Technische Universität München
Ingenieurfakultät Bau Geo Umwelt
Professur für Computational Mechanics

Optimizing Flexibility for Component Design in Systems Engineering under Epistemic Uncertainty

Marco Daub

Vollständiger Abdruck der von der Ingenieurfakultät Bau Geo Umwelt der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Kai-Uwe Bletzinger
Prüfende der Dissertation:	1. Prof. Dr.-Ing. habil. Fabian Duddeck
	2. Prof. Dr. ir. David Moens
	3. apl. Prof. Dr.-Ing. habil. Michael Hanss

Die Dissertation wurde am 18.02.2020 bei der Technischen Universität München eingereicht und durch die Ingenieurfakultät Bau Geo Umwelt am 29.06.2020 angenommen.

ABSTRACT

English

In systems engineering, complexity can be reduced by a decomposition of the design process. This includes the decoupling of design decisions for which this thesis presents a new framework. Innovative methods that take both a system's hierarchical structure and epistemic uncertainty in the early design phase into account are proposed. These methods provide optimal flexibility for decision making at the component level and yield robustness at the system level. The effectiveness of the overall approach is demonstrated via test-bed problems for crashworthiness.

German

Im Systems Engineering kann Komplexität durch eine Zerlegung des Entwicklungsprozesses reduziert werden. Dies beinhaltet die Entkopplung von Entscheidungen hinsichtlich der Entwicklung, für welche diese Doktorarbeit einen neuen Ansatz vorstellt. Es werden innovative Methoden vorgeschlagen, die sowohl die hierarchische Struktur eines Systems als auch epistemische Unschärfe in der frühen Entwicklungsphase berücksichtigen. Diese Methoden gewähren optimale Flexibilität für den Entscheidungsprozess auf Komponentenebene und erzielen Robustheit auf Systemebene. Die Wirksamkeit des Gesamtansatzes wird an Testproblemen für Crashsicherheit demonstriert.

NOTATIONS

The following list provides an overview of the major symbols used in this thesis. Symbols that are mathematical conventions are not included here:

A	transformation matrix for linear f
A^k	transformation matrix for linear g^k
a_{c}	critical acceleration threshold
a_j	row of A
a_j^k	row of A^k
b	system performance threshold for linear f
b_j	entry of b
b_j^k	component performance threshold for linear g^k
c	system cost function
c^k	component cost function
d	number of design variables of x
d^k	number of design variables of x^k
$E_{\mathrm{cf}}^{\mathrm{add}}$	energy correction factor of the additional load path
E_{cf}^k	energy correction factor of a component
E_{tot}^k	total internal energy of a component
f	vector of all f_j
F	vector of all F_j, or force
$F^{\mathrm{add}}(s)$	plastic deformation force of the additional load path at s
$F^k(s)$	plastic deformation force of a component at s
$F^{\mathrm{l},k}$	vector of all $F_i^{\mathrm{l},k}$
$F^{\mathrm{u},k}$	vector of all $F_i^{\mathrm{u},k}$
$F_{\mathrm{b},\theta}^{\mathrm{l},k}$	vector of all lower bounds of an intermediate box
$F_{\mathrm{b},\theta}^{\mathrm{u},k}$	vector of all upper bounds of an intermediate box
f_{c}	vector of all $f_{\mathrm{c},j}$
$f_{\mathrm{c},j}$	system performance threshold
$F_{\mathrm{ds}}^{\mathrm{l},k}$	minimum plastic deformation force of a component
$F_{\mathrm{ds}}^{\mathrm{u},k}$	maximum plastic deformation force of a component
F_i	design variable for constant force level
$\check{F}_i$	target design variable for constant force level
F_i^k	designable constant force level of a component
$F_i^{\mathrm{l},k}$	lower component solution space bound of F_i^k
$F_i^{\mathrm{u},k}$	upper component solution space bound of F_i^k
$F_{\mathrm{ib}}^{\mathrm{l},k}$	vector of all lower bounds of maximum volume inner box
$F_{\mathrm{ib}}^{\mathrm{u},k}$	vector of all upper bounds of maximum volume inner box
f_j	system performance function
F_j	constraint function

$F_{\mathrm{ob}}^{\mathrm{l},k}$	vector of all lower bounds of minimum volume outer box
$F_{\mathrm{ob}}^{\mathrm{u},k}$	vector of all upper bounds of minimum volume outer box
g^k	vector of all g_j^k
$g_{\mathrm{c}}^k s$	vector of all $g_{\mathrm{c},j}^k$
$g_{\mathrm{c},j}^k$	component performance threshold
g_j^k	component performance function
$\bar{L}$	average edge length
m	number of z_j
$m^*(s)$	active mass at s
m^k	number of y_j, or mass at the end of a component
m_l	discrete mass
n	number of components
n_{lp}	number of load paths
n_{hl}	number of hierarchical system levels
n_{m}	number of m_l
n_{s}	number of sections
p	vector of all p_l
$\check{p}$	vector of all $\check{p}_l$
P	fuzzy uncertainty set of p
p_l	uncontrollable parameter
$\check{p}_l$	nominal value of p_l
q	number of p_l
$r_{\mathrm{b},\theta}^k$	metrics for $[F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{u},k}]$
$r_{\mathrm{CSS},\theta}^k$	metrics for $[F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{u},k}]$
s	deformation position
$\bar{s}^k$	deformation length of a component
s_0	start of deformation of the vehicle
s_0^k	start of deformation of a component
s_{end}	end of deformation of the vehicle
s_{end}^k	end of deformation of a component
s_i	start and end of sections
$\bar{s}_i$	section length
t_{CPU}	CPU time
v_0	initial velocity
V	Volume
x	system design
$\check{x}$	system target design
$X(\check{x})$	fuzzy uncertainty set of x with respect to $\check{x}$
x^k	component design
$\check{x}^k$	component target design
x^{l}	vector of all x_i^{l}
$x^{\mathrm{l},k}$	vector of all $x_i^{\mathrm{l},k}$

x^{u}	vector of all x_i^{u}
$x^{\mathrm{u},k}$	vector of all $x_i^{\mathrm{l},k}$
$x^{\mathrm{r},k}$	vector of all component designs but x^k
$x^k_{\mathrm{cp},\iota}$	corner point of $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$
$x^{\mathrm{l}}_{\mathrm{ds}}$	vector of all $x^{\mathrm{l}}_{\mathrm{ds},i}$
$x^{\mathrm{u}}_{\mathrm{ds}}$	vector of all$x^{\mathrm{u}}_{\mathrm{ds},i}$
$x^{\mathrm{l},k}_{\mathrm{ds}}$	vector of all $x^{\mathrm{l},k}_{\mathrm{ds},i}$
$x^{\mathrm{u},k}_{\mathrm{ds}}$	vector of all $x^{\mathrm{u},k}_{\mathrm{ds},i}$
$x^{\mathrm{l}}_{\mathrm{ds},i}$	lower design space bound of x_i
$x^{\mathrm{u}}_{\mathrm{ds},i}$	upper design space bound of x_i
$x^{\mathrm{l},k}_{\mathrm{ds},i}$	lower design space bound of x_i^k
$x^{\mathrm{u},k}_{\mathrm{ds},i}$	upper design space bound of x_i^k
x_i	design variable
$\check{x}_i$	target design variable
$X_i(\check{x}_i)$	fuzzy uncertainty set of x_i with respect to $\check{x}_i$
x_i^k	entry of x^k
x_i^{l}	lower system solution space bound of x_i
x_i^{u}	upper system solution space bound of x_i
$x_i^{\mathrm{l},k}$	lower component solution space bound of x_i^k
$x_i^{\mathrm{u},k}$	upper component solution space bound of x_i^k
y	vector of all y_j
y_j	component response
z	vector of all z_j
z_j	system response
α	degree of membership
γ	vector of all γ_l
γ_l	uncertainty magnitude of $\check{p}_l$ for interval-type uncertainty
γ_α	vector of all $\gamma_{\alpha,l}$
$\gamma_{\alpha,l}$	uncertainty magnitude of $\check{p}_l$ for fuzzy-type uncertainty
δ	vector of all δ_i
$\bar{\delta}$	vector of all $\bar{\delta}_i$
δ'	optimization quantity for maximized uncertainty magnitudes
δ^k	vector of all δ_i^k
$\bar{\delta}^k$	vector of all upper estimates of δ_i^k
δ_i	uncertainty magnitude of $\check{x}_i$ for interval-type uncertainty
$\bar{\delta}_i$	upper estimate of δ_i
δ_i^k	uncertainty magnitude of $\check{x}_i^k$ for interval-type uncertainty
δ_α	vector of all $\delta_{\alpha,i}$
$\delta_{\alpha,i}$	uncertainty magnitude of $\check{x}_i$ for fuzzy-type uncertainty
ε	perturbation parameter
μ^P	membership function of P

$\mu^{X(\check{x})}$	membership function of $X(\check{x})$
$\mu^{X_i(\check{x}_i)}$	membership function of $X_i(\check{x}_i)$
ν	vector of all ν_i
ν_i	constant weight for maximized uncertainty magnitudes
ω	vector of all ω_i
ω_i	weighting factor for maximized uncertainty magnitudes
Ω	system solution space
Ω^k	component solution space
Ω_{c}	complete system solution space
$\Omega_{\mathrm{c,bc}}$	best-case complete system solution space
$\check{\Omega}_{\mathrm{c,bc}}$	best-case complete system solution space of target designs
$\Omega_{\mathrm{c,nec},\alpha}$	necessity-α complete system solution space
$\check{\Omega}_{\mathrm{c,nec},\alpha}$	necessity-α complete system solution space of target designs
$\Omega_{\mathrm{c,pos},\alpha}$	possibility-α complete system solution space
$\check{\Omega}_{\mathrm{c,pos},\alpha}$	possibility-α complete system solution space of target designs
$\Omega_{\mathrm{c,wc}}$	worst-case complete system solution space
$\check{\Omega}_{\mathrm{c,wc}}$	worst-case complete system solution space of target designs
Ω_{ds}	system design space
Ω^k_{ds}	component design space
Ω^k_{bc}	best-case component solution space
$\check{\Omega}^k_{\mathrm{bc}}$	best-case component solution space of target designs
$\Omega^k_{\mathrm{nec},\alpha}$	necessity-α component solution space
$\check{\Omega}^k_{\mathrm{nec},\alpha}$	necessity-α component solution space of target designs
$\Omega^k_{\mathrm{poc},\alpha}$	possibility-α component solution space
$\check{\Omega}^k_{\mathrm{poc},\alpha}$	possibility-α component solution space of target designs
Ω^k_{wc}	worst-case component solution space
$\check{\Omega}^k_{\mathrm{wc}}$	worst-case component solution space of target designs

$\mathcal{U}^P(\check{p},\gamma)$	uncertainty set of p
$\mathcal{U}^P_\alpha(\check{p},\gamma_{\alpha=0},\gamma_{\alpha=1})$	α-cut of P
$\mathcal{U}^P_0(\check{p},\gamma_{\alpha=0},\gamma_{\alpha=1})$	support of P
$\mathcal{U}^X(\check{x},\delta)$	uncertainty set of x with respect to $\check{x}$
$\mathcal{U}^X_\alpha(\check{x},\delta_{\alpha=0},\delta_{\alpha=1})$	α-cut of X with respect to $\check{x}$
$\mathcal{U}^X_0(\check{x},\delta_{\alpha=0},\delta_{\alpha=1})$	support of X with respect to $\check{x}$

$\mathrm{vol}(\Omega)$	volume of Ω
$\mathrm{proj}^k(\Omega_{\mathrm{c}})$	projection of Ω_{c} onto the coordinate space of x^k
$\mathrm{proj}_i(\Omega_{\mathrm{c}})$	projection of Ω_{c} onto the coordinate space of x_i

CONTENTS

1. INTRODUCTION

Systems engineering deals with the development and management of complex products, i.e., it guides their engineering, cf. [77]. Elementary in systems engineering is its perspective on systems thinking, see, e.g., [60, 130] for more information. In systems thinking, a system is considered as a set of components that operate together. Based on inputs, the system including its components perform tasks that are reflected in outputs. Every component has its specific properties and functions. Furthermore, the single components are interconnected by relations and can be seen as systems themselves. This leads to a hierarchical system structure. In this manner, even the original system can be viewed as a component of a larger system. The concept of a system can be applied to many engineering objects like vehicles or airplanes. Even smaller objects such as printers or screwdrivers are conceivable as systems. Further examples can be found in [127].

Because of their general complexity, only abstractions of systems can be analyzed in practice, see [106]. The point of view on a system depends on the observer and on the type of analysis that is conducted. Often, only a particular behavior of the system is of interest, like the performance of a vehicle in a crash scenario. This motivates a model-based framework for system engineering. Here, models are used as simplifications of real physical systems to represent particular aspects of them. In systems engineering, a distinction is made, for example, between physical and mathematical models, see [77]. Physical models represent the physical characteristics of systems and are usually physical constructs, e.g., scale models. In contrast, mathematical models use mathematical equations to describe systems. Fundamental mathematical models are based on engineering science principles and depict the input-output relations of systems.

The discipline of systems engineering that concerns with designing the system is also referred to as systems design. By assuming the structure of a system including its relations as given, the *design* of this system can be defined by characteristic quantities like geometric or material properties. Once values are assigned to these quantities, the design is modeled mathematically. The values can then be integrated as design variables into the equations describing specific performances of the system. This yields a design model, see [106] for more information. Design alternatives are represented by different values of the design variables. Using a design model, predictions about the performances of these alternatives can be made. Here, the design variables serve as controllable inputs for the design models that might be complemented by uncontrollable inputs mapping on the outputs of the system, see [107]. Clearly, design models are not unique. They are problem-oriented and depend on a designer's view on the system, including which design variables he wants to consider.

For a set of design alternatives that fulfill the essential system requirements, a *decision* is required in which one of the designs is selected. Hence, designing can be seen as a decision-making process, cf. [106]. In order to select an optimal design, the design alternatives must

be evaluated and ranked regarding an objective such as minimum costs. Again, this objective is subjective and may change over the design process.

There are various approaches that guide the engineering of a system yielding its design, see [60] for an overview. Some of them propose the decomposition of the design process, which allows different design teams to work on different components. An example is the V-model approach that decomposes system requirements into component requirements, enabling a decoupled development of the components in the early design phase. In doing so, decision making regarding the design becomes more flexible and agile. Here, the single components are modifiable to fulfill particular needs, which occur, for example, during their development. By breaking down complexity, the overall design process can also be simplified and sped up, cf. [108]. However, process models like the V-model are usually hypothetical and specific instructions on how to apply them, for instance, to mathematical system models are lacking, see [35].
In general, decomposition to yield a design is applied in different areas. In [80], it is differentiated between product, problem, and process decomposition. Approaches that involve mathematical system models usually belong to the category of problem decomposition. This category can be further subdivided into requirement decomposition, constraint-parameter decomposition, and decomposition-based design optimization, see [80] for more details and an overview of available methods. Requirement decomposition, see [81], and constraint-parameter decomposition, see [82], aim at decomposing a design problem into subproblems by not assigning values to the design variables. This allows designers to select their preferred procedure to solve the subproblems and provides *flexibility* for design decisions. In contrast, in decomposition-based design optimization, a system problem is broken into subproblems, which are then solved separately and coordinated to suit the original problem. Approaches that provide a hierarchical system decomposition are proposed, for example, in [79], and methods that coordinate the solutions of the subproblems, for example, in [2]. A further approach worth mentioning is analytical target cascading, cf. [75]. Here, system targets are cascaded iteratively to the subsystems, which are then designed to fulfill these targets. Theoretically, all methods are capable of finding an optimal design for the original system, i.e., to solve the original problem. Nevertheless, iterations are always required to coordinate the subproblem solution for the original system. This even holds for the mentioned approaches from requirement decomposition and constraint-parameter decomposition.
A method that abandons to find the most optimal design in favor of removing iterations for integrating the subproblem solutions is proposed in [140]. It can be affiliated with the category of constraint-parameter decomposition and computes permissible intervals for all design variables. Afterward, designers can select the values of the design variables independently within these intervals. In doing so, flexibility in selecting design variables for a system design that fulfills all requirements can be guaranteed, assuming the mathematical model is accurate enough. As the intervals are maximized with respect to an optimality criterion here, this approach provides optimal flexibility for design decisions which are decoupled and independent

between all design variables. Furthermore, the intervals provide an easy visualization of design alternatives, and uncertainties in the controllable design variables can be circumvented best if the designers target their centers. However, due to the complete decoupling, the intervals might become very small and exclude a lot of permissible designs, see [141]. This limits the flexibility for designing. First approaches to improve this drawback, by applying different decoupling schemes, are proposed in [38, 129].

Although a decomposition of the design process usually relates to the early phase in which uncertainty is omnipresent, cf. [48], uncertainty is not treated at all or only deficiently in the decomposition approaches. Hence, a design may deteriorate, for example, because of the use of an inaccurate model and violate the system requirements. That causes delays in the design process. In general, uncertainty in the early design phase can be characterized as *epistemic uncertainty* due to lack of knowledge. Approaches to model this type of uncertainty use, for example, interval analysis or possibility theory, see [124]. In contrast, aleatoric uncertainty caused by irreducible, intrinsic randomness is commonly modeled with probability distributions. Further ways to classify uncertainty that occur during the design process of a system can be found, for example, in [11, 94, 114, 126].
For different types of uncertainty, a survey of various approaches yielding an optimal design for the system can be found, e.g., in [10, 11, 49]. As the design becomes optimal under uncertainty, it is also called a robust design. However, a decomposition of the design process is not considered in these approaches, which is yet another drawback for the early phase of systems design. Still, a method for robust design proposed by [66] has similarities with the method of [140] in which permissible intervals for all design variables are computed. It maximizes the distance from a robust design to all alternative designs that violate the system requirements. Therefore, it can tolerate maximum uncertainties in the controllable design variables. Using the maximum metric for distance computation, a maximum box is spanned around the center comprising also intervals for the design variables. This motivates targeting the center of these intervals to circumvent negative effects of uncertainties in design variables.

Summarized, efficient methods that decompose the design process under appropriate consideration of uncertainty are lacking in the literature. Hence, the scope of this thesis is to provide an applicable methodology for decoupled design decisions in systems engineering under epistemic uncertainty. The methodology shall

- deploy a general framework to classify decoupled design decisions,
- provide flexibility for decoupled design decisions,
- decouple the decisions based on the hierarchical structure of the system,
- incorporate the treatment of epistemic uncertainty,

- provide algorithms for the numerical computation of the methods,
- be applicable to realistic problems in systems engineering.

In this regard, a framework based on [104], which differentiates between independent-decoupled decisions and dependent-decoupled decisions, is used and the existing approaches of [38, 129, 140] are embedded. As these approaches do not decouple decisions according to the structure of a system, the framework is geared towards decoupled design decisions for components. New methods are proposed that compute component solution spaces, which provide optimal flexibility for decisions regarding component designs. In order to circumvent also epistemic uncertainties in controllable variables and uncontrollable parameters in the early design phase, they are integrated into the methodology as interval- and fuzzy-type uncertainty. Here, fuzzy-type uncertainty is treated in the framework of possibility-theory, cf. [31]. This yields component solution spaces which account for different realizations of these uncertainties. To compute them, simplifications of the underlying problem statements are considered for specific performance functions, and required numerical tools are investigated. Similar to the method of [140], the new methodology is applied to crash design problems proposed by [43]. Furthermore, the existing crash models are enhanced to provide better results for the early design phase and a MATLAB app, comprising the complete methodology for crash design, is introduced. Note that even though the methodology is intended for the early phase of systems design, it can be transferred to further problems in systems engineering or different areas with the same problem structure.

Outline

This thesis is organized as follows:

Chapter 2 gives an introduction to systems design using mathematical models. Furthermore, a framework for design decisions is proposed, considering coupled and decoupled decisions, and uncertainties in systems design are classified. After discussing interval- and fuzz-type uncertainty as epistemic uncertainty models, existing approaches for coupled design decisions under these uncertainty types are reviewed and complemented. Subsequently, existing approaches for decoupled design decisions are integrated into the proposed framework, and research gaps in both the methodology and treatment of uncertainty are identified.

Chapter 3 proposes a new methodology for decoupled design decisions that is geared towards the component structure of a system. Therefore, the hierarchical levels of a system are investigated in more detail and it is described how flexibility for component design can be provided. For optimal flexibility, optimal component solution spaces are proposed for which different problem statements for an independent and dependent decoupling are considered. Moreover, the methodology is extended by integrating interval- and fuzzy type uncertainties in controllable variables and uncontrollable parameters into the framework.

Chapter 4 provides mathematical basics that are necessary to compute component solution spaces within the proposed framework. First, properties of the underlying problem statements are derived in order to check if an optimal solution exists. Then, they are simplified for specific system performance functions to enable their numerical computation. To further support this computation, useful numerical tools are provided.

Chapter 5 applies the methodology to crash design. Therefore, existing crash design models are reviewed and enhanced, and two test-bed problems are proposed. Using these problems, the different approaches of this thesis to compute component solution spaces are considered and compared. For this, epistemic uncertainty is taken into account as well. In addition, a MATLAB app is introduced that computes component solution spaces for enhanced crash design models. With this app, these crash design models can be built, too.

Chapter 6 discusses the results of this thesis for providing optimal flexibility for component design under epistemic uncertainty. Furthermore, recommendations for further research are suggested and the major results of this thesis are summarized.

2. BASICS: Design Decisions in Systems Engineering

This chapter discusses decision making in systems engineering using mathematical models. First, the basic ideas, which include the classification of design decisions and uncertainties in systems design, are investigated. Then, epistemic uncertainty models considering interval- and fuzzy-type uncertainty are reviewed and existing approaches for coupled and decoupled design decisions under these uncertainty types are considered. Furthermore, research gaps in the methodology and the treatment of uncertainty for decoupled design decisions are identified.

2.1. Systems Design

The first section gives an introduction to systems design with mathematical design models. Here, a general type of mathematical design model which incorporates continuous design variables is defined and is used throughout this thesis. Based on this type, a framework for different design decisions is proposed and occurring uncertainties are classified.

2.1.1. Design Model Definitions

Here, a vector $x \in \mathbb{R}^d$ is called a *system design*. It quantifies the design of the system using continuous variables in real coordinate space. More precisely, it consists of d independent *design variables* $x_i \in \mathbb{R}$, $i = 1, \dots, d$, i.e., $x = (x_1, \dots, x_d)$. Typical design variables include, for example, numerical representations of geometric or material properties. Their values can be selected by one or multiple designers, which can be viewed as decision-makers in this regard. Thus, the design variables are also called *controllable variables*.
Besides the controllable variables, there are q *uncontrollable parameters* $p_l \in \mathbb{R}$, $l = 1, \dots, q$, which are also assumed to be continuous. These cannot be controlled by designers in general, or at least not in the considered design phase. It holds $p \in \mathbb{R}^q$ with $p = (p_1, \dots, p_q)$. Note that the assignment of uncontrollable parameters and controllable variables to system quantities is not uniquely determined and depends on the considered design problem, see Chapter 1. Usually, the controllable variables and uncontrollable parameters are also restricted to subsets of $\mathbb{R}$.
In addition, there are m *system responses* $z_j \in \mathbb{R}$, $j = 1, \dots, m$, which are assumed to be uniquely determined for a given design model by the design variables and the uncontrollable parameters. It is $z \in \mathbb{R}^m$ with $z = (z_1, \dots, z_m)$. Similar to the controllable variables and uncontrollable parameters, the system responses might be also restricted to subsets of $\mathbb{R}$. The functions that map the design variables and the uncontrollable parameters to the responses are called *system performance functions* and are denoted by f_j, $j = 1, \dots, m$, and

$f = (f_1, \ldots, f_m)$. It holds

$$f : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^m,\ (x, p) \mapsto z = f(x, p). \tag{2.1}$$

This relationship is visualized in Figure 1. In this thesis, only system performance functions that consist of known analytic equations, i.e., white-box functions are considered. Alternatively, e.g., black-box functions or differential equations can be taken into account, too. If uncontrollable parameters are not considered explicitly in a problem statement, the corresponding argument in the system performance functions is often neglected. Thus, it is written $z = f(x)$, i.e., $f = f(\cdot, p)$ for $p \in \mathbb{R}^q$.

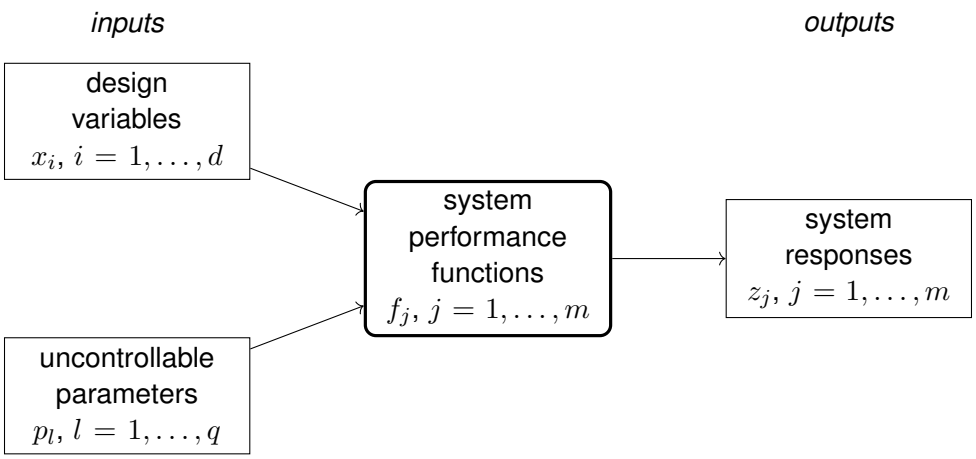


Figure 1 Diagram of a system design model.

In systems design, several constraints must be fulfilled. In this thesis, constraints on (a) design variables and (b) responses are considered:

(a) *Constraints on design variables*: Here, the design variables are bounded by lower bounds $x^{\mathrm{l}}_{\mathrm{ds},i} \in \mathbb{R}$ and upper bounds $x^{\mathrm{u}}_{\mathrm{ds},i} \in \mathbb{R}$, i.e., $x^{\mathrm{l}}_{\mathrm{ds},i} \leq x_i \leq x^{\mathrm{u}}_{\mathrm{ds},i}$ or $x_i \in [x^{\mathrm{l}}_{\mathrm{ds},i}, x^{\mathrm{u}}_{\mathrm{ds},i}]$, $i = 1, \ldots, d$, where the subscript ds stands for design space, see below. Note that this assumes that the bounds are independent between the single variables, which may not always be the case in general. In vector notation, it is $x^{\mathrm{l}}_{\mathrm{ds}} = (x^{\mathrm{l}}_{\mathrm{ds},1}, \ldots, x^{\mathrm{l}}_{\mathrm{ds},d})$, $x^{\mathrm{u}}_{\mathrm{ds}} = (x^{\mathrm{u}}_{\mathrm{ds},1}, \ldots, x^{\mathrm{u}}_{\mathrm{ds},d})$ with $x^{\mathrm{l}}_{\mathrm{ds}} \leq x \leq x^{\mathrm{u}}_{\mathrm{ds}}$ component-wise or $x \in [x^{\mathrm{l}}_{\mathrm{ds}}, x^{\mathrm{u}}_{\mathrm{ds}}]$. The d-dimensional interval $[x^{\mathrm{l}}_{\mathrm{ds}}, x^{\mathrm{u}}_{\mathrm{ds}}]$ is also called *system design space* Ω_{ds}, i.e.,

$$\Omega_{\mathrm{ds}} \quad = \quad [x^{\mathrm{l}}_{\mathrm{ds}}, x^{\mathrm{u}}_{\mathrm{ds}}] \tag{2.2}$$

or $\Omega_{\mathrm{ds}} = [x^{\mathrm{l}}_{\mathrm{ds},1}, x^{\mathrm{u}}_{\mathrm{ds},1}] \times \cdots \times [x^{\mathrm{l}}_{\mathrm{ds},d}, x^{\mathrm{u}}_{\mathrm{ds},d}]$. The general definition of a d-dimensional interval is provided in Section A.1.

(b) *Constraints on system responses*: The responses of the system must not exceed given *system performance thresholds*, which are denoted by $f_{\mathrm{c},j}(p)$ if they depend on the

uncontrollable parameters p_l, $l = 1, \ldots, q$, and by $f_{\mathrm{c},j}$ otherwise, $j = 1, \ldots, m$. In vector notation, it is $f_{\mathrm{c}}(p) = (f_{\mathrm{c},1}(p), \ldots, f_{\mathrm{c},m}(p))$ or $f_{\mathrm{c}} = (f_{\mathrm{c},1}, \ldots, f_{\mathrm{c},m})$. Furthermore, the argument p is also neglected if uncontrollable parameters are not considered explicitly in a problem statement and it is written $f_{\mathrm{c}} = f_{\mathrm{c}}(p)$ for $p \in \mathbb{R}^q$. Assuming a dependency on $p \in \mathbb{R}^q$, the constraints for $x \in \mathbb{R}^d$ can be expressed mathematically as $f_j(x, p) \leq f_{\mathrm{c},j}(p)$, $j = 1, \ldots, m$, or in vector notation as

$$f(x, p) \leq f_{\mathrm{c}}(p) \tag{2.3}$$

which must hold component-wise. These inequalities are also called *system performance requirements*. Here, only upper thresholds are considered as any lower threshold could be transformed into an upper one by multiplication of both the threshold and the performance function with -1.

A design that satisfies both constraint types is defined as *permissible*, otherwise, it is said to be *non-permissible*. The set of all permissible system designs is named *complete system solution space* Ω_{c} in [140] with

$$\Omega_{\mathrm{c}} = \{x \in \Omega_{\mathrm{ds}} \mid f(x, p) \leq f_{\mathrm{c}}(p)\}, \tag{2.4}$$

$\Omega_{\mathrm{c}} \subset \mathbb{R}^d$. In [95], Ω_{c} is also called feasible solution set, and in [57], permissible design space. A two- and a three-dimensional example of a complete system solution space are illustrated below. They are used for further investigations in the following.

Example 1. Given the system design space $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$ and a system performance function $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R},\ (x, p) \mapsto p_1 x_1 + p_2 x_2$. The uncontrollable parameters are $p_1 = 1$ and $p_2 = 2$, i.e., $f(x) = x_1 + 2x_2$ for $x \in \mathbb{R}^2$. Furthermore, a system performance threshold is given by $f_{\mathrm{c}} = 2$. Thus, the corresponding complete system solution space is

$$\Omega_{\mathrm{c}} = \{x \in [0, 2] \times [0, 1.5] \mid x_1 + 2x_2 \leq 2\}. \tag{2.5}$$

It is visualized in Figure 2.

Example 2. Given the system design space $\Omega_{\mathrm{ds}} = [0, 1.5]^3$ and a performance function $f : \mathbb{R}^3 \to \mathbb{R},\ x \mapsto -3x_1 - 2x_2 - 3x_3$ with threshold $f_{\mathrm{c}} = -9$. The corresponding complete system solution space is

$$\Omega_{\mathrm{c}} = \{x \in [0, 1.5]^3 \mid -3x_1 - 2x_2 - 3x_3 \leq -9\}. \tag{2.6}$$

It is visualized in Figure 3.

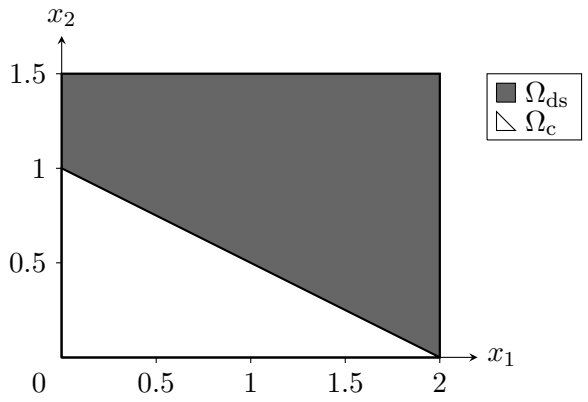


Figure 2 Two-dimensional complete system solution space Ω_{c} and system design space Ω_{ds} of Example 1. Note that their geometric shapes overlap due to their definitions. Similar findings can be observed in the following figures.

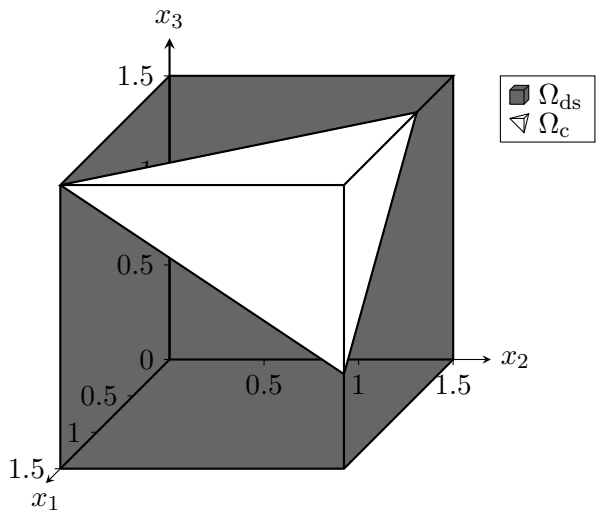


Figure 3 Three-dimensional complete system solution space Ω_{c} and system design space Ω_{ds} of Example 2.

2.1.2. Design Decisions Classification

In systems design, a permissible design is required. Therefore, decisions regarding the design variables are necessary, i.e., there must be decisions which values of the design variables x_i, $i = 1, \ldots, d$, shall be selected to obtain $x \in \Omega_{\mathrm{c}}$. As Ω_{c} contains all permissible system designs, i.e., design alternatives, it provides *flexibility* for system design decisions. These decisions are made by up to d designers. In general, there are several approaches to obtain a permissible system design. Subsequently, a categorization of possible approaches which uses perceptions of [36, 104] is proposed.

When selecting the value of one design variable, the values of the other variables must be usually considered, too, in order to obtain a permissible system design. This two-way flow of information implicates *coupled design decisions*. For coupled design decisions incorporating d designers, the designers must interact until a permissible system design $x \in \Omega_{\mathrm{c}}$ is found. This idea is visualized in Figure 4. Moreover, it is also possible that less than d designers

are involved in coupled design decisions if designers are allowed to select the values of multiple design variables. The extreme case is only one designer who selects the values of all design variables of the system together. He is then referred to as a system designer. This is also visualized in Figure 4. Still, a system designer must select the value of each design variable depending on the values of the other variables for a permissible system design $x \in \Omega_{\mathrm{c}}$. Representatively, one designer is assumed for coupled design decisions in the following.

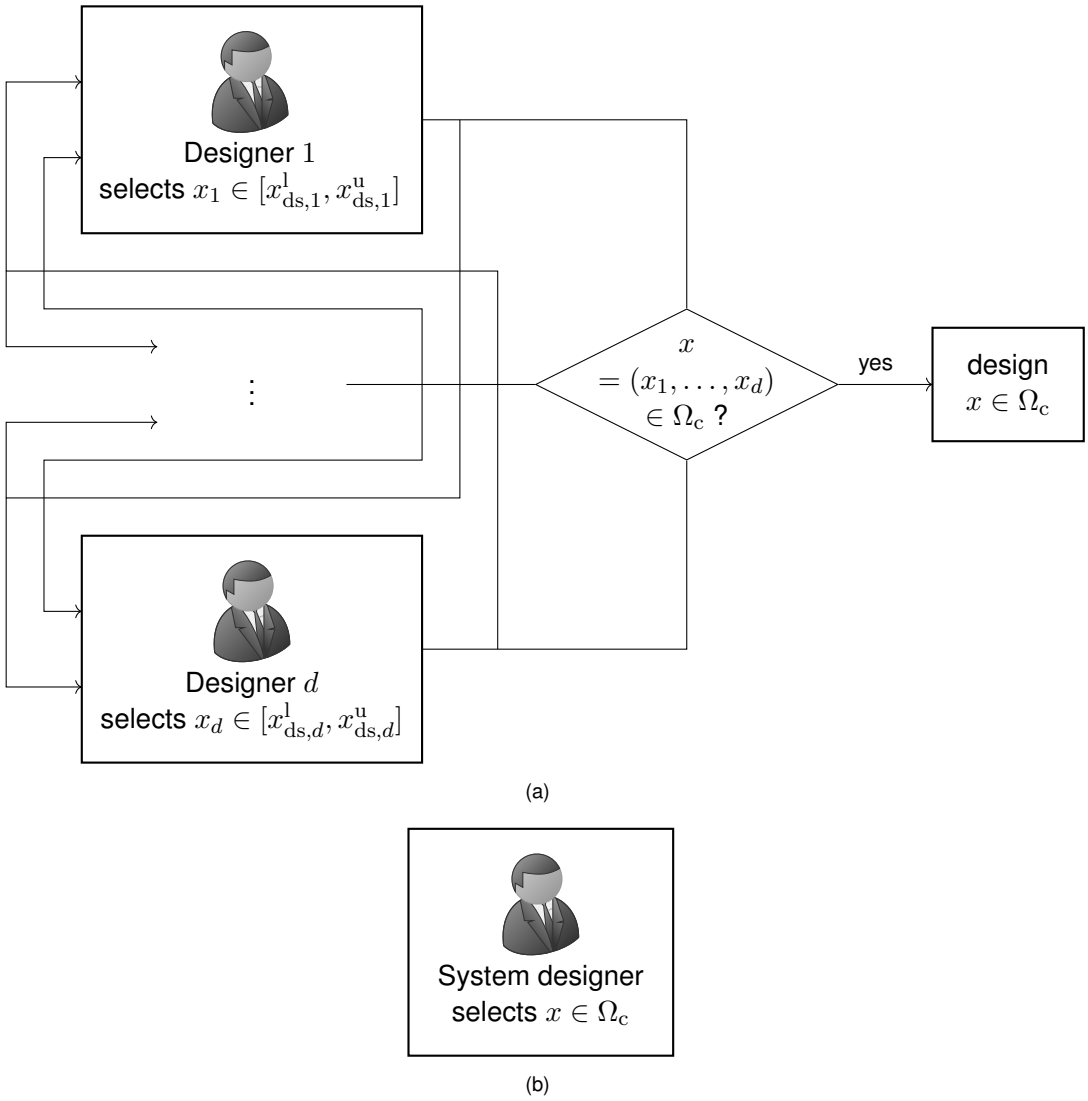


Figure 4 Coupled design decisions made by (a) d designers and (b) one system designer.

In contrast to coupled design decisions for the single design variables, there are *decoupled design decisions*. In this case, there is not only one overall decision for a system design but d decisions resulting from one decision for each design variable x_i, $i = 1, \ldots, d$. Representatively, these decisions are made by d designers, each responsible for selecting the values of a single design variable. Altogether, this procedure also results in a system design $x = (x_1, \ldots, x_d)$.

In order to provide flexibility for decoupled design decisions, each design variable value must

be selected individually from a corresponding one-dimensional set. For reasons of simplicity, these sets are assumed to be intervals which are defined by lower bounds $x_i^{\mathrm{l}} \in \mathbb{R}$ and upper bounds $x_i^{\mathrm{u}} \in \mathbb{R}$ with $x_i^{\mathrm{l}} \leq x_i^{\mathrm{u}}$, i.e., $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, holds. Hence, designer i selects $x_i \in [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$. Given the complete system solution space Ω_{c}, it is a priori not clear, how the bounds of the intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, must be defined to guarantee $x \in \Omega_{\mathrm{c}}$. This is addressed in Section 2.4.

In order to allow *independent-decoupled design decisions*, i.e., to select $x_i \in [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ for $i \in \{1, \ldots, d\}$ without taking other design decisions into account, the intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, must be independent of each other. Thus, there is no flow of information between the single design decisions. This can be understood as a concurrent engineering approach. The idea is visualized in Figure 5.

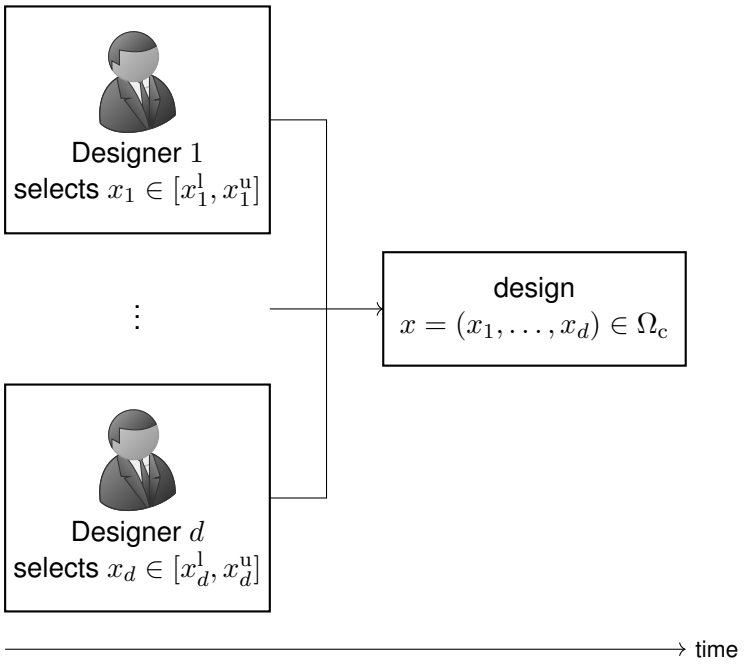


Figure 5 Independent-decoupled design decisions by d designers.

Classified between coupled design decisions and independent-decoupled design decisions, there are *dependent-decoupled design decisions* which correspond to a one-way flow of information. Here, the design decisions are made sequentially, i.e., one after the other. The intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, are dependent of each other. More precisely, the ${i_j}^{\mathrm{th}}$ interval, which corresponds to the j^{th} decision, made for the i^{th} design variable, always depends on the first $j - 1$ design decisions, $j \in \{2, \ldots, d\}$. This can be understood as a traditional or over-the-wall engineering approach. The idea is visualized in Figure 6.

Note that mixed approaches for selecting the single design variables are also conceivable. For example, there are dependent-decoupled design decisions to differentiate between early and

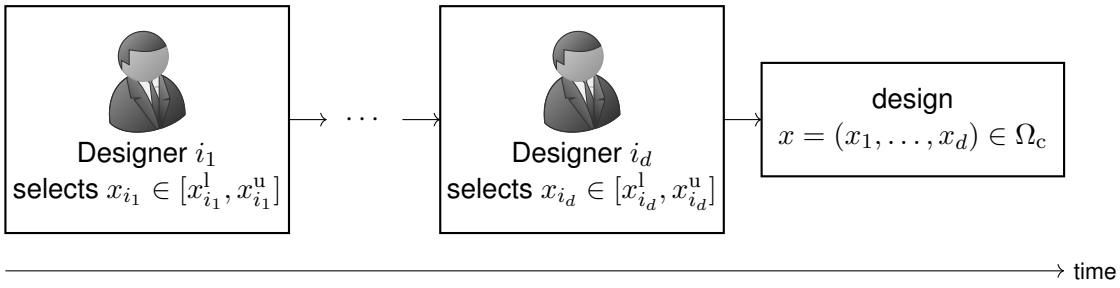


Figure 6 Dependent-decoupled design decisions by d designers.

late design variables mixed with independent-decoupled design decisions for the early decision variables, compare Section 2.4. Before addressing this further, fundamental classification schemes for uncertainties in systems design are discussed.

2.1.3. Uncertainty Classification

In general, there are many ways to classify uncertainties that occur during the design process of a system, see, e.g., [11, 94, 114, 126]. First, uncertainty is considered by means of a system's point of view, which is based on the above definitions.

So far, this section creates the impression that for any selected system design $x \in \Omega_{\mathrm{ds}}$ and given uncontrollable parameters p_l, $l = 1, \ldots, q$, there is a certain response of the system yielding precise statements about its permissibility. In realistic systems, however, the results are often inaccurate due to uncertainty. Uncertainties can occur in (a) controllable variables, (b) uncontrollable parameters, and (c) constraints. The following list gives an overview of plausible reasons and consequences of the occurrence of these uncertainties in systems design:

(a) *Uncertainties in controllable variables*: In the case of uncertainty in the controllable variables, the selected design variables cannot be realized exactly. Designers can only select nominal values expressed by $\check{x}_i \in \mathbb{R}$, $i = 1, \ldots, d$, which are called *target design variables*. However, the realized values of the design variables x_i, $i = 1, \ldots, d$, show deviations from the nominal values because of uncertainty. Note that x_i, $i = 1, \ldots, d$, are allowed to take different values for different realizations. Here, each x_i refers to one particular realization of the design variables, $i = 1, \ldots, d$.

Uncertainties in controllable variables mainly arise when the design is manufactured, i.e., in the late design phase, see [107]. If design variables represent geometric quantities, there can be, for example, manufacturing tolerances from imprecise machinery. Nevertheless, uncertainties in controllable variables can be already considered in the early design phase, e.g., when design variables are defined as responses of subsystems, cf. Section 3.1. Then, uncertainties are propagated from these subsystems. In the worst case, the subsystems are not defined in an early phase of systems design, see [34]. Thus, only the type of subsystem responses is known, but nothing about their lower-level design variables and corresponding functions, which is further discussed in Section 2.2. In addition, uncertainties

in controllable parameters can also account for non-realizable system designs within the system design space or for numerical errors that arise because of the quantification of the design.

(b) *Uncertainties in uncontrollable parameters*: Similar to uncertainties in controllable variables, the actual values of the parameters p_l are often unknown and only nominal values $\check{p}_l$ are given or can be assumed for $l = 1, \ldots, q$. Note again that each p_l refers to only one particular realization of the uncontrollable parameters, $l = 1, \ldots, q$.
As uncontrollable parameters can be, for example, former controllable variables whose values have been already selected, all examples of uncertainties in controllable variables are also possible for uncontrollable parameters. Furthermore, the uncontrollable parameters can address physical quantities, operational conditions, or parameter requirements, cf. [11]. Then, uncertainties due to natural variations, perturbations, or manual changes are present. Note that although each p_l represents only one realization, multiple realizations can be taken into account by considering sets of realizations, $l \in \{1, \ldots, q\}$. In particular, this is important for changing environmental, physical quantities, or system parameters that change during the system's life cycle. The latter circumstance may also affect controllable variables.

(c) *Uncertainties in constraints*: Although uncertainties in controllable variables and uncontrollable parameters have a direct impact on the outputs of the constraints as they are propagated through the system performance function, they are not considered as part of the uncertainties in the constraints. Despite that, the system performance functions of the constraints might be of the wrong mathematical structure, e.g., they are assumed to be linear instead of quadratic, or they consider the wrong variables and parameters as inputs. The typical representative for uncertainties in the performance functions, and therefore in the constraints, is model uncertainty due to the use of imprecise or inaccurate models, instead of real physical objects. This may also incorporate uncertainties in model parameters which are not considered as part of the uncontrollable parameters, compare [134]. Moreover, model errors due to the use of numerical approximations can be considered as uncertainty here, too, cf. [11]. The same considerations hold for the thresholds $f_{\mathrm{c},j}$, $j = 1, \ldots, m$, if they depend on uncontrollable parameters.
Furthermore, it is also possible that wrong system performances and thresholds are taken into account, some constraints are not required, and required ones are not considered. In addition, constraints may also change during the design process.

Note that this list neither claims completeness nor to be the only classification of uncertainty from a system's point of view. In this thesis, it is assumed that system performance functions are provided and there is no possibility to assess the underlying model uncertainty. This is typically accepted for the early phase in systems design. Thus, the focus of this thesis is put on uncertainties in the controllable variables and uncontrollable parameters.

In engineering applications, there is a complementary classification scheme to the one pre-

sented, which is often used and which distinguishes between aleatoric and epistemic uncertainty. This classification scheme motivates building mathematical models for parameters or variables describing these uncertainties.
Uncertainty caused by intrinsic randomness is called *aleatoric uncertainty*. Because of its stochastic nature, this randomness is considered as being irreducible, and aleatoric uncertainty is commonly modeled with probability distributions, compare [11, 124]. Thus, a probability can be assigned to finding a realization of a variable or parameter in a specific region, for example. Typical examples for aleatoric uncertainty include variations of physical nature, e.g., in wind load, humidity, and material properties.
Uncertainty due to lack of knowledge is called *epistemic uncertainty*. In principle, this lack of knowledge could be reduced by more research or collecting more specific data, see [65]. Examples include model uncertainty from inaccurate or imprecise models, which may be of mathematical or physical nature, statistical uncertainty from too little data, or non-investigated errors from applied numerical solution methods like discretization errors. In literature, there are various approaches how to model epistemic uncertainty. Amongst others, popular approaches use interval analysis [74, 96], fuzzy sets [136], possibility theory [31, 138], evidence theory [28, 118], and imprecise probabilities [131].

Furthermore, uncertainty that involves both aleatoric and epistemic characteristics can be treated in a common framework which is referred to as *polymorphic uncertainty* modeling in [55, 56]. Here, different approaches from aleatoric and epistemic modeling are usually combined, see e.g., [54, 105]. However, it shall be mentioned that sometimes, it is difficult or even not possible to distinguish whether a particular uncertainty is aleatoric, epistemic, or a combination of both, compare [76]. Thus, no model which can be considered as the most appropriate in these cases exists and it has to be decided based on the available knowledge, which model to choose.
As stated above, the epistemic uncertainty modeling can be improved theoretically by more research or gathering more specific data. However, in systems engineering, there are often deadlines that hinder gathering more knowledge about uncertainty in time. The state in which designers cannot get new information before the deadline is called a *no-more-knowledge* state in [42]. The other extreme is a *complete-knowledge* state in which all information is revealed before the deadline.

In this thesis, a no-more-knowledge state is considered in which only little knowledge about uncertainties is available. This is usually the case during the early design phase of a system. To comply with this, epistemic uncertainty is modeled as intervals and fuzzy sets. Here, *interval-type* uncertainty is regarded in an interval analysis context and *fuzzy-type* uncertainty in the framework of possibility theory. These epistemic uncertainty models have amongst others the advantage that multiple sources of uncertainties like unknown variations, numerical errors, and propagated uncertainty from subsystems can be directly embedded. For reasons of

simplicity, it is assumed that the uncertainties are uncorrelated between the single parameters and variables. More details on this modeling are provided below.

2.2. Epistemic Uncertainty Models

This section presents how epistemic uncertainties are modeled as intervals and fuzzy sets in this thesis. First, interval-type uncertainty is considered in the context of interval analysis. Then, an extension of the modeling is proposed yielding fuzzy-type uncertainty, which is treated in the framework of possibility theory.

2.2.1. Interval-Type Uncertainty

To begin with, the considerations of interval-type uncertainty are limited to controllable design variables. When the values of the target design variables $\check{x}_i$ are selected, the actual values of the design variables x_i can be only found within intervals because of uncertainty for $i = 1, \ldots, d$. In this thesis, it is assumed that the actual values are uncorrelated, symmetrically distributed, and bounded around these nominal values. The distances from the nominal values $\check{x}_i$ to the bounds of the actual values of the controllable variables are stated as $\delta_i \in \mathbb{R}_0^+$, $i = 1, \ldots, d$, which are also referred to as *uncertainty magnitudes* in the following. Note that each δ_i, $i \in \{1, \ldots, d\}$, can address multiple sources of uncertainties like stated above. For now, the uncertainty magnitudes δ_i, $i = 1, \ldots, d$, are assumed to be given and it holds

$$x_i \in [\check{x}_i - \delta_i, \check{x}_i + \delta_i], \tag{2.7}$$

$i = 1, \ldots, d$. This means that the realized system design x is always an element of the d-dimensional interval

$$\mathcal{U}^X(\check{x}, \delta) = \{x \in \mathbb{R}^d \mid x_i \in [\check{x}_i - \delta_i, \check{x}_i + \delta_i],\ i = 1, \ldots, d\}, \tag{2.8}$$

which depends on $\check{x} = (\check{x}_1, \ldots, \check{x}_d)$ and $\delta = (\delta_1, \ldots, \delta_d)$, where X refers to the controllable variables. Hence, the set $\mathcal{U}^X(\check{x}, \delta)$ is the *uncertainty set* of the controllable variables with respect to $\check{x} \in \mathbb{R}^d$.

Note that this framework for interval-type uncertainty has parallels to the conceptual approach in, e.g., [141] in which designers aim to realize arbitrary values within *target intervals*. In the above context, the smallest target intervals for which there is a guarantee that the actual values of the design variables can be found within, have size $2\delta_i$, $i = 1, \ldots, d$. Hence, targeting $\check{x}_i$ can always be considered as targeting $[\check{x}_i - \delta_i, \check{x}_i + \delta_i]$ for $i = 1, \ldots, d$ and vice versa.

Similar to interval-type uncertainties in controllable variables, interval-type uncertainties in uncontrollable parameters can be defined. It is assumed that the actual values of the uncontrol-

lable parameters, expressed by p_l are symmetrically distributed and bounded around nominal values, expressed by $\check{p}_l \in \mathbb{R}$, $l = 1, \ldots, q$. In contrast to $\check{x}_i$, $i = 1, \ldots, d$, the values of $\check{p}_l$, $l = 1, \ldots, q$, cannot be controlled by designers and are assumed to be given. The uncertainty magnitudes of the uncontrollable parameters are stated as $\gamma_l \in \mathbb{R}_0^+$ and are assumed to be given. It holds

$$p_l \in [\check{p}_l - \gamma_l, \check{p}_l + \gamma_l], \tag{2.9}$$

$l = 1, \ldots, q$. Thus, the actual values of p are always an element of the q-dimensional interval

$$\mathcal{U}^P(\check{p}, \gamma) = \{p \in \mathbb{R}^q \mid p_l \in [\check{p}_l - \gamma_l, \check{p}_l + \gamma_l],\ l = 1, \ldots, q\}, \tag{2.10}$$

which depends on $\check{p} = (\check{p}_1, \ldots, \check{p}_q)$ and $\gamma = (\gamma_1, \ldots, \gamma_q)$, where P refers to the uncontrollable parameters. The set $\mathcal{U}^P(\check{p}, \gamma)$ is the uncertainty set of the uncontrollable parameters.

For interval-type uncertainties in controllable variables and uncontrollable parameters, it is necessary to establish how the constraints on the design variables and the responses are interpreted. Approaches that interpret constraints for interval-type uncertainty can be found, for example, in [45, 119]. Similar to distinguishing between strong and weak solutions, this thesis differentiates between a worst- and a best-case scenario.
In order to account for the worst-case, the constraints must be fulfilled for all possible uncertainty realizations. Given $\check{x}$, δ, $\check{p}$, and γ, the constraints $x \in \Omega_{\mathrm{ds}}$ and $f(x, p) \leq f_{\mathrm{c}}(p)$ are met if they hold for all $p \in \mathcal{U}^P(\check{p}, \gamma)$ and all $x \in \mathcal{U}^X(\check{x}, \delta)$. The set of designs $x \in \Omega_{\mathrm{ds}}$ that fulfill $f(x, p) \leq f_{\mathrm{c}}(p)$ for all $p \in \mathcal{U}^P(\check{p}, \gamma)$ is called *worst-case complete system solution space* and is denoted by $\Omega_{\mathrm{c,wc}}$, i.e.,

$$\Omega_{\mathrm{c,wc}} = \{x \in \Omega_{\mathrm{ds}} \mid \forall p \in \mathcal{U}^P(\check{p}, \gamma) :\ f(x, p) \leq f_{\mathrm{c}}(p)\}. \tag{2.11}$$

Furthermore, the set of all system target designs $\check{x} \in \mathbb{R}^d$ that fulfill the constraints $x \in \Omega_{\mathrm{ds}}$ and $f(x, p) \leq f_{\mathrm{c}}(p)$ in the worst-case is

$$\check{\Omega}_{\mathrm{c,wc}} = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X(\check{x}, \delta) :\ x \in \Omega_{\mathrm{c,wc}}\} \tag{2.12}$$

and is named *worst-case complete system solution space of target designs*.
Opposite to the worst-case, there is the best-case in which the constraints must be only fulfilled for at least one uncertainty realization. Given $\check{x}$, δ, $\check{p}$, and γ, the constraints are met if they hold for at least one $x \in \mathcal{U}^X(\check{x}, \delta)$ and one $p \in \mathcal{U}^P(\check{p}, \gamma)$. The set of all designs $x \in \Omega_{\mathrm{ds}}$ that fulfill $f(x, p) \leq f_{\mathrm{c}}(p)$ for at least one $p \in \mathcal{U}^P(\check{p}, \gamma)$ is called *best-case complete system solution space* and is denoted by $\Omega_{\mathrm{c,bc}}$, i.e.,

$$\Omega_{\mathrm{c,bc}} = \{x \in \Omega_{\mathrm{ds}} \mid \exists p \in \mathcal{U}^P(\check{p}, \gamma) :\ f(x, p) \leq f_{\mathrm{c}}(p)\}. \tag{2.13}$$

Moreover, the set of all system target designs $\check{x} \in \mathbb{R}^d$ that fulfill the constraints $x \in \Omega_{\mathrm{ds}}$ and

$f(x,p) \leq f_{\mathrm{c}}(p)$ in the best-case is

$$\check{\Omega}_{\mathrm{c,bc}} = \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}^X(\check{x},\delta) :\ x \in \Omega_{\mathrm{c,bc}}\} \tag{2.14}$$

and is named *best-case complete system solution space of target designs*. Note that in Definition (2.14), the system target designs are not restricted to the system design space Ω_{ds}. Such a restriction could be done by adding system target design space constraints. Subsequently, examples of complete system solution spaces of designs and target designs for both scenarios are illustrated.

Example 3. Given the system design space $\Omega_{\mathrm{ds}} = [0,2] \times [0,1.5]$ and the performance function $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R},\ (x,p) \mapsto p_1x_1 + p_2x_2$ with threshold $f_{\mathrm{c}} = 2$ from Example 1. There are interval-type uncertainties in the controllable variables and the uncontrollable parameters. The nominal values of the uncontrollable parameters are given by $\check{p}_1 = 1$ and $\check{p}_2 = 2$. Regarding uncertainty, the different combinations of (a) $\delta_1, \delta_2 = 0$, (a') $\delta_1 = 0.3$, $\delta_2 = 0.1$, and (b) $\gamma_1, \gamma_2 = 0$, (b') $\gamma_1 = \gamma_2 = 0.2$ are considered. The corresponding worst- and best-case complete system solution spaces for the designs and target designs can be obtained from Equations (2.11)-(2.14) and are visualized in Figure 7.

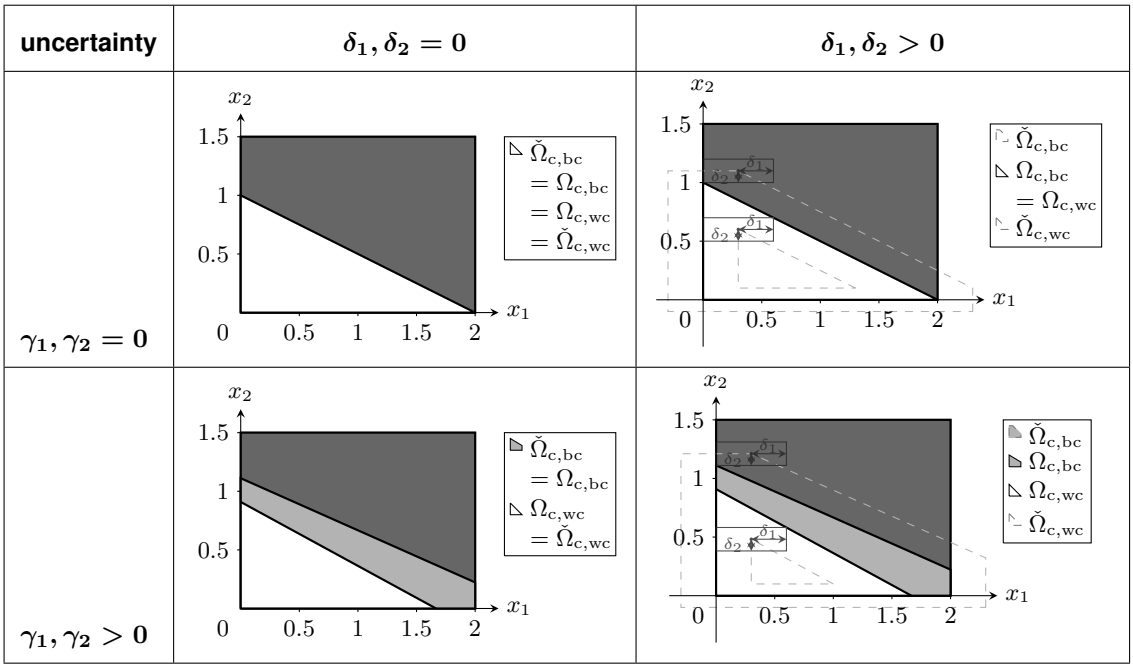


Figure 7 Worst- and best-case complete system solution spaces for the designs and target designs for different interval-type uncertainties of Example 3.

The case with $\delta_i = 0$, $i = 1,\ldots,d$, $\gamma_l = 0$, $l = 1,\ldots,q$, is equivalent to the case in which no uncertainties in controllable variables and uncontrollable parameters exist. Hence, the absence of uncertainty can be always considered as a special case of interval-type uncertainty in which $\Omega_{\mathrm{c}} = \check{\Omega}_{\mathrm{c,bc}} = \Omega_{\mathrm{c,bc}} = \Omega_{\mathrm{c,wc}} = \check{\Omega}_{\mathrm{c,wc}}$ holds. Furthermore, it is $\check{\Omega}_{\mathrm{c,bc}} = \Omega_{\mathrm{c,bc}}$ and

$\Omega_{\mathrm{c,wc}} = \check{\Omega}_{\mathrm{c,wc}}$ if there are no uncertainties in controllable variables and $\Omega_{\mathrm{c}} = \Omega_{\mathrm{c,bc}} = \Omega_{\mathrm{c,wc}}$ if there are no uncertainties in uncontrollable parameters, compare Figure 7.

Above, it is assumed that the values of δ and γ are always known. This might be reasonable for γ. Because the nominal values of $\check{p}$ do not change, the chances to gather knowledge about the associated uncertainty magnitudes might increase. For example, there can be knowledge from previously considered, similar systems, or expert knowledge for specific uncontrollable parameters. However, the assumption of fixed values for δ is often critical, cf. [25]. Possible reasons are:

- As $\check{x}_i$, $i = 1, \ldots, d$, can assume different values, the assumption of a fixed δ_i for all $\check{x}_i$ might be wrong, i.e., it is likely too small or too large for any $\check{x}_i$, $i = 1, \ldots, d$.
- If x_i are responses of a subsystem, δ_i depend on propagated uncertainty from this subsystem, $i = 1, \ldots, d$. If the subsystem is not defined, which might be the case in the early design phase, there is no knowledge about δ.

An approach that aims at circumventing these problems is proposed in Section 2.3.

2.2.2. Fuzzy-Type Uncertainty

Considering interval-type uncertainty, $x \in \mathbb{R}^d$ and $p \in \mathbb{R}^q$ either belong or do not belong to the uncertainty sets $\mathcal{U}^X(\check{x}, \delta)$ and $\mathcal{U}^P(\check{p}, \gamma)$ for given $\check{x}$, δ, $\check{p}$, and γ. In case there is further knowledge on the uncertainty magnitudes of the controllable variables and uncontrollable parameters, the uncertainties can also be modeled as fuzzy sets. In the following, a simple extension of the definitions for interval-type uncertainty is conducted, yielding fuzzy-type uncertainty that is parametrized by two uncertainty magnitudes. However, note that most of the results of this thesis for fuzzy-type uncertainty are not restricted to this simplification. Instead, the results can be transferred to more general fuzzy sets. A basic introduction to fuzzy sets which provides the fundamental definitions for this thesis is given in Section A.2.

Similar to the case of interval-type uncertainty, the considerations for fuzzy-type uncertainty are limited to controllable variables first. When the values of the target design variables $\check{x}_i$ are selected, all values in $\mathbb{R}$ can be assigned with a degree of membership of belonging to the fuzzy uncertainty sets of x_i, $i = 1, \ldots, d$. Here, the fuzzy uncertainty sets of x_i with respect to $\check{x}_i$ are denoted by $X_i(\check{x}_i)$ and the degrees of membership by α, reaching from zero to one. A degree of $\alpha = 1$ indicates that a design variable belongs for sure to $X_i(\check{x}_i)$ and a degree of $\alpha = 0$ indicates that a design variable belongs for sure not to $X_i(\check{x}_i)$, $i = 1, \ldots, d$, compare Section A.2. Again, it is assumed that the actual values are uncorrelated, symmetrically distributed, and bounded around the nominal values. In this thesis, the distances from the nominal values to the bounds of the values with $\alpha = 1$ are given by $\delta_{\alpha=1,i} \in \mathbb{R}_0^+$ and the distances from the nominal

values to the bounds of the closest values with $\alpha = 0$ by $\delta_{\alpha=0,i} \in \mathbb{R}_0^+$ where $\delta_{\alpha=0,i} \geq \delta_{\alpha=1,i}$, $i = 1, \ldots, d$. In between these bounds, the degrees of membership of belonging to the fuzzy uncertainty sets of x_i with respect to $\check{x}_i$ are assumed to behave linearly, $i = 1, \ldots, d$. Again, the quantities $\delta_{\alpha,i}$, $i = 1, \ldots, d$, $\alpha \in [0, 1]$, are referred to as uncertainty magnitudes. Overall, the properties of this fuzzy-type uncertainty can be represented by *membership functions*, defined as

$$\mu^{X_i(\check{x}_i)}(x_i) : \mathbb{R} \to [0,1],\ x_i \mapsto \begin{cases} 1 & \text{if } |x_i - \check{x}_i| \leq \delta_{\alpha=1,i}, \\ 1 - \frac{|x_i - \check{x}_i| - \delta_{\alpha=1,i}}{\delta_{\alpha=0,i} - \delta_{\alpha=1,i}} & \text{if } \delta_{\alpha=1,i} < |x_i - \check{x}_i| < \delta_{\alpha=0,i}, \\ 0 & \text{otherwise} \end{cases} \tag{2.15}$$

if $\delta_{\alpha=0,i} > \delta_{\alpha=1,i}$, and

$$\mu^{X_i(\check{x}_i)}(x_i) : \mathbb{R} \to [0,1],\ x_i \mapsto \begin{cases} 1 & \text{if } |x_i - \check{x}_i| \leq \delta_{\alpha=1,i}, \\ 0 & \text{otherwise} \end{cases} \tag{2.16}$$

if $\delta_{\alpha=0,i} = \delta_{\alpha=1,i}$, $i = 1, \ldots, d$. As remarked above, the membership functions and hence the fuzzy uncertainty sets of x_i with respect to $\check{x}_i$ are each parametrized by two uncertainty magnitudes, i.e., $\delta_{\alpha=0,i}$ and $\delta_{\alpha=0,i}$ for each $i \in \{1, \ldots, d\}$. In Figure 8, examples of the graph of $\mu^{X(\check{x}),i}$ are visualized for both $\delta_{\alpha=0,i} > \delta_{\alpha=1,i}$ and $\delta_{\alpha=0,i} = \delta_{\alpha=1,i}$, $i \in \{1, \ldots, d\}$.

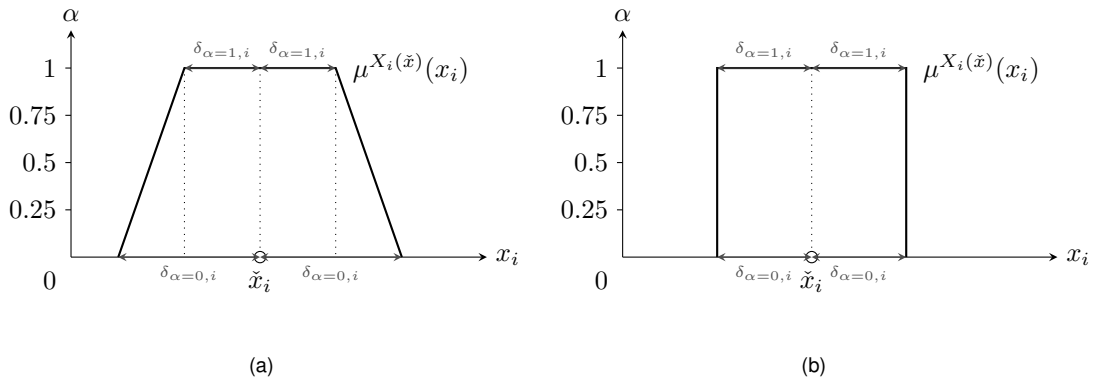


Figure 8 Examples of the graph of the membership function $\mu^{X(\check{x}),i}$ with (a) $\delta_{\alpha=0,i} > \delta_{\alpha=1,i}$ and (b) $\delta_{\alpha=0,i} = \delta_{\alpha=1,i}$, $i \in \{1, \ldots, d\}$.

The fuzzy uncertainty set of the controllable variables with respect to the system target design $\check{x}$ is denoted by X. For a system design x, the degree of membership of belonging to X is defined as the minimum of all membership function values of the single design variables, see Equation (A.12). For their joint membership function, it holds

$$\mu^{X(\check{x})}(x) = \min\{\mu^{X_1(\check{x}_1)}(x_d), \ldots, \mu^{X_d(\check{x}_d)}(x_d)\}. \tag{2.17}$$

The α-cuts of X are denoted by $\mathcal{U}_\alpha^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$ for $\alpha \in (0,1]$ and the support of X is denoted by $\mathcal{U}_0^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$ where $\delta_{\alpha=0,i}$,$\delta_{\alpha=1,i}$, $i = 1, \ldots, d$, are collected in $\delta_{\alpha=0}, \delta_{\alpha=1} \in \mathbb{R}^d$. It is

$$\mathcal{U}_\alpha^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) = \{x \in \mathbb{R}^d \mid \mu^{X(\check{x})}(x) \geq \alpha\} \tag{2.18}$$

for $\alpha \in (0,1]$, and

$$\mathcal{U}_0^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) = \{x \in \mathbb{R}^d \mid \mu^{X(\check{x})}(x) > 0\}. \tag{2.19}$$

For the special case of $\delta_{\alpha=0} = \delta_{\alpha=1}$, the uncertainty set is a crisp set with $\mathcal{U}_\alpha(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) = \mathcal{U}_{\alpha'}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$ for all $\alpha, \alpha' \in [0,1]$. This case corresponds to the case of interval-type uncertainty, compare Equation (2.8), and shows the property of fuzzy-type uncertainty to generalize interval-type uncertainty. For $\delta_{\alpha=0,i} > \delta_{\alpha=1,i}$, $i = 1, \ldots$, the support $\mathcal{U}_0^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$ is an open set, see Section 4.1.

Similar to fuzzy-type uncertainties in controllable variables, fuzzy-type uncertainty in uncontrollable parameters can be defined. The fuzzy uncertainty set of the uncontrollable parameters is denoted by P and its membership function μ^P depends on $\gamma_{\alpha=0,l}$,$\gamma_{\alpha=1,l}$, $l = 1, \ldots, q$, collected in $\gamma_{\alpha=0}, \gamma_{\alpha=1} \in \mathbb{R}^q$. It is defined similarly to $\mu^{X(\check{x})}$ via the single membership functions from Definitions (2.15) and (2.16) where x is substituted by p, i by l, d by q, and δ by γ. For the α-cuts of the fuzzy uncertainty set of the uncontrollable parameters, denoted by $\mathcal{U}_\alpha^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})$ for $\alpha \in (0,1]$, and its support, denoted by $\mathcal{U}_0^P(\check{p}, \delta_{\gamma=0}, \delta_{\gamma=1})$, it holds

$$\mathcal{U}_\alpha^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) = \{p \in \mathbb{R}^q \mid \mu^P(p) \geq \alpha\} \tag{2.20}$$

for $\alpha \in (0,1]$ and

$$\mathcal{U}_0^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) = \{p \in \mathbb{R}^q \mid \mu^P(p) > 0\}. \tag{2.21}$$

The properties of the fuzzy uncertainty set of the controllable variables transfer directly to the fuzzy uncertainty set of the uncontrollable parameters. Still, the values of $\check{p}_l$, $l = 1, \ldots, q$, cannot be controlled by designers.

Also for fuzzy-type uncertainty, it is necessary to establish how constraints on the design variables and the responses are interpreted. An overview of approaches that interpret constraints for fuzzy-type uncertainty can be found, for example, in [132, 133]. Similar to [30], this thesis uses possibility theory. In Section A.2, a basic introduction to possibility theory is given that emphasizes its interrelation with fuzzy sets.
In possibility theory, a possibility and necessity can be assigned to the fulfillment of constraints, which is similar to assigning a probability in probability theory. If thresholds are put on this possibility or necessity, so-called *chance constraints* are obtained, cf. [90]. Descriptively spoken, a possibility of zero means that it is absolutely not possible that the constraints are fulfilled, a possibility of one means that it is entirely possible, but not necessary that the

constraints are fulfilled, a necessity of zero means that it is not necessary, but possible that the constraints are fulfilled, and a necessity of one means that it is absolutely necessary that the constraints are fulfilled, compare [68]. Note that in general, possibility measures can also be interpreted as upper probability measures and necessity measures as lower probability measures, see [32], which is also discussed in Section A.2.

The set of all designs $x \in \Omega_{\mathrm{ds}}$ for which the possibility of fulfilling $f(x,p) \leq f_{\mathrm{c}}(p)$ is greater than or equal to α is called *possibility-α complete system solution space* and is denoted by $\Omega_{\mathrm{c,pos},\alpha}$, $\alpha \in (0,1)$. The set of all designs $x \in \Omega_{\mathrm{ds}}$ for which the necessity of fulfilling $f(x,p) \leq f_{\mathrm{c}}(p)$ is greater than or equal to α is called *necessity-α complete system solution space* and is denoted by $\Omega_{\mathrm{c,nec},\alpha}$, $\alpha \in (0,1)$. The set of all system target designs $\check{x} \in \mathbb{R}^d$ for which the possibility of fulfilling $x \in \Omega_{\mathrm{ds}}$ and $f(x,p) \leq f_{\mathrm{c}}(p)$ is greater than or equal to α is called *possibility-α complete system solution space of target designs* and is denoted by $\check{\Omega}_{\mathrm{c,pos},\alpha}$, $\alpha \in (0,1)$. And the set of all system target designs $\check{x} \in \mathbb{R}^d$ for which the necessity of fulfilling $x \in \Omega_{\mathrm{ds}}$ and $f(x,p) \leq f_{\mathrm{c}}(p)$ is greater than or equal to α is called *necessity-α complete system solution space of target designs* and is denoted by $\check{\Omega}_{\mathrm{c,nec},\alpha}$, $\alpha \in (0,1)$. They are all defined mathematically in Definition 2. For continuous system performance functions f it holds

$$\Omega_{\mathrm{c,nec},\alpha} = \{x \in \Omega_{\mathrm{ds}} \mid \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ f(x,p) \leq f_{\mathrm{c}}(p)\}, \tag{2.22}$$

$$\Omega_{\mathrm{c,pos},\alpha} = \{x \in \Omega_{\mathrm{ds}} \mid \exists p \in \mathcal{U}^P_{\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ f(x,p) \leq f_{\mathrm{c}}(p)\}, \tag{2.23}$$

$$\check{\Omega}_{\mathrm{c,nec},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) : \ x \in \Omega_{\mathrm{c,nec},\alpha}\}, \tag{2.24}$$

$$\check{\Omega}_{\mathrm{c,pos},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}^X_{\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) : \ x \in \Omega_{\mathrm{c,pos},\alpha}\}, \tag{2.25}$$

$\alpha \in (0,1)$, see Theorem 5. For $\alpha \in \{0,1\}$, the corresponding possibility-α and necessity-α solution spaces are defined by Equations (2.23)-(2.24). Note that for $\alpha \in (0,1)$, these equations are only properties of the possibility-α and necessity-α complete system solution spaces which hold for continuous system performance functions. In Theorem 7, it is shown for all system designs and system target designs of the possibility-α complete system solution spaces that the possibility of fulfilling $x \in \Omega_{\mathrm{ds}}$ and $f(x,p) \leq f_{\mathrm{c}}(p)$ is equal to one for $\alpha = 1$ and strictly greater than zero for $\alpha = 0$. For all system designs and system target designs of the necessity-α complete system solution spaces, the necessity of fulfilling $x \in \Omega_{\mathrm{ds}}$ and $f(x,p) \leq f_{\mathrm{c}}(p)$ is equal to one for $\alpha = 1$ and strictly greater than zero for $\alpha = 0$ if $\delta_{\alpha=1} = \delta_{\alpha=0}$ and $\gamma_{\alpha=1} = \gamma_{\alpha=0}$, and only greater than or equals to zero otherwise. Also, note that possibility-0 complete system solution spaces are usually non-closed sets, compare Section 4.1.

Further properties of the possibility-α and necessity-α complete system solution spaces are: If $\gamma_{\alpha=1} = 0$ holds, it is $\Omega_{\mathrm{c,pos},1} = \Omega_{\mathrm{c,nec},0}$, and if further $\delta_{\alpha=1} = 0$ holds, it is also $\check{\Omega}_{\mathrm{c,pos},1} = \Omega_{\mathrm{c,pos},1} = \Omega_{\mathrm{c,nec},0} = \check{\Omega}_{\mathrm{c,nec},0}$. In the case of interval-type uncertainty, the Equations (2.23) and (2.25) for possibility-α system solution spaces are consistent with the Equations (2.13) and (2.14) for best-case system solution spaces and the Equations (2.22) and (2.24) for necessity-α system solution spaces are consistent with the Equations (2.11) and (2.12) for worst-case system solution spaces and all $\alpha \in [0,1]$. In the following, examples of possibility-α and necessity-α complete system solution spaces are considered.

Example 4. Given the system design space $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$ and the performance function $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R},\ (x, p) \mapsto p_1 x_1 + p_2 x_2$ with threshold $f_{\mathrm{c}} = 2$ from Example 1. Furthermore, the nominal values of p are given by $\check{p}_1 = 1$, $\check{p}_2 = 2$. Here, two different cases are considered:

(a) There are fuzzy-type uncertainties in uncontrollable parameters for which $\gamma_{\alpha=0,1} = \gamma_{\alpha=0,2} = 0.2$ and $\gamma_{\alpha=1,1} = \gamma_{\alpha=1,2} = 0.1$ hold, and there is absence of uncertainties in controllable variables.

(b) There are fuzzy-type uncertainties in the controllable variables for which $\delta_{\alpha=0,1} = 0.3$, $\delta_{\alpha=0,2} = 0.1$ and $\delta_{\alpha=1,1} = 0.1$, $\delta_{\alpha=1,2} = 0$ hold, and there is absence of uncertainties in uncontrollable parameters.

The corresponding possibility-α and necessity-α complete system solution spaces of designs and target designs can be obtained from Equations (2.23)-(2.24) and are visualized in Figure 9 for $\alpha = 0$ and $\alpha = 1$.

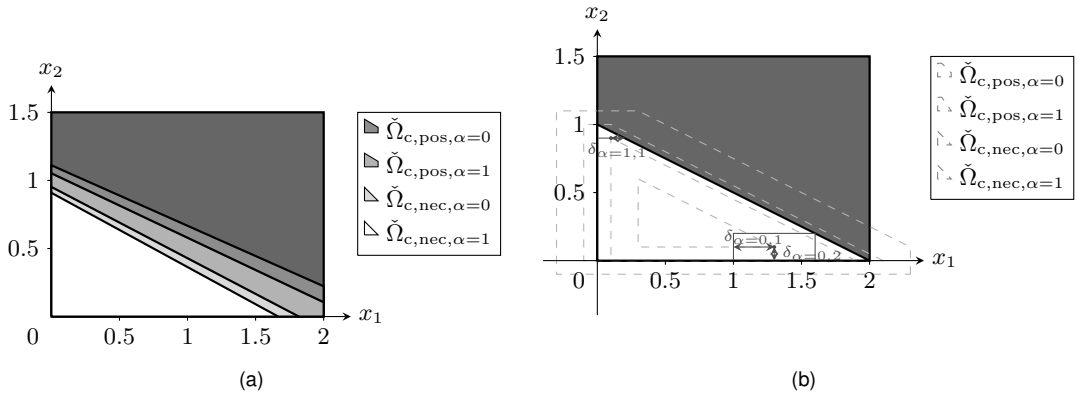


Figure 9 Possibility-α and necessity-α solution spaces with $\alpha \in \{0, 1\}$ for uncertainties in (a) uncontrollable parameters and (b) controllable variables of Example 4. Note that in the legend, the solution spaces are listed according to their size, the largest is on top.

Here, no explicit example for the case of both uncertainties in controllable variables and uncontrollable parameters is given. However, it can also be thought of as a combination of the results of Example 4.
Below, it is discussed how flexibility for both coupled and decoupled design decisions based on the complete system solution spaces can be provided.

2.3. Coupled Design Decisions

This section considers coupled design decisions under absence of uncertainty, under interval-type uncertainty, and under fuzzy-type uncertainty. It is based on the above framework in

which one designer is responsible to select a system design from the corresponding complete system solution space.

2.3.1. Coupled Design Decisions under Absence of Uncertainty

For coupled design decisions with a system designer, it is assumed that he has full knowledge about the complete system solution space Ω_{c}. Within Ω_{c}, he can select any system design, as any system target design is exactly realizable due to the absence of uncertainty, i.e., $x_i = \check{x}_i$, $i = 1, \ldots, d$. However, if Ω_{c} is not a singleton, there are multiple choices to select $x \in \Omega_{\mathrm{c}}$. As no preferences exist between the designs of the complete system solution space a priori, a ranking is desired in order to make a decision for a particular design.

Such a ranking can be provided by defining a *system cost function* c. This function measures the "costs" of the system. This can be, for example, material or manufacturing costs but also quantities like weight and rigidity. Similar to the performance functions, it is assumed that the cost function is uniquely determined by the design variables and uncontrollable parameters. It holds

$$c : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R},\ (x, p) \mapsto c(x, p). \tag{2.26}$$

Note that similar to above, the costs might also be restricted to subsets of $\mathbb{R}$. An intuitive system cost function regarding the system's performances can be, for example, a particular system response, i.e., $c = f_j$ for $j \in \{1, \ldots, m\}$. Another example is the minimum distance of all system responses to their thresholds, defined by

$$c(x, p) = \min_{j \in \{1, \ldots, m\}} \quad f_j(x, p) - f_{\mathrm{c},j}(p). \tag{2.27}$$

for $x \in \mathbb{R}^d$, $p \in \mathbb{R}^q$. In general, also multiple system cost functions can exist where c has multi-dimensional outputs. For reasons of simplicity, only one-dimensional outputs for system cost functions are considered in this thesis. As for system performance functions, the second argument in the system cost function is often neglected if uncontrollable parameters are not considered explicitly in a problem statement, i.e., $c = c(\cdot, p)$ for $p \in \mathbb{R}^q$. Furthermore, note that cost functions often contain subjective criteria which depend on the designer. They might be influenced by factors like the specific application, the time in the development process, or personal judgments of designers, compare [106].

Given a system cost function c, a system design $x \in \Omega_{\mathrm{c}}$ with lower costs $c(x)$ is preferred against a system design $x' \in \Omega_{\mathrm{c}}$ with higher costs $c(x')$ where $c(x) < c(x')$. Hence, often the design associated with the lowest costs is sought. This can be expressed as a mathematical

optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & c(x) \\ \text{subject to} \quad & x \in \Omega_{\mathrm{c}}. \end{aligned} \tag{2.28}$$

where the cost function c is the objective function and $x \in \Omega_{\mathrm{c}}$ expresses the optimization constraints. A solution of problem (2.28) is called an *optimal system design*. Due to its optimization constraints, problem (2.28) is also called a constraint optimization problem in mathematics. In literature, there are many approaches on how to solve constraint optimization problems. They mainly differ depending on the type of performance functions, which are included in Ω_{c}, and the type of cost function c. An overview can be found, for example, in [47, 100, 111]. Especially in [111], an emphasis is put on the application to engineering systems. For further information on optimization with multiple cost functions, read, e.g., [13]. Subsequently, an example of a coupled design decision in which an optimal system design is sought is considered. As the focus of this section is put on presenting different problem statements, a derivation of the corresponding solution is not provided for this and the following examples.

Example 5. Given the system design space $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$ and a performance function $f : \mathbb{R}^2 \to \mathbb{R},\ x \mapsto x_1 + 2x_2$ with threshold $f_{\mathrm{c}} = 2$ from Example 1. Furthermore, a system cost function $c : \mathbb{R}^2 \to \mathbb{R},\ x \mapsto 1.5 - 0.25x_1 - x_2$ exists. The optimal system design is unique and can be computed as $x = (0, 1)$ by solving problem (2.28). As this problem is linear here, linear programming techniques can be applied to find the solution numerically, see, e.g., [100]. The optimal system design is visualized in Figure 10.

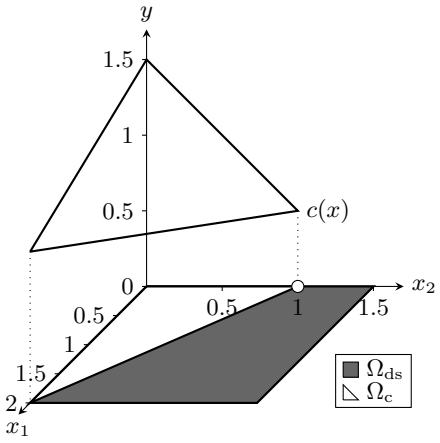


Figure 10 Optimal system design $x = (0, 1)$ (white dot) of Example 5.

If $d = 1$ holds for a system, a differentiation between coupled and decoupled design decision is not necessary. Nevertheless, the design decision for this case can be made similarly to the case of coupled design decisions for $d > 1$. Therefore, this case is also mentioned here.

Below, an example for $d = 1$ is given in which an optimal system design is sought.

Example 6. Given the system design space $\Omega_{\mathrm{ds}} = [0, 2]$ and a system performance function $f : \mathbb{R} \to \mathbb{R},\ x \mapsto -x$ with threshold $f_{\mathrm{c}} = -0.5$. Furthermore, a system cost function $c : \mathbb{R} \to \mathbb{R},\ x \mapsto \max(3.5 - 3x, 0.5x)$ exists. The optimal system design is unique and can be computed as $x = 1$ by solving problem (2.28). As the objective function is not differentiable here, methods from non-differentiable optimization must be applied to find the solution numerically, see, e.g., [120]. The optimal system design is visualized in Figure 11.

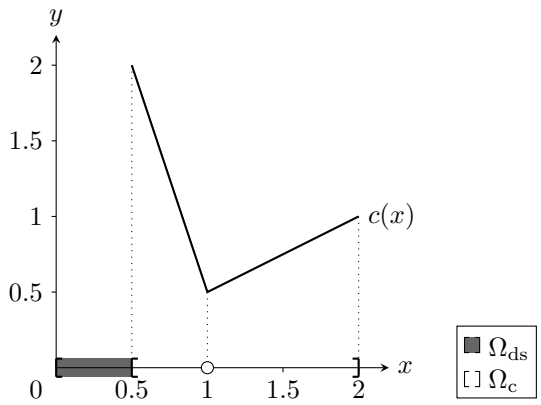


Figure 11 Optimal system design $x = 1$ (white dot) for Example 6.

2.3.2. Coupled Design Decisions under Interval-Type Uncertainty

For interval-type uncertainty, decisions which are made regarding the target design variables $\check{x}_i$, $i = 1, \ldots, d$, are considered. If no cost function is defined, any system target design within the worst- or best-case complete system solution space of target designs can be selected, depending on which case is taken into account. Again, there are no a-priori preferences between the permissible target designs in this case.

If a cost function c is introduced, preferences exist. As for uncertainties in constraints, there might be uncertainties in c due to considering a wrong mathematical structure, or the wrong variables and parameters as inputs, i.e., there are model uncertainties or model errors in c. It is also possible that the actual costs cannot be considered with the present cost function, as different costs are addressed. In addition, the cost function can change over the development process. In the following, uncertainties in the cost function are assumed to be unknown in the early design phase, and similar to uncertainties in constraints, they are not considered further. First, the focus is put on uncertainties in controllable variables. Here, the realized system design $x \in \Omega_{\mathrm{ds}}$ can have maximum costs of $\max_{x \in \mathcal{U}^x(\check{x},\delta)} c(x)$ for selected $\check{x} \in \mathbb{R}^d$. This corresponds to a worst-case scenario in terms of Section 2.2 and is named *robust regularization* in [89]. Hence, a goal is to find a design $\check{x} \in \check{\Omega}_{\mathrm{c,wc}}$ associated with a minimal maximum value

of $c(x)$ for $x \in \mathcal{U}^X(\check{x}, \delta)$. This is expressed in the mathematical optimization problem

$$\begin{aligned} \underset{\check{x}}{\text{minimize}} \quad & \max_{x \in \mathcal{U}^X(\check{x},\delta)} c(x) \\ \text{subject to} \quad & \check{x} \in \check{\Omega}_{\text{c,wc}}. \end{aligned} \tag{2.29}$$

As an optimal solution of problem (2.29) is sought under uncertainty, the optimization problem is called a *robust optimization problem*, and optimal solutions are named *robust system target designs*. Furthermore, problem (2.29) is also a mathematical constraint optimization problem that belongs to the class of minimax problems, see, e.g., [9, 17] in which approaches to solve minimax problems are presented. Subsequently, an example for problem (2.29) is considered.

Example 7. Given the system design space $\Omega_{\text{ds}} = [0, 2]$, the performance function $f : \mathbb{R} \to \mathbb{R},\ x \mapsto -x$ with threshold $f_{\text{c}} = -0.5$, and the cost function $c : \mathbb{R} \to \mathbb{R},\ x \mapsto \max(3.5 - 3x, 0.5x)$ from Example 6. Furthermore, there are interval-type uncertainties in the design variable with a magnitude of $\delta = 0.3$. Thus, the robust system target design in terms of problem (2.29) is unique and can be computed as $\check{x} = 1.2143$. It is visualized in Figure 12. In contrast to the optimal system design from Example 6, the robust system target design of this example is shifted to the right to avoid large costs if the realized system design becomes $x = \check{x} - \delta$.

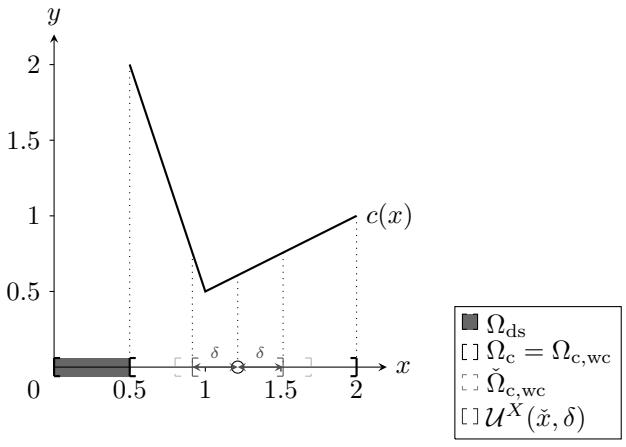


Figure 12 Robust system design $\check{x} = 1.2143$ (white dot) of Example 7.

In contrast to worst-case scenarios, best-case scenarios are usually not considered in robust optimization. Furthermore, robust optimization is not only restricted to uncertainties in controllable variables. For uncertainties in uncontrollable parameter under absence of uncertainties in controllable variables, the robust regularization can be formulated as $\max_{p \in \mathcal{U}^P(\check{p},\gamma)} c(x, p)$ for $x \in \mathbb{R}^d$ where $x_i = \check{x}_i$ holds for $i = 1, \ldots, d$. Here, the maximum costs of all possible realizations of $p \in \mathcal{U}^P(\check{p}, \gamma)$ are considered. The goal is to find a *robust system design* $x \in \Omega_{\text{c,wc}}$ associated with a minimal maximum value of $c(x, p)$ for $p \in \mathcal{U}^P(\check{p}, \gamma)$, which is

expressed in the minimax robust optimization problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \max_{p \in \mathcal{U}^P(\check{p},\gamma)} c(x,p) \\ \text{subject to} \quad & x \in \Omega_{\text{c,wc}}. \end{aligned} \tag{2.30}$$

Problem (2.30) is considered particularly in [7, 8] in which the problem statement is simplified in order to solve it with standard methods from mathematical optimization. This is also known as the *robust counterpart approach*. Subsequently, an example for problem (2.30) is given.

Example 8. Given the system design space $\Omega_{\text{ds}} = [0, 2]$, a performance function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ (x, p) \mapsto p - x$ with threshold $f_{\text{c}} = -0.5$, and a cost function $c : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ (x, p) \mapsto \max((3.5 - 2p) - (3 + 2p)x, p + 0.5x)$ similarly to Example 6. Furthermore, there are interval-type uncertainties in the uncontrollable parameters for which the nominal value of p is given by $\check{p} = 0$ and the uncertainty magnitude by $\gamma = 0.2$. Thus, the robust system design in terms of problem (2.30) is unique and can be computed as $x = 1.1935$. It is visualized in Figure 13. In contrast to the optimal system design from Example 6, the robust system design of this example is shifted to the right to avoid large costs if the uncontrollable parameter becomes $p = \check{p} - \gamma$. Note that this result is independent of the result from Example 7.

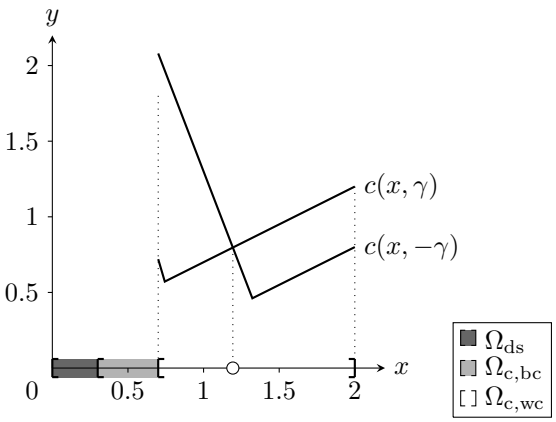


Figure 13 Robust system design $x = 1.1935$ (white dot) for Example 8.

From a mathematical point of view, the distinction between uncertainties in controllable variables and uncontrollable parameters is not absolutely necessary. Hence, approaches to solve Problems (2.29) and (2.30) can be sometimes transferred or they compensate for both types of uncertainty, depending on the given system cost and performance functions. Moreover, both can be combined in one robust optimization problem, i.e.,

$$\begin{aligned} \underset{\check{x}}{\text{minimize}} \quad & \max_{x \in \mathcal{U}^X(\check{x},\delta)} \max_{p \in \mathcal{U}^P(\check{p},\gamma)} c(x,p) \\ \text{subject to} \quad & \check{x} \in \check{\Omega}_{\text{c,wc}}. \end{aligned} \tag{2.31}$$

In [121], a neutral problem formulation is chosen in which both uncertainties in controllable variables and uncontrollable parameters can be incorporated. If no robust solution exists because of large uncertainty, the distance from the nominal values to the nearest realization that violates the constraints is proposed as a measure for the robustness of the design. This measure is called *radius of stability* in [121]. Then, the system target design with the largest radius of stability can be found via optimization.
A similar approach, limited to uncertainties in controllable variables, is proposed in [66]. Here, it is assumed that there is no information about the values of δ_i, $i = 1, \dots, d$. The system target design that allows the largest deviation such that the realized system design fulfills all constraints is defined as robust system target design, i.e., the target design with the largest radius of stability. If the maximum metric is used for distance computation, the corresponding optimization problem reads

$$\begin{aligned} &\underset{\check{x},\delta'}{\text{maximize}} \quad \delta' \\ &\text{subject to} \quad \mathcal{U}^X(\check{x}, \bar{\delta}) \subseteq \Omega_\mathrm{c} \end{aligned} \tag{2.32}$$

where δ' is the radius of stability with $\bar{\delta} = (\delta', \dots, \delta')$. Note that all entries of $\bar{\delta}$ are chosen to be the same to account for all system designs x with $\max_{i \in \{1,\dots,d\}} |x_i - \check{x}_i| = \delta'$. In contrast to the values of $\check{x}$, the value of δ' is unique. Problem (2.32) belongs to the class of *design centering problems*, as a system target design is centered in the complete system solution space Ω_c, see [64]. In [23], this is extended by also taking uncertainties in uncontrollable parameters with unknown γ into account and balancing them by solving a multi-objective optimization problem. Note that a system cost function c is not incorporated in the computation of robust system target designs with unknown δ_i, $i = 1, \dots, d$, in general. In the following, two examples are considered.

Example 9. (a) Given the system design space $\Omega_\mathrm{ds} = [0, 2] \times [0, 1.5]$ and the performance function $f : \mathbb{R}^2 \to \mathbb{R},\ x \mapsto x_1 + 2x_2$ with threshold $f_\mathrm{c} = 2$ from Example 1.

(b) Given the system design space $\Omega_\mathrm{ds} = [0, 2]$ and the performance function $f : \mathbb{R} \to \mathbb{R},\ x \mapsto -x$ with threshold $f_\mathrm{c} = -0.5$ from Example 6.

Furthermore, there are interval-type uncertainties in design variables with unknown δ. Hence, unique robust system target designs can be computed in terms of problem (2.32). For (a), it holds $\check{x} = (0.3334, 0.3334)$ with $\bar{\delta} = (0.3334, 0.3334)$ and for (b), it holds $\check{x} = 1.25$ with $\bar{\delta} = 0.75$. The robust system target designs are visualized in Figure 14.

Note that there are also approaches that use the volume of $\mathcal{U}^X(\check{x}, \delta)$ as a measure to find a robust system target design if δ_i, $i = 1, \dots, d$, are unknown, see, e.g., [64]. This can be examined critically because some δ_i might become very small in order to enlarge others, see

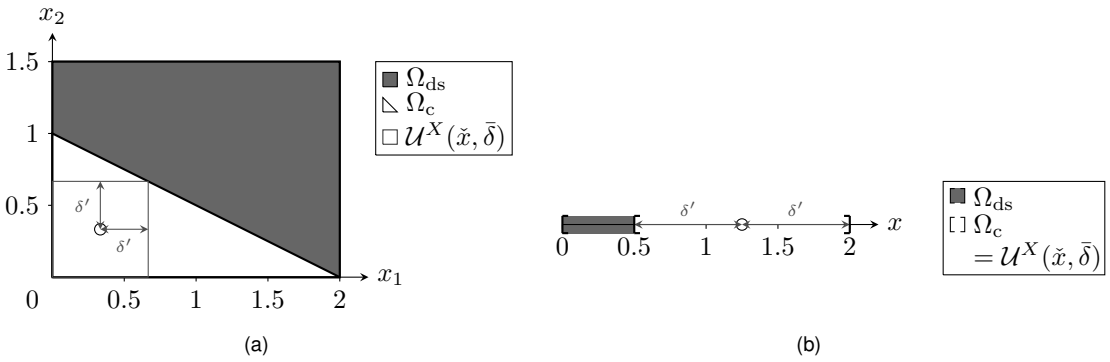


Figure 14 Robust system target designs (a) $\check{x} = (0.3334, 0.3334)$ and (b) $\check{x} = 1.25$ (white dots) of Example 9.

[43] for a discussion.

2.3.3. Coupled Design Decision under Fuzzy-Type Uncertainty

For fuzzy-type uncertainty, decisions which are made regarding the target design variables $\check{x}_i$, $i = 1, \ldots, d$, are considered again. Similar to above, any system target design within the possibility-α or necessity-α complete system solution space of target designs can be selected, depending on which case is taken into account, if no cost function is defined.

This changes again if a cost function is introduced. In the following, robust optimization problems for fuzzy-type uncertainty are proposed, which generalize the problems for interval-type uncertainty. According to the interpretation of the constraints, the system cost function is interpreted with the same measures from possibility theory here. This yields a consistent framework for optimization. Note that in general there are various approaches on how to compute a robust system design under fuzzy-type uncertainty, see, e.g., [91, 139]. The discussion on which approach to choose continues the above discussion on the interpretation of constraints under fuzzy-type uncertainty.

For a cost function c and fuzzy-type uncertainties in both controllable variables and uncontrollable parameters, a realized design $x \in \Omega_{\mathrm{ds}}$ from selected $\check{x} \in \mathbb{R}^d$ can have maximum costs of $\sup_{x \in \mathcal{U}_0(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})} \sup_{p \in \mathcal{U}_0(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})} c(x, p)$. This means that the necessity that the realized costs are smaller than or equal to the maximum costs is one, which can be shown similarly to the results of Section 4.1. If a necessity greater than or equals to α for $\alpha \in (0, 1)$ is sufficient, costs of $\sup_{x \in \mathcal{U}_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})} \sup_{p \in \mathcal{U}_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})} c(x, p)$ can be considered for selected $\check{x} \in \mathbb{R}^d$. Note that also the operator $\max$ instead of $\sup$ could be used for $\alpha \in (0, 1)$ as the α-cuts of the fuzzy uncertainty sets for the controllable variables and uncontrollable parameters are closed. These considerations can be extended to the case $\alpha = 1$, like done in Section 2.2, in which the results of Section 4.1 hold.

Compatible with these perceptions is selecting a system target design from the necessity-α complete system solution space of target designs. Hence, the goal becomes finding a design $\check{x} \in \check{\Omega}_{\mathrm{c,nec},\alpha}$ associated with a minimal maximum value of $c(x, p)$ for $x \in \mathcal{U}_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$

and $p \in \mathcal{U}_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})$. This is expressed in the robust optimization problem

$$\begin{aligned} \underset{\check{x}}{\text{minimize}} \quad & \sup_{x \in \mathcal{U}_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})} \; \sup_{p \in \mathcal{U}_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})} c(x,p) \\ \text{subject to} \quad & \check{x} \in \check{\Omega}_{\mathrm{c,nec},\alpha}. \end{aligned} \tag{2.33}$$

for $\alpha \in [0,1]$ for which an optimal solution is a robust system design. Like problems (2.29)-(2.31), problem (2.33) is a mathematical constraint optimization problem that belongs to the class of minimax problems. In the following, two examples for problem (2.33) are considered, one with uncertainties in controllable variables and one with uncertainties in uncontrollable parameters.

Example 10. Given the system design space $\Omega_{\mathrm{ds}} = [0,2]$, the performance function $f : \mathbb{R} \to \mathbb{R},\ x \mapsto -x$ with threshold $f_{\mathrm{c}} = -0.5$, and the cost function $c : \mathbb{R} \to \mathbb{R},\ x \mapsto \max(3.5 - 3x, 0.5x)$ from Example 6. Furthermore, there are fuzzy-type uncertainties in design variables given by Equation (2.17) with $\delta_{\alpha=0} = 0.3$ and $\delta_{\alpha=1} = 0$. Thus, robust system designs in terms of problem (2.33) for $\alpha \in [0,1]$ are unique and can be computed as $x_\alpha = 1 + 0.2143\alpha$. Here, the robust system designs for $\alpha = 0$ correspond to the case of no uncertainty from Example 6 and for $\alpha = 1$ to the case of no uncertainty from Example 7. They are visualized in Figure 15 for $\alpha \in \{0, 0.5, 1\}$.

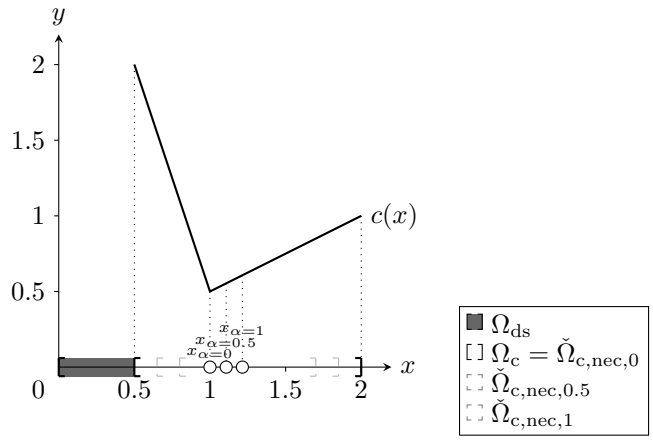


Figure 15 Robust system designs for $\alpha \in \{0, 0.5, 1\}$ (white dots) of Example 10. Note that in the legend, the solution spaces are listed according to their size, the largest is on top.

Example 11. Given the system design space $\Omega_{\mathrm{ds}} = [0,2]$, a performance function $f : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ (x,p) \mapsto p - x$ with threshold $f_{\mathrm{c}} = -0.5$, and a cost function $c : \mathbb{R} \times \mathbb{R} \to \mathbb{R},\ (x,p) \mapsto \max((3.5 - 2p) - (3 + 2p)x, p + 0.5x)$ from Example 8. Furthermore, there are fuzzy-type uncertainties in uncontrollable parameters given by Equation (2.17) with $\check{p} = 0$, $\gamma_{\alpha=0} = 0.2$, and $\gamma_{\alpha=1} = 0$. Thus, robust system designs in terms of problem (2.33) for $\alpha \in [0,1]$ are unique and can be computed as $x_\alpha = \frac{1+0.0571\alpha}{1-0.1143\alpha}$. Here, the robust system design for $\alpha = 0$

corresponds to the case of no uncertainty from Example 6 and for $\alpha = 1$ to the case of no uncertainty from Example 8. They are visualized in Figure 16 for $\alpha \in \{0, 0.5, 1\}$.

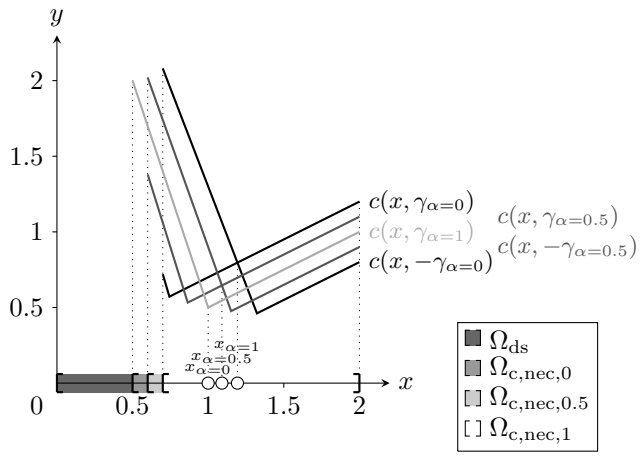


Figure 16 Robust system design for $\alpha \in \{0, 0.5, 1\}$ (white dots) of Example 11.

Besides problem (2.33), similar robust optimization problems under fuzzy-type uncertainty in literature use, for example, an objective function that measures the maximum possible error between the actual costs and the minimum costs for realized p, see [67, 70], or use possibility instead of necessity measures in the constraints, see, e.g., [29, 135]. Furthermore, a case in which there is no information about the values of $\delta_{\alpha=0}$, $\delta_{\alpha=1}$, $i = 1, \ldots, d$, can be handled similarly to interval-type uncertainty. This is considered in Section 3.5.

In the following, decoupled design decisions are considered. Here, the focus is put on approaches that provide optimal flexibility for these decisions. The decoupled design decisions themselves, e.g., for the single design variables, can then be made similarly to the coupled design decisions discussed in this section.

2.4. Decoupled Design Decisions

This section provides mathematical problem statements that provide optimal flexibility for decoupled design decisions under absence of uncertainty within the proposed framework. In order to do so, state-of-the-art approaches from literature that can be used for decoupled design decisions are reviewed and classified. Furthermore, existing research gaps are discussed, in particular regarding the consideration of epistemic uncertainty for decoupled design decisions.

2.4.1. Decoupled Design Decisions under Absence of Uncertainty

First, completely decoupled design decisions under absence of uncertainty are considered. Here, there are d designers, each responsible for one design variable, see Section 2.1. In order to provide flexibility for decoupled design decisions, intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, must be specified. The selection of the values of the corresponding design variables x_i, $i = 1, \ldots, d$, within these intervals can then be done similarly to the case of coupled design decisions for $d = 1$. Note that for an optimization of the design variables in $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, cost functions for the single design variables must be provided. These could be deduced, for example, from a given system cost function or could be directly defined for the single design variables. However, as cost functions depend on the designers' preferences in general, this aspect is not considered in this thesis. Instead, the focus is put on specifying the intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, given the complete system solution space Ω_{c}. This is investigated in the following.

For independent-decoupled design decisions, the designers must be able to select the values of the design variables independently of each other. As stated above, this also means that the intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, must be independent of each other. In order to guarantee a permissible system design $x \in \Omega_{\mathrm{c}}$ for $x_i \in [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, a necessary and also sufficient condition is that the Cartesian product of these intervals is a subset of Ω_{c}. As this Cartesian product defines a d-dimensional interval $[x^{\mathrm{l}}, x^{\mathrm{u}}] \subset \mathbb{R}^d$, the condition for independent-decoupled design decisions can be formulated as

$$[x^{\mathrm{l}}, x^{\mathrm{u}}] \subseteq \Omega_{\mathrm{c}}. \tag{2.34}$$

Multi-dimensional intervals that fulfill Condition (2.34) are called *box-shaped system solution spaces* in [140] and are denoted by Ω, i.e., $\Omega = [x^{\mathrm{l}}, x^{\mathrm{u}}]$. In general, there are many possibilities to choose box-shaped system solution spaces. Among all these, an optimal box-shaped system solution space that provides the most flexibility for independent-decoupled design decisions is preferred. This flexibility can be quantified, for instance, by the volume of $[x^{\mathrm{l}}, x^{\mathrm{u}}]$, see [86], for which

$$\mathrm{vol}([x^{\mathrm{l}}, x^{\mathrm{u}}]) = \prod_{i=1}^{d} (x_i^{\mathrm{u}} - x_i^{\mathrm{l}}) \tag{2.35}$$

holds. The larger the volume of the box-shaped system solution space, the more options exist for designers to select the values of the design variables for a system design $x \in [x^{\mathrm{l}}, x^{\mathrm{u}}]$. Furthermore, as Ω is only a subset of Ω_{c}, usually only a fraction of permissible system designs can be covered by Ω. This may exclude an optimal system design in terms of a system cost function c. Hence, a box-shaped system solution space with maximum volume is helpful

overall. By solving the mathematical optimization problem

$$\begin{aligned} \underset{x^{\mathrm{l}},x^{\mathrm{u}}}{\text{maximize}} \quad & \prod_{i=1}^{d}(x_i^{\mathrm{u}}-x_i^{\mathrm{l}}) \\ \text{subject to} \quad & [x^{\mathrm{l}},x^{\mathrm{u}}]\subseteq\Omega_{\mathrm{c}}, \end{aligned} \tag{2.36}$$

an optimal box-shaped system solution space Ω is obtained. Like optimization problem (2.32), problem (2.36) is a design centering problem in terms of [64]. This becomes clear when formulating a box-shaped system solution space with respect to its center and the distance to the bounds x^{l} and x^{u}. Furthermore, such a formulation shows the intrinsic property of box-shaped system solution spaces to provide also flexibility for design decisions under interval-type uncertainty, which is discussed further below. Approaches to solve problem (2.36) for different performance functions, which are incorporated in Ω_{c}, are widely discussed in the literature, see, for example, [58, 66, 95, 113, 140]. In [95, 113], optimal box-shaped system solution spaces are called *maximum volume inner boxes* as they form the largest box in terms of volume inside Ω_{c}.
Note that in general also different flexibility measures for optimizing box-shaped system solution spaces are conceivable and the choice of a particular measure should depend on the use case. For example, if the ratio of the single edge lengths shall be restricted, the minimum of the edge lengths, weighted by different factors, can be optimized, like introduced in [44]. If more weight shall be put on the size of a single edge length compared to another, Equation (2.35) can be used as an objective for which the single edge lengths are weighted by different exponents, see [37]. For reasons of simplicity, the considerations are limited to the volume in this thesis. Subsequently, an example for computing optimal box-shaped system solution space with maximum volume is considered. Similar to above, a detailed derivation of the corresponding solution is not provided for this and the following examples.

Example 12. Given the system design space $\Omega_{\mathrm{ds}} = [0,2]\times[0,1.5]$ and the performance function $f:\mathbb{R}^2\to\mathbb{R},\ x\mapsto x_1+2x_2$ with threshold $f_{\mathrm{c}}=2$ from Example 1. The optimal box-shaped system solution space can be computed as $[0,1]\times[0,0.5]$ using methods from Chapter 4. It is visualized in Figure 17.

For dependent-decoupled design decisions, the designers select the values of the design variables dependently of each other, which is expressed in the dependence of the intervals $[x_i^{\mathrm{l}},x_i^{\mathrm{u}}]$, $i=1,\dots,d$. For the first decision, intervals can be provided for all design variables. Then, any but only one designer, i.e., designer i_1, $i_1\in\{1,\dots,d\}$, can select $x_{i_1}\in[x_{i_1}^{\mathrm{l}},x_{i_1}^{\mathrm{u}}]$. Based on his decision, the intervals for the remaining design variables are updated and designer i_2 can select $x_{i_2}\in[x_{i_2}^{\mathrm{l}},x_{i_2}^{\mathrm{u}}]$ next, $i_2\in\{1,\dots,d\}\backslash\{i_1\}$. This is continued until a system design x is yielded.
After the value of the first design variable was selected, it must be ensured that there exists at least one combination of values of the remaining design variables $x_{i_2},\dots,x_{i_d}$ which can

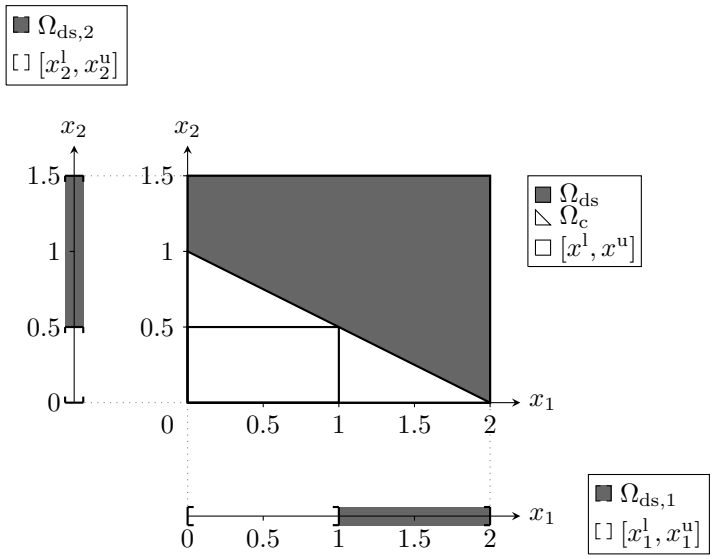


Figure 17 Optimal box-shaped system solution space $[x^{\mathrm{l}}, x^{\mathrm{u}}]$ of Example 12.

result in a permissible system design. Thus, $[x_{i_1}^{\mathrm{l}}, x_{i_1}^{\mathrm{u}}]$ must be a subset of the projection onto the coordinate space of the $i_1{}^{\mathrm{th}}$ design variable, $i_1 \in \{1, \ldots, d\}$. Here, this projection is denoted by proj_i, $i = 1, \ldots, d$, for which a general definition is provided in Section A.1. As i_1 can be any element from $\{1, \ldots, d\}$, intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, can be computed for all design variables. This means that $i_1 \in \{1, \ldots, d\}$, i.e., the design variable for which the first decision shall be made can be selected afterward. Similar to discussed above, maximizing these intervals can be used as a flexibility measure for dependent-decoupled design decisions. Note that in general other measures are conceivable here, too. By solving the mathematical optimization problems

$$\begin{aligned} &\underset{x_i^{\mathrm{l}}, x_i^{\mathrm{u}}}{\text{maximize}} \quad x_i^{\mathrm{u}} - x_i^{\mathrm{l}} \\ &\text{subject to} \quad [x_{i_1}^{\mathrm{l}}, x_{i_1}^{\mathrm{u}}] \subseteq \mathrm{proj}_i(\Omega_{\mathrm{c}}), \end{aligned} \tag{2.37}$$

$i = 1, \ldots, d$, optimal intervals are obtained that provide optimal flexibility for dependent-decoupled design decisions. For convex Ω_{c}, it holds $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}] = \mathrm{proj}_i(\Omega_{\mathrm{c}})$ with

$$\mathrm{proj}_i(\Omega_{\mathrm{c}}) = \left[\min_{x \in \Omega_{\mathrm{c}}} x_i, \max_{x \in \Omega_{\mathrm{c}}} x_i \right], \tag{2.38}$$

$i = 1, \ldots, d$. Here, the Cartesian product of the intervals $[x_i^{\mathrm{u}} - x_i^{\mathrm{l}}]$, $i = 1, \ldots, d$ forms box-shaped outer bounds of the complete system solution space. These bounds correspond to the minimum box that bounds the complete system solution space Ω_{c} and $[x^{\mathrm{l}}, x^{\mathrm{u}}]$ is called the *minimum volume outer box* in [95]. Minimum volume outer boxes are considered in various fields and approaches that compute solutions are presented, for example, in [62, 95, 102].
After $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, are calculated and $x_{i_1} \in [x_{i_1}^{\mathrm{l}}, x_{i_1}^{\mathrm{u}}]$ is selected, this procedure can be continued until a system design is obtained. In order to do so, the already selected design variables must be eliminated from Ω_{c} by treating their values as uncontrollable parameters.

In the following, only the intervals for the first decision are considered. Below, an example to compute these intervals for the first decision is given.

Example 13. Given the system design space $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$ and the performance function $f : \mathbb{R}^2 \to \mathbb{R},\ x \mapsto x_1 + 2x_2$ with threshold $f_{\mathrm{c}} = 2$ from Example 1. The box-shaped outer bounds of the complete system solution space for the first decision can be computed as $[0, 1] \times [0, 0.5]$ using methods from Chapter 4. It is visualized in Figure 18.

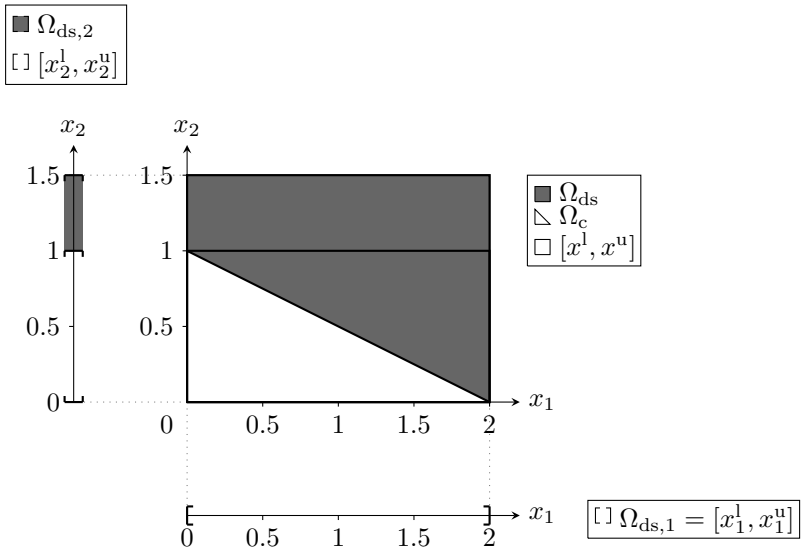


Figure 18 Box-shaped outer bounds of the complete system solution space $[x^{\mathrm{l}}, x^{\mathrm{u}}]$ for the first decision of Example 13.

Note again that there are various applications of maximum volume inner boxes and maximum volume outer boxes in literature. Here, they are used to decouple the design decisions. Besides that, it is described in Section 2.3 how maximum volume inner boxes are used to find a robust system design under interval-type uncertainty. Another possible application is, for example, given in [95, 113], in which maximum volume inner boxes and maximum volume outer boxes are used as an approximation of Ω_{c}, which facilitates the description and visualization of Ω_{c}. This aspect is taken up again in Section 5.3.

2.4.2. Mixed Approaches for Design Decisions under Absence of Uncertainty

A major shortcoming of using box-shaped system solution spaces for independent-decoupled design decisions is the loss of solution space, see, e.g., [141]. This states that box-shaped system solution spaces Ω can get small compared to the complete system solution space Ω_{c}. In this case, a lot of permissible system designs are not contained in Ω. For dependent-decoupled design decisions using box-shaped bounds of the complete system solution space, it can be argued that the development process is protracted as d subsequent decision must be made and hence, corresponding intervals must be computed d times. This motivates mixed approaches for design decisions which use combinations of coupled, independent-decoupled,

and dependent-decoupled approaches. As all mixed approaches still contain a decoupling of the design decisions, they provide flexibility for decoupled design decisions, too.

In [38], an approach that mixes coupled and independent-decoupled decisions is presented. If d is an even number, there are $\frac{d}{2}$ designers, each responsible for selecting two design variables from a corresponding set, independent of the selection of the designers. Hence, a coupled decision is required for the design variables x_i and x_{i+1} for which (x_i, x_{i+1}) is selected from a set $\Omega_{i,i+1}$ yielding $(x_i, x_{i+1}) \in \Omega_{i,i+1}$, $i = 1, 3, 5, \ldots, d-1$. In the case that d is an odd number, there are $\frac{d}{2} + 1$ designers and the design variable x_d must be selected from a one-dimensional set Ω_d. Again, these coupled design decisions can be made according to Section 2.3. In order to decouple the decisions between the single design variable pairs, the condition $\Omega_{1,2} \times \cdots \times \Omega_{d-1,d} \subseteq \Omega_\mathrm{c}$ must be always ensured to obtain a permissible system design $x \in \Omega_\mathrm{c}$ for even d. Then, $\Omega_{1,2} \times \cdots \times \Omega_{d-1,d}$ is a system solution space inside the complete system solution space. Similar to box-shaped system solution spaces, the flexibility for decoupled design decisions is optimized by maximizing the volume of $\Omega_{1,2} \times \cdots \times \Omega_{d-1,d}$. The corresponding optimization problem reads

$$\begin{aligned} \underset{\Omega_{1,2},\ldots,\Omega_{d-1,d}}{\text{maximize}} \quad & \mathrm{vol}(\Omega_{1,2} \times \cdots \times \Omega_{d-1,d}) \\ \text{subject to} \quad & \Omega_{1,2} \times \cdots \times \Omega_{d-1,d} \subseteq \Omega_\mathrm{c}. \end{aligned} \tag{2.39}$$

for even d. Approaches that solve problem (2.39) numerically are presented in [37, 38, 63] and an example is given below. In [21], this concept is extended by allowing couplings of arbitrary numbers of design variables.

Example 14. Given the system design space $\Omega_\mathrm{ds} = [0, 1.5]^3$ and the performance function $f : \mathbb{R}^3 \to \mathbb{R},\ x \mapsto -3x_1 - 2x_2 - 3x_3$ with threshold $f_\mathrm{c} = -9$ from Example 2. The optimal system solution space $\Omega_{1,2} \times \Omega_3$ can be computed using the methods mentioned above or the corresponding methods from Chapter 4. It is visualized in Figure 19.

In [129], an approach that mixes independent-decoupled and dependent-decoupled decisions is presented. Here, the design variables are divided into early decision variables collected in x_a and late decision variables collected in x_b, where $x = (x_\mathrm{a}, x_\mathrm{b})$ holds. First, the values of the early decision variables are selected independently of each other. In order to ensure that there is at least one permissible system design after the values of the early decision variables are selected, x_a must be located in the projection of the complete system solution space Ω_c onto the coordinate space of the early decision variables. This projection is denoted by proj_a. Within $\mathrm{proj}_\mathrm{a}(\Omega_\mathrm{c})$, any box $[x_\mathrm{a}^\mathrm{l}, x_\mathrm{a}^\mathrm{u}]$, called *box-shaped solution-compensation space* in [129], provides flexibility for independent-decoupled design decisions regarding the early decision variables. In order to optimize flexibility for these decisions, a maximum volume box-shaped

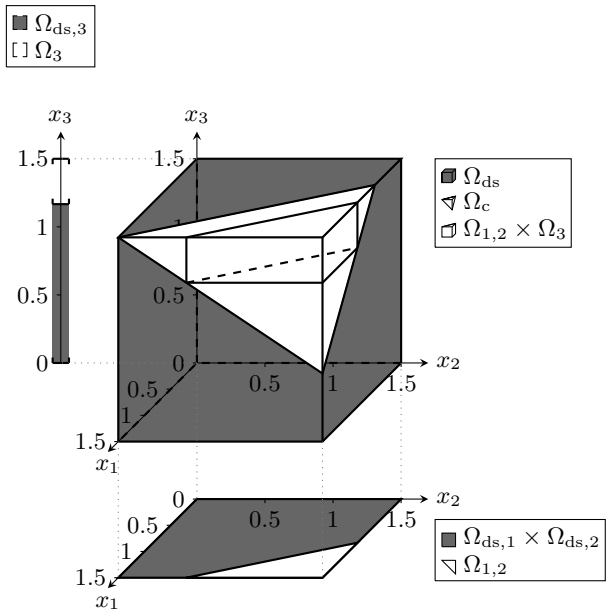


Figure 19 Optimal system solution space $\Omega_{1,2} \times \Omega_3$ for Example 14.

solution-compensation space is sought. The corresponding optimization problem reads

$$\begin{aligned} &\underset{x_{\mathrm{a}}^{\mathrm{l}}, x_{\mathrm{a}}^{\mathrm{u}}}{\text{maximize}} && \mathrm{vol}([x_{\mathrm{a}}^{\mathrm{l}}, x_{\mathrm{a}}^{\mathrm{u}}]) \\ &\text{subject to} && [x_{\mathrm{a}}^{\mathrm{l}}, x_{\mathrm{a}}^{\mathrm{u}}] \subseteq \mathrm{proj}_{\mathrm{a}}(\Omega_{\mathrm{c}}). \end{aligned} \tag{2.40}$$

Approaches that solve problem (2.40) numerically are presented in [128, 129] and an example is given in the following. Compared to dependent-decoupled design decisions presented in Section 2.4, this approach fixes a priori, which design variables must be specified first, i.e., the design variables collected in x_{a}. Thus, the design variables in x_{b} can be neglected for the first decision.

Example 15. Given the system design space $\Omega_{\mathrm{ds}} = [0, 1.5]^3$ and the performance function $f : \mathbb{R}^3 \to \mathbb{R},\ x \mapsto -3x_1 - 2x_2 - 3x_3$ with threshold $f_{\mathrm{c}} = -9$ from Example 2. The two design variables x_1 and x_2 are chosen to be the early decision variables, i.e., $x_{\mathrm{a}} = (x_1, x_2)$, and x_3 is chosen to be the late decision variable, i.e., $x_{\mathrm{b}} = x_3$. The optimal box-shaped solution-compensation space of the early decision variables $[x_{\mathrm{a}}^{\mathrm{l}}, x_{\mathrm{a}}^{\mathrm{u}}]$ can be computed using the methods mentioned above or the corresponding methods from Chapter 4. It is visualized in Figure 20.

Furthermore, there are also mixed approaches in literature which do not use optimization strategies to provide flexibility for decoupled design decisions. In [82], for example, an analysis of the system performance functions is used to define early decision variables, collected in x_{a}. When being fixed, these provide automatically a mix of coupled and independent-decoupled

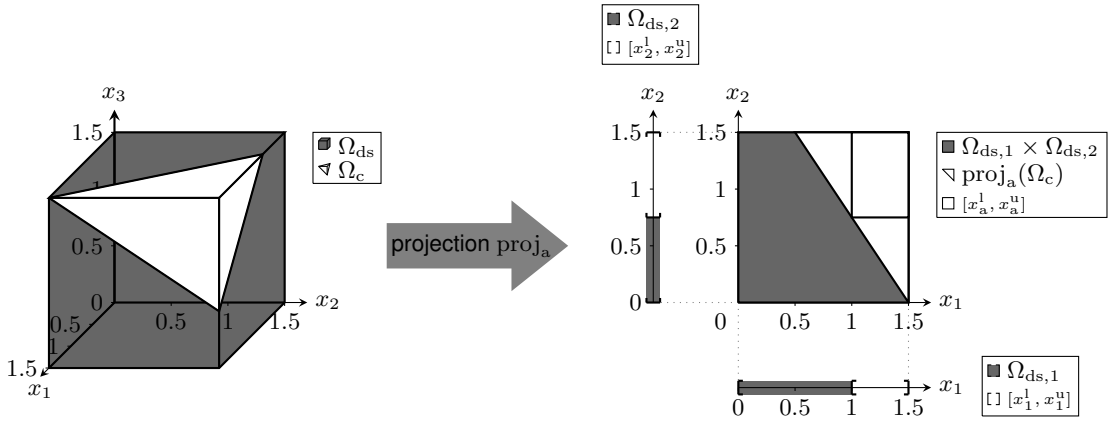


Figure 20 Optimal box-shaped solution-compensation space $[x_{\mathrm{a}}^{\mathrm{l}}, x_{\mathrm{a}}^{\mathrm{u}}]$ of Example 15.

design decisions for late decision variables, collected in x_{b}. By selecting the values of the design variables in x_{a}, the design decisions for the values of the variables in x_{b} become partially decoupled. Note that because of selecting the values of the early decision variables a priori, dependent-decoupled design decisions are involved here as well.

Overall, the grouping of design variables between which the design decisions shall be decoupled cannot generally be prearranged in the presented mixed approaches. In particular, this is the case for the approach of [82]. However, such a grouping of design variables is often desired in systems design as a system is usually composed of components, which comprise specific design variables, enabling the deployment of component designers. Hence, extensions of the mixed approaches from [38] and [129] which incorporate the composition of the underlying system are proposed in the next chapter. This allows component designers to make design decisions that are both decoupled between the single components and coupled for each component itself.

2.4.3. Decoupled Design Decisions under Uncertainty

The approaches for decoupled design decisions of this section can also be used for design decisions under uncertainty. The following considerations refer to those approaches which completely decouple the decisions for the single design variables, i.e., in which intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, are computed. Nevertheless, the considerations can also be transferred one-to-one to the mixed approaches for decoupled design decisions.

First, uncertainties in controllable variables are considered, assuming there is complete knowledge about the uncertainty magnitudes. In the case of independent-decoupled design decisions, the values of the target design variables $\check{x}_i$ can be selected such that the realized values of the design variables x_i are contained in $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ if the uncertainty magnitudes are small enough for $i = 1, \ldots, d$. For interval-type uncertainty, this can be ensured in the worst-

or best-case and for fuzzy-type uncertainty, this can be ensured with a particular necessity or possibility. The same holds for the target design variable $\check{x}_{i_1}$, $i_1 \in \{1, \ldots, d\}$ in the case of dependent-decoupled design decisions. The decision to select the value of a target design variable can be made according to the approach for coupled design decisions, presented in Section 2.3. Note however that this is not always guaranteed for the subsequently selected design variable x_{i_2}, $i_2 \in \{1, \ldots, d\} \backslash \{i_1\}$.
The larger the uncertainty magnitudes in controllable variables for worst-case scenarios or large necessities, the more centered in $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, must the values of the target design variables be chosen for independent-decoupled design decisions to ensure a permissible design. The centers of $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ are the target design variables in $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ which tolerate the largest uncertainties in controllable variables, compare Section 2.3, $i = 1, \ldots, d$. Thus, the values of these centers should always be chosen for the target design variables in case of independent-decoupled design decisions under unknown uncertainty. However, if only the uncertainty magnitude of one design variable is large enough such that the realized values of the design variable is outside the corresponding interval, i.e., $x \notin [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ for $i \in \{1, \ldots, d\}$, the whole system design is located outside the box-shaped system solution space, i.e., $x \notin [x^{\mathrm{l}}, x^{\mathrm{u}}]$. This means that there is a chance that the system design is non-permissible. Especially, if one of the intervals forming the box-shaped system solution space is small compared to the others, the chance of obtaining a non-permissible system design increases, when there are comparably large uncertainty magnitudes in the design variables. This was already discussed in Section 2.3. This problem could be avoided when using problem (2.32) to compute box-shaped system solution spaces as proposed in [43]. For small uncertainty magnitudes, however, this would lead to minor flexibility for decoupled design decisions. Similar arguments can be presented for dependent-decoupled design decisions in which the target design variables are selected one after the other.

Uncertainties in uncontrollable parameters are more difficult to handle with the presented approaches than uncertainties in controllable variables as the intervals $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$, do not incorporate any information about the uncontrollable parameters in p. Furthermore, the intervals are computed for fixed $p \in \mathbb{R}^d$, i.e., for nominal values $\check{p} \in \mathbb{R}^d$ if uncertainty is present. If p is uncertain, the realized system design might be non-permissible, although it is within the complete system solution space Ω_{c} computed with $\check{p} \in \mathbb{R}^d$. It might be assumed that the design variables near the center of $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ tolerate larger uncertainties in uncontrollable parameters than design variables near the bounds of $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$, $i = 1, \ldots, d$. However, there is no guarantee for this, as the effects of uncertain p on Ω_{c} are not a priori clear.
Besides being able to treat uncertainties in controllable variables, uncertainties in the constraints and also the system cost function which are not due to uncertainties in controllable variables or uncontrollable parameters can be efficiently treated a posteriori if the corresponding uncertain function depends on only one design variable, e.g., $f_j(x) = f_j(x_i)$ or $c(x) = c(x_i)$ for $i \in \{1, \ldots, d\}$, $j \in \{1, \ldots, m\}$. For a change in such a constraint which only depends on x_i, $i \in \{1, \ldots, d\}$, the corresponding interval $[x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ can be updated immedi-

ately for both independent-decoupled and dependent-decoupled design decisions and so the selection of the corresponding design variable. The same holds for the introduction of new constraints or the cancellation of old constraints. Note that in these cases, also uncertainties in uncontrollable parameters can be treated efficiently.

As stated above, uncertainties in the cost function and the constraints are not further considered in this thesis. Hence, the considerations are limited to uncertainties in controllable variables or uncontrollable parameters in the following. In order to establish an uncertainty consideration in uncontrollable parameters and to balance uncertainty and flexibility, new strategies to provide optimal flexibility for decoupled design decisions which include a-priori uncertainty considerations are presented in the next chapter.

3. METHODOLOGY: Component Solution Spaces (CSS)

This chapter proposes a new methodology for decoupled design decisions in systems engineering that is geared towards the component structure. For this purpose, the hierarchical levels of a system are investigated in more detail and it is discussed how flexibility for decoupled design decisions, i.e., component design can be provided. In order to obtain optimal flexibility for component design, optimal component solution spaces are proposed for which different problem statements for an independent and dependent decoupling are considered. Furthermore, it is distinguished between box-shaped and arbitrarily-shaped component solution spaces which comprise different geometric shapes for the component solution spaces. Moreover, the methodology is extended by integrating interval- and fuzzy type uncertainties in controllable variables and uncontrollable parameters.

3.1. Definitions and Assumptions

This section shows how the definitions of mathematical design models can be extended for systems that are composed of components, i.e., two-level systems. Furthermore, extensions to multi-level systems are considered, and it is discussed, how flexibility for decoupled design decisions between the components can be provided using the proposed definitions and assumptions. Note that all system examples of the previous chapter may be understood as a two- or a multi-level system, using the proposed extensions.

3.1.1. Two-Level Systems

In this thesis, systems which are composed of n components are considered, $n \leq d$. The following definitions are extensions of the definitions for systems design given in Section 2.1. Recall that there is a system design $x \in \mathbb{R}^d$. It is assumed that each design variable x_i, $i = 1, \ldots, d$, belongs to one of the components, i.e., there is no design variable that is shared between two components. In general, a component can, for example, be a structural component or anything else that can be described by a comprehensible grouping of design variables.

The k^{th} component comprises d^k design variables $x_i^k \in \mathbb{R}$, $i = 1, \ldots, d^k$. Note here that the superscript k states an upper index instead of an exponent. This is used throughout this thesis. The design variables $x_i^k \in \mathbb{R}$, $i = 1, \ldots, d^k$, are collected in a vector $x^k \in \mathbb{R}^{d^k}$ named *component design* of the k^{th} component and it holds $\sum_{k=1}^{n} d^k = d$. Thus, the k^{th} component has d^k degrees of freedom which sum up to the d degrees of freedom of the system. When all component designs are specified, a system design is obtained. For the system design, it

holds

$$x = (x^1, \ldots, x^n) \tag{3.1}$$

where the entries of x can be expressed by the entries of x^k via

$$x_{(d^1+\cdots+d^{k-1}+i)} = x_i^k, \tag{3.2}$$

$i = 1, \ldots, d^k$, $k = 1, \ldots, n$. In Figure 21, such a system is visualized.

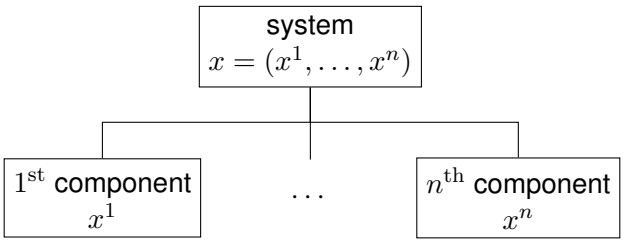


Figure 21 A system composed of n components. The design variables are grouped as system design at the system level and as component designs at the component level.

This two-level hierarchy, consisting of a system and component level, is the reason why the system is also called a *two-level system*. In Figure 21, these different levels are emphasized. The following example shows how component designs form a system design and vice versa.

Example 16. Given a system composed of two components. The component design of the first component contains two design variables, i.e., $d^1 = 2$, and is selected as $x^1 = (2, 1)$. The component design of the second component contains one design variable, i.e., $d^2 = 1$, and is selected as $x^2 = 1.5$. Therefore, the system design is $x = (2, 1, 1.5)$ with $d = 3$. The component designs are visualized together with the corresponding system design in Figure 22.

As all component designs together form a system design, the system performance functions can be used to map the component designs to the system responses. Besides the m system performance functions, m^k *component performance functions* g_j^k, $j = 1, \ldots, m^k$ can be defined additionally for every component, $k = 1, \ldots, n$. They map each component design to *component responses* $y_j^k \in \mathbb{R}$, $j = 1, \ldots, m^k$, via

$$g^k : \mathbb{R}^{d^k} \times \mathbb{R}^q \mapsto \mathbb{R}^{m^k},\ (x^k, p) \mapsto y^k = g^k(x^k, p) \tag{3.3}$$

where $g^k = (g_1^k, \ldots g_{m^k}^k)$ and $y^k = (y_1^k, \ldots, y_{m^k}^k)$ holds. Their hierarchy within the system is visualized in Figure 23.

Similar to the system responses, the component responses might also be restricted to subsets

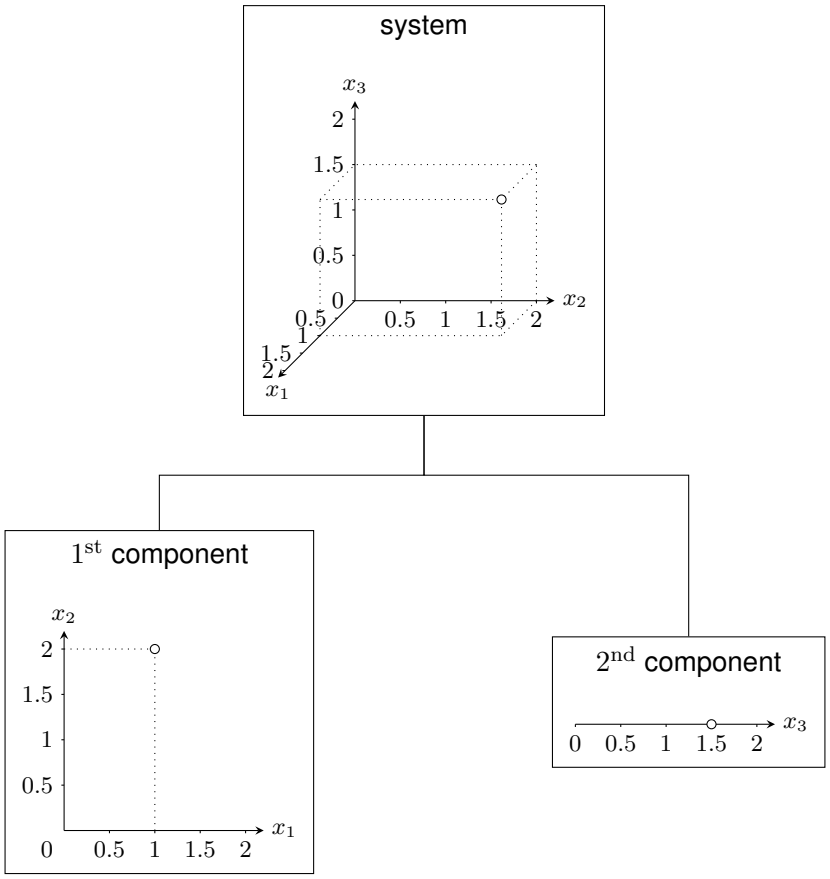


Figure 22 Component designs and corresponding system design of Example 16.

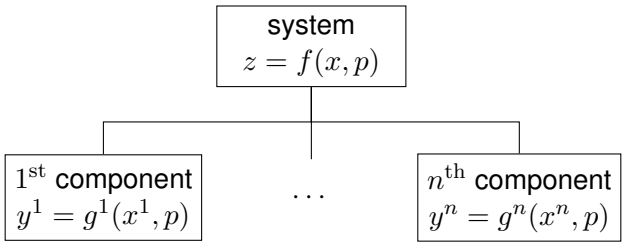


Figure 23 Hierarchy of the responses and performance functions with system responses and system performance functions at the system level and component responses and component performance functions at the component level.

of $\mathbb{R}$. Moreover, g^k, $k = 1, \ldots, n$, are defined such that they may depend on all uncontrollable parameters p_l, $l = 1, \ldots, q$, defined in Section 2.1. However, this does not need to be the case and corresponding uncontrollable parameters can also be excluded in the definition of component performance functions. If uncontrollable parameters are not considered explicitly in a problem statement, the corresponding argument in the component performance functions is neglected and it is written $y^k = g^k(x^k)$, i.e., $g^k = g^k(\cdot, p)$ for $p \in \mathbb{R}^q$.
In contrast to system performance functions, which are given by the system design model and usually have a physical meaning, component performance functions are not always predefined. In this thesis, they are mainly used as a helpful theoretical construct that may lack a physical meaning. Thus, the relation between y^k, $k = 1, \ldots, n$, and z is not a priori clear. Nevertheless,

a simple relation like $z = y^1 + \cdots + y^n$ can be established in some cases, i.e.,

$$f(x,p) \quad = \quad g^1(x^1,p) + \cdots + g^n(x^n,p) \tag{3.4}$$

with $m^k = m$, $k = 1,\ldots,m$. This can be done, for example, for linear system performance functions. In general, however, a property like Equation (3.4) cannot be assumed.

The same holds for *component cost functions* c^k, which can be defined as

$$c^k : \mathbb{R}^{d^k} \times \mathbb{R}^q \to \mathbb{R},\ (x^k,p) \mapsto c^k(x^k,p), \tag{3.5}$$

$k = 1,\ldots,n$. Again, if uncontrollable parameters are not considered explicitly in a problem statement, the corresponding argument in the component cost functions is neglected and it is $c^k = c^k(\cdot,p)$ for $p \in \mathbb{R}^q$. As discussed in Section 2.4, however, cost functions are not considered explicitly in the following as they are not required to decouple the design decisions. Rather, component cost functions help to optimize component designs after the decoupling.

Due to constraints on design variables, the design variables are bounded by lower and upper bounds. At the component level, it holds $x_{\mathrm{ds},i}^{\mathrm{l},k} \leq x_i^k \leq x_{\mathrm{ds},i}^{\mathrm{u},k}$ or $x_i^k \in [x_{\mathrm{ds},i}^{\mathrm{l},k}, x_{\mathrm{ds},i}^{\mathrm{u},k}]$, $i = 1,\ldots,d^k$, $k = 1,\ldots,n$. Therefore, it is in vector notation $x_{\mathrm{ds}}^{\mathrm{l},k} \leq x^k \leq x_{\mathrm{ds}}^{\mathrm{u},k}$ componentwise or $x^k \in [x_{\mathrm{ds}}^{\mathrm{l},k}, x_{\mathrm{ds}}^{\mathrm{u},k}]$. Each d^k-dimensional interval $[x_{\mathrm{ds}}^{\mathrm{l},k}, x_{\mathrm{ds}}^{\mathrm{u},k}]$ is also called *component design space* of the k^{th} component and is denoted by Ω_{ds}^k, i.e.,

$$\Omega_{\mathrm{ds}}^k \quad = \quad [x_{\mathrm{ds}}^{\mathrm{l},k}, x_{\mathrm{ds}}^{\mathrm{u},k}], \tag{3.6}$$

$k = 1,\ldots,n$. The Cartesian product of the component design spaces forms the system design space.

In contrast to system performance thresholds $f_{\mathrm{c}}(p) \in \mathbb{R}^m$, which are given for the system responses $z \in \mathbb{R}^m$, it is assumed that there are no a-priori *component performance thresholds* $g_{\mathrm{c}}^k(p) \in \mathbb{R}^{m^k}$ for the component responses, $p \in \mathbb{R}^q$. Nevertheless, requirements affecting single components can always be formulated as system requirements for which the system performance functions only depend on the design variables of the corresponding component.

3.1.2. Multi-Level Systems

As discussed in Chapter 1, each component can be considered as a system itself. Thus, the component designs of the two-level system may depend on further variables, i.e., lower-level design variables. Again, these lower-level design variables can be assigned to lower-level components for which it is assumed that no lower-level design variable belongs to more than one lower-level component. The lower-level component designs can be collected in lower-level system designs which are mapped to the original component designs by lower-level system

performance functions. Hence, the original component designs form the system responses of the corresponding subsystems.
This can be extended arbitrarily yielding a *multi-level system* with n_{hl} hierarchical levels. The structure of this multi-level system can be described by a rooted tree, see, e.g., [115] for more information on rooted trees. An example of a multi-level system can be found in Section 5.1. In the following, the relations between the single entities of multi-level systems are considered mathematically for which a notation scheme is introduced. Furthermore, it is discussed how flexibility for decoupled design decisions for a particular level can be provided.

First, the structure of a multi-level system is investigated further. The uppermost system is located at level n_{hl}, their components at level $n_{\mathrm{hl}}-1$ forming systems themselves, the components of these systems at level $n_{\mathrm{hl}}-2$, and so on. There is a unique path along the associated rooted tree from the system at level n_{hl} to each component of a system at level i_{hl} which can be described by the vector $(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}})\in\mathbb{N}^{n_{\mathrm{hl}}-i_{\mathrm{hl}}+1}$, $i_{\mathrm{hl}}\in\{2,\dots,n_{\mathrm{hl}}\}$. Here, the first entry refers to the ${k_{n_{\mathrm{hl}}}}^{\mathrm{th}}$ component of the system at level n_{hl}, the second entry to the ${k_{n_{\mathrm{hl}}-1}}^{\mathrm{th}}$ component of the system at level $n_{\mathrm{hl}}-1$ which belongs to the ${k_{n_{\mathrm{hl}}}}^{\mathrm{th}}$ component of the system at level n_{hl}, and so on. Then, a component design of a system at level i_{hl} can be denoted as $x^{(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}})}$ where the index states the mentioned path. Accordingly, the system performance functions of this system are denoted by $f^{(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1})}$ and their system responses by $z^{(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1})}$, $i_{\mathrm{hl}}\in\{2,\dots,n_{\mathrm{hl}}-1\}$. For $i_{\mathrm{hl}}=n_{\mathrm{hl}}$, no upper index is used. Neglecting uncontrollable parameters for $i_{\mathrm{hl}}\in\{2,\dots,n_{\mathrm{hl}}-1\}$, it holds

$$z^{\left(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1}\right)}=f^{\left(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1}\right)}\left(x^{(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1},1)},\dots,x^{\left(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1},n_{n_{\mathrm{hl}}}\right)}\right). \tag{3.7}$$

and furthermore

$$x^{\left(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1}\right)}=z^{\left(k_{n_{\mathrm{hl}}},\dots,k_{i_{\mathrm{hl}}+1}\right)} \tag{3.8}$$

where the lower-level responses correspond to the component designs of the consecutive level like discussed above. Note that the entire multi-level system is defined without component performance functions. These could be additionally introduced for every component. In Figure 24, the defined relations are visualized.

Moreover, note that the multi-level system presented here is a simplified multi-level system. At each level, only interactions between components belonging to one subsystem are assumed. Interactions between different levels are also not defined directly but could be established if there is a common path along the rooted tree to the responses of interest.
Furthermore, the following further assumptions are made for multi-level systems here:

- The multi-level system model is built up from the top to the bottom, i.e., the system model of the system at level n_{hl} is defined before lower-level system models are considered. Besides assigning design variables, system performance functions, and system responses

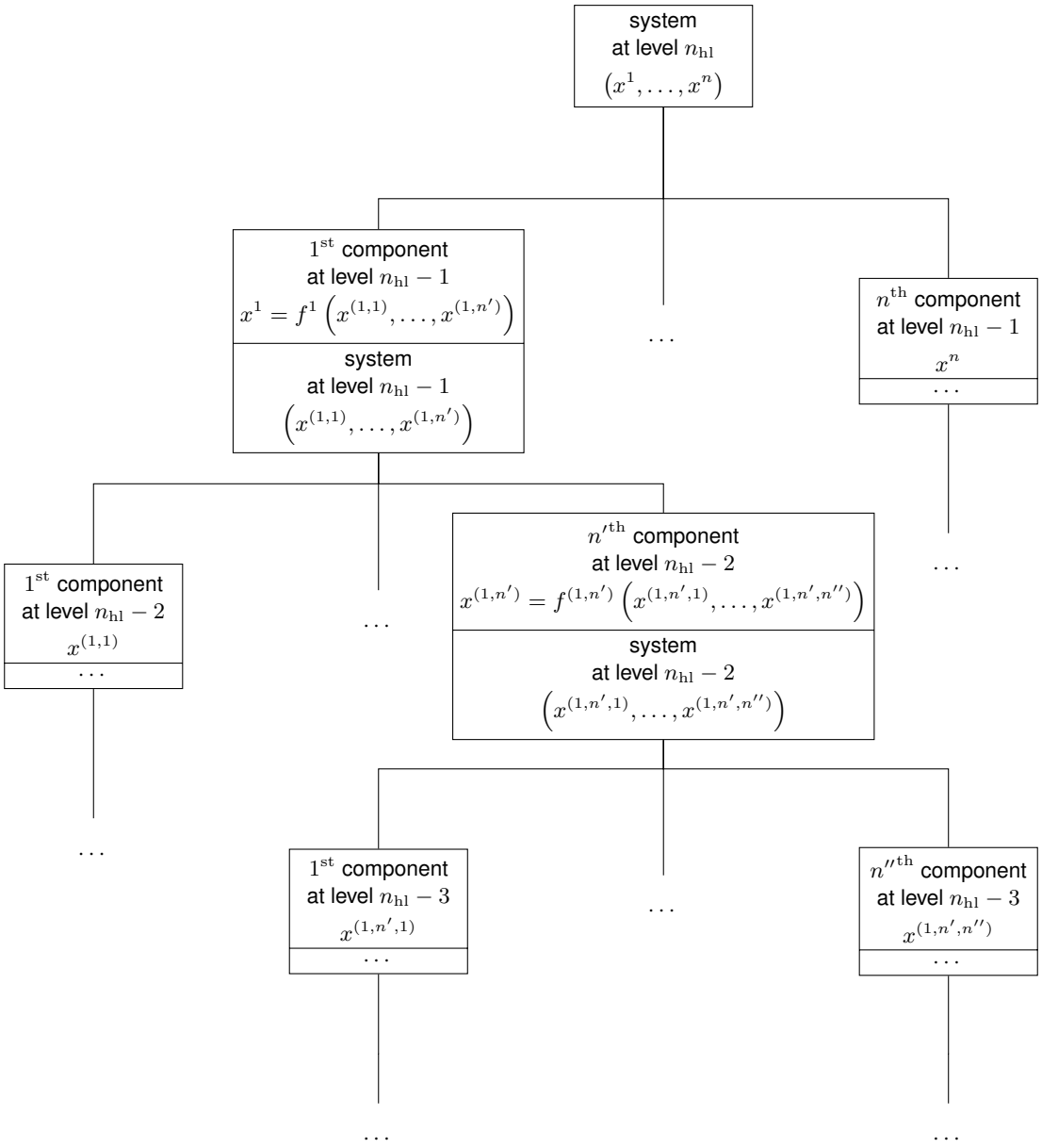


Figure 24 A multi-level system described by a rooted tree with corresponding component designs, system designs, system performance functions, and system responses. Here, n denotes the number of components of the system at level n_{hl}, n' the number of components of the system at level $n_{\mathrm{hl}} - 1$ with the path 1, and n'' the number of components of the system at level $n_{\mathrm{hl}} - 2$ with the path $(1, n')$. Note that the notations n' and n'' are chosen for reasons of simplicity. In general, notations for the number of components that indicate the path to their corresponding system could be used, too.

to system quantities, this includes the decomposition of the system into components which are located at the next lower level.

- System performance requirements are only defined for the system at level n_{hl}. Recall that possible system performance requirements for lower-level systems can always be expressed in terms of system performance requirements for the system at level n_{hl}.
- It is not necessary that all levels of the multi-level system are completely defined in the early design phase. Here, only the system models of the upper levels might be defined, and lower-level system models might be lacking.

- For every defined system model, system and component design spaces are available. If these design spaces are not known a priori and lower-level system models exist, they can be derived from the range of the corresponding system performance functions. If lower-level system models do not exist, they must be assumed.
- All system and component designs within their design spaces can be designed, i.e., realized. Furthermore, the system responses of a lower-level system are assumed to be approximately equally distributed in the corresponding component design space. This avoids an a-priori preference for any component design at the corresponding level.

As it is assumed that the system performance requirements and the system design space are given for level n_{hl}, the complete system solution space for level n_{hl} can be stated. Then, different types of design decisions are conceivable to obtain a permissible system design at level n_{hl}, i.e., a system design within the corresponding complete system solution space. For example, coupled design decisions can be made at level n_{hl} yielding component designs at level $n_{\mathrm{hl}} - 1$. As these component designs depend on lower-level design variables, the values of the lower-level design variables must be selected such that they are mapped onto the component designs by their corresponding system performance functions. In doing so, usually no flexibility in selecting the values of the lower-level design variables remains. Furthermore, coupled design decisions can be made directly for lower-level design variables at level $i_{\mathrm{hl}} + 1$, $i_{\mathrm{hl}} \in \{1, \ldots, n_{\mathrm{hl}} - 1\}$. For this purpose, system performance functions that map these lower-level design variables to the system responses at the level n_{hl} are required. These can be obtained by a function composition of the corresponding upper system performance functions if their system models are defined. In this context, all upper-level systems are reduced to a two-level system. Note that the total number of components at level i_{hl} remains here, $i_{\mathrm{hl}} \in \{1, \ldots, n_{\mathrm{hl}} - 1\}$.
Accordingly, the design decisions could be either decoupled between the components at level $n_{\mathrm{hl}} - 1$ first or decoupled between the components at level i_{hl}, $i_{\mathrm{hl}} \in \{1, \ldots, n_{\mathrm{hl}} - 1\}$ immediately. If flexibility for component design at level $(n_{\mathrm{hl}} - 1)$ is provided, the design decisions can also be made for components at level $(n_{\mathrm{hl}} - 2)$ as the component designs at level $n_{\mathrm{hl}} - 1$ form responses of the design variables at level $n_{\mathrm{hl}} - 2$. This decoupling can be continued until level i_{hl} which yields flexibility in component design at this level, $i_{\mathrm{hl}} \in \{1, \ldots, n_{\mathrm{hl}} - 1\}$. Moreover, by reducing the upper-level systems as described above, flexibility for component design at level i_{hl} can also be directly provided, $i_{\mathrm{hl}} \in \{1, \ldots, n_{\mathrm{hl}} - 1\}$. Note again that all relevant system models must be defined here.
In the following, decoupled design decisions between components are further investigated. As the focus on a multi-level system can always be put temporarily on two levels when considering design decisions, two-level systems are taken into account exclusively for which the corresponding notations from above are used.

3.1.3. Decoupled Designs Decision between Components

Instead of having one system designer who makes design decisions for a system design $x \in \mathbb{R}^d$, there are n *component designers* for decoupled design decisions between the components. Here, the component designers select their corresponding component designs $x^k \in \mathbb{R}^{d^k}$, $k = 1, \ldots, n$. Regarding the classification presented in Section 2.1, decoupled design decisions between the components correspond to a mixed approach of decoupled and coupled design decisions. In contrast to this, the mixed approaches presented in Section 2.4 are not geared towards the component structure of the system. For an efficient approach, however, the design process must be decomposed by enabling separate teams to design components with maximum flexibility, see [127]. The decomposition methodology proposed in this thesis can also be viewed as a hybrid approach for decoupled design decisions, as it takes the hierarchical structure of the system into account. Note that there are two extreme cases: one with $n = 1$ for which the design decisions for the system are made completely coupled, see Section 2.3, and one with $n = d$ for which the design decisions are made completely decoupled, see Section 2.4.

For coupled design decisions, the system designer selects x from the complete system solution space $\Omega_\mathrm{c} \subset \mathbb{R}^d$ in order to obtain a permissible design. After decomposing these coupled design decisions into decoupled design decisions between components, each of the n component designers is responsible for selecting x^k from a set $\Omega^k \subset \mathbb{R}^{d^k}$ for $k = 1, \ldots, n$. The sets Ω^k, $k = 1, \ldots, n$, are called *component solution spaces*, subsequently also abbreviated as CSS. The goal of the remaining thesis is to consider optimal CSS which provide optimal flexibility for decoupled design decisions at the component level The idea of decoupling the design decision from the system level to the component level is illustrated in Figure 25.

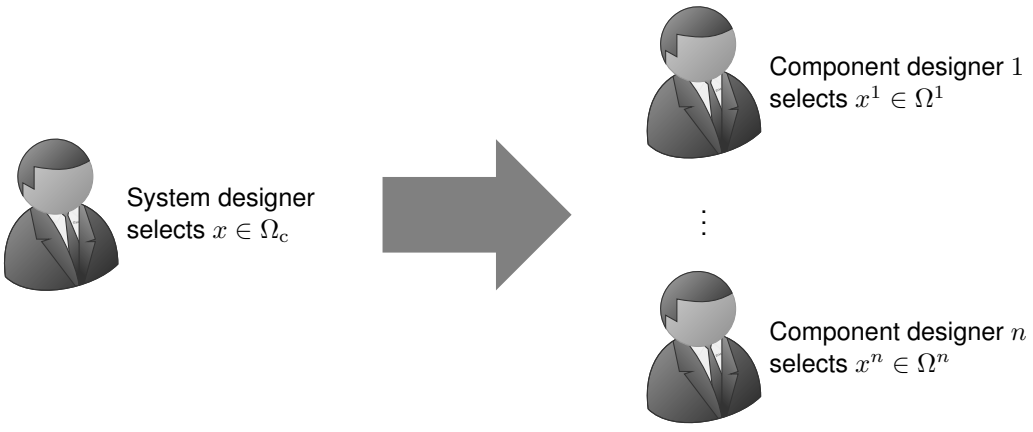


Figure 25 Decoupling of the design decisions from the system level with one system designer (left) to the component level with n component designers (right).

If $x^k \in \Omega^k$ holds, x^k is said to be *permissible*, otherwise, it is said to be *non-permissible*. The simplest way to obtain component solution spaces is to define lower bounds $x_i^{\mathrm{l},k}$ and upper bounds $x_i^{\mathrm{u},k}$ for the design variables of each component, $i = 1, \ldots, d^k$, $k = 1, \ldots, n$. In vector

notation, it is $x^{\mathrm{l},k} = (x_1^{\mathrm{l},k}, \ldots, x_{d^k}^{\mathrm{l},k})$ and $x^{\mathrm{u},k} = (x_1^{\mathrm{u},k}, \ldots, x_{d^k}^{\mathrm{u},k})$. Then, Ω^k is formed by the d^k-dimensional interval $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$, i.e.,

$$\Omega^k = [x^{\mathrm{l},k}, x^{\mathrm{u},k}] \tag{3.9}$$

where $\Omega^k \subseteq \Omega_{\mathrm{ds}}^k$ must be guaranteed for $k = 1, \ldots, n$. This type of component solution spaces is also referred to as *box-shaped CSS* in the following. Note that this approach completely decouples the design decisions for each component, yielding independent-decoupled instead of coupled design decisions for the single components.
Another approach to obtain component solution spaces Ω^k, $k = 1, \ldots, n$, is to define *component performance requirements* for each component by selecting performance thresholds $g_{\mathrm{c},j}^k(p)$ for component performance functions g_j^k, $j = 1, \ldots, m^k$, which can be either given or defined, $p \in \mathbb{R}^q$. This means $g_j^k(x^k, p) \leq g_{\mathrm{c},j}^k(p)$, $j = 1, \ldots, m^k$, or in vector notation $g^k(x^k, p) \leq g_{\mathrm{c}}^k(p)$, $p \in \mathbb{R}^q$, $k = 1, \ldots, n$. Thus, it is

$$\Omega^k \quad = \quad \{x^k \in \Omega_{\mathrm{ds}}^k \mid g^k(x, p) \leq g_{\mathrm{c}}^k(p)\} \tag{3.10}$$

for $p \in \mathbb{R}^q$ and $k = 1, \ldots, n$. If uncontrollable parameters are not considered explicitly in a problem statement, the corresponding argument in the system performance functions is often neglected, i.e., $g_{\mathrm{c}}^k = g_{\mathrm{c}}^k(p)$ for $p \in \mathbb{R}^q$. Note that the definition for CSS given in Equation (3.9) is a special case of the definition given in Equation (3.10) with

$$g^k(x^k) = \begin{pmatrix} -1 & & 0 \\ & \ddots & \\ 0 & & -1 \\ 1 & & 0 \\ & \ddots & \\ 0 & & 1 \end{pmatrix} x^k \leq \begin{pmatrix} -x^{\mathrm{l},k} \\ \\ x^{\mathrm{u},k} \end{pmatrix} = g_{\mathrm{c}}^k. \tag{3.11}$$

The question, how to choose $x^{\mathrm{l},k}$ and $x^{\mathrm{u},k}$ for Equation (3.9) or $g_{\mathrm{c}}^k(p)$, including $g^k(\cdot, p)$, for Equation (3.10), $p \in \mathbb{R}^q$, $k = 1, \ldots, n$, is addressed in the next sections. For variable $g^k(\cdot, p)$, $p \in \mathbb{R}^q$, $k = 1, \ldots, n$, the type of CSS is also referred to as *arbitrarily-shaped CSS*. Furthermore, it is distinguished between independent CSS that provide flexibility for independent-decoupled design decisions and dependent CSS that provide flexibility for dependent-decoupled design decisions. Note that the subsequent investigations are made for fixed $p \in \mathbb{R}^q$ first. An overview of where the corresponding problem statements are introduced is given in Table1.

Table 1 Overview of the proposed methodology for CSS under absence of uncertainty.

CSS	independent	dependent
box-shaped	Subsection 3.2.2	Subsection 3.3.2
arbitrarily-shaped	Subsection 3.2.3	Subsection 3.3.3

3.2. Independent CSS

This section considers mathematical problem statements for computing independent component solution spaces that provide optimal flexibility for independent-decoupled design decisions between the components. It is distinguished between box-shaped and arbitrarily-shaped independent CSS which comprise different geometric shapes. The corresponding problem statements were already published in [27] by the author of this thesis.

3.2.1. Problem Statement

In order to provide flexibility for independent-decoupled design decisions between the components, component solution spaces Ω^k, $k = 1, \ldots, n$ are required for which every component designer can select a permissible component design independently of the other component designers. Thus, for any selected component designs $x^k \in \Omega^k$, $k = 1, \ldots, n$, it must be guaranteed that the resulting system design is contained within the complete system solution space, i.e., $(x^1, \ldots, x^n) \in \Omega_\mathrm{c}$. The set of all system designs that can be realized by selecting only permissible component designs is built by the Cartesian product of the component solution spaces. It is denoted by $\Omega \subset \mathbb{R}^d$ with

$$\Omega = \Omega^1 \times \cdots \times \Omega^n. \tag{3.12}$$

If Ω is a subset of Ω_c, i.e.,

$$\Omega \subseteq \Omega_\mathrm{c}, \tag{3.13}$$

the resulting system design for any selected combination of permissible component designs is a permissible system design. In this case, Ω is also called a *system solution space*. Condition (3.13) is a necessary and sufficient condition to obtain flexibility for independent-decoupled design decisions between the components. Therefore, the sets Ω^k, $k = 1, \ldots, n$, are called *independent component solution spaces*. In Figure 26, their relationship is shown from a system perspective.

In general, there are multiple combinations of sets for independent CSS that fulfill condition (3.13). Among these, optimal component solution spaces which provide the most flexibility for independent-decoupled design decisions are preferred. Following up the discussion in Section 2.4, this flexibility for component design can be quantified by the volume of $\Omega^1 \times \cdots \times \Omega^n$, i.e.,

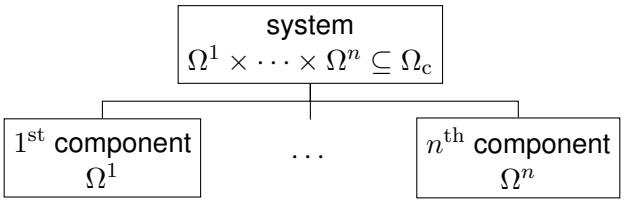


Figure 26 Independent component solution spaces at the component level and their corresponding system solution space at the system level.

$\mathrm{vol}(\Omega^1 \times \cdots \times \Omega^n)$. Thus, optimal independent CSS can be computed by searching for the CSS with maximum volume for which condition (3.13) is ensured. This can be expressed as a mathematical optimization problem which reads

$$\begin{aligned} \underset{\Omega^1,\ldots,\Omega^n}{\text{maximize}} \quad & \mathrm{vol}(\Omega^1 \times \cdots \times \Omega^n) \\ \text{subject to} \quad & \Omega^1 \times \cdots \times \Omega^n \subseteq \Omega_\mathrm{c}. \end{aligned} \tag{3.14}$$

Note again that in general other measures to optimize independent CSS are conceivable as well. Following up the discussion in Section 2.4, it would be also possible to optimize the minimum volume of Ω^k, weighted by a factor for $k = 1, \ldots, n$, or $\prod_{k=1}^{n} \mathrm{vol}(\Omega^k)$, where the single volumes are weighted by different exponents, for example.

In order to calculate the volume of $\Omega^1 \times \cdots \times \Omega^n$, it must be guaranteed that $\Omega^k \subset \mathbb{R}^{d^k}$, $k = 1, \ldots, n$, are Borel sets, compare [21]. If no restrictions on the geometric shapes of the component solution spaces are put, the CSS can take arbitrary shapes. As discussed above, they are therefore called arbitrarily-shaped CSS. Opposite to that, the geometric shapes of the component solution spaces may be predefined. In this case, an additional optimization constraint for the geometric shape must be introduced in problem (3.14).
Overall, the main challenge for solving problem (3.14) is that the optimization variables are sets instead of elements in real coordinate space. An approach to tackle this problem can be to transform or simplify the problem to a problem with real-valued optimization variables. This is discussed subsequently, and further simplifications are given in Section 4.2. Then, the obtained optimization problem belongs to the class of design centering problems, defined in [64].
After optimal independent CSS are computed, every component designer can select a component design within the corresponding CSS independently of the others. The design decision for a particular $x^k \in \Omega^k$ can be made as discussed in Section 2.3 using component cost functions. The associated workflow is visualized in Figure 27.

An important property of independent CSS is that after obtaining a system design, changes in the single component designs are still allowed as long as $x^k \in \Omega^k$ can be guaranteed

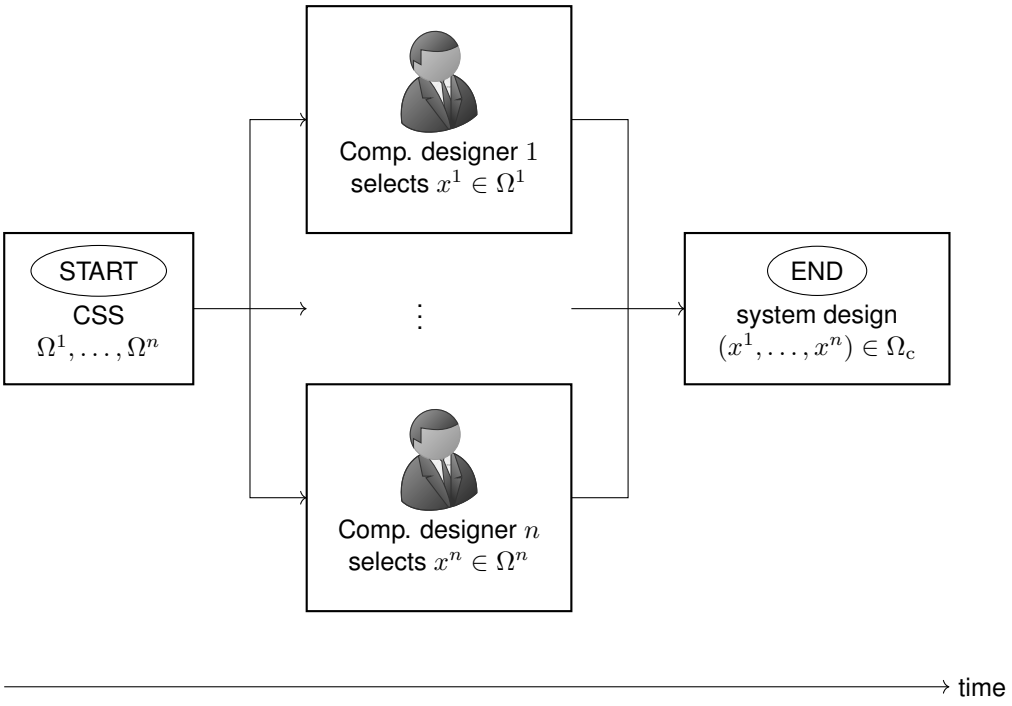


Figure 27 Workflow to obtain a system design based on independent CSS.

for $k = 1, \ldots, n$. This helps, for example, to treat efficiently uncertainties in the component performance functions or the component cost function that may occur later in the design process or if new requirements which affect only single components are introduced. Thus, a new component design can be selected by the corresponding component designer using the new component cost function and including the new constraints into the component solution spaces. The uncertainties that occur in the mentioned cases all belong to the class of model uncertainties, compare Section 2.2. Uncertainties concerning the system performance functions or possible system cost functions are more difficult to treat with CSS.
Due to the flexibility achieved by maximizing the volume of the system solution space Ω dealing with uncertainties in controllable design variables and to some extent also with uncertainties in uncontrollable parameters is possible, like discussed in Section 2.4. The larger the volume of Ω, the more possibilities exist that the realized component designs x^k, $k = 1, \ldots, n$, form a system design $x \in \Omega$ which fulfills all constraints. However, there are more efficient strategies to treat these kinds of uncertainties with CSS. These are discussed in Sections 3.4 and 3.5. Below, independent box-shaped and arbitrarily-shaped CSS are considered.

3.2.2. Box-Shaped CSS

Box-shaped CSS, which are represented by Equation (3.9), are CSS with predefined geometric shapes. In general, CSS with predefined geometric shapes can be described using Equation (3.10) for Ω^k and predefining the component performance functions g^k, possibly up to some parametrization, $k = 1, \ldots, n$. The definition of box-shaped CSS can be seen as a special case of this approach, compare Equation (3.11). In [22], CSS with predefined shape are

discussed using the example of ellipsoid-shaped CSS and compared to box-shaped CSS by the author of this thesis. Subsequently, the focus is only put on box-shaped component solution spaces.

Recall that box-shaped CSS Ω^k are d^k-dimensional-intervals $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$, $k = 1, \ldots, n$. They can be uniquely described by their lower bounds $x^{\mathrm{l},k}$ and their upper bounds $x^{\mathrm{u},k}$. The volume of each $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$ is the product of the d^k-dimensional interval's edge lengths $x_i^{\mathrm{u},k} - x_i^{\mathrm{l},k}$, $i = 1, \ldots, d^k$, i.e.,

$$\mathrm{vol}([x^{\mathrm{l},k}, x^{\mathrm{u},k}]) \quad = \quad \prod_{i=1}^{d^k} x_i^{\mathrm{u},k} - x_i^{\mathrm{l},k}. \tag{3.15}$$

Multiplication of the volumes of Ω^k, $k = 1, \ldots, n$, yields the volume of the Cartesian product $\Omega^1 \times \cdots \times \Omega^n$. Based on problem (3.14), the mathematical optimization problem to compute optimal box-shaped independent CSS then reads

$$\begin{aligned} &\underset{x^{\mathrm{l},1}, x^{\mathrm{u},1}, \ldots, x^{\mathrm{l},n}, x^{\mathrm{u},n}}{\text{maximize}} && \prod_{k=1}^{n} \prod_{i=1}^{d^k} (x_i^{\mathrm{u},k} - x_i^{\mathrm{l},k}) \\ &\text{subject to} && [x^{\mathrm{l},1}, x^{\mathrm{u},1}] \times \cdots \times [x^{\mathrm{l},n}, x^{\mathrm{u},n}] \subseteq \Omega_{\mathrm{c}} \end{aligned} \tag{3.16}$$

where the optimization variables are in real coordinate space. Note again that using box-shaped independent CSS corresponds to a fully independent-decoupled design decision approach, as also the decisions for each component design are decoupled independently. This becomes clear when noticing that the Cartesian product of the d^k-dimensional intervals is a d-dimensional interval. As further the objective function of problem (3.16) is the product of the edge lengths of this d-dimensional interval, problem (3.16) is equivalent to problem (2.36). As stated in Section 2.4, there are different approaches in the literature to solve this problem. In general, each of these approaches is useful for specific types of system performance functions which are used to describe the complete system solution space Ω_{c}. In Section 4.2, a simplification of problem (3.16) is presented for linear performance functions. Below, an example of optimal box-shaped independent CSS is regarded. As the focus of this chapter is put on presenting different problem statements, a derivation of the corresponding solution is not provided for this and the following examples. Nevertheless, all subsequent solutions can be derived using the methods of Chapter 4.

Example 17. Given a system composed of two components. The component design of the first component contains two design variables, i.e., $d^1 = 2$, and the component design of the second component contains one design variable, i.e., $d^2 = 1$, compare Example 16. Furthermore, the system design space is given by $\Omega_{\mathrm{ds}} = [0, 1.5]^3$ and a performance function by $f : \mathbb{R}^3 \to \mathbb{R},\ x \mapsto -3x_1 - 2x_2 - 3x_3$ with threshold $f_{\mathrm{c}} = -9$, known from Example 2.

Optimal box-shaped independent CSS can be computed as $\Omega^1 = [1.1667, 1.5] \times [1, 1.5]$ and $\Omega^2 = [1.1667, 1.5]$, see Example 12. They are visualized in Figure 28.

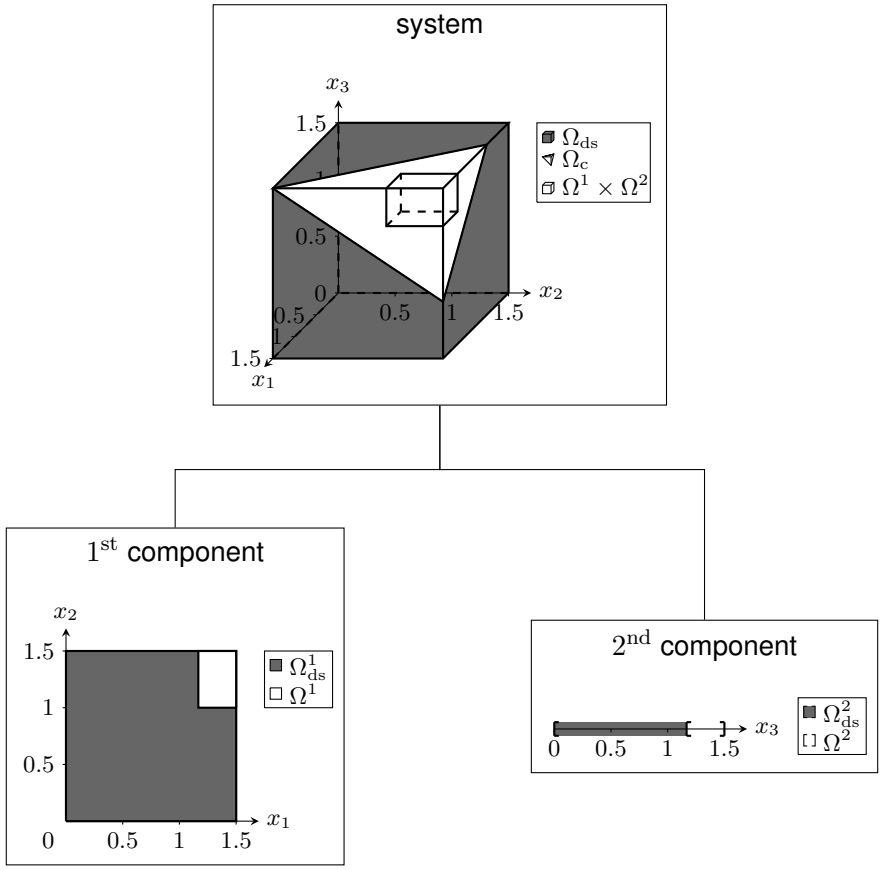


Figure 28 Optimal box-shaped independent CSS of Example 17 at the component level and their Cartesian product within the complete system solution space at the system level.

Using d^k-dimensional intervals to compute optimal independent CSS bears some advantages. One advantage is that the computation of their volume is simple, compare Equation (3.15). This is useful for solving problem (3.16) efficiently. Another advantage is that box-shaped CSS can easily be visualized by showing the one-dimensional intervals that build up the box-shaped CSS. This is even possible for high dimensions. However, a major drawback of box-shaped solution spaces is that their volume can be very small compared to the volume of Ω_{c}, like discussed above. This drawback can be improved by using arbitrarily-shaped CSS.

3.2.3. Arbitrarily-Shaped CSS

Equation (3.10) can also be used to describe arbitrarily-shaped CSS. In contrast to CSS with a predefined geometric shape, the component performance functions g^k, which are used for the definition of Ω^k, are not predefined but optimized together with the component performance thresholds g^k_{c}, $k = 1, \ldots, n$. Here, the optimization variables for g^k are usually not in real coordinate space. This makes the computation of arbitrarily-shaped CSS more complex than the computation of CSS with a predefined geometric shape.

Nevertheless, there are some cases for which optimal component performance functions can be deduced directly. This applies, for example, to cases for which Equation (3.4) holds for the system performance functions f, compare [21]. The following considerations will be limited to this case. Though, more complex cases are conceivable in general.
Equation (3.4) characterizes system performance functions f_j that can be decomposed into a sum of component performance functions g_j^k, $k = 1, \ldots, n$, $j = 1, \ldots, m$. These component performance functions are optimal, and the remaining optimization variables are the entries of the component performance thresholds, cf. Section 4.2. To express the dependency of Ω^k on g_c^k then, the notation $\Omega^k(g_\mathrm{c}^k)$ is used for $k = 1, \ldots, n$. The sum of the optimization variables must be smaller than or equal to the system performance threshold f_c to ensure that condition (3.13) is fulfilled, i.e.,

$$\sum_{k=1}^{n} g_\mathrm{c}^k \quad \leq \quad f_\mathrm{c}. \tag{3.17}$$

Hence, if the system performance functions f are a sum of component performance functions g^k, $k = 1, \ldots, n$, problem (3.14) is equivalent to

$$\begin{aligned} &\underset{g_\mathrm{c}^1, \cdots, g_\mathrm{c}^n}{\text{maximize}} && \mathrm{vol}(\Omega^1(g_\mathrm{c}^1) \times \cdots \times \Omega^n(g_\mathrm{c}^n)) \\ &\text{subject to} && \sum_{k=1}^{n} g_\mathrm{c}^k \leq f_\mathrm{c}. \end{aligned} \tag{3.18}$$

In problem (3.18), the optimization variables are in real coordinate space. Strategies to solve the problem numerically, which extend the ones proposed in [21] by the author of this thesis, are presented in Section 4.3. In general, the major difficulty for solving problem (3.18) is the computation of the volume of $\Omega^1(g_\mathrm{c}^1) \times \cdots \times \Omega^n(g_\mathrm{c}^n)$. Further properties of problem (3.18) are investigated for linear performance functions in Section 4.2. In the following, an example of optimal arbitrarily-shaped independent CSS is provided.

Example 18. Given the situation from Example 17. There is a system that is composed of two components. The component design of the first component contains two design variables, and the component design of the second component contains one design variable. The system design space is given by $\Omega_\mathrm{ds} = [0, 1.5]^3$ and the performance function by $f : \mathbb{R}^3 \to \mathbb{R},\ x \mapsto -3x_1 - 2x_2 - 3x_3$ with threshold $f_\mathrm{c} = -9$.
Therefore, optimal component performance functions are $g^1 : \mathbb{R}^2 \to \mathbb{R},\ (x_1, x_2) \mapsto -3x_1 - 2x_2$ and $g^2 : \mathbb{R} \to \mathbb{R},\ x_3 \mapsto -3x_3$, and the optimal component performance thresholds can be computed as $g_\mathrm{c}^1 = -5.5$ and $g_\mathrm{c}^2 = -3.5$. Then, optimal arbitrarily-shaped independent CSS are $\Omega^1 = \{(x_1, x_2) \in [0, 1.5]^2 \mid -3x_1 - 2x_2 \leq -5.5\}$ and $\Omega^2 = \{x_3 \in [0, 1.5] \mid -3x_3 \leq -3.5\}$. For this case, the result matches the results from Example 14. The corresponding CSS are visualized in Figure 29.

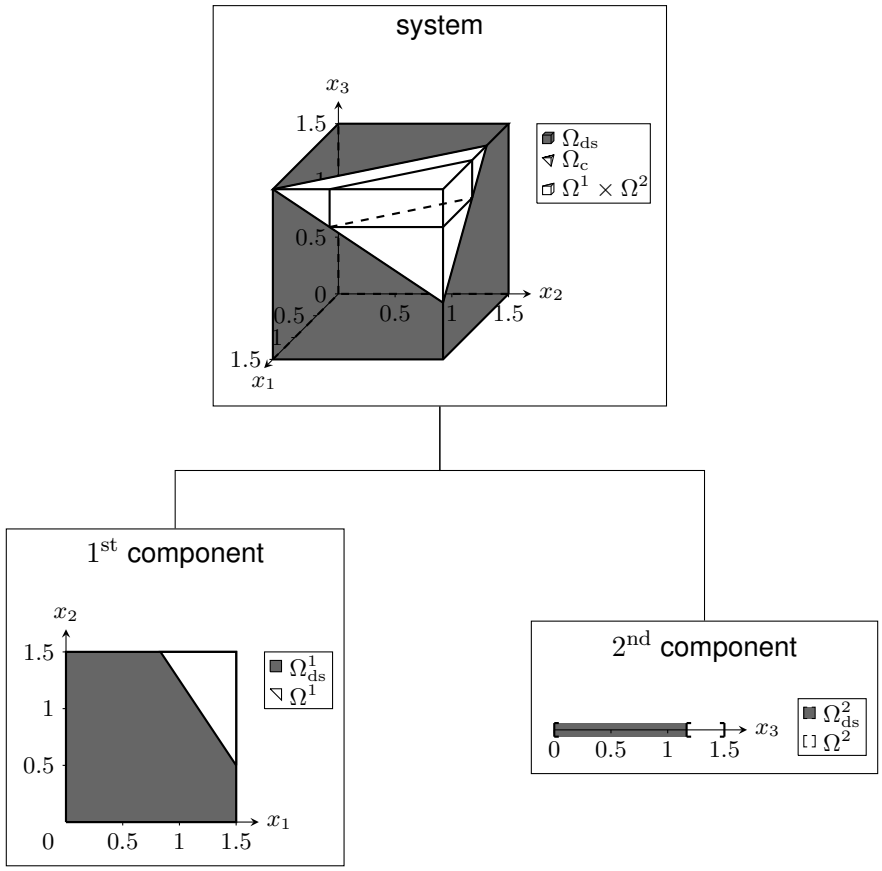


Figure 29 Optimal arbitrarily-shaped independent CSS of Example 18 at the component level and their Cartesian product within the complete system solution space at the system level.

The visualization of non-box-shaped CSS Ω^k is problematic for $d^k > 3$. One way to overcome this problem, which is related to parallel coordinate plots, is proposed in Section5.3. Nevertheless, more flexibility for component design is obtained by using arbitrarily-shaped CSS compared to box-shaped solution spaces. Also in Section 5.3, arbitrarily-shaped CSS are compared to box-shaped CSS in this regard and in terms of computation time. Before doing so, similar investigations are done for dependent CSS in the following.

3.3. Dependent CSS

This section considers mathematical problem statements for computing dependent component solution spaces that provide optimal flexibility for dependent-decoupled design decisions between the components. Here, it is distinguished between box- and arbitrarily-shaped dependent CSS again.

3.3.1. Problem Statement

In order to provide flexibility for dependent-decoupled design decisions between the components, component solution spaces Ω^k, $k = 1, \ldots, n$, are required for which any but only one of the n component designers, denoted by k^1 with $k^1 \in \{1, \ldots, n\}$, can select a permissible component design first. Based on this decision, the component solution spaces for the remaining component designs are updated and another component designer, denoted by k^2 with $k^2 \in \{1, \ldots, n\} \backslash \{k^1\}$, can select a permissible component design next. This is continued until the last decision for a permissible component design was made and hence, a system design is yielded. As the CSS for any component always depends on previous decisions, the CSS provide flexibility for dependent-decoupled design decisions between the components and are called *dependent component solution spaces*.

After the first component design $x^{k^1} \in \Omega^{k^1}$ was selected, it must be guaranteed that there are component designs $x^{k^2}, \ldots, x^{k^n}$ of the remaining components which can result in a permissible system design. Thus, Ω^{k^1} must be a subset of the projection of Ω_c onto the coordinate space of the $k^{1\,\mathrm{th}}$ component, i.e. $\Omega^{k^1} \subseteq \mathrm{proj}^{k^1}(\Omega_\mathrm{c})$. As k^1 can take any value in $\{1, \ldots, n\}$, the condition

$$\Omega^k \subseteq \mathrm{proj}^k(\Omega_\mathrm{c}) \tag{3.19}$$

is a necessary and sufficient condition for the CSS Ω^k, $k = 1, \ldots, n$, of the first decision. Note that the Cartesian product $\Omega^1 \times \cdots \times \Omega^n$ is usually no system solution space here. In Figure 30, the relationship of the CSS is shown from a system perspective.

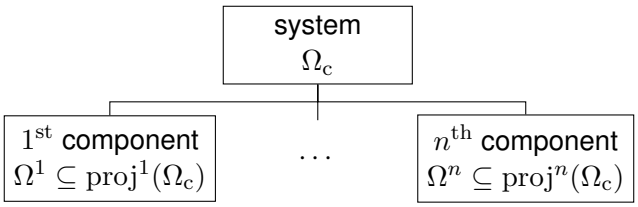


Figure 30 Dependent component solution spaces at the component level and their complete system solution space at the system level for the first decision.

In general, there are multiple sets for dependent CSS for the first decision that fulfill condition (3.19). Among these, optimal component solution spaces which offer the most flexibility for dependent-decoupled design decisions are preferred. In contrast to above, this flexibility for component design is not quantified by the Cartesian product $\Omega^1 \times \cdots \times \Omega^n$ here as this Cartesian product is usually no system solution space. Thus, the volumes of the single Ω^k, $k = 1, \ldots, n$, are used as a flexibility measure to optimize dependent CSS. However, other flexibility measures to optimize dependent CSS are conceivable, too. Optimizing dependent CSS for each component by ensuring condition (3.19) can be stated as mathematical optimization

problems. It reads

$$\begin{aligned} &\underset{\Omega^k}{\text{maximize}} && \text{vol}(\Omega^k) \\ &\text{subject to} && \Omega^k \subseteq \text{proj}^k(\Omega_\text{c}) \end{aligned} \tag{3.20}$$

for $k = 1, \ldots, n$. Similar to problem (3.14), $\Omega^k \subset \mathbb{R}^{d^k}$ must be Borel sets to enable the calculation of the volume of Ω^k, $k = 1, \ldots, n$. If no restrictions on the geometric shapes of the CSS are put, the unique solutions to problem (3.20) are given by $\Omega^k = \text{proj}^k(\Omega_\text{c})$ as they form the largest set within $\text{proj}^k(\Omega_\text{c})$ in terms of volume, $k = 1, \ldots, n$. Here, the CSS take arbitrary geometric shapes, depending on the projection of Ω_c, and are therefore called arbitrarily-shaped CSS. Opposite to that, the geometric shapes of the CSS can be predefined again. In this case, additional optimization constraints for the shapes must be introduced in problem (3.20).
For arbitrarily-shaped CSS, the main challenge is to calculate the projection of Ω_c onto the coordinate space of the k^th component. For CSS with predefined shape, this challenge usually occurs together with the problem of maximizing the volume of Ω^k, $k = 1, \ldots, n$. In the following, both cases are discussed.

As soon as dependent CSS are computed, the first component designer k^1 can choose a component design within the corresponding CSS, like proposed in Section 2.3 using a component cost function. Afterward, Ω_c is updated by fixing x^{k^1} to its selected values, which results in a $(d - d^{k^1})$-dimensional set, and is used to compute the CSS for the remaining components. The problem statement to compute the updated CSS is similar to problem (3.20). Thus, only the CSS that provide flexibility for the first decision for a component design are considered and visualized in this thesis. The associated workflow is shown in Figure 31.

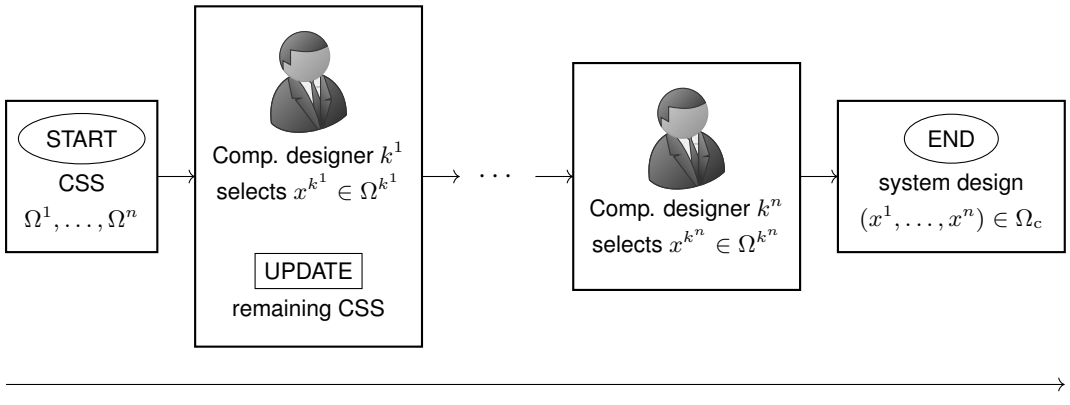


Figure 31 Workflow to obtain a system design based on dependent CSS.

Changing a single component design after the system design was composed of the component designs is also possible when using dependent CSS. However, it is more expensive than in the case of independent CSS. If the design of the k^th component shall be changed using

dependent CSS, the complete system solution space must be updated by fixing the values of the selected designs of the other components. The resulting d^k-dimensional set is a subset of the coordinate space of the k^{th} component and therefore corresponds to the projection onto this space. It can be used in problem (3.20) to compute a new CSS Ω^k for the k^{th} component, $k \in \{1, \ldots, n\}$. Then, a new decision for $x^k \in \Omega^k$ can be made. Again, this helps to treat uncertainties in the component performance functions or the component cost function that belong to the class of model uncertainty.
Due to the flexibility obtained from maximizing the volume of component solution spaces it is also possible to deal with uncertainties in design variables and to some extent with uncertainties in uncontrollable parameters, as discussed for independent CSS in Sections 2.4. However, there are also more efficient strategies to treat these uncertainties, which are discussed in Sections 3.4 and 3.5. Subsequently, box-shaped and arbitrarily-shaped dependent CSS are considered.

3.3.2. Box-Shaped CSS

Recall that box-shaped CSS represented by Equation (3.9) are a special type of CSS with predefined geometric shapes. They are d^k-dimensional-intervals $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$ and can be uniquely described by their lower bounds $x^{\mathrm{l},k}$ and their upper bounds $x^{\mathrm{u},k}$ for $k = 1, \ldots, n$. Their volume is the product of their edge lengths, compare Equation (3.15). Based on problem (3.20), the mathematical optimization problem to compute optimal box-shaped dependent CSS reads

$$\begin{aligned} \underset{x^{\mathrm{l},k}, x^{\mathrm{u},k}}{\text{maximize}} \quad & \prod_{i=1}^{d^k} (x_i^{\mathrm{u},k} - x_i^{\mathrm{l},k}) \\ \text{subject to} \quad & [x^{\mathrm{l},k}, x^{\mathrm{u},k}] \subseteq \mathrm{proj}^k(\Omega_{\mathrm{c}}), \end{aligned} \tag{3.21}$$

$k = 1, \ldots, n$, where the optimization variables are in real coordinate space. If only one $k \in \{1, \ldots, n\}$ is considered, problem (3.21) is similar to problem (2.40). Approaches that solve this problem are mentioned in Section 2.3. Moreover, a new approach to simplify problem (3.16) for linear performance functions is presented in Section 4.2. Note that using box-shaped dependent CSS corresponds to dependent-decoupled design decisions between components and independent-decoupled design decisions for each component. Below, an example for optimal box-shaped dependent CSS is considered.

Example 19. Given the situation from Example 17. There is a system that is composed of two components. The component design of the first component contains two design variables, and the component design of the second component contains one design variable. The system design space is given by $\Omega_{\mathrm{ds}} = [0, 1.5]^3$ and the performance function by $f : \mathbb{R}^3 \to \mathbb{R},\ x \mapsto -3x_1 - 2x_2 - 3x_3$ with threshold $f_{\mathrm{c}} = -9$.
Optimal box-shaped dependent CSS can be computed as $\Omega^1 = [1, 1.5] \times [0.75, 1.5]$, compare

Example 15, and $\Omega^2 = [0.5, 1.5]$. The CSS are visualized in Figure 32.

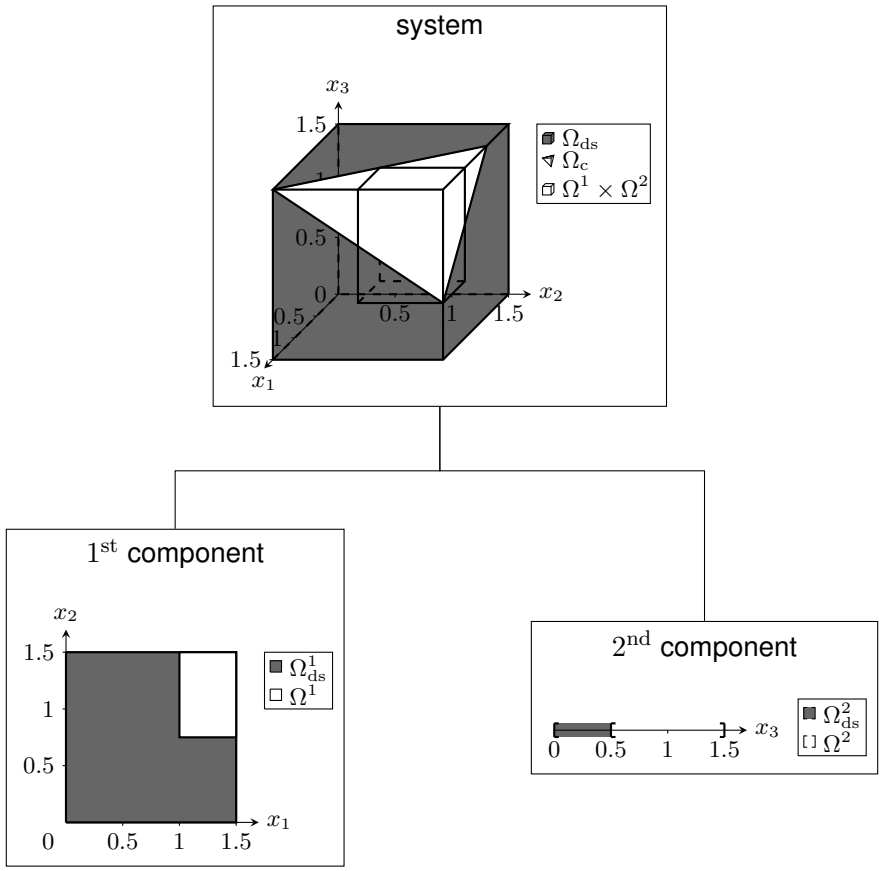


Figure 32 Optimal box-shaped dependent CSS of Example 19 at the component level and their Cartesian product within the system design space at the system level.

The properties of box-shaped dependent CSS are similar to box-shaped independent CSS. The computation of their volume is simple compared to other CSS with predefined geometric shapes, which is useful for solving problem (3.21) efficiently. Furthermore, simple visualizations as one-dimensional intervals that build up the box-shaped CSS are possible. Though, the volume of Ω^k can be small compared to the volume of $\mathrm{proj}^k(\Omega_{\mathrm{c}})$ for $k = 1, \ldots, n$. By using arbitrarily-shaped CSS, this problem can be improved.

3.3.3. Arbitrarily-Shaped CSS

When considering arbitrarily-shaped CSS Ω^k, $k = 1, \ldots, n$, no constraints on their geometric shapes are put. Hence, the optimal solutions of problem (3.21) are

$$\Omega^k \quad = \quad \mathrm{proj}^k(\Omega_{\mathrm{c}}), \tag{3.22}$$

$k = 1, \ldots, n$. Any other $\Omega^k \subseteq \mathrm{proj}^k(\Omega_{\mathrm{c}})$ has a smaller volume than $\mathrm{proj}^k(\Omega_{\mathrm{c}})$, which means that there is less flexibility for component design in a non-optimal case. Problem (3.21) is then reduced to compute the projection of the d-dimensional set Ω_{c} onto the d^k-dimensional coordinate space of the k^{th} component for $k = 1, \ldots, n$. The projected set can be represented by Equation (3.10) again, in which g^k are the component performance functions and g^k_{c} are

the component performance thresholds.
If the system performance functions are linear, there are multiple options to compute this projection, compare [71]. From these options, an enhanced Fourier-Motzkin method, which is based on an elimination of variables from the inequality system that describes Ω_{c}, is presented in Section 4.3. This can be extended to nonlinear cases, see, e.g., [4, 50]. In the following, an example for optimal arbitrarily-shaped dependent CSS is presented.

Example 20. Given the situation from Example 17. There is a system that is composed of two components. The component design of the first component contains two design variables, and the component design of the second component contains one design variable. The system design space is given by $\Omega_{\mathrm{ds}} = [0, 1.5]^3$ and the performance function by $f : \mathbb{R}^3 \to \mathbb{R},\ x \mapsto -3x_1 - 2x_2 - 3x_3$ with threshold $f_{\mathrm{c}} = -9$.
Optimal arbitrarily-shaped dependent CSS can be computed as $\Omega^1 = \mathrm{proj}^1(\Omega_{\mathrm{c}}) = \{(x_1, x_2) \in [0, 1.5]^2 \mid -3x_1 - 2x_2 \leq -4.5\}$ and $\Omega^2 = \mathrm{proj}^2(\Omega_{\mathrm{c}}) = \{x_3 \in [0, 1.5] \mid -x_3 \leq -0.5\}$. Here, $-x_3 \leq 0$ is a redundant constraint and therefore, it holds $\Omega^2 = [0, 1.5]$. The optimal CSS are visualized in Figure 33.

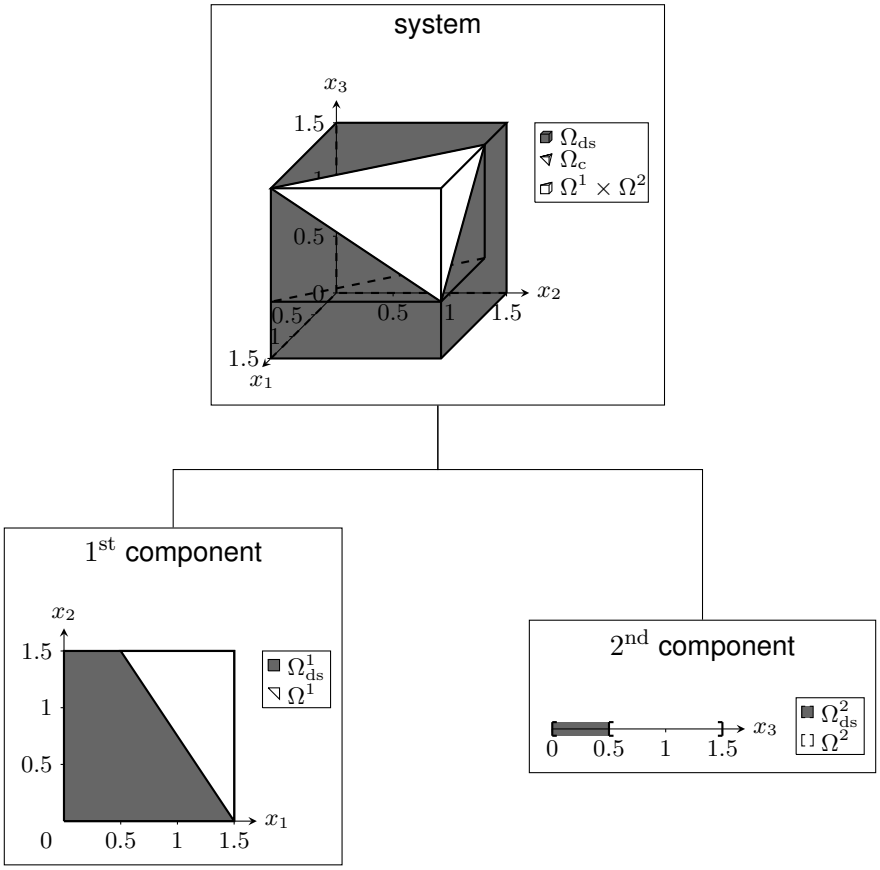


Figure 33 Optimal arbitrarily-shaped dependent CSS of Example 20 at the component level and their Cartesian product within the system design space at the system level.

Here, the properties of arbitrarily-shaped dependent CSS are similar to arbitrarily-shaped independent CSS. The visualization of non-box-shaped CSS Ω^k is problematic for $d^k > 3$.

Nevertheless, more flexibility for component design compared to box-shaped solution spaces is obtained. In Section 5.3, arbitrarily-shaped dependent CSS are compared to box-shaped dependent CSS regarding their volume and computation time together with independent CSS. In the subsequent sections, optimal CSS are computed for uncertainties in controllable variables and uncontrollable parameters.

3.4. CSS under Interval-Type Uncertainty

This section presents how epistemic uncertainty, modeled as interval-type uncertainty, can be considered in the problem statements for independent and dependent CSS. First, uncertainties in controllable variables for which it is distinguished between knowledge-based and maximized magnitudes are considered. Then, uncertainties in uncontrollable parameters are taken into account as well. This yields a complete methodology for computing CSS under interval-type uncertainty.

3.4.1. Uncertainties in Controllable Variables with Knowledge-Based Magnitudes

If uncertainties in controllable variables are present and component target designs are selected from their corresponding CSS, computed like above, the resulting realized system design might be non-permissible due to these uncertainties. As stated above, an a-priori uncertainty consideration can help to avoid this problem by computing appropriate component solution spaces.

To begin with, uncertainties in uncontrollable parameters are neglected. Recall that for interval-type uncertainties in controllable variables, designers can only select the values of the target design variables $\check{x}_i$, $i = 1, \ldots, d$, for which the realized values of the design variables can be found within the intervals $[\check{x}_i - \delta_i, \check{x}_i + \delta_i]$ for given magnitudes $\delta_i \in \mathbb{R}_0^+$, $i = 1, \ldots, d$. This means that component designers can select the values of the target design variables $\check{x}^k$, $k = 1, \ldots, n$, for which the realized component designs can be found within the d^k-dimensional intervals $[\check{x}^k - \delta^k, \check{x}^k + \delta^k]$. Here, δ^k have entries $\delta_i^k \in \mathbb{R}_0^+$, $i = 1, \ldots, d^k$, $k = 1, \ldots, n$, which are obtained from δ_i, $i = 1, \ldots, d$, similarly to Equation (3.2). As stated in Section 2.2 the assumption of fixed values for δ_i, $i = 1, \ldots, d$, is often crucial. Thus, the values of δ_i, $i = 1, \ldots, d$, are only assumed in this thesis. They are denoted by $\bar{\delta}_i \in \mathbb{R}_0^+$, $i = 1, \ldots, d$. These magnitudes are collected in the d-dimensional vector $\bar{\delta}$ at the system level and in the d^k-dimensional vectors $\bar{\delta}^k$, $k = 1, \ldots, n$, at the component level. In the following, it is distinguished between knowledge-based magnitudes and maximized magnitudes. The corresponding problem statements for box-shaped CSS in a worst-case scenario were already published in [25] by the author of this thesis.

First, knowledge-based uncertainty magnitudes are considered. Here, the values of $\bar{\delta}_i$ are

estimated based on the available knowledge about δ_i, $i = 1, \ldots, d$. For example, this can be the case if the uncertainties can be experimentally evaluated or if there is expert knowledge, e.g., on manufacturing tolerances about variations in material or geometry properties, which shall be already considered in the early design phase. If the design variables are responses of lower-level systems, $\bar{\delta}_i$, $i = 1, \ldots, d$, can be estimated from knowledge about the uncertainties of the lower-level systems. Note that all these cases may represent lack-of-knowledge situations in the early design phase for which epistemic uncertainty is present and precise uncertainty quantification of the uncertainties in the controllable variables is not available. An extreme case of knowledge-based $\bar{\delta}_i$ is $\bar{\delta}_i = 0$ for which no uncertainties are assumed, $i = 1, \ldots, d$. In order to avoid a non-permissible system design in a worst-case scenario, $\bar{\delta}_i$ must be chosen as upper bounds of δ_i, i.e., $\delta_i \leq \bar{\delta}_i$, $i = 1, \ldots, d$.

For knowledge-based $\bar{\delta}_i$, $i = 1, \ldots, d$, the set of all system target designs can be determined. Their corresponding uncertainty sets are subsets of the complete system solution space Ω_c. This set is the worst-case complete system solution space of target designs, see Equation (2.12), and it holds

$$\check{\Omega}_{\mathrm{c,wc}} = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X(\check{x}, \bar{\delta}) : \; x \in \Omega_\mathrm{c}\}. \tag{3.23}$$

Then, the design decisions to select a target design within $\check{\Omega}_{\mathrm{c,wc}}$ can be decoupled between the components using the approaches from Sections 3.2 and 3.3. This yields *worst-case CSS of target designs* $\check{\Omega}^k_\mathrm{wc}$, $k = 1, \ldots, n$. If all component designers select permissible component target designs, i.e., $\check{x}^k \in \check{\Omega}^k$, any realized system design is permissible if $\delta_i \leq \bar{\delta}_i$, $i = 1, \ldots, d$, holds. In doing so, uncertainties in controllable variables are treated efficiently. In order to provide optimal flexibility for component design under interval-uncertainty, the volume of the CSS of target designs is maximized. If $\check{\Omega}_{\mathrm{c,wc}}$ is a non-empty set with a positive volume, the optimization problem to obtain *worst-case independent CSS of target designs* reads

$$\begin{aligned} \underset{\check{\Omega}^1_\mathrm{wc},\ldots,\check{\Omega}^n_\mathrm{wc}}{\text{maximize}} \quad & \mathrm{vol}(\check{\Omega}^1_\mathrm{wc} \times \cdots \times \check{\Omega}^n_\mathrm{wc}) \\ \text{subject to} \quad & \check{\Omega}^1_\mathrm{wc} \times \cdots \times \check{\Omega}^n_\mathrm{wc} \subseteq \check{\Omega}_{\mathrm{c,wc}} \end{aligned} \tag{3.24}$$

and the optimization problems to obtain *worst-case dependent CSS of target designs* read

$$\begin{aligned} \underset{\check{\Omega}^k_\mathrm{wc}}{\text{maximize}} \quad & \mathrm{vol}(\check{\Omega}^k_\mathrm{wc}) \\ \text{subject to} \quad & \check{\Omega}^k_\mathrm{wc} \subseteq \mathrm{proj}^k(\check{\Omega}_{\mathrm{c,wc}}), \end{aligned} \tag{3.25}$$

$k = 1, \ldots, n$. The corresponding problems to compute CSS under absence of uncertainty are stated in problems (3.14) and (3.20). In Section 4.2, it is presented how solutions of problems (3.24) and (3.25) can be solved according to problems (3.14) and (3.20) for linear performance functions.

When worst-case CSS of target designs $\check{\Omega}^k_{\mathrm{wc}}$ are available, CSS Ω^k and *best-case CSS of target designs* $\check{\Omega}^k_{\mathrm{bc}}$ can be deduced, $k = 1, \ldots, n$. Here, the corresponding properties of the worst- and best-case complete system solution spaces must be ensured, i.e.,

$$\check{\Omega}^k_{\mathrm{wc}} = \{\check{x}^k \in \mathbb{R}^{d^k} \mid \forall x^k \in \mathcal{U}^X(\check{x}^k, \bar{\delta}^k) : \ x^k \in \Omega^k\}, \tag{3.26}$$

for worst-case CSS of target designs and

$$\check{\Omega}^k_{\mathrm{bc}} = \{\check{x}^k \in \mathbb{R}^{d^k} \mid \exists x^k \in \mathcal{U}^X(\check{x}^k, \bar{\delta}^k) : \ x^k \in \Omega^k\}, \tag{3.27}$$

for best-case CSS of target designs, $k = 1, \ldots, n$. The sets $\mathcal{U}^X(\check{x}^k, \bar{\delta}^k)$ are the uncertainty sets of the component designs with respect to $\check{x}^k \in \mathbb{R}^{d^k}$, $k = 1, \ldots, n$, similar to Equation (2.8).
Thus, an inverse scheme must be applied to obtain CSS Ω^k from $\check{\Omega}^k_{\mathrm{wc}}$. For both independent and dependent CSS,

$$\Omega^k = \{x^k \in \Omega^k_{\mathrm{ds}} \mid \exists \check{x}^k \in [x^k - \bar{\delta}^k, x^k + \bar{\delta}^k] : \ \check{x}^k \in \check{\Omega}^k_{\mathrm{wc}}\}, \tag{3.28}$$

$k = 1, \ldots, n$, can be used. However, for arbitrarily-shaped independent CSS Ω^k, $k = 1, \ldots, n$, obtained from Equation (3.28), there might be CSS Ω'^k, $k = 1, \ldots, n$, with $\Omega'^1 \times \cdots \times \Omega'^n \subseteq \Omega_{\mathrm{c}}$ and $\mathrm{vol}(\Omega'^1 \times \cdots \times \Omega'^n) > \mathrm{vol}(\Omega^1 \times \cdots \times \Omega^n)$. In this case, the flexibility of the CSS can be further optimized by solving problem (3.14) and using the CSS obtained from Equation (3.28) as sets which must be subsets of the optimal CSS. Similarly, the flexibility of arbitrarily-shaped dependent CSS can be optimized for each Ω^k by solving problem (3.25) and using a modified $\check{\Omega}_{\mathrm{wc}}$ with $\bar{\delta}^k = 0$ and $k \in \{1, \ldots, n\}$. Note that dependent CSS which are computed from the worst-case dependent CSS of target designs are aligned with the worst-case, i.e., for all $x^{k^1} \in \Omega^{k^1}$ there is at least one $(\check{x}^{k^2}, \ldots, \check{x}^{k^n})$ within the updated worst-case complete system solution space, see Example 21.
Then, best-case CSS of target designs $\check{\Omega}^k_{\mathrm{bc}}$ can be computed from the CSS Ω^k using Equation (3.27), $k = 1, \ldots, n$. Similar to Ω^k, the best-case CSS of target designs are aligned with the worst-case, i.e., for all $\check{x}^{k^1} \in \check{\Omega}^{k^1}_{\mathrm{bc}}$ there is at least one $x^{k^1} \in \Omega^{k^1}$ for that at least one $(\check{x}^{k^2}, \ldots, \check{x}^{k^n})$ exists within the updated worst-case complete system solution space, see Example 21. Note however that in the case of $\delta_i < \bar{\delta}_i$, there are target designs within $\check{\Omega}^1_{\mathrm{bc}} \times \cdots \times \check{\Omega}^n_{\mathrm{bc}}$ for which no $x \in \Omega_{\mathrm{c}}$ exists. Thus, the convention to use upper bounds $\bar{\delta}_i$ for δ_i can be regarded critically for best-case scenarios for which lower bounds of δ_i should be taken into account instead, $i = 1, \ldots, d$.

In general, the optimization of flexibility can also be done with respect to the best-case complete system solution space of target designs. Here, best-case CSS of target designs can be obtained by replacing $\check{\Omega}_{\mathrm{c,wc}}$ with $\check{\Omega}_{\mathrm{c,bc}}$ in problems (3.24) and (3.25). Note that this procedure usually yields different results. In order to obtain CSS Ω^k from $\check{\Omega}^k_{\mathrm{bc}}$, an inverse

scheme must be applied again. For both independent and dependent CSS,

$$\Omega^k = \{x^k \in \Omega^k_{\mathrm{ds}} \mid \forall \check{x}^k \in [x^k - \bar{\delta}^k, x^k + \bar{\delta}^k] : \ \check{x}^k \in \check{\Omega}^k_{\mathrm{bc}}\}, \tag{3.29}$$

$k = 1, \ldots, n$, can be used. However, if worst-case CSS of target designs are computed from best-case CSS of target designs, it is not necessarily guaranteed that the worst-case CSS of target designs are non-empty sets. This may even hold if the worst-case complete system solution space is a non-empty set. The same circumstances can occur for the CSS Ω^k, $k = 1, \ldots, n$. Note that dependent CSS which are computed from best-case dependent CSS of target designs are aligned with the best-case, i.e., for all $x^{k^1} \in \Omega^{k^1}$ there is at least one $(\check{x}^{k^2}, \ldots, \check{x}^{k^n})$ within the updated best-case complete system solution space. Similar results hold if worst-case CSS of target designs are computed from best-case CSS of target designs. In the following, an example is given in which both worst-case independent and dependent CSS of target designs $\check{\Omega}^k_{\mathrm{wc}}$ are optimized and corresponding CSS Ω^k and best-case CSS of target designs $\check{\Omega}^k_{\mathrm{bc}}$ are deduced using Equations (3.27) and (3.28), $k = 1, \ldots, n$.

Example 21. Given a system composed of two components. The component designs of both the first and the second component consist of one design variable each, i.e., $d^1 = d^2 = 1$, which means that box- and arbitrarily-shaped CSS coincide. Furthermore, the system design space is given by $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$ and a performance function by $f : \mathbb{R}^2 \to \mathbb{R}, \ x \mapsto x_1 + 2x_2$ with threshold $f_{\mathrm{c}} = 2$. There are interval-type uncertainties in the controllable variables with $\bar{\delta}_1 = 0.3$, $\bar{\delta}_2 = 0.1$, compare Example 3.
Optimal worst-case independent CSS of target designs can be computed as $\check{\Omega}^1_{\mathrm{wc}} = [0.3, 0.8]$ and $\check{\Omega}^2_{\mathrm{wc}} = [0.1, 0.35]$ and optimal worst-case dependent CSS of target designs as $\check{\Omega}^1_{\mathrm{wc}} = [0.3, 1.9]$ and $\check{\Omega}^2_{\mathrm{wc}} = [0.1, 0.6]$. They are visualized together with their corresponding CSS Ω^k and best-case CSS of target designs $\check{\Omega}^k_{\mathrm{bc}}$ in Figure 34, $k = 1, 2$.

The worst-case CSS from Example 21 provide optimal flexibility for component design under interval-type uncertainties in controllable variables. As the corresponding complete system solution spaces are identical to the ones of Examples 12 and 13, and all approaches fully decouple the design decisions for the single design variables, the Cartesian product of the CSS of Example 21 can be compared with the CSS of Examples 12 and 13. For the independent CSS of Example 21, the interval for x_1 is larger and the interval for x_2 is smaller compared to those of Example 12. As $\bar{\delta}_1 > \bar{\delta}_2$ holds, this property provides increased flexibility for decoupled decisions regarding the component target designs. For different $\bar{\delta}_1$ and $\bar{\delta}_2$, the size of these intervals would change. Regarding the dependent CSS of Example 21, the intervals for x_1 and x_2 are smaller in comparison with those of Example 13. The system designs of the complete system solution space which are excluded here cannot be realized by system target designs within the worst-case complete system solution space of target designs. As discussed above, this is reasonable, because only values of target design variables within $\check{\Omega}^{k^1}_{\mathrm{wc}}$, $k^1 \in \{1, 2\}$, can be selected then. For these, it can be ensured that a target design

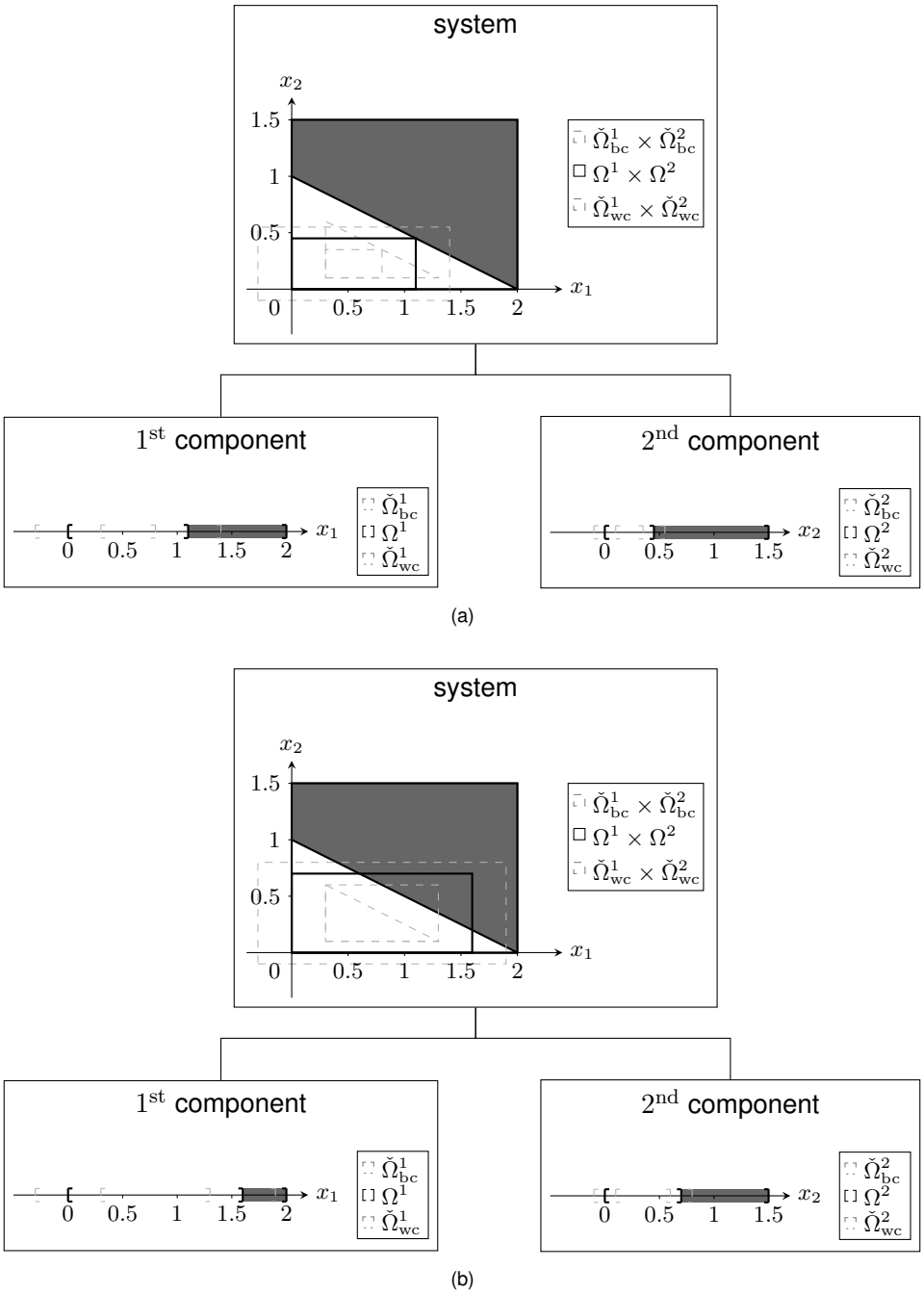


Figure 34 Optimal worst-case (a) independent and (b) dependent CSS of target designs and corresponding CSS and best-case CSS of target designs of Example 21 at the component level and their Cartesian product at the system level. Note that in the legend, the solution spaces are listed according to their size where the largest is on top.

variable exists in $\check{\Omega}^{k^2}_{\mathrm{wc}}$, $k^2 \in \{1,2\}\backslash\{k^1\}$, such that the realized system design is permissible. In Section 5.4, the comparison of CSS obtained from worst-case CSS of target designs with CSS that do not incorporate uncertainty is further investigated.

3.4.2. Uncertainties in Controllable Variables with Maximized Magnitudes

If there is no or very limited knowledge about the magnitude of δ_i or if the complete system solution space of target designs is empty, maximum magnitudes $\overline{\delta}_i$ can be computed for any

system target design $\check{x} \in \Omega_{\mathrm{c}}$. This is done in a way such that all system designs x with $x_i \in \mathcal{U}^X(\check{x}, \bar{\delta})$, i.e., $x_i \in [\check{x}_i - \bar{\delta}_i, \check{x}_i + \bar{\delta}_i]$, $i = 1, \ldots, d$, are permissible. In general, the single magnitudes $\bar{\delta}_i$, $i = 1, \ldots, d$, must be coupled in order to maximize them. One way of doing so is to assess the relation between the magnitudes of $\bar{\delta}_i$, $i = 1, \ldots, d$, which reduces the optimization quantities to one variable, denoted by $\delta' \in \mathbb{R}_0^+$. For example, this can be done by expert knowledge. In the case of linear relations, it holds

$$\bar{\delta}_i = \omega_i \delta' + \nu_i \tag{3.30}$$

with $\omega_i \in \mathbb{R}_0^+$, $\nu_i \in \mathbb{R}_0^+$, $i = 1, \ldots, d$. Then, the variable $\delta' \in \mathbb{R}_0^+$ can be maximized such that all system designs within the uncertainty set $\mathcal{U}^X(\check{x}, \bar{\delta})$ are permissible. The parameter ν_i can help to include partially known uncertainty, e.g., if multiple uncertainties are addressed in $\bar{\delta}_i$ and some of them are already known. The special case $\omega_i = 0$ and $\nu_i \geq 0$ corresponds to the situation of knowledge-based $\bar{\delta}$, $i = 1, \ldots, d$. Typical cases for $\omega_i > 0$, $i = 1, \ldots, d$ include:

(a) $\omega_i = 1$, $\nu_i \geq 0$: Here, all $\bar{\delta}_i$ are weighted the same. This can be the case if no information on the relation between the magnitudes of $\bar{\delta}_i$ is available. For $\nu_i = 0$, it then holds $\bar{\delta}_i = \delta'$.

(b) $\omega_i = \frac{x^{\mathrm{u}}_{\mathrm{ds},i} - x^{\mathrm{l}}_{\mathrm{ds},i}}{2}$, $\nu_i = 0$: Here, the length of the resulting intervals $[\check{x}_i - \frac{x^{\mathrm{u}}_{\mathrm{ds},i} - x^{\mathrm{l}}_{\mathrm{ds},i}}{2}\delta', \check{x}_i + \frac{x^{\mathrm{u}}_{\mathrm{ds},i} - x^{\mathrm{l}}_{\mathrm{ds},i}}{2}\delta']$ is proportional to the length of the intervals $[x^{\mathrm{l}}_{\mathrm{ds},i}, x^{\mathrm{u}}_{\mathrm{ds},i}]$ of the design space. For $\delta' = 1$, the length of these two types of intervals are the same.

(c) $\omega_i = \check{x}_i$, $\nu_i = 0$: With this property, the resulting intervals are $[(1 - \delta')\check{x}_i, (1 + \delta')\check{x}_i]$ which account for relative magnitudes of the target design variable $\check{x}_i$. For $\delta' = 1$, these intervals become $[0, 2\check{x}_i]$.

In order to maximize $\bar{\delta}_i$, $i = 1, \ldots, d$, for given ω_i and ν_i, a target design $\check{x}$ with maximum δ' is sought. The corresponding optimization problem reads

$$\begin{aligned} &\underset{\check{x}, \delta'}{\text{maximize}} && \delta' \\ &\text{subject to} && \mathcal{U}^X(\check{x}, \delta'\omega + \nu) \subseteq \Omega_{\mathrm{c}}, \end{aligned} \tag{3.31}$$

$\delta' \geq 0$, where $\omega = (\omega_1, \ldots, \omega_d)$, $\nu = (\nu_1, \ldots, \nu_d)$, and

$$\mathcal{U}^X(\check{x}, \delta'\omega + \nu) = \{x \in \mathbb{R}^d \mid x_i \in [\check{x}_i - (\omega_i\delta' + \nu_i), \check{x}_i + (\omega_i\delta' + \nu_i)],\ i = 1, \ldots d\}, \tag{3.32}$$

compare Equation (2.8). For $\nu_i = 0$, $i = 1, \ldots, d$, problem (3.31) is similar to the problem statement considered in [44]. If further $\omega_i = 1$ holds for $i = 1, \ldots, d$, problem (3.31) corresponds to problem (2.32). The system target design of the solution of problem (3.31) is a robust system target design which allows maximum uncertainties in the design variables, i.e., maximum magnitudes of $\bar{\delta}_i$, $i = 1, \ldots, d$.
Note that maximized $\bar{\delta}_i$, $i = 1, \ldots, d$, can be calculated using Equation (3.30) with the maximized δ'. Thus, the situation corresponds to the situation with knowledge-based magnitudes

for which $\bar{\delta}_i$, $i = 1, \ldots, d$, are assumed. Then, the worst-case complete system solution space of target designs $\check{\Omega}_{\mathrm{c,wc}}$ can be computed like above. If $\check{\Omega}_{\mathrm{c,wc}}$ is a singleton here, the robust target design is unique. Furthermore, the d-dimensional volume of $\check{\Omega}_{\mathrm{c,wc}}$ is always zero, as otherwise, δ' would not be optimal.

Nevertheless, flexibility in component target design can also be expressed using lower-dimensional volumes as a measure. To account for this, the objective function of problem (3.24) to obtain worst-case independent CSS of target designs must be perturbed for maximized $\bar{\delta}$. This can be done by replacing

$$\mathrm{vol}(\check{\Omega}^1_{\mathrm{wc}} \times \cdots \times \check{\Omega}^n_{\mathrm{wc}}) \text{ by } \prod_{k=1}^{n} (\mathrm{vol}(\check{\Omega}^k_{\mathrm{wc}}) + \varepsilon), \tag{3.33}$$

where $\varepsilon \geq 0$ is a *perturbation parameter*. The greater ε, the more weight is put on the lower-dimensional volumes, built by the Cartesian product of only parts of the CSS of target designs. Note however that $\check{\Omega}^k_{\mathrm{wc}} \neq \emptyset$ must be ensured here. Otherwise, no permissible system design exists within the Cartesian product of the CSS. If $\check{\Omega}^k_{\mathrm{wc}}$, $k = 1, \ldots, n$, are box-shaped, the replacement

$$\mathrm{vol}(\check{\Omega}^1_{\mathrm{wc}} \times \cdots \times \check{\Omega}^d_{\mathrm{wc}}) \text{ by } \prod_{i=1}^{d} (\check{x}^{\mathrm{u}}_{\mathrm{wc},i} - \check{x}^{\mathrm{l}}_{\mathrm{wc},i} + \varepsilon) \tag{3.34}$$

is preferred. In Equation (3.34), the single edge lengths of $\check{\Omega}^k_{\mathrm{wc}}$, $k = 1, \ldots, n$, i.e., one-dimensional volumes are weighted. Here, $\check{x}^{\mathrm{l}}_{\mathrm{wc},i}$ denote the lower bounds and $\check{x}^{\mathrm{u}}_{\mathrm{wc},i}$ the upper bounds of $\check{\Omega}^k_{\mathrm{wc}}$, $k = i$, $i = 1, \ldots, d$. For $\varepsilon = 0$, the real volume of the Cartesian product of CSS of target designs is computed in Equation (3.34). Similarly, the objective function to compute box-shaped worst-case dependent CSS of target designs can be replaced. For arbitrarily-shaped worst-case dependent CSS this is not necessary, as no optimization is required.
After computing the worst-case CSS of target designs, the CSS Ω^k, $k = 1, \ldots, n$, can be calculated using Equation (3.28). Note that for maximized $\bar{\delta}$, best-case CSS of target designs are usually not reasonable, as maximized $\bar{\delta}$ contains maximum upper bounds for uncertainties. However, they could be computed accordingly using Equation (3.27). In the following, an example of maximized $\bar{\delta}$ is considered.

Example 22. Given a system composed of two components. The component designs of both the first and the second component consist of one design variable each, i.e., $d^1 = d^2 = 1$, which means that box-shaped and arbitrarily-shaped CSS coincide, known from Example 21. Furthermore, the system design space is given by $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$ and a performance function by $f : \mathbb{R}^2 \to \mathbb{R},\ x \mapsto x_1 + 2x_2$ with threshold $f_{\mathrm{c}} = 2$. There are interval-type uncertainties in the controllable variables with unknown δ. Here, $\bar{\delta}$ is maximized with $\omega_1 = \omega_2 = 1$ and $\nu_1 = \nu_2 = 0$ by solving Equation (3.31), compare Example 9.

The worst-case complete system solution space is the singleton $\check{\Omega}_{\mathrm{c,wc}} = \{(0.3333, 0.3333)\}$. Thus the worst-case independent and worst-case dependent CSS coincide and are given by $\check{\Omega}^1_{\mathrm{wc}} = \{0.3333\}$ and $\check{\Omega}^2_{\mathrm{wc}} = \{0.3333\}$. They are visualized together with their corresponding CSS in Figure 35.

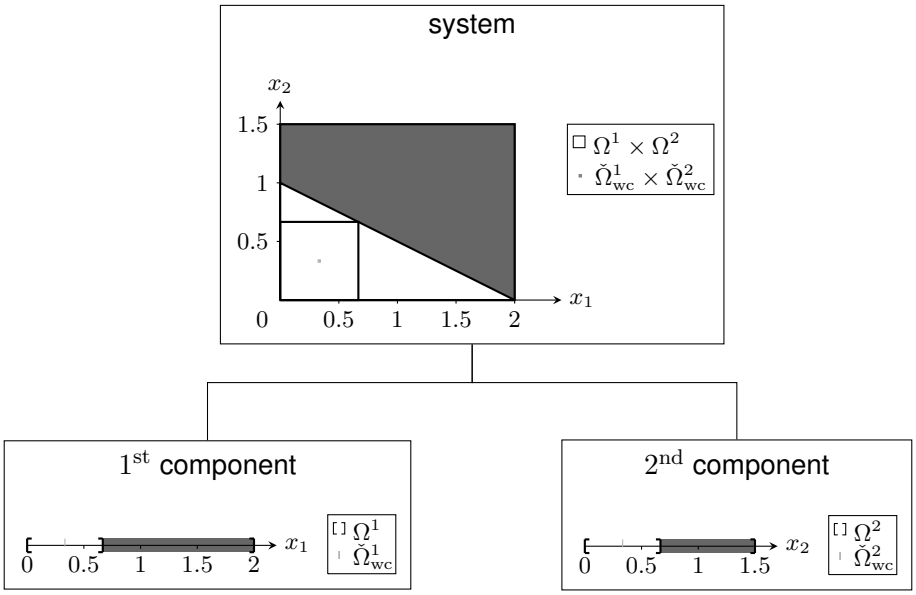


Figure 35 Optimal worst-case CSS of target designs and corresponding CSS of Example 22 at the component level and their Cartesian product at the system level.

3.4.3. Uncertainties in Uncontrollable Parameters

If uncertainties in uncontrollable parameters are present and the CSS are computed for nominal values $\check{p} \in \mathbb{R}^d$, a realized system design might be non-permissible due to these uncertainties, although it is within the complete system solution space Ω_{c} computed with $\check{p} \in \mathbb{R}^d$. In the following, uncertainties in uncontrollable parameters are considered and uncertainties in controllable variables are neglected at first. Recall that for interval-type uncertainties in uncontrollable variables, the true values of p_l can be found within intervals $[\check{p}_l - \gamma_l, \check{p}_l + \gamma_l]$ for which fixed values are assumed for $\check{p}_l \in \mathbb{R}$ and $\gamma_l \in \mathbb{R}^+_0$, $l = 1, \ldots, q$. The subsequent results regarding box-shaped solution spaces under interval-type uncertainties in uncontrollable parameters were already published in [24] for worst-case scenarios and extended in [26] for best-case scenarios by the author of this thesis.

Due to the absence of uncertainties in controllable variables, any designs within the worst-case complete system solution space $\Omega_{\mathrm{c,wc}}$ and the best-case complete system solution space $\Omega_{\mathrm{c,bc}}$, given by Equations (2.11) and (2.13), can be realized exactly. Within $\Omega_{\mathrm{c,wc}}$, every system design is permissible regardless of which values the entries of p assume in $\mathcal{U}^P(\check{p}, \gamma)$. For $\Omega_{\mathrm{c,bc}}$, at least one $p \in \mathcal{U}^P(\check{p}, \gamma)$ exists for every selected system design such that it is permissible, i.e., within the resulting complete system solution space for this $p \in \mathbb{R}^q$.
Again, the decisions to select a system design within $\Omega_{\mathrm{c,wc}}$ or $\Omega_{\mathrm{c,bc}}$ can be decoupled with the approaches from Sections 3.2 and 3.3 yielding *worst-case CSS* Ω^k_{wc} and *best-case CSS*

Ω_{bc}^k, $k = 1, \ldots, n$. This procedure ensures that uncertainties in uncontrollable parameters are considered in CSS and can be treated efficiently. The optimization problem to obtain *worst-case independent CSS* reads

$$\begin{aligned} &\underset{\Omega_{\mathrm{wc}}^1,\ldots,\Omega_{\mathrm{wc}}^n}{\text{maximize}} && \mathrm{vol}(\Omega_{\mathrm{wc}}^1 \times \cdots \times \Omega_{\mathrm{wc}}^n) \\ &\text{subject to} && \Omega_{\mathrm{wc}}^1 \times \cdots \times \Omega_{\mathrm{wc}}^n \subseteq \Omega_{\mathrm{c,wc}}. \end{aligned} \tag{3.35}$$

where $\Omega_{\mathrm{c,wc}}$ can be replaced by $\Omega_{\mathrm{c,bc}}$ to obtain best-case independent CSS Ω_{bc}^k, $k = 1, \ldots, n$. The optimization problem to obtain *worst-case dependent CSS* reads

$$\begin{aligned} &\underset{\Omega_{\mathrm{wc}}^k}{\text{maximize}} && \mathrm{vol}(\Omega_{\mathrm{wc}}^k) \\ &\text{subject to} && \Omega_{\mathrm{wc}}^k \subseteq \mathrm{proj}^k(\Omega_{\mathrm{c,wc}}) \end{aligned} \tag{3.36}$$

where again $\Omega_{\mathrm{c,wc}}$ can be replaced by $\Omega_{\mathrm{c,bc}}$ in order to obtain best-case dependent CSS Ω_{bc}^k, $k = 1, \ldots, n$. Given the worst-case and best-case complete system solution spaces, problems (3.35) and (3.36) can be solved similarly to problems (3.14) and (3.20) for linear performance functions which is presented in Section 4.2.
Using Problems (3.35) and (3.36) to compute worst-case and best-case CSS, it might be the case that $\Omega_{\mathrm{wc}}^k \not\subseteq \Omega_{\mathrm{bc}}^k$ holds for one or more $k \in \{1, \ldots, n\}$, although $\Omega_{\mathrm{c,wc}} \subseteq \Omega_{\mathrm{c,bc}}$ always holds. This arises from the fact that flexibility for component design is optimized independently for both worst-case and best-case CSS. One possibility to overcome this problem is to force

$$\Omega_{\mathrm{wc}}^k \subseteq \Omega_{\mathrm{bc}}^k, \tag{3.37}$$

$k = 1, \ldots, n$, by additional optimization constraints for the best-case CSS after computing worst-case CSS. The same could be done the other way around. Note that in this case, it is not necessarily guaranteed that the worst-case CSS are non-empty sets, even if the worst-case complete system solution space is a non-empty set. Similar results were obtained above for worst- and best-case CSS of target designs. In the following, an example is given in which both worst- and best-case, independent and dependent CSS are optimized.

Example 23. Given a system composed of two components. The component designs of both the first and the second component consist of one design variable each, i.e., $d^1 = d^2 = 1$, which means that box-shaped and arbitrarily-shaped CSS coincide, known from Example 21. Furthermore, the system design space is given by $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$ and a performance function by $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R},\ x \mapsto p_1 x_1 + p_2 x_2$ with threshold $f_{\mathrm{c}} = 2$. There are interval-type uncertainties in the uncontrollable parameters. Their nominal values are given by $\check{p}_1 = 1$ and $\check{p}_2 = 2$ and it holds $\gamma_1 = \gamma_2 = 0.2$, compare Example 3.
Optimal worst-case independent CSS can be computed as $\Omega_{\mathrm{wc}}^1 = [0, 0.8333]$ and $\Omega_{\mathrm{wc}}^2 =$

$[0, 0.4545]$, optimal best-case independent CSS as $\Omega^1_{\text{bc}} = [0, 1.25]$ and $\Omega^2_{\text{bc}} = [0, 0.5556]$, optimal worst-case dependent CSS as $\Omega^1_{\text{wc}} = [0, 1.6667]$ and $\Omega^2_{\text{wc}} = [0, 0.9091]$, and optimal best-case dependent CSS as $\Omega^1_{\text{bc}} = [0, 2]$ and $\Omega^2_{\text{bc}} = [0, 1.1111]$. They are visualized in Figure 36.

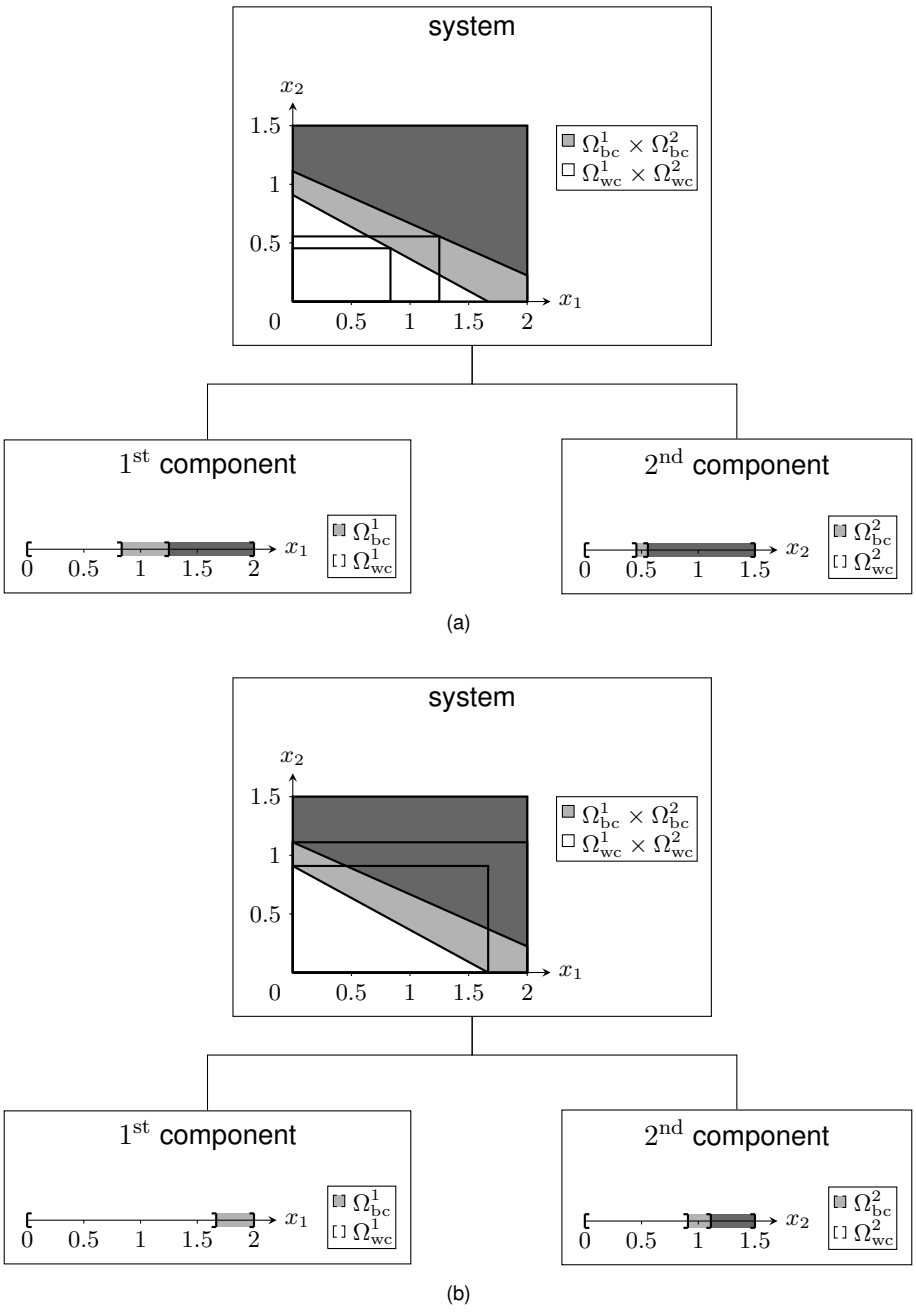


Figure 36 Optimal worst- and best-case (a) independent and (b) dependent CSS of Example 23 at the component level and their Cartesian product at the system level.

The worst-case and best-case CSS from Example 23 provide optimal flexibility for component design under interval-type uncertainties in uncontrollable parameters. In Figure 36, it is shown in which region flexibility can be added to worst-case CSS by computing best-case CSS. Outside the best-case CSS, no permissible component design exists. In particular, for independent CSS, it can be seen that the flexibility which is added to the first component

has a larger impact on the overall flexibility than the flexibility which is added to the second component.

3.4.4. Extensions and Summary of the General Approach

Furthermore, both uncertainties in controllable variables and uncertainties in uncontrollable parameters can be considered together, which was already published for box-shaped independent CSS in [26] by the author of this thesis. This can be accomplished by using the best- and worst-case complete system solution spaces of target designs for which both uncertainties in controllable variables and uncertainties in uncontrollable parameters are considered, i.e., by using Equations (2.12) and (2.14). In the case of knowledge-based magnitudes, the best- and worst-case CSS of target designs can be obtained by solving problems (3.24) and (3.25). Here, condition (3.37) must be fulfilled with

$$\Omega^k_{\mathrm{wc}} = \{x^k \in \Omega^k_{\mathrm{ds}} \mid \exists \check{x}^k \in [x^k - \bar{\delta}^k, x^k + \bar{\delta}^k] : \ \check{x}^k \in \check{\Omega}^k_{\mathrm{wc}}\}, \tag{3.38}$$

and

$$\Omega^k_{\mathrm{bc}} = \{x^k \in \Omega^k_{\mathrm{ds}} \mid \forall \check{x}^k \in [x^k - \bar{\delta}^k, x^k + \bar{\delta}^k] : \ \check{x}^k \in \check{\Omega}^k_{\mathrm{bc}}\}, \tag{3.39}$$

$k = 1, \ldots, n$, where the notes from above for inverse schemes must be considered. Furthermore, note that in doing so, the best-case is no longer aligned with the worst-case or vice versa for dependent CSS, compare Section 5.4.

In the case of maximized $\bar{\delta}$, problem (3.31) must be solved in which Ω_{c} is replaced by $\Omega_{\mathrm{c,wc}}$. Then, problems (3.24) and (3.25) can be solved by replacing the objective function according to (3.33) or (3.34).

In this section, no explicit example for the case of both uncertainties in controllable variables and uncontrollable parameters is given. Nevertheless, corresponding results are shown in Section 5.4. They merge the results of Examples 21 and 23.

After the worst- and best-case CSS of target designs are computed for $\bar{\delta}^k$, $k = 1, \ldots, n$, coupled design decisions under interval-type uncertainty for component target designs can be made similarly to the approaches shown in Section 2.3 by using component cost functions. If no further knowledge about the true values of δ_i, $i = 1, \ldots, d$ is available, component target designs $\check{x}^k$ should be always selected within $\check{\Omega}^k_{\mathrm{wc}}$ for $k = 1, \ldots, n$. Moreover, if the realized component designs x^k for the selected $\check{x}^k$ can be found within Ω^k_{wc} for $k = 1, \ldots, n$, they can be considered as permissible.

If further knowledge on δ_i becomes available after computing the worst- and best-case CSS of target designs, three different cases can be distinguished for selecting component target designs $\check{x}^k$, $k = 1, \ldots, n$:

(a) $\bar{\delta}_i > \delta_i$: The worst- and best-case CSS of target designs can be updated without computing new CSS. The volume of the corresponding worst-case CSS of target designs increases,

and so does the overall flexibility for component design in the worst-case. The volume of the corresponding best-case CSS of target designs decreases, and so does the overall flexibility for component design in the best-case. However, note that the updated flexibility is usually not optimal for the whole system.

(b) $\bar{\delta}_i < \delta_i$: The worst- and best-case CSS of target designs can be updated without computing new CSS of target designs if the updated worst-case CSS of target designs is non-empty. Otherwise, new CSS of target designs must be computed. Here, the volume of the corresponding worst-case CSS of target designs decreases, and so does the overall flexibility for component design in the worst-case. The volume of the corresponding best-case CSS of target designs increases, and so does the overall flexibility for component design in the best-case.

(c) $\bar{\delta}_i = \delta_i$: The worst- and best-case CSS of target designs remain the same and so does the overall flexibility for component design in both the worst- and best-case.

Note that these cases are only relevant for uncertainties in controllable variables. They are visualized in Figure 37 for $d^k = 1$, $k \in \{1, \ldots, n\}$, without taking uncertainties in uncontrollable parameters into account. Here, $\check{x}^{\mathrm{l}}_{\mathrm{wc},i}$ denote again the lower bounds and $\check{x}^{\mathrm{u}}_{\mathrm{wc},i}$ the upper bounds of $\check{\Omega}^k_{\mathrm{wc}}$, and $\check{x}^{\mathrm{l}}_{\mathrm{bc},i}$ denote the lower bounds and $\check{x}^{\mathrm{u}}_{\mathrm{bc},i}$ the upper bounds of $\check{\Omega}^k_{\mathrm{bc}}$, $k = i$, $i = 1, \ldots, d$.

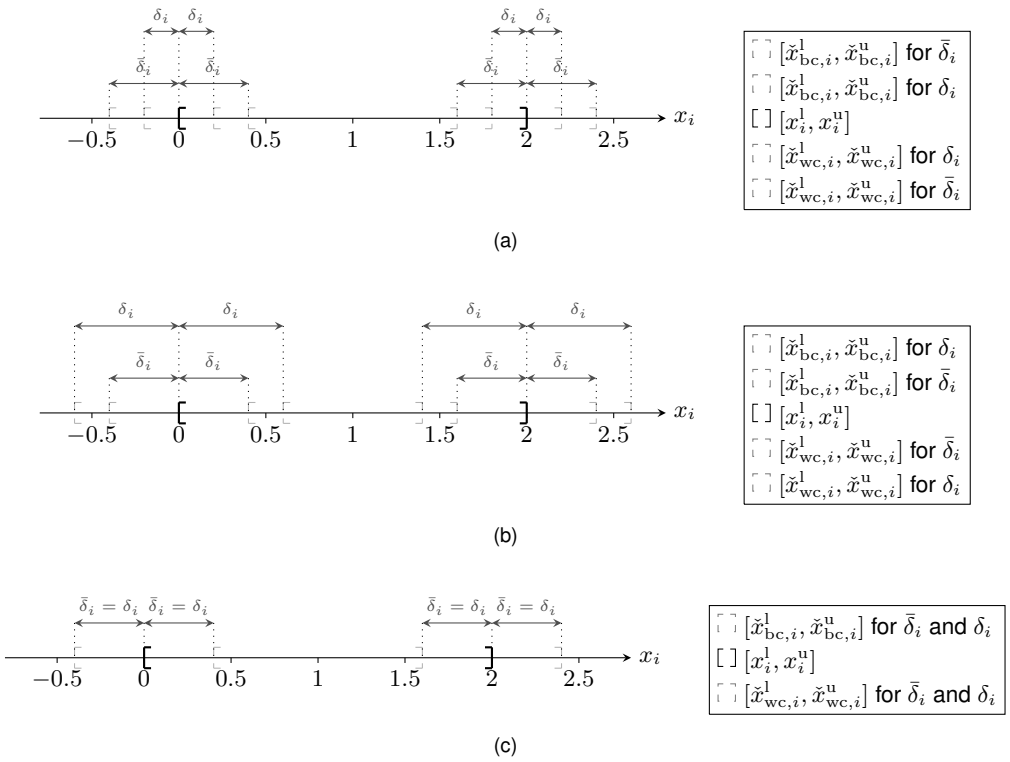


Figure 37 Changes in flexibility for component design with $d^k = 1$, $k \in \{1, \ldots, n\}$, due to gained knowledge about δ_i, $i = k$, with (a) $\bar{\delta}_i > \delta_i$ (b) $\bar{\delta}_i < \delta_i$ (c) $\bar{\delta}_i = \delta_i$. Note that in the legends, the corresponding CSS are listed according to their size, the largest is on top.

After updating the worst- and best-case CSS of target designs, coupled design decisions for component target designs can be made similarly to the approaches shown in Section 2.3. The overall procedure to obtain worst- and best-case CSS of designs and target designs for independent and dependent-decoupled design decisions under interval-type uncertainty is visualized in Figure 38.
In the following section, it is shown how the above perceptions must be adapted if fuzzy-type instead of interval-type uncertainty is considered for computing CSS.

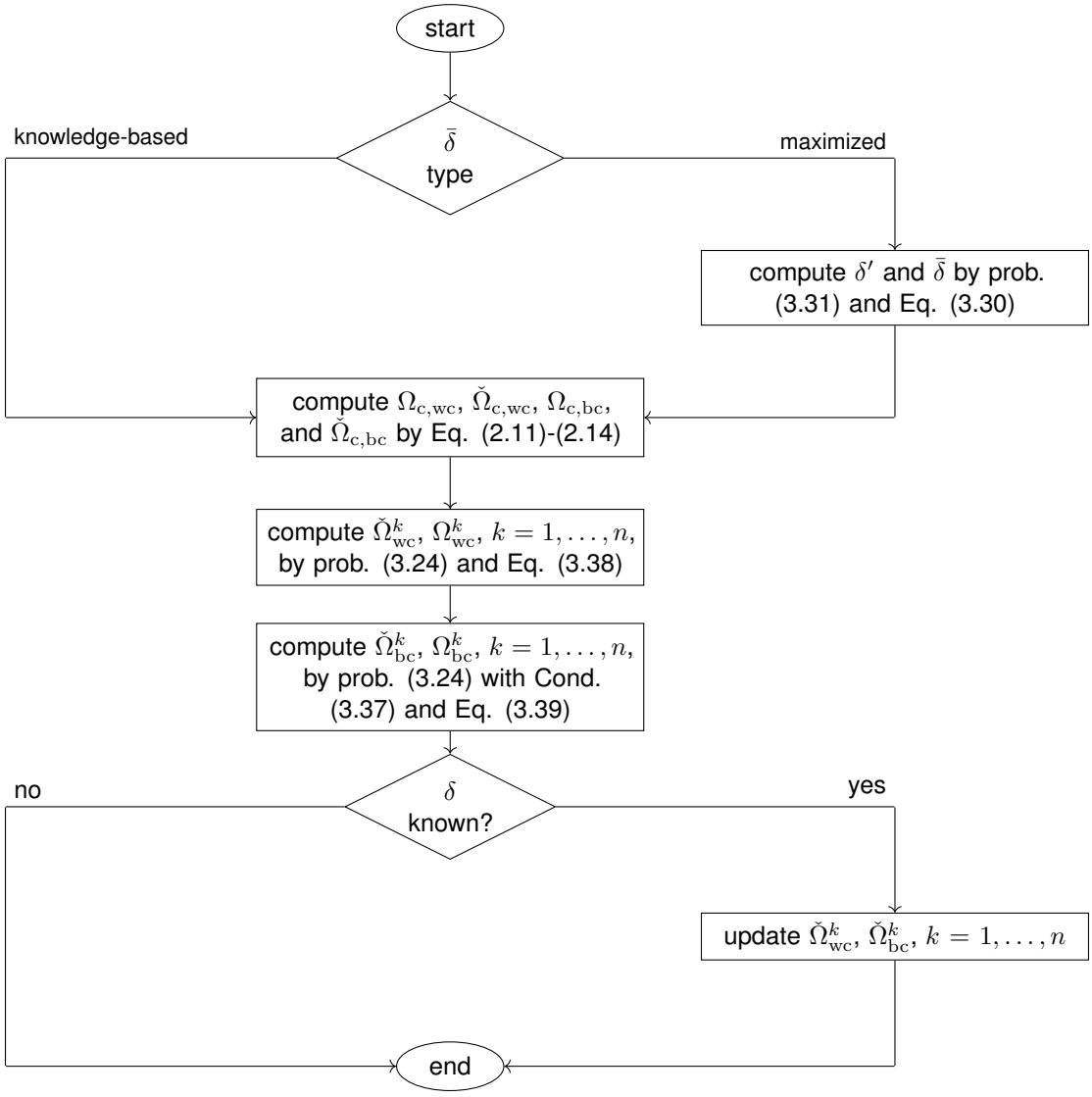


Figure 38 Flowchart to obtain worst- and best-case, independent and dependent CSS of designs and target designs. Note that dependent CSS are only computed for the first decision here.

3.5. CSS under Fuzzy-Type Uncertainty

Similar to the previous section, this section presents how epistemic uncertainty modeled as fuzzy-type uncertainty can be considered in the problem statements for independent and dependent CSS. Again, uncertainties in controllable variables are considered first. Then,

uncertainties in uncontrollable parameters are taken into account yielding a complete methodology for computing CSS under fuzzy-type uncertainty.

3.5.1. Uncertainties in Controllable Variables

As described above, the considerations are limited to uncertainties in controllable variables at first. Also, recall that for fuzzy-type uncertainties in controllable variables, the designers can only select the values of the target design variables $\check{x}_i$, $i = 1, \ldots, d$. Thus, a value α, given by Equation (2.15) or (2.16) with $\alpha \in [0, 1]$, can be assigned to any $x_i \in \mathbb{R}$. This indicates a degree of membership of belonging to the fuzzy uncertainty set of the controllable variable with respect to $\check{x}_i$, $i = 1, \ldots, d$. The membership functions are characterized by $\delta_{\alpha=0,i}$ and $\delta_{\alpha=1,i}$, which are only assumed here and denoted by $\bar{\delta}_{\alpha=0,i}$ and $\bar{\delta}_{\alpha=1,i}$, similar to δ_i for interval-type uncertainty, $i = 1, \ldots, d$. These magnitudes are collected in the d-dimensional vectors $\bar{\delta}_{\alpha=0}$ and $\bar{\delta}_{\alpha=1}$. Again, it is distinguished between knowledge-based and maximized magnitudes.

For knowledge-based $\bar{\delta}_{\alpha=0,i}$, $\bar{\delta}_{\alpha=0,i}$, $i = 1, \ldots, d$, the necessity-α complete system solution space of target designs is

$$\check{\Omega}_{\mathrm{c,nec},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \bar{\delta}_{\alpha=0}, \bar{\delta}_{\alpha=1}) : \; x \in \Omega_\mathrm{c}\}, \tag{3.40}$$

$\alpha \in [0, 1]$, see Equation (2.24). Then, the design decisions for selecting a target design within $\check{\Omega}_{\mathrm{c,nec},\alpha}$ can be decoupled yielding *necessity-α CSS of target designs* $\check{\Omega}^k_{\mathrm{nec},\alpha}$, $\alpha \in [0, 1]$, $k = 1, \ldots, n$. If $\check{\Omega}_{\mathrm{c,nec},\alpha}$ is a non-empty set with a positive volume, the optimization problem to obtain *necessity-α independent CSS of target designs* reads

$$\begin{aligned} \underset{\check{\Omega}^1_{\mathrm{nec},\alpha},\ldots,\check{\Omega}^n_{\mathrm{nec},\alpha}}{\text{maximize}} \quad & \mathrm{vol}(\check{\Omega}^1_{\mathrm{nec},\alpha} \times \cdots \times \check{\Omega}^n_{\mathrm{nec},\alpha}) \\ \text{subject to} \quad & \check{\Omega}^1_{\mathrm{nec},\alpha} \times \cdots \times \check{\Omega}^n_{\mathrm{nec},\alpha} \subseteq \check{\Omega}_{\mathrm{c,nec},\alpha} \end{aligned} \tag{3.41}$$

and the optimization problems to obtain *necessity-α dependent CSS of target designs* read

$$\begin{aligned} \underset{\check{\Omega}^k_{\mathrm{nec},\alpha}}{\text{maximize}} \quad & \mathrm{vol}(\check{\Omega}^k_{\mathrm{nec},\alpha}) \\ \text{subject to} \quad & \check{\Omega}^k_{\mathrm{nec},\alpha} \subseteq \mathrm{proj}^k(\check{\Omega}_{\mathrm{c,nec},\alpha}), \end{aligned} \tag{3.42}$$

$k = 1, \ldots, n$, for $\alpha \in [0, 1]$. In Section 4.2, it is discussed how Problems (3.41) and (3.42) can be solved according to Problems (3.14) and (3.20) for linear performance functions. Overall, a similar procedure as proposed for CSS under interval-type uncertainty can be applied to compute CSS under fuzzy-type uncertainty. Here, CSS Ω^k can be derived from $\check{\Omega}^k_{\mathrm{nec},\alpha}$ similarly to Equation (3.28), for which the flexibility for component design in the case of arbitrary-shaped CSS can be extended accordingly. Note that solving Problems (3.41) and (3.42) for different

$\alpha \in [0,1]$ does not necessarily yield the same deduced CSS Ω^k, $k = 1, \ldots, n$. However, this can be guaranteed by choosing a reference $\alpha \in [0,1]$, computing the corresponding $\check{\Omega}^k_{\mathrm{nec},\alpha}$ and CSS Ω^k, $k = 1, \ldots, n$, and deducing further necessity-α CSS of target designs for another α from these Ω^k, $k = 1, \ldots, n$. Similarly, *possibility-α CSS of target designs* $\check{\Omega}^k_{\mathrm{pos},\alpha}$ can be obtained.

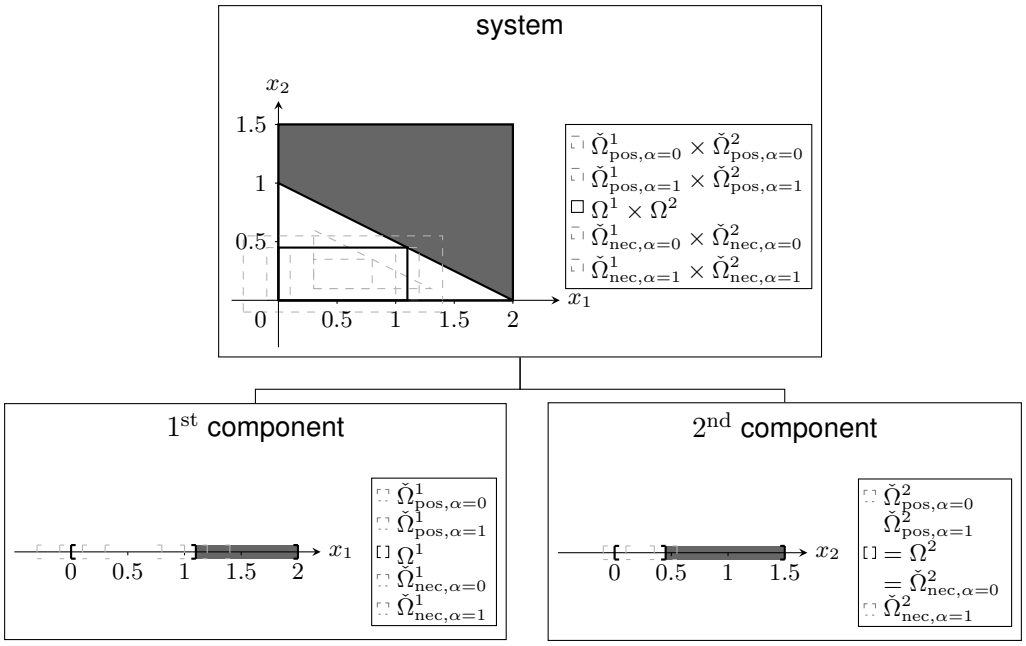
The necessity-1 case must be set as the reference to guarantee that the necessity-1 CSS of target designs are non-empty sets if the necessity-1 complete system solution space is a non-empty set. Then, all other necessity-α and possibility-α CSS of target designs are non-empty sets, too. In the following, an example which computes necessity-α and possibility-α CSS of target designs is given, $\alpha \in [0,1]$.

Example 24. Given a system composed of two components. The component designs of both the first and the second component consist of one design variable each, i.e., $d^1 = d^2 = 1$, which means that box-shaped and arbitrarily-shaped CSS coincide. Furthermore, the system design space is given by $\Omega_{\mathrm{ds}} = [0,2] \times [0,1.5]$ and a performance function by $f : \mathbb{R}^2 \to \mathbb{R},\ x \mapsto x_1 + 2x_2$ with threshold $f_{\mathrm{c}} = 2$. There are fuzzy-type uncertainties in the controllable variables with $\bar{\delta}_{\alpha=0,1} = 0.3$, $\bar{\delta}_{\alpha=0,2} = 0.1$ and $\bar{\delta}_{\alpha=1,1} = 0.1$, $\bar{\delta}_{\alpha=1,2} = 0$, compare Example 4. Choosing the necessity-α=1 case as a reference, optimal necessity-1 independent CSS of target designs can be computed as $\check{\Omega}^1_{\mathrm{nec},\alpha=1} = [0.3, 0.8]$ and $\check{\Omega}^2_{\mathrm{nec},\alpha=1} = [0.1, 0.35]$ and optimal necessity-1 dependent CSS of target designs as $\check{\Omega}^1_{\mathrm{nec},\alpha=1} = [0.3, 1.9]$ and $\check{\Omega}^2_{\mathrm{nec},\alpha=1} = [0.1, 0.6]$. They are visualized together with their corresponding CSS and necessity-0, possibility-1, and possibility-0 CSS of target designs in Figure 39. Note that the possibility-0 CSS of target designs must be modeled as open sets as $\check{\Omega}_{\mathrm{c,pos},\alpha=0}$ is an open set here, compare Sections 2.2 and 4.1.

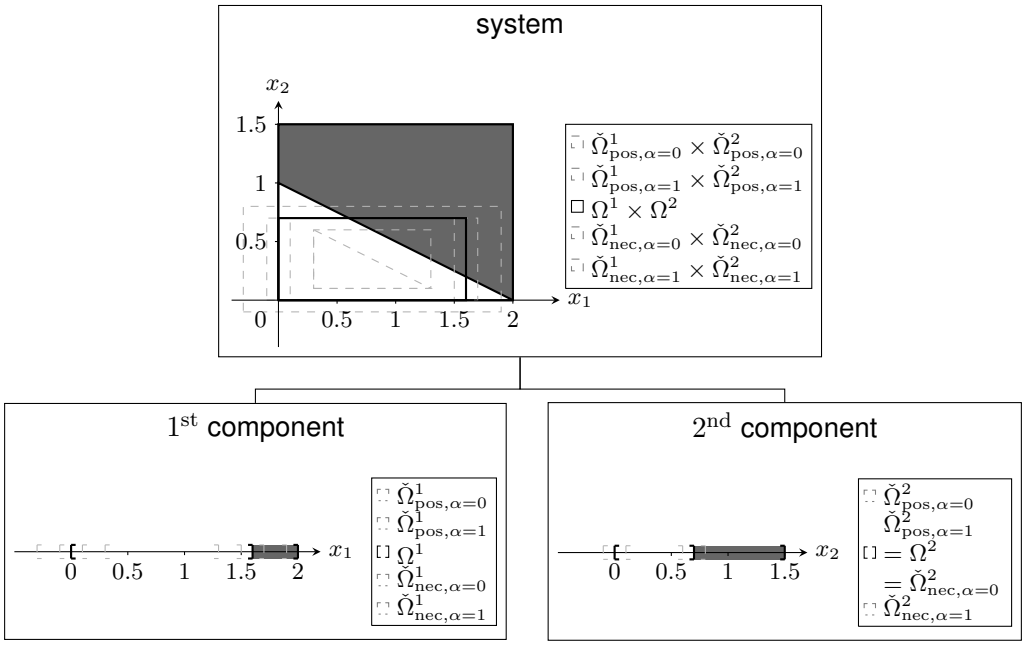
The results can be discussed similarly to Section 3.4. Furthermore, necessity-α and possibility-α CSS of target designs with $\alpha \in [0,1]$ are included in the follow-up discussion in Section 5.4. For maximized $\bar{\delta}_{\alpha=0,i}$, $\bar{\delta}_{\alpha=1,i}$, $i = 1, \ldots, d$, a maximum δ' must be computed for every $\alpha \in [0,1]$ similarly to problem (3.31) in which interval-type uncertainty is present. As no uncertainties in controllable parameters are considered here, the corresponding optimization problems are the same for every $\alpha \in [0,1]$ and so is δ'. Thus, for maximized $\bar{\delta}_{\alpha=0}$, $\bar{\delta}_{\alpha=1}$ without uncertainties in uncontrollable parameters, it holds $\bar{\delta}_{\alpha=0} = \bar{\delta}_{\alpha=1}$ which corresponds to the case of interval-type uncertainty. Therefore, Example 22 can be directly transferred to computing CSS under fuzzy-type uncertainty for which $\bar{\delta}_{\alpha=0,i}$, $\bar{\delta}_{\alpha=1,i}$, $i = 1, \ldots, d$, are maximized.

3.5.2. Uncertainties in Uncontrollable Parameters

Next, the focus is put on uncertainties in uncontrollable parameters and uncertainties in controllable variables are neglected. Recall that for fuzzy-type uncertainties in uncontrollable variables, a value $\alpha \in [0,1]$, is assigned to any $p_l \in \mathbb{R}$, $l = 1, \ldots, q$, that indicates a



(a)



(b)

Figure 39 Optimal necessity-1 (a) independent and (b) dependent CSS of target designs and corresponding CSS and necessity-0, possibility-1, and possibility-0 CSS of target designs of Example 24 at the component level and their Cartesian product at the system level. Note that in the legend, the solution spaces are listed according to their size, the largest is on top.

degree of membership of belonging to the fuzzy uncertainty set of the uncontrollable parameter.

Here, any design within the necessity-α and possibility-α complete system solution spaces $\Omega_{\mathrm{c,nec},\alpha}$ and $\Omega_{\mathrm{c,pos},\alpha}$, $\alpha \in [0,1]$, given by Equations (2.22) and (2.23), can be realized exactly. Again, the decisions to select a system design can be decoupled with the approaches from

Sections 3.2 and 3.3 yielding *necessity-α CSS* $\Omega^k_{\mathrm{nec},\alpha}$ and *possibility-α CSS* $\Omega^k_{\mathrm{pos},\alpha}$, $\alpha \in [0,1]$, $k = 1,\ldots,n$. The optimization problem to obtain *necessity-α independent CSS* reads

$$\begin{aligned} \underset{\Omega^1_{\mathrm{nec},\alpha},\ldots,\Omega^n_{\mathrm{nec},\alpha}}{\text{maximize}} \quad & \mathrm{vol}(\Omega^1_{\mathrm{nec},\alpha} \times \cdots \times \Omega^n_{\mathrm{nec},\alpha}) \\ \text{subject to} \quad & \Omega^1_{\mathrm{nec},\alpha} \times \cdots \times \Omega^n_{\mathrm{nec},\alpha} \subseteq \Omega_{\mathrm{c,nec},\alpha}. \end{aligned} \tag{3.43}$$

where $\Omega_{\mathrm{c,nec},\alpha}$ can be replaced by $\Omega_{\mathrm{c,pos},\alpha}$ in order to obtain possibility-α independent CSS $\Omega^k_{\mathrm{c,pos},\alpha}$, $k = 1,\ldots,n$, for $\alpha \in [0,1]$. The optimization problems to obtain *necessity-α dependent CSS* reads

$$\begin{aligned} \underset{\Omega^k_{\mathrm{wc}}}{\text{maximize}} \quad & \mathrm{vol}(\Omega^k_{\mathrm{nec},\alpha}) \\ \text{subject to} \quad & \Omega^k_{\mathrm{nec},\alpha} \subseteq \mathrm{proj}^k(\Omega_{\mathrm{c,nec},\alpha}) \end{aligned} \tag{3.44}$$

where again $\Omega_{\mathrm{c,nec},\alpha}$ can be replaced by $\Omega_{\mathrm{c,pos},\alpha}$ in order to obtain possibility-α dependent CSS $\Omega^k_{\mathrm{c,pos},\alpha}$ for $k = 1,\ldots,n$ and $\alpha \in [0,1]$. Given the necessity-α and possibility-α complete system solution spaces, problems (3.43) and (3.44) can be solved similarly to problems (3.14) and (3.20) for linear performance functions, see Section 4.2. To avoid $\Omega^k_{\mathrm{c,nec},\alpha} \not\subseteq \Omega^k_{\mathrm{c,nec},\alpha'}$ and $\Omega^k_{\mathrm{c,pos},\alpha'} \not\subseteq \Omega^k_{\mathrm{c,pos},\alpha}$, the conditions

$$\Omega^k_{\mathrm{c,nec},\alpha} \subseteq \Omega^k_{\mathrm{c,nec},\alpha'}, \tag{3.45}$$

and

$$\Omega^k_{\mathrm{c,pos},\alpha'} \subseteq \Omega^k_{\mathrm{c,pos},\alpha}, \tag{3.46}$$

can be introduced in problems (3.43) and (3.44) for $0 \leq \alpha' \leq \alpha \leq 1$, $k = 1,\ldots,n$. Similarly

$$\Omega^k_{\mathrm{c,nec},\alpha} \subseteq \Omega^k_{\mathrm{c,pos},\alpha'}, \tag{3.47}$$

can be introduced for all $\alpha, \alpha' \in [0,1]$ to avoid $\Omega^k_{\mathrm{c,nec},\alpha} \not\subseteq \Omega^k_{\mathrm{c,pos},\alpha'}$, $k = 1,\ldots,n$. Here, a reference α and the type of CSS must be chosen again. Choosing $\alpha = 1$ and necessity-α CSS guarantees that the necessity-1 CSS are non-empty sets if the necessity-1 complete system solution space is a non-empty set. Below, an example which computes necessity-α and possibility-α CSS is given.

Example 25. Given a system composed of two components. The component designs of both the first and the second component consist of one design variable each, i.e., $d^1 = d^2 = 1$, which means that box-shaped and arbitrarily-shaped CSS coincides. Furthermore, the system design space is given by $\Omega_{\mathrm{ds}} = [0,2] \times [0,1.5]$ and a performance function by $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R},\ x \mapsto p_1 x_1 + p_2 x_2$ with threshold $f_{\mathrm{c}} = 2$. There is fuzzy-type uncertainties in the uncontrollable parameters. Their nominal values are given by $\check{p}_1 = 1$ and $\check{p}_2 = 2$ and it

holds $\gamma_{\alpha=0,1}, \gamma_{\alpha=0,2} = 0.2$ and $\gamma_{\alpha=1,1}, \gamma_{\alpha=1,2} = 0.1$, compare Example 4.
Thus, optimal necessity-α independent CSS can be calculated as $\Omega^1_{\mathrm{c,nec},\alpha=1} = [0, 0.8333]$, $\Omega^2_{\mathrm{c,nec},\alpha=1} = [0, 0.4545]$, $\Omega^1_{\mathrm{c,nec},\alpha=0} = [0, 0.9091]$, $\Omega^2_{\mathrm{c,nec},\alpha=0} = [0, 0.4762]$, optimal possibility-$\alpha$ independent CSS as $\Omega^1_{\mathrm{c,pos},\alpha=1} = [0, 1.1111]$, $\Omega^2_{\mathrm{c,pos},\alpha=1} = [0, 0.5263]$, $\Omega^1_{\mathrm{c,pos},\alpha=0} = [0, 1.25)$, $\Omega^2_{\mathrm{c,pos},\alpha=0} = [0, 0.5556)$, optimal necessity-$\alpha$ dependent CSS can be calculated as $\Omega^1_{\mathrm{c,nec},\alpha=1} = [0, 1.6667]$, $\Omega^2_{\mathrm{c,nec},\alpha=1} = [0, 0.9091]$, $\Omega^1_{\mathrm{c,nec},\alpha=0} = [0, 1.8182]$, $\Omega^2_{\mathrm{c,nec},\alpha=0} = [0, 0.9524]$, and optimal possibility-α dependent CSS as $\Omega^1_{\mathrm{c,pos},\alpha=1} = [0, 2]$, $\Omega^2_{\mathrm{c,pos},\alpha=1} = [0, 1.0526]$, $\Omega^1_{\mathrm{c,pos},\alpha=0} = [0, 2]$, $\Omega^2_{\mathrm{c,pos},\alpha=0} = [0, 1.1111)$. They are visualized in Figure 40. Note that the possibility-0 CSS are modeled as non-closed sets as $\check{\Omega}_{\mathrm{c,pos},\alpha=0}$ is a non-closed set here, compare Sections 2.2 and 4.1.

The results can be discussed similarly to Section 3.4. Furthermore, necessity-α and possibility-α CSS with $\alpha \in [0, 1]$ are included in the follow-up discussion in Section 5.4.

3.5.3. Extensions and Summary of the General Approach

As done for interval-type uncertainty, uncertainties in controllable variables and uncertainties in uncontrollable parameters can also be considered together for fuzzy-type uncertainty. This is accomplished by using the necessity-α and possibility-α complete system solution spaces of target designs for which both uncertainties in controllable variables and uncertainties in uncontrollable parameters are considered, i.e., Equations (2.24) and (2.25).
In the case of knowledge-based $\bar{\bar{\delta}}_{\alpha=0}$, $\bar{\bar{\delta}}_{\alpha=1}$, the necessity-$\alpha$ and possibility-α CSS of target designs can be obtained by solving problems (3.41) and (3.42). Here, the conditions (3.45) and (3.46) must be taken into account if multiple α are considered and the condition (3.47) if both necessity-α and possibility-α solution spaces are considered. If $\bar{\bar{\delta}}_{\alpha=0}$, $\bar{\bar{\delta}}_{\alpha=1}$ shall be maximized, problem (3.31) must be solved in which Ω_c is replaced by $\Omega_{\mathrm{c,nec},\alpha}$, $\alpha \in [0, 1]$. For fixed values of α, $\check{\Omega}_{\mathrm{c,nec},\alpha}$ can be calculated using Equations (2.24) and (3.32). If multiple $\alpha \in [0, 1]$ are considered, condition (3.45) must also be taken into account. Even then, $\check{\Omega}_{\mathrm{c,nec},\alpha} \subseteq \check{\Omega}_{\mathrm{c,nec},\alpha'}$ is not necessarily fulfilled for $0 \leq \alpha' \leq \alpha \leq 1$. This could be introduced as an additional condition. Finally, problems (3.41) and (3.42) can be solved by replacing the objective function according to (3.33) or (3.34).
Again, no example for the case of both uncertainties in controllable variables and uncontrollable parameters is given here. However, the corresponding results merge the results of Examples 24 and 25 similarly to Section 3.4.

After the necessity-α and possibility-α CSS of target designs with $\alpha \in [0, 1]$ are obtained, coupled design decisions under fuzzy-type uncertainty for the component target design can be made similarly to the approaches shown in Section 2.3 by using component cost functions. If the realized component designs x^k for the selected $\check{x}^k$ can found be within $\Omega^k_{\mathrm{nec},\alpha=1}$ for $k = 1, \ldots, n$, they can be considered as permissible for all $\alpha \in [0, 1]$. Otherwise, they might be only permissible with a certain necessity or possibility $\alpha \in [0, 1]$. If more knowledge on the

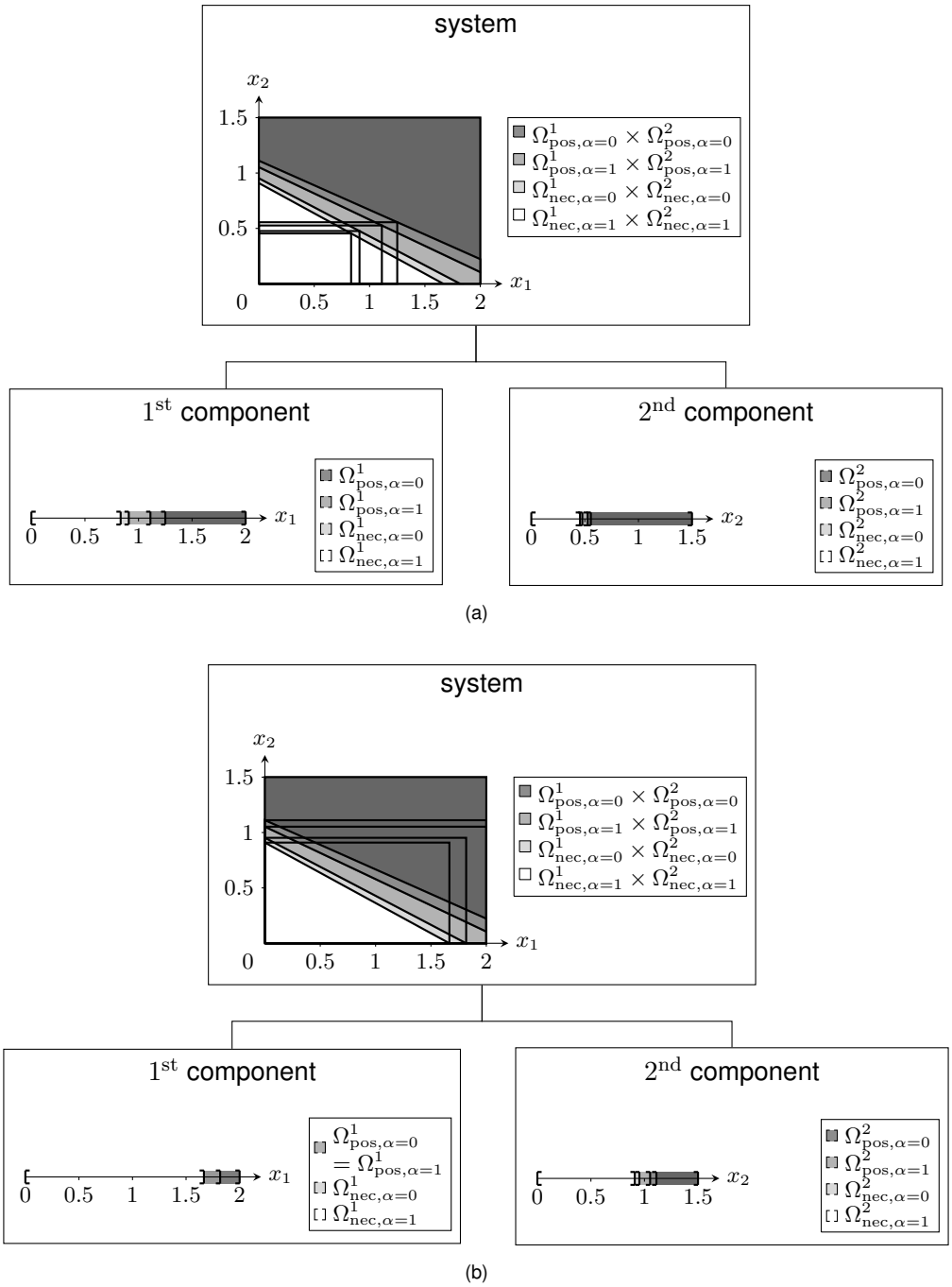


Figure 40 Optimal necessity-α and possibility-α (a) independent and (b) dependent CSS of Example 25 at the component level and their Cartesian product at the system level with $\alpha \in \{0, 1\}$.

values of $\delta_{\alpha=0,i}$, $\delta_{\alpha=1,i}$ becomes available, three different cases can be distinguished again for selecting component target designs $\check{x}^k$, $k = 1, \ldots, n$. Let $\bar{\delta}_{\alpha,i} = (1-\alpha)\bar{\delta}_{\alpha=0,i} + \alpha\bar{\delta}_{\alpha=1,i}$, $\delta_{\alpha,i} = (1-\alpha)\delta_{\alpha=0,i} + \alpha\delta_{\alpha=1,i}$ hold in the following for $i = 1, \ldots, d$:

(a) $\bar{\delta}_{\alpha,i} > \delta_{\alpha,i}$: The volume of the corresponding necessity-$1-\alpha$ CSS increases, and so does the overall flexibility for component design in the necessity-$1-\alpha$ case. The volume of the possibility-α CSS decreases, and so does the overall flexibility for component design in the possibility-α case.

(b) $\bar{\delta}_{\alpha,i} < \delta_{\alpha,i}$: The volume of the necessity-$1-\alpha$ CSS of target designs decreases, and so does the overall flexibility for component design in the necessity-$1-\alpha$ case. The volume of the possibility-α increases, and so does the overall flexibility for component design in the possibility-α case. If the corresponding necessity-$1-\alpha$ CSS of target designs is empty, new CSS of target designs must be computed.

(c) $\bar{\delta}_{\alpha,i} = \delta_{\alpha,i}$: The corresponding necessity-$1-\alpha$ and possibility-α CSS of target designs remain the same and so does the overall flexibility for component design in both the necessity-$1-\alpha$ and possibility-α case.

The three cases can be visualized similarly to Figure 37 for $d^k = 1$, $k \in \{1, \dots, n\}$. Note that the necessity-α and possibility-α CSS of target designs can both decrease and increase for different $\alpha \in [0, 1]$. After updating the necessity-α and possibility-α CSS of target designs with $\alpha \in [0, 1]$, coupled design decisions for each component target design can be made similarly as discussed in Section 2.3. As the overall procedure to obtain necessity-α and possibility-α CSS of designs and target designs for independent- and dependent-decoupled design decisions under fuzzy-type uncertainty with $\alpha \in [0, 1]$ is similar to interval-type uncertainty, see Figure 38, it is not visualized explicitly here.
The next chapter investigates important properties of complete system solution spaces, especially necessity-α and possibility-α complete system solution spaces with $\alpha \in [0, 1]$. Furthermore, simplifications and algorithms that solve the problem statements of this chapter are proposed. Note that no further methods related to systems engineering are presented in the next chapter. Thus, if the reader is rather interested in the application than in these mathematical considerations, the following chapter can be skipped.

4. PROPERTIES & ALGORITHMS: CSS for Specific Performance Functions

This chapter considers mathematical details that are relevant to compute component solution spaces. First, properties of the corresponding complete system solution spaces are investigated and the missing definitions for these solution spaces under fuzzy-type uncertainty are stated. Then, the problem statements to compute component solution spaces are simplified for specific system performance functions in order to enable their numerical computation. Furthermore, useful numerical tools are provided to obtain efficient algorithms for their computation.

4.1. Properties of Complete System Solution Spaces

This section shows that the complete system solution spaces under absence of uncertainty, and interval- and fuzzy-type uncertainty are compact for continuous system performance functions. This property can be transferred to component solution spaces. Furthermore, the missing definitions for the complete system solution spaces under fuzzy-type uncertainty are stated here.

4.1.1. Under Absence of Uncertainty

If the complete system solution space and component solution spaces defined by Equation (3.10) can be shown to be compact, the problem statements to compute CSS have an optimal solution. In Section A.1, the conditions for a compact set in $\mathbb{R}^d$ are stated. Otherwise, if the complete system solution space was open for example and the CSS were defined as compact sets, no optimal CSS would exist. Note that CSS defined by Equation (3.9) are always compact. In the following, continuous system performance functions are assumed. Under the absence of uncertainty, the complete system solution space Ω_{c} is compact for continuous system performance functions. This is shown in the subsequent theorem.

Theorem 1. *Given the complete system solution space Ω_{c}, defined by Equation (2.4), with continuous system performance functions f_j, $j = 1, \ldots, m$. Then, Ω_{c} is compact.*

Proof. For the set Ω_{c}, it holds

$$\begin{aligned} \Omega_{\mathrm{c}} &= \{x \in \Omega_{\mathrm{ds}} \mid f(x) \leq f_{\mathrm{c}}\} \\ &= \Omega_{\mathrm{ds}} \cap \{x \in \mathbb{R}^d \mid f(x) \leq f_{\mathrm{c}}\}. \end{aligned} \tag{4.1}$$

The set Ω_{c} is a subset of the compact set $\Omega_{\mathrm{ds}} = [x_{\mathrm{ds}}^{\mathrm{l}}, x_{\mathrm{ds}}^{\mathrm{u}}]$ and is therefore bounded. In addition, it is closed as the inverse image $f^{-1}(-\infty, f_{\mathrm{c}}]$ of the closed set $(-\infty, f_{\mathrm{c}}]$ is closed in $\mathbb{R}^d$ for continuous f and the intersection of closed sets, i.e., Ω_{ds} and $f^{-1}(-\infty, f_{\mathrm{c}}]$, is closed, too. □

This property transfers to CSS which are defined by Equation (3.10) for continuous component performance functions as they have the same mathematical structure. Note that in general more complex system performance functions could be taken into account for the complete system solution space and component solution spaces as well. In order to obtain a simple and more general mathematical formulation of the complete system solution space, the constraints on the design variables and the constraints on the responses from Section 2.1 can be merged using constraint functions.

Definition 1. *Given the system design space* $\Omega_{\mathrm{ds}} = [x_{\mathrm{ds}}^{\mathrm{l}}, x_{\mathrm{ds}}^{\mathrm{u}}]$ *with performance functions in* $f = (f_1, \ldots, f_m) : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^m : \ (x, p) \mapsto z = f(x, p)$ *and thresholds in* $f_{\mathrm{c}}(p) \in \mathbb{R}^m$*. The constraint functions* F_j*,* $j = 1, \ldots, m + 2d$*, with* $F = (F_1, \ldots, F_{m+2d})$ *are defined by*

$$\begin{aligned} F : \mathbb{R}^d \times \mathbb{R}^q &\to \mathbb{R}^{m+2d} : \\ (x, p) &\mapsto (f(x, p) - f_{\mathrm{c}}(p), -x + x_{\mathrm{ds}}^{\mathrm{l}}, x - x_{\mathrm{ds}}^{\mathrm{u}}). \end{aligned} \tag{4.2}$$

The constraint functions F_j, $j = 1, \ldots, m+2d$, have the property that they are continuous if the system performance functions f_j, $j = 1, \ldots, m$, are continuous. Furthermore, the constraint function can be used to define the constraints on the design variables and the responses as stated above.

Theorem 2. *For* $(x, p) \in \mathbb{R}^d \times \mathbb{R}^q$*, it is* $x \in \Omega_{\mathrm{ds}}$ *and* $f(x, p) \leq f_{\mathrm{c}}$ *if and only if* $F(x, p) \leq 0$ *holds component-wise.*

Proof. This follows directly from the definition of the system design space for which

$$x \in \Omega_{\mathrm{ds}} \Leftrightarrow -x + x_{\mathrm{ds}}^{\mathrm{l}} \leq 0 \wedge x - x_{\mathrm{ds}}^{\mathrm{u}} \leq 0 \tag{4.3}$$

holds component-wise. □

Thus, for fixed values of $p \in \mathbb{R}^q$, the complete system solution space can be reformulated as $\Omega_{\mathrm{c}} = \{x \in \mathbb{R}^d : \ F(x, p) \leq 0\}$, which is helpful for the following investigations. Next, properties of the complete system solution spaces under interval-type uncertainty are considered.

4.1.2. Interval-Type Uncertainty

Under interval-type uncertainty, worst- and best-case complete system solution spaces of designs and target designs are considered. These can also be reformulated using the constraint functions as stated in Theorem 3.

Theorem 3. *Given the worst- and best-case complete system solution spaces* $\Omega_{\mathrm{c,wc}}$ *and* $\Omega_{\mathrm{c,bc}}$ *and the worst- and best-case complete system solution spaces of target designs* $\check{\Omega}_{\mathrm{c,wc}}$ *and* $\check{\Omega}_{\mathrm{c,bc}}$*, defined by Equations (2.11)-(2.14). Then,* $\Omega_{\mathrm{c,wc}}$*,* $\Omega_{\mathrm{c,bc}}$*,* $\check{\Omega}_{\mathrm{c,wc}}$*, and* $\check{\Omega}_{\mathrm{c,bc}}$ *can be reformulated as*

(a)

$$\Omega_{\mathrm{c,wc}} = \{x \in \mathbb{R}^d \mid \forall p \in \mathcal{U}^P(\check{p}, \gamma) : \ F(x,p) \leq 0\}, \tag{4.4}$$

(b)

$$\Omega_{\mathrm{c,bc}} = \{x \in \mathbb{R}^d \mid \exists p \in \mathcal{U}^P(\check{p}, \gamma) : \ F(x,p) \leq 0\}, \tag{4.5}$$

(c)

$$\check{\Omega}_{\mathrm{c,wc}} = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X(\check{x}, \delta) \ \forall p \in \mathcal{U}^P(\check{p}, \gamma) : \ F(x,p) \leq 0\}, \tag{4.6}$$

(d)

$$\check{\Omega}_{\mathrm{c,bc}} = \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}^X(\check{x}, \delta) \ \exists p \in \mathcal{U}^P(\check{p}, \gamma) : \ F(x,p) \leq 0\}. \tag{4.7}$$

Proof. With Theorem 2, it holds

$$\begin{aligned}
\Omega_{\mathrm{c,wc}} &= \{x \in \Omega_{\mathrm{ds}} \mid \forall p \in \mathcal{U}^P(\check{p}, \gamma) : \ f(x,p) \leq f_{\mathrm{c}}(p)\} \\
&= \{x \in \mathbb{R}^d \mid x \in \Omega_{\mathrm{ds}} \ \wedge \ \forall p \in \mathcal{U}^P(\check{p}, \gamma) : \ f(x,p) \leq f_{\mathrm{c}}(p)\} \\
&= \{x \in \mathbb{R}^d \mid \forall p \in \mathcal{U}^P(\check{p}, \gamma) : \ x \in \Omega_{\mathrm{ds}} \ \wedge \ f(x,p) \leq f_{\mathrm{c}}(p)\} \\
&= \{x \in \mathbb{R}^d \mid \forall p \in \mathcal{U}^P(\check{p}, \gamma) : \ F(x,p) \leq 0\},
\end{aligned} \tag{4.8}$$

$$\begin{aligned}
\Omega_{\mathrm{c,bc}} &= \{x \in \Omega_{\mathrm{ds}} \mid \exists p \in \mathcal{U}^P(\check{p}, \gamma) : \ f(x,p) \leq f_{\mathrm{c}}(p)\} \\
&= \{x \in \mathbb{R}^d \mid x \in \Omega_{\mathrm{ds}} \ \wedge \ \exists p \in \mathcal{U}^P(\check{p}, \gamma) : \ f(x,p) \leq f_{\mathrm{c}}(p)\} \\
&= \{x \in \mathbb{R}^d \mid \exists p \in \mathcal{U}^P(\check{p}, \gamma) : \ x \in \Omega_{\mathrm{ds}} \ \wedge \ f(x,p) \leq f_{\mathrm{c}}(p)\} \\
&= \{x \in \mathbb{R}^d \mid \exists p \in \mathcal{U}^P(\check{p}, \gamma) : \ F(x,p) \leq 0\},
\end{aligned} \tag{4.9}$$

$$\begin{aligned}\check{\Omega}_{\mathrm{c,wc}} &= \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X(\check{x}, \delta) :\ x \in \Omega_{\mathrm{c,wc}}\} \\ &= \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X(\check{x}, \delta)\ \forall p \in \mathcal{U}^P(\check{p}, \gamma) :\ F(x,p) \leq 0\},\end{aligned} \tag{4.10}$$

$$\begin{aligned}\check{\Omega}_{\mathrm{c,bc}} &= \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}^X(\check{x}, \delta) :\ x \in \Omega_{\mathrm{c,bc}}\} \\ &= \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}^X(\check{x}, \delta)\ \exists p \in \mathcal{U}^P(\check{p}, \gamma) :\ F(x,p) \leq 0\}\end{aligned} \tag{4.11}$$

as $\mathcal{U}^X(\check{x}, \delta), \mathcal{U}^P(\check{p}, \gamma) \neq \emptyset$. □

These reformulations help to prove that the worst- and best-case complete system solution spaces of designs and target designs are also compact for continuous system performance functions. This is shown in the subsequent theorem.

Theorem 4. *Given the worst- and best-case complete system solution spaces $\Omega_{\mathrm{c,wc}}$ and $\Omega_{\mathrm{c,bc}}$ and the worst- and best-case complete system solution spaces of target designs $\check{\Omega}_{\mathrm{c,wc}}$ and $\check{\Omega}_{\mathrm{c,bc}}$, defined by Equations (2.11)-(2.14), with continuous system performance functions f. Then, $\Omega_{\mathrm{c,wc}}$, $\Omega_{\mathrm{c,bc}}$, $\check{\Omega}_{\mathrm{c,wc}}$, and $\check{\Omega}_{\mathrm{c,bc}}$ are compact.*

Proof. The sets $\Omega_{\mathrm{c,wc}}$, $\Omega_{\mathrm{c,bc}}$, $\check{\Omega}_{\mathrm{c,wc}}$, and $\check{\Omega}_{\mathrm{c,bc}}$ are subsets of the bounded d-dimensional interval $[x^{\mathrm{l}}_{\mathrm{ds}} - \delta, x^{\mathrm{u}}_{\mathrm{ds}} + \delta]$ and are therefore bounded.

First, assume that $\check{\Omega}_{\mathrm{c,wc}}$ is not closed. Then, there is a sequence $(\check{x}_n)_{n\in\mathbb{N}} \subset \check{\Omega}_{\mathrm{c,wc}}$ with $\lim_{n\to\infty} \check{x}_n = \check{x}$ and $\check{x} \in \mathbb{R}^d \backslash \check{\Omega}_{\mathrm{c,wc}}$.
Let $(\check{x}_n)_{n\in\mathbb{N}} \subset \check{\Omega}_{\mathrm{c,wc}}$ be such a sequence. Then, for $\check{x}_n \in \check{\Omega}_{\mathrm{c,wc}}$, $n \in \mathbb{N}$, it holds

$$\forall x \in \mathcal{U}^X(\check{x}_n, \delta)\ \forall p \in \mathcal{U}^P(\check{p}, \gamma)\ \forall j \in \{1, \ldots, m + 2d\} :\ F_j(x,p) \leq 0 \tag{4.12}$$

and for $\check{x} \in \mathbb{R}^d \backslash \check{\Omega}_{\mathrm{c,wc}}$, it holds

$$\exists x' \in \mathcal{U}^X(\check{x}, \delta)\ \exists p' \in \mathcal{U}^P(\check{p}, \gamma)\ \exists j' \in \{1, \ldots, m + 2d\} :\ F_{j'}(x', p') > 0. \tag{4.13}$$

The vector x' can be written as $x' = \check{x} + \delta'$ where $\delta' = (\delta'_1, \ldots, \delta'_d)$. Hence, $(x'_n)_{n\in\mathbb{N}}$ is a sequence defined by $x'_n = \check{x}_n + \delta'$, $n \in \mathbb{N}$, with $\lim_{n\to\infty} x'_n = x'$. Furthermore, it is $F_{j'}(x'_n, p') \leq 0$ for all $n \in \mathbb{N}$ and $F_{j'}(x', p') > 0$, which is a contradiction to the continuity of F. Therefore, $\check{\Omega}_{\mathrm{c,wc}}$ is closed and so is $\Omega_{\mathrm{c,wc}}$ as a special case of $\check{\Omega}_{\mathrm{c,wc}}$ with $\gamma = 0$.

Second, assume that $\check{\Omega}_{\mathrm{c,bc}}$ is not closed. Then, there is also a sequence $(\check{x}_n)_{n\in\mathbb{N}} \subset \check{\Omega}_{\mathrm{c,bc}}$ with $\lim_{n\to\infty} \check{x}_n = \check{x}$ and $\check{x} \in \mathbb{R}^d \backslash \check{\Omega}_{\mathrm{c,bc}}$.
Let $(\check{x}_n)_{n\in\mathbb{N}} \subset \check{\Omega}_{\mathrm{c,bc}}$ be such a sequence. Then, for $\check{x}_n \in \check{\Omega}_{\mathrm{c,bc}}$, $n \in \mathbb{N}$, it holds

$$\exists x \in \mathcal{U}^X(\check{x}_n, \delta)\ \exists p \in \mathcal{U}^P(\check{p}, \gamma)\ \forall j \in \{1, \ldots, m+2d\} :\ F_j(x,p) \leq 0. \tag{4.14}$$

and for $\check{x} \in \mathbb{R}^d \backslash \check{\Omega}_{\mathrm{c,bc}}$, it holds

$$\forall x \in \mathcal{U}^X(\check{x}, \delta)\ \forall p \in \mathcal{U}^P(\check{p}, \gamma)\ \exists j' \in \{1, \ldots, m+2d\} :\ F_{j'}(x,p) > 0. \tag{4.15}$$

Now, sequences $(\delta_n)_{n\in\mathbb{N}} \subset [-\delta, \delta]$ and $(\gamma_n)_{n\in\mathbb{N}} \subset [-\gamma, \gamma]$ can be defined such that $F(\check{x}_n + \delta_n, \check{p}_n + \gamma_n) \leq 0$ holds component-wise for all $n \in \mathbb{N}$. As $(\delta_n)_{n\in\mathbb{N}}$ and $(\gamma_n)_{n\in\mathbb{N}}$ are bounded sequences, there are convergent subsequences $(\delta_{n_k})_{k\in\mathbb{N}} \subset [-\delta, \delta]$ with $\lim_{k\to\infty} \delta_{n_k} = \delta'$, $\delta' \in [-\delta, \delta]$, and $(\gamma_{n_k})_{k\in\mathbb{N}} \subset [-\gamma, \gamma]$ with $\lim_{k\to\infty} \gamma_{n_k} = \gamma'$, $\gamma' \in [-\gamma, \gamma]$, after the Bolzano-Weierstrass theorem, cf. [51]. With the continuity of F, it is also $F(\check{x} + \delta', \check{p} + \gamma') \leq 0$, which is a contradiction to (4.15). Therefore, $\check{\Omega}_{\mathrm{c,bc}}$ is closed and so is $\Omega_{\mathrm{c,bc}}$ as a special case of $\check{\Omega}_{\mathrm{c,bc}}$ with $\gamma = 0$. □

In the following, properties of the complete system solution spaces under fuzzy-type uncertainty are investigated.

4.1.3. Under Fuzzy-Type Uncertainty with $\alpha \in (0,1)$

Under fuzzy-type uncertainty, necessity-α and possibility-α complete system solution spaces are considered, $\alpha \in [0,1]$. First, the case $\alpha \in (0,1)$ is considered, followed by the case $\alpha \in \{0,1\}$ in the subsequent subsection. Note that in Section 2.2, only properties of necessity-α and possibility-α complete system solution spaces for $\alpha \in (0,1)$ were stated. Here, the missing definitions are provided using also constraint functions.

Definition 2. *Given the constraint functions F from Definition 1, fuzzy sets $X(\check{x})$ and P, both defined with the membership functions from Equation (2.17), and $\alpha \in (0,1)$. Then,*

(a) the possibility-α complete system solution space is defined as

$$\Omega_{\mathrm{c,pos},\alpha} = \{x \in \mathbb{R}^d \mid \mathrm{pos}(\{P \mid F(x,P) \leq 0\}) \geq \alpha\}, \tag{4.16}$$

(b) the necessity-α complete system solution space is defined as

$$\Omega_{\mathrm{c,nec},\alpha} = \{x \in \mathbb{R}^d \mid \mathrm{nec}(\{P \mid F(x,P) \leq 0\}) \geq \alpha\}, \tag{4.17}$$

(c) the possibility-α complete system solution space of target designs is defined as

$$\check{\Omega}_{\mathrm{c,pos},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \mathrm{pos}(\{(X(\check{x}), P) \mid F(X(\check{x}), P) \leq 0\}) \geq \alpha\}, \tag{4.18}$$

(d) and the necessity-α complete system solution space of target designs is defined as

$$\check{\Omega}_{\mathrm{c,nec},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \mathrm{nec}(\{(X(\check{x}), P) \mid F(X(\check{x}), P) \leq 0\}) \geq \alpha\}. \tag{4.19}$$

In order to obtain the equivalence of the definitions of the necessity-α and possibility-α complete system solution spaces from Definition 2 and the properties, stated in Section 2.2, i.e., Equations (2.23)-(2.24), for $\alpha \in (0, 1)$, Lemma 1 can help.

Lemma 1. *Given the system design space $\Omega_{\mathrm{ds}} = [x^{\mathrm{l}}_{\mathrm{ds}}, x^{\mathrm{u}}_{\mathrm{ds}}]$, the performance functions with $f = (f_1, \ldots, f_m) : \mathbb{R}^d \times \mathbb{R}^q \to \mathbb{R}^m : \ (x, p) \mapsto z = f(x, p)$ and threshold $f_{\mathrm{c}}(p) \in \mathbb{R}^m$ which together define the constraint functions F, see Definition 1. Furthermore, let the α-cuts $\mathcal{U}^X_\alpha(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$ and $\mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})$ of the fuzzy uncertainty sets be defined according to Equations (2.18) and (2.20) with $\check{x}_i \in \mathbb{R}$, $\delta_{\alpha=0,i}, \delta_{\alpha=1,i} \in \mathbb{R}^+_0$, $i = 1 \ldots, d$, and $\check{p}_l \in \mathbb{R}$, $\gamma_{\alpha=0,l}, \gamma_{\alpha=1,l} \in \mathbb{R}^+_0$, $l = 1, \ldots, q$, for $\alpha \in (0, 1)$. Then, it holds*

(a)

$$\begin{aligned}
&\{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}^X_\alpha(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) \ \exists p \in \mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ F(x, p) \leq 0\} \\
&\quad = \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}^X_\alpha(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) : \\
&\qquad x \in \{x \in \Omega_{\mathrm{ds}} \mid \exists p \in \mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ f(x, p) \leq f_{\mathrm{c}}(p)\}\}.
\end{aligned} \tag{4.20}$$

(b)

$$\begin{aligned}
&\{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) \ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ F(x, p) \leq 0\} \\
&\quad = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) : \\
&\qquad x \in \{x \in \Omega_{\mathrm{ds}} \mid \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ f(x, p) \leq f_{\mathrm{c}}(p)\}\},
\end{aligned} \tag{4.21}$$

Proof. A proof can be done similarly to the proof of Theorem 3 for $\alpha \in (0, 1)$. □

Finally, the above-mentioned equivalences for the necessity-α and possibility-α complete system solution spaces, $\alpha \in (0, 1)$ can be proven. This is done in Theorem 5.

Theorem 5. *Given the necessity-α and possibility-α complete system solution spaces from Definition 2 and the corresponding α-cuts $\mathcal{U}^X_\alpha(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$ and $\mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})$ of the*

fuzzy uncertainty sets from Equations (2.18) and (2.20) with $\check{x}_i \in \mathbb{R}$, $\delta_{\alpha=0,i}, \delta_{\alpha=1,i} \in \mathbb{R}_0^+$, $i = 1 \ldots, d$, *and* $\check{p}_l \in \mathbb{R}$, $\gamma_{\alpha=0,l}, \gamma_{\alpha=1,l} \in \mathbb{R}_0^+$, $l = 1, \ldots, q$, *where* $\check{x}$ *is variable and* $\check{p}$ *is fixed. Furthermore, let the system performance functions in* f *be continuous. Thus, it holds*

(a)

$$\Omega_{\mathrm{c,pos},\alpha} = \{x \in \Omega_{\mathrm{ds}} \mid \exists p \in \mathcal{U}_\alpha^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ f(x,p) \leq f_{\mathrm{c}}(p)\}, \tag{4.22}$$

(b)

$$\Omega_{\mathrm{c,nec},\alpha} = \{x \in \Omega_{\mathrm{ds}} \mid \forall p \in \mathcal{U}_{1-\alpha}^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ f(x,p) \leq f_{\mathrm{c}}(p)\}, \tag{4.23}$$

(c)

$$\check{\Omega}_{\mathrm{c,pos},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}_\alpha^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) : \ x \in \Omega_{\mathrm{c,pos},\alpha}\}, \tag{4.24}$$

(d)

$$\check{\Omega}_{\mathrm{c,nec},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}_{1-\alpha}^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) : \ x \in \Omega_{\mathrm{c,nec},\alpha}\} \tag{4.25}$$

for $\alpha \in (0,1)$.

Proof. Let $X(\check{x}')$, P, and Z be fuzzy sets. Here, Z is defined as $Z = F(X,P)$. Using Definition 6, stated in Section A.2, the membership function of Z is

$$\mu^Z(z) = \begin{cases} \sup\limits_{(x,p) \in F^{-1}(z)} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) & \text{if } F^{-1}(z) \neq \emptyset \\ 0 & \text{else,} \end{cases} \tag{4.26}$$

$z \in \mathbb{R}^{m+2d}$.

First,

$$\check{\Omega}_{\mathrm{c,pos},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \exists x \in \mathcal{U}_\alpha^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) \ \exists p \in \mathcal{U}_\alpha^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) : \ F(x,p) \leq 0\}$$

is shown for $\alpha \in (0,1)$.
"$\subseteq$": Let $\check{\Omega}_{\mathrm{c,pos},\alpha} \neq \emptyset$ and $\check{x}' \in \check{\Omega}_{\mathrm{c,pos},\alpha}$. It holds

$$\begin{aligned}
& \mathrm{pos}(\{(X,P) : \ F(X,P) \leq 0\}) \geq \alpha && (4.27)\\
\Rightarrow\ & \mathrm{pos}(\{Z : \ Z \leq 0\}) \geq \alpha && (4.28)\\
\Rightarrow\ & \sup_{z \leq 0} \mu^Z(z) \geq \alpha && (4.29)\\
\Rightarrow\ & \sup_{z \leq 0} \sup_{(x,p) \in F^{-1}(z)} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) \geq \alpha && (4.30)\\
\Rightarrow\ & \sup_{(x,p) \in F^{-1}((-\infty,0]^m)} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) \geq \alpha, && (4.31)
\end{aligned}$$

where $(-\infty, 0]^m \subset \mathbb{R}^m$ is a closed set, defined by the Cartesian product over m sets which are all given by $(-\infty, 0] \subset \mathbb{R}$. As $\sup_{p\in T} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) = 0$ holds for $(x,p) \in ([\check{p}-\gamma_{\alpha=0}, \check{p}+\gamma_{\alpha=0}]\times[\check{x}'-\delta_{\alpha=0}, \check{x}'+\delta_{\alpha=0}])^{\mathrm{c}}$, where the superscript c denotes the complement, Equation (4.31) is equivalent to

$$\sup_{p\in T} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) \geq \alpha \tag{4.32}$$

for $T = F^{-1}((-\infty, 0]^m) \cap ([\check{x}' - \delta_{\alpha=0}, \check{x}' + \delta_{\alpha=0}] \times [\check{p} - \gamma_{\alpha=0}, \check{p} + \gamma_{\alpha=0}])$, where T is a closed set as an intersection of closed sets. Furthermore, $\min(\mu^{X(\check{x}')}(x), \mu^P(p))$ is continuous on the set $[\check{x}' - \delta_{\alpha=0}, \check{x}' + \delta_{\alpha=0}] \times [\check{p} - \gamma_{\alpha=0}, \check{p} + \gamma_{\alpha=0}]$. Thus, T has a maximum point and from Equation (4.32) it follows

$$\max_{p\in T} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) \geq \alpha \tag{4.33}$$

$$\Rightarrow \exists (x,p) \in T:\ \min(\mu^{X(\check{x}')}(x), \mu^P(p)) \geq \alpha \tag{4.34}$$

$$\Rightarrow \exists (x,p) \in T:\ \mu^{X(\check{x}')}(x) \geq \alpha \ \wedge\ \mu^P(p) \geq \alpha \tag{4.35}$$

$$\Rightarrow \exists (x,p) \in T:\ x \in \mathcal{U}^X_\alpha(\check{x}', \delta_{\alpha=0}, \delta_{\alpha=1}) \ \wedge\ p \in \mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) \tag{4.36}$$

$$\Rightarrow \exists x \in \mathcal{U}^X_\alpha(\check{x}', \delta_{\alpha=0}, \delta_{\alpha=1})\ \exists p \in \mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0. \tag{4.37}$$

Hence, it is $\check{x}' \in \{\check{x} \in \mathbb{R}^d:\ \exists x \in \mathcal{U}^X_\alpha(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \exists p \in \mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0\}$ and $\check{\Omega}_{\mathrm{c,pos},\alpha} \subseteq \{\check{x} \in \mathbb{R}^d:\ \exists x \in \mathcal{U}^X_\alpha(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \exists p \in \mathcal{U}^P_\alpha(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0\}$.
For $\check{\Omega}_{\mathrm{c,pos},\alpha} = \emptyset$, "$\supseteq$" can be used.
"$\supseteq$": A proof can be done similarly to "$\subseteq$".
With Lemma 1, (a) and (c) follow.

Second,

$$\check{\Omega}_{\mathrm{c,nec},\alpha} = \{\check{x} \in \mathbb{R}^d \mid \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0\}$$

is shown for $\alpha \in (0,1)$.
"$\subseteq$": Let $\check{\Omega}_{\mathrm{c,nec},\alpha} \neq \emptyset$ and $\check{x}' \in \check{\Omega}_{\mathrm{c,nec},\alpha}$. It holds,

$$\mathrm{nec}(\{(X,P):\ F(X,P) \leq 0\}) \geq \alpha \tag{4.38}$$

$$\Rightarrow \mathrm{nec}(\{Z:\ Z \leq 0\}) \geq \alpha \tag{4.39}$$

$$\Rightarrow \inf_{z>0} (1 - \mu^Z(z)) \geq \alpha \tag{4.40}$$

$$\Rightarrow \sup_{z>0} \mu^Z(z) \leq 1 - \alpha \tag{4.41}$$

$$\Rightarrow \sup_{(x,p)\in F^{-1}((0,\infty)^m)} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) \leq 1 - \alpha, \tag{4.42}$$

where $(0,\infty)^m \subset \mathbb{R}^m$ is an open set, defined by the Cartesian product over m sets which are all given by $(0,\infty) \subset \mathbb{R}$. Suppose there is a $(x',p') \in F^{-1}((0,\infty)^m)$ with $\min(\mu^{X(\check{x}')}(x'), \mu^P(p')) = 1 - \alpha$. Without loss of generality, consider $\delta_{\alpha=0,i} > \delta_{\alpha=1,i}$, $i = 1, \ldots, d$, and $\gamma_{\alpha=0,l} > \gamma_{\alpha=1,l}$,

$l = 1, \ldots, q$ as for $\alpha \in (0,1)$, there is no x_i with $\mu^{X(\check{x}'),i}(x_i) = 1-\alpha$ for $\delta_{\alpha=0,i} = \delta_{\alpha=1,i}$, $i \in \{1,\ldots,d\}$, or p_l with $\mu_l^P(p_l) = 1-\alpha$ for $\gamma_{\alpha=0,l} = \gamma_{\alpha=1,l}$, $l \in \{1,\ldots,q\}$. It holds $(x',p') \in U$ with $U = F^{-1}((0,\infty)^m) \cap ((\check{x}'-\delta_{\alpha=0}, \check{x}'+\delta_{\alpha=0}) \times (\check{p}-\gamma_{\alpha=0}, \check{p}+\gamma_{\alpha=0}))$. Furthermore, there is a $t \in \mathbb{R}$ with $(x',p')+t(\check{x}'-x', \check{p}-p') \in U$ with $\min(\mu^{X(\check{x}')}(\check{x}'+t(\check{x}'-x')), \mu^P(\check{p}+t(\check{p}-p'))) > 1-\alpha$ as U is open and $\min(\mu^{X(\check{x}')}(\cdot), \mu^P(\cdot))$ is monotonically increasing on U in direction $(\check{x}'-x', \check{p}-p')$. This is a contradiction and it follows

$$\forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}', \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ (x,p) \in U^{\mathrm{c}} \tag{4.43}$$

$$\Rightarrow \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}', \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0, \tag{4.44}$$

where the superscript c denotes the complement. Hence, it is

$$\check{x}' \in \{\check{x} \in \mathbb{R}^d :\ \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0\} \tag{4.45}$$

and

$$\begin{aligned}\check{\Omega}_{\mathrm{c,nec},\alpha} \subseteq \{\check{x} \in \mathbb{R}^d :\ &\forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\\ &F(x,p) \leq 0\}.\end{aligned} \tag{4.46}$$

For $\check{\Omega}_{\mathrm{c,nec},\alpha} = \emptyset$, "$\supseteq$" can be used.
"$\supseteq$": Let $\{\check{x} \in \mathbb{R}^d :\ \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0\} \neq \emptyset$ and $\check{x}' \in \{\check{x} \in \mathbb{R}^d :\ \forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0\}$. It holds,

$$\forall x \in \mathcal{U}^X_{1-\alpha}(\check{x}', \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}):\ F(x,p) \leq 0, \tag{4.47}$$

i.e., $\mathcal{U}^X_{1-\alpha}(\check{x}', \delta_{\alpha=0}, \delta_{\alpha=1}) \times \mathcal{U}^P_{1-\alpha}(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) \subseteq F^{-1}((-\infty,0]^m)$. Thus,

$$\sup_{(x,p) \in F^{-1}((0,\infty)^m)} \min(\mu^{X(\check{x}')}(x), \mu^P(p)) \leq 1-\alpha, \tag{4.48}$$

which is equal to Equation (4.42). The rest of this proof can be done similarly to the proof of "$\subseteq$".
With Lemma 1, (b) and (d) follow. □

Similar to the complete system solution spaces under absence of uncertainty and interval-type uncertainty, it can be shown for the necessity-α and possibility-α complete system solution spaces that they are compact if the system performance functions are continuous, $\alpha \in (0,1)$.

Theorem 6. *Given the necessity-α and possibility-α complete system solution spaces $\Omega_{\mathrm{c,nec},\alpha}$ and $\Omega_{\mathrm{c,pos},\alpha}$ and the necessity-α and possibility-α complete system solution spaces of target designs $\check{\Omega}_{\mathrm{c,nec},\alpha}$ and $\check{\Omega}_{\mathrm{c,pos},\alpha}$, defined in Equations (4.16)-(4.19) for $\alpha \in (0,1)$, with continuous system performance functions in f, i.e., continuous constraint functions in F. Then, $\Omega_{\mathrm{c,nec},\alpha}$, $\Omega_{\mathrm{c,pos},\alpha}$, $\check{\Omega}_{\mathrm{c,nec},\alpha}$, and $\check{\Omega}_{\mathrm{c,pos},\alpha}$ are compact for $\alpha \in (0,1)$.*

Proof. Using Theorem 5, a proof for $\alpha \in (0,1)$ can be done similarly to the proof of Theorem 4. □

Next, necessity-α and possibility-α complete system solution spaces are considered for $\alpha \in \{0,1\}$.

4.1.4. Under Fuzzy-Type Uncertainty with $\alpha \in \{0,1\}$

As done in Section 2.2, the necessity-α and possibility-α complete system solution spaces for $\alpha \in \{0,1\}$ can be defined using Equations (2.23)-(2.24), i.e., Equations (4.22)-(4.25), and the corresponding α-cuts of the fuzzy uncertainty sets from Equations (2.18)-(2.21). Then, the properties stated in the following theorem hold.

Theorem 7. *Let the necessity-α and possibility-α complete system solution spaces be defined by Equations (4.22)-(4.25) with corresponding α-cuts of the fuzzy uncertainty sets from Equations (2.18)-(2.21) for $\alpha \in \{0,1\}$. Furthermore, let the system performance functions in f be continuous, P be the fuzzy set of the uncontrollable parameters, and $X(\check{x})$ be the fuzzy set of the target designs $\check{x}$, both defined with the membership functions from Equation (2.17). Then, it holds*

(a)

$$\mathrm{pos}(\{P \mid F(x,P) \leq 0\}) > 0 \tag{4.49}$$

for all $x \in \Omega_{\mathrm{c,pos},0}$ and

$$\mathrm{pos}(\{(X(\check{x}),P) \mid F(X(\check{x}),P) \leq 0\}) > 0 \tag{4.50}$$

or all $\check{x} \in \check{\Omega}_{\mathrm{c,pos},0}$,

(b)

$$\mathrm{pos}(\{P \mid F(x,P) \leq 0\}) = 1 \tag{4.51}$$

for all $x \in \Omega_{\mathrm{c,pos},1}$ and

$$\mathrm{pos}(\{(X(\check{x}),P) \mid F(X(\check{x}),P) \leq 0\}) = 1; \tag{4.52}$$

for all $\check{x} \in \check{\Omega}_{\mathrm{c,pos},1}$,

(c)

$$\mathrm{nec}(\{P \mid F(x,P) \leq 0\}) = 1 \tag{4.53}$$

for all $x \in \Omega_{\mathrm{c,nec},0}$ if $\gamma_{\alpha=0} = \gamma_{\alpha=1}$ and

$$\mathrm{nec}(\{(X(\check{x}),P) \mid F(X(\check{x}),P) \leq 0\}) = 1 \tag{4.54}$$

for all $\check{x} \in \check{\Omega}_{\mathrm{c,nec},0}$ *if additionally* $\delta_{\alpha=0} = \delta_{\alpha=1}$,

$$\mathrm{nec}(\{P \mid F(x,P) \leq 0\}) \geq 0 \tag{4.55}$$

for all $x \in \Omega_{\mathrm{c,nec},0}$ *and*

$$\mathrm{nec}(\{(X(\check{x}),P) \mid F(X(\check{x}),P) \leq 0\}) \geq 0 \tag{4.56}$$

for all $\check{x} \in \check{\Omega}_{\mathrm{c,nec},0}$ *else, with*

$$\mathrm{nec}(\{P \mid F(x,P) \leq 0\}) > 0 \tag{4.57}$$

for all $x \in \Omega_{\mathrm{c,nec},0}$ *with* $F(x,p) < 0$ *for all* $p \in \mathcal{U}_1^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})$ *and*

$$\mathrm{nec}(\{(X(\check{x}),P) \mid F(X(\check{x}),P) \leq 0\}) > 0 \tag{4.58}$$

for all $\check{x} \in \check{\Omega}_{\mathrm{c,nec},0}$ *with* $F(x,p) < 0$ *for all* $x \in \mathcal{U}_1^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$, $p \in \mathcal{U}_1^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})$,

(d)

$$\mathrm{nec}(\{P \mid f(x,P) \leq 0\}) = 1 \tag{4.59}$$

for all $x \in \Omega_{\mathrm{c,nec},1}$ *and*

$$\mathrm{nec}(\{(X(\check{x}),P) :\ F(X(\check{x}),P) \leq 0\}) = 1 \tag{4.60}$$

for all $\check{x} \in \check{\Omega}_{\mathrm{c,nec},1}$.

Proof. Properties (a) and (b) can be proven similarly to "$\supseteq$" of the first part of the proof of Theorem 5. Equations (4.53) and (4.54) of Property (c) can be proven similarly to "$\supseteq$" of the second part of the proof of Theorem 5 as $\mathcal{U}_1^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) = \mathcal{U}_\alpha^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})$ and $\mathcal{U}_1^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) = \mathcal{U}_\alpha^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1})$ for $\alpha \in [0,1)$.

Let $\alpha = 0$ and $\check{x} \in \check{\Omega}_{\mathrm{c,nec},0}$ with

$$\forall x \in \mathcal{U}_1^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}_1^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) :\ F(x,p) < 0, \tag{4.61}$$

i.e., $\mathcal{U}_1^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) \times \mathcal{U}_1^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) \subseteq F^{-1}((-\infty, 0)^m)$. Thus,

$$\sup_{(x,p) \in F^{-1}([0,\infty)^m)} \min(\mu^{X(\check{x})}(x), \mu^P(p)) < 1 \tag{4.62}$$

$$\Rightarrow \sup_{(x,p) \in F^{-1}((0,\infty)^m)} \min(\mu^{X(\check{x})}(x), \mu^P(p)) < 1 \tag{4.63}$$

$$\Rightarrow \mathrm{nec}(\{(X(\check{x}),P) :\ F(X(\check{x}),P) \leq 0\}) > 0 \tag{4.64}$$

holds similarly to "$\supseteq$" of the first part of the proof of Theorem 5. Hence, the Equations (4.57) and (4.58) from Property (c) follow. Note that the Equations (4.55) and (4.56) are always

fulfilled.

Now, let $\alpha = 1$ and $\check{x} \in \check{\Omega}_{\mathrm{c,nec},1}$. Then, it holds

$$\forall x \in \mathcal{U}_0^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1})\ \forall p \in \mathcal{U}_0^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) :\ F(x,p) \leq 0, \tag{4.65}$$

i.e., $\mathcal{U}_0^X(\check{x}, \delta_{\alpha=0}, \delta_{\alpha=1}) \times \mathcal{U}_0^P(\check{p}, \gamma_{\alpha=0}, \gamma_{\alpha=1}) \subseteq F^{-1}((-\infty, 0]^m)$. Thus,

$$\sup_{(x,p)\in F^{-1}((0,\infty)^m)} \min(\mu^{X(\check{x})}(x), \mu^P(p)) = 0 \tag{4.66}$$

$$\Rightarrow \mathrm{nec}(\{(X(\check{x}), P) :\ F(X(\check{x}), P) \leq 0\}) = 1 \tag{4.67}$$

holds similarly to "$\supseteq$" of the first part of the proof of Theorem 5. Hence, property (d) follows. □

Like the complete system solution spaces considered before, it can be shown that the necessity-α and possibility-α complete system solution spaces, $\alpha \in \{0, 1\}$, are compact if the system performance functions are continuous. However, this does not hold for the possibility-0 complete system solution space and the possibility-0 complete system solution space of target designs in general. The corresponding results are stated in the subsequent theorem.

Theorem 8. *Given the necessity-α and possibility-α complete system solution spaces $\Omega_{\mathrm{c,nec},\alpha}$ and $\Omega_{\mathrm{c,pos},\alpha}$ and the necessity-α and possibility-α complete system solution spaces of target designs $\check{\Omega}_{\mathrm{c,nec},\alpha}$ and $\check{\Omega}_{\mathrm{c,pos},\alpha}$, defined by (4.22)-(4.25) with $\alpha \in \{0, 1\}$, with continuous system performance functions f, i.e., continuous constraint functions F. Then, $\Omega_{\mathrm{c,nec},\alpha}$ and $\check{\Omega}_{\mathrm{c,nec},\alpha}$ are compact for $\alpha \in \{0, 1\}$, $\Omega_{\mathrm{c,pos},\alpha}$ and $\check{\Omega}_{\mathrm{c,pos},\alpha}$ are compact for $\alpha = 1$ and if $\delta_{\alpha=0} = \delta_{\alpha=1}$ and $\gamma_{\alpha=0} = \gamma_{\alpha=1}$ also for $\alpha = 0$.*

Proof. A proof can be done similarly to the proof of Theorem 4. □

Usually, the possibility-0 complete system solution space $\Omega_{\mathrm{c,pos},0}$ and the possibility-0 complete system solution space of target designs $\check{\Omega}_{\mathrm{c,pos},0}$ are not compact as in general, there exists a sequence $(\check{x}_n)_{n\in\mathbb{N}} \subset \check{\Omega}_{\mathrm{c,pos},0}$ with $\lim_{n\to\infty} \check{x}_n = \check{x}$ and $\check{x} \in \mathbb{R}^d \backslash \check{\Omega}_{\mathrm{c,pos},0}$, which shows that they are non-closed sets. In this case, the problem statements for computing possibility-0 CSS of designs and target designs must be reformulated using non-closed sets for these CSS, see Examples 24 and 25.

After checking the compactness of the complete system solution spaces under absence of uncertainty, interval- and fuzzy type uncertainty, which guarantees the existence of optimal

CSS, the underlying problem statements are investigated further in the following to enable an efficient computation. In this regard, the problem statements to compute CSS are simplified for specific continuous system performance functions. Furthermore, corresponding properties of the resulting problem statements are investigated in order to apply suitable numerical algorithms to solve them.

4.2. Simplification and Properties of CSS Problem Statements

This section simplifies the problem statements to compute box- and arbitrarily-shaped independent, and box- and arbitrarily-shaped dependent CSS under absence of uncertainty, interval- and fuzzy type uncertainty. Here, system performance functions which are linear in the controllable variables and monotonic in the uncontrollable parameters are taken into account. For this case, it is shown that the problem statements are convex optimization problems for which any local optimum is a global one.

4.2.1. Box-Shaped Independent CSS

First, the absence of uncertainty is considered for which uncontrollable parameters are not taken into account in the system performance functions, i.e., $f = f(\cdot, p)$ for $p \in \mathbb{R}^q$. The following investigations are limited to linear system performance functions of the form

$$f_j(x) - f_{\mathrm{c},j} = a_j^{\mathrm{T}} x - b_j \tag{4.68}$$

with $a_j \in \mathbb{R}^d$, $b_j \in \mathbb{R}$, $j = 1, \ldots, m$. These are relevant in systems engineering, see Chapter 5. Note that for box-shaped independent CSS, more complex system performance functions are considered in, e.g., [59]. In general, more complex considerations are lacking for other types of CSS. Nevertheless, this thesis points out when the simplifications can be directly transferred to more complex performance functions.
For reasons of simplicity, $f_j(x) = a_j^{\mathrm{T}} x$ and $f_{\mathrm{c},j} = b_j$, $j = 1, \ldots, m$ are assumed. Thus, the constraints $f(x) \leq f_{\mathrm{c}}$ build a system of linear inequalities, i.e., $Ax \leq b$, where a_j are the rows of $A \in \mathbb{R}^{m \times d}$ and b_j the entries of $b \in \mathbb{R}^m$ for $j = 1, \ldots, m$. For linear system performance functions, problem (3.16) reads

$$\begin{aligned} \underset{x^{\mathrm{l}}, x^{\mathrm{u}}}{\text{maximize}} \quad & \prod_{i=1}^{d} (x_i^{\mathrm{u}} - x_i^{\mathrm{l}}) \\ \text{subject to} \quad & x^{\mathrm{l}} - x^{\mathrm{u}} \leq 0, \\ & \forall x \in [x^{\mathrm{l}}, x^{\mathrm{u}}] : \; -x \leq -x_{\mathrm{ds}}^{\mathrm{l}}, \; x \leq x_{\mathrm{ds}}^{\mathrm{u}}, \; Ax \leq b. \end{aligned} \tag{4.69}$$

Defining $\log(0) = -\infty$, the problem can be reformulated as

$$\begin{aligned} \underset{x^{\mathrm{l}}, x^{\mathrm{u}}}{\text{minimize}} \quad & -\log\left(\prod_{i=1}^{d}(x_i^{\mathrm{u}} - x_i^{\mathrm{l}})\right) \\ \text{subject to} \quad & x^{\mathrm{l}} - x^{\mathrm{u}} \leq 0, \\ & -x^{\mathrm{l}} \leq -x_{\mathrm{ds}}^{\mathrm{l}},\ x^{\mathrm{u}} \leq x_{\mathrm{ds}}^{\mathrm{u}}, \\ & a_j^{\mathrm{T}} V_j^{\mathrm{l}} x^{\mathrm{l}} + a_j^{\mathrm{T}} V_j^{\mathrm{u}} x^{\mathrm{u}} \leq b_j,\ j = 1, \ldots, m, \end{aligned} \tag{4.70}$$

where $V_j^{\mathrm{l}}, V_j^{\mathrm{u}} \in \mathbb{R}^{d \times d}$ are diagonal matrices for which the i^{th} entries on the diagonals are given by

$$v_{j,i}^{\mathrm{l}} = \begin{cases} 1 & \text{if } a_{j,i} \leq 0, \\ 0 & \text{else}, \end{cases} \qquad v_{j,i}^{\mathrm{u}} = \begin{cases} 0 & \text{if } a_{j,i} \leq 0, \\ 1 & \text{else}, \end{cases} \tag{4.71}$$

$i = 1, \ldots, d$, $j = 1, \ldots, m$. This is shown in Theorem 9. In problem (4.70), the m optimization constraints in the condition $\forall x \in [x^{\mathrm{l}}, x^{\mathrm{u}}]$: $-x \leq -x_{\mathrm{ds}}^{\mathrm{l}}$, $x \leq x_{\mathrm{ds}}^{\mathrm{u}}$, $Ax \leq b$ of problem (4.69) are simplified, which helps to solve the optimization problem numerically. Overall, problem (4.70) has $2d$ optimization variables and $3d + m$ optimization constraints, for which each of $x^{\mathrm{l}} - x^{\mathrm{u}} \leq 0$, $-x^{\mathrm{l}} \leq -x_{\mathrm{ds}}^{\mathrm{l}}$, and $x^{\mathrm{u}} \leq x_{\mathrm{ds}}^{\mathrm{u}}$ forms d optimization constraints.

Theorem 9. *Problem (4.69) is equivalent to problem (4.70).*

Proof. First, it is shown that the optimization constraints $\forall x \in [x^{\mathrm{l}}, x^{\mathrm{u}}]$: $-x \leq -x_{\mathrm{ds}}^{\mathrm{l}}$, $x \leq x_{\mathrm{ds}}^{\mathrm{u}}$, $Ax \leq b$ from problem (4.69) and $-x^{\mathrm{l}} \leq -x_{\mathrm{ds}}^{\mathrm{l}}$, $x^{\mathrm{u}} \leq x_{\mathrm{ds}}^{\mathrm{u}}$, $a_j^{\mathrm{T}} V_j^{\mathrm{l}} x^{\mathrm{l}} + a_j^{\mathrm{T}} V_j^{\mathrm{u}} x^{\mathrm{u}} \leq b_j$, $j = 1, \ldots, m$, from problem (4.70) are equivalent. For all $x \in [x^{\mathrm{l}}, x^{\mathrm{u}}]$, it holds

$$\begin{aligned} -x_i \leq -x_{\mathrm{ds},i}^{\mathrm{l}} &\Leftrightarrow \max_{x_i \in [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]} -x_i \leq -x_{\mathrm{ds},i}^{\mathrm{l}} \\ &\Leftrightarrow\ -x_i^{\mathrm{l}} \leq -x_{\mathrm{ds},i}^{\mathrm{l}}, \end{aligned} \tag{4.72}$$

$$\begin{aligned} x_i \leq x_{\mathrm{ds},i}^{\mathrm{u}} &\Leftrightarrow \max_{x_i \in [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]} x_i \leq x_{\mathrm{ds},i}^{\mathrm{u}} \\ &\Leftrightarrow x_i^{\mathrm{u}} \leq x_{\mathrm{ds},i}^{\mathrm{u}}, \end{aligned} \tag{4.73}$$

$i = 1, \ldots, d$, and

$$\begin{aligned} a_j^{\mathrm{T}} x \leq b_j &\Leftrightarrow \max_{x \in [x^{\mathrm{l}}, x^{\mathrm{u}}]} a_j^{\mathrm{T}} x \leq b_j \\ &\Leftrightarrow a_j^{\mathrm{T}} V_j^{\mathrm{l}} x^{\mathrm{l}} + a_j^{\mathrm{T}} V_j^{\mathrm{u}} x^{\mathrm{u}} \leq b_j, \end{aligned} \tag{4.74}$$

where $V_j^{\mathrm{l}}, V_j^{\mathrm{u}} \in \mathbb{R}^{d \times d}$ are diagonal matrices with entries given by Equation (4.71), $j = 1, \ldots, m$. Therefore, the feasible sets of problems (4.69) and (4.70) are equivalent.

Furthermore, the logarithm with $\log(0) = -\infty$ is monotonically increasing on $[0, \infty)$ and any maximization problem can be reformulated as a minimization problem by changing the sign of the objective function. Thus, any optimal solution of problem (4.70) is also an optimal solution of problem (4.69) and vice versa, which means that the two problems are equivalent. □

Equivalent simplifications of the optimization constraints of problem (4.69) for linear system performance functions are, for example, considered in [6, 39, 64], too. Moreover, the above simplification can be generalized to monotonic system performance functions, as the proof of Theorem 9 only uses the monotony of linear system performance functions. A simplification for specific convex system performance functions can be found in [21]. For more general system performance functions, a simplification is more complex.
Using the objective function of problem (4.70), it can be shown that problem (4.70) is convex and differentiable. This is stated in the subsequent theorem.

Theorem 10. *The objective function of problem (4.70), defined by* $-\log(\prod_{i=1}^{d}(x_i^{\mathrm{u}} - x_i^{\mathrm{l}}))$ *for* $x^{\mathrm{l}}, x^{\mathrm{u}} \in \mathbb{R}^d$*, is convex and differentiable for* $x^{\mathrm{l}} < x^{\mathrm{u}}$*.*

Proof. Using the monotony and concavity of the logarithm, it holds

$$\begin{aligned} &-\log\left(\prod_{i=1}^{d}(\theta(x_i^{\mathrm{u}} - x_i^{\mathrm{l}})) + (1-\theta)(x_i'^{\mathrm{u}} - x_i'^{\mathrm{l}})\right) \\ &\quad\leq\ -\log\left(\theta\prod_{i=1}^{d}(x_i^{\mathrm{u}} - x_i^{\mathrm{l}}) + (1-\theta)\prod_{i=1}^{d}(x_i'^{\mathrm{u}} - x_i'^{\mathrm{l}})\right) \\ &\quad\leq \theta\left(-\log\left(\prod_{i=1}^{d}(x_i^{\mathrm{u}} - x_i^{\mathrm{l}})\right)\right) + (1-\theta)\left(-\log\left(\prod_{i=1}^{d}(x_i'^{\mathrm{u}} - x_i'^{\mathrm{l}})\right)\right) \end{aligned} \tag{4.75}$$

for all $x^{\mathrm{l}}, x^{\mathrm{u}}, x'^{\mathrm{l}}, x'^{\mathrm{u}} \in \mathbb{R}^d$, and $\theta \in (0,1)$ with $x^{\mathrm{l}} < x^{\mathrm{u}}$ and $x'^{\mathrm{l}} < x'^{\mathrm{u}}$. Therefore, the objective function of problem (4.70) is convex. By ordering the entries of $x^{\mathrm{l}}, x^{\mathrm{u}} \in \mathbb{R}^d$ as $x_1^{\mathrm{l}}, \ldots, x_d^{\mathrm{l}}, x_1^{\mathrm{u}}, \ldots, x_d^{\mathrm{u}}$, the gradient of the objective function becomes

$$\nabla\left(-\log\left(\prod_{i=1}^{d}(x_i^{\mathrm{u}} - x_i^{\mathrm{l}})\right)\right) = \left(\frac{1}{x_1^{\mathrm{u}} - x_1^{\mathrm{l}}}, \ldots, \frac{1}{x_d^{\mathrm{u}} - x_d^{\mathrm{l}}}, -\frac{1}{x_1^{\mathrm{u}} - x_1^{\mathrm{l}}}, \ldots, -\frac{1}{x_d^{\mathrm{u}} - x_d^{\mathrm{l}}}\right). \tag{4.76}$$

Hence, the objective function of problem (4.70) is differentiable for $x^{\mathrm{l}} < x^{\mathrm{u}}$. □

Moreover, the feasible set of problem (4.70) is bounded due to the design space constraints and is determined by a system of linear inequalities for x^{l} and x^{u}. Hence, problem (4.70) is a convex optimization problem for and any local minimum is a global one. It can be solved

numerically by using standard methods for differentiable convex optimization problems, see, e.g. [12]. This is further discussed in Section 4.3. Next, simplifications for arbitrarily-shaped independent CSS are considered.

4.2.2. Arbitrarily-Shaped Independent CSS

As discussed for Equation (3.4), linear system performance functions can be decomposed into a sum of linear component performance functions, i.e.,

$$a_j^{\mathrm{T}} x = \sum_{k=1}^{n} (a_j^k)^{\mathrm{T}} x^k \tag{4.77}$$

for $x \in \mathbb{R}^d$, where the corresponding entries of a_j are collected in (a_j^k), $k = 1, \ldots, n$, $j = 1, \ldots, m$. In matrix notation, Equation (4.77) is equal to

$$Ax = \sum_{k=1}^{n} A^k x^k \tag{4.78}$$

where (a_j^k), $j = 1, \ldots, m$, are the rows of $A^k \in \mathbb{R}^{m \times d^k}$, $k = 1, \ldots, n$. In the case of linear system performance functions, it can be shown that the component performance functions defined by $g^k(x^k) = A^k x^k$ for $x^k \in \mathbb{R}^k$ are the optimal component performance functions for the definitions of the CSS Ω^k, $k = 1, \ldots, n$, that solve problem (3.14), see Theorem 11. Then, these CSS have the mathematical structure

$$\Omega^k(b^k) = \{x \in \Omega_{\mathrm{ds}}^k \mid A^k x^k \leq b^k\}, \tag{4.79}$$

where $b^k \in \mathbb{R}^m$ are the component performance thresholds, which correspond to g_{c}^k, $k = 1, \ldots, n$.

Theorem 11. *For linear system performance functions, given by Equation (4.68), the optimal CSS that solve problem (3.14) have the mathematical structure* $\Omega^k(b^k) = \{x \in \Omega_{\mathrm{ds}}^k \mid A^k x^k \leq b^k\}$, $k = 1, \ldots, n$, *as stated in Equation (4.79).*

Proof. Let $\Omega^k(b^k)$ be CSS of the mathematical structure given by Equation (4.79) that depend on b^k, $k = 1, \ldots, n$. Suppose there exist component solution spaces $\Omega'^k = \{x \in \Omega_{\mathrm{ds}}^k \mid g^k(x^k) \leq g_{\mathrm{c}}^k\}$, $k = 1, \ldots, n$, with $\Omega'^1 \times \cdots \times \Omega'^n \subseteq \Omega_{\mathrm{c}}$ and $\Omega'^k \neq \Omega^k(b^k)$ for all $b^k \in \mathbb{R}^m$ and at least one $k \in \{1, \ldots, n\}$. Furthermore, suppose it holds $\mathrm{vol}(\Omega'^1 \times \cdots \times \Omega'^n) \geq \mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n))$ for all b^k, $k = 1, \ldots, n$, with $\sum_{k=1}^n b^k \leq b$.
Because of $\Omega'^1 \times \cdots \times \Omega'^n \subseteq \Omega_{\mathrm{c}}$, it is

$$b_j \geq \sup_{x \in \Omega'^1 \times \cdots \times \Omega'^n} a_j^{\mathrm{T}} x = \sum_{k=1}^{n} \sup_{x^k \in \Omega'^k} (a_j^k)^{\mathrm{T}} x^k, \tag{4.80}$$

$j = 1, \ldots, m$. Now, define $b'^k_j = \sup_{x^k \in \Omega'^k} (a^k_j)^{\mathrm{T}} x^k$ for $j = 1, \ldots, m$, $k = 1, \ldots, n$. Thus, it holds that $\Omega'^k \subseteq \Omega^k(b'^k)$ and $\sum_{k=1}^n b'^k \leq b$. This is a contradiction to the assumption that $\mathrm{vol}(\Omega'^1 \times \cdots \times \Omega'^n) \geq \mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n))$ for b^k, $k = 1, \ldots, n$, with $\sum_{k=1}^n b^k \leq b$. Hence, the optimal component performance functions g^k, $k = 1, \ldots, n$, must be linear and the optimal CSS have the mathematical structure that is given in Equation (4.79). □

Theorem 11, can be generalized to all system performance functions that can be decomposed into a sum of component performance functions as described in Section 3.2, compare [21]. Overall, problem (3.14) simplifies to problem (3.18). Incorporating again the negative logarithm of the objective function, the problem reads

$$\begin{aligned} \underset{b^1,\ldots,b^n}{\text{minimize}} \quad & -\log(\mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n))) \\ \text{subject to} \quad & \sum_{k=1}^n b^k \leq b. \end{aligned} \tag{4.81}$$

for linear performance functions. The objective function of problem (4.81) is convex, see the subsequent theorem. Problem (4.81) has mn optimization variables and m optimization constraints stemming from $b^k \in \mathbb{R}^m$, $k = 1, \ldots, n$. Note that the number of optimization variables might be reduced depending on the detailed structure of the linear system performance functions.

Theorem 12. *The objective function of problem (4.81) given by* $-\log(\mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n)))$ *for* $b^k \in \mathbb{R}^{d^k}$, $k = 1, \ldots, n$, *is convex.*

Proof. For $b^k \in \mathbb{R}^{d^k}$, $k = 1, \ldots, n$, it holds

$$\mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n)) = \prod_{k=1}^n \mathrm{vol}(\Omega^k(b^k)). \tag{4.82}$$

Thus, it is sufficient to show that $\mathrm{vol}(\Omega^k(b^k))$, $k = 1, \ldots, n$ is log-concave, because the product of log-concave functions is log-concave, see Section A.1. Then, $\log(\mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n)))$ is concave and $-\log(\mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n)))$ is convex for $b^k \in \mathbb{R}^{d^k}$, $k = 1, \ldots, n$.

Let $k \in \{1, \ldots, n\}$ be arbitrary. The condition $x^k \in \Omega^k$ can be reformulated as a system of linear inequalities, cf. Theorem 2, i.e.,

$$\Omega^k(b^k) = \{x^k \in \mathbb{R}^{d^k} \mid A^k x^k \leq b^k\}. \tag{4.83}$$

Defining

$$\chi^k : \mathbb{R}^{d^k} \times \mathbb{R}^m \to \{0,1\},\ (x^k, b^k) \mapsto \chi^k(x^k, b^k) = \begin{cases} 1 & \text{if } A^k x^k \leq b^k, \\ 0 & \text{else,} \end{cases} \tag{4.84}$$

it is

$$\mathrm{vol}(\Omega^k(b^k)) = \int\limits_{x^k \in \mathbb{R}^{d^k}} \chi^k(x^k, b^k)\, \mathrm{d}x^k. \tag{4.85}$$

Now, it will be shown that the function χ^k is log-concave, i.e.,

$$\begin{aligned} \chi^k(\theta x^k + (1-\theta)x'^k, \theta b^k + (1-\theta)b'^k) &\geq \chi^k(x^k b^k)^\theta \chi^k(x'^k b'^k)^{1-\theta} \\ &= \chi^k(x^k b^k)\chi^k(x'^k b'^k) \end{aligned} \tag{4.86}$$

for all $(x^k, b^k), (x'^k, b'^k) \in \mathbb{R}^{d^k} \times \mathbb{R}^m$ and $\theta \in (0,1)$. Inequality (4.86) is always fulfilled for either $\chi^k(x^k b^k) = 0$ or $\chi^k(x'^k b'^k) = 0$. Hence, let $\chi^k(x^k, b^k) = 1$ and $\chi^k(x'^k, b'^k) = 1$, i.e., $A^k x^k \leq b^k$ and $A^k x'^k \leq b'^k$. Then, it holds

$$\begin{aligned} A^k(\theta x^k + (1-\theta)x'^k) &= \theta A^k x^k + (1-\theta)A^k x'^k \\ &= \theta b^k + (1-\theta)b'^k, \end{aligned} \tag{4.87}$$

i.e., $\chi^k(\theta x^k + (1-\theta)x'^k, \theta b^k + (1-\theta)b'^k) = 1$ and inequality (4.86) is fulfilled for all $(x^k, b^k), (x'^k, b'^k) \in \mathbb{R}^{d^k} \times \mathbb{R}^m$ and $\theta \in (0,1)$. As furthermore the integration of the log-concave function χ^k remains log-concave, compare [109, 110], $\mathrm{vol}(\Omega^k(b^k))$ is log-concave. This holds for all $k = 1, \ldots, n$ because k was selected arbitrarily. □

The feasible set of problem (4.81) is defined by linear inequalities. There is a part of the feasible set for which $-\log(\mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n))) > -\infty$ holds for $b^k \in \mathbb{R}^{d^k}$. It is bounded by

$$\min_{x^k \in \Omega^k_{\mathrm{ds}}} A^k x^k \leq b^k \leq b - \sum_{\substack{k'=1 \\ k' \neq k}}^{n} \min_{x^{k'} \in \Omega^{k'}_{\mathrm{ds}}} A^{k'} x^{k'}, \tag{4.88}$$

$k = 1, \ldots, n$. As the objective function of problem (4.81) is convex, problem (4.81) is a convex optimization problem for which again every local optimum is a global optimum.
Furthermore, from the convexity of the objective function, its continuity follows. However, it is not necessarily differentiable as the following example shows.

Example 26. Given a system composed of two components. The component design of component 1 contains two design variables and the component design of component 2 contains one design variable. The system design space is given by $\Omega_{\mathrm{ds}} = [0,1]^3$ and the system performance functions by $f : \mathbb{R}^3 \to \mathbb{R}^2,\ x \mapsto (x_1 + x_2 + x_3, x_1 + x_2)$ with performance

thresholds in $f_c = (2, 1)$.
According to Theorem 11, the optimal component performance functions are given by $g^1 : \mathbb{R}^2 \to \mathbb{R}^2,\ (x_1, x_2) \mapsto (x_1 + x_2, x_1 + x_2)$ and $g^2 : \mathbb{R} \to \mathbb{R}^2,\ x_3 \mapsto (x_3, 0)$. Thus, the optimal CSS have the mathematical structure $\Omega^1(b^1) = \{(x_1, x_2) \in [0, 1]^2 \mid (x_1 + x_2, x_1 + x_2) \leq b^1\}$ and $\Omega^2(b^2) = \{(x_1, x_2) \in [0, 1]^2 \mid (x_3, 0) \leq b^2\}$, and depend on $b^1, b^2 \in \mathbb{R}^2$. For $b^1 = (1.5, 1) - t(1, 0)$, $b^2 = (0.5, 0) + t(1, 0)$ and $t \in [0, 1]$ where $b^1 + b^2 = (2, 1)$, it holds

$$
\begin{aligned}
&-\log(\mathrm{vol}(\Omega^1(b^1(t)) \times \Omega^2(b^2(t)))) \\
&\quad = \begin{cases} -\log(0.5(0.5 + t)) & \text{if } 0 \leq t \leq 0.5, \\ -\log(0.5(1.5 - t)^2(0.5 + t)) & \text{else.} \end{cases}
\end{aligned} \tag{4.89}
$$

The corresponding graph is visualized in Figure 41. For $t = 0.5$ the derivative

$$
\frac{\mathrm{d}(-\log(\mathrm{vol}(\Omega^1(b^1(t)) \times \Omega^2(b^2(t)))))}{\mathrm{d}t} = \begin{cases} \frac{-1}{0.5+t} & \text{if } 0 \leq t \leq 0.5, \\ \frac{-3t^2+5t-0.75}{(1.5-t)^2(0.5+t)} & \text{else.} \end{cases} \tag{4.90}
$$

is not continuous at $t = 0.5$, which shows that the objective function of problem (4.81) is not differentiable in general.

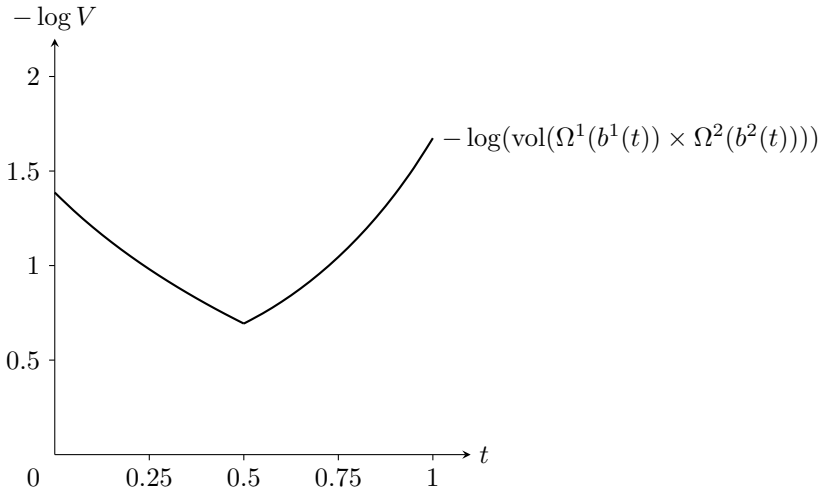


Figure 41 Graph of the function defined by $-\log(\mathrm{vol}(\Omega^1(b^1(t)) \times \Omega^2(b^2(t))))$ over the domain $[0, 1]$ for Example 26.

Hence, problem (4.81) can be solved numerically using techniques for non-differentiable convex optimization problems, compare, e.g., [120]. If the two cases that a constraint appears more than once, compare Example 26, and that there are redundant constraints can be excluded, then $\Omega^k(b^k)$ can be shown to be differentiable at b^k, $k \in \{1, \ldots, n\}$, see [87]. Hence, also $-\log(\mathrm{vol}(\Omega^1(b^1) \times \cdots \times \Omega^n(b^n)))$ becomes differentiable and problem (4.81) can be solved numerically using techniques for differentiable convex optimization problems, see [12]. Furthermore, the objective function of problem (4.81) is monotonically decreasing in b^k,

$k = 1, \ldots, n$, i.e., it holds

$$\begin{aligned} & b_j'^k \leq b_j''^k,\ j = 1, \ldots, m,\ k = 1, \ldots, n, \\ \Rightarrow\ & -\log(\mathrm{vol}(\Omega^1(b'^1) \times \cdots \times \Omega^n(b'^n))) \geq -\log(\mathrm{vol}(\Omega^1(b''^1) \times \cdots \times \Omega^n(b''^n))). \end{aligned} \tag{4.91}$$

Thus, an optimal solution of problem (4.81) can always be found at $\sum_{k=1}^{n} b^k = b$ and problem (4.81) can be reformulated as an unconstrained optimization problem, which reads, for example,

$$\underset{b^1,\ldots,b^{n-1}}{\text{minimize}}\ -\log\left(\mathrm{vol}\left(\Omega^1(b^1) \times \cdots \times \Omega^{n-1}(b^{n-1}) \times \Omega^n\left(b - \sum_{k=1}^{n-1} b^k\right)\right)\right). \tag{4.92}$$

Problem (4.92) can be solved using methods for unconstrained optimization problems, see, e.g., [100]. This is further discussed in Section 4.3.

For linear performance functions and $d^k = 1$, box-shaped and arbitrarily-shaped independent CSS coincide. Here, the CSS Ω^k with $\mathrm{vol}(\Omega^k) > 0$ are intervals due to their convexity, i.e.,

$$\Omega^k = [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}] \tag{4.93}$$

with $i = k$, $k = 1, \ldots, n$, for which the constraints (4.72)-(4.74) hold. Next, simplifications for dependent CSS are considered.

4.2.3. Dependent CSS

The problem of computing box-shaped dependent CSS, i.e., problem (3.21), can be tackled in different ways. One approach is to first compute the projection $\mathrm{proj}^k(\Omega_{\mathrm{c}})$, which is a convex polytope in the case of linear system performance functions that is described by linear inequalities for all $k \in \{1, \ldots, n\}$, compare Section A.1. Then, the resulting problem is similar to problem (4.69) and can be solved by its reformulation, i.e., problem (4.69). This approach is, for example, taken in [128].

Another approach is to reformulate problem (3.21) such that it can be solved directly. Using the properties of projections, problem (3.21) reads

$$\begin{aligned} \underset{x^{\mathrm{l},k}, x^{\mathrm{u},k}}{\text{maximize}}\quad & \prod_{i=1}^{d^k} (x_i^{\mathrm{u},k} - x_i^{\mathrm{l},k}) \\ \text{subject to}\quad & x^{\mathrm{l},k} - x^{\mathrm{u},k} \leq 0, \\ & \forall x^k \in [x^{\mathrm{l},k}, x^{\mathrm{u},k}]\ \exists x^{k'},\ k' = 1, \ldots, n,\ k' \neq k: \\ & \qquad -x \leq -x_{\mathrm{ds}}^{\mathrm{l}},\ x \leq x_{\mathrm{ds}}^{\mathrm{u}},\ Ax \leq b \end{aligned} \tag{4.94}$$

for linear performance functions where x is formed by x^k and $x^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$ for $k = 1, \ldots, n$. For every $k \in \{1, \ldots, n\}$, a solution of problem (4.94) can, for example, be achieved by finding remaining component designs $x^{k'} \in \Omega_{\mathrm{ds}}^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$, for

all corner points of $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$ such that all resulting system designs are permissible. Here, each resulting system designs is denoted by $x_{\mathrm{cp},\iota}$ and depends on the vectors $x^{\mathrm{l},k}$ and $x^{\mathrm{u},k}$, which form the ι^{th} corner point of $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$, and the remaining component designs $x^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$, collected in $x^{\mathrm{r},k}_{\mathrm{cp},\iota}$, $\iota = 1, \ldots, 2^{d^k}$. These dependencies can be expressed as $x_{\mathrm{cp},\iota}(x^{\mathrm{l},k}, x^{\mathrm{u},k}, x^{\mathrm{r},k}_{\mathrm{cp},\iota})$, $\iota = 1, \ldots, 2^{d^k}$. Then, problem (4.94) can be simplified mathematically as

$$
\begin{aligned}
\underset{x^{\mathrm{l},k}, x^{\mathrm{u},k}, x^{\mathrm{r},k}_{\mathrm{cp},\iota}, \iota = 1, \ldots, 2^{d^k}}{\text{minimize}} \quad & -\log\left(\prod_{i=1}^{d^k} (x_i^{\mathrm{u},k} - x_i^{\mathrm{l},k})\right), \\
\text{subject to} \quad & x^{\mathrm{l},k} - x^{\mathrm{u},k} \leq 0, \\
& -x^{\mathrm{l},k} \leq -x^{\mathrm{l},k}_{\mathrm{ds}},\ x^{\mathrm{u},k} \leq x^{\mathrm{u},k}_{\mathrm{ds}}, \\
& -x^{\mathrm{r},k}_{\mathrm{cp},\iota} \leq -x^{\mathrm{l,r},k}_{\mathrm{ds}},\ x^{\mathrm{r},k}_{\mathrm{cp},\iota} \leq x^{\mathrm{u,r},k}_{\mathrm{ds}},\ \iota = 1, \ldots, 2^{d^k} \\
& a_j^{\mathrm{T}} x_{\mathrm{cp},\iota}(x^{\mathrm{l},k}, x^{\mathrm{u},k}, x^{\mathrm{r},k}_{\mathrm{cp},\iota}) \leq b_j,\ j = 1, \ldots, m,\ \iota = 1, \ldots, 2^{d^k},
\end{aligned}
\tag{4.95}
$$

where the lower component design space bounds of the remaining component vectors are collected in $x^{\mathrm{l,r},k}_{\mathrm{ds}} \in \mathbb{R}^{d-d^k}$ and the upper design space bounds in $x^{\mathrm{l,r},k}_{\mathrm{ds}} \in \mathbb{R}^{d-d^k}$, $k = 1, \ldots, n$. This is shown in Theorem 13. For every $k \in \{1, \ldots, n\}$, problem (4.95) has $2d^k + 2^{d^k}(d - d^k)$ optimization variables and $3d^k + 2^{d^k}(2(d - d^k) + m)$ optimization constraints. Here, each of $x^{\mathrm{l},k} - x^{\mathrm{u},k} \leq 0$, $-x^{\mathrm{l},k} \leq -x^{\mathrm{l},k}_{\mathrm{ds}}$, and $x^{\mathrm{u},k} \leq x^{\mathrm{u},k}_{\mathrm{ds}}$ forms d^k, each of $-x^{\mathrm{r},k}_{\mathrm{cp},\iota} \leq -x^{\mathrm{l,r},k}_{\mathrm{ds}}$, $\iota = 1, \ldots, 2^{d^k}$, and $x^{\mathrm{r},k}_{\mathrm{cp},\iota} \leq x^{\mathrm{u,r},k}_{\mathrm{ds}}$, $\iota = 1, \ldots, 2^{d^k}$, form $2^{d^k}(d - d^k)$, and $a_j^{\mathrm{T}} x_{\mathrm{cp},\iota}(x^{\mathrm{l},k}, x^{\mathrm{u},k}, x^{\mathrm{r},k}_{\mathrm{cp},\iota}) \leq b_j$, $j = 1, \ldots, m$, $\iota = 1, \ldots, 2^{d^k}$, form $m2^{d^k}$ optimization constraints. As d^k is an exponent for both the number of optimization variables and constraints, problem (4.95) is in general only suitable for small d^k, $k \in \{1, \ldots, n\}$. However, the number of optimization variables and constraints might be reduced depending on the detailed structure of linear system performance functions.

Theorem 13. *Problem (4.95) is equivalent to problem (4.94).*

Proof. The optimization constraints $-x^{\mathrm{l},k} \leq -x^{\mathrm{l},k}_{\mathrm{ds}}$, $x^{\mathrm{u},k} \leq x^{\mathrm{u},k}_{\mathrm{ds}}$, $-x^{\mathrm{r},k}_{\mathrm{cp},\iota} \leq -x^{\mathrm{l,r},k}_{\mathrm{ds}}$, $x^{\mathrm{r},k}_{\mathrm{cp},\iota} \leq x^{\mathrm{u,r},k}_{\mathrm{ds}}$, $a_j^{\mathrm{T}} x_{\mathrm{cp},\iota}(x^{\mathrm{l},k}, x^{\mathrm{u},k}, x^{\mathrm{r},k}_{\mathrm{cp},\iota}) \leq b_j$, $j = 1, \ldots, m, \iota = 1, \ldots, 2^{d^k}$ of problem (4.95) correspond to the optimization constraints of problem (4.94) for all corner points of $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$ where $x^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$, must exist with $-x \leq -x^{\mathrm{l}}_{\mathrm{ds}}$, $x \leq x^{\mathrm{u}}_{\mathrm{ds}}$, and $Ax \leq b$. Hence, it must be shown that the optimization constraints of (4.94) are also fulfilled for all non-corner points if the corner points fulfill them.

Let $x^k \in [x^{\mathrm{l},k}, x^{\mathrm{u},k}]$ be a non-corner point for $k \in \{1, \ldots, n\}$. With the theorem of Carathéodory,

cf. [52], there exist $d^k + 1$ corner points of $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$, i.e., $x^k_{\mathrm{cp},\iota'}$, $\iota' = 1, \ldots, d^k + 1$, with

$$x^k = \sum_{\iota'=1}^{d^k+1} \lambda^{\iota'} x^k_{\mathrm{cp},\iota'} \tag{4.96}$$

where $\lambda^{\iota'} \in (0, 1)$, $\iota' = 1, \ldots, d^k + 1$, and $\sum_{\iota'=1}^{d^k+1} \lambda^{\iota'} = 1$. With $\lambda^{\iota'}$ of Equation (4.96), the vectors $x^{\mathrm{r},k}_{\mathrm{cp},\iota'}$, which contain the remaining component designs $x^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$ for $\iota' = 1, \ldots, d^k + 1$, can be combined to

$$x^{\mathrm{r},k} = \sum_{\iota'=1}^{d^k+1} \lambda^{\iota'} x^{\mathrm{r},k}_{\mathrm{cp},\iota'}. \tag{4.97}$$

Then, the vectors x^k and $x^{\mathrm{r},k}$ define a system design x given by

$$x = \sum_{\iota'=1}^{d^k+1} \lambda^{\iota'} x_{\mathrm{cp},\iota'}(x^{\mathrm{l},k}, x^{\mathrm{u},k}, x^{\mathrm{r},k}_{\mathrm{cp},\iota'}) \tag{4.98}$$

where $x \in \Omega_{\mathrm{c}}$. This holds due to the convexity of the complete system solution space Ω_{c}. Hence for every $x^k \in [x^{\mathrm{l},k}, x^{\mathrm{u},k}]$ that is not a corner point, there exist $x^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$, given by $x^{\mathrm{r},k}$ in Equation (4.97), with $-x \leq -x^{\mathrm{l}}_{\mathrm{ds}}$, $x \leq x^{\mathrm{u}}_{\mathrm{ds}}$, and $Ax \leq b$.
Furthermore, the objective functions of problems (4.94) and (4.95) are consistent, which can be shown similarly to the proof of Theorem 9. Therefore, the two problems are equivalent. □

The above simplification can be generalized to convex system performance functions, as the proof of Theorem 13 only uses the convex property of linear system performance functions. The objective function of problem (4.95) is convex and differentiable, compare Theorem 10, and its feasible set is bounded. Thus, problem (4.70) is also a convex optimization problem and can be solved numerically by using standard methods for differentiable convex optimization problems, which is further discussed in the next section.

For $d^k = 1$, $k \in \{1, \ldots, n\}$, the volume of $[x^{\mathrm{l},k}, x^{\mathrm{u},k}]$ can be maximized by minimizing $x^{\mathrm{l},k}$ and maximizing $x^{\mathrm{u},k}$. In this case, problem (4.95) can be reformulated as two independent optimization problems

$$\begin{aligned} \underset{x^{\mathrm{l},k}, x^{\mathrm{r},k}}{\text{minimize}} \quad & x^{\mathrm{l},k} \\ \text{subject to} \quad & -x(x^{\mathrm{l},k}, x^{\mathrm{r},k}) \leq -x^{\mathrm{l}}_{\mathrm{ds}},\ x(x^{\mathrm{l},k}, x^{\mathrm{r},k}) \leq x^{\mathrm{u}}_{\mathrm{ds}}, \\ & a_j^{\mathrm{T}} x(x^{\mathrm{l},k}, x^{\mathrm{r},k}) \leq b_j,\ j = 1, \ldots, m, \end{aligned} \tag{4.99}$$

where $x(x^{\mathrm{l},k}, x^{\mathrm{r},k}) \in \mathbb{R}^d$ is the resulting system design of $x^{\mathrm{l},k} \in \mathbb{R}^{d^k}$ and $x^{k'} \in \mathbb{R}^{d^{k'}}$, $k' =$

$1, \ldots, n$, $k' \neq k$, collected in $x^{\mathrm{r},k} \in \mathbb{R}^{d-d^k}$, and

$$\begin{aligned} \underset{x^{\mathrm{u},k}, x^{\mathrm{r},k}}{\text{minimize}} \quad & -x^{\mathrm{u},k} \\ \text{subject to} \quad & -x(x^{\mathrm{u},k}, x^{\mathrm{r},k}) \leq -x^{\mathrm{l}}_{\mathrm{ds}},\ x(x^{\mathrm{u},k}, x^{\mathrm{r},k}) \leq x^{\mathrm{u}}_{\mathrm{ds}}, \\ & a_j^{\mathrm{T}} x(x^{\mathrm{u},k}, x^{\mathrm{r},k}) \leq b_j,\ j = 1, \ldots, m, \end{aligned} \tag{4.100}$$

where $x(x^{\mathrm{u},k}, x^{\mathrm{r},k}) \in \mathbb{R}^d$ is the resulting system design of $x^{\mathrm{u},k} \in \mathbb{R}^{d^k}$ and $x^{k'} \in \mathbb{R}^{d^{k'}}$, $k' = 1, \ldots, n$, $k' \neq k$, collected in $x^{\mathrm{r},k} \in \mathbb{R}^{d-d^k}$ again. Equivalent simplification for the $d^k = 1$, $k \in \{1, \ldots, n\}$, can be found in [6]. Problems (4.99) and (4.100) have d^k optimization variables and $2d^k + m$ optimization constraints each. Representatively for problem (4.99), each of $-x(x^{\mathrm{l},k}, x^{\mathrm{r},k}) \leq -x^{\mathrm{l}}_{\mathrm{ds}}$ and $x(x^{\mathrm{l},k}, x^{\mathrm{r},k}) \leq x^{\mathrm{u}}_{\mathrm{ds}}$ forms d^k and $a_j^{\mathrm{T}} x(x^{\mathrm{l},k}, x^{\mathrm{r},k}) \leq b_j$, $j = 1, \ldots, m$, form m optimization constraints. Both problems (4.99) and (4.100) are linear optimization problems that can be solved by using standard methods for linear optimization, cf., [100].

Furthermore, these problems can also be used to compute arbitrarily-shaped dependent CSS for $d^k = 1$, $k \in \{1, \ldots, n\}$, as the CSS are intervals in this case. In general, no optimization problem needs to be solved to compute arbitrarily-shaped dependent CSS because they are the projection of the complete system solution space onto the corresponding d^k-dimensional coordinate space. This is discussed in Section 4.3 for linear system performance functions. In the following, computing CSS under uncertainty is considered.

4.2.4. CSS under Uncertainty

First, problem statements under interval-type uncertainty with uncertainties in controllable variables are investigated in which uncertainties in uncontrollable parameters are neglected. In the case of linear system performance functions, problem (3.31), which accounts for uncertainties in controllable variables with unknown magnitudes, can be reformulated as

$$\begin{aligned} \underset{\check{x}, \delta'}{\text{maximize}} \quad & \delta' \\ \text{subject to} \quad & -\delta' \leq 0, \\ & -\check{x} + (\delta'\omega + \nu) \leq -x^{\mathrm{l}}_{\mathrm{ds}},\ \check{x} + (\delta'\omega + \nu) \leq x^{\mathrm{u}}_{\mathrm{ds}}, \\ & a_j^{\mathrm{T}} \check{x} + a_j^{\mathrm{T}} W_j (\delta'\omega + \nu) \leq b_j,\ j = 1, \ldots, m, \end{aligned} \tag{4.101}$$

where $W_j \in \mathbb{R}^{d \times d}$ are diagonal matrices for that the i^{th} entries on the diagonals are given by

$$w_{j,i} = \begin{cases} -1 & \text{if } a_{j,i} \leq 0, \\ 1 & \text{else}, \end{cases} \tag{4.102}$$

$i = 1, \ldots, d$, $j = 1, \ldots, m$. This is shown in Theorem 14. Overall, problem (4.101) has $d + 1$ optimization variables and $1 + 2d + m$ optimization constraints. Here, $-\delta' \leq 0$ forms one, each of $-\check{x} + (\delta'\omega + \nu) \leq -x^{\mathrm{l}}_{\mathrm{ds}}$ and $\check{x} + (\delta'\omega + \nu) \leq x^{\mathrm{u}}_{\mathrm{ds}}$ forms d, and $a_j^{\mathrm{T}}\check{x} + a_j^{\mathrm{T}} W_j(\delta'\omega + \nu) \leq b_j$, $j = 1, \ldots, m$, form m optimization constraints.

Theorem 14. *Problem (3.31) is equivalent to problem (4.101) for linear system performance functions.*

Proof. For linear system performance functions, the optimization constraint $\mathcal{U}^X(\check{x}, \delta'\omega + \nu) \subseteq \Omega_{\mathrm{c}}$ of problem (3.31), in which $\mathcal{U}^X(\check{x}, \delta'\omega + \nu)$ is given by equation (3.32), reads

$$\forall x \in [\check{x} - (\delta'\omega + \nu), \check{x} + (\delta'\omega + \nu)] : \; -x \leq -x^{\mathrm{l}}_{\mathrm{ds}}, \; x \leq x^{\mathrm{u}}_{\mathrm{ds}}, \; Ax \leq b. \tag{4.103}$$

Similar to the proof of Theorem 9, it can be shown that the optimization constraints given by Equation (4.103) are equivalent to the $-\check{x} + (\delta'\omega + \nu) \leq -x^{\mathrm{l}}_{\mathrm{ds}}$, $\check{x} + (\delta'\omega + \nu) \leq x^{\mathrm{u}}_{\mathrm{ds}}$, $a_j^{\mathrm{T}}\check{x} + a_j^{\mathrm{T}} W_j(\delta'\omega + \nu) \leq b_j$, $j = 1, \ldots, m$, of problem (3.31). For all $x \in [\check{x} - (\delta'\omega + \nu), \check{x} + (\delta'\omega + \nu)]$, it holds

$$\begin{aligned} -x_i \leq -x^{\mathrm{l}}_{\mathrm{ds},i} \Leftrightarrow & \max_{x_i \in [\check{x}_i - (\delta'\omega_i + \nu_i), \check{x} + (\delta'\omega_i + \nu_i)]} -x_i \leq -x^{\mathrm{l}}_{\mathrm{ds},i} \\ \Leftrightarrow & \; -\check{x}_i + (\delta'\omega_i + \nu_i) \leq -x^{\mathrm{l}}_{\mathrm{ds},i}, \end{aligned} \tag{4.104}$$

$$\begin{aligned} x_i \leq x^{\mathrm{u}}_{\mathrm{ds},i} \Leftrightarrow & \max_{x_i \in [\check{x}_i - (\delta'\omega_i + \nu_i), \check{x} + (\delta'\omega_i + \nu_i)]} x_i \leq x^{\mathrm{u}}_{\mathrm{ds},i} \\ \Leftrightarrow & \; \check{x}_i + (\delta'\omega_i + \nu_i) \leq x^{\mathrm{u}}_{\mathrm{ds},i}, \end{aligned} \tag{4.105}$$

$i = 1, \ldots, d$, and

$$\begin{aligned} a_j^{\mathrm{T}} x \leq b_j \Leftrightarrow & \max_{x \in [\check{x}_i - (\delta'\omega_i + \nu_i), \check{x} + (\delta'\omega_i + \nu_i)]} a_j^{\mathrm{T}} x \leq b_j \\ \Leftrightarrow & \; a_j^{\mathrm{T}}\check{x} + a_j^{\mathrm{T}} W_j(\delta'\omega + \nu) \leq b_j, \end{aligned} \tag{4.106}$$

where W_j are diagonal matrices with entries given by Equation (4.102), $j = 1, \ldots, m$. Therefore, the feasible sets of problems (4.69) and (4.70) are equivalent. As furthermore $\delta' \geq 0 \;\Leftrightarrow\; -\delta' \leq 0$ and any maximization problem can be reformulated as a minimization problem by changing the sign of the objective function., problems (3.31) and (4.101) are equivalent for linear system performance functions. □

Equivalent simplifications of problem (3.31) for linear system performance functions are, for example, considered in [66]. Again, the above simplification can be generalized to monotonic system performance functions, as the proof of Theorem 14 only uses the monotony of linear system performance functions. For more complex system performance functions, a simplification is more complex. Problem (4.101) is a linear optimization problem, which can be solved

numerically by using standard methods for linear optimization problems as both the objective function and the constraint functions are linear.

After solving problem (4.101), $\bar{\delta}$ can be computed by Equation (3.30) and the situation corresponds to the situation of uncertainty in controllable variables with known magnitudes, as pointed out in Section 3.4. For known $\bar{\delta}$ and and linear system performance functions, the worst-case solution space of target designs is equivalent to

$$\check{\Omega}_{\mathrm{c,wc}} = \{\check{x} \in [x_{\mathrm{ds}}^{\mathrm{l}} + \bar{\delta}, x_{\mathrm{ds}}^{\mathrm{u}} - \bar{\delta}] \mid a_j^{\mathrm{T}} \check{x} \leq b_j - a_j^{\mathrm{T}} W_j \bar{\delta},\ j = 1, \ldots, m\}, \tag{4.107}$$

where W_j, $j = 1, \ldots, m$, are given by Equation (4.102). This can be shown similarly to the proof of Theorem 14. Then, box- or arbitrarily-shaped worst-case CSS $\check{\Omega}_{\mathrm{wc}}^k$, $k = 1, \ldots, n$, can be computed using the simplifications from above because $\check{\Omega}_{\mathrm{c,wc}}$, given by Equation (4.107), has the same structure as Ω_{c}, given by Equation (2.4). Note that this does not change if a perturbation parameter $\varepsilon \geq 0$ is included in the objective function, see replacements (3.33) and (3.34). In this subsection, the vectors of lower bounds for box-shaped worst-case CSS are denoted by $\check{x}_{\mathrm{wc}}^{\mathrm{l},k}$, the vectors of their upper bounds by $\check{x}_{\mathrm{wc}}^{\mathrm{u},k}$, and the vectors of component performance thresholds for arbitrarily-shaped worst-case CSS by b_{wc}^k, $k = 1, \ldots, n$.
Box-shaped CSS Ω^k can be calculated from box-shaped worst-case CSS $\check{\Omega}_{\mathrm{wc}}^k$ via

$$\Omega^k = [\check{x}_{\mathrm{wc}}^{\mathrm{l},k} - \bar{\delta}^k, \check{x}_{\mathrm{wc}}^{\mathrm{u},k} + \bar{\delta}^k], \tag{4.108}$$

$k = 1, \ldots, n$. Arbitrarily-shaped CSS Ω^k can be calculated from arbitrarily-shaped worst-case CSS $\check{\Omega}_{\mathrm{wc}}^k$ via

$$\Omega^k = \{x^k \in \Omega_{\mathrm{ds}}^k \mid (a_j^k)^{\mathrm{T}} x^k \leq b_{\mathrm{wc},j}^k + (a_j^k)^{\mathrm{T}} W_j^k \bar{\delta}^k,\ j = 1, \ldots, m\}, \tag{4.109}$$

where $W_j^k \in \mathbb{R}^{d^k \times d^k}$ are diagonal matrices for that the i^{th} entries on the diagonals are given by

$$w_{j,i}^k = \begin{cases} -1 & \text{if } a_{j,i}^k \leq 0, \\ 1 & \text{else,} \end{cases} \tag{4.110}$$

$i = 1, \ldots, d^k$, $k = 1, \ldots, n$. Note that in doing so, the constraints for independent CSS, expressed in $\Omega^1 \times \cdots \times \Omega^n \subset \Omega_{\mathrm{c}}$, are always fulfilled and dependent CSS are aligned with the worst-case, as described in Section 3.4. Furthermore, an additional optimization to maximize flexibility for component design in the case of arbitrarily-shaped CSS, like also described in Section 3.4, is redundant if arbitrarily-shaped CSS are computed by Equation (4.109).
Similarly, best-case CSS $\check{\Omega}_{\mathrm{bc}}^k$ can be computed from CSS Ω^k, $k = 1, \ldots, n$. Note that for arbitrarily-shaped CSS however, best-case CSS are sometimes only subsets of the resulting sets, i.e.,

$$\check{\Omega}_{\mathrm{bc}}^k \subseteq \{\check{x}^k \in [x_{\mathrm{ds}}^{\mathrm{l},k} - \bar{\delta}^k, x_{\mathrm{ds}}^{\mathrm{u},k} + \bar{\delta}^k] \mid (a_j^k)^{\mathrm{T}} \check{x}^k \leq b_{\mathrm{wc},j}^k + 2(a_j^k)^{\mathrm{T}} \bar{\delta}^k,\ j = 1, \ldots, m\}, \tag{4.111}$$

$k = 1, \ldots, n$, compare Example 27.

Instead of deducing best-case CSS from worst-case CSS, also worst-case CSS can be deduced from best-case CSS in general, compare Section 3.4. However, the same problem as stated in Equation (4.111) occurs if the best-case complete system solution space of target designs $\check{\Omega}_{\mathrm{c,bc}}$ is calculated from the complete system solution space Ω_{c}. It is

$$\check{\Omega}_{\mathrm{c,bc}} \subseteq \{\check{x} \in [x^{\mathrm{l}}_{\mathrm{ds}} - \bar{\delta}, x^{\mathrm{u}}_{\mathrm{ds}} + \bar{\delta}] \mid a_j^{\mathrm{T}} \check{x} \leq b_j + a_j^{\mathrm{T}} W_j \bar{\delta},\ j = 1, \ldots, m\}, \tag{4.112}$$

see Example 27, and the superset is denoted by $\check{\Omega}_{\mathrm{s,bc}}$. Nevertheless, best-case CSS might be computed using the simplifications from above in which $\check{\Omega}_{\mathrm{c,bc}}$ is replaced by the set $\check{\Omega}_{\mathrm{s,bc}}$. Then, $\check{\Omega}^1_{\mathrm{bc}} \times \cdots \times \check{\Omega}^n_{\mathrm{bc}} \subseteq \check{\Omega}_{\mathrm{s,bc}}$ must be fulfilled from which $\check{\Omega}^1_{\mathrm{bc}} \times \cdots \times \check{\Omega}^n_{\mathrm{bc}} \subseteq \check{\Omega}_{\mathrm{c,bc}}$ might follow. However, in doing so, it must be tested whether the designs within $\check{\Omega}^1_{\mathrm{bc}} \times \cdots \times \check{\Omega}^n_{\mathrm{bc}}$ are within $\check{\Omega}_{\mathrm{c,bc}}$ as well. If this is the case, CSS Ω^k, and worst-case CSS Ω^k_{bc} can be calculated from best-case CSS Ω^k_{bc} similarly to the inverse formulations of Equations (4.108) and (4.109). Note that for an exact description of $\check{\Omega}_{\mathrm{c,bc}}$, further constraints could be included. These would circumvent this drawback. However, as the procedure to obtain these constraints is more complex in the general case than the approach considered here, it is not taken into account in the following.

Example 27. Given a situation from Example 3 with the system design space $\Omega_{\mathrm{ds}} = [0, 2] \times [0, 1.5]$, the performance function $f : \mathbb{R}^2 \times \mathbb{R}^2 \to \mathbb{R},\ (x, p) \mapsto x_1 + 2x_2$ with threshold $f_{\mathrm{c}} = 2$, and interval-type uncertainties in the controllable variables with $\bar{\delta}_1 = 0.3$, $\bar{\delta}_2 = 0.1$.
Using Equation (2.14), it holds

$$\begin{aligned}
\check{\Omega}_{\mathrm{c,bc}} &= \{(\check{x}_1, \check{x}_2) \in [-0.3, 2.3] \times [-0.1, 1.6] \mid \\
&\qquad \exists (x_1, x_2) \in \mathcal{U}^X((\check{x}_1, \check{x}_2), (0.3, 0.1)) :\ x_1 + 2x_2 \leq 2\} \\
&\subseteq \{(\check{x}_1, \check{x}_2) \in [-0.3, 2.3] \times [-0.1, 1.6] \mid \check{x}_1 + 2\check{x}_2 \leq 2.5\} \qquad (4.113) \\
&= \check{\Omega}_{\mathrm{s,bc}}. \qquad (4.114)
\end{aligned}$$

Here, the subset relation with $\check{\Omega}_{\mathrm{c,bc}} \neq \check{\Omega}_{\mathrm{s,bc}}$ holds as, for example, $(-0.3, 1.4) \in \check{\Omega}_{\mathrm{s,bc}}$, but also $(-0.3, 1.4) \notin \check{\Omega}_{\mathrm{c,bc}}$. This can be seen in Figure 7.

Next, interval uncertainty in uncontrollable parameters is considered. Therefore, uncontrollable parameters must be included in the system performance functions and thresholds. In the following, system performance functions which are linear in x, i.e.,

$$f_j(x, p) - f_{\mathrm{c},j}(p) = a_j(p)^{\mathrm{T}} x - b_j(p) \tag{4.115}$$

with $a_j(p) \in \mathbb{R}^d$, $b_j(p) \in \mathbb{R}$ for $p \in \mathbb{R}^q$, $j = 1, \ldots, m$, are taken into account again. Then, the constraints $f(x, p) \leq f_{\mathrm{c}}(p)$ build a system of linear inequalities in x, i.e., $A(p)x \leq b(p)$

where $a_j(p)$ are the rows of $A(p) \in \mathbb{R}^{m \times d}$ and $b_j(p)$ the entries of $b(p) \in \mathbb{R}^m$ for $j = 1, \ldots, m$. Furthermore, functions concerning $f(x,p) - f_{\mathrm{c}}(p)$ which are monotonic in p_l are considered, i.e., they are either *monotonically increasing* in p_l if

$$p_l' \leq p_l'' \quad \Rightarrow \quad A(p')x - b(p') \leq A(p'')x - b(p'') \tag{4.116}$$

or *monotonically decreasing* in p_l if

$$p_l' \leq p_l'' \quad \Rightarrow \quad A(p')x - b(p') \geq A(p'')x - b(p'') \tag{4.117}$$

holds for all $x \in \mathbb{R}^d$ with $l \in \{1, \ldots, q\}$ where $p', p'' \in [\check{p} - \gamma, \check{p} + \gamma]$ only differ in the l^{th} entry. Then, it holds

$$\begin{aligned} \Omega_{\mathrm{c,wc}} &= \{x \in \Omega_{\mathrm{ds}} \mid \forall p \in \mathcal{U}^P(\check{p}, \gamma) : \ A(p)x \leq b(p)\} \\ &= \{x \in \Omega_{\mathrm{ds}} \mid A(\check{p} + W_{\mathrm{wc}}^P \gamma) \leq b(\check{p} + W_{\mathrm{wc}}^P \gamma)\} \end{aligned} \tag{4.118}$$

and

$$\begin{aligned} \Omega_{\mathrm{c,bc}} &= \{x \in \Omega_{\mathrm{ds}} \mid \exists p \in \mathcal{U}^P(\check{p}, \gamma) : \ A(p)x \leq b(p)\} \\ &= \{x \in \Omega_{\mathrm{ds}} \mid A(\check{p} + W_{\mathrm{bc}}^P \gamma) \leq b(\check{p} + W_{\mathrm{bc}}^P \gamma)\} \end{aligned} \tag{4.119}$$

where $W_{\mathrm{wc}}^P, W_{\mathrm{bc}}^P \in \mathbb{R}^{p \times p}$ are diagonal matrices for that the l^{th} entries on the diagonals are given by

$$w_{\mathrm{wc},l}^P = \begin{cases} 1 & \text{if } A(p)x - b(p) \text{ mon. increasing in } p_l, \\ -1 & \text{if } A(p)x - b(p) \text{ mon. decreasing in } p_l, \end{cases} \tag{4.120}$$

$$w_{\mathrm{bc},l}^P = \begin{cases} -1 & \text{if } A(p)x - b(p) \text{ mon. increasing in } p_l, \\ 1 & \text{if } A(p)x - b(p) \text{ mon. decreasing in } p_l, \end{cases} \tag{4.121}$$

$l = 1, \ldots, p$. Hence, $\Omega_{\mathrm{c,wc}}$ and $\Omega_{\mathrm{c,bc}}$ have the same structure as Ω_{c} and CSS can be computed using the simplifications from above. Note that similar results are obtained for the worst-case complete system solution space if the functions concerning $f(x,p) - f_{\mathrm{c}}(p)$ are only component-wise monotonic in p, i.e., $a_j^{\mathrm{T}}(p')x - b_j(p') \leq a_j^{\mathrm{T}}(p'')x - b_j(p'')$ or $a_j^{\mathrm{T}}(p')x - b_j(p') \geq a_j^{\mathrm{T}}(p'')x - b_j(p'')$ for $p_l' \leq p_l''$, $j = 1, \ldots, m$, where $a_{j'}^{\mathrm{T}}(p')x - b_{j'}(p') \leq a_{j'}^{\mathrm{T}}(p'')x - b_{j'}(p'')$ and $a_{j''}^{\mathrm{T}}(p')x - b_{j''}(p') \geq a_{j''}^{\mathrm{T}}(p'')x - b_{j''}(p'')$ is possible for $j' \neq j''$, $j', j'' \in \{1, \ldots, m\}$. However, in doing so, only a superset might be yielded for the best-case complete system solution space due to similar reasons as discussed for the calculation of the best-case complete system solution space of target designs. Nevertheless, this superset can also be used for computing CSS if the permissibility of the resulting system designs is tested a posteriori. This is taken up again in Section 5.4. Furthermore, the methods to compute CSS under interval-uncertainty in both controllable variables and uncontrollable parameters can be combined, as discussed in

Section 3.4, using the corresponding simplifications of this section.

As the structures of the necessity-α and possibility-α complete system solution spaces are similar to the ones of worst- and best-case complete system solution spaces, $\alpha \in [0, 1]$, similar simplifications can be used if the system performance functions are linear in the controllable variables and monotonic in the uncontrollable parameters. Furthermore, this holds for the corresponding complete system solution spaces of target designs, too. The only exceptions are the possibility-0 complete system solution space and the possibility-0 complete system solution space of target design, which are usually non-closed sets, see Sections 2.2 and 4.1. Hence, it is preferred to define the possibility-0 CSS as open sets in the corresponding problem statements.
Although the simplification of the CSS problem statements for the specific system performance functions considered in this section solves the main obstacles for their numerical computation, there are still some open questions. These concern, for example, numerical tools for the computation of volumes and projections. They are discussed along with suitable numerical optimization algorithms to compute CSS in the next section.

4.3. Numerical Tools for Computing CSS

This section provides numerical tools for the computation of volumes and projections, which are used to optimize CSS. Furthermore, suitable and efficient optimization algorithms for the different CSS problem statements are presented.

4.3.1. Numerical Volume Computation

In the following, the numerical computation of the volume of CSS is addressed. For box-shaped CSS, the volume can be calculated analytically by Equation (2.35). This calculation is more complex for arbitrarily-shaped CSS. Here,

$$\mathrm{vol}(\Omega^1(g_\mathrm{c}^1) \times \cdots \times \Omega^n(g_\mathrm{c}^n)) = \prod_{k=1}^{n} \mathrm{vol}(\Omega^k(g_\mathrm{c}^k)) \tag{4.122}$$

holds, where $\Omega^k(g_\mathrm{c}^k)$, $k = 1, \ldots, n$, is given by Equation (3.10) for measurable component performance functions. Then, the volume of $\Omega^k(g_\mathrm{c}^k)$, $k = 1, \ldots, n$, can be approximated or computed analytically. As $\Omega^k(g_\mathrm{c}^k)$, $k = 1, \ldots, n$, have the same structure as Ω_c, the numerical volume computation is subsequently discussed for the complete system solution space, which can then be directly transferred to CSS.

In [21], a grid approximation method is proposed. For this purpose, Ω_ds is divided into a

uniform grid yielding N_t d-dimensional hyperrectangles. The function f is evaluated at the center of each hyperrectangle. If $f(x) \leq f_\mathrm{c}$ holds, the hyperrectangle that corresponds to the center x is considered as permissible and as part of the volume of Ω_c. Finally, $\mathrm{vol}(\Omega_\mathrm{c})$ is yielded by summing up the volumes of the permissible hyperrectangles. Denoting the amount of permissible hyperrectangles by N_p, it holds

$$\mathrm{vol}(\Omega_\mathrm{c}) = \frac{N_\mathrm{p}}{N_\mathrm{t}} \prod_{i=1}^{d} (x_{\mathrm{ds},i}^\mathrm{u} - x_{\mathrm{ds},i}^\mathrm{l}), \tag{4.123}$$

see [21]. The smaller the grid step, the more precise is the approximation of this volume. In general, this method gets computationally expensive for large d. Hence, it is only useful for the computation of arbitrarily-shaped CSS if d^k is small, e.g., $d^k \leq 3$, $k = 1, \ldots, n$.

To avoid this curse of dimensionality, Monte Carlo integration, see [40], can be used for an efficient approximation of the volume of Ω_c. Here, N_t independent, uniformly distributed sample points are generated in Ω_ds for which f is again evaluated at each sample point. Similar to the grid approximation method, a sample point x is considered as permissible if $f(x) \leq f_\mathrm{c}$ holds. Then, the volume of Ω_c is computed by dividing the number of permissible sample points N_p by the total number of sample points N_t and multiplying it with the volume of Ω_ds, which also yields Equation (4.123).
Nevertheless, optimizing the volume of CSS $\Omega^k(g_\mathrm{c}^k)$ using the Monte Carlo method might still be difficult because the objective function gets piece-wise constant as a possible increase in the volume of $\Omega^k(g_\mathrm{c}^k)$ for an increase in g_c^k depends on the number of permissible sample points which are added, $k = 1, \ldots, n$. In order to avoid a possible decrease in the volume when increasing g_c^k, it can be useful to create the sample points for every component only once during the optimization of CSS. If optimal CSS can be guaranteed in a neighborhood around an initial $\Omega^k_{f_{\mathrm{c},0}^k}$, $k = 1, \ldots, n$, a local Monte Carlo sampling further enhances the optimization.

Besides methods that approximate the volume of Ω_c, there are also exact methods to compute the volume Ω_c. In the following, the method presented in [87] that is suitable for linear system performance functions, cf. Equation (4.68), is considered in which the complete system solution space is a d-dimensional polytope. For reasons of simplicity, it is assumed in the following that the design space constraints are already included or rather covered by the system of linear inequalities $Ax \leq b$, $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, which describes the polytope Ω_c, i.e.,

$$\Omega_\mathrm{c} = \{x \in \mathbb{R}^d \mid Ax \leq b\}. \tag{4.124}$$

In [87], the volume of a d-dimensional polytope is computed via a recursion scheme which is based on Euler's theorem concerning homogeneous functions. Here, the volume of Ω_c is denoted by $V(d, A, b)$, i.e., $\mathrm{vol}(\Omega_\mathrm{c}) = V(d, A, b)$. It is shown that if $V(d, A, b)$ is differentiable

at b,

$$V(d, A, b) = \frac{1}{d} \sum_{j=1}^{m} \frac{b_j}{\|a_j\|} V_j(d-1, A, b) \tag{4.125}$$

holds where $V_j(d-1, A, b)$ is the volume of the j^{th} face of Ω_{c}, which is $\{x \in \mathbb{R}^d \mid a_j^{\text{T}} x = b_j, Ax \leq b\}$, $j \in \{1, \ldots, m\}$. The algorithm to compute $V(d, A, b)$ is presented in the following. Let $a_{j,i(j)} \neq 0$, $j \in \{1, \ldots, m\}$, $i \in \{1, \ldots, d\}$. First, $x_{i(j)}$ is removed from the inequality system $Ax \leq b$ using the equality $a_j^{\text{T}} x = b_j$ which holds for x on the j^{th} face, $j \in \{1, \ldots, m\}$. Then, a new inequality system $\tilde{A}_{i(j)} \tilde{x} \leq \tilde{b}$ is obtained with $\tilde{a}_{j'}^{\text{T}} \tilde{x} \leq \tilde{b}_{j'}$, $j' = 1, \ldots, m$, $j' \neq j$, where x_i, $i = 1, \ldots, d$, $i \neq i(j)$, are collected in $\tilde{x} \in \mathbb{R}^{d-1}$. The inequality system $\tilde{A}_{i(j)} \tilde{x} \leq \tilde{b}$ has $d-1$ variables and $m-1$ constraints and describes a convex polytope for which $V_j'(d-1, \tilde{A}_{i(j)}, \tilde{b})$ denotes its volume. It holds

$$V_j(d-1, A, b) = \frac{\|a_j\|}{|a_{j,i(j)}|} V_j'(d-1, \tilde{A}_{i(j)}, \tilde{b}), \tag{4.126}$$

see [87] for a proof. With Equation (4.126), Equation (4.125) can be reformulated as

$$V(d, A, b) = \frac{1}{d} \sum_{j=1}^{m} \frac{b_j}{|a_{j,i(j)}|} V_j'(d-1, \tilde{A}_{i(j)}, \tilde{b}), \tag{4.127}$$

which is also valid if redundant constraints are incorporated in $Ax \leq b$. Overall, a recursive scheme to compute $V(d, A, b)$ is obtained by Equation (4.127). After $d-1$ eliminations, the volume of the interval, defined by $\tilde{a}_{j'} \tilde{x} \leq \tilde{b}_{j'}$, $\tilde{a}_{j'}, \tilde{a}_{j'} \in \mathbb{R}$, $j' = 1, \ldots, m-d+1$, for $\tilde{x} \in \mathbb{R}$, must be computed. This volume is given by

$$\begin{aligned} &\text{vol}(\{\tilde{x} \in \mathbb{R} \mid \tilde{a}_{j'} \tilde{x} \leq \tilde{b}_{j'},\ j' = 1, \ldots, m-d+1\}) \\ &\qquad = \max \left\{ 0, \left(\min_{\substack{\tilde{a}_{j'} \in \mathbb{R}^+, \\ j' \in \{1, \ldots, m-d+1\}}} \frac{\tilde{b}_{j'}}{\tilde{a}_{j'}} - \max_{\substack{\tilde{a}_{j'} \in \mathbb{R}^-, \\ j' \in \{1, \ldots, m-d+1\}}} \frac{\tilde{b}_{j'}}{\tilde{a}_{j'}} \right) \right\}. \end{aligned} \tag{4.128}$$

The computational effort of this algorithm depends on the number of design variables d as well as on the number of constraints for which $b_j \neq 0$, $j = 1, \ldots, m$, hold. Its efficiency is compared to the volume approximation with the Monte Carlo method for the computation of arbitrarily-shaped CSS in Section 5.3.

Other analytical methods to compute the volume of Ω_{c} often require the values of its corner points, see [14] for an overview. For non-linear functions f, an analytic calculation of Ω_{c} is more complex and not considered here. Subsequently, a method for the numerical projection of Ω_{c} is presented.

4.3.2. Numerical Projection

If Ω_{c} is described by linear system performance functions, its projection onto the coordinate space of the k^{th} component, i.e., $\text{proj}^k(\Omega_{\text{c}})$, $k \in \{1, \ldots, n\}$, is a convex polytope, see Section A.1. As already discussed in Section 3.3, the projection of a linear polytope can be done in several ways, compare [71]. In the following, the *Fourier-Motzkin elimination method*, cf.,

[97], is discussed. It is an analytic method that computes a system of linear constraints. Here, the idea is to remove design variables $x_{i_1}, \ldots, x_{i_{d^k-k}}$ that are entries of the vectors $x^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$, successively from the inequality system $Ax \leq b$ such that $A^k \in \mathbb{R}^{\tilde{m} \times d^k}$ and $b^k \in \mathbb{R}^{\tilde{m}}$ with

$$\exists x \in \mathbb{R}^d : \; Ax \leq b \quad \Leftrightarrow \quad \exists x^k \in \mathbb{R}^{d^k} : A^k x^k \leq b^k, \tag{4.129}$$

$k \in \{1, \ldots, n\}$, are obtained. Equivalence (4.129) motivates the primary intention of the Fourier-Motzkin elimination to test if there exists at least one $x \in \mathbb{R}^d$ that fulfills the system of linear inequalities $Ax \leq b$, i.e., if Ω_c given by Equation (4.124) is a non-empty set. The algorithm to obtain $A^k x^k \leq b^k$, $k \in \{1, \ldots, n\}$, is presented in the following.

First, x_{i_1} is removed from the inequality system $Ax \leq b$. Therefore, the rows of A and the corresponding entries of b are multiplied by positive constants yielding A', b' such that $a'_{j,i_1} \in \{1, -1, 0\}$ holds. Let J_1 be the set of indices with $a'_{j,i_1} = 1$, J_{-1} be the set of indices with $a'_{j,i_1} = -1$, and J_0 be the set of indices with $a'_{j,i_1} = 0$, $j = 1, \ldots, m$. For $a'_{j,i}$ collected in $\tilde{a}_j \in \mathbb{R}^{d-1}$ and x_i collected in $\tilde{x} \in \mathbb{R}^{d-1}$, $i = 1, \ldots, d$, $i \neq i_1$, $j = 1, \ldots, m$, it is

$$x_{i_1} \leq -\tilde{a}_{j'}^\mathrm{T} \tilde{x} + b_{j'} \tag{4.130}$$

for all $j' \in J_1$ and

$$-x_{i_1} \leq -\tilde{a}_{j''}^\mathrm{T} \tilde{x} + b_{j''} \tag{4.131}$$

for all $j'' \in J_{-1}$. The inequalities (4.130) and (4.131) are equivalent to

$$\max_{j' \in J_1} (\tilde{a}_{j'}^\mathrm{T} \tilde{x} - b_{j'}) \leq x_{i_1} \leq \min_{j'' \in J_{-1}} (-\tilde{a}_{j''}^\mathrm{T} \tilde{x} + b_{j''}). \tag{4.132}$$

Hence, x_{i_1} can be removed and $A'x \leq b'$ is equivalent to the system of linear inequalities $\tilde{A}\tilde{x} \leq \tilde{b}$ given by

$$(\tilde{a}_{j'}^\mathrm{T} + \tilde{a}_{j''}^\mathrm{T})\tilde{x} \leq b_{j'} + b_{j''}, \tag{4.133}$$

$$\tilde{a}_{j'''}^\mathrm{T} \tilde{x} \leq b_{j'''} \tag{4.134}$$

for all $j' \in J_1$, $j'' \in J_{-1}$, and $j''' \in J_0$, compare [116]. The inequality system $\tilde{A}\tilde{x} \leq \tilde{b}$ has $d - 1$ variables and consists of $m'(m'' - m') + m - m''$ inequalities where m' denotes the number of indices contained in J_1 and m'' denotes the number of indices contained in J_{-1}. It describes the convex polytope that is yielded when Ω_c is projected along the $i_1{}^{\mathrm{th}}$ coordinate axis. Thus, it holds that there exists at least one $x_{i_1} \in \mathbb{R}$ that fulfills (4.132) for all $\tilde{x} \in \mathbb{R}^{d-1}$ that fulfill $\tilde{A}\tilde{x} \leq \tilde{b}$.

This procedure can be continued by a further, successive elimination of the design variables $x_{i_2}, \ldots, x_{i_d}$, which yields the projection of Ω_c onto the coordinate space of the k^{th} component, i.e., $\mathrm{proj}^k(\Omega_\mathrm{c})$ for which $A^k x^k \leq b^k$ holds for all $x^k \in \mathrm{proj}^k(\Omega_\mathrm{c})$, $k \in \{1, \ldots, n\}$.

The computational effort of this algorithm depends on $d - d^k$, $k \in \{1, \ldots, n\}$, as well as on

the number of constraints. As the Fourier-Motzkin elimination produces redundant constraints when eliminating a variable, it can be combined with a redundancy removal strategy to speed up computation time. Here, the goal is to remove all the constraints from an inequality system for which the set of solutions remains unchanged when the constraints are removed from the system. As suggested in [128], the strategy proposed in [18] is useful here. It is presented for Ω_c given by Equation (4.124) in the following and can be directly transferred to any reduced system $\tilde{A}\tilde{x} \leq \tilde{b}$, obtained during the Fourier-Motzkin elimination.

Assume that an interior design of Ω_c exists, i.e., $x_0 \in \Omega_\mathrm{c}$ with $Ax_0 < b$. Such a design can, for example, be found by solving the unconstrained optimization problem

$$\underset{x}{\text{minimize}} \max_{j\in\{1,\dots,m\}} a_j^\mathrm{T} x - b, \tag{4.135}$$

which is equivalent to the linear optimization problem

$$\begin{aligned} \underset{x,\zeta}{\text{minimize}} \quad & \zeta \\ \text{subject to} \quad & a_j^\mathrm{T} x \leq b + \zeta,\ j = 1,\dots,m. \end{aligned} \tag{4.136}$$

Problem (4.136) can be solved using standard methods for linear optimization, cf., [100]. If $\zeta < 0$ holds or $\max\{a_j^\mathrm{T} x - b \mid j = 1,\dots,m\} < 0$ in case of problem (4.135), an interior design can be guaranteed.
Let J_nr be the set of indices for which the constraints $a_j^\mathrm{T} x \leq b_j$ are non-redundant in $Ax \leq b$, and J_nt be the set of indices for which it is not tested yet whether the constraints $a_j^\mathrm{T} x \leq b_j$ are redundant or non-redundant in $Ax \leq b$, $j = 1,\dots,m$. Initially, it holds $J_\mathrm{nt} = \{1,\dots,m\}$, and $J_\mathrm{nr} = \emptyset$ is assumed. The set J_nr is updated by a recursive scheme.
Let $j' \in J_\mathrm{nt}$ be an index for which it shall be tested whether the constraint $a_{j'}^\mathrm{T} x \leq b_{j'}$ is redundant in the inequality system $a_j^\mathrm{T} x \leq b_j$, $\forall j \in J_\mathrm{nr}$. Thus, the problem

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & -a_{j'}^\mathrm{T} x \\ \text{subject to} \quad & a_j^\mathrm{T} x \leq b_j,\ \forall j \in J_\mathrm{nr}, \\ & a_{j'}^\mathrm{T} x \leq b_{j'} + 1 \end{aligned} \tag{4.137}$$

is considered, which is again a linear optimization problem. If $-a_{j'}^\mathrm{T} x < -b_{j'}$ holds for an optimal solution $x \in \mathbb{R}^d$ of problem (4.137), the constraint $a_{j'}^\mathrm{T} x \leq b_{j'}$ is non-redundant in $a_j^\mathrm{T} x \leq b_j$, $\forall j \in J_\mathrm{nr}$. In this case, also an index $j'' \in \{1,\dots,m\}$ can be found for which $a_{j''}^\mathrm{T} x \leq b_{j''}$ is non-redundant in $Ax \leq b$. It is the index of the smallest $t_j > 0$ for which

$$a_j^\mathrm{T}(x_0 + t_j(x - x_0)) = b_j, \tag{4.138}$$

$j = 1, \ldots, m$, holds, compare [125]. Then, J_{nr} is updated by its union with $\{j''\}$, i.e., $J_{\mathrm{nr}} \cup \{j''\}$, and J_{nt} is updated by subtracting $\{j''\}$, i.e., $J_{\mathrm{nt}} \backslash \{j''\}$. This procedure is continued further until $J_{\mathrm{nt}} = \emptyset$ is obtained. Then, the set J_{nr} contains all non-redundant constraints.
The efficiency of the Fourier-Motzkin algorithm for computing box-shaped dependent CSS with the presented redundancy removal strategy, referred to as two-step method in the following, is compared to the direct computation of box-shaped dependent CSS by solving problem (4.95) in Section 5.3. Note that in general there are further redundancy removal strategies, cf. [125]. Moreover, for non-linear functions f, the computation of the projection $\mathrm{proj}^k(\Omega_{\mathrm{c}})$, $k \in \{1, \ldots, n\}$, is more complex.

Overall, all prerequisites to solve numerically the CSS optimization problems for system performance functions which are linear in the controllable variables and monotonic in the uncontrollable parameters are met. Thus, suitable and efficient numerical optimization algorithms, which can be used to solve the simplified problems of Section 4.2, are considered in the following.

4.3.3. Optimization Algorithms to Compute CSS

The subsequent considerations provide an overview of numerical optimization algorithms which can be used to compute CSS using the simplifications of Section 4.2 and the numerical tool from above. Here, the focus is put on algorithms that are embedded in the software MATLAB. As all simplified problem statements form convex optimization problems for which every local optimum is also a global one, only local optimization methods are addressed. Corresponding MATLAB algorithms that help in solving these problems are mentioned and described briefly below. Furthermore, suitable initial values, which are usually required for the numerical algorithms, are discussed.

Linear optimization problems can be solved using the MATLAB `linprog` command. Examples of such problems are problems (4.99) and (4.100) to compute dependent CSS for $d^k = 1$, $k \in \{1, \ldots, n\}$, problem (4.101) to determine maximum magnitudes for unknown uncertainty magnitudes in controllable variables, problem (4.136) to find an interior design of Ω_{c}, and problem (4.137), which is used for redundancy removal.
The `'dual-simplex'` algorithm of `linprog` performs a simplex algorithm on the dual problem. Here, the constrained linear minimization problem that has to be solved is reduced to an equality-constrained minimization problem, compare [100]. Then, its dual problem is considered for which the objective function is perturbed. First, a dual feasible point is sought by solving an auxiliary linear optimization problem. Then, entering and leaving variables are chosen iteratively until the solution to the perturbed optimization problem is both primal and dual feasible, i.e., an optimal solution is found. See [93] for more details. Here, no initial values are required.

Differentiable convex optimization problems can be solved using the `fmincon` command. Examples of such problems are problem (4.70) to compute box-shaped independent CSS, problem (4.95) to compute box-shaped dependent CSS, and problem (4.81) to compute arbitrarily-shaped independent CSS if the objective function is differentiable, as discussed in Section 4.2.

The `'interior-point'` algorithm of `fmincon` finds an optimal solution of a differentiable constraint optimization problem by solving a sequence of approximate optimization problems for which initial values must be provided. Each approximate problem uses slack variables to transform inequality constraints into equality constraints and adds a barrier function to the objective function. The barrier function is described by the sum of the logarithms of the slack variables and a parameter which leads the optimal solution of an approximate problem towards the solution of the original problem as it decreases to zero, compare [100]. In order to solve an approximate problem, a direct step, using linear approximation, and a conjugate gradient step, using a trust region, are performed at each iteration. See [93] for more details.

Initial values for problem (4.70) can be computed, for instance, by solving problem (4.135) or (4.136) and setting $x_{0,i}^{\mathrm{l}} = x_i - \varepsilon_i$, $x_{0,i}^{\mathrm{u}} = x_i + \varepsilon_i$ where x_i form an optimal solution of problem (4.135) and ε_i are small enough so that the optimization constraints of problem (4.70) are fulfilled, $i = 1, \ldots, d$. This can, for example, be achieved for

$$\varepsilon_i = \frac{1}{d} \min_{j \in \{j' \in \{1,\ldots,m\} \mid |a_{j',i}| \neq 0\}} \frac{-a_j^{\mathrm{T}} x + b_j}{|a_{j,i}|}, \tag{4.139}$$

where there exists at least one $j \in \{1, \ldots, m\}$ with $|a_{j,i}| > 0$, $i = 1, \ldots, d$. This is due to the fact that Ω_{c} is bounded as

$$\begin{aligned} \sum_{i=1}^{d} a_{j,i} v_{j,i}^{\mathrm{l}} (x_i - \varepsilon_i) + \sum_{i=1}^{d} a_{j,i} v_{j,i}^{\mathrm{u}} (x_i + \varepsilon_i) &= \sum_{i=1}^{d} (a_{j,i} x_i + |a_{j,i}| \varepsilon_i) \\ &\leq a_j^{\mathrm{T}} x + \sum_{i=1}^{d} \frac{-a_j^{\mathrm{T}} x + b_j}{d} \\ &= b, \end{aligned} \tag{4.140}$$

where $v_{j,i}^{\mathrm{l}}$, $v_{j,i}^{\mathrm{u}}$ are given by Equation (4.71), $i = 1, \ldots, d$, $j = 1, \ldots, m$.

For problem (4.95), initial values can be computed similarly via $x_{0,i}^{\mathrm{l},k} = x_i^k - \varepsilon_i$, $x_{0,i}^{\mathrm{u},k} = x_i^k + \varepsilon_i$ with, e.g.,

$$\varepsilon_i = \frac{1}{d^k} \min_{j \in \{j' \in \{1,\ldots,m\} \mid |a_{j',i}| \neq 0\}} \frac{-a_j^{\mathrm{T}} x + b_j}{|a_{j,i}|}, \tag{4.141}$$

for which there is at least one $j \in \{1, \ldots, m\}$ with $|a_{j,i}| > 0$, $i = 1, \ldots, d^k$, and $x_{\mathrm{cp},\iota,0}^{\mathrm{r},k} = x^{\mathrm{r},k}$, $\iota = 1, \ldots, 2^{d^k}$. The entries of x that do not belong to the k^{th} component are collected in $x^{\mathrm{r},k}$, $k \in \{1, \ldots, n\}$. Moreover, initial values for problem (4.95) can also be obtained by solving problem (4.70) first and setting $x_{0,i}^{\mathrm{l},k} = x_i^{\mathrm{l},k}$, $x_{0,i}^{\mathrm{u},k} = x_i^{\mathrm{u},k}$, and $x_{\mathrm{cp},\iota,0}^{\mathrm{r},k} = (x^{\mathrm{l},\mathrm{r},k} + x^{\mathrm{u},\mathrm{r},k})/2$, $\iota = 1, \ldots, 2^{d^k}$. Here, x^{l}, x^{u} are an optimal solution of problem (4.70) for which the entries that

do not belong to the k^{th} component are collected in $x^{\text{l,r},k}$ and $x^{\text{u,r},k}$, $k \in \{1,\ldots,n\}$.
For problem (4.81), initial values can be chosen as $b_{0,j}^k = (a_j^k)^{\text{T}} x^k + \varepsilon_j^k$ where x is again an optimal solution of problem (4.135) or (4.136) and ε_k^j are small enough so that the optimization constraints of problem (4.81) are satisfied, e.g.,

$$\varepsilon_k^j = \frac{-a_j^{\text{T}} x + b_j}{n}, \tag{4.142}$$

$j = 1,\ldots,m$, $k = 1,\ldots,n$. Moreover, initial values for problem (4.81) can also be obtained by solving problem (4.70) first and choosing $b_{0,j}^k = (a_j^k)^{\text{T}} V_j^{\text{l},k} x^{\text{l},k} + (a_j^k)^{\text{T}} V_j^{\text{u},k} x^{\text{u},k}$ where $V_j^{\text{l},k}, V_j^{\text{u},k} \in \mathbb{R}^{d^k \times d^k}$ are diagonal matrices for that the i^{th} entries on the diagonals are given by

$$v_{j,i}^{\text{l},k} = \begin{cases} 1 & \text{if } a_{j,i}^k \leq 0, \\ 0 & \text{else,} \end{cases} \qquad v_{j,i}^{\text{u},k} = \begin{cases} 0 & \text{if } a_{j,i}^k \leq 0, \\ 1 & \text{else,} \end{cases} \tag{4.143}$$

$i = 1,\ldots,d$, $j = 1,\ldots,m$, $k = 1,\ldots,n$.

Differentiable, unconstrained, convex optimization problems can be solved using the `fminunc` command. An example of such a problem is problem (4.92) to compute arbitrarily-shaped independent CSS if the objective function is computed analytically and is furthermore differentiable, as discussed in Section 4.2.
The `'quasi-newton'` algorithm of `fminunc` finds an optimal solution of a differentiable unconstrained optimization problem using the Broyden-Fletcher–Goldfarb-Shanno (BFGS) Quasi-Newton method with a cubic line search procedure for which initial values must be provided. The problem is approximated quadratically by incorporating curvature information at each iteration. Here, the Hessian is updated by a formula that uses gradient information, compare [100]. In order to find a new iterate, the line search algorithm uses a bracketing phase for its range and a sectioning phase that divides the range and approximates the optimum using cubic interpolation. See [93] for more details. Initial values for problem (4.92) can be computed similarly to initial values for problem (4.81) as described above, for which only $b_{0,j}^k$, $j = 1,\ldots,m$, $k = 1,\ldots,n-1$, are used.

Non-differentiable, unconstrained, convex optimization problems can be solved using the `fminsearch` command. An example of such a problem is problem (4.92) to compute arbitrarily-shaped independent CSS if the objective function is non-differentiable, e.g., when using the Monte-Carlo method for volume computation.
The algorithm implemented in `fminsearch` attempts to find an optimal solution of an unconstrained optimization problem using the Nelder-Mead simplex search method described in [83]. It is a direct search method that does not require gradient information. Here, a simplex is spanned around the initial values and modified repeatedly afterward. Note that this algorithm does not necessarily converge to a local minimum. See [93] for more details. Initial values for problem (4.92) can be computed as described above.

The optimization algorithms presented in this subsection complete the discussion on how to compute the different types of CSS numerically. Note again that the results of this and the previous section can only be used for system performance functions which are linear in the controllable variables and monotonic in the uncontrollable parameters. For more complex system performance functions, further investigations must be done to enable the computations of CSS. In the next chapter, the methods proposed in this thesis are applied to crash design by using the simplifications and numerical tools of this chapter.

5. APPLICATION: CSS for Crash Design

This chapter applies the methods of this thesis to crash design. Therefore, existing crash design models are reviewed and enhanced, and two test-bed problems are defined. For these problems, box-shaped and arbitrarily-shaped CSS are compared and CSS under epistemic uncertainty are computed. In addition, a MATLAB app is introduced that is capable of computing the CSS of this thesis for enhanced crash design models, which can also be built with this app.

5.1. Crash Design Basics

This section gives an introduction to crash design. Here, the focus is put on deformation space models for which details on their modeling are provided. Furthermore, a simple test-bed problem is defined in order to evaluate the methods of this thesis in the subsequent sections.

5.1.1. Introduction to Crash Design

Designing a vehicle which must perform in a crash can be considered as a systems design problem. In particular, the focus of this thesis is put on the early design phase of the vehicle's frontal structure and a frontal crash at full overlap against a rigid wall. A frontal crash can be simulated, for example, by the finite element method (FEM), cf. [1]. In doing so, a finite element (FE) model of the vehicle is required. In Figure 42, the results of a frontal crash simulation at full overlap against a rigid wall are shown for a detailed HONDA ACCORD FE model from [98].

However, the drawbacks of using the FEM for crash simulations are that it is computationally very expensive and that detailed FE models are usually not available in the early design phase. This gives rise to surrogate modeling. In general, surrogate models can be either classified as physical surrogates, for which simplified physical characteristics are used to obtain mathematical models, or mathematical surrogates, for which the system responses are approximated by mathematical functions without taking physical characteristics for the approximation into account. An overview of surrogate models for crash design can be found in, e.g., [34] in which also a discussion concerning optimization and robustness is provided.
Mathematical surrogates are usually based on high-fidelity FE models. For example, there are response surface methods (RSM) that approximate the system responses using samples that are generated from high-fidelity models. These methods include polynomial models, Gaussian processes or Kriging, support vector machines (SVM), and artificial neural networks (ANN), see [46] for an overview. Besides, there is model order reduction (MOR) which aims at reducing the computational complexity of the mathematical models. Here, advanced approaches

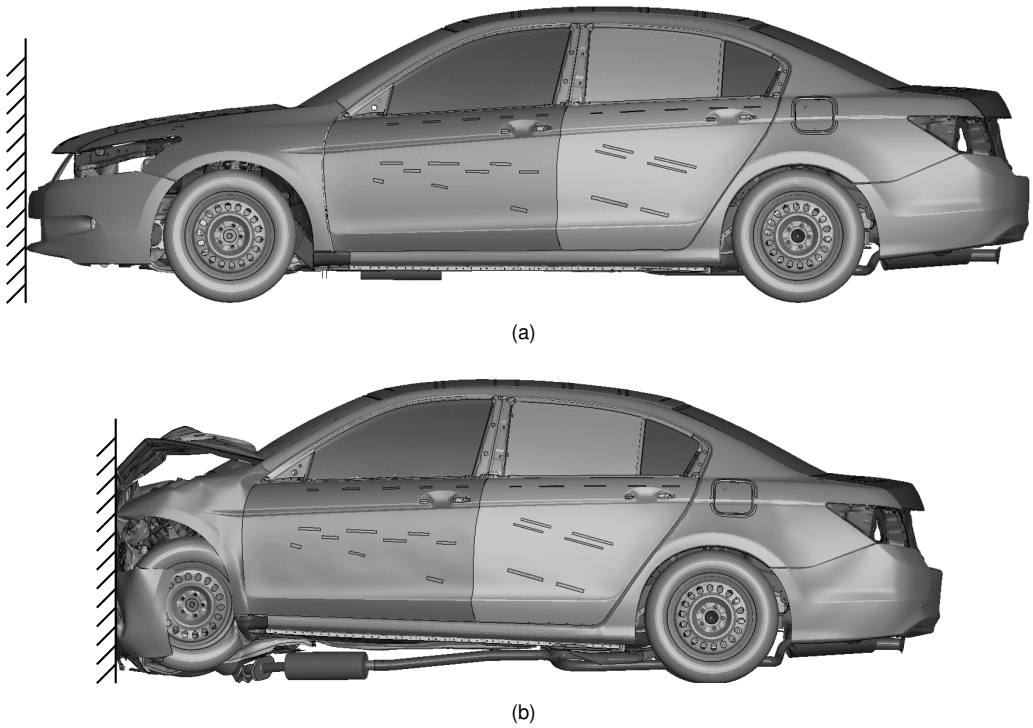


Figure 42 Example of an FEM frontal crash simulation at full overlap against a rigid using the software LS-DYNA and a HONDA ACCORD FE model from [98] (a) before and (b) after the crash.

are, for example, system equivalent reduction expansion processes (SEREP), see [101] and Krylov subspace methods, see [3, 5].
Low-fidelity physical surrogate models can be used to avoid high-fidelity models. Amongst others, there are multi-body system approaches that use, for example, lumped masses connected by dampers and springs, i.e., lumped mass-spring models (LMS), see [16, 73], or that model parts of the simplified model by finite elements and the rest, for instance, by rigid bodies, i.e., hybrid FE approaches, see [19, 92]. Furthermore, there are substructure approaches in which a part of the vehicle is cut out, cf. [41, 112]. This list is by far not complete, see [1, 34, 85] for more information.

A type of model for which box-shaped solution spaces were computed in literature are deformation space models (DSM). These are models for the early design phase that can be classified as a multi-body system approach. DSM were introduced in [43] and are considered in, e.g., [44, 85] as well. In order to follow up this discussion, DSM are also taken into account for decoupled design decisions in crash design here.
Deformation space models are crash design models for which the responses of the crash system, like energy absorption and acceleration, depend on the force-deformation characteristics of $n \in \mathbb{N}$ structural, crash-relevant components. For DSM, only forces and deformations in the longitudinal direction are taken into account in accordance with the longitudinal impact direction. Deformations in other directions and rotations are not considered. For other crash load cases, this is taken into account in, e.g., [20, 122]. By neglecting the elastic deformation,

it is assumed that for every point in the deformation space $s \in [s_0, s_{\text{end}})$, there are forces $F^k(s)$, $k = 1, \ldots, n$, responsible for plastic deformation. Here, $s_0 \in \mathbb{R}_0^+$ denotes the start and $s_{\text{end}} \in \mathbb{R}_0^+$ the end of deformation of the vehicle. Thus, the force-deformation characteristics are described by the graphs $\{(s, F^k(s)) \mid s \in [s_0, s_{\text{end}})\}$, $k = 1, \ldots, n$. Note that force-deformation characteristics of the components usually depend on boundary conditions like the initial velocity of the vehicle before the impact, cf. [117].
Using DSM for crash design, it is assumed that the forces $F^k(s)$ can be designed for every $s \in [s_0, s_{\text{end}})$, $k = 1, \ldots, n$. This is in general not the case, as DSM only represent the uppermost system performance functions of a multi-level crash system. Recall that multi-level systems are considered Section 3.1. In this thesis, a three-level crash system is considered with system performance functions at level 3, given by a DSM, and n structural components at level 2 that form systems themselves, i.e., there are lower-level design variables at level 1. Thus, the force-deformation characteristics are responses of the systems at level 2. Each of the systems at level 2 is defined by a corresponding model and design variables, which are grouped as components at level 1. These design variables relate, for example, to geometric or material properties of the structural components. In Figure 43, such a crash system is visualized.

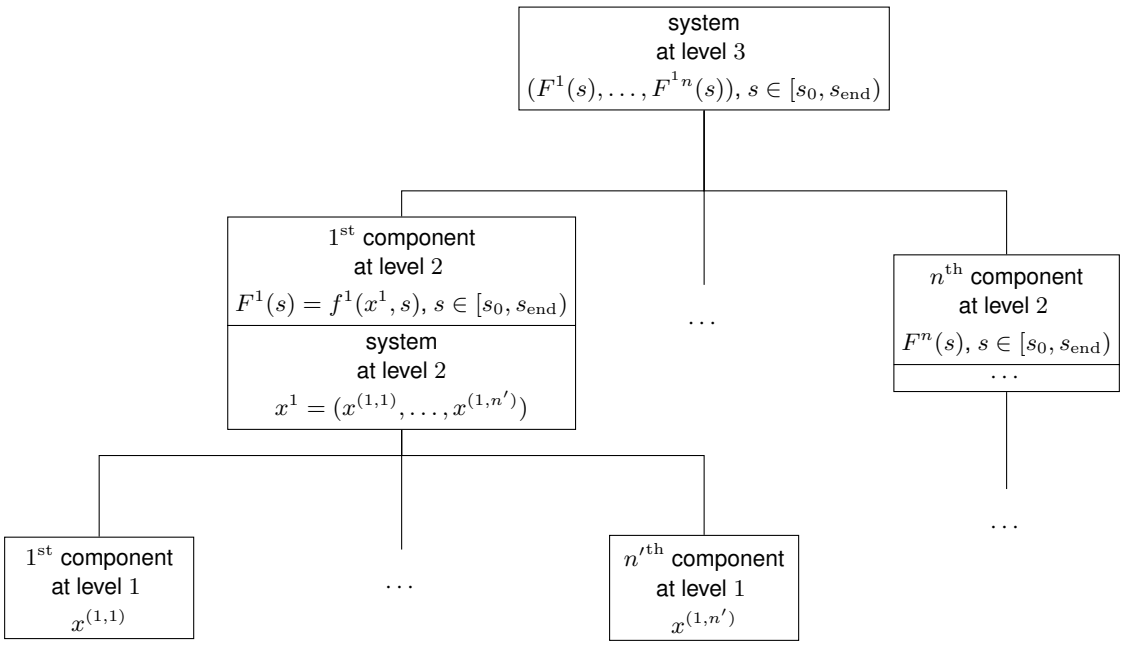


Figure 43 Three-level crash system for which the design variables at each level are grouped as system designs at the corresponding system level and as component designs at the corresponding component level. Note that the notation n' is chosen for reasons of simplicity.

Note that the design models for systems at level 2 might not be available in the early design phase. Still, CSS can be computed for the force-deformation characteristics of the structural components using DSM. The design models of the systems at level 2 must then follow in a second step, which is further discussed in Chapter 6. In the following, only the system at level 3 with components at level 2 is regarded for which two-level system notations are used, see Section 3.1.

In order to compute CSS, requirements on the system responses of DSM must be formulated. Here, requirements regarding the minimum energy absorption, maximum acceleration, and progressive order of deformation are considered. They are motivated by crash test criteria like considered in the US-NCAP crash load case to obtain a 5-star safety rating, see [99], and are investigated below. Furthermore, the state-of-the-art of DSM is discussed in more detail subsequently. Compared to previous work which limits its consideration to parametrized force-deformation characteristics by modeling the forces as piece-wise constant, this thesis discusses DSM for non-parametrized force-deformation characteristics as well.

5.1.2. Test Bed 1: Simple Crash Design Problem

At first, a simple crash design problem is investigated, for which the vehicle front structure is divided into two sections, see Figure 44. This example is based on the model introduced in [140] and is used as a test-bed problem for computing CSS for crash design later.

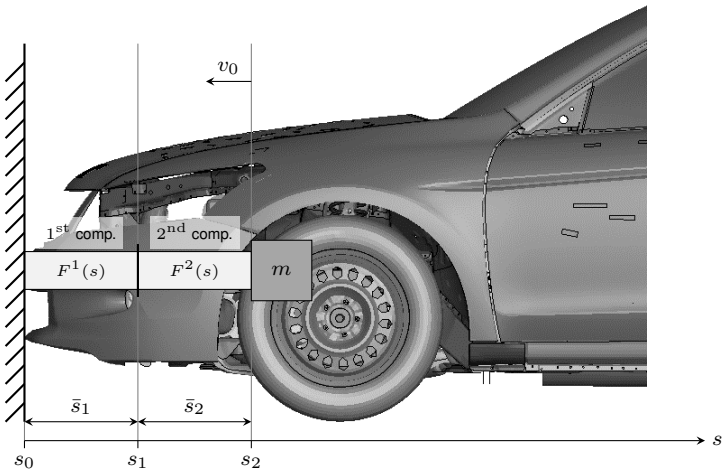


Figure 44 Model of a vehicle front structure with two components and underlying vehicle chassis, after [140].

In a local vehicle coordinate system, section 1 starts deforming at $s_0 \in \mathbb{R}_0^+$ and ends at $s_1 \in \mathbb{R}^+$ and section 2 starts deforming at s_1 and ends at $s_2 \in \mathbb{R}^+$, where $s_0 < s_1 < s_2$ and $s_{\text{end}} = s_2$. In this example, the front structure of the vehicle in each section is modeled as a component. This means there are two front end components here. At the end of these components, i.e., at s_2, the total *vehicle mass* $m \in \mathbb{R}^+$ is lumped. The lengths of the sections are $\bar{s}_1 = s_1 - s_0$ and $\bar{s}_2 = s_2 - s_1$, which correspond to the *deformation lengths* of the components in longitudinal impact direction here. As described above, the components can be represented by force-deformation characteristics. For the first component, it is $F^1(s) \in \mathbb{R}_0^+$ if $s \in [s_0, s_1)$ and $F^1(s) = 0\,\text{kN}$ otherwise, and for the second component, it is $F^2(s) \in \mathbb{R}_0^+$ if $s \in [s_1, s_2)$ and $F^2(s) = 0\,\text{kN}$ otherwise. After hitting the barrier at full overlap with initial velocity v_0, the structure deforms. If $F^1(s) < F^2(s_1)$ holds for $s \in [s_0, s_1)$, the first component deforms completely before the second component starts deforming.

For the early design stage, the vehicle mass and the deformation lengths are assumed. Thus, the mass and the deformation lengths are uncontrollable parameters. The assumptions for these uncontrollable parameters can be either made without deeper knowledge or based on knowledge from similar vehicles that are already fully developed. Note that this is a source of uncertainty, which is treated in the next section. In contrast to the vehicle mass and the deformation lengths, the force-deformation characteristics of the components are controllable by selecting the values of the lower-level design variables in a subsequent step. There are minimum forces that $F^1(s)$ can assume in $[s_0, s_1)$ and $F^2(s)$ can assume in $[s_1, s_2)$. These are $F_{\mathrm{ds}}^{\mathrm{l},k} \in \mathbb{R}_0^+$, $k = 1, 2$, and represent the component design spaces. Similarly, there are maximum forces $F_{\mathrm{ds}}^{\mathrm{u},k} \in \mathbb{R}_0^+$ with $F_{\mathrm{ds}}^{\mathrm{l},k} \leq F_{\mathrm{ds}}^{\mathrm{u},k}$, $k = 1, 2$.

Using the vehicle mass, the deformation lengths, and the force-deformation characteristics, the requirements on the minimum energy absorption, the maximum acceleration, and the progressive order of deformation, which are mentioned above, can be formulated mathematically as shown below. Here, the requirements which are stated in [140] are generalized for non-parametrized force-deformation characteristics:

- *Minimum energy absorption*: The impact energy $\frac{1}{2}mv_0^2$ must be completely absorbed in the front structure, meaning

$$-\int_{s_0}^{s_1} F^1(s)\,\mathrm{d}s - \int_{s_1}^{s_2} F^2(s)\,\mathrm{d}s \leq -\frac{1}{2}mv_0^2. \tag{5.1}$$

  Then, there is no intrusion into the occupant compartment.

- *Maximum acceleration*: The acceleration must be smaller than a critical acceleration threshold a_{c}, i.e.,

$$F^1(s) \leq ma_{\mathrm{c}} \tag{5.2}$$

  for all $s \in [s_0, s_1)$ and

$$F^2(s) \leq ma_{\mathrm{c}} \tag{5.3}$$

  for all $s \in [s_1, s_2)$. This requirement is also injury-related and regulated protection thresholds for different parts of the human can be found in, e.g., [78]. As a conservative threshold, $a_{\mathrm{c}} = 0.3\,\frac{\mathrm{mm}}{\mathrm{ms}^2}$ is used throughout this thesis.

- *Progressive order of deformation*: The ordered deformation of the vehicle must start at the front, meaning

$$F^1(s) \leq F^2(s_1). \tag{5.4}$$

  for all $s \in [s_0, s_1)$. In contrast to the other two requirements, this requirement is usually intended for the reparability of the vehicle in a low-speed crash, see [88].

The values of the design space parameters and the uncontrollable parameters which are used for the requirements (5.1)-(5.4) in this test-bed problem are stated in Table 2 and 3. Without loss of generality, s_0 is always set to $s_0 = 0\,\mathrm{mm}$ for the remaining thesis.

Table 2 Design space parameters for test-bed problem 1

quantity	$F_{\mathrm{ds}}^{\mathrm{l},1}$	$F_{\mathrm{ds}}^{\mathrm{l},2}$	$F_{\mathrm{ds}}^{\mathrm{u},1}$	$F_{\mathrm{ds}}^{\mathrm{u},2}$
value	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$	$500\,\mathrm{kN}$	$500\,\mathrm{kN}$

Table 3 Uncontrollable parameters for test-bed problem 1

quantity	$\bar{s}_1$	$\bar{s}_2$	m	v_0	a_{c}
value	$350\,\mathrm{mm}$	$350\,\mathrm{mm}$	$1500\,\mathrm{kg}$	$15.6\,\frac{\mathrm{mm}}{\mathrm{ms}}$	$0.3\,\frac{\mathrm{mm}}{\mathrm{ms}^2}$

In general, the goal for the simple crash design problem is to provide optimal flexibility for component design regarding the force-deformation characteristics of each component such that a permissible system design can be obtained. To compute optimal CSS by applying the concepts of this thesis, the considerations are limited to parametrized force-deformation characteristics. Thus, their degrees of freedom can be used as design variables.

Force-deformation characteristics which are constant during the deformation of each component are examples of characteristics with one degree of freedom. They can be defined as the graphs that belong to $F^1(s) = F_1^1$, $F_1^1 \in [F_{\mathrm{ds}}^{\mathrm{l},1}, F_{\mathrm{ds}}^{\mathrm{u},1}]$ if $s \in [s_0, s_1)$, and $F^1(s) = 0$ otherwise, and $F^2(s) = F_1^2$, $F_1^2 \in [F_{\mathrm{ds}}^{\mathrm{l},2}, F_{\mathrm{ds}}^{\mathrm{u},2}]$ if $s \in [s_1, s_2)$, and $F^2(s) = 0$ otherwise. Thus, there are two design variables, which are F_1^1 and F_1^2, overall. In this case, the test-bed problem 1 matches the crash design example from [140] in which no values for the uncontrollable parameters are provided. The requirements (5.1)-(5.4) become a system of linear inequalities of the form $A^1 F_1^1 + A^2 F_1^2 \leq b$, i.e.,

$$\underbrace{\begin{pmatrix} -\bar{s}_1 \\ 1 \\ 0 \\ 1 \end{pmatrix}}_{=A^1} \begin{pmatrix} F_1^1 \end{pmatrix} + \underbrace{\begin{pmatrix} -\bar{s}_2 \\ 0 \\ 1 \\ -1 \end{pmatrix}}_{=A^2} \cdot \begin{pmatrix} F_1^2 \end{pmatrix} \leq \underbrace{\begin{pmatrix} -\frac{1}{2} m v_0^2 \\ m a_{\mathrm{c}} \\ m a_{\mathrm{c}} \\ 0 \end{pmatrix}}_{=b} \tag{5.5}$$

where the first row of the inequality system (5.5) belongs to requirement (5.1), the second row to (5.2), the third row to (5.3), and the last row to (5.4). Using the simplifications of Section 4.2, optimal CSS Ω^1 and Ω^2 can be computed for F_1^1 and F_1^2. Here, the CSS can be visualized either as geometric shapes in force space or as regions of permissible force-deformation characteristics in force-deformation space. This is investigated in more detail in Section 5.3. Exemplary, the corresponding independent CSS are visualized in Figure 45 for both visualization methods. Note that box- and arbitrarily-shaped CSS coincide in this example.

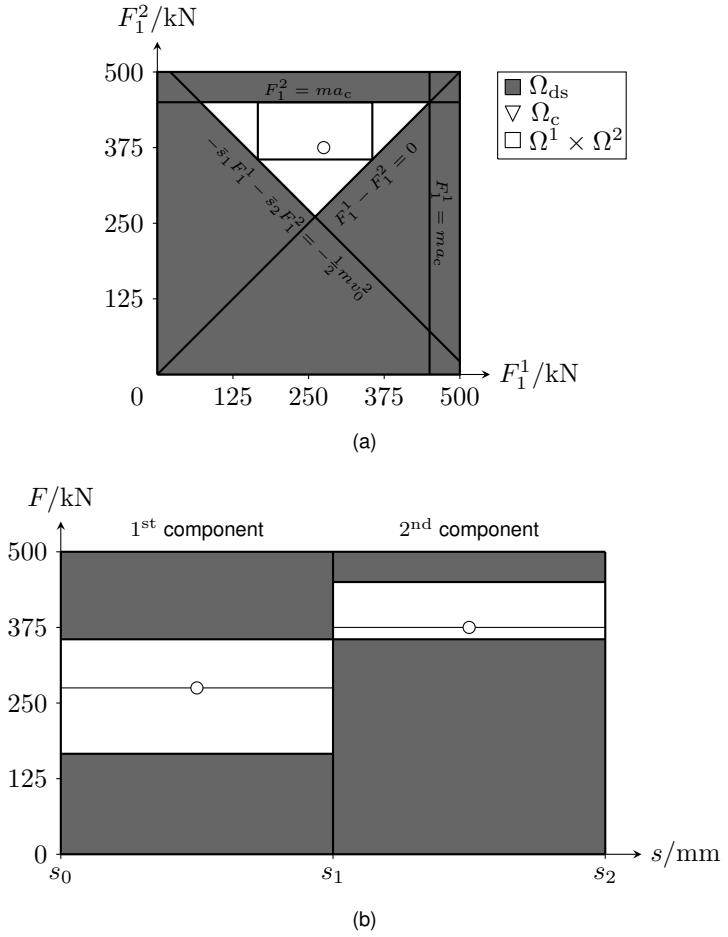


Figure 45 Optimal independent CSS as (a) geometric shapes in force space and (b) regions of permissible force-deformation characteristics for test-bed problem 1 with one degree of freedom per component and an example of a permissible design (white dot).

In Figure 45(a), the CSS $\Omega^1 = [F^{\mathrm{l},1}, F^{\mathrm{u},1}]$ and $\Omega^2 = [F^{\mathrm{l},2}, F^{\mathrm{u},2}]$ are displayed as their Cartesian product $\Omega = \Omega^1 \times \Omega^2$, i.e., $[F^{\mathrm{l}}, F^{\mathrm{u}}] = [F^{\mathrm{l},1}, F^{\mathrm{u},1}] \times [F^{\mathrm{l},2}, F^{\mathrm{u},2}]$, inside the complete system solution space Ω_{c}, which is defined by the requirements and the design space constraints. It can be seen that the computed lower bound $F^{\mathrm{l},1}$ is influenced by the minimum-energy-absorption requirement, the computed upper bound $F^{\mathrm{u},1}$ by the progressive-order-of-deformation requirement, the computed lower bound $F^{\mathrm{l},2}$ by both the minimum-energy-absorption and the progressive-order-of-deformation requirement, and the computed upper bound $F^{\mathrm{u},1}$ by the maximum-acceleration requirement.

In the following, not only force-deformation characteristics with one degree of freedom per component but two and more degrees of freedom are investigated. Corresponding box- and arbitrarily-shaped CSS are compared in Section 5.3. Furthermore, test-bed problem 1 is used to compute CSS under interval- and fuzzy-type uncertainty in Section 5.4.

5.1.3. Deformation Space Models

In order to model a vehicle front structure in more detail, more general DSM can be used. In [43], DSM were introduced as an extension of the simple crash model from [140]. First, n crash-relevant, structural components that shall be designed are determined. Amongst others, examples may include front rails, crash boxes, aprons, the firewall, and the front bumper. Here, the longitudinal lengths of the components must be selected a priori. It is assumed that the components have certain deformation lengths $\bar{s}^k \in \mathbb{R}^+$, $k = 1, \ldots, n$, under impact and behave as if they were rigid afterward. This means that the longitudinal length of every component can be represented by the sum of a deformable and a non-deformable length. Like above, the assumption for $\bar{s}^k$, $k = 1, \ldots, n$, in the early design stage can be either made without deeper knowledge or based on knowledge from similar vehicles that are already developed. A way to get knowledge-based assumptions is proposed in Section 5.2.

In order to obtain a DSM, a *geometry space model* (GSM) of the vehicle must be built first. Whereas DSM show the deformation space of the components, GSM are based on their geometric lengths. Therefore, the components are assigned to one of $n_{\mathrm{lp}} \in \mathbb{N}$ load paths and positioned in the geometry space in longitudinal impact direction for GSM. This is done in accordance with their positions within the vehicle's front structure. If the right side of the vehicle front structure is symmetrical to its left side, i.e., there are identical components which exist twice at the same position in the geometry space, these components can be represented by the same component in GSM and later in DSM. If the load paths contain further rigid parts which do not deform and are not part of the components, these parts are also included in GSM.
The mass of the rear vehicle is lumped behind the components across all load paths and the mass of the vehicle front is lumped as $n_{\mathrm{m}} - 1 \in \mathbb{N}$ discrete mass points behind the components on the load paths. The number of discrete mass points and their positions behind one component are assumed as well as the values of the masses $m_1, \ldots, m_{n_{\mathrm{m}}}$, which are uncontrollable parameters. An example of a GSM is visualized in Figure 46.

Using a GSM, the parts of the vehicle structure which deform simultaneously after hitting the barrier at full overlap can be identified. Here, it is assumed that the components fulfill a progressive order of deformation requirement and that the deformation rate is constant across all load paths at any position in the deformation space. Hence, the parts which deform simultaneously can be aligned vertically in deformation space. By further removing non-deformable parts and possible empty spaces from the load paths and stacking the deformable parts, a DSM is yielded. The DSM that corresponds to the GSM in Figure 46 is shown in this Figure as well.

The deformation space is divided into n_{s} sections where a new section always starts at

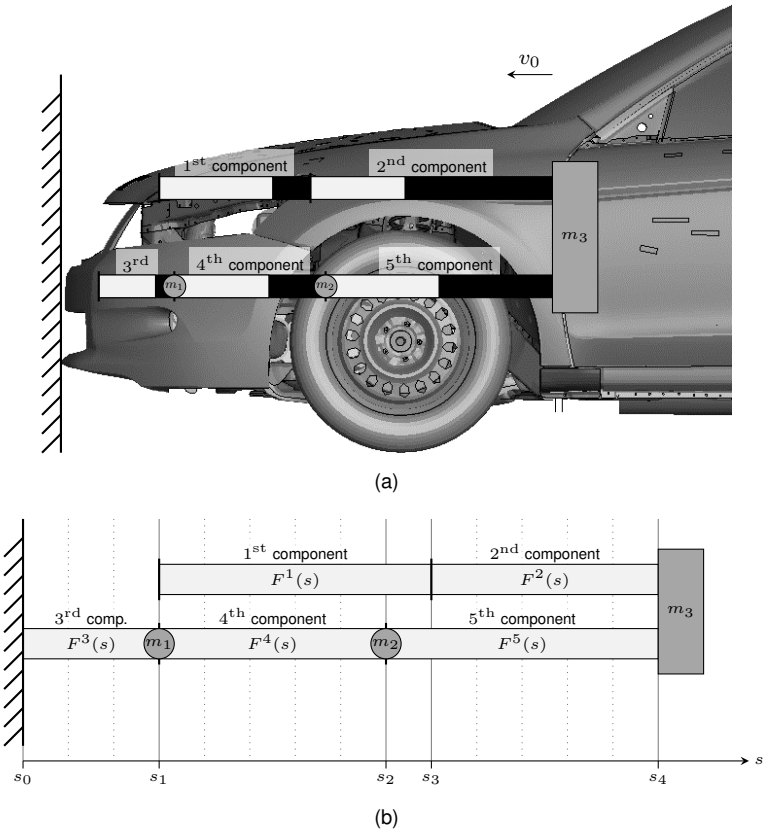


Figure 46 Vehicle front structure models (a) GSM with underlying vehicle chassis and (b) corresponding DSM (stretched) with five components, two load paths, and three discrete masses. There are deformable (gray) and non-deformable lengths (black).

the beginning of the deformation of each component. The last section ends when the front structure has deformed. Hence, there are four sections in the DSM example in Figure 46. In general, section 1 can be defined as the half-open interval $[s_0, s_1)$, section 2 as the half-open interval $[s_2, s_3)$, and so on until section n_s with $[s_{n_\mathrm{s}-1}, s_{n_\mathrm{s}})$. Like above, the position s_{n_s} is also referred to as s_end. The section lengths are $\bar{s}_1, \ldots, \bar{s}_{n_\mathrm{s}}$. Representatively, these are considered as the uncontrollable parameters for the deformation lengths of DSM in the following.

The position s_i, $i \in \{0, \ldots, n_\mathrm{s}\}$ where the deformation of the k^{th} component, $k \in \{1, \ldots, n\}$, starts is also denoted as s_0^k. Similarly, the position s_i, $i \in \{0, \ldots, n_\mathrm{s}\}$ where the deformation of the k^{th} component ends is denoted as s_end^k. For the deformation lengths of the components, it holds $\bar{s}^k = s_\mathrm{end}^k - s_0^k$, $k = 1, \ldots, n$.

Using these definitions, the force-deformation characteristics of the components can be defined via $F^k(s) \in [F_\mathrm{ds}^{\mathrm{l},k}, F_\mathrm{ds}^{\mathrm{u},k}]$ if $s \in [s_0^k, s_\mathrm{end}^k)$ and $F^k(s) = 0\,\mathrm{kN}$ otherwise, $k = 1, \ldots, n$. Here, $F_\mathrm{ds}^{\mathrm{l},k} \in \mathbb{R}_0^+$ is the minimum and $F_\mathrm{ds}^{\mathrm{u},k} \in \mathbb{R}_0^+$ the maximum force that $F^k(s)$ can assume in $[s_0^k, s_\mathrm{end}^k)$ with $F_\mathrm{ds}^{\mathrm{l},k} \leq F_\mathrm{ds}^{\mathrm{u},k}$, $k = 1, \ldots, n$. Furthermore, the sum of forces over all load paths, i.e.,

$$F(s) = \sum_{k=1}^{n} F^k(s) \tag{5.6}$$

at a position $s \in [s_0, s_\mathrm{end})$ represents a joint force-deformation characteristic of all crash-

relevant components.

During deformation, a discrete mass point m_l, $l \in \{1, \dots, n_\mathrm{m}\}$, located at position s_i, $i \in \{0, \dots, n_\mathrm{s} - 1\}$ in deformation space is only accelerated for $s < s_i$ where it is called active. For $s \geq s_i$, the discrete mass point m_l has hit the barrier and has been brought to an abrupt halt. Note that an elastic rebound is not considered in the DSM approach. Thus, $m_l^*(s) = m_l$ for $s < s_i$ and $m_l^*(s) = 0\,\mathrm{kg}$ otherwise indicates if m_l is active at $s \in [s_0, s_{n_\mathrm{s}})$. The total *active mass of the vehicle* at $s \in [s_0, s_{n_\mathrm{s}})$ can then be computed as

$$m^*(s) = m_{n_\mathrm{m}} + \sum_{l=1}^{n_\mathrm{m}-1} m_l^*(s). \tag{5.7}$$

Moreover, a discrete mass point m_l, $l \in \{1, \dots, n_\mathrm{m} - 1\}$ that is located between the k^th and $(k+1)^\mathrm{th}$ component in the same load path is also denoted by m^k, $k \in \{1, \dots, n-1\}$.

Using these definitions, the requirements on the minimum energy absorption, the maximum acceleration, and the progressive order of deformation can be formulated mathematically as follows, see Section A.3, which generalizes the requirements (5.1)-(5.4) from above:

- *Minimum energy absorption*: The vehicle must be brought to halt during the deformation. This means that its kinetic energy must be absorbed. Considering the change in active mass, it must hold that the integral of the sum of deformation forces over all load paths divided by the active mass from $s = s_0$ to $s = s_{n_\mathrm{s}}$ is smaller or equal to $\frac{1}{2}v_0^2$, i.e.,

$$-\int_{s_0}^{s_\mathrm{end}} \frac{F(s)}{m^*(s)}\,\mathrm{d}s \leq -\frac{1}{2}v_0^2. \tag{5.8}$$

- *Maximum acceleration*: The acceleration must be smaller than the critical acceleration threshold a_c at any position in deformation space. Here, the acceleration is the sum of deformation forces over all load paths divided by the active mass. Thus,

$$\frac{F(s)}{m^*(s)} \leq a_\mathrm{c} \tag{5.9}$$

  must hold for all $s \in [s_0, s_\mathrm{end})$.

  Progressive order of deformation: The ordered deformation of the vehicle must start at the front. This means that the force which is necessary to deform the k^th component is smaller or equal to the deformation force which is necessary to start the deformation of the $(k+1)^\mathrm{th}$ component for all $s \in [s_0^k, s_\mathrm{end}^k)$ in consideration of the local inertia force for m^k. Therefore,

$$F^k(s) - m^k \frac{F(s)}{m^*(s)} \leq F^{k+1}(s_0^{k+1}) \tag{5.10}$$

must hold for all $s \in [s_0^k, s_{\text{end}}^k)$ if the k^{th} and $(k+1)^{\text{th}}$ component share the same load path, $k = 1, \ldots, n-1$.

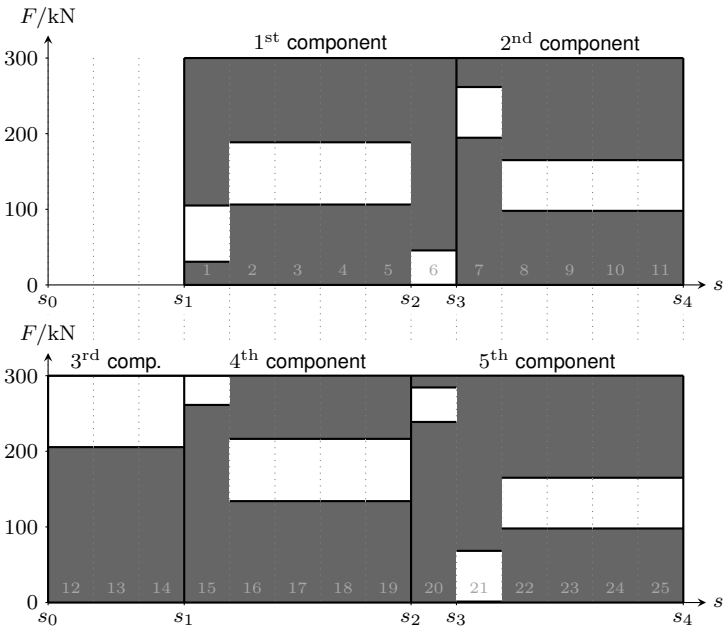
If piece-wise constant functions are considered as parametrized force-deformation characteristics, the requirements (5.9)-(5.10) simplify to a system of linear inequalities similarly to (5.5). In the following, an example of computing optimal box-shaped CSS for the components is considered for which the complete system solution space is based on the above constraints. Given the DSM of a vehicle front structure from Figure 46 with five components and three discrete masses, two cases are regarded:

(a) The masses are given by $m_1 = 200\,\text{kg}$, $m_2 = 250\,\text{kg}$, and $m_3 = 1100\,\text{kg}$, the initial velocity by $v_0 = 15.6\,\frac{\text{mm}}{\text{ms}}$ and the critical acceleration by $a_c = 0.3\,\frac{\text{mm}}{\text{ms}^2}$. The deformation lengths of the components are given by $\bar{s}^1 = 300\,\text{mm}$, $\bar{s}^2 = 250\,\text{mm}$, $\bar{s}^3 = 150\,\text{mm}$, $\bar{s}^4 = 250\,\text{mm}$, and $\bar{s}^5 = 300\,\text{mm}$, see Figure 46 in which the deformation lengths of components are colored gray.

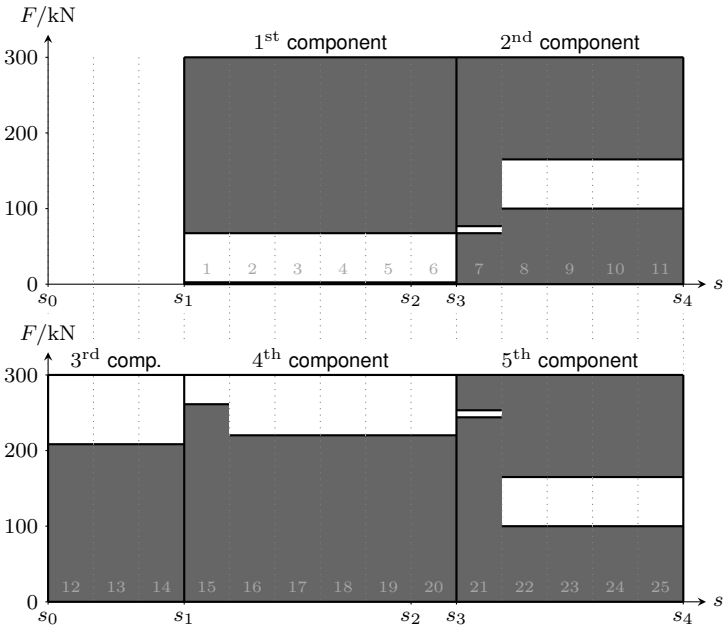
(b) The parameters are given similarly to (a) but $\bar{s}^4$ is increased by $50\,\text{mm}$ to $\bar{s}^4 = 300\,\text{mm}$. Hence, it is sufficient to use $\bar{s}^5 = 250\,\text{mm}$ as the last $50\,\text{mm}$ of the 5^{th} component will not deform then.

The force-deformation characteristics are modeled by piece-wise constant functions for which a piece-wise constant segment has a length of $50\,\text{mm}$. Hence, there are 25 constant force levels, i.e., 25 design variables, which are assigned to the components in lexicographical order from left to right, starting at the upper load path. Furthermore, there are one linear inequality for (5.8), 14 linear inequalities for (5.9), and 14 inequalities for (5.10) in case (a) and 15 in case (b), i.e., 29 linear inequalities in total in case (a) and 30 in case (b). The box-shaped CSS can be computed using the methods from Section 4.2 and the solutions for cases (a) and (b) are visualized in Figure 47.

Although only the deformation length of the 4^{th} component was increased by $50\,\text{mm}$, the changes in the box-shaped CSS for the other components are significant. In particular, the intervals of the design variables 7 and 21, are very different, both in size and location. As a result, there are changes in the development of these components.
These circumstances give rise to use more realistic assumptions for building DSM. Furthermore, it is assumed for DSM that only the crash-relevant components, in particular their force-deformation characteristics, determine the performance of the vehicle, which is in general not the case. These drawbacks of DSM are improved in the subsequent section.



(a)



(b)

Figure 47 Optimal independent box-shaped CSS visualized as regions of permissible force-deformation characteristics for cases (a) and (b) in which the deformation lengths of the 4^{th} component are different.

5.2. Enhanced Deformation-Space Models

This section derives enhanced deformation space models. First, knowledge-based DSM models which are based on realistic data for deformation lengths and masses are considered. As they do not represent the energy absorption and acceleration of the overall vehicle correctly, energy- and acceleration-corrected DSM are introduced and a further test-bed problem is defined. Furthermore, possible epistemic uncertainties for energy- and acceleration-corrected

DSM are considered.

5.2.1. Knowledge-based DSM

For knowledge-based DSM, realistic information about the vehicle's front structure, including its mass distribution and deformation lengths of its components, is used. In this thesis, the focus is put on building a knowledge-based DSM based on measured deformation lengths and masses of a single vehicle that is already fully developed. Such a knowledge-based DSM might help, for example, to improve the development and the crash performances of the vehicle type's next generation. Other than that, also realistic data from multiple vehicles can be combined to get knowledge-based DSM for developing new vehicle types. Before a knowledge-based DSM can be obtained, a corresponding GSM must be built. Therefore, the following steps are proposed:

(a) *Identification of crash-relevant components, load paths, and lumped masses*: Crash-relevant components are usually structural entities like crash boxes or front rails. Sometimes, these entities can also be defined as several components, for example, due to changing shapes or material properties in the longitudinal direction or because multiple designers work on different parts of these entities. This might be the case for large front rails. Crash-relevant components are usually arranged in distinct load paths. In these, the load is passed through connected components. Components that connect different load paths are not considered in this thesis. However, they are also conceivable in general, see [85]. Besides the components, lumped masses are assigned to the load paths, too. These masses originate from heavy parts of the vehicle like the engine. Furthermore, the mass of the rear vehicle is lumped behind all components.

(b) *Measurement of lumped masses, the geometric lengths of the components, and their deformation lengths in the longitudinal impact direction*: In order to get the values for the lumped masses, the corresponding parts are weighted. The longitudinal lengths of the components are measured before and after the crash in the longitudinal impact direction. Considering the loading and unloading of the components, it is assumed that the differences between these lengths yield the lengths of plastic deformation. Note that the measured longitudinal lengths after the crash depend on the impact velocity of the vehicle as discussed above. Thus, these lengths must be considered as the non-deformable lengths of the components for a certain impact velocity. Only if the velocity is large enough and the lengths do not change for larger velocities, they can be considered as the maximum non-deformable lengths of the components.

(c) *Arrangement of further possible non-deformable lengths in the load paths (if applicable)*: Besides the non-deformable lengths of the components that represent the non-deformable parts in the GSM, there might be further non-deformable parts which do not belong to the components. If these are part of the load paths, they must be included in the GSM to obtain DSM for which parts of simultaneous deformation are aligned vertically. Like non-deformable lengths of the components, these lengths can also be measured after the

crash.

In this thesis, relevant crash properties are gathered from analyzing a HONDA ACCORD FE model taken from [98] and running a crash simulation of this model in the software LS-DYNA at full overlap with an initial velocity of $v_0 = 15.6\,\frac{\text{mm}}{\text{ms}}$ like done in the US-NCAP crash test.
For this model, seven crash-relevant components are identified that shall be designed and that share two load paths, compare Figure 48. Here, the vehicle front structure is symmetrical and only components on the vehicle's left side are considered. In order to take also the right side into account, the force levels of the components on the left must be multiplied by two when using the corresponding GSM or DSM, and divided by two the other way around. Note that the front bumper as a crash-relevant component is not considered here as it spans transversely across the vehicle. Furthermore, there are four relevant masses which are lumped and distributed on the load paths, compare Figure 48. The mass m_1 represents the drivetrain, m_2 the radiator, m_3 the engine, and m_4 the remaining mass of the vehicle, which shall approximate the mass of the rear vehicle.
The measurements in LS-DYNA can be done according to Section A.3 for which the measured masses and deformation lengths for the HONDA ACCORD FE model are stated in Table 4. Note that these values are non-deterministic, i.e., there is uncertainty in these values, which is investigated later in this section.

Table 4 Measured deformation lengths in the longitudinal impact direction of the seven crash-relevant components and measured masses of the HONDA ACCORD FE model.

quantity	$\bar{s}^1$	$\bar{s}^2$	$\bar{s}^3$	$\bar{s}^4$	$\bar{s}^5$	$\bar{s}^6$	$\bar{s}^7$
value	$313\,\text{mm}$	$156\,\text{mm}$	$78\,\text{mm}$	$54\,\text{mm}$	$101\,\text{mm}$	$150\,\text{mm}$	$270\,\text{mm}$

quantity	m_1	m_2	m_3	m_4
value	$104\,\text{kg}$	$19\,\text{kg}$	$277\,\text{kg}$	$1250\,\text{kg}$

Besides the non-deformable lengths of the components, there is a further non-deformable length in the second load path behind the seventh component which is considered in the GSM, see Figure 48. From the GSM, the knowledge-based DSM for the HONDA ACCORD FE model can be derived as before. It is visualized in Figure 48.

For knowledge-based DSM, the same system performance functions as for general DSM, i.e., the underlying performance functions of requirements (5.1)-(5.4), hold. However, when measuring, for example, the acceleration or energy absorption of the overall HONDA ACCORD FE model in the software LS-DYNA, significant differences with the computed system responses from the force-deformation characteristics of the components exist. This is mainly because not all components in the frontal vehicle structure are considered and that the DSM only considers the longitudinal direction, which does not account for the real energy absorption in three

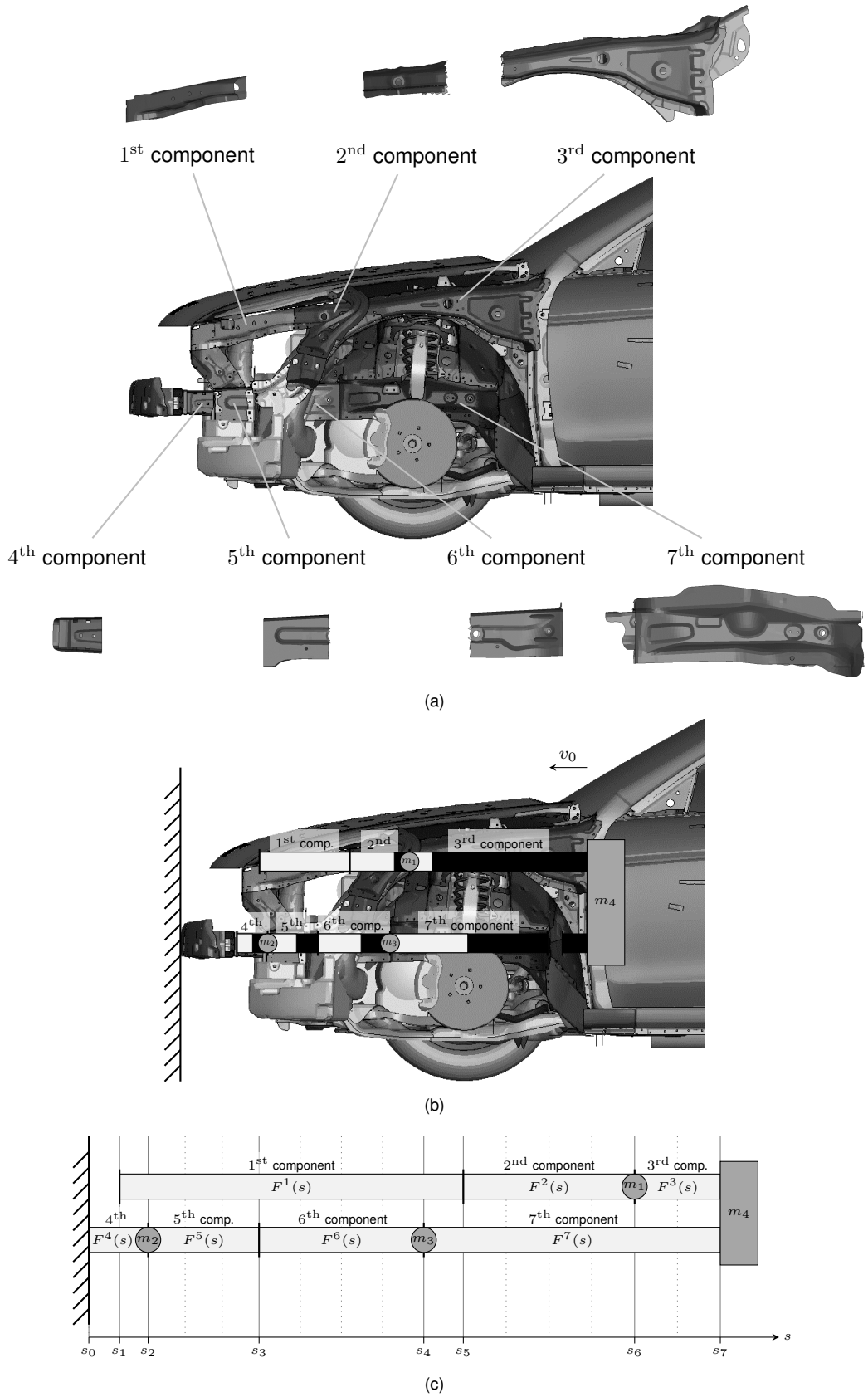


Figure 48 The HONDA ACCORD FE model's (a) crash-relevant components (b) GSM, and (c) DSM with deformable (gray) and non-deformable lengths (black). Note that a different scale for GSM and DSM is used here.

dimensions. In order to circumvent also these drawbacks, energy- and acceleration-corrected DSM are introduced.

5.2.2. Energy- and Acceleration-Corrected DSM

For energy- and acceleration-corrected DSM, measured force-deformation characteristics in the longitudinal impact direction are used for calibration. In Section A.3, an approach is described, how they can be obtained from an FEM simulation of the full HONDA ACCORD FE model using the software LS-DYNA. As an example, the corresponding force-deformation characteristic of the fourth component is shown in Figure 50.

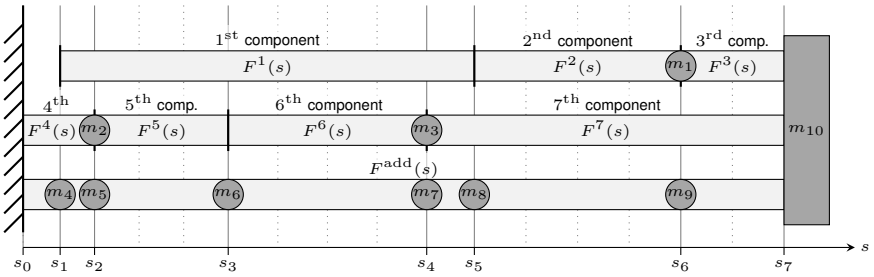


Figure 49 DSM of the HONDA ACCORD FE model with an additional load path and additional masses as a representation of the remaining parts of its front structure.

The underlying assumption of DSM is that $F(s)$, computed from Equation (5.6), over the active mass $m^*(s)$ is a good estimation for the acceleration of the rear vehicle at any deformation position in deformation space. However, as usually more parts of the frontal structure than the ones represented in the DSM are responsible for the acceleration of the vehicle, the estimated acceleration is in general too low and must be corrected. This can be done by calibrating a force-deformation characteristic that is integrated into an additional load path, see Figure 49. Its force values are denoted by $F^{\mathrm{add}}(s)$ and represent the force in the longitudinal impact direction on the remaining parts of the front structure which are not considered in the DSM for $s \in [s_0, s_{\mathrm{end}})$. The force-deformation characteristic of the additional load path can be obtained from calibration at the acceleration of the rear vehicle, as described in Section A.3.
Furthermore, the values of the active mass $m^*(s)$ at $s \in [s_0, s_{\mathrm{end}})$ can be enhanced by assigning the remaining masses of the frontal structure to the additional load path. They must be lumped at $s \in \{s_0, \ldots, s_{\mathrm{end}}\}$ by considering Equation (5.7), which can be done according to Section A.3. These masses are visualized in Figure 49 as well.

Using these perceptions, the requirement on the maximum acceleration, stated in Equation (5.9), then reads

$$\frac{F(s) + F^{\mathrm{add}}(s)}{m^*(s)} \leq a_{\mathrm{c}} \tag{5.11}$$

for all $s \in [s_0, s_{\mathrm{end}})$. Note that the HONDA ACCORD FE model violates the progressive order of deformation requirement. Thus, using $F(s)$, computed by Equation (A.37), and $F^{\mathrm{add}}(s)$, calibrated as proposed in Section A.3, does not exactly represent the acceleration of the rear

vehicle for $s \in [s_0, s_{\text{end}})$. This changes when the progressive order requirement is fulfilled. The corresponding requirement, previously given by Equation (5.10), can here be formulated as

$$F^k(s) - m^k \frac{F(s) + F^{\text{add}}(s)}{m^*(s)} \leq F^{k+1}(s_0^{k+1}) \tag{5.12}$$

for all $s \in [s_0^k, s_{\text{end}}^k)$ if the k^{th} and $(k+1)^{\text{th}}$ component share the same load path, $k = 1, \ldots, n-1$.

A further assumption of DSM is that if the force-deformation characteristic which represents all crash-relevant components is inserted into requirement (5.8), the left-hand side of (5.8) approximates the right-hand side. This means that the sum of component forces integrated over their deformation lengths is a good estimation of the total absorbed energy of the vehicle. However, there is an error due because there are remaining parts of the vehicle which are not considered in the DSM, i.e., their energy absorption is not included in requirement (5.8). This can be circumvented by integrating the forces of the additional load path over the deformation length of the load paths, too.
Moreover, there is a further error because the DSM only considers energy absorption in the longitudinal direction. In order to take the real energy absorption in three dimensions into account, energy-correction factors are introduced for all components, denoted by E_{cf}^k, $k = 1, \ldots, n$, and for the additional load path, denoted by $E_{\text{cf}}^{\text{add}}$. Here, it is assumed that the energy absorbed in three dimensions is proportional to the energy absorbed in the longitudinal impact direction. Thus, E_{cf}^k can be calibrated at the force-deformation characteristics of the components and the total internal energy of the components E_{tot}^k, $k = 1, \ldots, n$. It holds

$$E_{\text{cf}}^k = \frac{E_{\text{tot}}^k}{\int_{s_0^k}^{s_{\text{end}}^k} F^k(s)\,\mathrm{d}s}, \tag{5.13}$$

$k = 1, \ldots, n$. The natural condition $E_{\text{tot}}^k \geq \int_{s_0^k}^{s_{\text{end}}^k} F^k(s)\,\mathrm{d}s$ and therefore $E_{\text{cf}}^k \geq 1$ should always be fulfilled, $k = 1, \ldots, n$. Moreover, the energy-correction factor of the additional load path is calibrated such that

$$E_{\text{cf}}^{\text{add}} = \frac{\frac{1}{2}v_0^2 - \sum_{k=1}^{n} \left(E_{\text{cf}}^k \int_{s_0^k}^{s_{\text{end}}^k} \frac{F^k(s)}{m^*(s)}\,\mathrm{d}s \right)}{\int_{s_0}^{s_{\text{end}}} \frac{F^{\text{add}}(s)}{m^*(s)}\,\mathrm{d}s} \tag{5.14}$$

holds using the measured force-deformation characteristics of the components. Accordingly, the requirement on the minimum energy absorption, previously stated in requirement (5.8), then reads

$$-\sum_{k=1}^{n} \left(E_{\text{cf}}^k \int_{s_0^k}^{s_{\text{end}}^k} \frac{F^k(s)}{m^*(s)}\,\mathrm{d}s \right) - E_{\text{cf}}^{\text{add}} \int_{s_0}^{s_{\text{end}}} \frac{F^{\text{add}}(s)}{m^*(s)}\,\mathrm{d}s \leq -\frac{1}{2}v_0^2. \tag{5.15}$$

This means that the calibration of $E_{\text{cf}}^{\text{add}}$ guarantees an equal sign in requirement (5.15) for the measured force-deformation characteristics of the components of the HONDA ACCORD

FE model. Summarized, the requirements for energy- and acceleration-corrected DSM are the requirements on minimum energy absorption (5.15), maximum acceleration (5.11), and progressive order of deformation (5.12).

In addition to the uncontrollable parameters in the requirements of non-corrected DSM v_0, a_c, $\bar{s}_i$, $i = 1, \ldots, n_\mathrm{s}$, and m_l, $l = 1, \ldots, n_\mathrm{m}$, uncontrollable parameters in the requirements for energy- and acceleration-corrected DSM are additionally formed by the force-deformation characteristic of the additional load path, i.e., $F^\mathrm{add}(s)$ for $s \in [s_0, s_\mathrm{end})$ and the energy-correction factors E_cf^k, $k = 1, \ldots, n$, and $E_\mathrm{cf}^\mathrm{add}$. The constraint functions that belong to the requirements are component-wise monotonic in the uncontrollable parameters p_l, $l = 1, \ldots, q$, as it holds

$$\frac{\mathrm{d}\left(-\sum_{k=1}^{n}\left(E_\mathrm{cf}^k \int_{s_0}^{s_\mathrm{end}} \frac{F^k(s)}{m^*(s)}\,\mathrm{d}s\right) - E_\mathrm{cf}^\mathrm{add} \int_{s_0}^{s_\mathrm{end}} \frac{F^\mathrm{add}(s)}{m^*(s)}\,\mathrm{d}s + \frac{1}{2}v_0^2\right)}{\mathrm{d}p_l}$$

$$\begin{cases} \geq 0 & \text{if } p_l \in \{v_0, m_1, \ldots, m_{n_\mathrm{m}}\}, \\ \leq 0 & \text{if } p_l \in \{s_1, \ldots, s_\mathrm{end}, E_\mathrm{cf}^1, \ldots, E_\mathrm{cf}^n, E_\mathrm{cf}^\mathrm{add}\} \cup \{F^\mathrm{add}(s) \mid s \in [s_0, s_\mathrm{end})\}, \end{cases} \tag{5.16}$$

$$\frac{\mathrm{d}\left(\frac{F(s)+F^\mathrm{add}(s)}{m^*(s)} - a_\mathrm{c}\right)}{\mathrm{d}p_l} \begin{cases} \geq 0 & \text{if } p_l \in \{F^\mathrm{add}(s)\}, \\ \leq 0 & \text{if } p_l \in \{a_\mathrm{c}, m_1, \ldots, m_{n_\mathrm{m}}\} \end{cases} \tag{5.17}$$

for all $s \in [s_0, s_\mathrm{end})$, and

$$\frac{\mathrm{d}\left(F^k(s) - m^k \frac{F(s)+F^\mathrm{add}(s)}{m^*(s)} - F^{k+1}(s_0^{k+1})\right)}{\mathrm{d}p_l}$$

$$\begin{cases} \geq 0 & \text{if } p_l \in \{m_1, \ldots, m_{n_\mathrm{m}}\} \backslash \{m^k\}, \\ \leq 0 & \text{if } p_l \in \{m^k, F^\mathrm{add}(s)\} \end{cases} \tag{5.18}$$

for all $s \in [s_0^k, s_\mathrm{end}^k)$ if the k^th and $(k+1)^\mathrm{th}$ component share the same load path, $k = 1, \ldots, n-1$, $l \in \{1, \ldots, q\}$. However, they are not monotonic in the uncontrollable parameters as they are, for example, both component-wise monotonically increasing and decreasing in m_l, $l = 1, \ldots, m_{n_\mathrm{m}}$. These properties will be used for DSM under uncertainties in uncontrollable parameters later. Moreover, they transfer immediately to non-corrected DSM including test-bed problem 1 which are special cases of energy- and acceleration-corrected DSM.
Subsequently, a new test-bed problem representing the energy- and acceleration-corrected DSM of the HONDA ACCORD FE model is defined.

5.2.3. Test bed 2: Realistic Crash Design Problem

In general, the values of the uncontrollable parameters for the energy- and acceleration-corrected DSM of the HONDA ACCORD FE model can be obtained by following the instructions above. In Table 5, the values of the section lengths and discrete mass points are stated. The initial velocity of the HONDA ACCORD FE model in the LS-DYNA simulation was chosen as $15.6\,\frac{\mathrm{mm}}{\mathrm{ms}}$. However, as there are parts of the frontal structure that deform and accelerate the vehicle before the considered load paths start deforming, the velocity at the beginning of the load paths' deformation is chosen here. Its value is specified in Table 5. Furthermore, the critical acceleration represents a threshold, i.e., a predetermined value, which shall not be exceeded. Its value is chosen as above and is also stated in Table 5.

Table 5 Section lengths, discrete mass points, initial velocity, and critical acceleration for the DSM of the HONDA ACCORD FE model.

quantity	$\bar{s}_1$	$\bar{s}_2$	$\bar{s}_3$	$\bar{s}_4$	$\bar{s}_5$	$\bar{s}_6$	$\bar{s}_7$
value	$28\,\mathrm{mm}$	$26\,\mathrm{mm}$	$101\,\mathrm{mm}$	$150\,\mathrm{mm}$	$36\,\mathrm{mm}$	$156\,\mathrm{mm}$	$78\,\mathrm{mm}$
quantity	m_1	m_2	m_3	m_4	m_5	m_6	m_7
value	$104\,\mathrm{kg}$	$19\,\mathrm{kg}$	$277\,\mathrm{kg}$	$18\,\mathrm{kg}$	$3\,\mathrm{kg}$	$26\,\mathrm{kg}$	$26\,\mathrm{kg}$
quantity	m_8	m_9	m_{10}	v_0	a_c		
value	$17\,\mathrm{kg}$	$16\,\mathrm{kg}$	$1144\,\mathrm{kg}$	$15.0\,\frac{\mathrm{mm}}{\mathrm{ms}}$	$0.3\,\frac{\mathrm{mm}}{\mathrm{ms}^2}$		

Before the uncontrollable parameters regarding the additional load path and the energy-correction factors are considered in more detail, the lower and upper bounds for the component design spaces are investigated. Here, the corresponding values are obtained by rounding up the maximum force values of components' associated load paths to the next $100\,\mathrm{kN}$. They are stated in Table 6.

Table 6 Design space parameters for the DSM of the HONDA ACCORD FE model.

quantity	$F_\mathrm{ds}^{\mathrm{l},1}$	$F_\mathrm{ds}^{\mathrm{l},2}$	$F_\mathrm{ds}^{\mathrm{l},3}$	$F_\mathrm{ds}^{\mathrm{l},4}$	$F_\mathrm{ds}^{\mathrm{l},5}$	$F_\mathrm{ds}^{\mathrm{l},6}$	$F_\mathrm{ds}^{\mathrm{l},7}$
value	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$
quantity	$F_\mathrm{ds}^{\mathrm{u},1}$	$F_\mathrm{ds}^{\mathrm{u},2}$	$F_\mathrm{ds}^{\mathrm{u},3}$	$F_\mathrm{ds}^{\mathrm{u},4}$	$F_\mathrm{ds}^{\mathrm{u},5}$	$F_\mathrm{ds}^{\mathrm{u},6}$	$F_\mathrm{ds}^{\mathrm{u},7}$
value	$100\,\mathrm{kN}$	$100\,\mathrm{kN}$	$100\,\mathrm{kN}$	$200\,\mathrm{kN}$	$200\,\mathrm{kN}$	$200\,\mathrm{kN}$	$200\,\mathrm{kN}$

In contrast to test-bed problem 1, test-bed problem 2 will be directly defined for parametrized force-deformation characteristics. These characteristics are necessary to compute CSS Ω^k, $k = 1, \ldots, 7$, for the components of the HONDA ACCORD FE model in this thesis. Again, piece-wise constant force-deformation characteristics, which are constant in any segment, are considered. The number of equidistant segments per section is defined in a way that all

segments have approximately the same length. Hence, one segment is chosen for the first, second, and fifth section, two for the seventh, three for the third, and four for the fourth and sixth.
When designing the force-deformation characteristics of the components, the constant force levels within these segments are their design variables. Their number corresponds to the degrees of freedom. Thus, there are two degrees of freedom for the third and fourth component, three for the fifth, four for the second and sixth, seven for the seventh, and nine for the first, which sums up to 31 degrees of freedom in total. Again, the design variables are denoted by F_i, $i = 1, \ldots, 31$, and are assigned to the components in lexicographical order that goes from left to right and starts at the top, compare Figure 51. As an example, $F_{16} = F_1^4$ and $F_{17} = F_2^4$ with $d^4 = 2$ hold and the piece-wise constant force-deformation characteristic of the fourth component is given by the graph that belongs to $F^4(s) = F_{16}$ for $s \in [s_0, s_1)$, $F^4(s) = F_{17}$ for $s \in [s_1, s_2)$, and $F^4(s) = 0$ otherwise. Here, $(F_{16}, F_{17}) \in [0\,\mathrm{kN}, 200\,\mathrm{kN}]^2$ must hold in order to fulfill the design space constraints.
However, when measuring, for example, the force-deformation characteristics of the components with the methods from above, it can be recognized that they are not parametrized, see Figure 50. Hence, two options are conceivable for incorporating measured force-deformation characteristics into CSS: either the measured force-deformation characteristics must be parametrized a posteriori such that it can be tested if they are contained in the CSS or the CSS must be transformed a posteriori such that they form constraints for the non-parametrized force-deformation characteristics. Both options are briefly discussed in the following.

In general, there are various methods to obtain piece-wise constant force-deformation characteristics from measured characteristics. In this thesis, a method that conserves the energy which is absorbed in the longitudinal impact direction is chosen. This can be reached by integrating the force over the deformation of a segment and dividing the integral by the segment length. Then, the resulting value is set as the constant force level for the corresponding segment. This method is visualized for the fourth component of the HONDA ACCORD FE model in Figure 50.
Note that the force-deformation characteristic in Figure 50 was measured at the left side of the vehicle. In order to take also the right side into account, the forces must be multiplied by two. Also, recall that the measured force-deformation characteristic represents only one possible design of the corresponding component and that the force levels can be modified by a designer. Other parametrization methods, which do not necessarily conserve the energy, select, for example, the force value that belongs to the center of the segment or the maximum or minimum force value to represent the constant force level of the segment.
All parametrization methods have in common that they usually involve errors which propagate to the system responses. For example, an energy-conservative parametrized force-deformation characteristic induces an error regarding acceleration as it does not represent the maximum forces well enough. Furthermore, there is an error in the force values that are necessary to start the components' plastic deformation. These facts must be considered when evaluating

parametrized force-deformation characteristics with the CSS. Thus, even a permissible force-deformation characteristic regarding a CSS might cause a violation of the system constraints. This problem can be circumvented by considering the deviations from the constant force levels as uncertainties in the controllable variables, compare the following subsection.

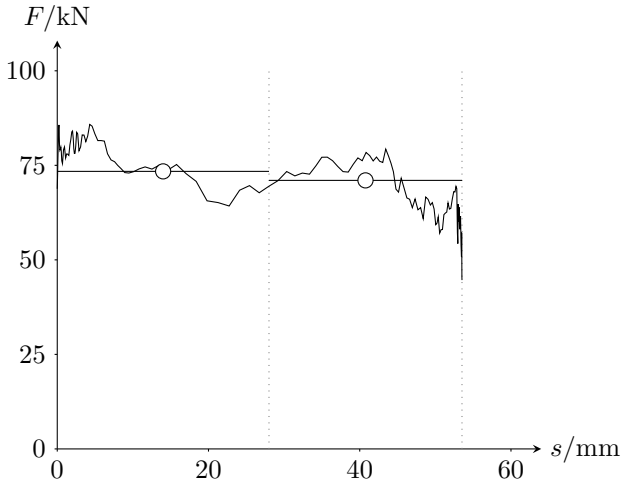


Figure 50 Measured (continuous force levels) and parametrized (constant force levels with white dots) force-deformation characteristic of the fourth component (left side only) of the HONDA ACCORD FE model for which the parametrization is done with respect to the segments of the underlying DSM (dotted lines).

Besides parametrizing the force-deformation characteristics, the CSS can be transformed into constraints for the non-parametrized characteristics as well. The intervals of box-shaped CSS can be used as point-wise lower and upper bounds for the non-parametrized characteristics within their corresponding segment, for example. Here, possible drawbacks are jumps for which the lower bound of a certain segment is above the upper bound of the previous or subsequent segment. This is, for example, the case for the CSS of the third component of the HONDA ACCORD FE model, visualized in Figure 51. Ideas on how to overcome these drawbacks are discussed in [43]. They can also be avoided by using piece-wise linear parametrized force-deformation characteristics, see [27]. Furthermore, a non-parametrized characteristic may violate the transformed lower bounds of CSS and still be permissible if the integrated force over the deformation length of the component is greater than or equal to the integrated lower bounds over the same length. Using arbitrarily-shaped CSS of the form (3.10) and assigning the computed component threshold values b^k to the component performance functions for the non-parametrized force-deformation characteristics, transformed CSS which circumvent this problem are obtained, $k = 1, \ldots, n$.

All in all, both options to incorporate measured force-deformation characteristics into computed CSS have their advantages and disadvantages, and it depends on the use-case to decide which option to choose. Whereas only the second option is discussed in [43], the first option is used in this thesis to focus on the property of CSS to provide optimal flexibility for component design with a finite number of degrees of freedom, i.e., non-transformed CSS.

In order to do so, also the force-deformation characteristic of the additional load path obtained by Equation (A.37) must be parametrized accordingly with respect to the segments of the section. Here, there are 16 segments, i.e., 16 constant force levels for the additional load path. They are labeled from left to right and their computed values are stated in Table 7. As the parametrized force-deformation characteristics are conservative regarding energy, the computation of the energy-correction factor for the components and the additional load path can be done using both the parametrized or the non-parametrized characteristics in Equations (5.13) and (5.14). This is not the case for general parametrized force-deformation characteristics. In Table 7, the corresponding values are stated.

Table 7 Force values of the additional load path and energy-correction factors of the HONDA ACCORD FE model.

quantity	F_1^{add}	F_2^{add}	F_3^{add}	F_4^{add}	F_5^{add}	F_6^{add}	F_7^{add}
value	$2\,\mathrm{kN}$	$15\,\mathrm{kN}$	$71\,\mathrm{kN}$	$116\,\mathrm{kN}$	$184\,\mathrm{kN}$	$170\,\mathrm{kN}$	$161\,\mathrm{kN}$
quantity	F_8^{add}	F_9^{add}	F_{10}^{add}	F_{11}^{add}	F_{12}^{add}	F_{13}^{add}	F_{14}^{add}
value	$148\,\mathrm{kN}$	$127\,\mathrm{kN}$	$143\,\mathrm{kN}$	$75\,\mathrm{kN}$	$124\,\mathrm{kN}$	$182\,\mathrm{kN}$	$238\,\mathrm{kN}$
quantity	F_{15}^{add}	F_{16}^{add}	E_{cf}^1	E_{cf}^2	E_{cf}^3	E_{cf}^4	E_{cf}^5
value	$242\,\mathrm{kN}$	$287\,\mathrm{kN}$	1.10	1.25	1.18	1.16	1.02
quantity	E_{cf}^6	E_{cf}^7	$E_{\mathrm{cf}}^{\mathrm{add}}$				
value	1.23	1.00	1.26				

Using the piece-wise constant force-deformation characteristics described above, the requirements (5.15), (5.11), and (5.12) for the energy- and acceleration-corrected DSM turn into linear inequalities for the force levels F_i, $i = 1, \ldots, 31$. There are one linear inequality for (5.15), 16 linear inequalities for (5.11), and 22 inequalities for (5.12), i.e., 39 linear inequalities in total. For the DSM of the HONDA ACCORD FE model which is not energy- and acceleration-corrected, the same number of constraints is yielded using the piece-wise constant force-deformation characteristics described above. In Figure 51, both the optimal box-shaped independent CSS for the non-corrected and corrected DSM of the HONDA ACCORD FE model are faced and the parametrized force-deformation characteristics of the components are included. Again, the box-shaped independent CSS are computed using the methods from Section 4.2. Note that the upper design space constraints for the non-corrected DSM are multiplied by two in order to compensate for the energy absorption of further parts in the frontal structure that are not considered here.

Although both non-corrected and energy- and acceleration-corrected DSM shall describe the same model, significant differences in the CSS are present, see Figure 51. Both the lower and upper bounds of the non-corrected DSM are much higher than the corresponding bounds of the corrected DSM. This is due to the significant influence of the additional load path in the

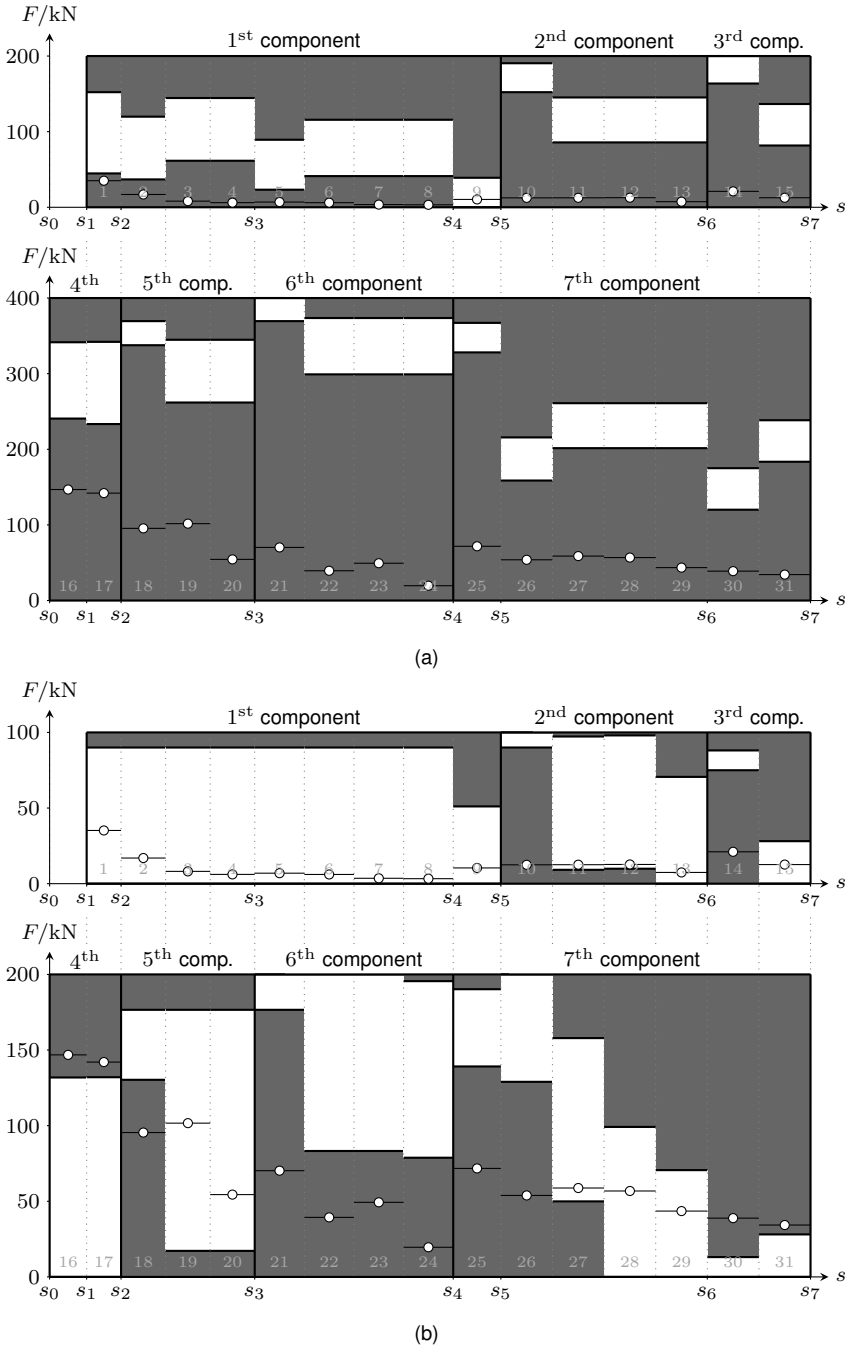


Figure 51 Optimal box-shaped independent CSS visualized as regions of permissible force-deformation characteristics for the non-corrected (a) and energy- and acceleration-corrected (b) DSM of the HONDA ACCORD FE model and parametrized force-deformation characteristics of its components.

corrected model, which induces that the components must participate less in the acceleration of the vehicle and absorb less energy. The effect on the lower bounds is even reinforced by the energy-correction factors, which account for energy absorption in three dimensions. The local influence of the additional load path on the upper bounds can especially be seen for the sixth and seventh component.

Furthermore, it can be concluded from Figure 51 that the force levels of the parametrized force-deformation characteristics from the components of the HONDA ACCORD FE model are

generally too low for non-corrected DSM although the required energy absorption is fulfilled in the simulation. This is different for energy- and acceleration-corrected DSM, which circumvent this drawback. Hence, only energy- and acceleration-corrected DSM are considered in the following to yield realistic CSS. Note that there are also force levels of the parametrized force-deformation characteristics of the HONDA ACCORD FE model that are outside the bounds of the CSS in Figure 51. Here, the progressive order of deformation requirement is violated. Thus, the force levels can be improved for the vehicle's next generation using the provided box-shaped CSS or different CSS types of this thesis, for example. Furthermore, test-bed problem 2, representing the energy- and acceleration-corrected DSM for the HONDA ACCORD FE model, is used to compute various types of CSS including CSS under epistemic uncertainty. In order to do so, epistemic uncertainty is considered for enhanced DSM subsequently.

5.2.4. Enhanced DSM under Epistemic Uncertainty

There are various reasons for uncertainty in crash design. This subsection focuses on uncertainty that occurs when using DSM or rather energy- and acceleration-corrected DSM. Here, uncertainty is classified from a system point of view, like done in Section 2.1. As DSM provide only constraints but no cost function, it is only differentiated between uncertainties in controllable variables, uncontrollable parameters, and constraints. The following list provides an overview of possible reasons for the occurrence of these uncertainties:

(a) *Uncertainties in controllable variables*: The design variables for DSM describe the force-deformation characteristics of the considered, crash-relevant components. These force-deformation characteristics are responses at level 2 of a three-level system and are controlled indirectly by modifying the components' design models and their design variables at level 1, compare Section 5.1. This means that the uncertainties at level 1 propagate to the force-deformation characteristics, i.e., the controllable variables at level 2. In general, these multiple sources of uncertainty complicates the uncertainty quantification for the force-deformation characteristics. Moreover, DSM can be even used if the component models are undefined in the early design phase. Thus, the result of the propagated uncertainty to the force-deformation characteristics is not known and must be assumed in a no-more-knowledge state, see Section 2.1. If in contrast a completely or almost completely developed vehicle is considered, the corresponding uncertainty propagation can possibly be reduced to the propagation of uncertainty in material or geometric properties.
Although it is assumed in this thesis that the component design models which map the design variables at level 1 to the force-deformation characteristics at level 2 are surjective with respect to the design spaces of the force-deformation characteristics, this might not always be the case in reality. Using the maximum metric, the distance of all target force-deformation characteristics to their next realizable characteristics can be considered as the uncertainty magnitudes of the controllable variables. Moreover, if parametrized characteristics that shall represent measured characteristics are used, e.g., in an FEM simulation like done above, there is usually an error which propagates to the responses of the crash system. This error depends on the parametrization and can also be considered

as uncertainty in controllable variables.

As discussed in Section 5.1, also different initial conditions of the vehicle before the impact and different interaction modes with the remaining frontal vehicle structure can lead to different force-deformation characteristics of the components, i.e., they depend on boundary conditions. If these conditions are changed, different deformation modes of the components, which imply a sustainable change in the force-deformation characteristics of the components, may occur. This uncertainty occurs, for example, when there are variations in these conditions or when measuring forces-deformation characteristics with simplified boundary conditions, e.g., in a drop tower test.

(b) *Uncertainties in uncontrollable parameters*: The parameters that cannot be controlled by designers using DSM are the section lengths, the lumped masses, the initial velocity, and the critical acceleration for the non-corrected DSM. For energy- and acceleration-corrected DSM, the force-deformation characteristic of the additional path and the energy-correction factors add to these parameters. If the values for these parameters are gathered, for example, from an FEM simulation, there are usually measurement errors, which can be considered as uncertainties in uncontrollable parameters. If these parameters must be assumed in the early phase, uncertainty in these parameters is also present. In both cases, the uncertainty is hard to quantify and must often be assumed.

Like the controllable variables, the uncontrollable parameters also depend on the boundary conditions. Here, this might directly affect the masses, and the initial velocity, and indirectly affect deformation length, the additional load path, and the correction factors. In addition, also the critical acceleration might be considered as uncertain, as it shall represent the critical acceleration on specific occupant parts, which might be different.

(d) *Uncertainties in constraints*

In addition to the design space constraints, constraints for crash design using DSM are the requirements on the energy absorption, the critical acceleration, and the progressive order of deformation. When building enhanced DSM from realistic crash properties of a completely developed vehicle, see test-bed problem 2, there are model errors that arise from using DSM as a simplified, low-fidelity model instead of the high-fidelity model that supplied the crash properties. Here, possible changes to the high-fidelity during development are further sources of uncertainty. In addition, the model error also depends on the details of the simplified model, like its assumed mass distribution and the alignment of simultaneously deforming parts in different load paths. For DSM built from assumed crash properties, high-fidelity models and therefore knowledge about model errors are not available. However, when corresponding high-fidelity models are built a posteriori, these errors occur, too. Note that in general, there is also uncertainty in higher-level models, e.g., when using FE models for crash tests instead of real vehicles. Overall, the considered errors and uncertainties can be conceived as model uncertainty, which is part of the uncertainty in the constraints.

Furthermore, it is also possible that requirements will be added, adapted, or removed during the development of a system, which affects the set of permissible force-deformation characteristics. Here, this is, for example, the case if new requirements are introduced for

crash tests.

Note that this list does not claim completeness and can be extended by further examples of uncertainty in controllable variables, uncontrollable parameters, and constraints. Like in the previous sections, only uncertainties in controllable variables and uncontrollable parameters are considered and uncertainties in constraints are neglected in the following.
According to the reasons above, the force-deformation characteristics as controllable variables comprise severe epistemic uncertainty in the early phase of crash design. This motivates the use of CSS, besides its property, to provide optimal flexibility for component design, as discussed above. In order to further take this severity into account and allow multiple sources of uncertainty for one parameter, epistemic uncertainty is modeled as intervals and fuzzy sets like above. As the magnitudes of the uncertainties are hard to estimate, especially in the early phase, these are assumed in this thesis, cf. Section 5.4. Before investigating the effects of epistemic uncertainty on DSM and corresponding optimal CSS, box- and arbitrarily-shaped CSS are compared for test-bed problems 1 and 2 without taking uncertainty into account.

5.3. Box-Shaped vs. Arbitrarily-Shaped CSS

This section compares box- and arbitrarily-shaped CSS for test-bed problems 1 and 2. Regarding test-bed problem 1, parametrized force-deformation characteristics are considered starting with components with one degree of freedom. This degree is increased successively and challenges in visualizing arbitrarily-shaped CSS are addressed.

5.3.1. Test Bed 1: Components with one degree of freedom

Similar to Section 5.1, the parametrization for test-bed problem 1 is done by dividing the deformation lengths of the components into equidistant segments and modeling the force-deformation characteristics as constant within these segments. Then, the number of segments per component is equal to the number of its design variables and its degrees of freedom. In the following, optimal box- and arbitrarily-shaped CSS are computed and compared for both independent- and dependent-decoupled design decisions. An overview of the underlying problem statements can be found in Table 1. The results of this section for independent CSS were already published in [27] for piece-wise linear force-deformation characteristics by the author of this thesis.

First, the simple crash design problem is considered for which each of the two components has one degree of freedom. For force-deformation characteristics that are constant during the

deformation of each component, it holds

$$F^1(s) = \begin{cases} F_1^1 \in [F_{\mathrm{ds},1}^{\mathrm{l},1}, F_{\mathrm{ds},1}^{\mathrm{u},1}] & \text{if } s \in [s_0, s_1), \\ 0 & \text{else}, \end{cases} \tag{5.19}$$

and

$$F^2(s) = \begin{cases} F_1^2 \in [F_{\mathrm{ds},1}^{\mathrm{l},2}, F_{\mathrm{ds},1}^{\mathrm{u},2}] & \text{if } s \in [s_1, s_2), \\ 0 & \text{else}, \end{cases} \tag{5.20}$$

where $F_{\mathrm{ds},1}^{\mathrm{l},1} = F_{\mathrm{ds}}^{\mathrm{l},1}$, $F_{\mathrm{ds},1}^{\mathrm{u},1} = F_{\mathrm{ds}}^{\mathrm{u},1}$, $F_{\mathrm{ds},1}^{\mathrm{l},2} = F_{\mathrm{ds}}^{\mathrm{l},2}$, and $F_{\mathrm{ds},1}^{\mathrm{u},2} = F_{\mathrm{ds}}^{\mathrm{u},2}$. This type of force-deformation characteristic has already been considered in Section 5.1 and examples are shown in Figures 45 and 56. For force-deformation characteristics with one degree of freedom per component, the system performance functions become linear, i.e., $f(x^1, x^2) = A^1x^1 + A^2x^2$, see Section 5.1. Thus, for F_1^k, $k = 1, 2$, optimal independent CSS can be computed by solving problem (4.70), and optimal dependent CSS by solving problems (4.99) and (4.100). Here, the maximum acceleration requirement for the second component, i.e., the third row of the inequality system (5.5), and the design space constraints for the parameter values, given in Table 3, are redundant and can be removed for computing optimal CSS. Note that box-shaped and arbitrarily-shaped CSS coincide for components with one degree of freedom as the constraints are linear and $d^k = 1$, $k = 1, 2$, holds, compare Section 4.2. The corresponding optimal independent CSS are visualized in Figure 52 and the corresponding optimal dependent CSS are visualized in Figure 53 both as geometric shapes in force space.

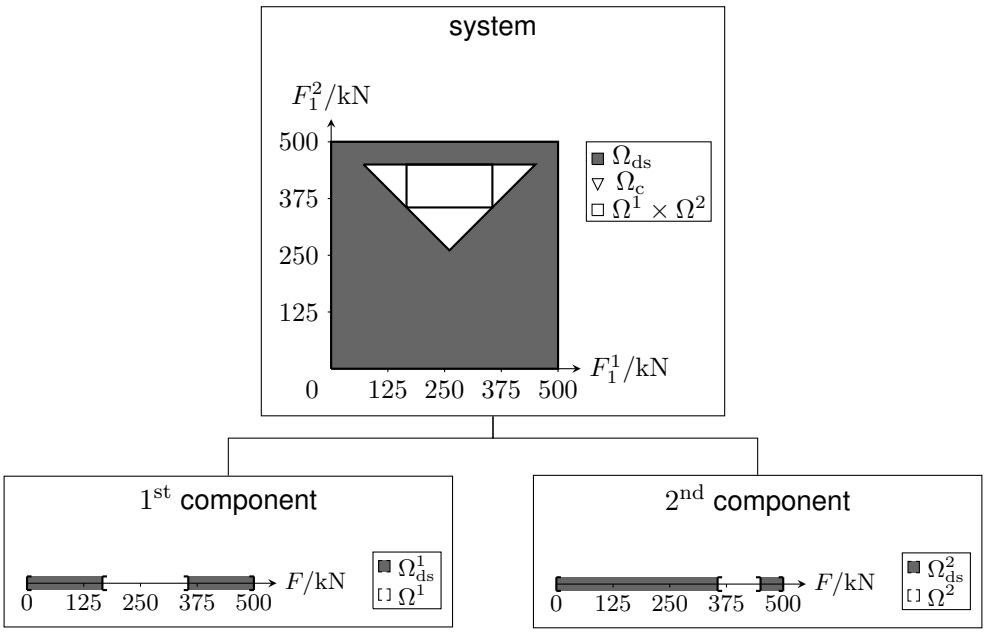


Figure 52 Independent CSS: Optimal independent CSS as regions of permissible component designs for test-bed problem 1 with components with one degree of freedom

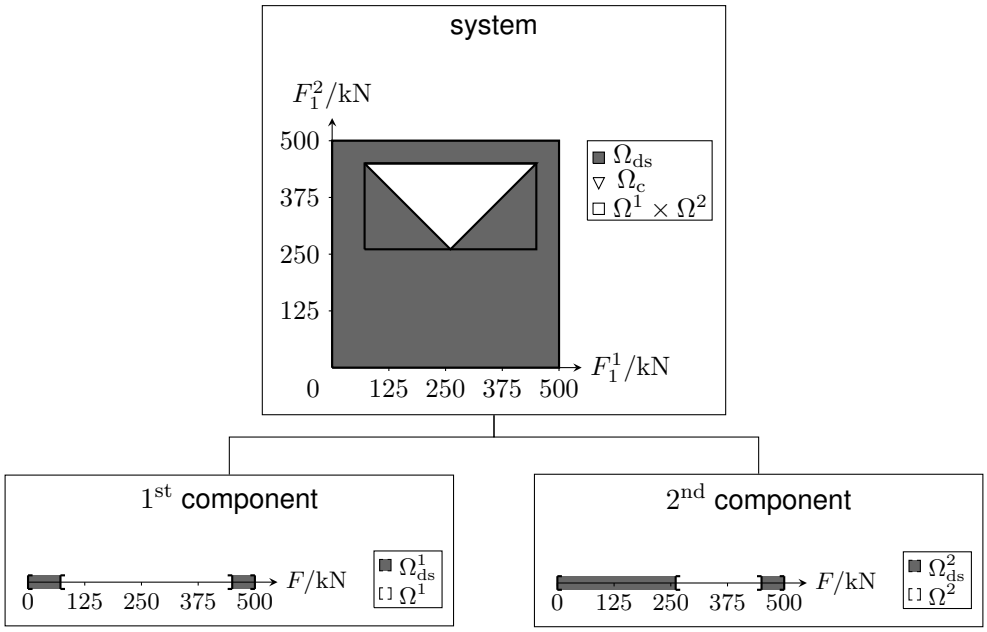


Figure 53 Dependent CSS: Optimal dependent CSS as geometric shapes in force space for test-bed problem 1 with components with one degree of freedom

5.3.2. Test Bed 1: Components with two degrees of freedom

Second, test-bed problem 1 is reconsidered for which each of the components has two degrees of freedom. Here, two equidistant segments for each component are used in which the corresponding force-deformation characteristic is modeled as constant. Therefore, it holds

$$F^1(s) = \begin{cases} F_1^1 \in [F_{\mathrm{ds},1}^{\mathrm{l},1}, F_{\mathrm{ds},1}^{\mathrm{u},1}] & \text{if } s \in [s_0, \frac{\bar{s}^1}{2} + s_0), \\ F_2^1 \in [F_{\mathrm{ds},2}^{\mathrm{l},1}, F_{\mathrm{ds},2}^{\mathrm{u},1}] & \text{if } s \in [\frac{\bar{s}^1}{2} + s_0, s_1), \\ 0 & \text{else,} \end{cases} \tag{5.21}$$

and

$$F^2(s) = \begin{cases} F_1^2 \in [F_{\mathrm{ds},1}^{\mathrm{l},2}, F_{\mathrm{ds},1}^{\mathrm{u},2}] & \text{if } s \in [s_1, \frac{\bar{s}^2}{2} + s_1), \\ F_2^2 \in [F_{\mathrm{ds},2}^{\mathrm{l},2}, F_{\mathrm{ds},2}^{\mathrm{u},2}] & \text{if } s \in [\frac{\bar{s}^2}{2} + s_1, s_2), \\ 0 & \text{else,} \end{cases} \tag{5.22}$$

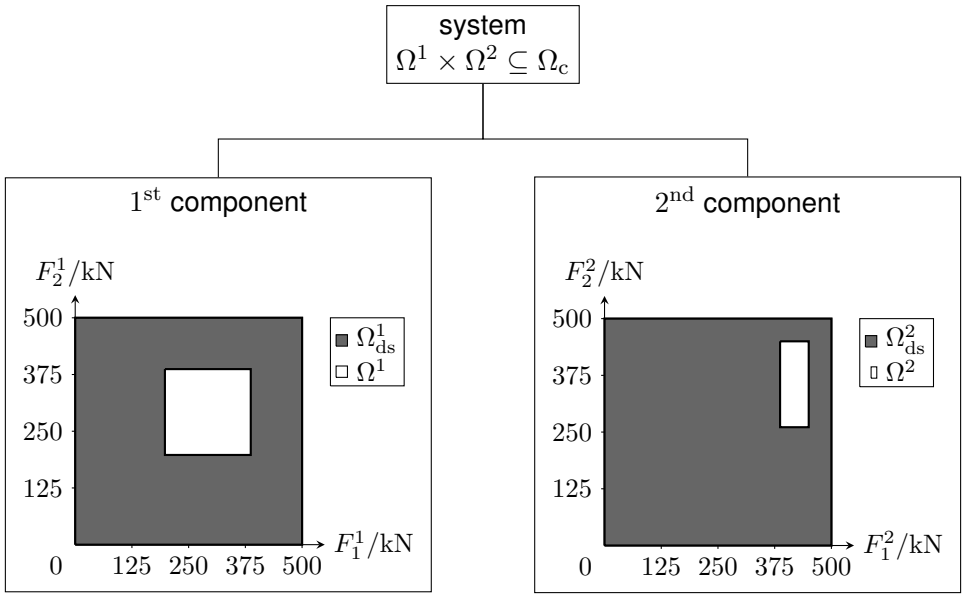
where $F_{\mathrm{ds},1}^{\mathrm{l},1} = F_{\mathrm{ds},2}^{\mathrm{l},1} = F_{\mathrm{ds}}^{\mathrm{l},1}$, $F_{\mathrm{ds},1}^{\mathrm{u},1} = F_{\mathrm{ds},2}^{\mathrm{u},1} = F_{\mathrm{ds}}^{\mathrm{u},1}$, $F_{\mathrm{ds},1}^{\mathrm{l},2} = F_{\mathrm{ds},2}^{\mathrm{l},2} = F_{\mathrm{ds}}^{\mathrm{l},2}$, and $F_{\mathrm{ds},1}^{\mathrm{u},2} = F_{\mathrm{ds},2}^{\mathrm{u},2} = F_{\mathrm{ds}}^{\mathrm{u},2}$. An example of this type of force-deformation characteristic is also shown in Figure 56. Similar to force-deformation characteristics with one degree of freedom per component, the system performance functions become linear, i.e., $f(x^1, x^2) = A^1x^1 + A^2x^2$. This means that

the requirements simplify to

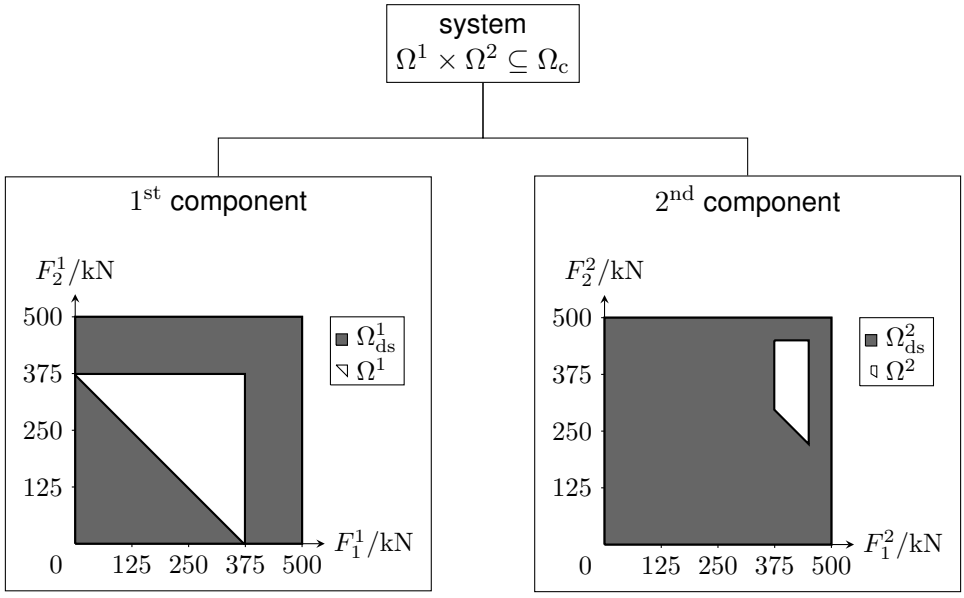
$$\underbrace{\begin{pmatrix} -\frac{\bar{s}^1}{2} & -\frac{\bar{s}^1}{2} \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}}_{=A^1} \begin{pmatrix} F_1^1 \\ F_2^1 \end{pmatrix} + \underbrace{\begin{pmatrix} -\frac{\bar{s}^2}{2} & -\frac{\bar{s}^2}{2} \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ -1 & 0 \end{pmatrix}}_{=A^2} \begin{pmatrix} F_1^2 \\ F_2^2 \end{pmatrix} \leq \underbrace{\begin{pmatrix} -\frac{1}{2}mv_0^2 \\ ma_{\mathrm{c}} \\ ma_{\mathrm{c}} \\ ma_{\mathrm{c}} \\ ma_{\mathrm{c}} \\ 0 \\ 0 \end{pmatrix}}_{=b}, \tag{5.23}$$

where the first row belongs to inequality (5.1), the second and third row to (5.2), the following two rows to (5.3), and the last two rows to (5.4). Thus, for (F_1^k, F_2^k), $k = 1, 2$, optimal box-shaped independent CSS can be computed by solving problem (4.70), optimal arbitrarily-shaped independent CSS by solving problem (4.92) and using the exact method for computing the volume from Section 4.3, optimal box-shaped dependent CSS by solving problem (4.95), and optimal arbitrarily-shaped dependent CSS by the projection method for polytopes from Section 4.3. Here, the lower design space constraint for F_1^2 and all upper design space constraints are redundant for the parameter values given in Table 3 and can be removed for computing optimal CSS. Moreover, the optimization variables for problem (4.92), i.e., b^1 and b^2, can be reduced to b_1^1, i.e., the component performance thresholds of the first component regarding the energy absorption, and b_6^1, i.e., the component performance thresholds of the first component regarding the order of deformation for its first segment, due to the detailed structure of the linear system performance functions. In addition, it is $b_2^1 = b_3^1 = ma_{\mathrm{c}}$, $b_4^1 = b_5^1 = 0$, $b_7^1 = b_6^1$, and $b^2 = b - b^1$. For problem (4.95), the optimization variables, i.e., $F^{\mathrm{l},k}$, $F^{\mathrm{u},k}$, $F_{\mathrm{cp},\iota}^{\mathrm{r},k}$, $\iota = 1, \ldots, 4$, can be reduced due to the detailed structure of the linear system performance functions as well. For each $k \in \{1, 2\}$, finding remaining component designs collected in $F_{\mathrm{cp},\iota}^{\mathrm{r},k}$ for $(F_1^{\mathrm{u},k}, F_2^{\mathrm{u},k})$ can be eliminated from the optimization problem as the corresponding system design automatically fulfills the constraints if the corner points $(F_1^{\mathrm{l},k}, F_2^{\mathrm{u},k})$ and $(F_1^{\mathrm{u},k}, F_2^{\mathrm{l},k})$, together with a corresponding $F_{\mathrm{cp},\iota}^{\mathrm{r},k}$ for each of these two corner points, fulfill the constraints. In Figure 54, the corresponding optimal independent CSS are visualized as geometric shapes in force space. Accordingly, the corresponding optimal dependent CSS are visualized in Figure 55.

Regarding optimal flexibility for independent-decoupled design decisions, the volume of $\Omega^1 \times \Omega^2$ for arbitrarily-shaped CSS is about 2.333 times larger than the volume of $\Omega^1 \times \Omega^2$ for box-shaped



(a)



(b)

Figure 54 Independent CSS: Optimal (a) box-shaped and (b) arbitrarily-shaped independent CSS as geometric shapes in force space for test-bed problem 1 with components with two degrees of freedom

CSS. For dependent-decoupled design decisions, the volume of Ω^1 for arbitrarily-shaped CSS is about 1.342 times larger and the volume of Ω^2 is about 1.671 times larger than the volumes of Ω^1 and Ω^2 for box-shaped CSS. This increase in volume offers more flexibility for component design in both cases. Note that the geometric shapes of box-shaped dependent CSS fit into the shape of arbitrarily-shaped dependent CSS. In general, this does not hold for independent CSS, which is, for example, the case in this example or the example in [27].

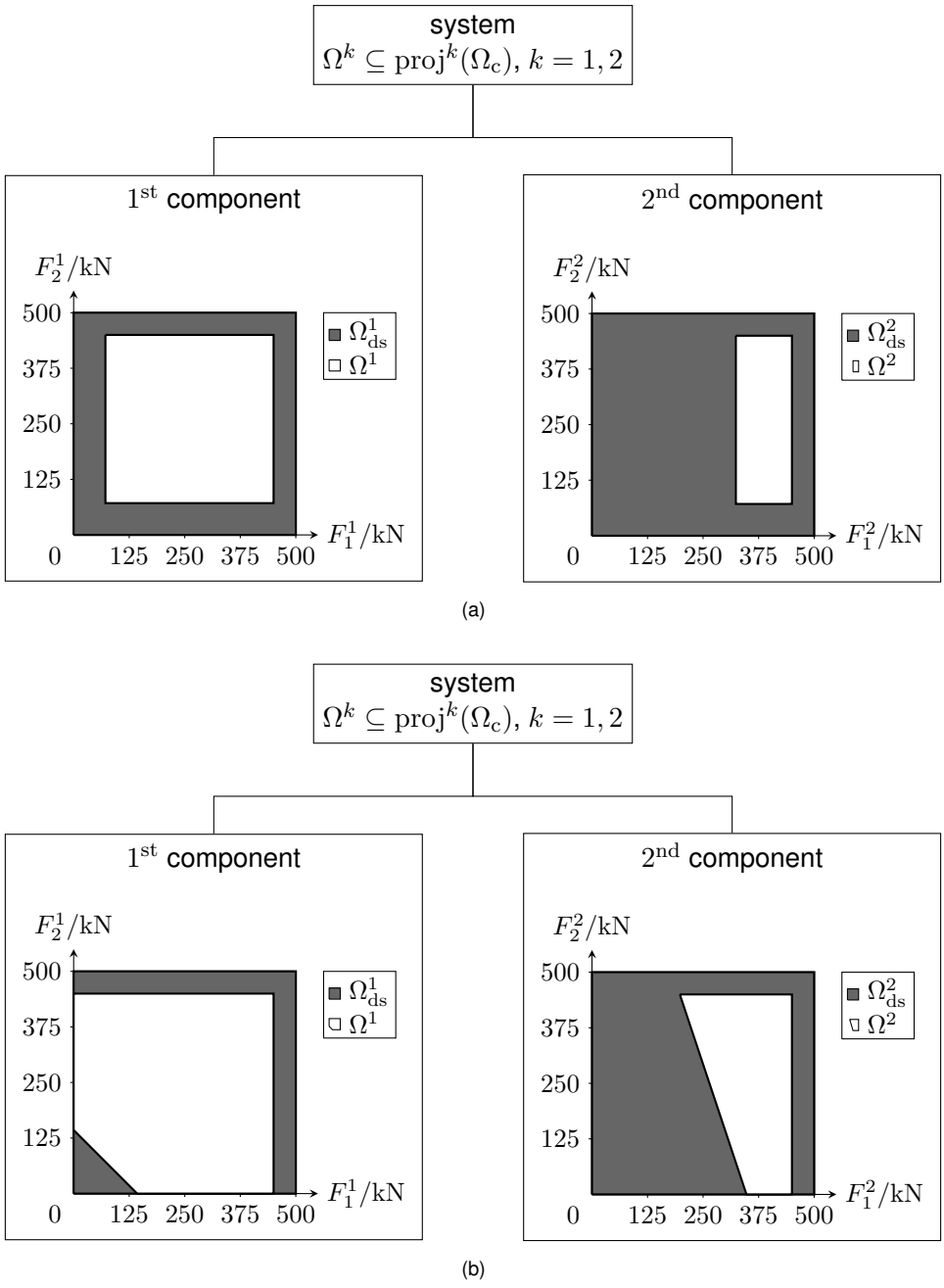


Figure 55 Dependent CSS: Optimal (a) box-shaped and (b) arbitrarily-shaped dependent CSS as regions of permissible component designs for test-bed problem 1 with components with two degrees of freedom

5.3.3. Test Bed 1: Components with arbitrary many degrees of freedom

Similar investigations can be done for components with arbitrary many degrees of freedom. If d^k, $k = 1, 2$, equidistant segments for the components are used in which the force-deformation characteristics are modeled as constant, components with arbitrary many degrees of freedom are obtained for test-bed problem 1. Here, the consideration are limited to $d^1 = d^2 = d/2$.

Thus, it holds

$$F^1(s) = \begin{cases} F_i^1 \in [F_{\mathrm{ds},i}^{\mathrm{l},1}, F_{\mathrm{ds},i}^{\mathrm{u},1}] & \text{if } s \in [(i-1)\frac{\bar{s}^1}{d^1} + s_0, i\frac{\bar{s}^1}{d^1} + s_0),\ i = 1, \dots, d^1 \\ 0 & \text{else,} \end{cases} \tag{5.24}$$

and

$$F^2(s) = \begin{cases} F_i^2 \in [F_{\mathrm{ds},i}^{\mathrm{l},2}, F_{\mathrm{ds},i}^{\mathrm{u},2}] & \text{if } s \in [(i-1)\frac{\bar{s}^2}{d^2} + s_1, i\frac{\bar{s}^2}{d^2} + s_1),\ i = 1, \dots, d^2 \\ 0 & \text{else,} \end{cases} \tag{5.25}$$

where $F_{\mathrm{ds},i}^{\mathrm{l},1} = F_{\mathrm{ds}}^{\mathrm{l},1}$, $F_{\mathrm{ds},i}^{\mathrm{u},1} = F_{\mathrm{ds}}^{\mathrm{u},1}$, $F_{\mathrm{ds},i}^{\mathrm{l},2} = F_{\mathrm{ds}}^{\mathrm{l},2}$, and $F_{\mathrm{ds},i}^{\mathrm{u},2} = F_{\mathrm{ds}}^{\mathrm{u},2}$. Examples of this type of force-deformation characteristics are also shown in Figure 56. Like for force-deformation characteristics with one or two degrees of freedom per component, the system performance functions for force-deformation characteristics with arbitrary many degrees of freedom per component become linear and are of the form $f(x^1, x^2) = A^1x^1 + A^2x^2$ with

$$A^1 = \begin{pmatrix} -\frac{\bar{s}^1}{d^1} & -\frac{\bar{s}^1}{d^1} & \cdots & -\frac{\bar{s}^1}{d^1} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}, \quad A^2 = \begin{pmatrix} -\frac{\bar{s}^2}{d^2} & -\frac{\bar{s}^2}{d^2} & \cdots & -\frac{\bar{s}^2}{d^2} \\ 0 & 0 & \cdots & 0 \\ 0 & 0 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \\ -1 & 0 & \cdots & 0 \\ -1 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & \cdots & 0 \end{pmatrix}, \tag{5.26}$$

where the system performance thresholds in b must be adapted accordingly. The first row of the corresponding system of inequalities $Ax \leq b$ belongs to inequality (5.1), the following $d/2$ rows to (5.2), the then following $d/2$ rows to (5.3), and the last $d/2$ rows to (5.4). Again, for $(F_1^k, \ldots, F_{d^k}^k)$, $k = 1, 2$, optimal box-shaped independent CSS can be computed by solving problem (4.70), and optimal arbitrarily-shaped independent CSS by solving problem (4.92) and using the exact method for computing the volume from Section 4.3. Optimal box-shaped dependent CSS can be computed by solving problem (4.95), and optimal arbitrarily-shaped dependent CSS by the projection method for polytopes from Section 4.3.

For components with arbitrary many degrees of freedom, all upper design space constraints are redundant for the parameter values given in Table 3 and can be removed for computing optimal CSS. Moreover, the optimization variables b^1 and b^2 for problem (4.92) can be reduced to the component performance thresholds of the first component regarding the energy absorption and regarding the order of deformation for its first segment. This holds due to the detailed structure of the linear system performance functions. Hence, only two optimization variables remain for problem (4.92). Also for problem (4.95), the optimization variables $F^{\mathrm{l},k}$, $F^{\mathrm{u},k}$, $F_{\mathrm{cp},\iota}^{\mathrm{r},k}$, $\iota = 1, \ldots, 2^{d^k}$ can be reduced due to the detailed structure of the linear system performance functions. For corner points of a box-shaped CSS of the k^{th} component that considers more than one upper bound, i.e., more than one entry of $(F_1^{\mathrm{u},k}, \ldots, F_{d^k}^{\mathrm{u},k})$, $k \in \{1, 2\}$, finding a remaining component design of the other component can be eliminated from the optimization problem as the resulting system design automatically fulfills the constraints if the other corner points, together with a corresponding remaining component design, fulfill the constraints. Thus, $d^2/2 + 3d/2$ optimization variables remain for problem (4.92).

However, the corresponding optimal arbitrarily-shaped independent and dependent CSS are difficult to visualize as geometric shapes in force space for $d^k > 3$, $k = 1, \ldots, n$. This is addressed in the following for piece-wise constant force-deformation-characteristics for which the deformation lengths are divided into d^k segments, $k = 1, \ldots, n$.

5.3.4. Visualization of permissible force-deformation characteristics

To overcome the problem of visualizing CSS as geometric shapes in force space for $d^k > 3$, $k = 1, \ldots, n$, regions of possibly permissible force-deformation characteristics can be used. Recall that force-deformation characteristics, e.g., given by Equation (5.24) or (5.25) are said to be permissible if their design variables are permissible, i.e., if $(F_1^k, \ldots, F_{d^k}^k) \in \Omega^k$, $k = 1, \ldots, n$ holds. For every component, the *region of all possibly permissible force-deformation characteristics* is the region between the two bounding characteristics defined by the design variables $F_{\mathrm{ob},i}^{\mathrm{l},k} = \min\{F_i^k \mid (F_1^k, \ldots, F_{d^k}^k) \in \Omega^k\}$ and $F_{\mathrm{ob},i}^{\mathrm{u},k} = \min\{F_i^k \mid (F_1^k, \ldots, F_{d^k}^k) \in \Omega^k\}$ for $i = 1, \ldots, d^k$, $k = 1, \ldots, n$. Here, the subscript ob stands for "outer box" as the d^k-dimensional intervals $[F_{\mathrm{ob}}^{\mathrm{l},k}, F_{\mathrm{ob}}^{\mathrm{u},k}]$ form minimum volume outer boxes of the CSS Ω^k, $k = 1, \ldots, n$, which can be computed by solving problems (4.99) and (4.100). The projections $\mathrm{proj}_i^k([F_{\mathrm{ob}}^{\mathrm{l},k}, F_{\mathrm{ob}}^{\mathrm{u},k}]) = [F_{\mathrm{ob},i}^{\mathrm{l},k}, F_{\mathrm{ob},i}^{\mathrm{u},k}]$ are not intended to obtain optimal flexibility for component design but for the purpose of visualization of Ω^k, $k = 1, \ldots, n$, compare [6]. Note however that it cannot be guaranteed that all force-deformation characteristics within these regions are permissible.

Also for the purpose of visualization, maximum volume inner boxes $[F_{\mathrm{ib}}^{\mathrm{l},k}, F_{\mathrm{ib}}^{\mathrm{u},k}]$ can be computed to define a *region of only permissible force-deformation characteristics* for every component. Here, however, it cannot be guaranteed that all permissible characteristics are within these regions. Similarly, boxes defined, e.g., by

$$[F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{u},k}] = [\theta F_{\mathrm{ib}}^{\mathrm{l},k} + (1-\theta) F_{\mathrm{ob}}^{\mathrm{l},k}, \theta F_{\mathrm{ib}}^{\mathrm{l},k} + (1-\theta) F_{\mathrm{ob}}^{\mathrm{u},k}], \tag{5.27}$$

$\theta \in [0,1]$, $k = 1, \ldots, n$, can be used to visualize further regions of possibly permissible force-deformation characteristics. For $\theta = 0$, the minimum volume outer box and for $\theta = 1$, the maximum volume inner box is obtained.
Note that for box-shaped CSS, the regions of all possibly permissible characteristics and only permissible characteristics, defined by the minimum volume outer and maximum volume inner boxes, coincide as the CSS are d^k -dimensional intervals, i.e., $[F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{u},k}] = [F_{\mathrm{b},\theta'}^{\mathrm{l},k}, F_{\mathrm{b},\theta'}^{\mathrm{u},k}]$ for $\theta, \theta' \in [0,1]$, $k = 1, \ldots, n$. Thus, there is only one region of force-deformation characteristics for every component, which was already used in Sections 5.1 and 5.2 for the visualization of box-shaped CSS as regions of permissible characteristics.

In order to get an idea about the quality of a region of possibly permissible characteristics for arbitrarily-shaped CSS, the two key metrics

$$r_{\mathrm{CSS},\theta}^{k} = \frac{\mathrm{vol}(\Omega^k \cap [F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{l},k}])}{\mathrm{vol}(\Omega^k)}, \tag{5.28}$$

$$r_{\mathrm{b},\theta}^{k} = \frac{\mathrm{vol}(\Omega^k \cap [F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{l},k}])}{\mathrm{vol}([F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{l},k}])} \tag{5.29}$$

are introduced, where $0 \leq r_{\mathrm{CSS},\theta}^{k}, r_{\mathrm{b},\theta}^{k} \leq 1$, $k = 1, \ldots, n$. The metrics $r_{\mathrm{CSS},\theta}^{k}$ represent the ratio of permissible force-deformation characteristics within the corresponding region of possibly permissible characteristics to all permissible characteristics and the metrics $r_{\mathrm{b},\theta}^{k}$ represent the ratio of permissible characteristics within the corresponding region of possibly permissible characteristics to all characteristics within the corresponding region of possibly permissible characteristics, $k = 1, \ldots, n$. For the region of all possibly permissible force-deformation characteristics, it holds $r_{\mathrm{CSS},\theta=0}^{k} = 1$ and for a region of only permissible force-deformation characteristics, it holds $r_{\mathrm{b},\theta=1}^{k} = 1$, $k = 1, \ldots, n$. In the following, $r_{\mathrm{b},\theta}^{k}$ are used to assign a grayscale value to $[F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{u},k}]$ and to the corresponding regions of possibly permissible force-deformation characteristics for $\theta \in [0,1]$, $k = 1, \ldots, n$. They are colored using a dot pattern of the specified color. For $r_{\mathrm{b},\theta}^{k} = 0$, the color black and for $r_{\mathrm{b},\theta}^{k} = 1$, the color white is assigned. In Figure 56, examples of Ω^k are visualized for different degrees of freedom as geometric shapes in force space together with $[F_{\mathrm{b},\theta}^{\mathrm{l},k}, F_{\mathrm{b},\theta}^{\mathrm{u},k}]$, $\theta = 0, 0.5, 1$, and the corresponding regions of possibly permissible force-deformation characteristics, $k \in \{1, \ldots, n\}$.

Whereas component designs within $[F_{\mathrm{b},\theta=0}^{\mathrm{l},k}, F_{\mathrm{b},\theta=0}^{\mathrm{u},k}]$ are always permissible, and so are

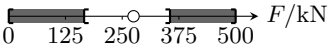
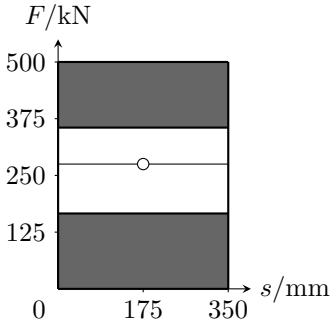
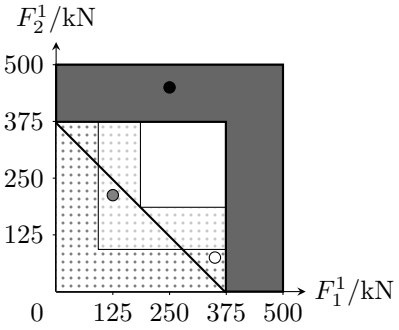
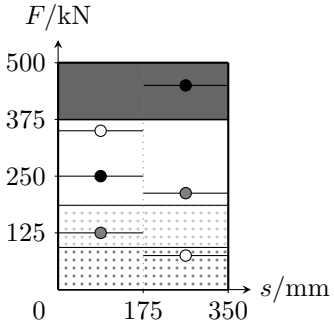
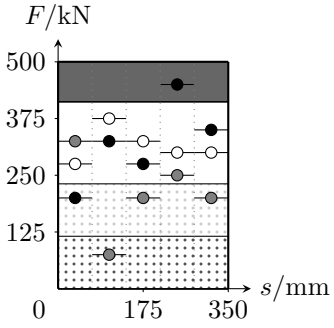
d^k	geometric shapes in force space	regions of possibly permissible characteristics
1		 $r^k_{\mathrm{CSS},\theta=0.5} = r^k_{\mathrm{CSS},\theta=1} = 1.00$
2		 $r^k_{\mathrm{CSS},\theta=0.5} = 0.87,\ r^k_{\mathrm{CSS},\theta=1} = 0.50$
arbitrary		example ($d^k = 5$)  $r^k_{\mathrm{CSS},\theta=0.5} = 0.48,\ r^k_{\mathrm{CSS},\theta=1} = 0.05$

Figure 56 Examples of Ω^k with one, two, and arbitrary many degrees of freedom as geometric shapes in force space together with $[F^{\mathrm{l},k}_{\mathrm{b},\theta}, F^{\mathrm{u},k}_{\mathrm{b},\theta}]$ for $\theta = 0$ (dark gray dot pattern), $\theta = 0.5$ (light gray dot pattern), and $\theta = 1$ (filled white) on the left-hand side, $k \in \{1, \ldots, n\}$. On the right-hand side, the corresponding regions of possibly permissible force-deformation characteristics are shown for which the same coloring scheme is used. Within these regions, there are permissible component designs or characteristics (large white dots) or non-permissible (large gray dots). Outside, they are always non-permissible (large black dots).

the corresponding characteristics, it must always be tested whether component designs within $[F^{\mathrm{l},k}_{\mathrm{b},\theta}, F^{\mathrm{u},k}_{\mathrm{b},\theta}]$, $\theta \in (0, 1]$ or corresponding characteristics are permissible, i.e., whether $(F^k_1, \ldots, F^k_{d^k}) \in \Omega^k,\ k = 1, \ldots, n$ holds. Subsequently, general properties of box- and

arbitrarily-shaped CSS are compared as well as their volumes, average edge lengths, and computation time. First, investigations are done for test-bed problem 1.

5.3.5. Test Bed 1: Comparison of box-shaped and arbitrarily-shaped CSS

Compared to arbitrarily-shaped CSS, box-shaped CSS are easy to visualize as regions of permissible force-deformation characteristics due to the given lower and upper bounds for each design variable, i.e., F_i^{l} and F_i^{u}, $i = 1, \ldots, d$. Furthermore, regions of possibly permissible characteristics are always regions of only permissible characteristics for box-shaped CSS which do not contain non-permissible characteristics. For arbitrarily-shaped CSS, regions of only possible characteristics can be computed along with further regions of possibly permissible characteristics, as described above. These multiple regions of possibly permissible force-deformation characteristics yield additional information for designing components.
In order to provide optimal flexibility for independent-decoupled design decisions in the case of test-bed problem 1, the volume of $\Omega^1 \times \Omega^2$ for arbitrarily-shaped CSS grows faster with the degrees of freedom d^k, $k = 1, 2$, than the volume of $\Omega^1 \times \Omega^2$ for box-shaped CSS in test-bed problem 1. This means the greater d^k, $k = 1, 2$, the more flexibility for component design is obtained when using independent arbitrarily-shaped instead of box-shaped independent CSS. This difference can also be asserted in terms of the average edge length of $\Omega^1 \times \Omega^2$, denoted by $\bar{L}$. For the simple crash design problem, it holds

$$\bar{L} = \sqrt[d]{\mathrm{vol}(\Omega^1 \times \cdots \times \Omega^n)} \tag{5.30}$$

with $n = 2$. Similar results are obtained regarding optimal flexibility for dependent-decoupled design decisions in terms of the volumes of Ω^1 and Ω^2. A comparison of the volumes of $\Omega^1 \times \Omega^2$ and their corresponding edge lengths is shown for independent CSS in Figure 57 and for dependent CSS in Figure 58. However, the computation times for box-shaped and arbitrarily-shaped CSS differ. In Figure 59, a comparison is shown for independent CSS and in Figure 60, a comparison for dependent CSS.

With increasing d^k, computing arbitrarily-shaped independent CSS with the exact method from Section 4.3 requires much more time than the computation of box-shaped solution spaces due to the expensive volume computation. It increases exponentially on a logarithmic scale and is therefore unsuitable for large d^k, $k = 1, 2$. Hence, further approaches like the Monte Carlo method for volume computation are also considered in the comparison for test-bed problem 2 below.
For dependent CSS, the difference in computation time between box-shaped and arbitrarily-shaped CSS is not that significant. Although computing box-shaped CSS with the direct method from Section 4.2 requires less time than computing arbitrarily-shaped CSS for $2 \leq d^k \leq, 10$, the difference gets smaller on the logarithmic scale with increasing d^k, $k = 1, 2$. This means that for crash design problems with a higher number of degrees of freedom, it is not a priori clear if the direct method from Section 4.2 is faster than the two-step method from Section 4.3.

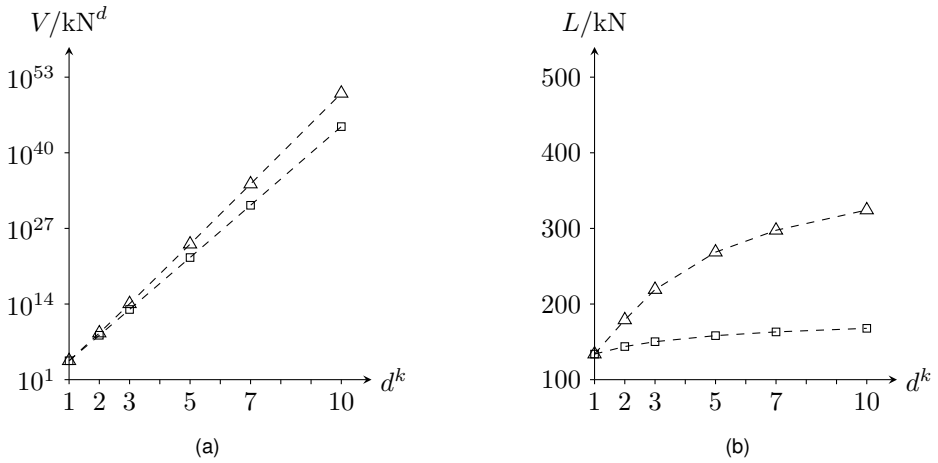


Figure 57 Independent CSS: Volume of $\Omega^1 \times \Omega^2$ (a) and average edge length (b) of optimal independent box-shaped (square marks) and arbitrarily-shaped (triangle marks) CSS as a function of the components' degrees of freedom for test-bed problem 1.

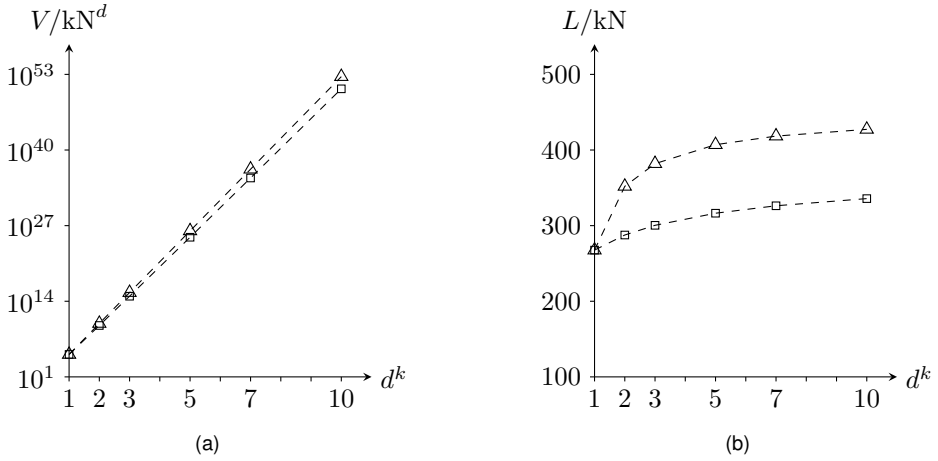


Figure 58 Dependent CSS: Volume of $\Omega^1 \times \Omega^2$ (a) and average edge length (b) of optimal box-shaped (square marks) and arbitrarily-shaped (triangle marks) dependent CSS as a function of the components' degrees of freedom for test-bed problem 1.

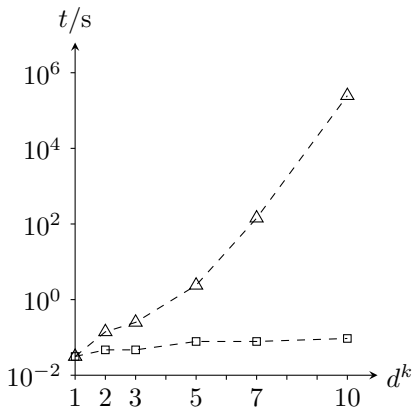


Figure 59 Independent CSS: CPU time for Intel(R) Xeon(R) CPU E5-1660 v4 @ 3.20 GHz to compute optimal box-shaped (square marks) and arbitrarily-shaped (triangle marks) independent CSS as a function of the components' degrees of freedom for test-bed problem 1 for which the exact method from Section 4.3 is used to compute arbitrarily-shaped CSS.

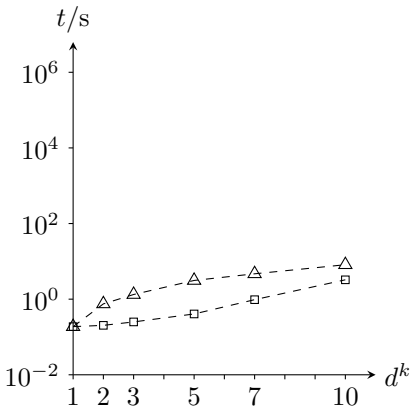


Figure 60 Dependent CSS: CPU time for Intel(R) Xeon(R) CPU E5-1660 v4 @ 3.20 GHz to compute optimal box-shaped (square marks) and arbitrarily-shaped (triangle marks) dependent CSS as a function of the components' degrees of freedom for test-bed problem 1 for which the direct method from Section 4.2 is used to compute box-shaped CSS.

5.3.6. Test Bed 2: Comparison of box-shaped and arbitrarily-shaped CSS

For test-bed problem 2, the system performance functions are also linear and hence, optimal CSS can be computed like above. In Figure 61, optimal box- and arbitrarily-shaped independent CSS are visualized, for which the exact method was used to compute arbitrarily-shaped CSS. Note that for test-bed problem 2, the number of optimization variables to compute arbitrarily-shaped CSS can be reduced, similar to above. The corresponding key metrics $r^k_{\mathrm{b},\theta}$, $k = 1, \ldots, 7$, introduced in Equation (5.28), are stated in Table 8. In Figure 62, optimal box- and arbitrarily-shaped dependent CSS are visualized, for which the direct method was used to compute box-shaped CSS. The corresponding metrics $r^k_{\mathrm{b},\theta}$, $k = 1, \ldots, 7$ are stated in Table 9.

Table 8 Independent CSS: Ratio of permissible characteristics within the region of possibly permissible characteristics to all characteristics within the region of possibly permissible characteristics for arbitrarily -shaped independent CSS of test-bed problem 2.

quantity	$r^1_{\mathrm{CSS},\theta=0.5}$	$r^2_{\mathrm{CSS},\theta=0.5}$	$r^3_{\mathrm{CSS},\theta=0.5}$	$r^4_{\mathrm{CSS},\theta=0.5}$	$r^5_{\mathrm{CSS},\theta=0.5}$	$r^6_{\mathrm{CSS},\theta=0.5}$	$r^7_{\mathrm{CSS},\theta=0.5}$
value	0.26	0.81	1.00	1.00	0.91	0.79	0.67
quantity	$r^1_{\mathrm{CSS},\theta=1}$	$r^2_{\mathrm{CSS},\theta=1}$	$r^3_{\mathrm{CSS},\theta=1}$	$r^4_{\mathrm{CSS},\theta=1}$	$r^5_{\mathrm{CSS},\theta=1}$	$r^6_{\mathrm{CSS},\theta=1}$	$r^7_{\mathrm{CSS},\theta=1}$
value	0.05	0.65	1.00	0.99	0.82	0.61	0.41

Table 9 Dependent CSS: Ratio of permissible characteristics within the region of possibly permissible characteristics to all characteristics within the region of possibly permissible characteristics for arbitrarily -shaped dependent CSS of test-bed problem 2.

quantity	$r^1_{\mathrm{CSS},\theta=0.5}$	$r^2_{\mathrm{CSS},\theta=0.5}$	$r^3_{\mathrm{CSS},\theta=0.5}$	$r^4_{\mathrm{CSS},\theta=0.5}$	$r^5_{\mathrm{CSS},\theta=0.5}$	$r^6_{\mathrm{CSS},\theta=0.5}$	$r^7_{\mathrm{CSS},\theta=0.5}$
value	1.00	1.00	1.00	1.00	1.00	1.00	0.87
quantity	$r^1_{\mathrm{CSS},\theta=1}$	$r^2_{\mathrm{CSS},\theta=1}$	$r^3_{\mathrm{CSS},\theta=1}$	$r^4_{\mathrm{CSS},\theta=1}$	$r^5_{\mathrm{CSS},\theta=1}$	$r^6_{\mathrm{CSS},\theta=1}$	$r^7_{\mathrm{CSS},\theta=1}$
value	1.00	1.00	1.00	1.00	1.00	1.00	0.74

In general, the results for test-bed problem 2 are similar to those of test-bed problem 1. For

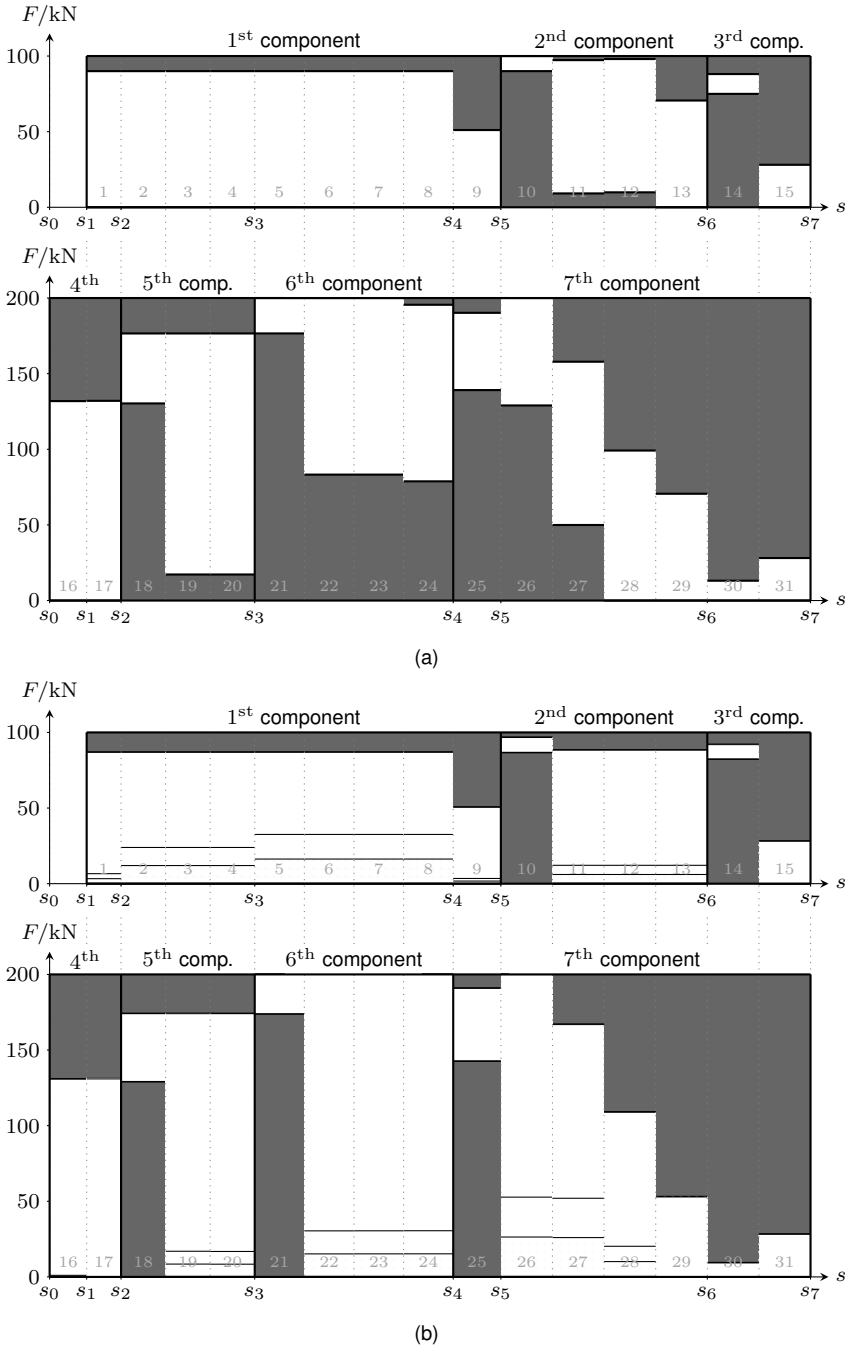


Figure 61 Independent CSS: Optimal (a) box-shaped and (b) arbitrarily-shaped independent CSS visualized as regions of possibly permissible force-deformation characteristics with $\theta = 0, 0.5, 1$ for test-bed problem 2. Note the dot pattern for $\theta = 0, 0.5$ is not visible as the dots are almost white.

both independent and dependent CSS, the volume of $\Omega^1 \times \cdots \times \Omega^7$ for arbitrarily-shaped CSS is again greater than that for box-shaped CSS. This is also expressed in the average edge length. The detailed values are stated in Tables 10 and 11. Nevertheless, it must be tested whether a design within the regions of possibly permissible force-deformation characteristics is permissible, i.e., whether it is in Ω^k, $k = 1, \ldots, 7$. Here, the different design regions defined by $[F^{\mathrm{l},k}_{\mathrm{b},\theta}, F^{\mathrm{u},k}_{\mathrm{b},\theta}]$, which are visualized for $\theta = 0, 0.5, 1$ in Figures 61 and 62 support this testing where the inner regions with $\theta = 1$ comprise only permissible designs, $k = 1, \ldots, 7$. The

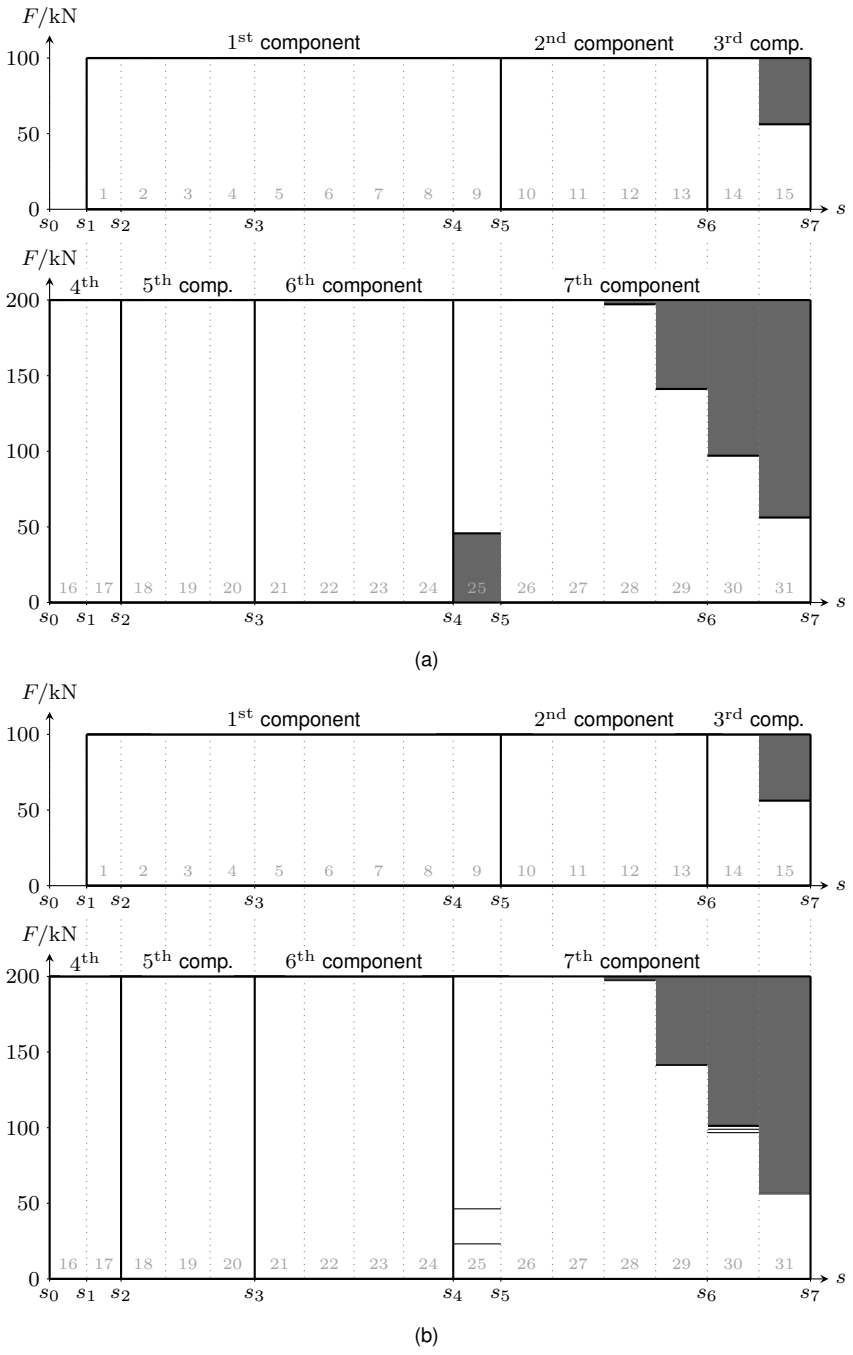


Figure 62 Dependent CSS: Optimal (a) box-shaped and (b) arbitrarily-shaped dependent CSS visualized as regions of possibly permissible force-deformation characteristics with $\theta = 0, 0.5, 1$ for test-bed problem 2. Note the dot pattern for $\theta = 0, 0.5$ is not visible as the dots are almost white.

dot patterns, which are used in these figures to visualize $[F^{\mathrm{l},k}_{\mathrm{b},\theta}, F^{\mathrm{u},k}_{\mathrm{b},\theta}]$, are almost white and therefore not visible, i.e., $r^k_{\mathrm{b},\theta} \approx 1$, $k = 1, \ldots, 7$. Hence, the chances are high to obtain also permissible designs when designing in these regions. Recall that for box-shaped CSS, these regions do not exist or rather, they coincide with the region of only permissible characteristics. Therefore, testing is redundant for box-shaped CSS.

Similar to test-bed problem 1, the computation time of arbitrarily-shaped CSS is greater than the computation time of box-shaped CSS for test-bed problem 2. In Table 10, the results for independent CSS are stated. Here, the exact method for computing arbitrarily-shaped CSS is complemented by the Monte Carlo approach for which 10^4, 10^5, 10^6, and 10^7 local sample points per component are used to compute the volume of each CSS Ω^k, $k = 1, \ldots, 7$. Furthermore, details on the initial values for optimizing arbitrarily-shaped CSS obtained from box-shaped solution spaces, see Section 4.3, are added, too. Besides their computation times, also the 31-dimensional volume of $\Omega^1 \times \cdots \times \Omega^7$ and the average edge length is stated using these approaches. In order to compare the different approaches, the exact method is used to determine the volume of the computed CSS Ω^k, $k = 1, \ldots, 7$, in Table 10. In Table 11, the same results for dependent CSS are stated in which the direct method to compute box-shaped CSS is complemented by the two-step method.

Table 10 Independent CSS: Results for $\mathrm{vol}(\Omega^1 \times \cdots \times \Omega^7)$, average edge length $\bar{L}$, and CPU time t_{CPU} using different methods to compute box-shaped and arbitrarily-shaped independent CSS for test-bed problem 2 (Intel(R) Xeon(R) CPU E5-1660 v4 @ 3.20 GHz).

quantity	$\mathrm{vol}(\Omega^1 \times \cdots \times \Omega^7)/\mathrm{kN}^{31}$	$\bar{L}/\mathrm{kN}$	$t_{\mathrm{CPU}}/\mathrm{s}$
box-shaped	$3.42 \cdot 10^{56}$	66.63	1.25
arbitrarily-shaped: initial values	$1.84 \cdot 10^{57}$	70.35	1.25
arbitrarily-shaped: Monte Carlo 10^4	$2.35 \cdot 10^{57}$	70.90	$3.61 \cdot 10^3$
arbitrarily-shaped: Monte Carlo 10^5	$2.64 \cdot 10^{57}$	71.17	$3.34 \cdot 10^4$
arbitrarily-shaped: Monte Carlo 10^6	$2.71 \cdot 10^{57}$	71.23	$3.43 \cdot 10^5$
arbitrarily-shaped: Monte Carlo 10^7	$2.82 \cdot 10^{57}$	71.33	$3.55 \cdot 10^6$
arbitrarily-shaped: exact	$3.21 \cdot 10^{57}$	71.62	$3.66 \cdot 10^5$

Table 11 Dependent CSS: Results for $\mathrm{vol}(\Omega^1 \times \cdots \times \Omega^7)$, average edge length $\bar{L}$, and CPU time t_{CPU} using different methods to compute box-shaped and arbitrarily-shaped dependent CSS for test-bed problem 2 (Intel(R) Xeon(R) CPU E5-1660 v4 @ 3.20 GHz).

quantity	$\mathrm{vol}(\Omega^1 \times \cdots \times \cdots \times \Omega^7)/\mathrm{kN}^{31}$	$\bar{L}/\mathrm{kN}$	$t_{\mathrm{CPU}}/\mathrm{s}$
box-shaped: direct	$2.70 \cdot 10^{65}$	129.02	$2.36 \cdot 10$
box-shaped: two-step	$2.68 \cdot 10^{65}$	128.99	$1.29 \cdot 10^2$
arbitrarily-shaped	$3.64 \cdot 10^{65}$	130.27	$1.28 \cdot 10^2$

For arbitrarily-shaped independent CSS, $\Omega^1 \times \cdots \times \Omega^7$ comprises the largest volume if the exact method is used, see Table 10. When using the Monte Carlo method for optimizing CSS, the volume is smaller, although the same optimization problem is solved. The fewer sample points are used, the more likely is an early termination of the optimization solver, which comes along with a smaller volume of $\Omega^1 \times \cdots \times \Omega^7$. Here, changes in the optimization variables are not necessarily reflected in the value of the objective function because no new sample points

are included or excluded in Ω^k, $k = 1, \ldots, 7$. However, it also holds that the computation time is smaller, the fewer sample points are used. The computation time of box-shaped CSS and of the initial values for optimizing arbitrarily-shaped CSS is the same because the initial values are directly obtained from the box-shaped CSS. Still, the volume of $\Omega^1 \times \cdots \times \Omega^7$ for the initial values is around five times larger than the corresponding volume for box-shaped CSS. Thus, it is overall a trade-off between the volume and the CPU time when choosing an approach to compute arbitrarily-shaped CSS. If the volume of box-shaped CSS is already sufficient for the specific application, box-shaped CSS are more preferable.
For box-shaped dependent CSS, the direct method outperforms the two-step method for test-bed problem 2 as both the volume of $\Omega^1 \times \cdots \times \Omega^7$ is slightly larger and the computation time is less. Here, the difference in volume is due to numerical errors, as both methods are exact. In comparison to arbitrarily-shaped CSS, the difference in volume is small as the CSS match the component design spaces for the first, second, fourth, fifth, and sixth component and are very similar for the other components.
Note that in this section, optimal flexibility for component design was considered under absence of uncertainty. In the subsequent section, uncertainty is included in the considerations.

5.4. CSS under Epistemic Uncertainty

This section computes CSS using the DSM of test-bed problems 1 and 2 under epistemic uncertainty. Both uncertainties in controllable variables and uncontrollable parameters are investigated, for which uncertainty magnitudes are assumed. For a simpler visualization, the considerations are limited to box-shaped CSS. Nevertheless, they could be done similarly for arbitrarily-shaped CSS.

5.4.1. Test Bed 1: Uncertainties in Controllable Variables

First, interval-type uncertainty in controllable variables, i.e., in the force values of the force-deformation characteristics, is taken into account for test-bed problem 1. Here, components with one degree of freedom are considered for which box- and arbitrarily-shaped CSS coincide. Like before, the values of Table 2 are used for the design space parameters and the values of Table 3 for the uncontrollable parameters. For them, no uncertainty is assumed in this subsection. It is distinguished between three cases of uncertainty in controllable variables, which are stated in the following:

(a) *No uncertainties*: The uncertainty magnitudes are zero, which means there is also no uncertainty in controllable variables. For this case, the CSS match the ones visualized in Figures 52 and 53 and are only used for the purpose of comparison.

(b) *Uncertainties with known magnitudes*: The uncertainty magnitudes are assumed to be $\bar{\delta}_1 = \bar{\delta} = 25\,\mathrm{kN}$ for the forces F_1 and F_2, i.e., the uncertainty magnitudes are known.

(c) *Uncertainties with unknown magnitudes*: The uncertainty magnitudes are assumed to be unknown. Thus, maximum magnitudes are computed using the weighting factors $\omega_1 = \omega_2 = 1$, for which the same uncertainty magnitudes will be obtained for the forces F_1 and F_2.

As the constraints are linear in the controllable variables, results from Section 4.2 can be used to compute optimal CSS Ω^1, Ω^2 in case (a), optimal worst-case CSS $\check{\Omega}^1_{\mathrm{wc}}$, $\check{\Omega}^2_{\mathrm{wc}}$ in case (b) for which optimal CSS Ω^1, Ω^2 and optimal best-case CSS $\check{\Omega}^1_{\mathrm{bc}}$, $\check{\Omega}^2_{\mathrm{bc}}$ are deduced from $\check{\Omega}^1_{\mathrm{wc}}$, $\check{\Omega}^2_{\mathrm{wc}}$, and maximum uncertainty magnitudes $\bar{\delta}_1$, $\bar{\delta}_2$ and optimal worst-case CSS $\check{\Omega}^1_{\mathrm{wc}}$, $\check{\Omega}^2_{\mathrm{wc}}$ in case (c) for which optimal CSS Ω^1, Ω^2 are deduced. Note that in (c), no best-case CSS are computed because its interpretation is critical, see Section 3.4. The corresponding maximum uncertainty magnitudes are $\bar{\delta}_1 = \bar{\delta}_2 = 63.09\,\mathrm{kN}$ and $\check{\Omega}^1_{\mathrm{wc}} \times \check{\Omega}^2_{\mathrm{wc}}$ is a singleton. All in all, the optimal independent CSS are shown in Figure 63, and the optimal dependent CSS in Figure 64, both as geometric shapes in force space and regions of permissible force-deformation characteristics.

The worst-case CSS of target designs $\check{\Omega}^1_{\mathrm{wc}}$ and $\check{\Omega}^2_{\mathrm{wc}}$ show the target designs which are permissible for all uncertainty scenarios, and the best-case CSS of target designs $\check{\Omega}^1_{\mathrm{bc}}$ and $\check{\Omega}^2_{\mathrm{bc}}$ show the designs which are permissible for at least one uncertainty scenario. In both Figures 63 and 64, it is visualized where the component designers can select their target design variables $\check{F}_1$ and $\check{F}_2$ in the worst- and the best-case by dashed lines. If there are uncertainties in controllable variables which are neglected in the computation of optimal CSS, this can lead to the case that a permissible design cannot be guaranteed. Hence, the intervals of the computed optimal CSS Ω^1 and Ω^2 for test-bed problem 1 under interval-type uncertainty in controllable variables have different sizes.

For independent CSS, this stems from the objective of optimizing flexibility for component target design under uncertainties in controllable variables, compare Section 3.4. The smaller $\bar{\delta}_2$, the smaller intervals are computed for independent Ω^2, see Figure 63. This result transfers to the best-case CSS of target designs. Furthermore, the intervals of $\check{\Omega}^1_{\mathrm{wc}}$ and $\check{\Omega}^2_{\mathrm{wc}}$ become smaller with increasing $\bar{\delta}_1$ and $\bar{\delta}_2$ which limits the flexibility of designers in selecting target design variables $\check{F}_1$ and $\check{F}_2$ in the worst-case. However, the intervals of $\check{\Omega}^1_{\mathrm{bc}}$ and $\check{\Omega}^2_{\mathrm{bc}}$ become larger which enlarges the flexibility in the best-case. In case (a), the worst- and best-case CSS coincide with Ω^1 and Ω^2. In case (c), the best-case is not considered and there is no more flexibility in selecting $\check{F}_1$ and $\check{F}_2$ in the worst-case if maximum uncertainty should be tolerable.

For dependent CSS, the decrease of the worst-case complete system solution space causes also a decrease of the intervals of $\check{\Omega}^1_{\mathrm{wc}}$ and $\check{\Omega}^2_{\mathrm{wc}}$, which implies a decrease of the intervals of Ω^1, Ω^2, $\check{\Omega}^1_{\mathrm{bc}}$, and $\check{\Omega}^2_{\mathrm{bc}}$. This guarantees that a target design variable $\check{F}_i$, $i^1 \in \{1,2\}$, is selected for which a target design variable $\check{F}_i \in \Omega^{i^2}$, $i^2 \in \{1,2\}\backslash\{i^1\}$, exists such that the realized design is always permissible. If worst-case CSS were deduced from best-case CSS, both would be larger but also the interpretation would change. In case (c), independent and dependent CSS coincide.

The results for fuzzy-type uncertainty are similar, compare Section 3.5, and are not considered

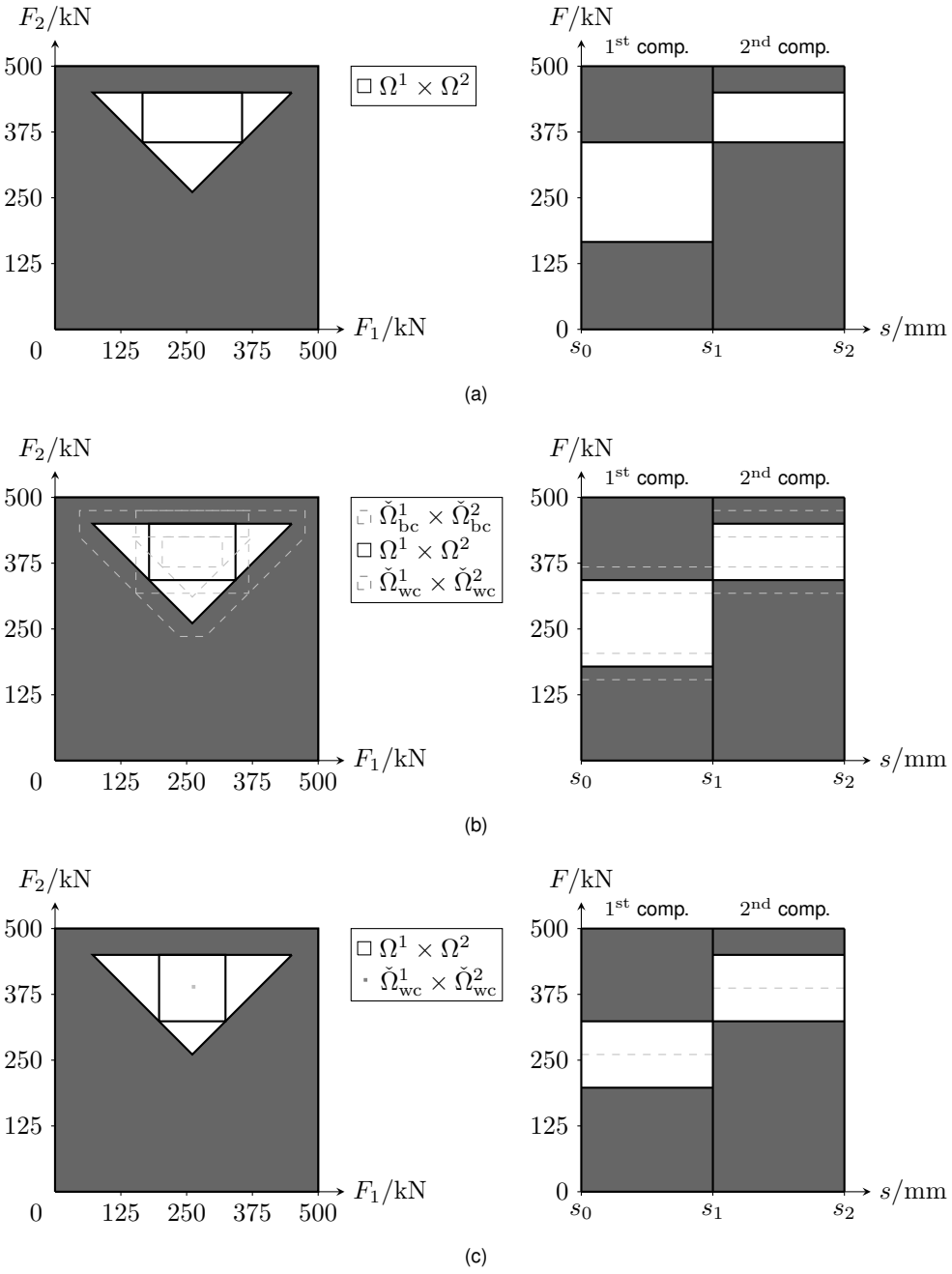


Figure 63 Independent CSS: Optimal independent CSS visualized as geometric shapes in force space (left) and permissible force-deformation characteristics (right) for test-bed problem 1 for which the components have one degree of freedom and (a) no uncertainties, (b) uncertainties with known magnitudes, and (c) uncertainties with unknown magnitudes in controllable variables are considered.

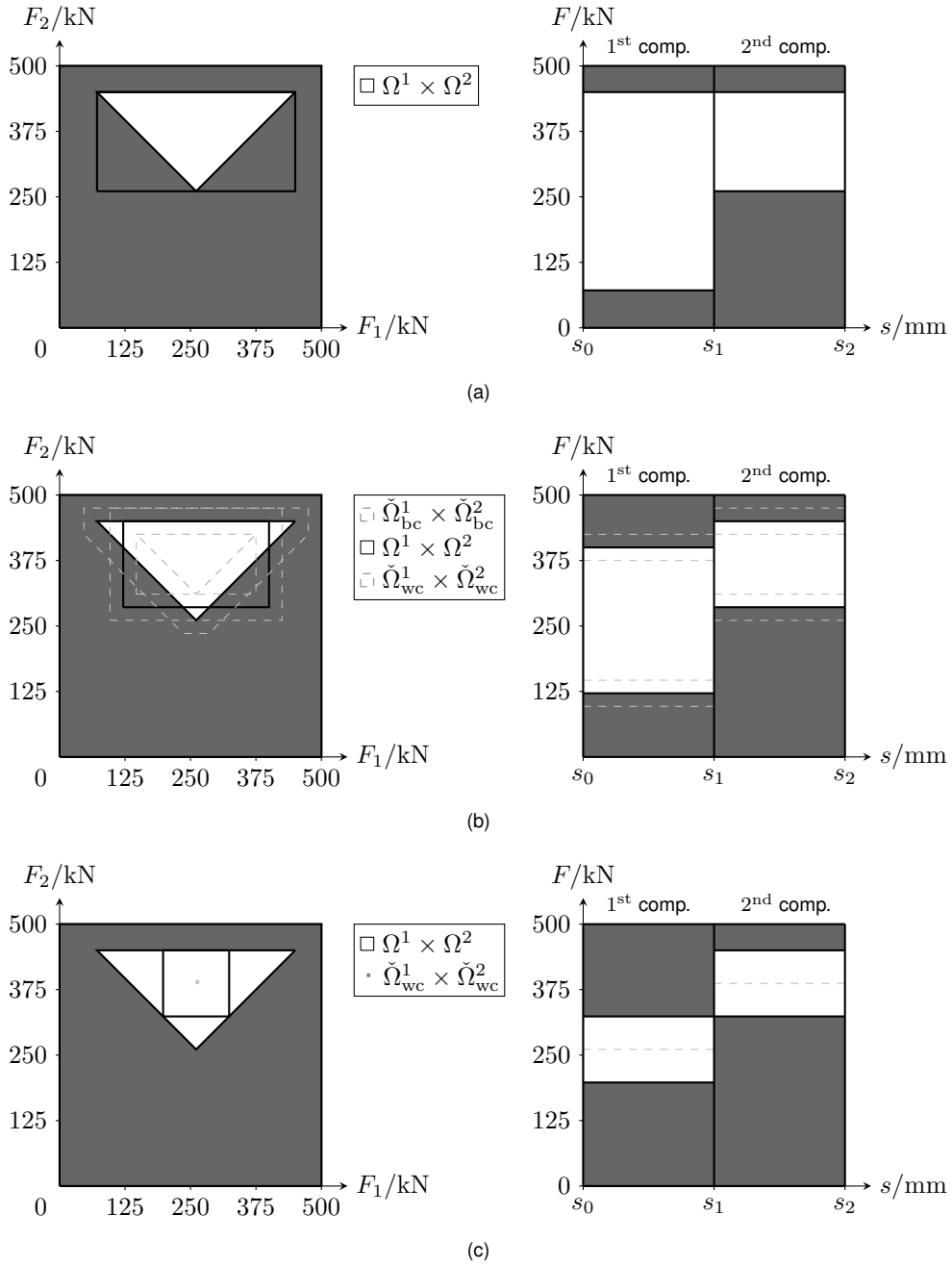


Figure 64 Dependent CSS: Optimal dependent CSS visualized as geometric shapes in force space (left) and permissible force-deformation characteristics (right) for test-bed problem 1 for which the components have one degree of freedom and (a) no uncertainties, (b) uncertainties with known magnitudes, and (c) uncertainties with unknown magnitudes in controllable variables are considered.

for test-bed problem 1 here. Nevertheless, they are investigated for test-bed problem 2 below.

5.4.2. Test Bed 1: Uncertainties in Uncontrollable Parameters

Next, interval-type uncertainty in uncontrollable parameters is taken into account for test-bed problem 1, for which again components with one degree of freedom are considered. These investigations were already published in [26] by the author of this thesis. In order to do so, the values of Table 3 are used as nominal values for the uncontrollable parameters. For the design space parameters given in Table 2, no uncertainty is assumed. In the following, uncertainties in uncontrollable parameters are considered both with the presence and absence of uncertainties in controllable variables. It is distinguished between two cases:

(a) *Uncertainties in uncontrollable parameters only*: There are only uncertainties in uncontrollable parameters. For $p = (m, v_0, a_c)$, it is assumed that $\gamma_l = (50\,\mathrm{kg}, 0.1\,\frac{\mathrm{mm}}{\mathrm{ms}}, 0.01\,\frac{\mathrm{mm}}{\mathrm{ms}^2})$ holds. Furthermore, the uncertainty magnitudes for the controllable variables, and the other uncontrollable parameters are assumed to be zero.

(b) *Uncertainties in both uncontrollable parameters and controllable variables*: In addition to the uncertainties in the uncontrollable parameters given in case (a), uncertainties in the controllable variables are assumed with $\bar{\delta}_1 = \bar{\delta}_2 = 25\,\mathrm{kN}$.

As the constraints are component-wise monotonic in the uncontrollable parameters and linear in the controllable variables, results from Section 4.2 can be used to compute optimal worst-case CSS Ω^1_{wc}, Ω^2_{wc} in case (a) and optimal worst-case CSS of target designs $\check{\Omega}^1_{\mathrm{wc}}$, $\check{\Omega}^2_{\mathrm{wc}}$ in case (b). Then, optimal CSS Ω^1, Ω^2 and optimal best-case CSS Ω^1_{bc}, Ω^2_{bc} can be deduced in (a), and optimal worst-case CSS Ω^1_{wc}, Ω^2_{wc}, optimal best-case CSS Ω^1_{bc}, Ω^2_{bc}, and optimal best-case CSS of target designs $\check{\Omega}^1_{\mathrm{bc}}$, $\check{\Omega}^2_{\mathrm{bc}}$ for independent CSS in (b). For dependent CSS in case (b), optimal best-case CSS of target designs are computed independently as condition (3.37), i.e., $\Omega^k_{\mathrm{wc}} \subseteq \Omega^k_{\mathrm{bc}}$, $k = 1, \ldots, 7$, is fulfilled here. All in all, the optimal independent CSS are shown in Figure 65, and the optimal dependent CSS in Figure 66, both as geometric shapes in force space and regions of permissible force-deformation characteristics.

The worst-case CSS Ω^1_{wc} and Ω^2_{wc} show the permissible designs which are permissible for all uncertainty scenarios, and the best-case CSS Ω^1_{bc} and Ω^2_{bc} show the designs which are permissible for at least one uncertainty scenario. If there are uncertainties in uncontrollable parameters and only the nominal values are used to model the uncertainties, this information is missing, compare Figures 63 and 64.
If uncertainties in controllable variables are present as well, the intervals of $\check{\Omega}^1_{\mathrm{wc}}$, $\check{\Omega}^2_{\mathrm{wc}}$, $\check{\Omega}^1_{\mathrm{bc}}$, and $\check{\Omega}^2_{\mathrm{bc}}$ provide additional information where to select the target design variables and influence the size of the intervals of the worst- and best-case CSS. Thus, for independent CSS, the interval of Ω^1_{wc} is smaller in case (a) than in case (b) and the interval of Ω^2_{wc} is larger in (a) than in (b). This transfers to the best-case CSS Ω^1_{bc} and Ω^2_{bc}. For dependent CSS, the optimal

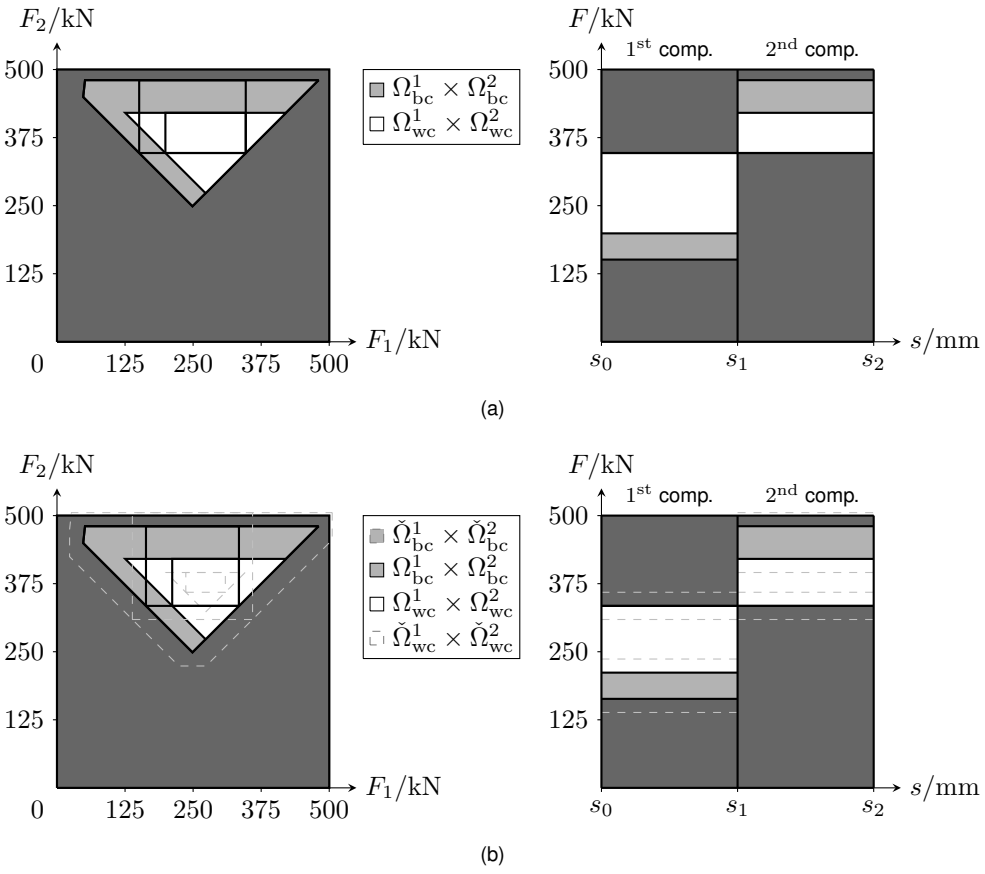


Figure 65 Independent CSS: Optimal independent CSS visualized as geometric shapes in force space (left) and permissible force-deformation characteristics (right) for test-bed problem 1 for which the components have one degree of freedom and uncertainties are considered in (a) uncontrollable parameters only and (b) both uncontrollable parameters and controllable variables.

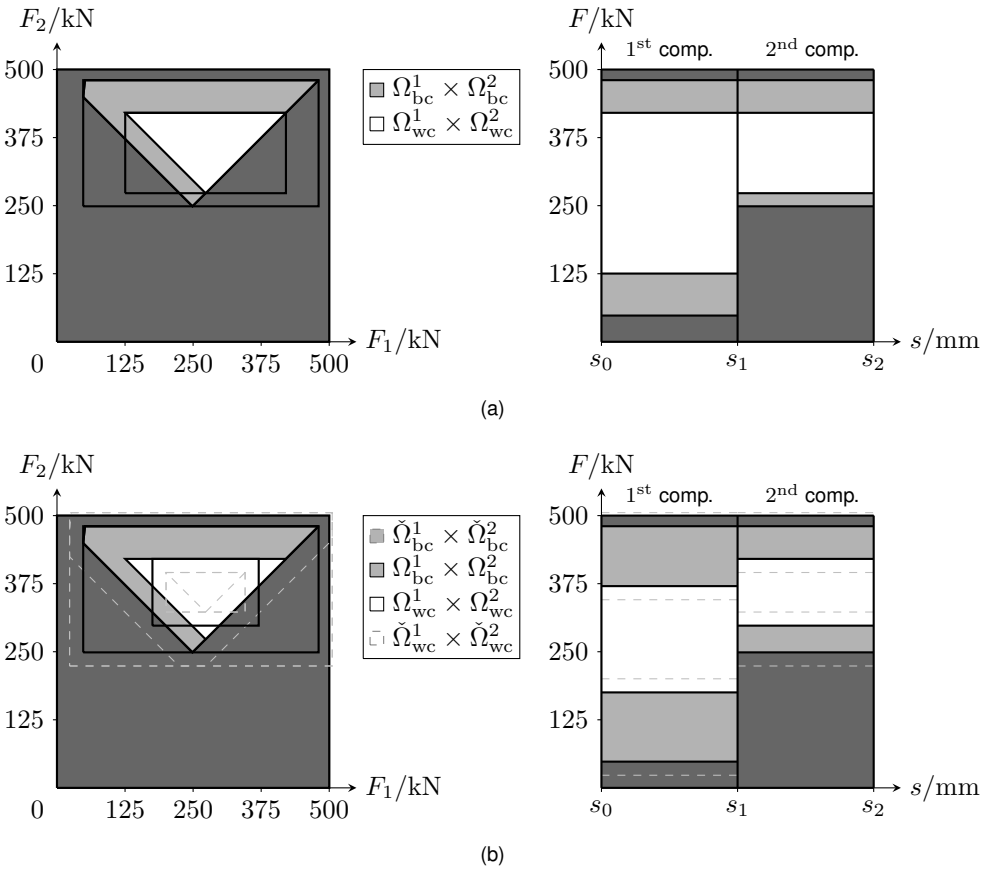


Figure 66 Dependent CSS: Optimal dependent CSS visualized as geometric shapes in force space (left) and permissible force-deformation characteristics (right) for test-bed problem 1 for which the components have one degree of freedom and uncertainties are considered in (a) uncontrollable parameters only and (b) both uncontrollable parameters and controllable variables.

best-case CSS can be computed independently of the optimal worst-case CSS and so can best-case CSS of target designs $\check{\Omega}^1_{\mathrm{bc}}$ and $\check{\Omega}^2_{\mathrm{bc}}$. Note that this was not the case above. Hence, the interpretation of the best-case CSS of target designs is different here. It guarantees that a target design variable $\check{F}_{i^1}$, $i^1 \in \{1,2\}$, is selected for which a target design variable $\check{F}_{i^2} \in \Omega^{i^2}$, $i^2 \in \{1,2\}\backslash\{i^1\}$, exists such that there is at least one uncertainty scenario for which the realized design is permissible.

The effect of component-wise monotonic constraints in the uncontrollable parameters can be seen in the upper left part of the best-case complete system solution space in Figures 63 and 64. For example, there is an additional constraint introduced that guarantees the existence of a mass $m \in [1450\,\mathrm{kg}, 1550\,\mathrm{kg}]$ for which both the constraints (5.1) and (5.3) can be fulfilled. It is neglected in the computation of Ω^1_{bc}, Ω^2_{bc}, $\check{\Omega}^1_{\mathrm{bc}}$, and $\check{\Omega}^2_{\mathrm{bc}}$ using the results of Section 4.2. For independent CSS, this is reasonable here as the designs within $\Omega^1_{\mathrm{bc}} \times \Omega^2_{\mathrm{bc}}$ do not violate these constraints. This is different for dependent CSS, for which the lower bound of Ω^1_{bc} would be lower than visualized in Figures 63 and 64 if the results of Section 4.2 are used.

Note that uncertainties in the uncontrollable section length are not considered here as the integrated plots in Figures 63 and 64 for permissible force-deformation characteristics could not be used in this case. However, there are no differences in the procedure if uncertainties in the section lengths are also incorporated. Similar to above, the results for fuzzy-type uncertainty in uncontrollable variables are not considered for test-bed problem 1 but investigated for test-bed problem 2.

5.4.3. Test Bed 2: Uncertainties in Controllable Variables

In the following, interval- and fuzzy-type uncertainties in controllable variables and uncontrollable parameters are taken into account for test-bed problem 2. The uncertainties in controllable variables and uncontrollable parameters are investigated separately, however, their assembly could be done like above. In addition to limiting the considerations to box-shaped CSS, they are also limited to independent CSS in the following, as the optimal dependent CSS for test-bed problem 2 completely or almost completely cover their corresponding component design spaces, see Figure 62. Nevertheless, the corresponding investigations can be done similarly for dependent CSS.

First, uncertainties in controllable variables are taken into account, and uncertainties in uncontrollable parameters are neglected. Here, two cases are considered for which the magnitudes are assumed to be known:

(a) *Interval-type uncertainties*: The magnitudes of the interval-type uncertainties of the controllable variable are given by $\bar{\delta}_i = 5\,\mathrm{kN}$ for the first load path, i.e., for the first until the third component with $i = 1, \ldots, 15$, and $\bar{\delta}_i = 10\,\mathrm{kN}$ for the second load path, i.e., for the fourth until the seventh component with $i = 16, \ldots, 31$.

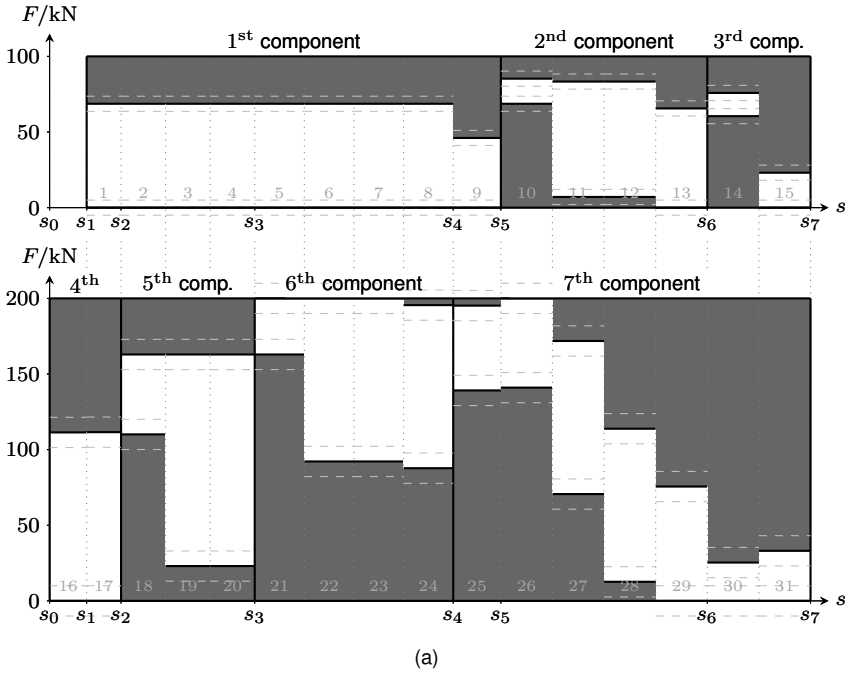
(b) *Fuzzy-type uncertainties*: The magnitudes of the fuzzy-type uncertainties of the controllable

variables are given by $\bar{\delta}_{\alpha=0,i} = 5\,\mathrm{kN}$ and $\bar{\delta}_{\alpha=1,i} = 3\,\mathrm{kN}$ for the first load path, $i = 1, \ldots, 15$, and $\bar{\delta}_{\alpha=0,i} = 10\,\mathrm{kN}$ and $\bar{\delta}_{\alpha=1,i} = 5\,\mathrm{kN}$ for the second load path, $i = 16, \ldots, 31$.

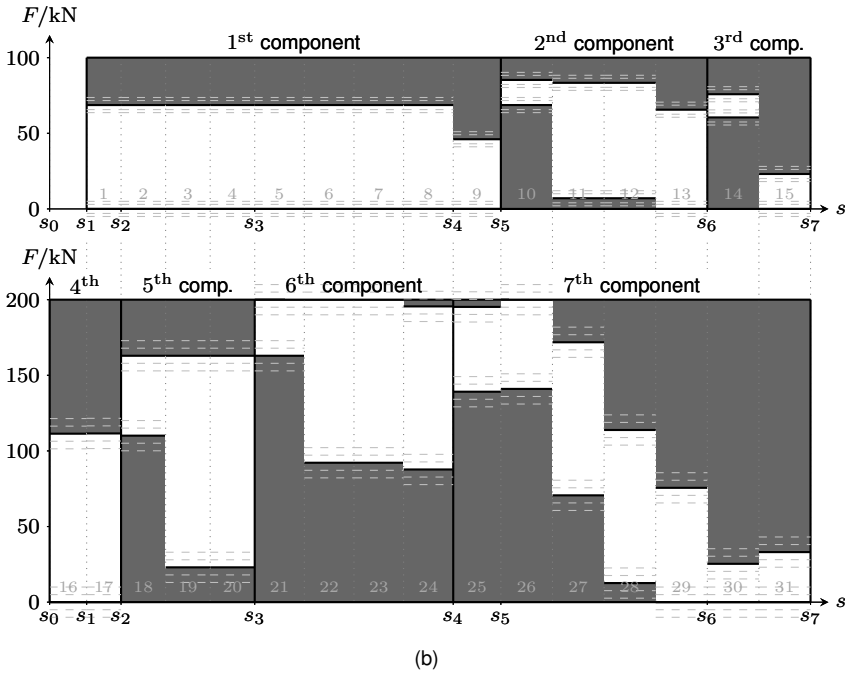
Like for test-bed problem 1, the results from Section 4.2 can be used to compute optimal worst-case CSS of target designs $\check{\Omega}^k_{\mathrm{wc}}$ in case (a), for which CSS Ω^k and best-case CSS of target designs $\check{\Omega}^k_{\mathrm{bc}}$ are deduced from $\check{\Omega}^k_{\mathrm{wc}}$, $k = 1, \ldots, 7$. Similarly, optimal necessity-$\alpha = 1$ CSS of target designs $\check{\Omega}^k_{\mathrm{nec},\alpha=1}$ can be computed in case (b), for which necessity-$\alpha = 1$ CSS of target designs $\check{\Omega}^k_{\mathrm{nec},\alpha=0}$, CSS Ω^k, possibility-1 CSS of target designs $\check{\Omega}^k_{\mathrm{pos},\alpha=1}$, and possibility-$0$ CSS of target designs $\check{\Omega}^k_{\mathrm{pos},\alpha=0}$ are deduced from $\check{\Omega}^k_{\mathrm{nec},\alpha=1}$, $k = 1, \ldots, 7$. The corresponding CSS are shown in Figure 67 as regions of permissible force-deformation characteristics.
Furthermore, the case is considered in which the uncertainty magnitudes are unknown and shall be maximized for the worst-case and the necessity-α case. Here, the weights $\omega_i = 1$ are used for $i = 1, \ldots, 31$. Using the results from Section 4.2, the maximum uncertainty magnitudes can be computed as $\bar{\delta}_i = \bar{\delta}_{\alpha,i} = 14.05\,\mathrm{kN}$, $\alpha \in [0, 1]$, i.e., the worst-case complete system solution space of target designs and all necessity-α complete system solution spaces of target designs coincide. Note that these complete system solution spaces are not singletons here. Thus, a perturbation parameter in the objective function, see (3.34), must be considered when computing the corresponding optimal CSS of target designs with the results from Section 4.2. From them, CSS Ω^k, $k = 1, \ldots, 7$ are deduced. In Figure 67, the corresponding CSS are shown as regions of permissible force-deformation characteristics.

In Figure 67, it can be seen that the CSS Ω^k, $k = 1, \ldots, 7$, coincide for interval- and fuzzy-type uncertainty as well as that the worst-case CSS of target designs match the necessity-1 CSS of target designs, and that the best-case CSS of target designs match the closure of the possibility-0 CSS of target designs. This is because the closures of the supports of the corresponding uncertainty sets are the same, i.e., $\delta_i = \delta_{\alpha=0,i}$ for $i = 1, \ldots, 31$, and interval-type uncertainty can be interpreted as fuzzy-type uncertainty with $\bar{\delta}_{\alpha,i} = \bar{\delta}_i$ for all $\alpha \in [0, 1]$. Moreover, the optimization of the corresponding box-shaped independent CSS is done for the worst-case and the necessity-1 CSS of target designs and the remaining CSS are deduced for which no further optimization is possible. For computed maximum magnitudes in the case of unknown uncertainty magnitudes, the worst-case and the necessity-α CSS of target designs coincide and as well as the CSS Ω^k, $k = 1, \ldots, 7$, which are deduced. This holds as the computed uncertainty magnitudes are all the same, $\alpha \in [0, 1]$.

Overall, the interpretation of the results can be done similarly to test-bed problem 1. The CSS of target designs under interval-type uncertainties provide additional information for component design in the case of uncertainty in controllable variables, i.e., where to select target design variables. This can be even extended to fuzzy-type uncertainty if more information on the uncertainty is available. Depending on the uncertainty magnitudes, the sizes of the intervals of Ω^k, $k = 1, \ldots, 7$, differ. However, there are no fundamental differences in their location. Especially, the narrow intervals for the first design variables of the second and third component,



(a)



(b)

Figure 67 Independent CSS: Optimal box-shaped (a) worst- and best-case and (b) necessity-1, and necessity-0, possibility-1, and possibility-0 independent CSS of target designs visualized as regions of permissible force-deformation characteristics (dashed lines).

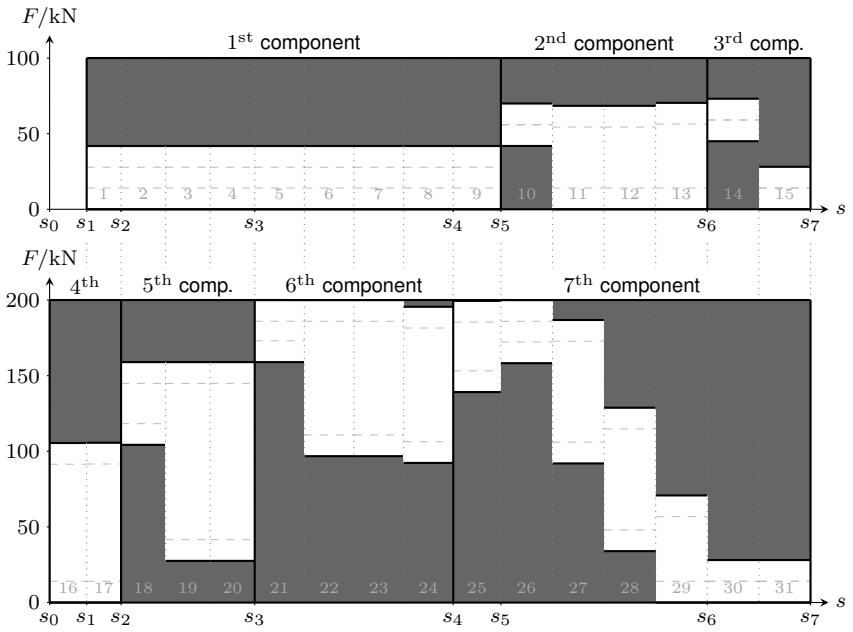


Figure 68 Independent CSS: Optimal box-shaped worst-case independent CSS of target designs visualized as regions of permissible force-deformation characteristics (dashed lines), which corresponds to the optimal box-shaped necessity-α CSS of target designs, $\alpha \in [0, 1]$.

i.e., of F_{10} and F_{14} in the case of no uncertainty, compare Figure 61, are enlarged for known uncertainty magnitudes and even further enlarged for maximized uncertainty magnitudes. Thus, in the case of large uncertainty magnitudes, non-permissible designs can be avoided more efficiently. However, the volume $\Omega^1 \times \cdots \times \Omega^7$, decreases in these cases and it should be decided, based on the available information about the uncertainties, which case is taken into account. Note that the bounds of best-case CSS and possibility-α CSS with $\alpha \in [0, 1]$ violate the design space constraints here, see in Figure 67. This could be avoided by bounding the target design variables additionally as discussed in Section 2.2.
Compared to the other examples of this thesis, there is flexibility for component target design in the case of maximized uncertainty magnitudes. This is reflected in Ω^k, $k = 1, \ldots, 7$, by using a perturbation parameter in the objective function for computing CSS as discussed in Section 3.4. Therefore, the corresponding method of [43], in which no perturbation parameter is considered, is extended.

5.4.4. Test Bed 2: Uncertainties in Uncontrollable Parameters

Next, uncertainties in uncontrollable parameters are considered for test-bed problem 2, and uncertainties in controllable variables are neglected. Here, the provided values for the uncontrollable parameters from Tables 5 and 6 are used as nominal values. Again, it is distinguished between two cases in which interval-type is considered in one of them and fuzzy-type uncertainty in the other. Furthermore, it is $\gamma_l = \gamma_{\alpha=0,l}$, $l = 1, \ldots, q$, i.e., the closures of the supports of the corresponding uncertainty sets are the same:

(a) *Interval-type uncertainties*: For for the magnitudes of the interval-type uncertainties in the

uncontrollable parameters, it is assumed that $\gamma = (0.05p_1, \ldots, 0.05p_{10}, 0.1\,\frac{\text{mm}}{\text{ms}}, 0.01\,\frac{\text{mm}}{\text{ms}^2}, 0.05p_{13}, \ldots, 0.05p_{28}, 0.1(p_{29} - 1), \ldots, 0.1(p_{36} - 1))$ holds for $p = (m_1, \ldots, m_{10}, v_0, a_\text{c}, F_1^\text{add}, \ldots, F_{16}^\text{add}, E_\text{cf}^1, \ldots, E_\text{cf}^7, E_\text{cf}^\text{add})$. In addition, their values are stated in Table 12 and the magnitudes of the remaining uncontrollable parameters are assumed to be zero.

(b) *Fuzzy-type uncertainties*: The magnitudes of the fuzzy-type uncertainties in the uncontrollable parameters in $\gamma_{\alpha=0}$ are assumed to match those in γ of case (a). Furthermore, it is assumed that $\gamma_{\alpha=1} = (0.01p_1, \ldots, 0.01p_{10}, 0.0\,\frac{\text{mm}}{\text{ms}}, 0.00\,\frac{\text{mm}}{\text{ms}^2}, 0.01p_{13}, \ldots, 0.01p_{28}, 0.00\ldots, 0.00)$ holds. In addition, their values are stated in Table 12 again.

As above, the results from Section 4.2 can be used to compute optimal worst-case CSS Ω_wc^k and optimal best-case CSS, Ω_bc^k in case (a), in which condition (3.37) must be fulfilled. Similarly, optimal necessity-α CSS $\Omega_{\text{nec},\alpha}^k$ and optimal possibility-α CSS $\Omega_{\text{pos},\alpha}^k$, can be computed in case (b) in which conditions (3.45)-(3.47) must be fulfilled for $\alpha \in [0, 1]$. The corresponding CSS are shown in Figure 69 as regions of permissible force-deformation characteristics.

Due to similar reasons as discussed for uncontrollable variables, the worst-case CSS coincide with the necessity-1 CSS and the best-case CSS coincide with the possibility-1 CSS here, see Figure 69. Furthermore, the interpretation of the results can be done similarly to test-bed problem 1. Compared to standard CSS, CSS under interval-type uncertainty in uncontrollable parameters provide additional information for component design, i.e., worst- and best-case CSS. This is even further extended considering fuzzy-type uncertainty if more information on the uncertainty is available. For that, necessity-α and possibility-α CSS are obtained. Note that for all component designs within best-case and possibility-α CSS, it must be tested whether at least one $p \in [\check{p} - \gamma, \check{p} + \gamma]$ exists such that all constraints are fulfilled when using the approach of Section 4.2. As this is the case, this strategy to compute the corresponding CSS is appropriate here.
In general, both uncertainties in controllable variables and controllable variables can be taken into account at the same time, which is demonstrated, for example, in [26] and for the test-bed problem 1 above. This yields an overall approach for crash design under epistemic uncertainty. However, an example is skipped here as the results merge the previous results and the visualization becomes more complicated.
In the final section, a MATLAB app for crash design which is capable of creating DSM accounting for epistemic uncertainty and of computing the CSS introduced in this thesis is presented.

5.5. MATLAB App: CSS Solver for Crash Design

This section presents a MATLAB app that computes CSS for crash design using DSM. It was developed for MATLAB version R2019a. In this app, all methods of this thesis are included, i.e., the user can choose between computing box- and arbitrarily-shaped and between independent and dependent CSS. In addition, enhanced DSM can be built using the integrated *DSM Creator*

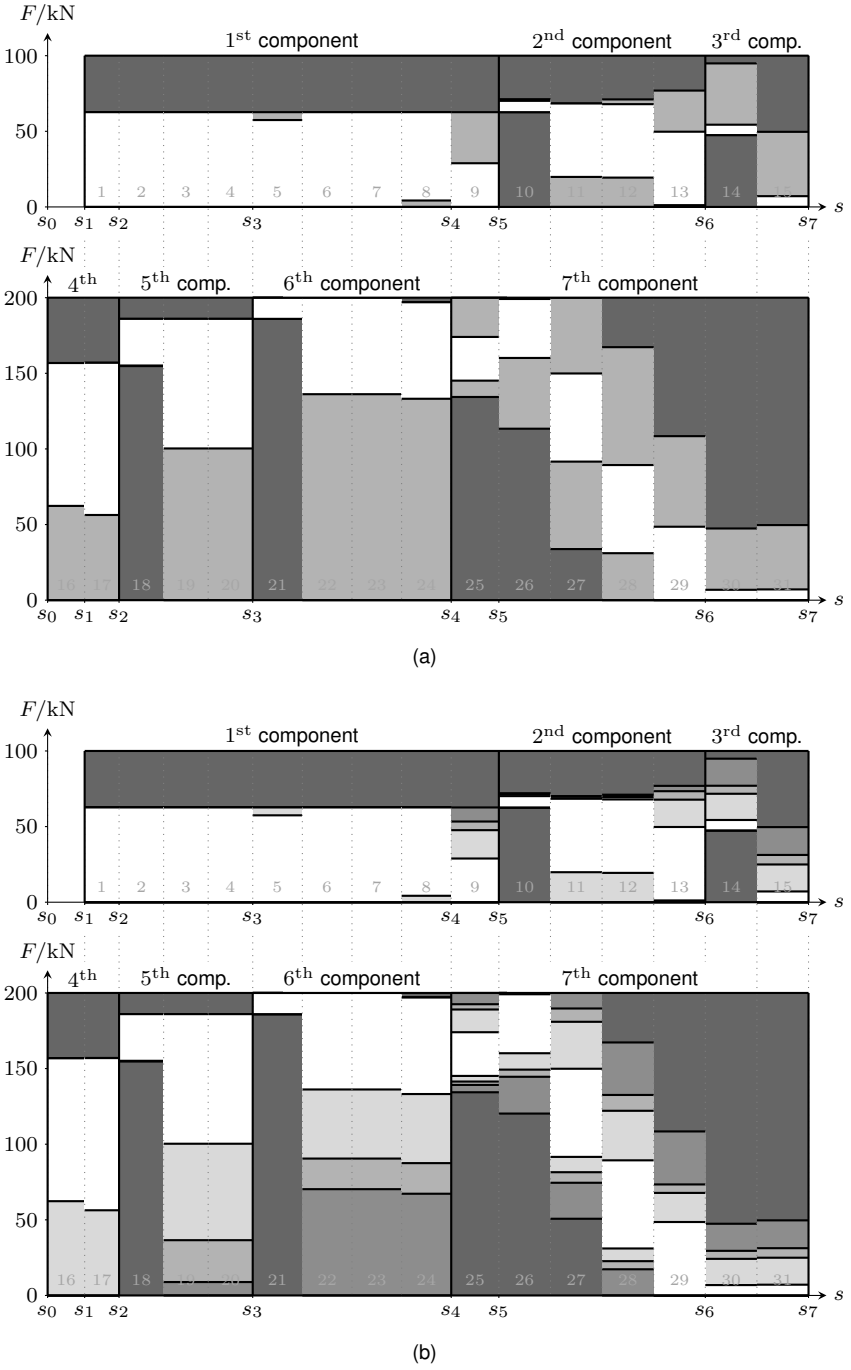


Figure 69 Independent CSS: Optimal box-shaped (a) worst- and best-case and (b) necessity-1, and necessity-0, possibility-1, possibility-0, independent CSS visualized as regions of permissible force-deformation characteristics.

Table 12 Values of the magnitudes γ_l for interval-type and $\gamma_{\alpha=0,l}$ and $\gamma_{\alpha=1,l}$ for fuzzy-type uncertainty of the HONDA ACCORD FE model.

quantity	m_1	m_2	m_3	m_4	m_5	m_6	m_7
value of γ_l, $\gamma_{\alpha=0,l}$	$5\,\mathrm{kg}$	$1\,\mathrm{kg}$	$14\,\mathrm{kg}$	$1\,\mathrm{kg}$	$0\,\mathrm{kg}$	$1\,\mathrm{kg}$	$1\,\mathrm{kg}$
value of $\gamma_{\alpha=1,l}$	$1\,\mathrm{kg}$	$0\,\mathrm{kg}$	$3\,\mathrm{kg}$	$0\,\mathrm{kg}$	$0\,\mathrm{kg}$	$0\,\mathrm{kg}$	$0\,\mathrm{kg}$
quantity	m_8	m_9	m_{10}	v_0	a_c	F_1^add	F_2^add
value of γ_l, $\gamma_{\alpha=0,l}$	$1\,\mathrm{kg}$	$1\,\mathrm{kg}$	$57\,\mathrm{kg}$	$0.1\,\frac{\mathrm{mm}}{\mathrm{ms}}$	$0.01\,\frac{\mathrm{mm}}{\mathrm{ms}^2}$	$0\,\mathrm{kN}$	$1\,\mathrm{kN}$
value of $\gamma_{\alpha=1,l}$	$0\,\mathrm{kg}$	$0\,\mathrm{kg}$	$11\,\mathrm{kg}$	$0.0\,\frac{\mathrm{mm}}{\mathrm{ms}}$	$0.00\,\frac{\mathrm{mm}}{\mathrm{ms}^2}$	$0\,\mathrm{kN}$	$0\,\mathrm{kN}$
quantity	F_3^add	F_4^add	F_5^add	F_6^add	F_7^add	F_8^add	F_9^add
value of γ_l, $\gamma_{\alpha=0,l}$	$4\,\mathrm{kN}$	$6\,\mathrm{kN}$	$9\,\mathrm{kN}$	$9\,\mathrm{kN}$	$8\,\mathrm{kN}$	$7\,\mathrm{kN}$	$6\,\mathrm{kN}$
value of $\gamma_{\alpha=1,l}$	$0\,\mathrm{kN}$	$1\,\mathrm{kN}$	$2\,\mathrm{kN}$	$2\,\mathrm{kN}$	$2\,\mathrm{kN}$	$1\,\mathrm{kN}$	$1\,\mathrm{kN}$
quantity	F_{10}^add	F_{11}^add	F_{12}^add	F_{13}^add	F_{14}^add	F_{15}^add	F_{16}^add
value of γ_l, $\gamma_{\alpha=0,l}$	$7\,\mathrm{kN}$	$4\,\mathrm{kN}$	$6\,\mathrm{kN}$	$9\,\mathrm{kN}$	$12\,\mathrm{kN}$	$12\,\mathrm{kN}$	$14\,\mathrm{kN}$
value of $\gamma_{\alpha=1,l}$	$1\,\mathrm{kN}$	$0\,\mathrm{kN}$	$1\,\mathrm{kN}$	$2\,\mathrm{kN}$	$2\,\mathrm{kN}$	$2\,\mathrm{kN}$	$3\,\mathrm{kN}$
quantity	E_cf^1	E_cf^2	E_cf^3	E_cf^4	E_cf^5	E_cf^6	E_cf^7
value of γ_l, $\gamma_{\alpha=0,l}$	0.01	0.03	0.02	0.02	0.00	0.02	0.00
value of $\gamma_{\alpha=1,l}$	0.00	0.00	0.00	0.00	0.00	0.00	0.00

quantity	$E_\mathrm{cf}^\mathrm{add}$
value of γ_l, $\gamma_{\alpha=0,l}$	0.03
value of $\gamma_{\alpha=1,l}$	0.00

which allows to consider epistemic uncertainty as interval- or fuzzy-type uncertainty. First, the general features of the *CSS Solver* are explained before its advanced settings are discussed

and details of the *DSM Creator* are provided.

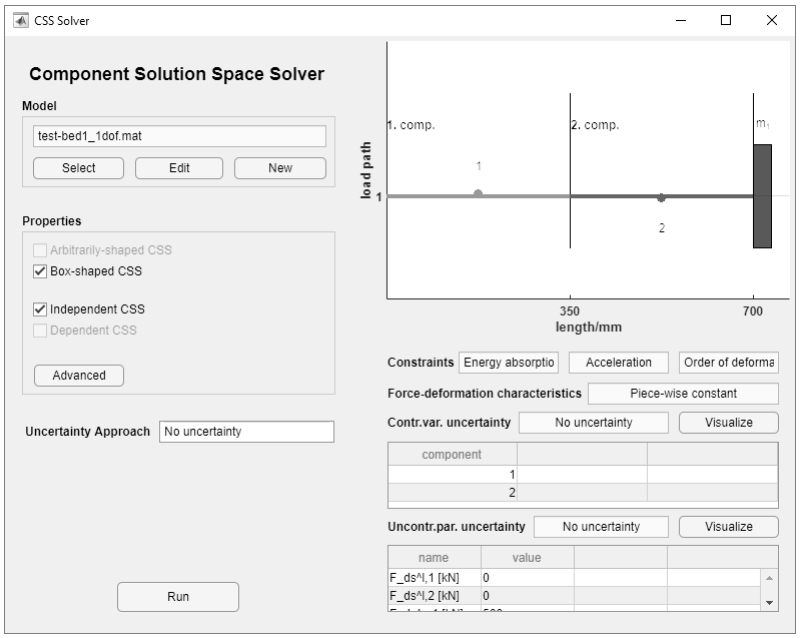
5.5.1. The CSS Solver

The *CSS Solver* is a MATLAB app that was developed within the scope of this thesis which computes CSS for crash design. Together with its integrated *DSM Creator* it consists of over 50 MATLAB files and more than 11,000 lines of source code. This is more than two times the number of text lines of this thesis.

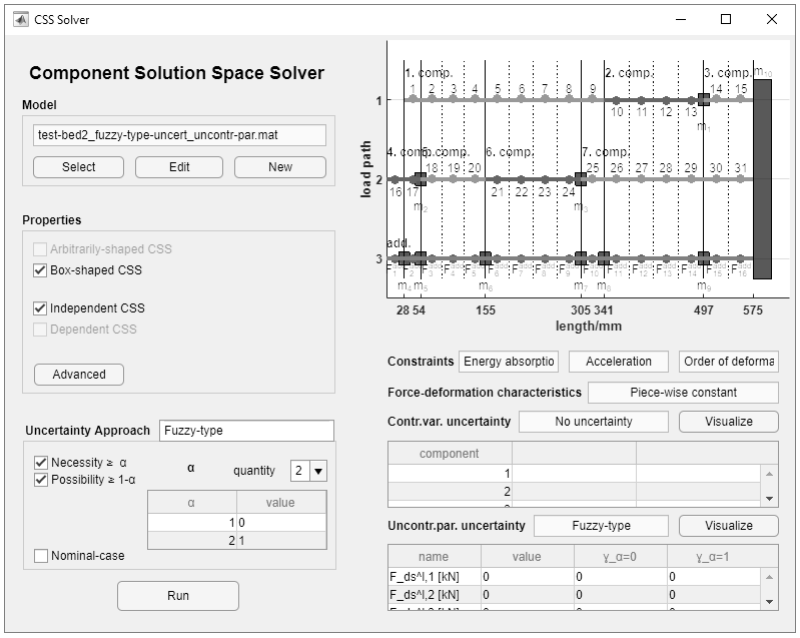
When starting the *CSS Solver*, the graphical user interface of the MATLAB app opens. Here, the user can either select, edit, or create a new DSM. The 'New' button opens the *DSM Creator* in which a new DSM can be built and saved. It is presented below. The 'Select' button opens a list of all previously created DSM of which the user can select a DSM for that he wants to compute CSS. The 'Edit' button enables the user to edit existing DSM. Then, the *DSM Creator* is opened with the previously saved model specifications.
The details on the selected DSM are provided on the right-hand side of the graphical user interface. This includes the visualization of the DSM on the top, the selected constraints, the type of parametrization of the force-deformation characteristic of the components, the type of uncertainty of the controllable variables including their magnitudes, and the type of uncertainty of the uncontrollable parameters including their magnitudes and nominal values. Furthermore, the uncertainties can be visualized according to Section 2.2 by clicking on the corresponding 'Visualize' buttons. In Figure 70, the graphical user interface of the *CSS Solver* with selected test-bed problems 1 and 2 is shown.

After selecting a DSM, the user can specify which type of CSS he wants to compute. He can choose between box- and arbitrarily-shaped and between independent and dependent CSS, yielding four options in total. As there are multiple methods for computing CSS, compare Section 4.3, the user's preferred method can be selected after clicking the 'Advanced' button. Here, also the specifications for the visualization of the CSS can be selected. The advanced settings are discussed in more detail below. If no specifications are selected, the default settings are chosen.

Depending on the type of uncertainty in controllable variables and uncontrollable parameters, the overall uncertainty approach is established. It states 'No uncertainty' if there is neither uncertainty in controllable variables nor uncontrollable parameters, 'Interval-type' if there is interval uncertainty in either controllable variables or uncontrollable parameters and interval-type or no uncertainty in the others, and 'Fuzzy-type' if there is fuzzy-type uncertainty in either controllable variables or uncontrollable parameters. This is because the absence of uncertainty can be treated in the framework of interval-type uncertainty and interval-type uncertainty in



(a)



(b)

Figure 70 Graphical user interface of the *CSS Solver* with the selected DSM of (a) test-bed problem 1 with components with one degree of freedom and (b) test-bed problem 2 under fuzzy-type uncertainties in uncontrollable parameters.

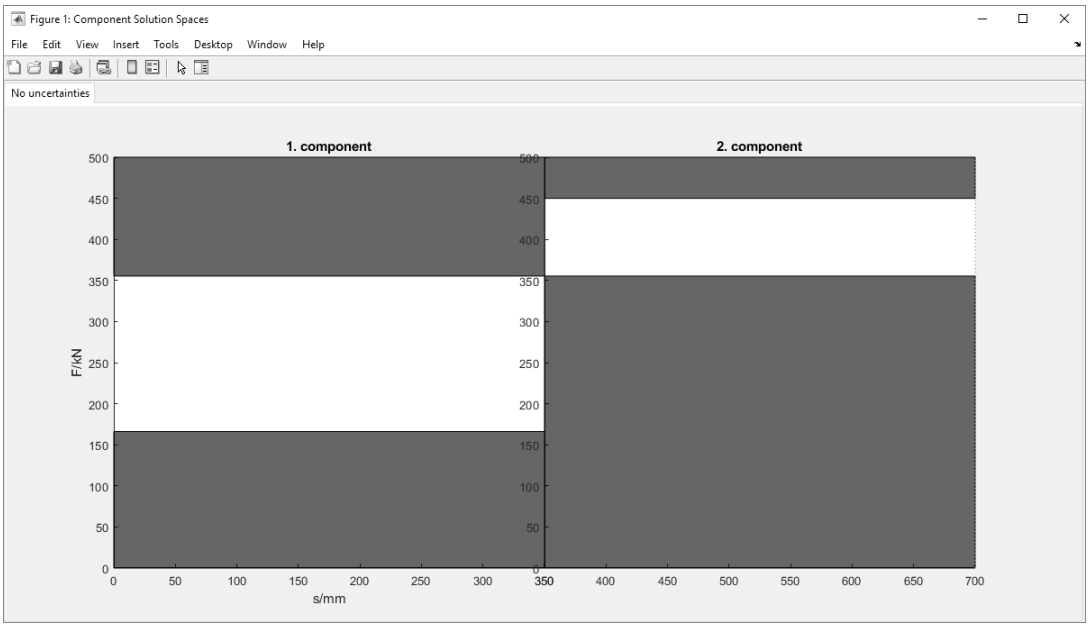
the framework of fuzzy-type uncertainty, see Section 2.2.
For interval-uncertainty, the user can select if both worst- and best-case CSS shall be computed. Moreover, a nominal-case can be selected in which all uncertainty magnitudes are set to zero. For fuzzy-type uncertainty, the user can choose if both necessity-α and possibility-α CSS shall be computed and can specify how many α-levels and which values of α shall be taken into account, $\alpha \in [0, 1]$. Here, a nominal-case can be selected, too.
When these user preferences are set, the corresponding CSS are computed by clicking on the 'Run' button. The results for test-bed problems 1 and 2 using the settings of Figure 70 are visualized in Figure 71. In addition, the numerical values of the results are stored in an extra file.

5.5.2. Advanced Settings for the CSS Solver

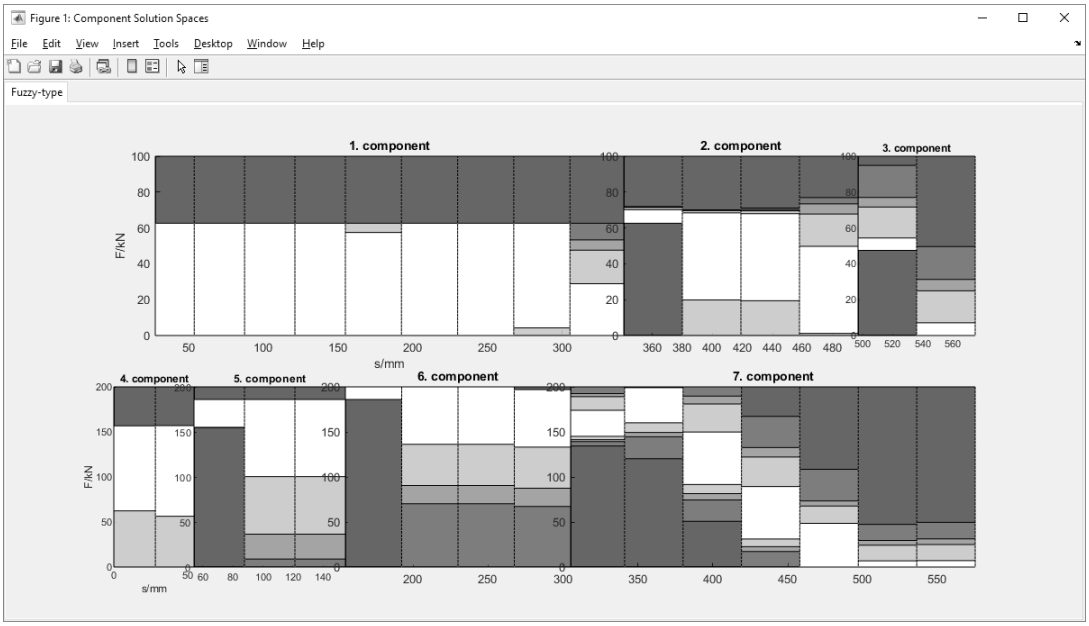
As discussed above, the specifications for the methods for computing and visualizing CSS can be selected in the advanced settings. The 'Advanced' button opens a new window which is customized regarding the type of CSS and the type of uncertainty. In Figures 72 and 73, the advanced settings are visualized.

For arbitrarily-shaped independent CSS, the algorithm to compute CSS can be selected. The default is "no optimization" for which arbitrarily-shaped CSS are computed directly from box-shaped CSS, like described in Section 4.3. In Section 5.3, these arbitrarily-shaped CSS are used as the initial values for optimization. For optimizing CSS, this setting for the initial values is also possible besides obtaining them from solving problem (4.136) and using Equation (4.142), compare Section 4.3. For computing the volume of the CSS, either the exact method can be used, or it can be approximated using Monte Carlo sampling, see also Section 4.3 for details. If the sampling method is selected, the number of sampling points per component can be specified as well as if the sampling shall be done in the whole design space or locally in the neighborhood of the current iteration. For the local option, the size of this neighborhood and the number of optimization runs can be specified additionally. Note that the sampling is only created once for each optimization run, as discussed in Section 4.3.
For box-shaped dependent CSS, the user can choose between the direct method and the two-step method, see Section 4.3. Here, the default is the direct method. For box-shaped independent and arbitrarily-shaped dependent CSS, no optimization settings can be selected.

For all types of CSS, the CSS of target designs can be visualized as dashed lines, as done throughout this thesis. In the case of no uncertainty in controllable variables, the CSS of target designs coincide with the CSS. Regarding the coloring of CSS, the user can either select the coloring scheme of this thesis, for which the outside of CSS is colored, or select to color their bounds, like done, for example, in [25, 26, 27]. Using the *CSS Solver*, the CSS are always visualized as regions of permissible or possibly permissible force-deformation characteristics. Their plots can be grouped in accordance with the DSM, like done in this

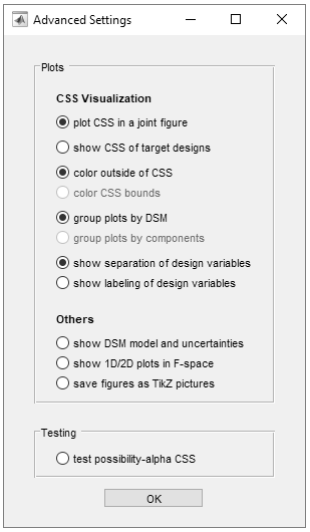


(a)

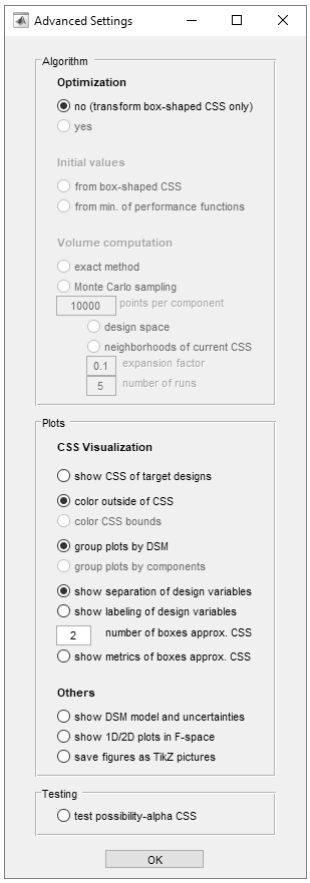


(b)

Figure 71 Visualized CSS using the *CSS Solver* for (a) test-bed problem 1 with components with one degree of freedom and (b) test-bed problem 2 under fuzzy-type uncertainty in uncontrollable parameters.

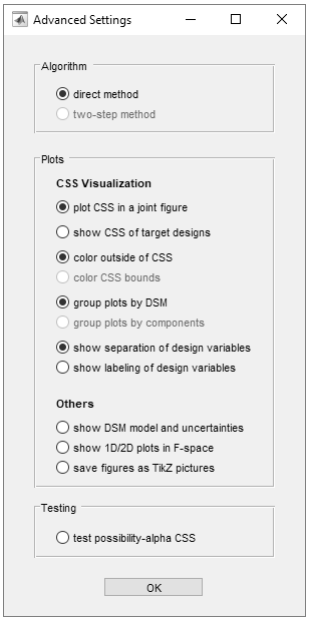


(a)

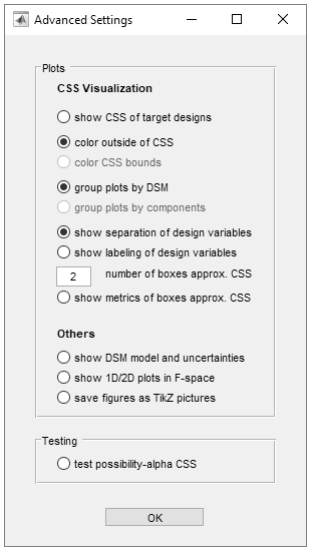


(b)

Figure 72 Independent CSS: Advanced settings for computing, visualizing, and testing (a) box-shaped and (b) arbitrarily-shaped independent CSS



(a)



(b)

Figure 73 Dependent CSS: Advanced settings for computing, visualizing, and testing (a) box-shaped and (b) arbitrarily-shaped dependent CSS

thesis, or separated into component plots, see [27]. Moreover, the user can select to include dotted lines into the CSS plots to mark new segments, i.e., design variables. These describe the parametrization of the force-deformation characteristics and can be labeled numerically by selecting the corresponding option.
Besides the visualization of CSS, the user can also visualize the DSM model and the uncertainties in controllable variables and uncontrollable parameters in separate figures, show the CSS as geometrical shapes for one- or two-dimensional CSS, and save all figures as TikZ pictures.
In the case of arbitrarily-shaped CSS, the number of boxes, i.e., regions of possibly permissible characteristics, describing the CSS can be selected. Also, the key metrics, given by Equations (5.28) and (5.29) can be displayed.
If uncertainties are present, worst- and best-case or necessity-α and possibility-α CSS can be visualized in one plot, $\alpha \in [0, 1]$. Here, the conditions (3.37) in the case of interval-type uncertainty and conditions (3.45) - (3.47) in the case of fuzzy-type uncertainty are applied. As best-case CSS and possibility-α CSS, are computed using the simplifications of Section 4.2, it can be tested whether they contain non-permissible designs, $\alpha \in [0, 1]$. The windows showing the advanced settings for the different types of CSS under fuzzy-type uncertainty are visualized in Figures 72 and 73.

5.5.3. The DSM Creator

A DSM can be created by the *DSM Creator*, which uses again a graphical user interface. The details of the models can be specified in the 'Geometry', 'Model', Contr.var.', and 'Uncontr.par.' tabs. It can be saved by clicking the 'Save as' button, for which a name for the model can be provided. If the 'Save as Temp' button is clicked, the model is saved as a temporary model. Furthermore, the details of the model are presented on the right-hand side of the graphical interface similarly to the right-hand side of the *CSS Solver*. The 'Update' button updates the right-hand side if any changes are applied to the model. In case there is an error in defining the DSM, there is an error message stating the specific error when clicking on any of the 'Save as Temp', 'Save', or 'Update' button.

After starting the *DSM Creator*, the 'Geometry' tab is opened. Here, the number of load paths, sections, components, and discretized masses must be provided. Furthermore, an additional load path can be included for building an enhanced DSM. For the components, the user must enter their associated load paths and the sections of their front and rear ends. Similarly, the associated load path and the location must be provided for the discretized masses, too. The layout of the 'Geometry' tab of the *DSM Creator* is shown in Figure 74.

Next, the constraints that shall be considered can be selected in the 'Model' tab. Here, the constraints on the minimum energy absorption, the maximum acceleration, and the progres-

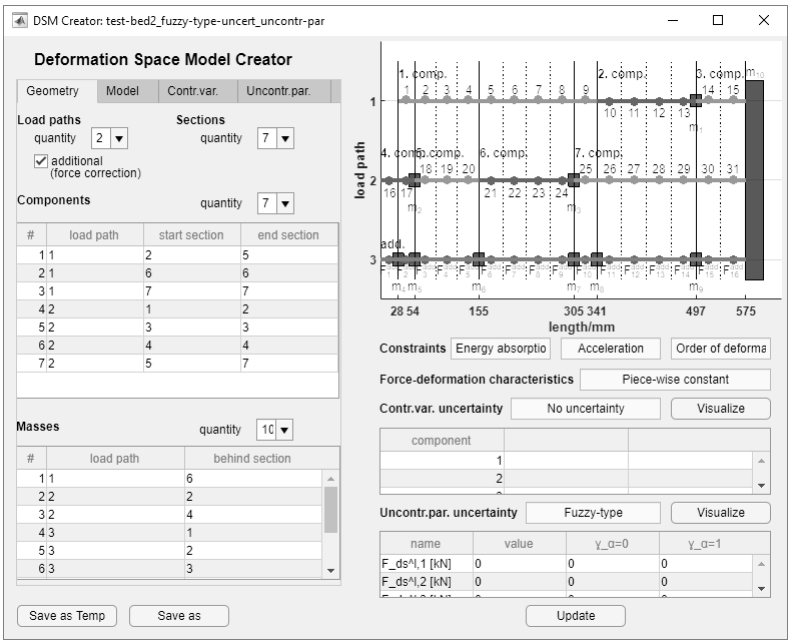


Figure 74 The 'Geometry' tab of the *DSM Creator* with specifications for test-bed problem 2 under fuzzy-type uncertainties in uncontrollable parameters.

sive order of deformation can be activated. Furthermore, the user can decide to include energy-correction factors to build an enhanced DSM. The layout of the 'Model' tab of the *DSM Creator* is shown in Figure 75.

The 'Contr.var.' tab considers the controllable variables. Here, the user can select the type of force-deformation characteristics for computing CSS, i.e., piece-wise constant or linear force-deformation characteristics. Whereas only piece-wise constant characteristics are considered in this thesis, piece-wise linear characteristics are considered, e.g., in [27]. For any section, it can be specified how many design variables shall be taken into account, which defines the number of equidistant segments per section.
Furthermore, the type of uncertainty can be specified by the user. Besides the option of absence of uncertainty, there is the option of interval- and fuzzy-type uncertainty with known magnitudes for which δ^k or $\delta^k_{\alpha=0}$ and $\delta^k_{\alpha=1}$, $k = 1, \ldots, n$, must be provided. Furthermore, there is also the option of unknown magnitudes for which weighting factors ω^k, $k = 1, \ldots, n$, must be provided. Note that δ^k, $\delta^k_{\alpha=0}$, $\delta^k_{\alpha=1}$, and ω^k, $k = 1, \ldots, n$, assume the same values for all design variables of one component. The layout of the 'Contr.var.' tab of the *DSM Creator* is shown in Figure 76.

The 'Uncontr.par.' tab considers the uncontrollable parameters. The user must specify the type of uncertainty first, i.e., no uncertainty, interval-type uncertainty, or fuzzy-type uncertainty. Then, he must enter the nominal values for all uncontrollable parameters along with the magnitudes γ_l for interval-type uncertainty and $\gamma_{\alpha=0,l}$ and $\gamma_{\alpha=1,l}$ for fuzzy-type uncertainty,

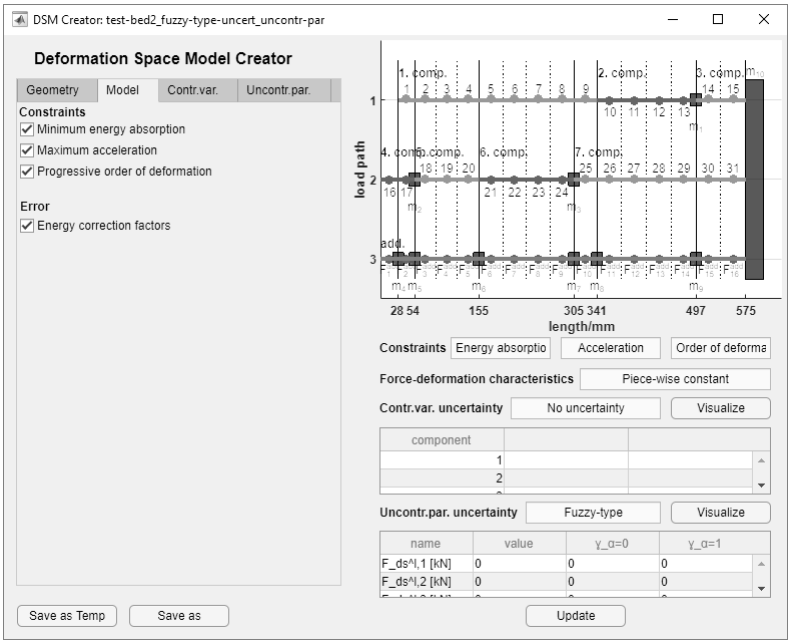


Figure 75 The 'Model' tab of the *DSM Creator* with specifications for test-bed problem 2 under fuzzy-type uncertainties in uncontrollable parameters.

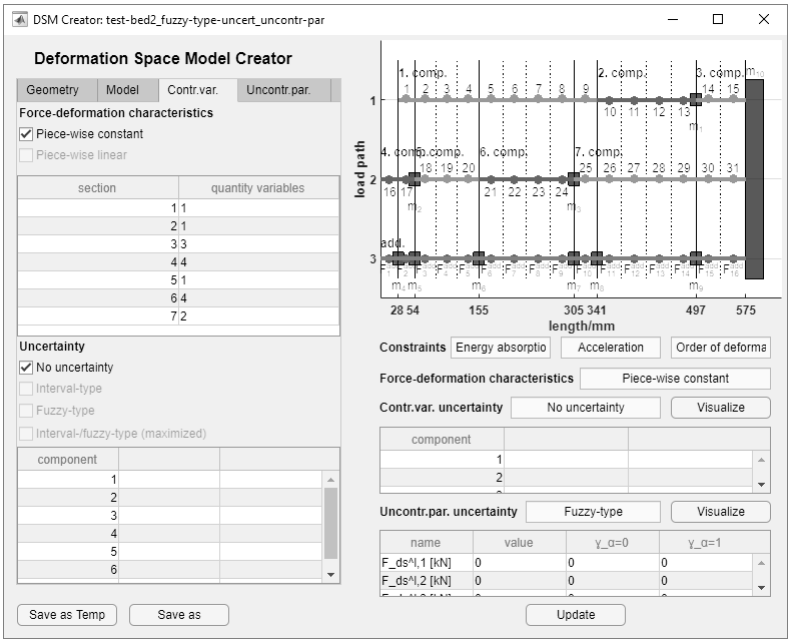


Figure 76 The 'Contr.var.' tab of the *DSM Creator* with specifications for test-bed problem 2 under fuzzy-type uncertainties in uncontrollable parameters.

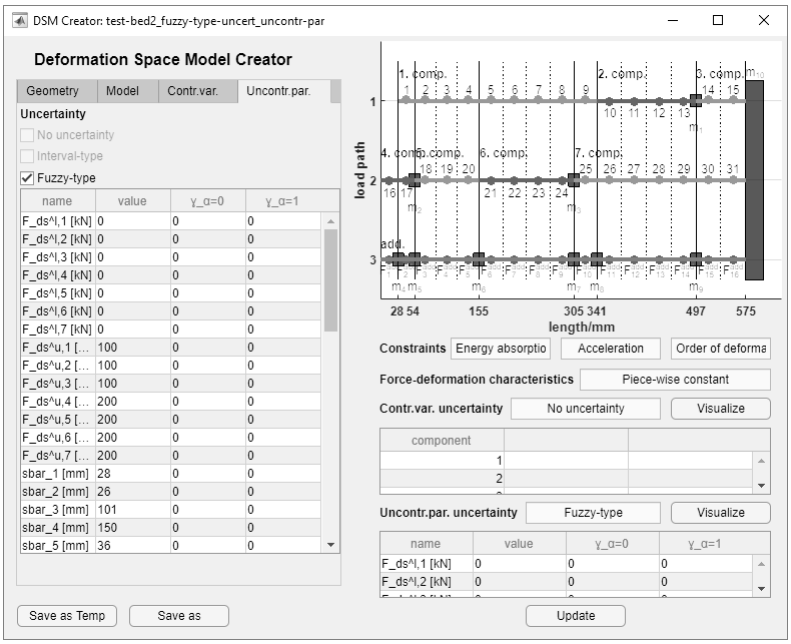


Figure 77 The 'Uncontr.par.' tab of the *DSM Creator* with specifications for test-bed problem 2 under fuzzy-type uncertainties in uncontrollable parameters.

$l = 1, \ldots, p$. Here, also the values that bound the design space are included together with the section lengths, the masses, the initial velocity, and the critical acceleration. Furthermore, the force values of the additional path and the energy-correction factors must be defined when considering enhanced DSM. The layout of the 'Uncontr.par.' tab of the *DSM Creator* is shown in Figure 77.

6. CONCLUSIONS

This section discusses the outcome of this thesis which aims to provide a methodology for decoupled design decisions in systems engineering under epistemic uncertainty. Therefore, the proposed methodology is reviewed in terms of how far its demands are fulfilled, and its limitations are addressed. Afterward, an outlook for further research is given and the major results of this thesis are summarized.

Discussion

The main goal of this thesis was to provide a methodology for decoupled design decisions in systems engineering that enables optimal flexibility for component design under epistemic uncertainty. Based on mathematical design models, fundamentals for decoupled design decisions were discussed, for which coupled design decisions were taken into account, too. In Chapter 2, the state of the art incorporating the modeling of epistemic uncertainties as intervals and fuzzy sets, robust optimization problems, and approaches for decoupled design decisions were reviewed. In order to obtain a complete framework, the methods for robust optimization were complemented and research gaps in the approaches for decoupled design decisions were identified. In the following it is discussed, if the proposed methodology in Chapter 3 including the algorithms of Chapter 4 are appropriate to fulfill the demands stated in Chapter 1. First, the particular contributions to literature are considered after summarizing how the proposed methodology...

... deploys a general framework to classify decoupled design decisions: In this thesis, it is distinguished between independent-decoupled decisions and dependent-decoupled decisions. In the case of independent-decoupled decisions, designers make decisions independent on each other. There is no flow of information between the single design decisions, which can be understood as a concurrent engineering approach. In the case of dependent-decoupled decisions, designers make decisions dependent on each other and one after the other. There is a one-way flow of information between the single design decisions, which can be understood as a traditional engineering approach. This is delimited from coupled design decisions for which there is a two-way flow of information between the single design decisions.

This framework helps systems engineers to structure the development of a system. Depending on the problem and the possibilities to design, they can select if independent-decoupled, dependent-decoupled, or coupled design decisions are preferred. These options were missing in previous approaches for decoupled design decisions, and in this regard, they enhance systems engineering.

... provides flexibility for decoupled design decisions: Based on mathematical design models and constraints on design variables and system responses, the complete system solution space can be computed. This is the set of all permissible system designs which fulfill the constraints. In order to provide flexibility for decoupled design decisions, sets for the corresponding design variables containing design alternatives for one designer must be provided. Independent-decoupled decisions are enabled if the Cartesian product of these sets is a subset of the complete system solution space. Dependent-decoupled decisions are enabled if the set for the first decision is a subset of the projection of the complete system solution space. For the subsequent decisions, this procedure can be repeated with the updated complete system solution space. In this thesis, flexibility for decoupled design decisions is optimized by maximizing the volume of these subsets.

Flexibility for decoupled design decisions enables designers to choose between different design alternatives, which helps to improve the overall design. Furthermore, uncertainties can be circumvented by increased flexibility and the design decisions can also be changed a posteriori. The quantification of flexibility in this thesis is consistent with previous approaches from the literature and extends them to the proposed problem statements of this thesis.

... decouples the decisions based on the hierarchical structure of the system: It is assumed that the system comprises several components which relate to the design variables. Hence, the design variables can be grouped as component designs at the component level. Furthermore, multi-level systems are defined for which the components form subsystems themselves. As the multi-level systems regarded in this thesis can be reduced to two-level systems, the focus is put on two-level systems, which consist of a system and a component level. In order to enable decoupled design decisions between the components, sets for the component design, which are called component solution spaces, must be provided. Here, the geometric shapes of the sets can be predefined, e.g., as boxes yielding box-shaped component solution spaces, or can be optimized yielding arbitrarily-shaped component solution spaces. These two types are considered for both independent- and dependent-decoupled design decisions in this thesis.

Taking the hierarchical structure of a system that consists of components for decoupled design decisions into account allows the deployment of component designers. Thus, the development of the system is aligned with its structure, which may yield a more efficient and cost-saving design process than in previous approaches that completely decouple the decisions for the single design variables. Box-shaped component solution spaces come along with a relatively simple computation and an easy visualization of design alternatives. As they often lack flexibility, arbitrarily-shaped component solution can improve this drawback. Hence, constraint relaxation strategies which are proposed in the literature to increase flexibility for box-shaped solution spaces yielding also non-permissible designs, see, e.g., [84], may not be required. The testing, if a design is permissible or not, can always be done at the component level for component solution spaces. For constraint relaxation strategies this must be done at the system level.

... incorporates the treatment of epistemic uncertainty: Epistemic uncertainties addressing the early design phase are modeled as either intervals, treated in the framework of interval analysis, or fuzzy sets, treated in the framework of possibility theory. These types of uncertainties can be incorporated into the mathematical design models as uncertainties in controllable design variables or uncertainties in uncontrollable parameters. In the case of interval-type uncertainty, worst-case complete system solution spaces which account for all realizations of the uncertainties and best-case complete system solution spaces which account for at least one realization of the uncertainties are yielded. In the case of fuzzy-type uncertainty, necessity-α complete system solution spaces, for which the necessity that a system design is permissible is at least α, and possibility-α complete system solution spaces, for which the possibility that a system design is permissible is at least α, are yielded. Without loss of generality, the considerations of the membership functions of the fuzzy sets are limited to membership functions parametrized by two uncertainty magnitudes as a simple extension of the interval modeling. From these different complete system solution spaces, corresponding component solution spaces that provide flexibility for component design under epistemic uncertainty can be computed.

Incorporating epistemic uncertainties into the proposed methodology, i.e., into the computation of component solution spaces, yield different design regions. Thus, designers can base decisions on whether the design should be sufficient for all or at least one uncertainty scenario in the case of interval-type uncertainty. In the case of fuzzy-type uncertainty, they can base decisions on a minimum necessity or possibility for a design to be permissible. This opportunity was missing in the previous approaches for decoupled design decisions and enhances systems engineering under epistemic uncertainty.

... provides algorithms for the numerical computation of the methods: For a numerical computation of component solution spaces under absence and presence of uncertainty, the considerations in this thesis are limited to specific system performance functions of the mathematical design models. These are linear in the controllable variables and monotonic in the uncontrollable parameters for which the corresponding complete system solution spaces form convex polytopes. It is shown that the optimization problems to compute component solution spaces can be simplified and become convex optimization problems, which can be solved using local numerical optimization solvers, e.g., provided by the software MATLAB, in this case. As the problem statements include also the computation of the volumes of the component solution spaces and the projection of the complete system solution space, suitable algorithms from the literature for this regard are also presented.

The proposed algorithms enable the numerical computation of component solutions spaces for system performance functions that are linear in the controllable variables and monotonic in the uncontrollable parameters. It is demonstrated how complex problem statements that use sets as optimization variables can be simplified to problem statements that use optimization variables in real coordinate space. In the considered examples, the algorithms outperform state-of-the-art algorithms for particular problem statements in terms of computation time and maximized flexibility.

... is applicable to realistic problems in systems engineering: In this thesis, the methodology is applied to crash design problems. Here, deformation space models which provide constraints for controllable force-deformation characteristics of the components are used. The constraints incorporate system performance functions which are linear in the controllable variables and monotonic in the uncontrollable parameters and hence allow the use of the considered numerical computation. In order to compare the different types of component solution spaces, one simple and one realistic test-bed crash problem are proposed. For the realistic test-bed problem, the existing deformation space modeling is enhanced by introducing energy- and acceleration-corrected deformation space models which are calibrated at results from a finite-element-method simulation in the software LS-DYNA.
By applying the methodology to realistic problems in crash design, the potential for a decomposed design process in systems engineering is demonstrated. As the accuracy of the results is always limited by the quality of the mathematical design model, the proposed, enhanced deformation space models are used here. They yield more reasonable component solution spaces for crash design compared to state-of-the-art deformation space models and, therefore, enhance the design for crashworthiness. In general, the proposed methodology is not limited to crash design and can be transferred to other problems in systems engineering for which a component-based development and the incorporation of epistemic uncertainty is appreciated.

Overall, the proposed methodology for decoupled design decisions, which provides optimal flexibility for component design in systems engineering under epistemic uncertainties, complies with its demands. It is able to close significant research gaps and enhances systems engineering regarding the design of components. As all details of the numerical computation of component solution spaces including parameter values are provided, all results of this thesis can be reproduced. Furthermore, the methodology can be directly applied to engineering problems which incorporate system performance functions that are linear in the controllable variables and monotonic in the uncontrollable parameters by using the proposed algorithms. For systems engineering problems with different performance functions, the problem statements of the methodology in its general form can be used as well. However, appropriate algorithms must be proposed, before component solution spaces can be computed. This leads to the limitations of the methodology, which are addressed subsequently and guide to the outlook of this thesis in the next section. Note that the demands of Chapter 1 for the proposed methodology are not discussed here.

The methodology incorporates various simplifications to avoid unnecessary complexity. First of all, it shall be mentioned that the proposed methodology is based on mathematical design models which assume their inputs, controllable design variables and uncontrollable parameters, to be continuous in real coordinate space. If this is not the case for a systems engineering

problem, the methods of this thesis cannot be used in general or it must be investigated how they can be transferred.
Regarding the framework of decoupled decisions for component design, only completely independent- and dependent-decoupled decision approaches were considered. A different sequencing of the decisions for component design is also reasonable for systems engineering, i.e., mixed approaches that take both independent- and dependent-decoupled decisions between the components into account. This was only mentioned briefly in this thesis but could be improved by merging the corresponding problem statements. Hence, this limitation is not that crucial.
Moreover, only the volumes of the component solution spaces or the volumes of their Cartesian product were used as flexibility measures in this thesis. Although it was discussed that the choice of flexibility measures should depend on the use case and examples were provided, embedding different measures into the methodology is lacking. Other measures may also circumvent the long computation time for optimizing high-dimensional volumes of arbitrarily-shaped component solution spaces.
In this thesis, it was assumed that a system consists of components, which are clearly separable from each other, and that each design variable can be assigned to one of them. In general, this is not the case for all systems. Components might not be separable, or even identifiable as such, and multiple components might share one design variable. This discussion can be extended to the multi-level systems considered in this thesis. Furthermore, the assumption of rectangular design spaces in which all design variables can be realized equally well can be assessed critically. Hence, for each system design model, it must be analyzed first, if the methodology of this thesis is applicable. If this is not the case, it must be investigated again how the methods can be transferred.
The modeling of epistemic uncertainty was limited to intervals and fuzzy sets in this thesis. In addition, further limitations were made like the assumption of fully uncorrelated uncertainties and the limitation to uncertainties in controllable variables and uncontrollable parameters, and to simple membership functions for fuzzy-type uncertainty. For a methodology that accounts for further types of epistemic uncertainty, these limitations must be abolished. In particular, model uncertainty should also be taken into account as the accuracy of a system design depends on the underlying mathematical design model, for which uncertainties usually exist. Nevertheless, the considered uncertainties can cover various uncertainty scenarios and provide first meaningful results for decoupled design decisions under epistemic uncertainty.
As discussed above, only algorithms to compute component solution spaces for system performance functions that are linear in the controllable variables and monotonic in the uncontrollable parameters were provided in this thesis. On the one hand, this limits the applicability of the methodology to problems in systems engineering and new algorithms need to be proposed for different performance functions. On the other hand, first-order approximations, i.e., linear performance functions, are often used for engineering applications, and in doing so, they can be incorporated, too. Moreover, the provided algorithms can be seen as the first tools to compute component solution spaces.
For the application of the methodology to crash design, the considerations were limited to

deformation space models. This approach could be critically discussed in general. However, this criticism is not addressed here. In contrast to the enhanced modeling with calibrated uncontrollable parameters, the uncertainty magnitudes were only assumed in this thesis. Therefore, they might be inappropriate for the considered crash design problem. This limits the validity of the results. For complete applicability of the proposed approach to crash design, it must also be discussed how component designs for deformation space models can be optimized after decoupling the decisions. This is addressed in the next section together with further topics for future research.

Outlook

In the following, major topics for further research are proposed. First, topics that relate to the proposed methodology for decoupled design decisions and may improve its discussed drawbacks are considered. Then, fundamentals for the optimization of component designs for crashworthiness, using enhanced deformation space models and component solution spaces, are discussed.

Sequencing strategies for the decoupled decisions could be further investigated in combination with different flexibility measures for component solution spaces. The sequencing strategy determines when each component designer makes his decision. This can be simultaneous to others or consecutive. Thereby, the applicability of the methodology in systems engineering, for example, for large organizations, could be improved. The sequencing strategy and the definition of a flexibility measure should be based on, e.g., the engineering problem, the structure of the organization with its component designers, and economic and regulatory factors. Overall, this can be considered as a complex optimization problem that requires a deliberate framework for its general applicability.
Moreover, the complexity of the mathematical design model could be increased. This could further help to improve the applicability of the methodology in systems engineering. As discussed, more complex system performance functions could be incorporated for which appropriate algorithms to compute component solution spaces must be proposed. Besides that, mathematical design models could be extended to account for, e.g., discrete, time-dependent, or spatially dependent variables. The force-deformation characteristics considered for deformation space models are an example of such variables. In this thesis, they were parametrized, i.e., reduced to continuous variables in real coordinate space. However, different approaches might be investigated yielding component solution spaces for functional relations.
In addition, a more general modeling of epistemic uncertainty could be incorporated. Thus, the methodology could also account for correlated uncertainties, model uncertainties, and more complex membership functions of fuzzy sets, for instance. Further investigations may investigate decoupled design decisions for other epistemic uncertainty models for which aleatoric uncertainty could be taken into account as well. First investigations regarding independent-

decoupled design decisions under pure aleatoric uncertainty were considered in [123]. Overall, this could help to improve the applicability of the methodology in systems engineering under polymorphic uncertainty.
Of course, this list is not complete and future research can follow up any question raised in this thesis. In particular, there are various open, narrow questions, which were partially addressed in the discussion of the limitations of this thesis, too. In the following, the optimization of crash-relevant components is considered for which the constraints are provided by component solution spaces.

In the modeling approach of this thesis, the optimization of crash-relevant components corresponds to coupled design decisions at the second level of a three-level crash system, see Section 5.1. First, a design model for the crash-relevant component at level 2 must be provided, e.g., a high-fidelity FE model, for which design variables, grouped as a subsystem design at level 2 and as subsystem component designs at level 1, and potential uncontrollable parameters need to be specified. Here, the design variables might be geometric or material properties, and the subsystem responses of the component design model must be force-deformation characteristics at level 2. Using component solution spaces which are based on an enhanced deformation space model and which are intended for parametrized force-deformation characteristics, performance thresholds on the force-deformation characteristics can be provided. If they are transferred to the design variable of the component design model, a complete subsystem solution space at level 2 defining all permissible design variables is yielded. In order to make an optimal decision for the permissible design alternatives, an appropriate cost function needs to be defined. Then, the optimal component design at level 2 can be sought.
What sounds to be simple here is rather complex in reality. For example, there are various methodical questions that concern the boundary conditions for the FE model to obtain reasonable results in combination with deformation space models. Furthermore, it must be discussed how the constraints provided by the component solution spaces should be interpreted for continuous force-deformation characteristics, which follows up the discussion from Section 5.2. In order to account for epistemic uncertainty, potential sources of uncertainty, including the modeling of uncertainty magnitudes for controllable design variables and uncontrollable parameters, must be investigated. For an efficient optimization for high-fidelity FE models, suitable mathematical surrogate models must be considered as well.
Overall, extensive research is required for the optimization of crash-relevant components using component solution spaces. A first approach that uses simplified boundary conditions and neglects uncertainty can be found in [15]. Exemplary, the influence of varying the values of a design variable of a component on its force-deformation characteristic is visualized in Figure 78. Here, the wall thickness of the left crash box of the HONDA ACCORD FE model from [98] is diminished by 20-percent. Note that this is not an optimization. However, it is sufficient to yield a permissible from a non-permissible parametrized force-deformation characteristic in this special case. In general, making decoupled decisions at level 1 instead of coupled decisions at level 2 for the design of crash-relevant components is also conceivable if more

design variables are taken into account. In the remaining section, the major results of this thesis are summarized.

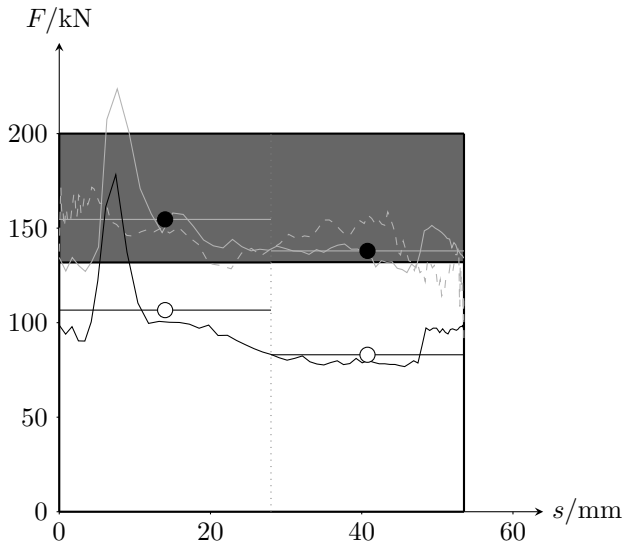


Figure 78 Measured (continuous force levels) and parametrized (constant force levels) force-deformation characteristic of the fourth component of the HONDA ACCORD FE model in a full vehicle simulation (dashed lines) and a drop-tower test for the component (solid line) with original (gray lines) and diminished (black lines) wall thicknesses. The parametrization is done with respect to the segments of the underlying enhanced DSM (dotted vertical lines). The corresponding box-shaped component solution space of Figure 51 is visualized (white region) inside the corresponding component design space (gray region). The white dots for the constant force levels indicate a permissible and the black dots a non-permissible component design. Note that the force-deformation characteristics were multiplied by two to account for the right crash box, too.

Main Results

This thesis provides a new methodology for decoupled design decisions in systems engineering that enables optimal flexibility for component design under epistemic uncertainty. The main results are summarized below. Note that most of them were already addressed from a different viewpoint in the discussion above. Here, the focus is put on the novel contributions of this thesis to literature.

In this thesis, a framework that distinguishes between coupled, dependent-decoupled, and independent-decoupled design decisions is introduced. Here, coupled decisions require a two-way flow of information between the designers making design decisions, dependent-decoupled require a one-way flow of information, and independent-decoupled require no flow of information. In order to enable designing according to this framework with mathematical design models by also providing flexibility for decoupled design decisions, fundamental prerequisites are discussed. Furthermore, the consideration of epistemic uncertainties, accounting for uncertainties in the early design phase and addressing controllable design variables and uncontrollable parameters, is incorporated into the framework. These uncertainties are modeled as intervals, treated in the framework of interval analysis, and fuzzy sets, treated in

the framework of possibility theory. Previous approaches from literature are presented and classified according to this framework. In order to obtain a complete framework for design decisions under epistemic uncertainty modeled as intervals and fuzzy sets, the methods for coupled design decisions are complemented directly and research gaps regarding decoupled design decisions are identified. These gaps concern the alignment of decoupled design decisions with the hierarchical structure of the system and the consideration of uncertainty.
The definitions of mathematical design models are extended such that they account for systems consisting of components, i.e., two-level systems. Additionally, it is discussed how this can be generalized to multi-level systems. In accordance with these definitions and the introduced framework, component solution spaces, as sets of permissible component designs that provide flexibility for designing at the component level, are proposed. They complement the complete system solution space for permissible system designs at the system level. It is distinguished between independent component solution spaces that provide flexibility for independent-decoupled design decisions and dependent component solution spaces that provide flexibility for independent-decoupled design decisions. Corresponding problem statements are proposed to optimize the flexibility of these decisions. In this thesis, the geometric shapes of component solution spaces are either predefined as boxes, yielding box-shaped component solution spaces, or optimized, yielding arbitrarily-shaped component solution spaces. Box-shaped component solution spaces have the advantage of a relatively simple computation and an easy visualization and arbitrarily-shaped ones provide more flexibility. In order to incorporate also epistemic uncertainty modeled as intervals, worst-case component solution spaces that account for all realizations of the uncertainties and best-case component solution spaces that account for at least one realization of the uncertainties are proposed. If they are modeled as fuzzy sets, necessity-α component solution spaces, for which the necessity that a component design is permissible is at least α, and possibility-α component solution spaces, for which the possibility that a component design is permissible is at least α, are yielded. These different types of component solution spaces provide multiple design regions for component designers and support their decisions regarding the uncertainty scenario they intend to design for. In this thesis, the uncertainties may address uncontrollable variables and uncontrollable parameters. For the computation of component solution spaces, the corresponding uncertainty magnitudes must be provided. Furthermore, the case of unknown uncertainty magnitudes in controllable variables is taken into account in the case of more severe uncertainty in these variables, stemming, for example, from undefined subsystems in the early design phase.
If the system performance functions incorporated in the mathematical design models are continuous, the complete system solution spaces for the epistemic uncertainty modeling of this thesis, besides the possibility-0 complete system solution space, are shown to be compact. This justifies the modeling of component solution spaces as compact sets. Moreover, it is demonstrated how the problem statements to compute component solution spaces can be simplified for performance functions which are linear in the controllable variables and monotonic in the uncontrollable parameters yielding convex optimization problems. For these, every local optimum is a global optimum and the problems can be solved by using local optimization solvers. As in particular the problem statements for arbitrarily-shaped component solution

spaces include a complex computation of the volumes of the component solution spaces and the projection of the complete system solution space, suitable algorithms from literature are reviewed and incorporated for this regard.
In this thesis, the methodology is applied to crash design problems. Here, deformation space models as mathematical design models which provide constraints for controllable force-deformation characteristics of the crash-relevant components are used. They are enhanced to knowledge-based deformation space models for which system-related parameters are obtained from realistic crash tests or simulations. Furthermore, energy- and acceleration-corrected deformation space models using additional load paths and energy-correction factors to calibrate energy and acceleration responses on results from finite-element-method simulations are introduced. In order to compare the different types of component solution spaces, two test-bed examples are proposed, a simple one, based on a simple deformation space model, and a realistic one based on an energy- and acceleration-corrected deformation space model. The computations can be done using the corresponding MATLAB app that was created within the scope of this thesis. It covers all methods for decoupled design decisions under epistemic uncertainty presented in this thesis. Furthermore, energy- and acceleration-corrected deformation space models can be created using this app. Overall, the application of the methodology to crash design illustrates its full potential for decoupled design decisions under epistemic uncertainty in systems engineering. This motivates transferring the approach to further problems in systems engineering for which decoupled design decisions under epistemic uncertainty shall be addressed.

A. APPENDIX

A.1. Important Classes of Sets and Functions

In this thesis, the underlying sets for the controllable variables, uncontrollable parameters, and responses together with the standard scalar products form Euclidean spaces. The subsequent considerations are made representatively for $\mathbb{R}^d$ for which the *standard scalar product* is

$$\langle x, y \rangle = \sum_{i=1}^{d} x_i y_i, \tag{A.1}$$

$x, y \in \mathbb{R}^d$. The scalar product $\langle \cdot, \cdot \rangle$ from Equation (A.1) induces a norm on $\mathbb{R}^d$, i.e., $\|x\| = \sqrt{\langle x, x \rangle}$ for $x \in \mathbb{R}^d$, which is called the *Euclidean norm*. In general, there are more norms on $\mathbb{R}^d$ which are all equivalent, see [53]. In $\mathbb{R}^d$, open and closed sets can be defined using open balls.

Definition 3. *Let $\| \cdot \|$ be a norm on $\mathbb{R}^d$.*

(a) An open ball with center $x_0 \in \mathbb{R}^d$ of radius δ, $\delta \in \mathbb{R}^+$ is defined as

$$B(x_0, \delta) = \{x \in \mathbb{R}^d \mid \|x - x_0\| < \delta\}. \tag{A.2}$$

(b) A set $U \subseteq \mathbb{R}^d$ is open in $\mathbb{R}^d$ if for every $x \in \mathbb{R}^d$ there exists a ball $B(x, \delta)$ such that $B(x, \delta) \subseteq U$.

(c) A set $U \subseteq \mathbb{R}^d$ is closed in $\mathbb{R}^d$ if $\mathbb{R}^d \backslash U$ is open in $\mathbb{R}^d$.

Moreover, it holds that a set $U \subseteq \mathbb{R}^d$ is closed in $\mathbb{R}^d$ if and only if every convergent sequence with values in U converges to a point in U. The union of arbitrarily many open sets and the intersection of finitely many open sets is open in $\mathbb{R}^d$. The union of finitely many closed sets and the intersection of arbitrarily many closed sets is closed in $\mathbb{R}^d$. Moreover, a set $U \subset \mathbb{R}^d$ is *bounded* if it is contained in some open ball and a set $U \subset \mathbb{R}^d$ is *compact* if it is bounded and closed in $\mathbb{R}^d$. [51]
An example for an open set is the open interval $(x^{\mathrm{l}}, x^{\mathrm{u}}) \subset \mathbb{R}$ and an example for a closed set, which is also compact, is the closed interval $[x^{\mathrm{l}}, x^{\mathrm{u}}] \subset \mathbb{R}$ where $x^{\mathrm{l}}, x^{\mathrm{u}} \in \mathbb{R}$ with $x^{\mathrm{l}} < x^{\mathrm{u}}$. This can be directly transferred to d*-dimensional intervals*, which are defined as

$$(x^{\mathrm{l}}, x^{\mathrm{u}}) = \{x \in \mathbb{R}^d \mid x_i^{\mathrm{l}} < x < x_i^{\mathrm{u}},\ i = 1, \ldots, d\}, \tag{A.3}$$

$$[x^{\mathrm{l}}, x^{\mathrm{u}}] = \{x \in \mathbb{R}^d \mid x_i^{\mathrm{l}} \leq x \leq x_i^{\mathrm{u}},\ i = 1, \ldots, d\}, \tag{A.4}$$

with $x^{\mathrm{l}} = (x_1^{\mathrm{l}}, \ldots, x_d^{\mathrm{l}}) \in \mathbb{R}^d$ and $x^{\mathrm{u}} = (x_1^{\mathrm{u}}, \ldots, x_d^{\mathrm{u}}) \in \mathbb{R}^d$. The d-dimensional intervals can be also represented by Cartesian products of one-dimensional intervals, i.e.,$(x^{\mathrm{l}}, x^{\mathrm{u}}) = (x_1^{\mathrm{l}}, x_1^{\mathrm{u}}) \times \cdots \times (x_d^{\mathrm{l}}, x_d^{\mathrm{u}})$ and $[x^{\mathrm{l}}, x^{\mathrm{u}}] = [x_1^{\mathrm{l}}, x_1^{\mathrm{u}}] \times \cdots \times [x_d^{\mathrm{l}}, x_d^{\mathrm{u}}]$.
Furthermore, all open and closed sets, including compact sets, belong to the Borel σ-algebra of $\mathbb{R}^d$, i.e., they are Borel sets for which the d-dimensional volume can be calculated using the Lebesgue measure, see [52]. Another class of sets that is important in this thesis are convex sets in $\mathbb{R}^d$.

Definition 4. *A set $U \subseteq \mathbb{R}^d$ is convex if either $U = \emptyset$ or if for all $\theta \in [0, 1]$ and all $x, x' \in U$, it holds*

$$\theta x + (1 - \theta)x' \in U. \tag{A.5}$$

This means that if any segment that connects two points of a set is entirely contained in the set, the set is convex. The intersection of convex sets is convex. For example, all intervals and all polytopes of the mathematical structure $\{x \in \mathbb{R}^d \mid Ax \leq b\}$, $A \in \mathbb{R}^{m \times d}$, $b \in \mathbb{R}^m$, are convex. [12]

Subsequently, important classes of functions are considered for $f : \mathbb{R}^d \to \mathbb{R}^m$. This generalizes to system or component performance functions that map controllable variables and uncontrollable parameters to responses.
Continuous functions can be defined as in [103], for which Theorem 15 holds.

Theorem 15. *Let $f : \mathbb{R}^d \to \mathbb{R}^m$. The following statements are equivalent:*

(a) At every point of $\mathbb{R}^d$, f is continuous

(b) For every open subset $V \subseteq \mathbb{R}^m$, the inverse image $f^{-1}(V)$ is open in $\mathbb{R}^d$.

(c) For every closed subset $V \subseteq \mathbb{R}^m$, the inverse image $f^{-1}(V)$ is closed in $\mathbb{R}^d$.

(d) For every sequence $(x_n)_{n \in \mathbb{N}}$ that converges to $x \in \mathbb{R}^d$, the sequence $(f(x_n))_{n \in \mathbb{N}}$ converges to $f(x) \in \mathbb{R}^m$.

Proof. Compare [103]. □

Furthermore, $f : \mathbb{R}^d \to \mathbb{R}^m$ is continuous if and only if $f_j : \mathbb{R}^d \to \mathbb{R}^m$, $j = 1, \ldots, m$, are continuous. Next, convex and concave functions of the form $f : \mathbb{R}^d \to \mathbb{R}$ are considered.

Definition 5. *A function $f : \mathbb{R}^d \to \mathbb{R}$ is convex if for all $\theta \in [0,1]$ and all $x, x' \in U$, it holds*

$$f(\theta x + (1-\theta)x') \leq \theta f(x) + (1-\theta) f(x'). \tag{A.6}$$

If $-f$ is convex, f is concave.

Definition 5 can be generalized to functions of the form $f : \mathbb{R}^d \to \mathbb{R}^m$. They are said to be convex if and only if $f_j : \mathbb{R}^d \to \mathbb{R}^m$, $j = 1, \ldots, m$, are convex. A convex function $f : \mathbb{R}^d \to \mathbb{R}^m$ is also continuous and the set $f^{-1}((-\infty, f_\mathrm{c}]) = \{x \in \mathbb{R}^d \mid f(x) \leq f_\mathrm{c}\}$, $f_\mathrm{c} \in \mathbb{R}^m$, is convex, see [12].
Moreover, a function $f : \mathbb{R}^d \to \mathbb{R}$ with $f(x) > 0$ for all $x \in \mathbb{R}^d$ is said to be *logarithmically convex* if $\log f$ is convex and *logarithmically concave* if $\log f$ is concave. Here, $f(x) = 0$ can be allowed by defining $\log(0) = -\infty$. Logarithmic concavity of a function f can also be expressed without the logarithm, i.e. f is logarithmically concave if and only if for all $\theta \in [0,1]$ and all $x, x' \in U$, it holds

$$f(\theta x + (1-\theta)x') \geq f(x)^\theta f(x')^{1-\theta}. \tag{A.7}$$

Furthermore, the product of logarithmically concave functions is logarithmically concave and in some special cases, the logarithmic concavity is also preserved by integration. [12]

A component design $x^k \in \mathbb{R}^{d^k}$ can be considered as the projection of the system design $x \in \mathbb{R}^d$ onto the coordinate space of the k^{th} component. Here, the projection is denoted by proj^k with

$$\mathrm{proj}^k : \mathbb{R}^d \to \mathbb{R}^{d^k},\ x \mapsto x^k, \tag{A.8}$$

$k \in \{1, \ldots, n\}$. The projections proj^k, $k = 1, \ldots, n$, are continuous. For the projection $\mathrm{proj}^k(U)$ of a set $U \subseteq \mathbb{R}^d$, it holds

$$\mathrm{proj}^k(U) = \{x^k \in \mathbb{R}^{d^k} \mid \exists x^{\mathrm{r},k} : x \in U\}, \tag{A.9}$$

where $x^{\mathrm{r},k}$ is a vector that contains the remaining component designs $x^{k'}$, $k' = 1, \ldots, n$, $k' \neq k$, and x is formed by x^k and $x^{\mathrm{r},k}$ for $k = 1, \ldots, n$. Furthermore, the projection of a convex polytope is convex, see [72].

A.2. Fuzzy Sets and Possibility Theory

Fuzzy sets, as introduced in [136], are one method to model epistemic uncertainty, see Section 2.1. In fuzzy set theory, the elements of a reference set have degrees of membership of belonging to a fuzzy set compared to classical set theory, in which the elements of a reference set either belong or do not belong to a set. This gradual membership is expressed by membership functions. For the reference set $\mathbb{R}$, a *membership function* for the fuzzy set X

is defined by

$$\mu^X : \mathbb{R} \to [0,1],\ x \mapsto \mu^X(x). \tag{A.10}$$

Moreover, the fuzzy set X itself is defined as the pair $(\mathbb{R}, \mu^X)$ where $x \in \mathbb{R}$ is called *fully included* in X if $\mu^X(x) = 1$, *partially included* in X if $0 < \mu^X(x) < 1$, and *not included* in X if $\mu^X(x) = 0$. Examples of membership functions for the reference set $\mathbb{R}$ are given in Equations (2.15) and (2.16). The membership function from Equation (2.16) only takes the values 0 and 1 and therefore corresponds to a characteristic function of a classical set. Note that there are also more general reference sets.
The set of all elements whose membership grade is at least α is $\{x \in \mathbb{R} \mid \mu^X(x) \geq \alpha\}$, which is a crisp set, i.e., a classical set, and also called α*-cut* for $\alpha \in [0,1]$. In addition, the set $\{x \in \mathbb{R} \mid \mu^X(x) > 0\}$ is called the *support* and $\{x \in \mathbb{R} \mid \mu^X(x) = 1\}$ is called the *core* of X. They are also crisp sets.

In order to perform arithmetic operations on fuzzy sets $X_1, \ldots, X_d$, the extension principle, introduced in [137], can be used.

Definition 6. *Let $X_1, \ldots, X_d$ be fuzzy sets, for which $\mathbb{R}$ is the reference set for each of these fuzzy sets and μ^{X_i}, $i = 1, \ldots, d$, are their membership functions. Furthermore, let f be of the form $f : \mathbb{R}^d \to \mathbb{R}^m$. Then, $Z = f(X_1, \ldots, X_d)$ is a fuzzy set with reference set $\mathbb{R}^m$ and its membership function can be defined as*

$$\mu^Z : \mathbb{R}^m \to [0,1],$$
$$z \mapsto \mu^Z(z) = \begin{cases} \sup\limits_{x \in f^{-1}(z)} \min\{\mu^{X_1}(x_1), \ldots, \mu^{X_d}(x_d)\} & \text{if } f^{-1}(z), \neq \emptyset, \\ 0 & \text{else}, \end{cases} \tag{A.11}$$

see [137].

Note that other extension principles are also conceivable, cf. [33]. For multi-dimensional fuzzy sets, the supports, cores, and α-cuts, $\alpha \in [0,1]$, can be defined similarly to above. Further details on fuzzy arithmetic using Definition 6 are, for example, given in [61].
From Definition 6, a membership function for the fuzzy number $X = (X_1, \ldots, X_d)$ with reference set $\mathbb{R}^d$ is defined implicitly. It reads

$$\mu^X : \mathbb{R}^d \to [0,1],\ x \mapsto \mu^X(x) = \min\{\mu^{X_1}(x_1), \ldots, \mu^{X_d}(x_d)\}. \tag{A.12}$$

If a fuzzy set X is *normalized*, i.e., there exists at least one x within the reference set with $\mu^X(x) = 1$, its membership function can be considered as a possibility distribution, see [31, 138].

Possibility theory can be viewed as a framework to treat epistemic uncertainty by imprecise probabilities, cf. [131]. Its mathematical basics were mainly introduced in [31, 33]. In possibility theory, a *possibility distribution* π is used to represent a state of knowledge. In this thesis, $\pi : \mathbb{R}^d \to [0,1],\ x \mapsto \pi(x)$ is considered, where π can be identified mathematically with μ^X. Similar to the interpretation of fuzzy sets, a state $x \in \mathbb{R}^d$ is totally possible if $\pi(x) = 1$ and is impossible if $\pi(x) = 0$, see [31].
Degrees of possibilities and necessities that an element of the set $U \subset \mathbb{R}^d$ occurs can be computed by the *possibility measure* Pos and the *necessity measure* Nec via

$$\mathrm{Pos}(U) = \sup_{x \in U} \pi(x), \tag{A.13}$$

$$\mathrm{Nec}(U) = \inf_{x \in \mathbb{R}^d \setminus U} (1 - \pi(x)). \tag{A.14}$$

The two measures are dual, which is expressed by

$$\mathrm{Nec}(U) = 1 - \mathrm{Pos}(\mathbb{R}^d \setminus U) \tag{A.15}$$

for $U \subseteq \mathbb{R}^d$. More general, possibility and necessity measures are mappings from the power set of $\mathbb{R}^d$ to $[0,1]$ with

$$\mathrm{Pos}(\emptyset) = 0, \tag{A.16}$$

$$\mathrm{Pos}(\mathbb{R}^d) = 1, \tag{A.17}$$

$$\mathrm{Pos}(U \cup V) = \max(\mathrm{Pos}(U), \mathrm{Pos}(V)), \tag{A.18}$$

and

$$\mathrm{Nec}(\emptyset) = 0, \tag{A.19}$$

$$\mathrm{Nec}(\mathbb{R}^d) = 1, \tag{A.20}$$

$$\mathrm{Nec}(U \cap V) = \min(\mathrm{Nec}(U), \mathrm{Nec}(V)), \tag{A.21}$$

for all $U, V \subseteq \mathbb{R}^d$, cf. [31]. Here, the duality from Equation (A.15) can be used to derive Equations (A.19)-(A.21) from Equations (A.16)-(A.18) and vice versa. From these equations follows that $\mathrm{Pos}(U) < 1$ implies $\mathrm{Nec}(U) = 0$ and $\mathrm{Nec}(U) > 0$ implies $\mathrm{Pos}(U) = 1$ for $U \subseteq \mathbb{R}^d$. Furthermore, it is argued in [32] that possibility measures can also be considered as upper probability measures and necessity measures as lower probability measures, i.e., they form bounds for imprecise probabilities. However, note that the propagation of these bounds generally does not yield bounds for propagated probabilistic uncertainty if the extension principle defined in Definition 6 is used. Approaches to circumvent this problem are proposed in [67].
For a given possibility distribution π, the possibility and necessity measures of, e.g., the set $(-\infty, x^{\mathrm{u}}] \subset \mathbb{R}^d$ can be interpreted as the degrees of possibility and necessity that the elements

of the corresponding fuzzy set X are smaller than or equal to upper thresholds $x^{\mathrm{u}} \in \mathbb{R}^d$. Without loss of generality, let the elements of the zero vector be these upper thresholds. Using Equations (A.13) and (A.14), it holds

$$\begin{aligned} \mathrm{Pos}(\{X \mid X \leq 0\}) &= \mathrm{Pos}((-\infty, 0]) && \text{(A.22)} \\ &= \sup_{x \leq 0} \mu^X(x) && \text{(A.23)} \\ \mathrm{Nec}(\{X \mid X \leq 0\}) &= \mathrm{Nec}((-\infty, 0]) && \text{(A.24)} \\ &= \inf_{x > 0} (1 - \mu^X(x)), && \text{(A.25)} \\ & && \text{(A.26)} \end{aligned}$$

compare [69].

A.3. Requirements and FE Measurements for Crash Design

In the following, the requirements for deformation space models, which are stated in Section 5.1 and proposed in [43], are derived. Here, the notations introduced in Section 5.1 are used:

- *Maximum acceleration*: If the vehicle deforms in deformation space, the active mass $m^*(s)$ is accelerated by the sum of deformation forces of all components $F(s)$ at any position $s \in [s_0, s_{\mathrm{end}})$. Thus, the resulting acceleration $a(s)$ is determined by the ratio of $F(s)$ to $m^*(s)$, i.e.,

  $$a(s) = \frac{F(s)}{m^*(s)} \tag{A.27}$$

  for $s \in [s_0, s_{\mathrm{end}})$. As the acceleration must not exceed the critical acceleration a_{c}, it must hold $a(s) \leq a_{\mathrm{c}}$ or

  $$\frac{F(s)}{m^*(s)} \leq a_{\mathrm{c}} \tag{A.28}$$

  for all $s \in [s_0, s_{\mathrm{end}})$, see inequality (5.9).

  Progressive order of deformation: In order to ensure that there is a deformation front in deformation space that starts at the front of the vehicle and moves in longitudinal impact direction, the deformation of the component must be ordered progressively, starting at the front. Hence, the force which is responsible for the deformation of the k^{th} component $F^k(s)$ must be smaller than the force that is necessary to start the deformation of the $(k+1)^{\mathrm{th}}$ component $F^{k+1}(s_0^{k+1})$, for which additionally the inertia force $F_{m^k}(s)$ for the mass m^k must be considered, at any position $s \in [s_0^k, s_{\mathrm{end}}^k)$. If $F^k(s)$ is greater than the sum of $F^{k+1}(s_0^{k+1})$ and $F_{m^k}(s)$, the $(k+1)^{\mathrm{th}}$ component starts to deform before the k^{th} component has completely deformed, i.e., the requirement for a progressively ordered

deformation can be formulated as

$$F^k(s) \leq F^{k+1}(s_0^{k+1}) + m^k a(s) \tag{A.29}$$

for $s \in [s_0^k, s_{\text{end}}^k)$. Note that this must only hold if the k^{th} and $(k+1)^{\text{th}}$ component share the same load path, $k = 1, \ldots, n-1$. Using equation (A.27), inequality (A.29) can also be formulated as

$$F^k(s) - m^k \frac{F(s)}{m^*(s)} \leq F^{k+1}(s_0^{k+1}) \tag{A.30}$$

for all $s \in [s_0^k, s_{\text{end}}^k)$ if the k^{th} and $(k+1)^{\text{th}}$ component share the same load path, $k = 1, \ldots, n-1$, see inequality (5.10).

- *Minimum energy absorption*: If the progressive order of deformation requirement is fulfilled, the first section of the vehicle completely deforms in a vehicle crash before the other sections start to deform. At the start of the deformation of section 1 at s_0, the vehicle has the velocity $v_0 \in \mathbb{R}_0^+$ and the velocity $v_1 \in \mathbb{R}_0^+$ at the end of the deformation of section 1 at s_1. As the active mass does not change during the deformation of section 1, i.e., $m^*(s) = m^*(s_0)$ for $s \in [s_0, s_1)$, see Equation (5.7), it holds

$$\int_{s_0}^{s_1} F(s)\,\mathrm{d}s = \frac{1}{2} m^*(s_0)(v_0^2 - v_1^2) \tag{A.31}$$

for the absorbed kinetic energy, i.e.,

$$\int_{s_0}^{s_1} \frac{F(s)}{m^*(s)}\,\mathrm{d}s = \frac{1}{2}(v_0^2 - v_1^2). \tag{A.32}$$

Then, it is assumed that the masses located at s_1 are brought to an abrupt halt and that the remaining kinetic energy of these masses is not absorbed by the components of the vehicle structure. An elastic rebound of the vehicle is neglected. The sum of the masses located at s_1 is represented by the difference in the active mass $(m^*(s_0) - m^*(s_1))$. Thus, the corresponding kinetic energy which is not absorbed by the components is $\frac{1}{2}(m^*(s_0) - m^*(s_1))v_1^2$.
Afterward, section 2 starts deforming at s_1, where the vehicle has still the velocity v_1, and ends deforming at s_2 with velocity $v_2 \in \mathbb{R}_0^+$. Similar to above, it holds

$$\int_{s_1}^{s_2} \frac{F(s)}{m^*(s)}\,\mathrm{d}s = \frac{1}{2}(v_1^2 - v_2^2). \tag{A.33}$$

Then, the masses located at s_2 are brought to an abrupt halt, too. This is continued and for an arbitrary position $s' \in [s_0, s_{\text{end}})$ in deformation space with velocity $v(s') \in \mathbb{R}_0^+$, it is

$$\int_{s_0}^{s'} \frac{F(s)}{m^*(s)}\,\mathrm{d}s = \frac{1}{2}(v_0^2 - v(s')^2). \tag{A.34}$$

If all components of the vehicle deform completely and $v(s_{\text{end}}) > 0$ holds, the initial kinetic energy of the vehicle is not fully absorbed. Hence, to absorb this minimum required energy and to bring the vehicle to a halt at $s \in [s_0, s_{\text{end}})$, it must hold that

$$\int_{s_0}^{s_{\text{end}}} \frac{F(s)}{m^*(s)} \, \mathrm{d}s \geq \frac{1}{2} v_0^2, \tag{A.35}$$

i.e.,

$$-\int_{s_0}^{s_{\text{end}}} \frac{F(s)}{m^*(s)} \, \mathrm{d}s \leq -\frac{1}{2} v_0^2, \tag{A.36}$$

see inequality (5.8).

Furthermore, it is proposed how measurements in FEM simulations for frontal crash at full overlap against a rigid wall can be done in order to obtain properties like force-deformation characteristics, energies, and masses which are required for enhanced DSM. This is applied in Section 5.2 for the HONDA ACCORD FE using the software LS-DYNA.

Part masses of an FE model can be measured directly in an FEM simulation by considering the corresponding parts. The longitudinal lengths of the components can be computed, for example, by the average distances between four different nodes at the front and four different nodes at the back of each component. From measuring these lengths over time, a deformation-time graph can be deduced. The deformation lengths of the components can then be set as the deformation values at the end time of the simulation, or as the difference of the deformation values at the end and beginning of plastic deformation.
Similarly, forces can be measured over time and combined with the deformation measurements over time to force-deformation characteristics afterward. In order to account only for plastic deformation, both the deformation-time graph and force-time graph are considered in the time interval of plastic deformation here. In doing so, the deformation-time graph is measured as described above. In theory, the force-time graph of a component can be measured at an arbitrary cross-section of the component, see [43]. Moreover, the force-time graph should also not be dependent on the position of the cross-section on a load path if the active mass does not change.

However, as each considered component is usually in contact with other parts of the structure during deformation, i.e., there is load transmission along the component, this no longer holds. Therefore, a procedure to obtain a force-time graph that considers especially the forces measured at deforming segments of the components is proposed and used in this thesis: First, each component is divided into several segments of approximately the same length. At the beginning and end of each segment, four nodes are placed to measure the deformation of the

segment over time. In between, force-over-time in longitudinal impact direction is measured by also taking self-contact of the load paths into account. For each segment, the force is integrated over the deformation of the segment and transformed back into time-space. Thus, an energy-time graph of the segment that accounts for the energy absorbed in the longitudinal direction is obtained. Then, the energies of all segments are summed up to get the total energy of the component absorbed in the longitudinal direction over time. By taking the derivative with respect to the deformation of the component, a force-deformation characteristic with the desired properties is yielded. A natural upper bound of this force-deformation characteristic is the derivative of the total internal energy of the component, which accounts for the absorbed energy in all directions, with respect to the deformation of the component. Due to uncertainties in deformation and force measurements, this is not always an upper-bounding force. Hence, the point-wise minimum of these two force-deformation characteristics can be chosen to get a force-deformation characteristic of the component that is conservative regarding energy absorption and acceleration.

If the crash relevant components deform progressively, a force-deformation characteristic that represents all considered components can be computed from Equation (5.6). If this is not the case, a procedure similar to the computation of force-deformation characteristics of the single components can be used here: From the force-deformation characteristics of the components, energy-time graphs for the components are computed and summed up. Its derivative with respect to the deformation of the load paths provides a force-deformation characteristic for all components represented in the DSM. Here, it is used that the deformation rate is approximately the same across all load paths. The corresponding force at deformation position $s \in [s_0, s_{\mathrm{end}})$ is denoted by $F'(s)$.
For calibrating the force $F^{\mathrm{add}}(s)$ for the additional load path of an enhanced DSM, the equation

$$F^{\mathrm{add}}(s) = m^*(s)a_{\mathrm{rv}}(s) - F'(s), \tag{A.37}$$

where $a_{\mathrm{rv}}(s)$ denotes the acceleration of the rear vehicle at deformation position $s \in [s_0, s_{\mathrm{end}})$, can be used. Thus, the corresponding acceleration-deformation and active-mass-deformation graphs must be computed first. The acceleration-deformation graph of the rear vehicle is obtained in an FEM simulation by measuring the acceleration of an associated node over time and performing a geometric transformation of the domain from time into deformation of the load paths. Furthermore, the active mass $m^*(s)$ of the vehicle is estimated here by removing the parts with zero velocity at deformation position $s \in [s_0, s_{\mathrm{end}})$ and measuring the remaining mass in an FEM simulation. This is done for $s \in \{s_0, \ldots, s_{\mathrm{end}}\}$. In between, the active mass is set as constant for the DSM approach.
Alternatively, it is also conceivable to estimate the active mass of the vehicle, for example, by $m^*(s) = \frac{2E_{\mathrm{kin}}(s)}{v_{\mathrm{rv}}(s)^2}$ where $E_{\mathrm{kin}}(s)$ denotes its remaining kinetic energy and $v_{\mathrm{rv}}(s)$ the velocity of the rear vehicle at deformation position $s \in [s_0, s_{\mathrm{end}})$.
The energies of the components E^k_{tot}, $k = 1, \ldots, n$, in an FEM simulation can be measured as the difference in internal energy in the time interval of plastic deformation, for example.

REFERENCES

[1] J. A. C. Ambrosio, ed. *Crashworthiness: energy management and occupant protection*. Vienna, Austria: Springer, 2001. ISBN: 9783211833346.

[2] S. Azarm and W.-C. Li. "A two-level decomposition method for design optimization." In: *Engineering Optimization* 13.3 (1988), pp. 211–224. ISSN: 0305-215X. DOI: 10.1080/03052158808940956.

[3] C. Bach, D. Ceglia, L. Song, and F. Duddeck. "Randomized low–rank approximation methods for projection–based model order reduction of large nonlinear dynamical problems." In: *International Journal for Numerical Methods in Engineering* 118.4 (2019), pp. 209–241. ISSN: 00295981. DOI: 10.1002/nme.6009.

[4] A. Baharev, F. Domes, and A. Neumaier. "A robust approach for finding all well-separated solutions of sparse systems of nonlinear equations." In: *Numerical Algorithms* 76.1 (2017), pp. 163–189. ISSN: 1017-1398. DOI: 10.1007/s11075-016-0249-x.

[5] Z. Bai. "Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems." In: *Applied Numerical Mathematics* 43.1-2 (2002), pp. 9–44. ISSN: 01689274. DOI: 10.1016/S0168-9274(02)00116-2.

[6] A. Bemporad, C. Filippi, and F. D. Torrisi. "Inner and outer approximations of polytopes using boxes." In: *Computational Geometry* 27.2 (2004), pp. 151–178. ISSN: 09257721. DOI: 10.1016/S0925-7721(03)00048-8.

[7] A. Ben-Tal, L. El Ghaoui, and A. Nemirovski. *Robust optimization*. Princeton, USA: Princeton University Press, 2009. ISBN: 9780691143682.

[8] A. Ben-Tal and A. Nemirovski. "Robust optimization - methodology and applications." In: *Mathematical Programming* 92.3 (2002), pp. 453–480. ISSN: 0025-5610. DOI: 10.1007/s101070100286.

[9] D. P. Bertsekas. *Constrained optimization and Lagrange multiplier methods*. Belmont, USA: Athena Scientific, 1996. ISBN: 9781886529045.

[10] D. Bertsimas, D. B. Brown, and C. Caramanis. "Theory and applications of robust optimization." In: *SIAM Review* 53.3 (2011), pp. 464–501. DOI: 10.1137/080734510.

[11] H.-G. Beyer and B. Sendhoff. "Robust optimization – a comprehensive survey." In: *Computer Methods in Applied Mechanics and Engineering* 196.33-34 (2007), pp. 3190–3218. ISSN: 00457825. DOI: 10.1016/j.cma.2007.03.003.

[12] S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge, UK: Cambridge University Press, 2004. ISBN: 9780521833783.

[13] J. Branke. *Multiobjective optimization: interactive and evolutionary approaches*. Berlin, Germany: Springer, 2008. ISBN: 9783540889076.

[14] B. Büeler and A. Enge. "Vinci version 1.0.5." Technical report. Palaiseau, France: École Polytechnique, 2003.

[15] L. Burmberger. “Efficient global optimization of structural components in the context of solution spaces for crashworthiness.” M.Sc. thesis. Munich, Germany: Technische Universität München, 2020.

[16] M. Carvalho, J. Ambrósio, and P. Eberhard. “Identification of validated multibody vehicle models for crash analysis using a hybrid optimization procedure.” In: *Structural and Multidisciplinary Optimization* 44.1 (2011), pp. 85–97. ISSN: 1615-147X. DOI: 10.1007/s00158-010-0590-y.

[17] C. Charalambous and A. R. Conn. “An efficient method to solve the minimax problem directly.” In: *SIAM Journal on Numerical Analysis* 15.1 (1978), pp. 162–187. ISSN: 0036-1429. DOI: 10.1137/0715011.

[18] K. L. Clarkson. “More output-sensitive geometric algorithms.” In: *Proceedings of the 35th Annual Symposium on Foundations of Computer Science, Santa Fe, USA, November 1994* (1994). DOI: 10.1109/SFCS.1994.365723.

[19] D. Cornette, J. L. Thirion, E. Markiewicz, P. Drazetic, and Y. Ravalard. “Localization of collapse mechanisms for simplified vehicle crash simulation.” In: *Integrated design and manufacturing in mechanical engineering ’98*. Ed. by J.-L. Batoz, P. Chedmail, G. Cognet, and C. Fortin. Dordrecht, Netherlands: Springer, 1999. ISBN: 9789048153428.

[20] I. Cuevas Salazar, F. Duddeck, and L. Song. “Small overlap assessment for early design phases based on vehicle kinematics.” In: *International Journal of Crashworthiness* 58 (2019), pp. 1–16. ISSN: 1358-8265. DOI: 10.1080/13588265.2018.1514689.

[21] M. Daub. “Konvexe Optimierung am Beispiel volumenmaximaler einbeschriebener Rechtecksmengen.” M.Sc. thesis. Konstanz, Germany: Universität Konstanz, 2017. URL: http://nbn-resolving.de/urn:nbn:de:bsz:352-0-420923.

[22] M. Daub and F. Duddeck. “Complex systems design under non-reducible lack-of-knowledge uncertainties.” In: *Proceedings of the ICVRAM-ISUMA UNCERTAINTIES Conferences 2018, Florianópolis, Brazil, April 2018* (2018).

[23] M. Daub and F. Duddeck. “Robust target design for complex systems using multi-objective optimization.” In: *Proceedings of the 13th World Congress on Computational Mechanics (WCCM2018), New York City, USA, July 2018* (2018).

[24] M. Daub and F. Duddeck. “Worst-case solution spaces for systems design under uncertainties.” In: *Proceedings of the 8th International Workshop on Reliable Engineering Computing (REC2018), Liverpool, UK, July 2018* (2018).

[25] M. Daub and F. Duddeck. “Maximizing flexibility for complex systems design to compensate lack-of-knowledge uncertainty.” In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems: Part B. Mechanical Engineering* 5.4 (2019), p. 041008. DOI: 10.1115/1.4044045.

[26] M. Daub and F. Duddeck. “A decoupled design approach for complex systems under lack-of-knowledge uncertainty.” In: *International Journal of Approximate Reasoning* 119 (2020), pp. 408–420. DOI: 10.1016/j.ijar.2020.01.006.

[27] M. Daub, F. Duddeck, and M. Zimmermann. “Optimizing component solution spaces for systems design.” In: *Structural and Multidisciplinary Optimization* (2020). ISSN: 1615-1488. DOI: 10.1007/s00158-019-02456-8.

[28] A. P. Dempster. “A generalization of Bayesian inference.” Technical report. Cambridge, USA: Harvard University, 1967.

[29] L. Du, K. K. Choi, B. D. Youn, and D. Gorsich. “Possibility-based design optimization method for design problems with both statistical and fuzzy input data.” In: *Journal of Mechanical Design* 128.4 (2006), pp. 928–935. DOI: 10.1115/1.2204972.

[30] D. Dubois and H. Prade. “Ranking fuzzy numbers in the setting of possibility theory.” In: *Information Sciences* 30.3 (1983), pp. 183–224. ISSN: 00200255. DOI: 10.1016/0020-0255(83)90025-7.

[31] D. Dubois and H. Prade. *Possibility theory*. Boston, USA: Springer, 1988. ISBN: 9781468452891.

[32] D. Dubois and H. Prade. “When upper probabilities are possibility measures.” In: *Fuzzy Sets and Systems* 49.1 (1992), pp. 65–74. ISSN: 01650114. DOI: 10.1016/0165-0114(92)90110-P.

[33] D. J. Dubois and H. Prade. *Fuzzy sets and systems: theory and applications*. New York, USA: Academic Press, 1980. ISBN: 9780122227509.

[34] F. Duddeck and E. Wehrle. “Recent advances on surrogate modeling for robustness assessment of structures with respect to crashworthiness requirements.” In: *Proceedings of the 10th European LS-DYNA Conference, Würzburg, Germany, June 2015* (2015).

[35] M. Eigner, D. Roubanov, and R. Zafirov. *Modellbasierte virtuelle Produktentwicklung*. Berlin: Springer Vieweg, 2014. ISBN: 9783662438152.

[36] S. D. Eppinger, D. E. Whitney, R. P. Smith, and D. A. Gebala. “A model-based method for organizing tasks in product development.” In: *Research in Engineering Design* 6.1 (1994), pp. 1–13. ISSN: 1435-6066. DOI: 10.1007/BF01588087.

[37] S. Erschen. “Optimal decomposition of high-dimensional solution spaces for chassis design.” Ph.D. thesis. Munich, Germany: Technische Universität München, 2018.

[38] S. Erschen, F. Duddeck, M. Gerdts, and M. Zimmermann. “On the optimal decomposition of high-dimensional solution spaces of complex systems.” In: *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems: Part B. Mechanical Engineering* 4.2 (2018), pp. 1–15. DOI: 10.1115/1.4037485.

[39] S. Erschen, F. Duddeck, and M. Zimmermann. “Robust design using classical optimization.” In: *Proceedings in Applied Mathematics and Mechanics (PAMM)* 15.1 (2015), pp. 565–566. DOI: 10.1002/pamm.201510272.

[40] M. Evans and T. Swartz. *Approximating integrals via Monte Carlo and deterministic methods*. Oxford, UK: Oxford University Press, 2000. ISBN: 9780198502784.

[41] C. J. Falconi D., A. F. Walser, H. Singh, and A. Schumacher. “Automatic generation, validation and correlation of the submodels for the use in the optimization of crashworthy structures.” In: *Advances in structural and multidisciplinary optimization*. Ed. by A. Schumacher, T. Vietor, S. Fiebig, K.-U. Bletzinger, and K. Maute. Cham, Switzerland: Springer International Publishing, 2018. ISBN: 9783319679877.

[42] H. Fargier, J. Lang, R. Martin-Clouaire, and T. Schiex. “A constraint satisfaction approach to decision under uncertainty.” In: *Proceedings of the 11th Conference on Uncertainty in Artificial Intelligence (UAI1995), Montreal, Canada, August 1995* (1995).

[43] J. Fender. “Solution spaces for vehicle crash design.” Ph.D. thesis. Munich, Germany: Technische Universität München, 2013.

[44] J. Fender, F. Duddeck, and M. Zimmermann. “Direct computation of solution spaces.” In: *Structural and Multidisciplinary Optimization* 55.5 (2016), pp. 1787–1796. ISSN: 1615-147X. DOI: 10.1007/s00158-016-1615-y.

[45] M. Fiedler, J. Nedoma, J. Ramík, J. Rohn, and K. Zimmermann. *Linear optimization problems with inexact data*. Boston, USA: Springer, 2006. ISBN: 9780387326979.

[46] A. Forrester, A. Sóbester, and A. Keane. *Engineering design via surrogate modelling: a practical guide*. Hoboken, USA: John Wiley & Sons, 2008. ISBN: 9780470060681.

[47] W. Forst and D. Hoffmann. *Optimization-theory and practice*. New York, USA: Springer, 2010. ISBN: 9780387789767.

[48] J. Frishammar, H. Floren, and J. Wincent. “Beyond managing uncertainty: insights from studying equivocality in the fuzzy front end of product and process innovation projects.” In: *IEEE Transactions on Engineering Management* 58.3 (2011), pp. 551–563. ISSN: 0018-9391. DOI: 10.1109/TEM.2010.2095017.

[49] V. Gabrel, C. Murat, and A. Thiele. “Recent advances in robust optimization: an overview.” In: *European Journal of Operational Research* 235.3 (2014), pp. 471–483. ISSN: 03772217. DOI: 10.1016/j.ejor.2013.09.036.

[50] A. Galántai. “Projection methods for linear and nonlinear equations.” Ph.D. thesis. Miskolc, Hungary: University of Miskolc, 2003.

[51] M. Giaquinta and G. Modica. *Mathematical analysis: linear and metric structures and continuity*. Basel, Switzerland: Birkhäuser, 2007. ISBN: 9780817645144.

[52] M. Giaquinta and G. Modica. *Mathematical analysis: foundations and advanced techniques for functions of several variables*. Basel, Switzerland: Birkhäuser, 2012. ISBN: 9780817683108.

[53] M. S. Gockenbach. *Finite-dimensional linear algebra*. Discrete Mathematics and Its Applications. Boca Raton, USA: CRC Press, 2010. ISBN: 9781439815632.

[54] M. Götz. “Numerische Entwurfsmethoden unter Berücksichtigung polymorpher Unschärfe: Aspekte zeitlicher und räumlicher Abhängigkeiten.” Ph.D. thesis. Dresden, Germany: Technische Universität Dresden, 2017.

[55] M. Götz, W. Graf, and M. Kaliske. “Structural design with polymorphic uncertainty models.” In: *International Journal of Reliability and Safety* 9.2-3 (2015), pp. 112–131. ISSN: 1479-389X. DOI: 10.1504/IJRS.2015.072715.

[56] W. Graf, M. Götz, and M. Kaliske. “Analysis of dynamical processes under consideration of polymorphic uncertainty.” In: *Structural Safety* 52 (2015), pp. 194–201. ISSN: 0167-4730. DOI: 10.1016/j.strusafe.2014.09.003.

[57] W. Graf, M. Götz, and M. Kaliske. “Computing permissible design spaces under consideration of functional responses.” In: *Advances in Engineering Software* 117 (2018), pp. 95–106. ISSN: 09659978. DOI: 10.1016/j.advengsoft.2017.05.015.

[58] L. Graff. “A stochastic algorithm for the identification of solution spaces in high-dimensional design spaces.” Ph.D. thesis. Basel, Switzerland: Universität Basel, 2013.

[59] L. Graff, H. Harbrecht, and M. Zimmermann. “On the computation of solution spaces in high dimensions.” In: *Structural and Multidisciplinary Optimization* 54.4 (2016), pp. 811–829. ISSN: 1615-147X. DOI: 10.1007/s00158-016-1454-x.

[60] R. Haberfellner, O. de Weck, E. Fricke, and S. Vössner. *Systems engineering: fundamentals and applications*. Cham, Switzerland: Springer International Publishing, 2019. ISBN: 9783030134310.

[61] M. Hanss. *Applied fuzzy arithmetic*. Berlin, Germany: Springer-Verlag, 2005. ISBN: 9783540242017.

[62] S. Har-Peled. *Geometric approximation algorithms*. Providence, USA: American Mathematical Society, 2011. ISBN: 9780821849118.

[63] H. Harbrecht, D. Tröndle, and M. Zimmermann. “A sampling-based optimization algorithm for solution spaces with pair-wise-coupled design variables.” In: *Structural and Multidisciplinary Optimization* (2019). ISSN: 1615-1488. DOI: 10.1007/s00158-019-02221-x.

[64] S. M. Harwood and P. I. Barton. “How to solve a design centering problem.” In: *Mathematical Methods of Operations Research* 86.1 (2017), pp. 215–254. ISSN: 1432-2994. DOI: 10.1007/s00186-017-0591-3.

[65] J. C. Helton, J. D. Johnson, and W. L. Oberkampf. “An exploration of alternative approaches to the representation of uncertainty in model predictions.” In: *Reliability Engineering & System Safety* 85.1-3 (2004), pp. 39–71. ISSN: 09518320. DOI: 10.1016/j.ress.2004.03.025.

[66] E. M. Hendrix, C. J. Mecking, and T. H. Hendriks. “Finding robust solutions for product design problems.” In: *European Journal of Operational Research* 92.1 (1996), pp. 28–36. ISSN: 03772217. DOI: 10.1016/0377-2217(95)00082-8.

[67] D. Hose and M. Hanss. “Possibilistic calculus as a conservative counterpart to probabilistic calculus.” In: *Mechanical Systems and Signal Processing* 133 (2019), p. 106290. ISSN: 08883270. DOI: 10.1016/j.ymssp.2019.106290.

[68] D. Hose, M. Mäck, and M. Hanss. “A possibilistic approach to the optimization of uncertain systems.” In: *Proceedings of the ICVRAM-ISUMA UNCERTAINTIES Conferences 2018, Florianópolis, Brazil, April 2018* (2018).

[69] M. Inuiguchi and J. Ramík. “Possibilistic linear programming: a brief review of fuzzy mathematical programming and a comparison with stochastic programming in portfolio selection problem.” In: *Fuzzy Sets and Systems* 111.1 (2000), pp. 3–28. ISSN: 01650114. DOI: 10.1016/S0165-0114(98)00449-7.

[70] K. Jamison and W. A. Lodwick. “Minimizing unconstrained fuzzy functions.” In: *Fuzzy Sets and Systems* 103.3 (1999), pp. 457–464. ISSN: 01650114. DOI: 10.1016/S0165-0114(97)00183-8.

[71] C. N. Jones, E. C. Kerrigan, and J. M. Maciejowski. “Equality set projection: a new algorithm for the projection of polytopes in halfspace representation.” Technical report. Cambridge, UK: University of Cambridge, 2004.

[72] G. Kalai and G. M. Ziegler, eds. *Polytopes: combinatorics and computation*. Basel, Switzerland: Birkhäuser, 2000. ISBN: 9783764363512.

[73] M. M. Kamal. “Analysis and simulation of vehicle to barrier impact.” In: *SAE Technical Paper* (1970), p. 700414. DOI: 10.4271/700414.

[74] R. B. Kearfott and V. Kreinovich, eds. *Applications of interval computations*. Boston, USA: Springer, 1996. ISBN: 9781461334422.

[75] H. M. Kim. “Target cascading in optimal system design.” Ph.D. thesis. Ann Arbor, USA: The University of Michigan, 2001.

[76] A. D. Kiureghian and O. Ditlevsen. “Aleatory or epistemic? does it matter?” In: *Structural Safety* 31.2 (2009), pp. 105–112. ISSN: 0167-4730. DOI: 10.1016/j.strusafe.2008.06.020.

[77] A. Kossiakoff, W. N. Sweet, S. J. Seymour, and S. M. Biemer. *Systems engineering principles and practice*. Hoboken, USA: John Wiley & Sons, 2011. ISBN: 9781118001028.

[78] F. Kramer. *Passive Sicherheit von Kraftfahrzeugen: Biomechanik – Simulation – Sicherheit im Entwicklungsprozess*. Wiesbaden, Germany: Vieweg+Teubner, 2009. ISBN: 9783834892546.

[79] R. S. Krishnamachari and P. Y. Papalambros. “Hierarchical decomposition synthesis in optimal systems design.” In: *Journal of Mechanical Design* 119.4 (1997), pp. 448–457. ISSN: 10500472. DOI: 10.1115/1.2826389.

[80] A. Kusiak and N. Larson. “Decomposition and representation methods in mechanical design.” In: *Journal of Mechanical Design* 117.B (1995), pp. 17–24. ISSN: 10500472. DOI: 10.1115/1.2836453.

[81] A. Kusiak and E. Szczerbicki. “A formal approach to specifications in conceptual design.” In: *Journal of Mechanical Design* 114.4 (1992), pp. 659–666. ISSN: 10500472. DOI: 10.1115/1.2917057.

[82] A. Kusiak and J. Wang. “Decomposition of the design process.” In: *Journal of Mechanical Design* 115.4 (1993), pp. 687–695. ISSN: 10500472. DOI: 10.1115/1.2919255.

[83] J. C. Lagarias, J. A. Reeds, M. H. Wright, and P. E. Wright. *Convergence properties of the Nelder-Mead simplex method in low dimensions*. 1998.

[84] V. A. Lange, J. Fender, and F. Duddeck. “Relaxing high-dimensional constraints in the direct solution space method for early phase development.” In: *Optimization and Engineering* 19.4 (2018), pp. 887–915. DOI: 10.1007/s11081-018-9381-x.

[85] V. A. Lange, J. Fender, L. Song, and F. Duddeck. “Early phase modeling of frontal impacts for crashworthiness: from lumped mass–spring models to deformation space models.” In: *Proceedings of the Institution of Mechanical Engineers, Part D: Journal of Automobile Engineering* 233.12 (2019), pp. 3000–3015. DOI: 10.1177/0954407018814034.

[86] J. Lasserre and F. Roubellat. “Measuring decision flexibility in production planning.” In: *IEEE Transactions on Automatic Control* 30.5 (1985), pp. 447–452. DOI: 10.1109/TAC.1985.1103984.

[87] J. B. Lasserre. “An analytical expression and an algorithm for the volume of a convex polyhedron in Rn.” In: *Journal of Optimization Theory and Applications* 39.3 (1983), pp. 363–377. ISSN: 0022-3239. DOI: 10.1007/BF00934543.

[88] F. Leimbach and H. Kiebach. “Reparability and insurance ratings in the development of cars.” In: *Encyclopedia of automotive engineering*. Ed. by D. Crolla, D. E. Foster, T. Kobayashi, and N. Vaughan. Chichester, UK: John Wiley & Sons, 2014. ISBN: 9781118354179.

[89] A. S. Lewis. “Robust regularization.” Technical report. Burnaby, Canada: Simon Fraser University, 2002.

[90] B. Liu and K. Iwamura. “Chance constrained programming with fuzzy parameters.” In: *Fuzzy Sets and Systems* 94.2 (1998), pp. 227–237. ISSN: 01650114. DOI: 10.1016/S0165-0114(96)00236-9.

[91] B. Liu and J. Kacprzyk. *Theory and practice of uncertain programming*. Berlin, Germany: Springer, 2009. ISBN: 9783540894834.

[92] D. Ma and H. M. Lankarani. “A multibody/finite element analysis approach for modeling of crash dynamic responses.” In: *Journal of Mechanical Design* 119.3 (1997), pp. 382–387. ISSN: 10500472. DOI: 10.1115/1.2826359.

[93] MathWorks. *Optimization toolbox user’s guide (R2019a)*. 2019. URL: https://www.mathworks.com/help/pdf_doc/optim/optim_tb.pdf (visited on 02/02/2020).

[94] R. E. Melchers and A. T. Beck. *Structural reliability analysis and prediction*. Hoboken, USA: John Wiley & Sons, 2018. ISBN: 9781119265993.

[95] M. Milanese, J. Norton, H. Piet-Lahanier, and E. Walter, eds. *Bounding approaches for system identification*. New York, USA: Springer, 1996. ISBN: 9781475795479.

[96] R. E. Moore. *Methods and applications of interval analysis*. Philadelphia, USA: Society for Industrial and Applied Mathematics, 1979. ISBN: 9780898711615.

[97] T. S. Motzkin. “Beiträge zur Theorie der linearen Ungleichungen.” Ph.D. thesis. Basel, Switzerland: Universität Basel, 1936.

[98] National Highway Traffic Safety Administration. *Crash simulation vehicle models*. 2020. URL: https://www.nhtsa.gov/crash-simulation-vehicle-models (visited on 02/02/2020).

[99] National Highway Traffic Safety Administration. *Ratings*. 2020. URL: https://www.nhtsa.gov/ratings (visited on 02/02/2020).

[100] J. Nocedal and S. J. Wright. *Numerical optimization*. New York, USA: Springer, 2006. ISBN: 9780387303031. DOI: 10.1007/978-0-387-40065-5.

[101] J. C. O'Callahan, P. Avitabile, and R. Riemer. “System equivalent reduction expansion process.” In: *Proceedings of the 7th International Modal Analysis Conference, Las Vegas, US, February 1989* (1989).

[102] J. O'Rourke. “Finding minimal enclosing boxes.” In: *International Journal of Computer & Information Sciences* 14.3 (1985), pp. 183–199. ISSN: 0091-7036. DOI: 10.1007/BF00991005.

[103] M. O'Searcoid. *Metric spaces*. London, UK: Springer, 2007. ISBN: 9781846283697.

[104] J. H. Panchal, C. J. J. Paredis, J. K. Allen, and F. Mistree. “Managing design-process complexity: a value-of-information based approach for scale and decision decoupling.” In: *Journal of Computing and Information Science in Engineering* 9.2 (2009), p. 021005. DOI: 10.1115/1.3130791.

[105] I. Papaioannou, M. Daub, M. Drieschner, F. Duddeck, M. Ehre, L. Eichner, M. Eigel, M. Götz, W. Graf, L. Grasedyck, R. Gruhlke, D. Hömberg, M. Kaliske, D. Moser, Y. Petryna, and D. Straub. “Assessment and design of an engineering structure with polymorphic uncertainty quantification.” In: *Surveys for Applied Mathematics and Mechanics (GAMM-Mitteilungen)* 42.2 (2019), e201900009. DOI: 10.1002/gamm.201900009.

[106] P. Y. Papalambros and D. J. Wilde. *Principles of optimal design: modeling and computation*. Cambridge, UK: Cambridge University Press, 2017. ISBN: 9781107132672.

[107] A. Parkinson, C. Sorensen, and N. Pourhassan. “A general approach for robust optimal design.” In: *Journal of Mechanical Design* 115.1 (1993), p. 74. ISSN: 10500472. DOI: 10.1115/1.2919328.

[108] T. U. Pimmler and S. D. Eppinger. “Integration analysis of product decompositions.” In: *Proceedings of the 6th International Conference on Design Theory and Methodology Conference, Minneapolis, USA, September 1994* (1994).

[109] A. Prékopa. “On logarithmic concave measures and functions.” In: *Acta Scientiarum Mathematicarum* 34 (1973), pp. 335–343.

[110] A. Prékopa. “Logarithmic concave measures and related topics.” In: *Proceedings of the 1st International Conference on Stochastic Programming, Oxford, England, July 1974* (1974).

[111] S. S. Rao. *Engineering optimization*. Hoboken, USA: John Wiley & Sons, 2009. ISBN: 9780470549124.

[112] M. Rayamajhi. “Efficient robust shape optimization for crashworthiness.” Ph.D. thesis. London, UK: Queen Mary University of London, 2014.

[113] C. M. Rocco, J. A. Moreno, and N. Carrasquero. “Robust design using a hybrid-cellular-evolutionary and interval-arithmetic approach: a reliability application.” In: *Reliability Engineering & System Safety* 79.2 (2003), pp. 149–159. ISSN: 09518320. DOI: 10.1016/S0951-8320(02)00226-0.

[114] S. Roeser, ed. *Handbook of risk theory: epistemology, decision theory, ethics, and social implications of risk*. Dordrecht, Netherlands: Springer, 2012. ISBN: 9789400714335.

[115] K. H. Rosen. *Discrete mathematics and its applications*. New York City, USA: McGraw-Hill, 2019. ISBN: 9781259676512.

[116] A. Schrijver. *Theory of linear and integer programming*. Chichester, UK: John Wiley & Sons, 2000. ISBN: 9780471982326.

[117] W. Scott, E. Bonugli, H. Guzman, and D. Swartzendruber. “Reconstruction of low-speed crashes using the quasi-static force vs. deformation characteristics of the bumpers involved in the crashes.” In: *SAE International Journal of Passenger Cars - Mechanical Systems* 5.1 (2012), pp. 592–611. ISSN: 1946-4002. DOI: 10.4271/2012-01-0598.

[118] G. Shafer. *A mathematical theory of evidence*. Princeton, USA: Princeton University Press, 1976. ISBN: 9780691100425.

[119] S. P. Shary. “A new technique in systems analysis under interval uncertainty and ambiguity.” In: *Reliable Computing* 8.5 (2002), pp. 321–418. ISSN: 1573-1340. DOI: 10.1023/A:1020505620702.

[120] N. Z. Shor. *Nondifferentiable optimization and polynomial problems*. Boston, USA: Springer, 1998. ISBN: 9781441947925.

[121] M. Sniedovich. “From statistical decision theory to robust optimization: a maximin perspective on robust decision-making.” In: *Robustness analysis in decision aiding, optimization, and analytics*. Ed. by M. Doumpos. Cham, Switzerland: Springer International Publishing, 2016. ISBN: 9783319331195.

[122] L. Song, M. Pabst, F. Duddeck, and J. Fender. “A simplified model for barrier–vehicle interaction in a rear crash for early phase development and solution spaces.” In: *International Journal of Crashworthiness* 23.5 (2018), pp. 507–520. ISSN: 1358-8265. DOI: 10.1080/13588265.2017.1350091.

[123] M. M. Strauß. “Stochastic solution spaces for vehicle crash design.” B.Sc. thesis. Munich, Germany: Technische Universität München, 2019.

[124] L. P. Swiler, T. L. Paez, and A. L. Mayes. “Epistemic uncertainty quantification tutorial.” In: *Proceedings of the 27th Conference and Exposition on Structural Dynamics (IMAC-XXVII), Orlando, USA, February 2009* (2009).

[125] M. Szedlak. “Redundancy in linear systems: combinatorics, algorithms and analysis.” Ph.D. thesis. Zurich, Switzerland: ETH Zürich, 2017. DOI: 10.3929/ETHZ-B-000167108.

[126] D. Thunnissen. “Uncertainty classification for the design and development of complex systems.” In: *Proceedings of the 3rd Annual Predictive Methods Conference, Newport Beach, USA, June 2003* (2003).

[127] K. T. Ulrich, S. D. Eppinger, and M. C. Yang. *Product design and development*. New York City, USA: McGraw-Hill, 2020. ISBN: 9781260566437.

[128] M. E. Vogt, F. Duddeck, H. Harbrecht, F. Stutz, M. Wahle, and M. Zimmermann. “Computing solution-compensation spaces using an enhanced Fourier-Motzkin algorithm.” In: *Proceedings in Applied Mathematics and Mechanics (PAMM)* 18.1 (2018), e201800103. DOI: 10.1002/pamm.201800103.

[129] M. E. Vogt, F. Duddeck, M. Wahle, and M. Zimmermann. “Optimizing tolerance to uncertainty in systems design with early- and late-decision variables.” In: *IMA Journal of Management Mathematics* 30.3 (2018), pp. 269–280. ISSN: 1471-678X. DOI: 10.1093/imaman/dpy003.

[130] D. D. Walden, G. J. Roedler, K. J. Forsberg, R. D. Hamelin, and T. M. Shortell. *Incose systems engineering handbook: a guide for system life cycle processes and activities*. Hoboken, USA: John Wiley & Sons, 2015. ISBN: 9781118999400.

[131] P. Walley. *Statistical reasoning with imprecise probabilities*. Boston, USA: Springer, 1991. ISBN: 9780412286605.

[132] X. Wang and E. E. Kerre. “Reasonable properties for the ordering of fuzzy quantities (i).” In: *Fuzzy Sets and Systems* 118.3 (2001), pp. 375–385. ISSN: 01650114. DOI: 10.1016/S0165-0114(99)00062-7.

[133] X. Wang and E. E. Kerre. “Reasonable properties for the ordering of fuzzy quantities (ii).” In: *Fuzzy Sets and Systems* 118.3 (2001), pp. 387–405. ISSN: 01650114. DOI: 10.1016/S0165-0114(99)00063-9.

[134] W. Yao, X. Chen, W. Luo, M. van Tooren, and J. Guo. “Review of uncertainty-based multidisciplinary design optimization methods for aerospace vehicles.” In: *Progress in Aerospace Sciences* 47.6 (2011), pp. 450–479. ISSN: 03760421. DOI: 10.1016/j.paerosci.2011.05.001.

[135] B. D. Youn, K. K. Choi, L. Du, and D. Gorsich. “Integration of possibility-based optimization and robust design for epistemic uncertainty.” In: *Journal of Mechanical Design* 129.8 (2007), p. 876. ISSN: 10500472. DOI: 10.1115/1.2717232.

[136] L. A. Zadeh. “Fuzzy sets.” In: *Information and Control* 8.3 (1965), pp. 338–353. ISSN: 00199958. DOI: 10.1016/S0019-9958(65)90241-X.

[137] L. A. Zadeh. “The concept of a linguistic variable and its application to approximate reasoning—I.” In: *Information Sciences* 8.3 (1975), pp. 199–249. ISSN: 00200255. DOI: 10.1016/0020-0255(75)90036-5.

[138] L. A. Zadeh. “Fuzzy sets as a basis for a theory of possibility.” In: *Fuzzy Sets and Systems* 100 (1999), pp. 9–34. ISSN: 01650114. DOI: 10.1016/S0165-0114(99)80004-9.

[139] H.-J. Zimmermann. *Fuzzy set theory - and its applications*. Dordrecht, Netherlands: Springer, 2001. ISBN: 9789401038706.

[140] M. Zimmermann and J. E. von Hoessle. “Computing solution spaces for robust design.” In: *International Journal for Numerical Methods in Engineering* 94.3 (2013), pp. 290–307. ISSN: 00295981. DOI: 10.1002/nme.4450.

[141] M. Zimmermann, S. Königs, C. Niemeyer, J. Fender, C. Zeherbauer, R. Vitale, and M. Wahle. “On the design of large systems subject to uncertainty.” In: *Journal of Engineering Design* 28.4 (2017), pp. 233–254. ISSN: 0954-4828. DOI: 10.1080/09544828.2017.1303664.