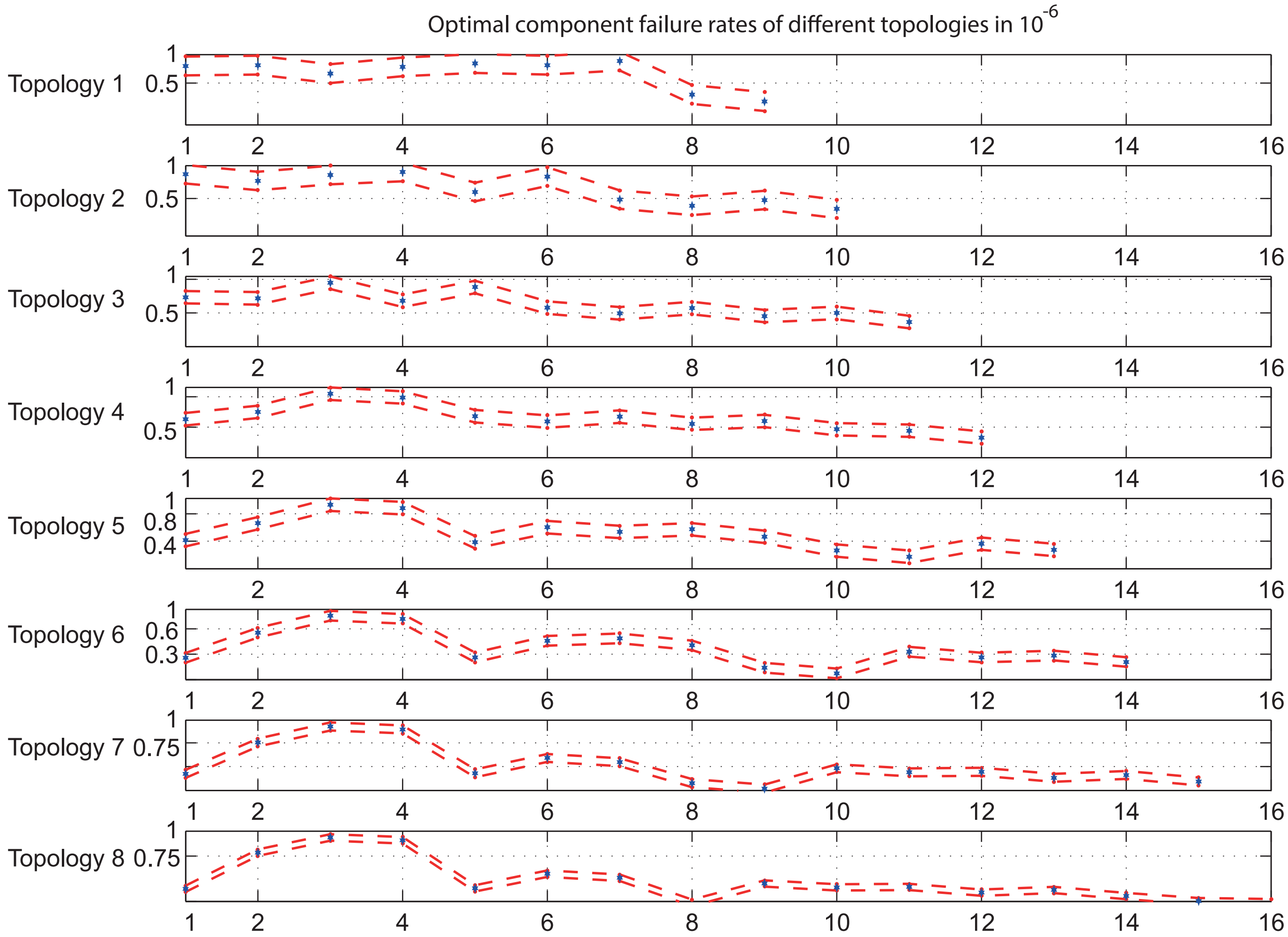


Figure 5.4.: Optimal reliabilities of different topologies

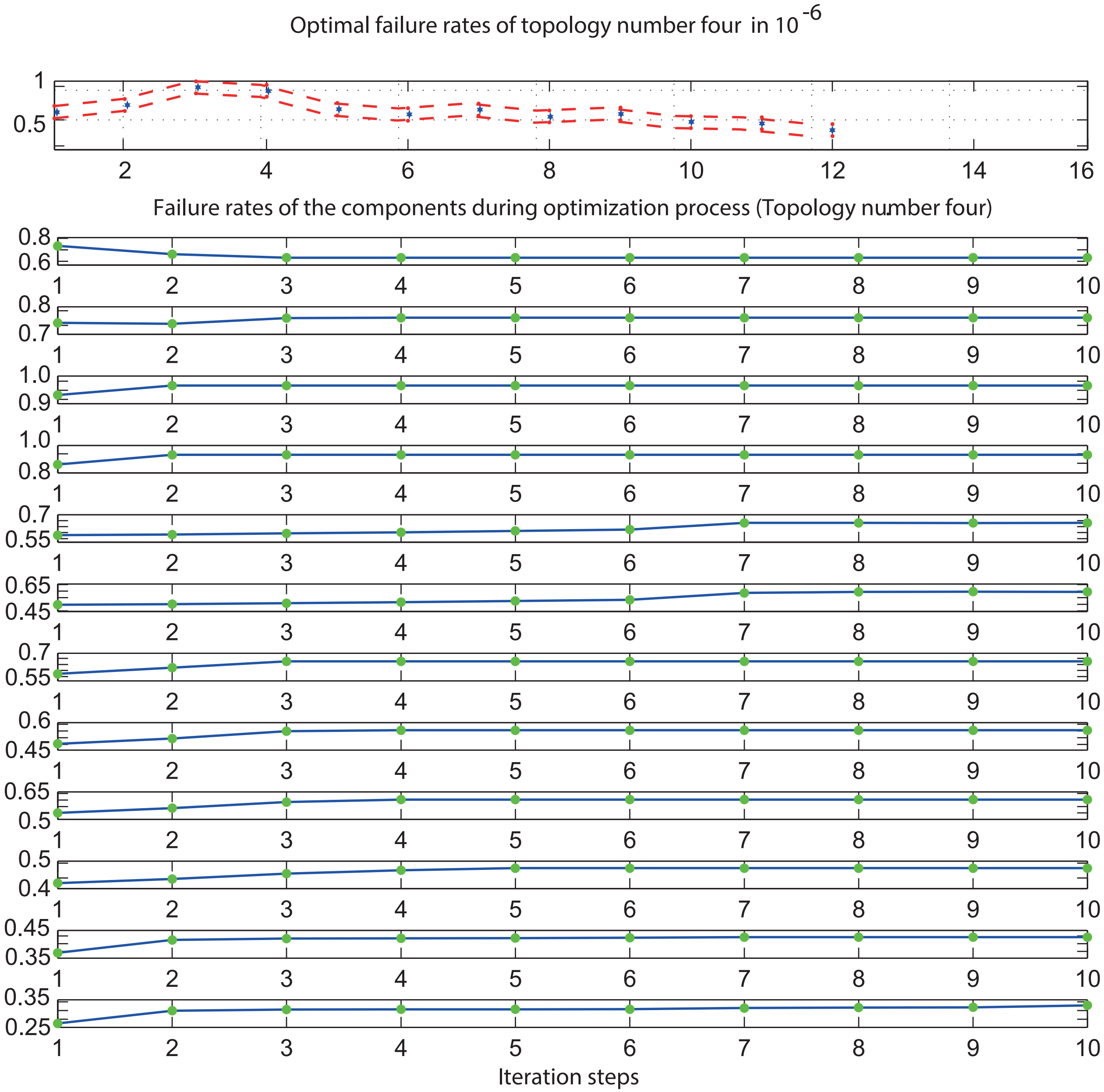


Figure 5.5.: Reliability of the components during optimization process (Topology number four)

Representative number	Component	Failure rate
1	Handwheel angle sensor	$5.3 \cdot 10^{-6}$
2	ECU	$7.1 \cdot 10^{-6}$
3	Communication	$9.4 \cdot 10^{-6}$
4	Motor	$9.2 \cdot 10^{-6}$
5	Steering angle sensor	$6.7 \cdot 10^{-6}$
6	Motor position sensor	$6.1 \cdot 10^{-6}$
7	Handwheel feedback motor	$6.6 \cdot 10^{-6}$
8	Power supply	$5.8 \cdot 10^{-6}$
9	Second ECU	$6.1 \cdot 10^{-6}$
10	Second power supply	$4.8 \cdot 10^{-6}$
11	Second motor	$4.1 \cdot 10^{-6}$
12	Second handwheel angle sensor	$3.5 \cdot 10^{-6}$

Table 5.3.: Proposed components for the topology 4

The proposed optimized components for topology 4 is given in the table 5.3. The RBD of the optimized topology is illustrated in the figure 5.6. The optimization problem at the component level is solved using a numerical deterministic approach.

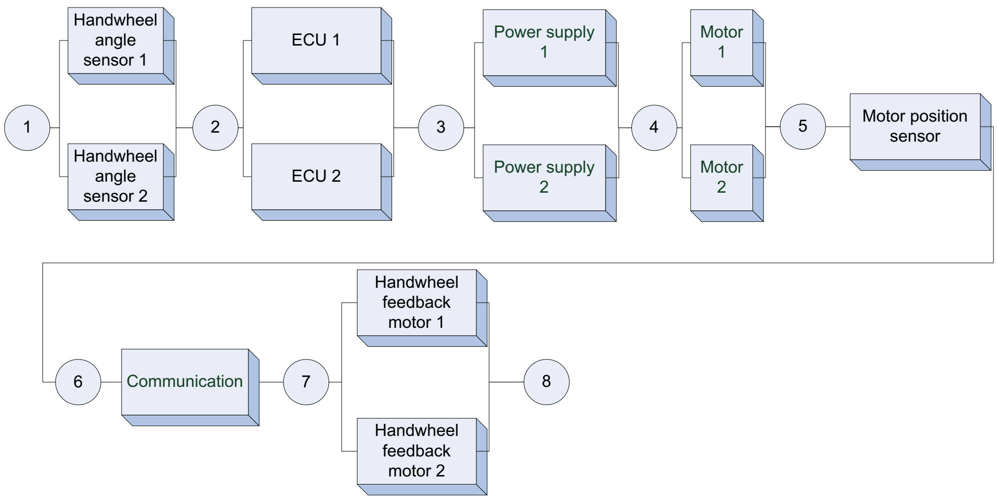


Figure 5.6.: The optimized RBD for steer-by-wire system - Topology 4

5.2. Brake-by-wire example

Conventional hydraulic brakes are currently a well established solution in order to generate the needed braking torque to stop an automobile. Such conventional systems consist of a mechanical pedal, hydraulic fluid, different pipes and the brake actuators. Once the brake pedal is pushed by the driver, the hydraulic brake fluid deliver the pressure which presses the pads on the brake discs [Kar12].

In order to improve active safety system in the vehicle, the drive by wire trend has been reached the braking domains as well. The principle of EMB is based on electrical actuating of brakes by a high performance electric motor (electromagnet) mounted in the drum brake instead of a hydraulic mechanism [Kar12] [Lin07].

Pressing the brake pedal is substituted by an actuation unit consisting of a pedal feel simulator and divers sensors to capture driver requests. The braking power is provided directly at each wheel by high performance electric motor. The motor is actuated by an ECU which processes the signals from an electronic pedal unit and other sensor clusters. The EMB system also could offer all known active safety functions such as ABS, EBD, TCS, ESP, etc [Kar12].

Benefits of the EMB can be expressed as follows [Bal04] [Kar12] [Del05] [Aut07]:

- shorter deceleration distances and quicker response during braking process
- lower cost,
- higher safety,
- reduced number of components and cutting wiring volume,
- avoiding pedal vibration by using of ABS,
- removal of brake fluid (environmentally friendly product, elimination of hazardous brake),
- reducing occupied space,
- integration of all the required braking and stability functions like ABS, EBD, TCS, ESP, ACC, etc., and
- simpler maintenance.

The minimal hardware subsystems which are required to perform EMB brake activity, are illustrated in the figure 5.7. The to be accomplished functions and data transformation among hardware element are given in the figure 5.8. Based on the function description, an initial RBD is suggested which is shown in the figure 5.9.

The optimization problem is expressed by

$$
\begin{aligned}
& J = \alpha \cdot \frac{1}{(1 - f(R_i(\lambda_i)))} + \beta \cdot \sum_i^n C_i(\lambda_i) + \\
& \qquad \mu \cdot \sum_i^n S_i(\lambda_i) + \gamma \cdot \sum_i^n W_i(\lambda_i) \\
& J \rightarrow \min. \\
& n: \qquad \text{number of components} \\
& \alpha,\ \beta,\ \mu,\ \text{and}\ \gamma: \qquad \text{weighting factors} \\
& \begin{cases}
\sum_i^n C_i \ \leq \ C_{\max} = 390, \\
\sum_i^n S_i \ \leq \ S_{\max} = 300, \\
\sum_i^n W_i \ \leq \ W_{\max} = 700, \\
\prod_{i=1}^k R_i \ \geq 0.3 \quad \text{and} \\
R_{\text{sys}} \ \geq \ 0.8
\end{cases}
\end{aligned}
\tag{5.5}
$$

The cost and failure rate relation of components

$$
\begin{aligned}
C_(\lambda_i) =& 8032 \cdot \lambda_i^7 - 4215 \cdot \lambda_i^6 + 65000 \cdot \lambda_i^5 - 48739 \cdot \lambda_i^4 + 17363 \cdot \lambda_i^3 \\
& - 1500 \cdot \lambda_i^2 + 30 \cdot \lambda_i .
\end{aligned}
\tag{5.6}
$$

is fitted by the polynomial method. While size and failure rate relation of components, weight and failure rate relationship of components

$$
\begin{aligned}
S_(\lambda_i) =& - 55930 \cdot \lambda_i^5 + 38030 \cdot \lambda_i^4 + 12722 \cdot \lambda_i^3 - 1500 \cdot \lambda_i^2 + 35 \cdot \lambda_i , \\
W_(\lambda_i) =& - 16642 \cdot \lambda_i^7 + 70522 \cdot \lambda_i^6 + 76251 \cdot \lambda_i^5 - 71982 \cdot \lambda_i^4 + 31001 \cdot \lambda_i^3 \\
& - 3200 \cdot \lambda_i^2 + 56 \cdot \lambda_i
\end{aligned}
\tag{5.7}
$$

are constructed by a spline interpolation method.

The following subsystems are subject to change:

- MC,
- ASIC,
- battery,
- lateral acceleration sensor,
- steering angle sensor,
- communication, and
- yaw sensor.

Components	Choice 1 $[10^{-6}]$	Choice 2 $[10^{-6}]$	Choice 3 $[10^{-6}]$	Choice 4 $[10^{-6}]$
Battery	[0.45 ; 0.75]	[0.2 ; 0.45]	[0.5 ; 0.76]	[0.2 ; 0.4]
MC	[0.3 ; 0.6]	[0.5 ; 0.8]	[0.4 ; 0.75]	[0.5 ; 0.9]
Communication	[0.45 ; 0.65]	[0.5 ; 0.8]	[0.3 ; 0.65]	[0.4 ; 0.75]
Steering angle sensor	[0.55 ; 0.75]	[0.25 ; 0.65]	[0.35 ; 0.7]	[0.4 ; 0.75]
Yaw sensor	[0.45 ; 0.75]	[0.5 ; 0.7]	[0.25 ; 0.5]	[0.35 ; 0.65]
ASIC	[0.2 ; 0.5]	[0.3 ; 0.55]	[0.45 ; 0.8]	[0.65 ; 0.9]
Lateral acceleration sensor	[0.45 ; 0.75]	[0.3 ; 0.6]	[0.3 ; 0.7]	[0.55 ; 0.85]

Table 5.4.: Failure rate interval of various choices for EMB

For some applications, it is desired to have a Pareto frontier instead of an exact solution which offers the designers the flexibility for better decision. Indeed designers are required to evaluate which topology is feasible due to other constraints like functional safety, software architecture, energy consumption, and etc. After several reviews, one or two topologies can be considered as feasible during design. After using the proposed platform, six different topologies based their RBDs are classified as optimal topologies. The figure 5.10, figure 5.12, and figure 5.14 demonstrate the Pareto frontier for size and weight in association with cost and reliability. While the figure 5.11, figure 5.13, and figure 5.15 show the RBD for each topologies respectively. The investigated optimization problem is solved by application of the genetic algorithm. In the given example, the topology 5 seems to be an optimal topology which can be implemented for large-scale production. Pareto frontier for topology 5 are plotted in the figure 5.14.

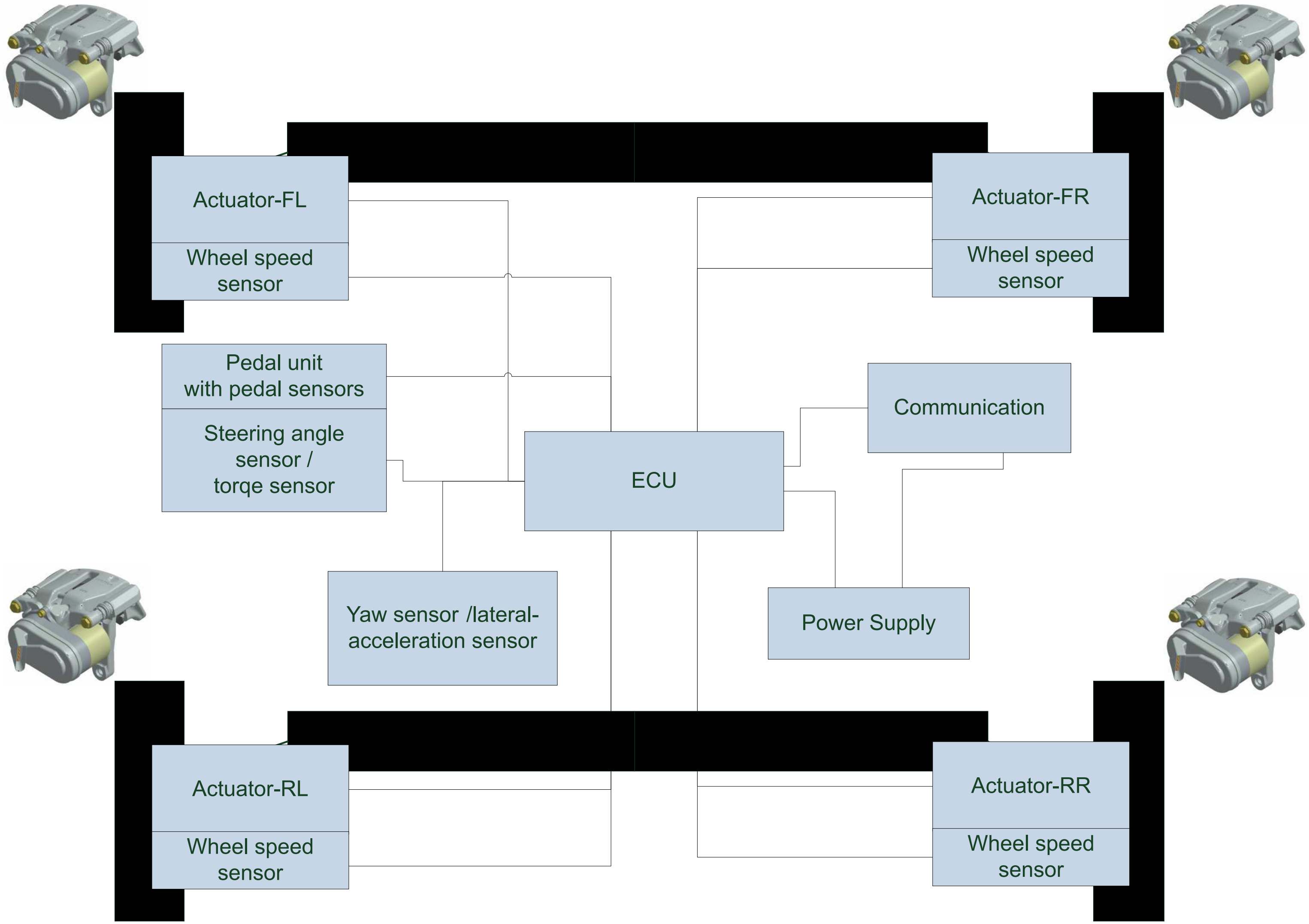


Figure 5.7.: Initial typical EMB schematic

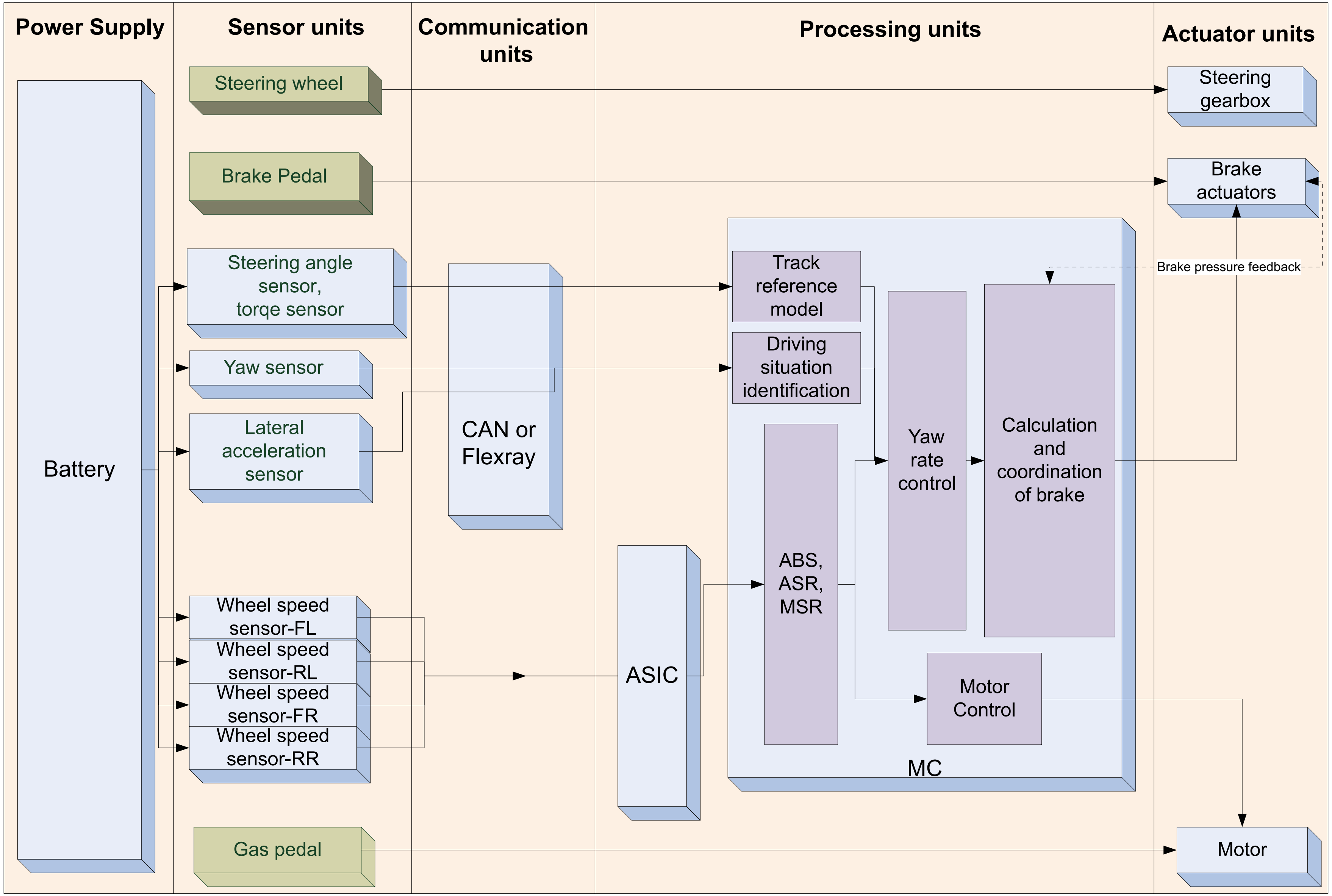


Figure 5.8.: Description of typical EMB functions

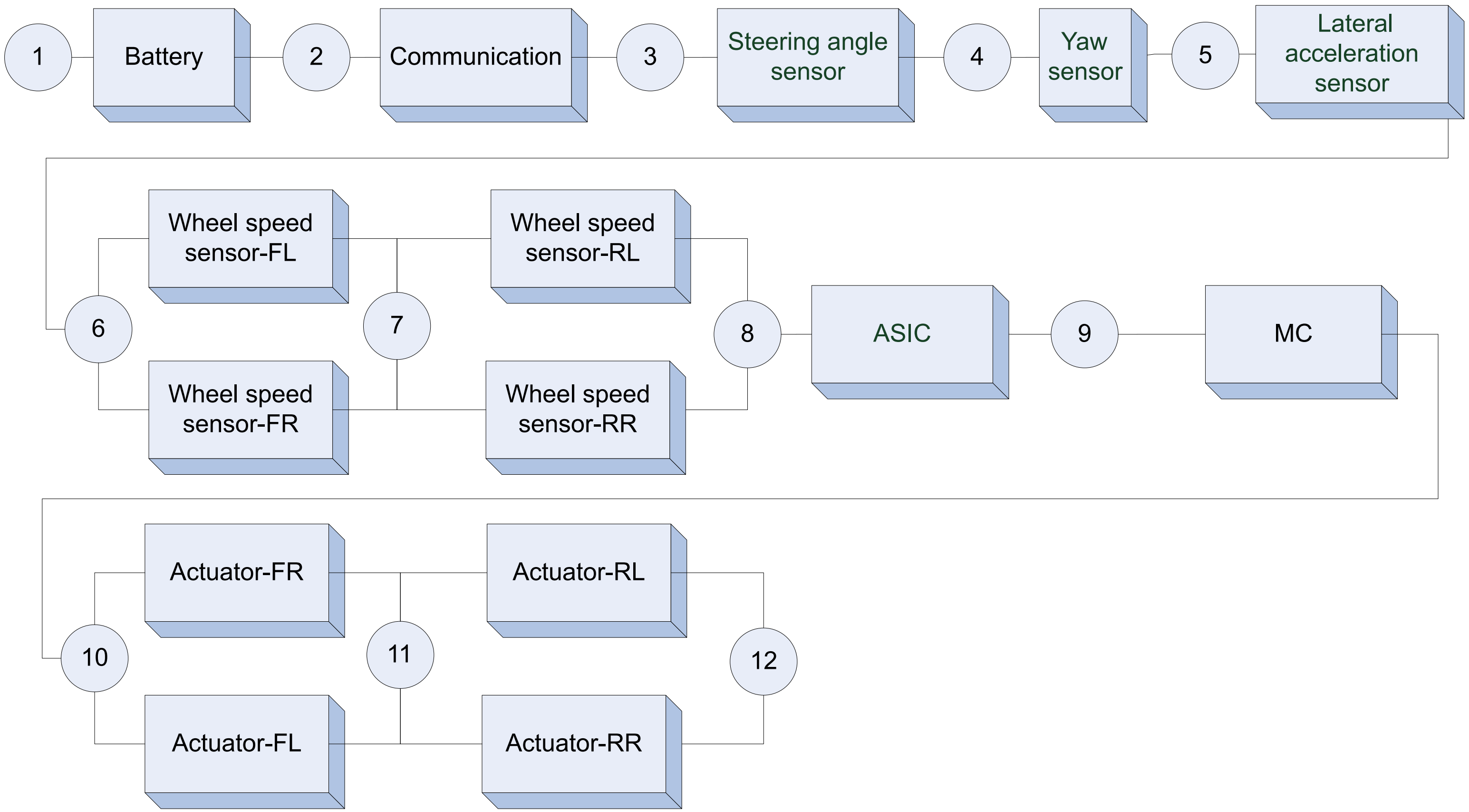


Figure 5.9.: RBD of EMB system based on the given function

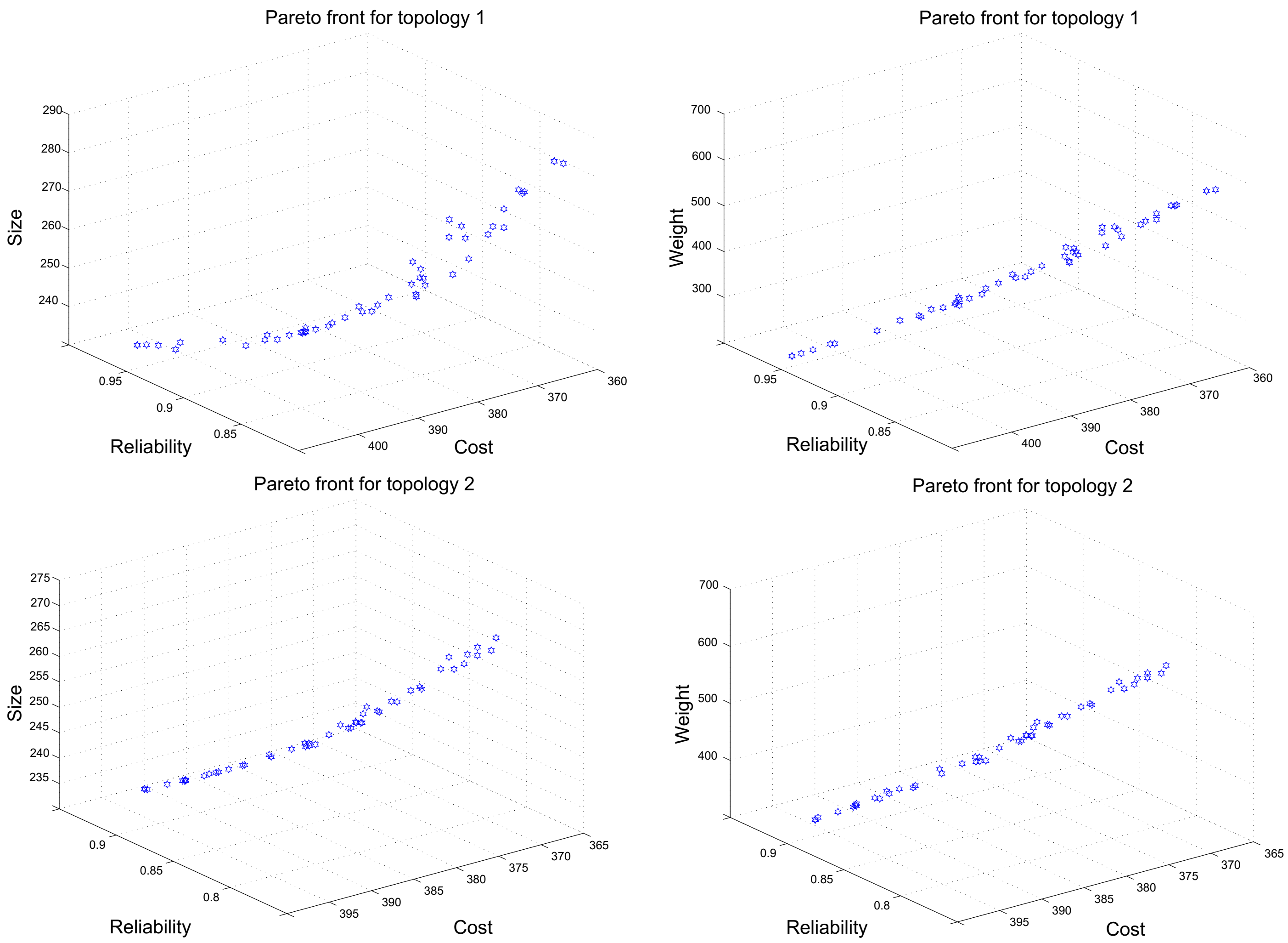


Figure 5.10.: Pareto front for topology 1 and topology 2

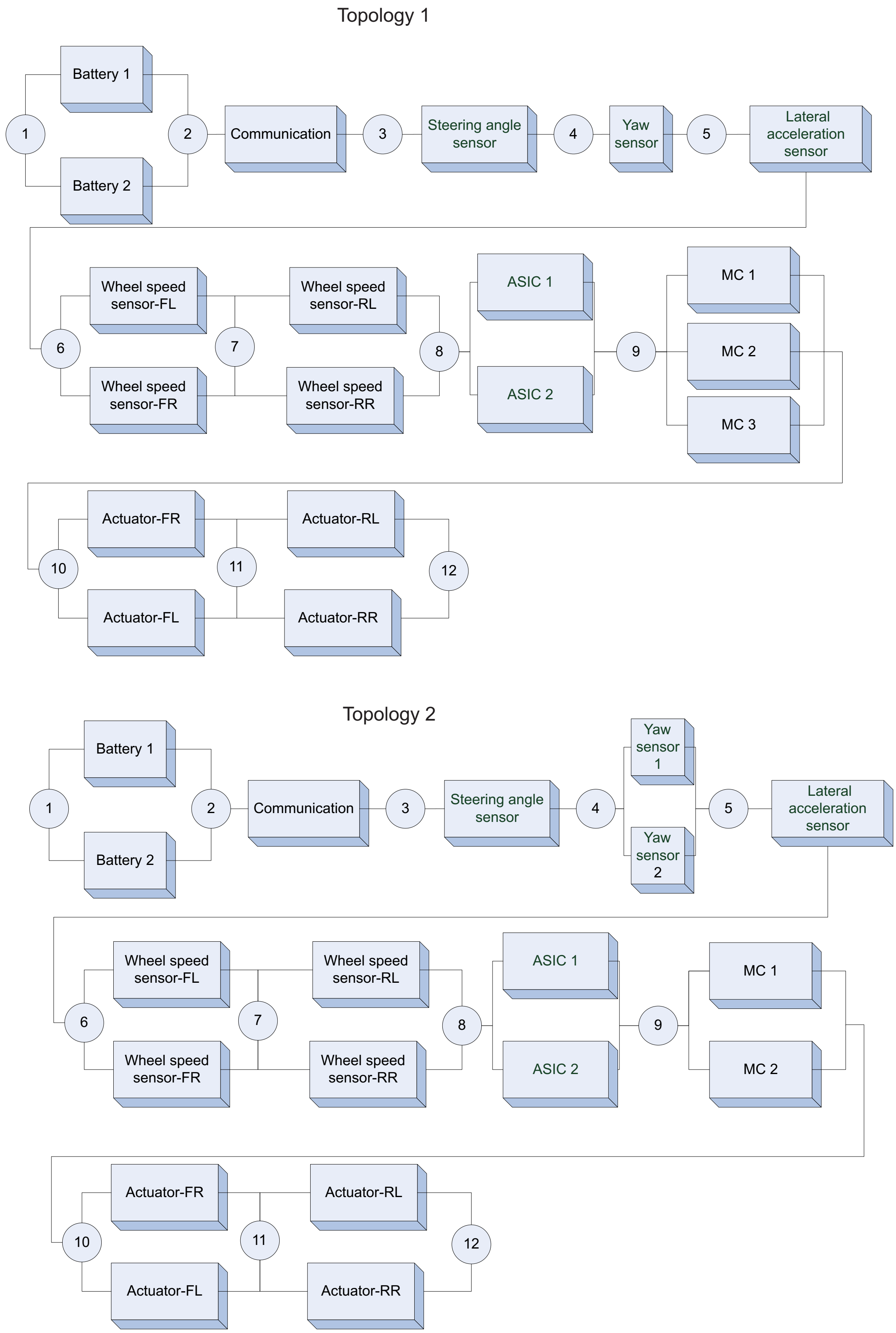


Figure 5.11.: RBD for topology 1 and topology 2

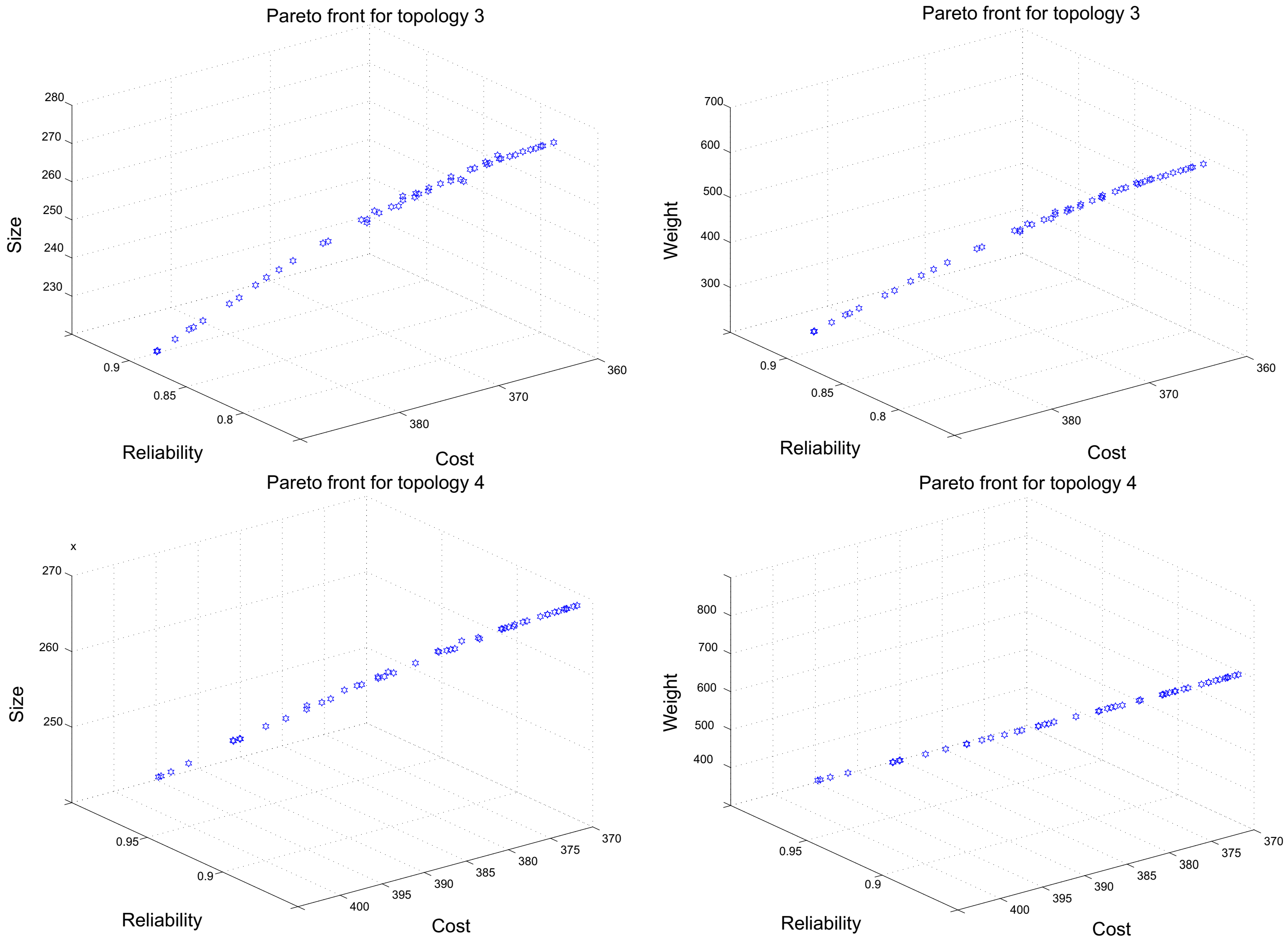


Figure 5.12.: Pareto front for topology 3 and topology 4

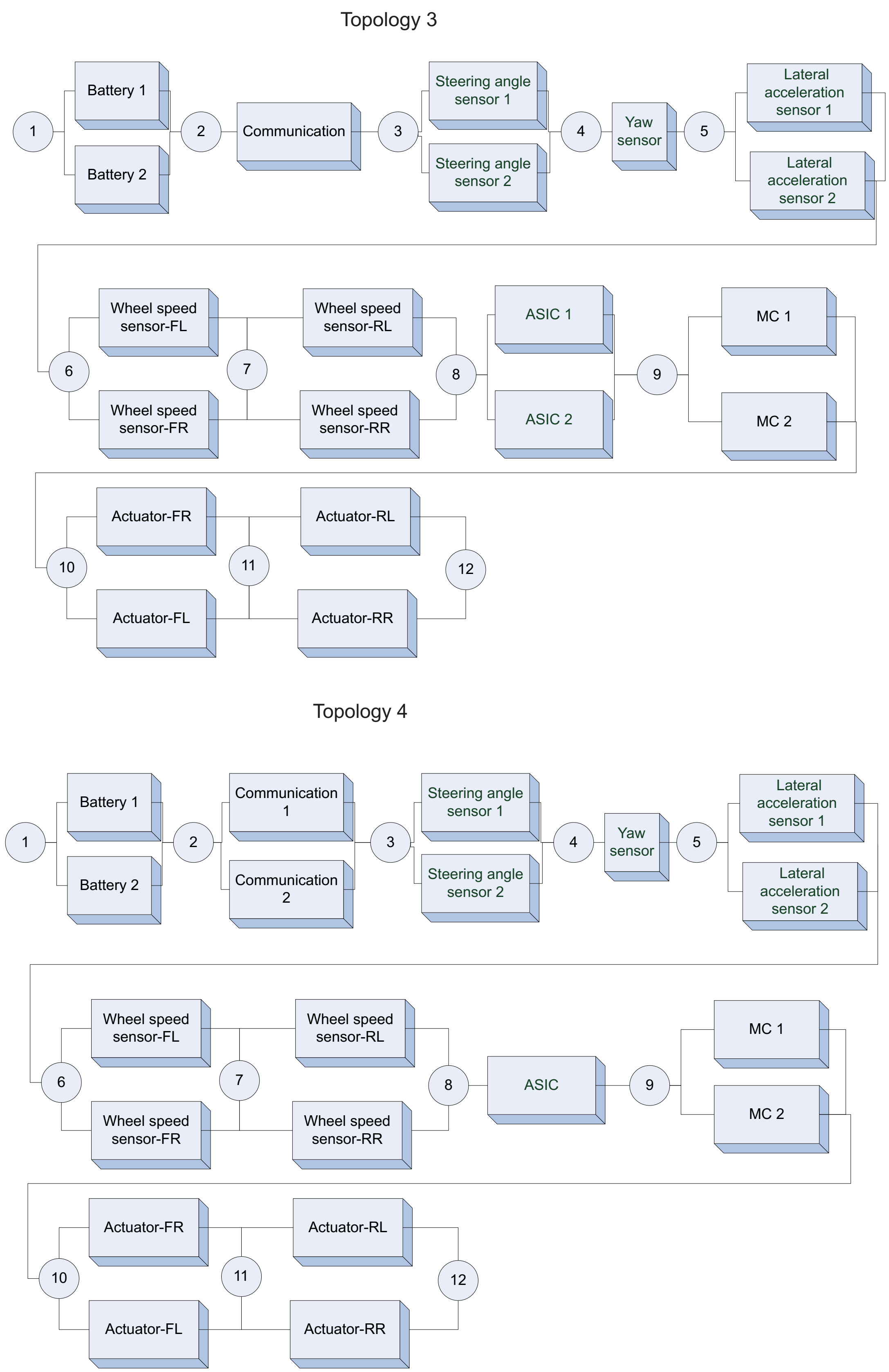


Figure 5.13.: RBD for topology 3 and topology 4

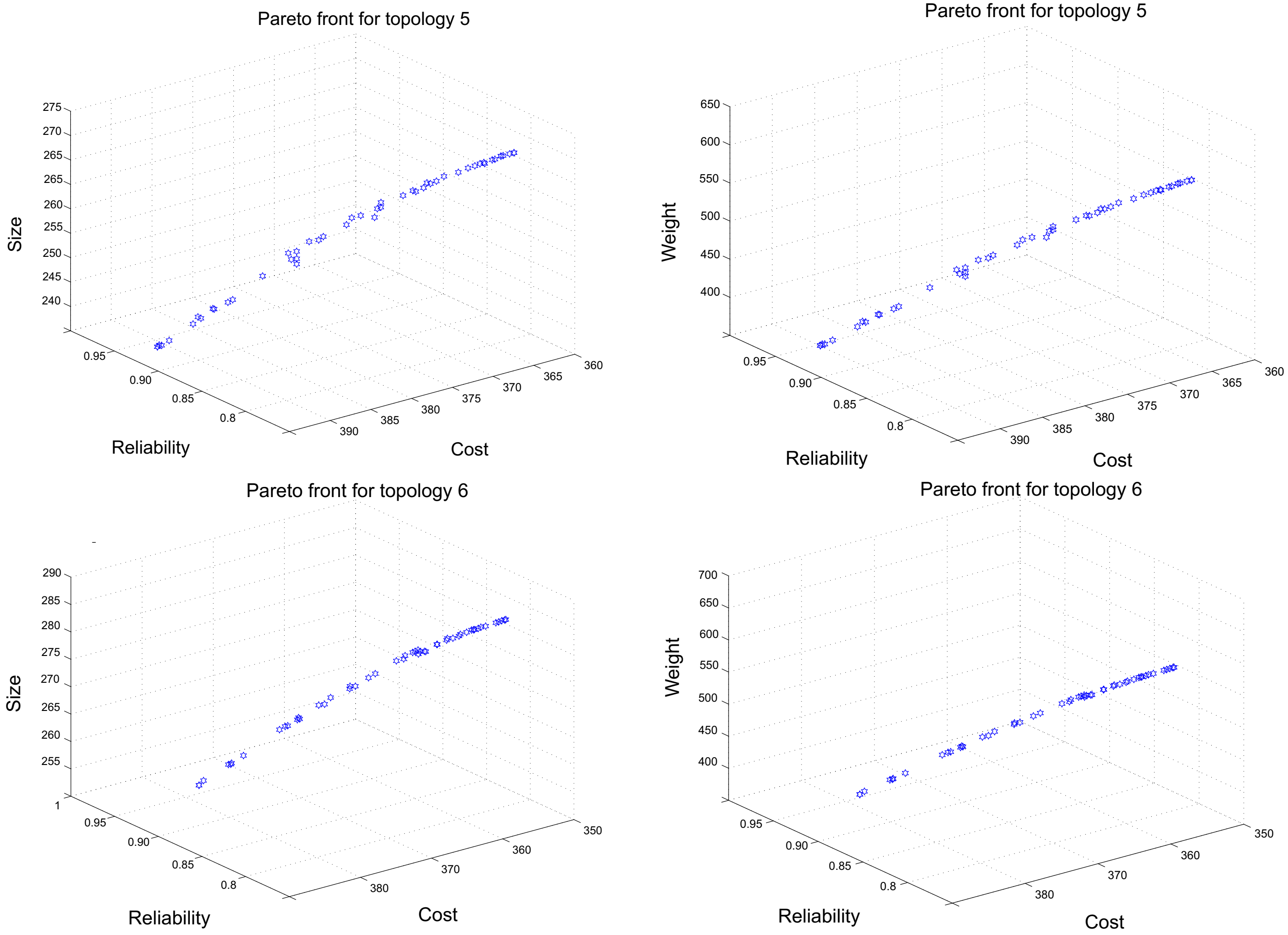


Figure 5.14.: Pareto front for topology 5 and topology 6

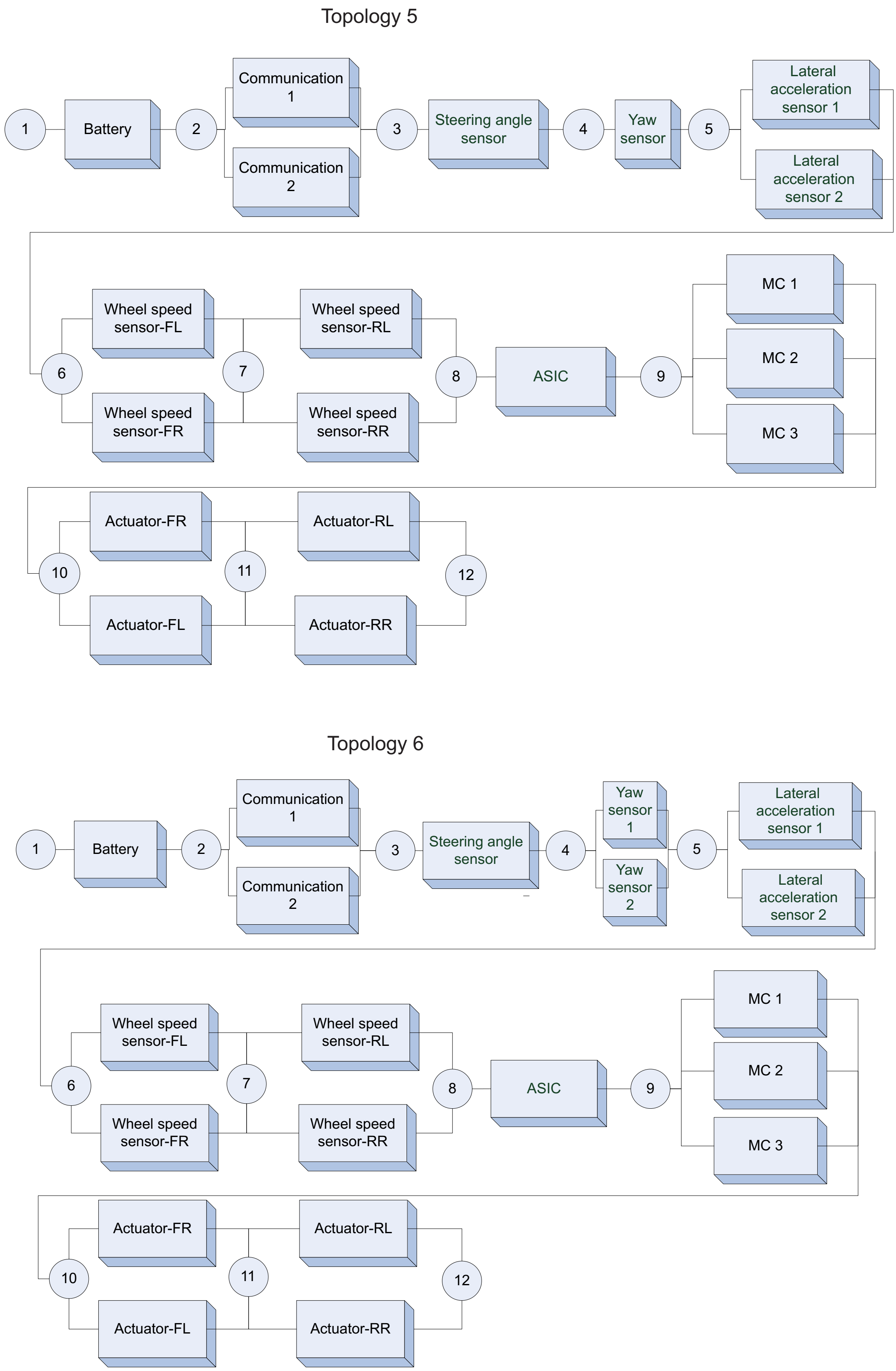


Figure 5.15.: RBD for topology 5 and topology 6

6. Summary and outlook

6.1. Summary

Reliability requirements are great challenges or showstoppers for the introduction and acceptance of many innovative products like drive-by-wire systems. In order to design and build a reliable system it is pivotal to precisely investigate the specific requirements for reliability in all the subsystems during early design phase.

Besides to this aspect, the engineering challenge currently is to design mass-producible systems with the typical constraints like low cost, low weight, small size, using modularity and maintainability etc. All these objectives and constraints require a smart component and topology selection strategy. Due to numerous conflicting design objectives and constraints, designers are required to be supported by an optimization tool. Exploring all design options can not be undertaken manually due to complexity. The designs investigated by such coherent optimization tool resonate more likely with customer and stakeholders expectations, law organizations than manual choosing of a design based on the past technical experience.

This dissertation basically proposes an efficient reliability optimization framework which helps the design team to determine the optimal hardware topology with optimal set of components under typical technical and financial restrictions.

In this context, the reliability and optimization basics have been introduced which are used during optimization framework. For modeling of system reliability, RBD has been applied because the design teams are more familiar with this method in practice. After a reliability model has been developed and checked by experts, an approach for reliability calculation has been developed which lends itself easily to automation. The proposed approach for reliability calculating detects parallel and serial structure in the RBD and converts bridge structure into the parallel and serial structures. The developed reliability calculation approach is based on a system reliability matrix (SRM). This matrix is used in the optimization process during both steps namely:

- topology varying and determination of optimal feasible topologies and
- component characteristic optimization.

In the first step, the proposed reliability optimization framework begins with topology variation. The introduced SRM-description allows the possibility to alter the system topology. Each column of SRM represents a hardware component. By omitting an arbitrary matrix column and adding new columns, new SRMs (new topologies) can be created.

In the second step, the framework involves optimizing component characteristics. The introduced SRM calculates up-to-date reliability value of the ongoing topology whenever the optimization process at the component level requires it. The numerical deterministic methods and genetic algorithms are deployed for component properties optimization and allows minimizing or maximizing any arbitrary objective function under consideration of given constraints.

The design of objective functions is an important step because inadequate weighting factors or improper choosing of the sub-functions could lead to non-optimal components configuration. In this dissertation, the weighted sum method are used to convert a multi-objective optimization problem to a single-objective mathematical optimization problem. This single objective function is constructed as a sum of objective functions like reliability and cost, weight, size multiplied by weighting coefficients. It is also suggested to specify all sub objective functions (cost, size, weight, etc.) in association with failure rates like reliability sub function. Reducing the number of optimization parameters and shedding light on relationship between the technical and economic objective functions and failure rates of each hardware components are the most important benefits of this strategy. Some instances have been analyzed to investigate the influence of an appropriate objective function formulation strategy and advantages/disadvantages of the applied numerical optimization methods at the component level.

Besides two practical applications with fictive data have been analyzed by using proposed approach. Consequently some optimal feasible designs have been determined.

6.2. Outlook

In this dissertation, RBD has been suggested for reliability modeling and SRM approach illustrates RBD in a mathematical way. A possible area which can be discussed and advanced in the future, is the using of other reliability modeling possibilities like FTA, Markov modeling, and Monte Carlo simulation.

In the case of application of other reliability modeling methods such as FTA, Markov modeling, and Monte Carlo simulation, it is mandatory to find a reliability calculation way like SRM approach once the optimization process requires the current reliability value of system. The reliability evaluation approach must be capable of being automated. This domain can be extended in the future works.

The other topic, which can be investigated more deeply, is related to extension of number of optimization parameters. For instance, it is interesting to include availability of system, mean time to repair, the system energy consumption into optimization process.

In this contribution, it is suggested to change the topology of system by using hardware redundancy strategy in order to enhance the system reliability. Besides it is attractive for design team to investigate if it is possible to increase the reliability by using other strategy regardless to component redundancy strategy. It is worthwhile to check if a group of components can be substituted by an another group of components with different structure. In other words, the platform is not supposed to merely add some components in parallel in order to increase the reliability.

Finally it is reasonable to investigate if it is possible to consider software reliability parameter during implementation of such reliability optimization framework.

Im Rahmen von Forschungs- und Projektarbeiten im Lehrstuhl SRS wurden von Herrn Univ.-Prof. Dr.-Ing. Dirk Söffker und Herrn Amir Kazeminia die Studienarbeit von Frau Olga Muthig [Mut12] und die Bachelorarbeit von Frau Lisa Heistermann [Hei09] inhaltlich betreut, wobei Bestandteile und Ergebnisse aus den Forschungs- und Projektarbeiten sowie den studentischen Qualifikationsarbeiten wechselseitig in die jeweiligen Arbeiten und somit auch in diese Promotionsarbeit eingeflossen sind.

Bibliography

[ADM99] Amari, S. V. ; Dugan, J. B. ; Misra, R. B.: Optimal reliability of systems subject to imperfect fault-coverage. In: *IEEE Transactions on reliability* Bd. 53, 1999, S. 275–284.

[ADS03] Altiparmak, F. ; Dengiz, B. ; Smith, A. E.: Optimal design of reliable computer networks: A comparison of metaheuristics. In: *J. Heuristics* Bd. 9, 2003, S. 471–487.

[Ams77] Amstadter, B. L.: *Reliability mathematics: fundamentals, practices, procedures.* McGraw-Hill, 1977.

[And01] Andersson, J.: *Multiobjective Optimization in Engineering Design – Applications to Fluid Power Systems.* Linköping, Sweden, Linköping University, Linköping studies in science and Technology, Dissertation No. 675, 2001.

[APD04] Amari, S. V. ; Pham, H. ; Dill, G.: Optimal design of k-out-of-n: G subsystems subjected to imperfect fault-coverage. In: *IEEE Transactions on reliability* Bd. 53, 2004, S. 567–575.

[Aut07] Automotive Systems Division of Continental AG: *Continental Electronic Brake Systems, Auburn Hills, MI.* 2007. – Technical report.

[AUT12] AUTOSAR: *AUTOSAR (AUTomotive Open System ARchitecture).* 2012. – Online available at `http://www.autosar.org/`; last checked: 28.07.2012.

[Bal04] Balazovic, P.: *56F8300 Hybrid Controller Used in Control of Electro-Mechanical Brake.* 2004. – Freescale Semiconductor, Application note.

[BBC] BBC: *Toyota recalls 110,000 hybrid cars on safety concerns.* – Online available at `http://www.bbc.co.uk/news/mobile/business-13970431`; last checked: 28.07.2012.

[Bec11] Bechtold, L. E.: Industry consensus approach to physics of failure in reliability prediction. In: *Reliability and Maintainability Symposium (RAMS)*, 2011.

[Bes06] Besharati, B.: *Deterministic multi-objective robust optimization for single product and product line engineering with design and marketing considerations*, University of Maryland, College Park, Diss., 2006.

[BFM+03] Baleani, M. ; Ferrari, A. ; Mangeruca, L. ; Sangiovanni-Vincentelli, A. ; Peri, M. ; Pezzini, S.: Fault tolerant platforms for automotive safety-critical systems, 2003.

[BOS] BOSCH: *Mechatronic systems in vehicles*. – Online available at `http://www.bosch.de`; last checked: 28.07.2012.

[BS79] Bazaraa, M. S. ; Shetty, C. M.: *Nonlinear Programming theory and Practice*. Wiley New York, 1979.

[BSS93] Bazaraa, M. S. ; Sherali, H. D. ; Shetty, C. M.: *Nonlinear Programming – theory and algorithms*. John Wiley & Sons, New York, 1993.

[Che08] Chen, X.: *Requirements and concepts for future automotive electronic architectures from the view of integrated safety*, University of Karlsruhe, Diss., 2008.

[CJW04] Coit, D. W. ; Jin, T. ; Wattanapongsakorn, N.: System Optimization With Component Reliability Estimation Uncertainty: A Multi-Criteria Approach. In: *IEEE Transactions on reliability* Bd. 53, 2004, S. 369–380.

[CL00] Coit, D. W. ; Liu, J.: System reliability optimization with k-out-of-n subsystems. In: *International Journal Reliability Quality Safety Engineering* Bd. 7, 2000, S. 129–142.

[CM07] Caserza Magro, M.: Advanced Power Converters for Universal and Flexible Power Management in Future electricity Networks, 2007

[Coe08] Coelho, L. .: Reliability-redundancy optimization by means of a chaotic differential evolution approach, 2008, S. 594–602.

[Coi01] Coit, D. W.: Cold-standby redundancy optimization for nonrepairable systems. In: *IEEE Transactions on reliability* Bd. 33, 2001, S. 471–478.

[Coi03] Coit, D. W.: Maximization of system reliability with a choice of redundancy strategies. In: *IEEE Transactions on reliability* Bd. 35, 2003, S. 535–543.

[CP96] Campbell, J. E. ; Painton, L. A.: Optimization of reliability allocation strategies through use of genetic algorithms. In: *Proceedings 6th. symposium multidisciplinary design and optimization*, 1996, S. 1233–1242.

[CS96a] Coit, D. W. ; Smith, A.: Adaptive penalty method for genetic optimization of constraind combinatorial problems. In: *INFORMS journal on computing*, 1996, S. 173–184.

[CS96b] Coit, D. W. ; Smith, A.: Reliability optimization for series-parallel systems using a genetic algorithm. In: *IEEE Transactions on reliability* Bd. 45, 1996, S. 895–904.

[CVVL02] Coello, C. ; Van Veldhuizen, D. ; Lamont, G.: *Evolutionary Algorithms for Solving Multi-Objective Problems*. Springer-Verlag, 2002.

[Deb01] Deb, K.: *Multi-Objective Optimization using Evolutionary Algorithms*. John Wiley & Sons, 2001.

[Del05] Delphi, Co.: *Delphi Brake Modules - Electric Caliper, Troy, MI*. 2005. – Technical report.

[Den04] Denston, T.: *Automobile electrical and electronic systems*. Elsevier Butterworth-Heinemann, 2004.

[Dhi99] Dhillon, B. S.: *Design Reliability: Fundamentals and Applications*. CRC Press, 1999.

[FF95] Fonsecay, C. M. ; Fleming, P. J.: An overview of evolutionary algorithms in multiobjective optimization, 1995, S. 1–16.

[FGF80] Frair, L. C. ; Ghare, P. M. ; Frair, K. L.: Optimization of System Reliability via Redundancy and/or Design Considerations. In: *IEEE Transactions on reliability* Bd. R-29, 1980, S. 33–35.

[FGW02] Forsgren, A. ; Gill, P. E. ; Wright, M. H.: Interior Methods for Nonlinear Optimization. In: *SIAM Rev. 44*, 2002, S. 525–597.

[FHL68] Fyffe, D. E. ; Hines, W. W. ; Lee, N. K.: System Reliability Allocation and a Computational Algorithm. In: *IEEE Transactions on reliability* Bd. R-17, 1968.

[Gei06] Geis, S. R.: *Integrated Methodolgy for Production related risk management of vehicle electronic (IMPROVE)*, University of Karlsruhe, Diss., 2006.

[GSS98] Galla, J.J. ; Schedl, T ; Sphohr, J: X-by Wire safety related fault-tolerant systems in vehicles. In: *Europian commission-Britte-EuRam III*, 1998.

[GW10] Guertin, S. ; White, M.: CMOS Reliability Challenges - The Future of Commercial Digital Electronics and NASA. In: *NEPP Electronic Technology Workshop*, 2010. – Available at `http://nepp.nasa.gov/workshops/etw2010/talks/21_Thu_Guertin_CMOS%20Reliability%20Challenges%20-%20The%20Future%20of%20Commercial%20Digital%20Electronics%20and%20NASA.pdf`.

[Han92] Hansen, E. R.: Global optimization using interval analysis, 1992.

[Hei09] Heistermann, L.: *Zuverlässigkeitsanalyse mechatronischer Systeme mittels Markov-Modellen.* October 2009. – University of Duisburg-Essen, Chair of Dynamics and Control, Bacherlorarbeit.

[HKK04] Hardung, B. ; Kölzow, T. ; Krüger, A.: Reuse of Software in Distributed Embedded Automotive Systems. In: *EMSOFT '04*, 2004.

[HTF96] Harashima, F. ; Tomizuka, F. ; Fukuda, T.: Mechatronics- What Is It, Why, and How? An Editorial, 1996, S. 1–3.

[Hwa79] Hwang, C. L.: Reliability Optimization by Generalized Lagrangian-Function and Reduced-Gradient Methods, 1979, S. 316 – 319.

[HY81] Hwang, C.-L. ; Yoon, K.: Multiple objective decision making, methods and applications: a state-of-the-art survey. In: Beckmann, M. (Hrsg.) ; Kunzi, H. P. (Hrsg.): *Lecture notes in economics and mathematical systems* Bd. 186, Springer-Verlag, Berlin, 1981.

[IAU76] Institute, IIT R. ; Anderson, R.T. ; (U.S.), Reliability Analysis C.: *Reliability Design Handbook.* Rome Air Development Center, 1976. (Reliability analysis center: Catalog)

[Ign78] Ignizio, J. P.: A review of goal programming: A tool for multiobjective analysis, 1978, S. 1109–1119.

[Int04] International Electrotechnical Commission: *IEC TR 62380, Reliabilitydata handbook- Universal model for reliability prediction of electronic components, PCBs, and equipment.* 2004.

[Int11] International Organization for Standardization: *ISO 26262-5 Road vehicles - Functional safety, Part 5: Product development at the hardware level.* 2011.

[ISS02] Iserman, R. ; Schwarz, R. ; Stolzl, S.: Fault-tolerant Drive-by-Wire Systems. In: *IEEE Control Systems Magazine* Bd. 22, 2002, S. 64–81.

[JT10] Jones, D. ; Tamiz, M.: *Practical goal programming.* Springer, 2010.

[Kar12] Karakoc, K.: *Modeling and Design Optimization of Electromechanical Brake Actuator Using Eddy Currents*, Dept. of Mechanical Engineering, University of Victoria, Diss., 2012.

[KCS06] Konak, A. ; Coit, D. W. ; Smith, A. E.: Multi-objective optimization using genetic algorithms: A tutorial. In: *Reliability Engineering and System Safety 91*, 2006, S. 992–1007.

[KJS10] Kazeminia, A. ; Junglas, M. ; Soeffker, D.: An approach for reliability optimization of mechatronic systems during design phase. In: *Proc. ESREL 2009, Taylor & Francis Group*, 2010, S. 1505–1512.

[KL01] Kasprzak, E.M. ; Lewis, K.E.: Pareto analysis in multi-objective optimization using the colinearity theorem and scaling method, 2001, S. 208–218.

[KLXZ87] Kuo, W. ; Lin, H. ; Xu, Z. ; Zhang, W.: Reliability optimization with Lagrange multiplier and branch and bound technique. In: *IEEE transactions on reliabilty* Bd. R-36, 1987, S. 624–630.

[Kon07] Konnov, A.: *Zuverlässigkeitsberechnung und vorbeugende Wartung von komplexen technischen Systemen mittels modifizierter Markov-Modellen*, Universität Karlsruhe (TH), Diss., 2007.

[Kos84] Koski, J.: Multicriterion optimization in structural design. In: Atrek, E. (Hrsg.) ; Gallagher, R. H. (Hrsg.) ; Ragsdell, K. M. (Hrsg.) ; Zienkiewicz, O. C. (Hrsg.): *New directions in optimum structural design*, John Wiley & Sons, New, 1984, S. 483–503.

[KPTH01] Kuo, W. ; Prasad, V. R. ; Tillman, F. A. ; Hwang, C. L.: *Optimal Reliability Design: Fundamentals and Applications.* Cambridge University Press, 2001.

[KPTH06] Kuo, W. ; Prasad, V. R. ; Tillman, F. A. ; Hwang, C.: *Optimal Reliability Design: Fundamentals and Applications.* Cambridge University Press, 2006.

[Kra10] Kraftfahrzeugbundesamt (KBA): *Jahresbericht 2010.* 2010.

[KS87] Koski, J. ; Silvennoinen, R.: Norm methods and partial weighting in multicriterion optimization of structures, 1987, S. 1101–1121.

[Kuo87] Kuo, W.: Reliability Optimization with the Lagrange-Multiplier and Branch-and-Bound Technique, 1987, S. 624–630.

[KW07] Kuo, W. ; Wan, R.: Recent Advances in Optimal Reliability Allocation. In: *IEEE Transactions on systems, man, and cybernetics- Part A: Systems and humans* Bd. 53, 2007, S. 143–156.

[LH02] Leen, G. ; Heffernan, D.: Expanding Automotive Electronic Systems. In: *IEEE Computer* Bd. 35, 2002, S. 88–93.

[Lin07] Line, C. L. J.: *Modelling and control of an automotive electromechanical brake*, University of Melbourne, Department of Mechanical and Manufacturing Engineering, Diss., 2007.

[LNF11] Lanigan, P.E. ; Narasimhan, P. ; Fuhrman, T. E.: Diagnostic Fusion for Dependable Vehicle Architectures. In: *The Fourth International Conference on Dependability*, 2011.

[Lue90] Luenberger, D. G.: *Linear and Non-linear Programming*. Addison-Wesley, New York, 1990.

[MA04] Marler, R. T. ; Arora, J. S.: Survey of multi-objective optimization methods for engineering, Springer Berlin / Heidelberg, April 2004., S. 369–395.

[Met00] Mettas, A.: Reliability allocation and optimization for complex systems. In: *Proceedings Annual Reliability and Maintainability Symposium*, 2000, S. 216–221.

[ML73] Misra, K. ; Ljubojevic, M. D.: Optimal Reliability Design of a System: A New Look. In: *IEEE Transactions on reliability* Bd. R-22, 1973, S. 983–993.

[MMS07] Magro, M. C. ; Mazzucchelli, M. ; Savio, S.: Advanced power converters for universal and flexible power management in future electricity networks. In: *EPE Journal* Bd. 19, 2007.

[MP06] Munoz, H. ; Pierre, E.: Interval arithmetic technique for constraints reliability optimization problems. In: *NSF workshop on reliable engineering computing*, 2006.

[MP10] Meyna, A. ; Pauli, B.: *Zuverlässigkeitstechnik, Quantitative Bewertungsverfahren*. Carl Hanser Verlag GmbH & Co. KG, 2010.

[MS91] Misra, K. ; Sharma, U.: An effective approach for multiple criteria redundancy optimization problems. In: *Microelectron. Reliab.* Bd. 31, 1991, S. 303–321.

[Mut12] Muthig, O.: *Automatisierte Berechnung und Optimierung der Zuverlässigkeit eines mechatronischen Systems*. January 2012. – University of Duisburg-Essen, Chair of Dynamics and Control, Studienarbeit.

[OHTS11] O Halloran, B. M. ; Tumer, I. Y. ; Stone, R. B.: Early design stage reliability analysis using function-flow failure rate. In: *Proceedings of the ASME 2011 International Design Engineering Technical Conferences & Computers and Information in Engineering Conference*, 2011.

[Osy84] Osyczka, A.: *Multicriterion optimization in engineering with fortran programs*. John Wiley & Sons, New York, 1984.

[PSQX01] Proos, K. A. ; Steven, G. P. ; Querin, O. M. ; Xie, Y. M.: Multicriterion evolutinary structural optimization using the weighted and the global criterion methods. In: *AIAA Journal* Bd. 39, 2001.

[Rel] Reliability Engineering Resource Website: *Reliability Cost Relationship*. – Online available at `http://www.weibull.com`; last checked: 02.11.2012.

[RMCK04] Ramirez-Marquez, J. E. ; Coit, D.W. ; Konak, A.: Redundancy allocation for series-parallel systems using a maxmin approach, 2004, S. 891–898.

[Rom91] Romero, C.: *Handbook of critical issues in goal programming*. Oxford: Pergamon Press., 1991.

[RR89] Rao, J. R. ; Roy, N.: Fuzzy set theorotic approach of assigning weights to objectives in multicriteria decision making, 1989, S. 1381–1386.

[Sal79] Salukvadze, M. E.: *Vector valued optimization problems in control theory*. Academic Press, New York, 1979.

[SBMR+12] Suresh Babu, S. V. ; Maheswar, D. ; Ranganath, G. ; Vijaya Kumard, Y. ; Sankaraiah, G.: Redundancy Allocation for Series Parallel Systems with Multiple Constraints and Sensitivity Analysis, 2012, S. 424–428.

[SFP+12] Svancara, K. ; Forbes, W. ; Priddy, J. ; Kudanowski, M ; Lovric, T. ; Miller, J: Experience with second method for EPS hardware analysis: Evaluation of each cause of safety goal violation due to random hardware failures, 2012.

[SMWM09] Stumpf, F. ; Meves, C. ; Weyl, B. ; M., Wolf: A security architecture for multipurpose ECUs in vehicles. In: *25th joint VDI/VW automotive security conference*, 2009.

[Sta09] Stapelberg, R. F.: *Handbook of reliability, availability, maintainability and safety in engineering design*. Springer, 2009.

[TEC08] Taboada, H. A. ; Espiritu, J. F. ; Coit, D. W.: MOMS-GA: A Multi-Objective Multi-State Genetic Algorithm for System Reliability Optimization Design Problems. In: *IEEE transaction on reliability* Bd. 57, 2008, S. 182–191.

[The02] Theis, I.: *Das Steer-by-Wire-System im Kraftfahrzeug - Analyse der menschlichen Zuverlässigkeitsanalyse*, Lehrstuhl für Ergonomie, Technische Universität München, Diss., 2002.

[The04] The Association of German Engineers (VDI): *VDI 2206 – Design methodology for mechatronic systems*. Beuth Verlag, 2004.

[TRW09] TRW: *IAA Exhibition,Frankfurt, (NYSE:TRW)*. 2009.. – Online available at `http://trw.mediaroom.com/index.php?s=43&item=423`; last checked: 28.07.2012.

[VSVL08] Van Sickel, J.H. ; Venkatesh, P. ; Lee, K.Y.: Analysis of the pareto front of a multi-objective optimization problem for a fossil fuel power plant. In: *Power and Energy Society General Meeting - Conversion and Delivery of Electrical Energy in the 21st Century, IEEE*, 2008, S. 1–8.

[VV99] Van Veldhuizen, D. A.: *Multi-objective Evolutionary Algorithms: Classifications, Analyses, and New Innovations*, Department of Electrical and Computer Engineering. Graduate School of Engineering. Air Force Institute of Technology, Wright-Patterson AFB, Diss., 1999.

[WB11] Wyrwas, E. ; Bernstein, J.: Quantitatively Analyzing the Performance of Integrated Circuits and Their Reliability. In: *IEEE Instrumentation & measurement*, 2011.

[WCH11] Wyrwas, E. ; Condra, L. ; Hava, A.: Accurate Quantitative Physics-of-Failure Approach to Integrated Circuit Reliability. In: *IPC APEX*, 2011.

[Yih05] Yih, P.: *Steer-by-Wire: Implications for vehicle handling and safety*, Stanford University, Diss., 2005.

[ZT99] Zitzler, E. ; Thiele, L.: Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. In: *IEEE Transactions on Evolutionary Computation* Bd. 3, 1999, S. 257–271.

A. Source code

```
%% DEFINITION OF GLOABAL VARIABLES
clc, clear all, close all

global T;           % Initial topology
global L;           % Initial component reliability
global C;           % Initial costs
global S;
global W;
global Cost_Max;    % Violation level of costs (budget)
global history
global R_temp
global N_temp
global k
global Tp_hist
global Lp_hist
global TT_hist
global LL_hist

L0_hist = [];
L_hist = [];
fval_hist = [];
mflag_hist = [];
output_hist = [];
history.x = [];
history.fval = [];
R_temp = [];
N_temp = [];

%% DATA INPUT
T = input('Please enter the initial topology:\n');
L = input('Please enter the initial component reliability:\n');
C = input('Please enter the initial costs:\n');
S= input('Please enter the initial sizes:\n');
W = input('Please enter the initial weights:\n');
Cost_Max = input('Please enter the violation level of costs:\n');
Size_Max = input('Please enter the violation level of sizes:\n');
Weight_Max = input('Please enter the violation level of ...
    weights:\n');

%% GENERATION OF NEW TOPOLOGY AND OPTIMIZATION AT THE COMPONENT ...
    LEVEL
tol = (rand(1, 'double') * L(1)) / 10.0;   % Tolerance for ...

% Generierung der mÃ¶glichen Topologien
for k = 1:size(T, 2);

     % Selection of the ith—component
```

```
        temp_T = T(:,k);
        temp_L = L(:,k);
        temp_C = C(:,k);
        temp_S = S(:,k);
        temp_W = W(:,k);

        if (sum(C) < Cost_Max) || (sum(W) < Weight_Max) || (sum(S) ...
            < Size_Max) % Maximal cost level not reached ?
            C(:,k) =[];
            C = [C rand(1, 'double')*((temp_C))+2.0 (1.0-rand(1, ...
                'double'))*((temp_C))+1.0];

            T(:,k) = [];
            T = [T temp_T temp_T];

            L(:,k) = [];
            L =[L (temp_L)-rand(1, 'double')*((temp_L)/4.0) ...
                temp_L-rand(1, 'double')*((temp_L)/5.0)];

            W(:,k) =[];
            W = [W (temp_W)-rand(1, 'double')*((temp_W)/4.0) ...
                temp_W-rand(1, 'double')*((temp_W)/5.0) ];

            S(:,k) =[];
            S = [S (temp_S)-rand(1, 'double')*((temp_S)/4.0) ...
                temp_S-rand(1, 'double')*((temp_S)/5.0)];
            %Optimierung auf der Komponente Ebene...Berechnung mittels
                    %GA
            ub = L+tol;
            lb = L-tol;
            L0 = L;
            A = []; b = [];
Aeq = []; beq = [];
FitnessFunction = @multi_objectives;
numberOfVariables = length(L);
[x fval flag output population] = ...
    gamultiobj(FitnessFunction,numberOfVariables, ...
    A,b,Aeq,beq,lb,ub, options);

% Figure 1-2-3
figPareto=figure;
plot3(fval(:,1),fval(:,2),fval(:,3),'h');
title_string=sprintf('Pareto front for topology %d', k);
title(title_string);
xlabel('Costs');
ylabel('1/Reliability');
zlabel('Size function');
grid ON;
clear title_string;
filename = sprintf('development_pareto_front_size%d.eps', k);
print(figPareto,'-depsc2', filename);
```

```
close(figPareto);

% Figure 1-2-4
figPareto=figure;
plot3(fval(:,1),fval(:,2),fval(:,4),'h');
title_string=sprintf('Pareto front for topology %d', k);
title(title_string);
xlabel('Costs');
ylabel('1/Reliability');
zlabel('Weight function');
grid ON;
clear title_string;
filename = sprintf('development_pareto_front_weight%d.eps', k);
print(figPareto,'-depsc2', filename);
close(figPareto);
disp('GA finished')
    end
end
```

Source code A.1: Main generic optimization program

```
%% MAIN PROGRAMM

%% DEFINITION OF GLOABAL VARIABLES
clc, clear all, close all

global T;           % Initial topology
global L;           % Initial component reliability
global C;           % Initial costs
global Cost_Max;    % Violation level of costs (budget)
global history
global R_temp
global N_temp
global k
global Tp_hist
global Lp_hist
global TT_hist
global LL_hist

L0_hist = [];
L_hist = [];
tolL_hist = [];
tolC_hist = [];
fval_hist = [];
mflag_hist = [];
output_hist = [];
history.x = [];
history.fval = [];
R_temp = [];
N_temp = [];
```

```
%% Set the starting values
initVal;
ub_hist = zeros(1,length(L));
lb_hist = zeros(1,length(L));

%% GENERATION OF NEW TOPOLOGY AND OPTIMIZATION AT THE COMPONENT ...
    LEVEL

% Generierung der mÃ¶glichen Topologien

for k = 1:size(T, 2);

     % Selection of the ith-component
     temp_T = T(:,k);
     temp_L = L(:,k);
     temp_C = C(:,k);

     if sum(C) < Cost_Max % Maximal cost level not reached ?
        C(:,k) =[];
        C = [C randomVal*((temp_C))+5 (1.0-randomVal)*((temp_C))+5];

        T(:,k) = [];
        T = [T temp_T temp_T];

        L(:,k) = [];
        L =[L (temp_L/2.0) (temp_L/3.0)];

        tolL = calc_tol(L, randomVal, skalVal);
        tolL_hist(k) = tolL;

        %Optimierung auf der Komponente Ebene...Berechnung mittels
        %F MINCON
        ub = L+tolL;
        if size(ub_hist, 2) < length(ub) && ¬isempty(ub)
                ub_hist = [ub_hist zeros(size(ub_hist, 1),1)];
        end
        ub_hist(k,1:length(ub_hist)) = ub(1:length(ub));

        lb = L-tolL;
        if size(lb_hist, 2) < length(lb) && ¬isempty(lb)
                lb_hist = [lb_hist zeros(size(lb_hist, 1),1)];
        end
        lb_hist(k,1:length(lb_hist)) = lb(1:length(lb));
        L0 = L;
        options = ...
           optimset('outputfcn',@outfun);%,'Display','iter-detailed');
        [L fval mflag output]  = ...
           fmincon(@objfun,L0,[],[],[],[],lb,ub,@confun, options);

        % History of optimization output
```

```
            L0_hist = [L0_hist L0];
            if size(L_hist, 2) < length(L) && ¬isempty(L_hist)
                    L_hist = [L_hist zeros(size(L_hist, 1),1)];
            end
            L_hist = [L_hist; L];
            fval_hist = [fval_hist fval];
            mflag_hist = [mflag_hist mflag];
            output_hist = [output_hist output];
            clear N_temp R_temp
        end
end
clf;
fig=figure;
for i=1:size(L_hist,1)
 plot(L_hist(((i-1)*length(L)+1):length(L)),'x')
 plot(1:length(L_hist(i,:)),L_hist(i,:), 'x' ,'LineWidth',2)
   findNonZero=find(L_hist(i,:));
 plot_tol_range(1:length(L_hist(i,:)),L_hist(i,:),tolL_hist(i))
   subplot(size(L_hist,1),1,i), ...
       plot_tol_range(findNonZero,L_hist(i,findNonZero),tolL_hist(i))
   xlim([1 length(L_hist(i,:))])
   ylim([0 1.1*max(L_hist(i,findNonZero))]);
   grid on;
   hold on;
   axes('Position',[0 0 1 1],'Visible','off');
   set(gca,'XTick',1:length(L_hist(i,:)))
   if i == size(L_hist,1)
      xlabel('Iteration steps');
   end
   clear Δ
end
hold off;
text_title = sprintf('Development of L');
tx = text(0.4,0.95,text_title);
filename = sprintf('development_of_L.eps', k);
print(fig, '-depsc2', filename)
clf;
clear filename
close all;
```

Source code A.2: Main deterministic optimization program

```
clc; clear all; close all;
num_points = 100;

R_min = 0.5;
R_max = 0.9;
epsilon_min = 0.0;
epsilon_max = 0.5;
r = linspace(R_min, R_max, num_points);
```

```
epsilon = linspace(epsilon_min, epsilon_max, num_points);
[Epsilon, R] = meshgrid(epsilon, r);

c_0 = 1.0;
Costs = c_0 .* (sin(20.0 .* R + Epsilon) .* cos(R+ 30.0 .* ...
    Epsilon) - cos(40.0 .* R + 70.0 .* Epsilon) .* sin(30.0 .* R ...
    + 50.0 .* Epsilon));

s_0 = 1.0;
Size = s_0 .* (sin(20.0 .* R + Epsilon) .* cos(R+ 30.0 .* ...
    Epsilon) - cos(40.0 .* R + 70.0 .* Epsilon) .* sin(30.0 .* R ...
    + 50.0 .* Epsilon));

w_0 = 1.0;
Weight = w_0 .* (sin(20.0 .* R + Epsilon) .* cos(R+ 30.0 .* ...
    Epsilon) - cos(40.0 .* R + 70.0 .* Epsilon) .* sin(30.0 .* R ...
    + 50.0 .* Epsilon));

alpha = 0.4;
beta = 0.3;
gamma = (1.0 - alpha - beta);
Sum = alpha .* Costs + beta .* Size + gamma .* Weight;

fig = figure;
surf(Epsilon, R, Sum);
hold off;
xlabel('Improvement epsilon', 'interpreter','latex','fontsize',12);
ylabel('Reliability R', 'interpreter','latex','fontsize',12);
zlabel('Objective Function J' , ...
    'interpreter','latex','fontsize',12);
set(fig,'PaperPositionMode','auto')
filename = sprintf('sum_weighted_C_S_W.eps');
print(fig,'-depsc2', filename);
clear filename fig;
```

Source code A.3: Visualization of the objective functions

```
function f = test_objective(L)
global T;
global f_hist;

[tempT, tempL]= SRM(T,L); % tempT geht verloren!

%f(1)= ((tempL/0.9)+(0.9/tempL))*sin(tempL)+sin(tempL)
f(1) = sum(L);
f(2)= 1./tempL;
f(3) = tempL .* tempL .* tempL - tempL .* tempL;
f(4) = tempL .* sin(tempL);
%f(3)=1./sum(L);
%f_hist = [f_hist f];
```

Source code A.4: Declaration of the multiobjective functions

```
function [c, ceq] = confun(L)
  % Nonlinear inequality constraints
  c(1) = sum(L)-9;
  c(2) = prod(L(1:3))-0.5;
  c(3) = prod(L(4:6))-0.4;
  % Nonlinear equality constraints
  ceq = [];
```

Source code A.5: Constraints

```
function f = objfun (L)
global T;
global f_hist;

[tempT, tempL]= SRM (T,L); % tempT geht verloren!
% f = 1./tempL;
f = sin (tempL)*(1./tempL)+cos (tempL);
% f= ((tempL/0.9)+(0.9/tempL))*sin (tempL)+sin (tempL)
f_hist = [f_hist f];
```

Source code A.6: Objective function

```
function [Tb Lb] =  seri(TT, LL)

% T: ursprÃ¼ngliche Topologiematrix
% L: ursprÃ¼nglicher Lambdavektor
% Tb: bearbeitete Topologiematrix
% Lb: bearbeitete Lambdavektor

l = 1;  % Laufindex T_r bzw. l_r
ex = 0; % Exit-Bedingung wenn keine Parallelitaeten mehr vorliegen

Tb = zeros(size(TT,1),1);
Lb = zeros(size(LL,1));
while ex ≠ 1
     if size(TT,2) == 1      % TT and LL have only one column
         ex = 1;
         Tb = TT;
         Lb = LL;
    else
    % PARALLELISM CHECK
      if l == size(TT,2)      % last row of TT reached?
            ex = 1;
            Tb(:,l) = col(:,:);
```

```
            Lb(l) = LL(l);
            TT(:,l) = [];
            LL(l) = [];
        elseif l > size(TT,2)
            ex = 1;
        else
            %disp('not last row')
            col = TT(:,l);                  % Referenzspalte
                 pos1 = find(col == 1); %zeilenindex startknoten
                 pos2 = find(col == 2); %zeilenindex zielknoten

                 seriCheck1 = find(TT(pos2,(l+1):size(TT,2)) == ...
                    1); %RZOK
                 seriCheck2 = find(TT(pos2,:) == 2);
                 if (length(seriCheck1) < eps || ...
                    length(seriCheck1) > 1 || length(seriCheck2) ...
                    > 1 )    % More than '2' found in a row?
                     Tb(:,l) = col;
                     Lb(l) = LL(l);
                     l = l + 1;
                     continue;
                 end

                 pos21 = find(TT(pos2,:)  == 1); %spaltenindex ...
                    reihenstartknoten
                 pos22 = find(TT(:,pos21) == 2);  %zeilenindex ...
                    reihenendknoten

        end
        l = l + 1;
    end
end
```

Source code A.7: Detection of components in series

```
function [Tout Lout] =  SRM(TT,LL)
global Tp_hist
global Lp_hist
global TT_hist
global LL_hist

% TT besagt die Topologie der Struktur und LL is einen skalaren ...
   Wert-->Zuverlaessigkeit des Systems
iteration = 1; % RZOK
while size(TT,2)>1
    % parallele Reduktion
    [Tp, Lp] = para(TT, LL);
    Tp_hist = [Tp_hist Tp];
    Lp_hist = [Lp_hist Lp];
```

```
        % serielle Reduktion
       [TT, LL] = seri(Tp,Lp);
       iteration = iteration + 1; % RZOK
    end


    Tout=TT;
    Lout=LL;
```

Source code A.8: Reliability calculation using SRM

```
function [Tb Lb] =  para(TT, LL)

% T: ursprÃ¼ngliche Topologiematrix
% L: ursprÃ¼nglicher Lambdavektor
% Tb: bearbeitete Topologiematrix
% Lb: bearbeitete Lambdavektor

l = 1;  % Laufindex T_r bzw. l_r
ex = 0; % Exit-Bedingung wenn keine Parallelitaeten mehr vorliegen

while ex ≠ 1

    col = TT(:,l);                  % ReferenzspaLe
    mat = zeros(size(TT));          % Initmatrix
    [ir ic val] = find(col);        % ....

    mat(ir(1), :) = val(1);
    mat(ir(2), :) = val(2);
    ident = find(abs(sum((TT - mat).*rand(size(TT))))<eps);
    sizeofident = size(LL(ident),2);
    co = LL(ident);%content of Iden

    switch sizeofident
        case 5
            reli = sum(LL(ident)) ...
                 - co(:,1) * co(:,2) ...
                 - co(:,1) * co(:,3) ...
                 - co(:,1) * co(:,4) ...
                 - co(:,1) * co(:,5) ...
                 - co(:,2) * co(:,3) ...
                 - co(:,2) * co(:,4) ...
                 - co(:,2) * co(:,5) ...
                 - co(:,3) * co(:,4) ...
                 - co(:,3) * co(:,5) ...
                 - co(:,4) * co(:,5) ...
                 + co(:,1) * co(:,2) * co(:,3) ...
                 + co(:,1) * co(:,2) * co(:,4) ...
                 + co(:,1) * co(:,2) * co(:,5) ...
                 + co(:,1) * co(:,3) * co(:,4) ...
```

```
                    + co(:,1) * co(:,3) * co(:,5) ...
                    + co(:,1) * co(:,4) * co(:,5) ...
                    + co(:,2) * co(:,3) * co(:,4) ...
                    + co(:,2) * co(:,3) * co(:,5) ...
                    + co(:,3) * co(:,4) * co(:,5) ...
                    - co(:,1) * co(:,2) * co(:,3) * co(:,4) ...
                    - co(:,1) * co(:,2) * co(:,3) * co(:,5) ...
                    - co(:,2) * co(:,3) * co(:,4) * co(:,5) ...
                    - co(:,1) * co(:,3) * co(:,4) * co(:,5) ...
                    - co(:,1) * co(:,2) * co(:,4) * co(:,5) ...
                    - co(:,1) * co(:,2) * co(:,3) * co(:,5) ...
                    + co(:,1) * co(:,2) * co(:,3) * co(:,4) * co(:,5);
        case 4
            reli = sum(LL(ident)) ...
                     - co(:,1) * co(:,2) ...
                     - co(:,1) * co(:,3) ...
                     - co(:,3) * co(:,2) ...
                     - co(:,1) * co(:,4) ...
                     - co(:,2) * co(:,4) ...
                     - co(:,3) * co(:,4) ...
                     + co(:,1) * co(:,2) * co(:,3) ...
                     + co(:,1) * co(:,2) * co(:,4) ...
                     + co(:,1) * co(:,3) * co(:,4) ...
                     + co(:,2) * co(:,3) * co(:,4) ...
                     - co(:,1) * co(:,2) * co(:,3) * co(:,4);
        case 3
            reli = sum(LL(ident)) ...
                     - co(:,1) * co(:,2) ...
                     - co(:,1) * co(:,3) ...
                     - co(:,2) * co(:,3) ...
                     + co(:,1) * co(:,2) * co(:,3);
        case 2
            reli = sum(LL(ident)) ...
                      - co(:,1) * co(:,2);
        case 1
            reli = sum(LL(ident));
        otherwise
    end
    Tb(:,l) =  col;
    Lb(l) = reli;
    LL(l) = reli;

    del_ident = ident(2:length(ident));
    TT(:,del_ident) = [];
    LL(del_ident) = [];
    if size(TT,2) == 1 || l == size(TT,2)
        ex = 1;
    end
    l = l + 1;

end
```

Source code A.9: Detection of components in parallel