Structural and Multidisciplinary Optimization (2020) 61:2097–2109
https://doi.org/10.1007/s00158-019-02456-8

RESEARCH PAPER

Optimizing component solution spaces for systems design

Marco Daub[1] · **Fabian Duddeck**[1,2] · **Markus Zimmermann**[1]

Received: 9 July 2019 / Revised: 11 November 2019 / Accepted: 17 November 2019 / Published online: 28 February 2020

Abstract

Systems design is concerned with breaking down large and complex problems on the system level into smaller and simpler problems on a component level. This can be accomplished by decomposing quantitative system requirements into requirements for each component. These new requirements can be expressed by component solution spaces, i.e., regions of permissible component design variables. They serve as design goals for component development and support decisions regarding component design. Component solution spaces should be chosen such that components can be designed independently of each other. This is the case when satisfying all component requirements implies satisfying the higher level system requirements. In addition, component solution spaces should be as large as possible to provide maximum flexibility for component design, and to encompass uncertainty. Motivated by related work, their shapes can be predefined as boxes, i.e., multi-dimensional intervals, which enables a simple visual and numerical representation. Unfortunately, this will make solution spaces small and associated requirements unnecessarily restrictive. In a new approach, the shapes of component solution spaces are also optimized to further enlarge their size. This is accomplished by decomposing the system performance function as a sum of predefined component performance functions and optimizing their individual contribution. For both the old and new approach, optimization schemes are presented which focus on linear system performance functions. Their effectiveness is demonstrated for a crash design problem.

Keywords Solution spaces · Systems engineering · Concurrent engineering · Uncertainty · Crashworthiness

1 Introduction

There are many approaches to the design of technical systems like vehicles and airplanes. One of them is designing the system as a whole by using classical optimization where a system performance function is minimized while fulfilling quantitative system requirements, see, e.g., (Avriel et al. 1973). However, if uncertainties are present, it is possible that the optimal design deteriorates, shows poor performance or requirement violation. This is especially the case if the design is selected in an early phase of development where epistemic uncertainties exist. Uncertainties are associated, for example, with controllable design variables and uncontrollable parameters, cf. Parkinson et al. (1993). Reasons for uncertainties in controllable variables are, for instance, variations from target design values due to manufacturing or more detailed design modifications in later development stages. Variations in parts of the system where the system designer has limited control, like changing operation conditions or modifications in the system not in the responsibility of the designer, can be considered as uncertainties in uncontrollable parameters. These circumstances may result in a time-consuming and cost-intensive design process.

In Beyer and Sendhoff (2007), a survey of various approaches, related to robust design, is given on how uncertainties can be treated in order to obtain an optimal system design. What these approaches have in common is that they are based on specific mathematical quantifications of uncertainty which are often non-available in the early phase of systems design. An approach that is not based on

Communicated by:Responsible Editor: Somanath Nagendra

✉ Marco Daub
marco.daub@tum.de

Fabian Duddeck
duddeck@tum.de

Markus Zimmermann
zimmermann@tum.de

[1] Technische Universität München, Arcisstr. 21, 80333 München, Germany

[2] Queen Mary University of London, Mile End Road, London E14NS, UK

a mathematical quantification of uncertainty is presented in Hendrix et al. (1996). Here, a feasible design with maximum distance to the region that violates the system requirements is sought in order to tolerate maximum uncertainties in controllable variables. If the Chebyshev distance is used, a maximum box that is entirely contained in the permissible region is spanned around the robust design. In Zimmermann and von Hoessle (2013), a box inside the permissible region that is maximized with respect to its volume is called an optimal box-shaped solution space. Along with being able to encompass uncertainties, box-shaped solution spaces have the property to fully decouple system requirements as they consist of permissible intervals for each design variable. This allows a flexible and independent selection of design variables. Hence, this approach is capable of improving the design process regarding costs and time, cf. Zimmermann et al. (2017).

However, a major drawback of using box-shaped solution spaces is that many designs which do not violate the constraints cannot be found within box-shaped solution spaces. This limits the options for systems design. To reduce this loss, optimal *component solution spaces* are proposed. They are defined as the set of permissible design variable values associated with components. Each component specifies a group of design variables that may interact with each other. Like box-shaped solution spaces, component solution spaces are based on an optimal system requirement decomposition. However, the requirements are not fully decoupled for component solution spaces; they are only decoupled between components. This increases the available solution space and therefore the freedom for design work. Moreover, larger uncertainties can be treated.

The paper is organized as follows: In Section 2, basics used for systems design are defined and the method of optimal component solution spaces is introduced. A differentiation is made between box-shaped and arbitrarily shaped component solution spaces. Furthermore, numerical algorithms are proposed to obtain optimal component solution spaces. A simple crash design problem is described in Section 3 and optimal component solution spaces are computed and compared. In Section 4, the same is done for a realistic crash design problem. In Section 5, a conclusion is given.

2 Component solution spaces

2.1 Definitions

Assume a system designer's perspective in the early development phase. Suppose that a system model exists; however, design variables have not yet assumed a fixed value. The system is composed of n components where each design variable belongs to one of the components. Thus, the kth component is described by d^k-independent design variables $x_i^k \in \mathbb{R}$, $i = 1, \ldots, d^k$, which assigns d^k degrees of freedom to the component. Its design variables are collected in a vector $x^k \in \mathbb{R}^{d^k}$ and named *component design* of the kth component. Together, the component designs constitute the *system design* $(x^1, \ldots, x^n)$. In Fig. 1, such a system is visualized.

For given design variables, the system responses, denoted by $z \in \mathbb{R}^m$, are determined uniquely by the system model. This relationship is expressed by the system performance function f where design variables serve as inputs and system responses as outputs, typically $z = f(x^1, \ldots, x^n)$.

In systems design, several constraints must be met. In general, constraints on design variables and on system responses are distinguished:

- The set of system designs with $x_{\mathrm{ds},i}^{\mathrm{l},k} \leq x_i^k \leq x_{\mathrm{ds},i}^{\mathrm{u},k}$ for $i = 1, \ldots, d^k$, $k = 1, \ldots, n$ is called *design space* Ω_{ds} of the system. Similarly, the design space for the kth component design is named Ω_{ds}^k.
- The system responses must fulfill performance requirements which can be expressed mathematically as $f_j(x^1, \ldots, x^n) \leq f_{\mathrm{c},j}$ for $j = 1, \ldots, m$. Here, only upper bounds are considered as thresholds because any lower bound can be transformed into an upper bound by multiplication with -1.

A system design that satisfies both constraints 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}, i.e.,

$$\Omega_{\mathrm{c}} = \{(x^1, \ldots, x^n) \in \Omega_{\mathrm{ds}} : \ f(x^1, \ldots, x^n) \leq f_{\mathrm{c}}\}. \tag{1}$$

In the literature, Ω_{c} is also called feasible solution set (Milanese et al. 1996) or permissible design space (Graf et al. 2018).

2.2 Problem statement

Systems design is concerned with enabling independent component development to reduce overall design complexity. This can be realized by decoupling the performance requirements for the system into requirements for the

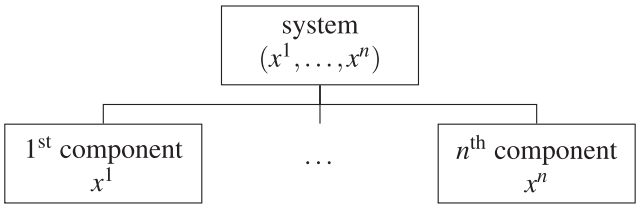


Fig. 1 A system composed of n components each specified by a design variable vector x^k, $k = 1, \ldots, n$

components yielding *component solution spaces* Ω^k, $k = 1, \ldots, n$. This means that instead of having one system designer who targets designs within the complete system solution space Ω_c, there are n component designers who target component designs within component solution spaces Ω^k, $k = 1, \ldots, n$.

In order to guarantee that the component designers can work independently, it must be ensured that $(x^1, \ldots, x^n) \in \Omega_c$ holds for $x^k \in \Omega^k$, $k = 1, \ldots, n$. Therefore, it is necessary and sufficient that a system solution space formed by the Cartesian product of the single Ω^k is a subset of Ω_c, i.e.,

$$\Omega^1 \times \cdots \times \Omega^n \subseteq \Omega_c. \tag{2}$$

The overall idea is visualized in Fig. 2.

Among all possibilities to choose component solution spaces, optimal component solution spaces that provide most the flexibility for component designers are preferred. Here, flexibility is quantified by an objective function, which measures the infinite amount of permissible system designs covered in $\Omega^1 \times \cdots \times \Omega^n$. This can be represented by the volume of $\Omega^1 \times \cdots \times \Omega^n$. The resulting mathematical optimization problem reads as follows:

$$\begin{aligned} &\underset{\Omega^1, \ldots, \Omega^n}{\text{maximize}} \ \operatorname{vol}(\Omega^1 \times \cdots \times \Omega^n) \\ &\text{subject to} \ \ \Omega^1 \times \cdots \times \Omega^n \subseteq \Omega_c. \end{aligned} \tag{3}$$

In Fig. 3, an example of two optimal component solution spaces Ω^1 and Ω^2 is shown.

The particular challenge of solving problem (3) is that the optimization variables are sets instead of values in real coordinate space. An approach to tackle this problem is to simplify the problem such that a so-called design centering problem, see Harwood and Barton (2017), is obtained where the optimization variables are in real coordinate space.

2.3 Box-shaped component solution spaces

In this approach, component solution spaces are considered which are box-shaped, i.e., high-dimensional intervals. Box-shaped Ω^k can be uniquely defined by the vectors $x^{l,k}, x^{u,k} \in \mathbb{R}^{d^k}$ via

$$\Omega^k = [x^{l,k}, x^{u,k}], \tag{4}$$

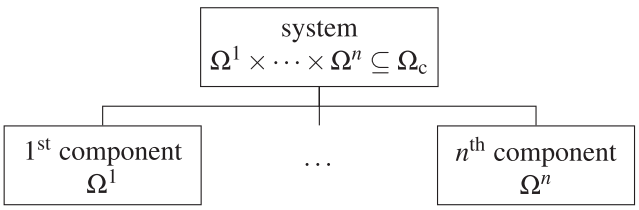


Fig. 2 Decomposition of the complete system solution space into component solution spaces

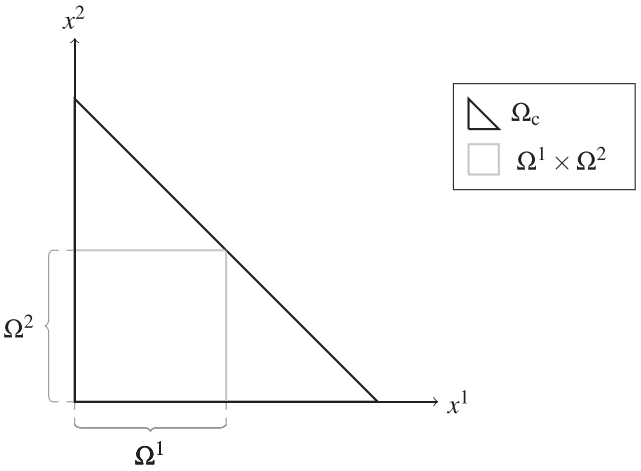


Fig. 3 Example of optimal component solution spaces Ω^1 and Ω^2 for a system composed of two components with one degree of freedom each

where $[x^{l,k}, x^{u,k}]$ is a d^k-dimensional interval for $k = 1, \ldots, n$, providing a simple and real-valued description for the component solution spaces. Thus, corresponding shape constraints must be included in problem (3), and the new optimization problem reads as follows:

$$\begin{aligned} &\underset{x^{l,k}, x^{u,k}}{\text{maximize}} \ \prod_{k=1}^{n} \prod_{i=1}^{d^k} (x_i^{u,k} - x_i^{l,k}) \\ &\text{subject to} \ \ [x^{l,1}, x^{u,1}] \times \cdots \times [x^{l,n}, x^{u,n}] \subset \Omega_c \end{aligned} \tag{5}$$

where the optimization variables are in real coordinate space. Problem (5) is identical to the optimization problem for computing box-shaped solution spaces proposed in Zimmermann and von Hoessle (2013). Hence, the Cartesian product of box-shaped component solution spaces can always be considered as a box-shaped solution space in terms of Zimmermann and von Hoessle (2013) and the other way around. In the literature, the calculation of solutions to problems similar to problem (5) has been studied extensively and is used for systems design. Further examples can be found in Bemporad et al. (2004), Daub (2017), Fender (2013), Götz et al. (2012), Graf et al. (2018), Graff et al. (2016), Harwood and Barton (2017), and Milanese et al. (1996). The associated algorithms mainly differ depending on the type of the system performance function f, e.g., linear, non-linear, and black-box. Particularly, if the functions f_j, $j = 1, \ldots, m$ are linear, i.e.,

$$f(x^1, \ldots, x^n) = \sum_{k=1}^{n} A^k x^k \tag{6}$$

with $A^k \in \mathbb{R}^{m\times d^k}$, $k = 1,\dots,n$, problem (5) can be simplified, see Harwood and Barton (2017). Defining $b = f_c$, it yields:

$$\begin{aligned}&\underset{x^{l,k},x^{u,k}}{\text{maximize}} \ \prod_{k=1}^{n}\prod_{i=1}^{d^k}(x_i^{u,k}-x_i^{l,k})\\ &\text{subject to } \ x^{l,k}\le x^{u,k},\ -x^{l,k}\le -x_{ds}^{l,k},\ x^{u,k}\le x_{ds}^{u,k},\ k=1,\dots,n,\\ &\qquad \sum_{k=1}^{n}(a_j^{k,T}W_j^{l,k}x^{l,k}+a_j^{k,T}W_j^{u,k}x^{u,k})\le b_j,\ j=1,\dots,m,\end{aligned} \tag{7}$$

where $a_j^{k,T}$ is the jth row of A^k and $W_j^{l,k}$, $W_j^{u,k}$ are diagonal $d^k\times d^k$ matrices where their ith entries on the diagonals are given by the following:

$$w_{j,i}^{l,k}=\begin{cases}1 \text{ for } a_{j,i}^k\le 0,\\ 0 \text{ for } a_{j,i}^k>0\end{cases},\quad w_{j,i}^{u,k}=\begin{cases}0 \text{ for } a_{j,i}^k\le 0,\\ 1 \text{ for } a_{j,i}^k>0\end{cases} \tag{8}$$

for $i = 1,\dots,d^k$, $k = 1,\dots,n$, $j = 1,\dots,m$. Problem (7) can be solved numerically by using standard gradient-based optimization algorithms.

2.4 Arbitrarily shaped component solution spaces

In contrast to box-shaped component solution spaces, there are no shape constraints in this approach. Thus, arbitrary shapes can be assumed. If Ω^k, $k = 1,\dots,n$, are represented by

$$\Omega^k=\{x^k\in\Omega_{ds}^k:\ f^k(x^k)\le f_c^k\}, \tag{9}$$

the goal is to find optimal functions f^k and corresponding thresholds f_c^k for $k = 1,\dots,n$ to obtain an optimal solution of problem (3). The functions f^k, $k = 1,\dots,n$ can be interpreted as component performance functions that contribute to the total system performance. If the optimal functions f^k are known a-priori, only the values of f_c^k must be sought in order to obtain optimal component solution spaces. This, for example, is the case when the system performance function can be written in the form as follows:

$$f(x^1,\dots,x^n)=\sum_{k=1}^{n}f^k(x^k), \tag{10}$$

cf. Daub (2017). To express the dependency of Ω^k on f_c^k in here, the notation $\Omega^k(f_c^k)$ is used. Then, problem (3) reduces to the optimization problem as follows:

$$\begin{aligned}&\underset{f_c^1,\dots,f_c^n}{\text{maximize}} \ \text{vol}(\Omega^1(f_c^1)\times\cdots\times\Omega^n(f_c^n))\\ &\text{subject to } \ \sum_{k=1}^{n}f_c^k\le f_c.\end{aligned} \tag{11}$$

Again, this is an optimization problem where the optimization variables are in real coordinate space. The optimization constraints guarantee that condition (2) is fulfilled. As the objective function increases if any entry of f_c^k, $k = 1,\dots,n$, is increased, a maximum of problem (11) is always obtained for $\sum_{k=1}^{n}f_c^k = f_c$. Note that linear system performance functions f_j are incorporated in this approach as they can be written in the form of (10), compare equation (6). For more general system performance functions, a reduction of problem (3) is more difficult.

2.5 Numerical implementation

A challenge for solving problem (11) is the computation of the volume of $\Omega^1(f_c^1)\times\cdots\times\Omega^n(f_c^n)$. If the single d^k-dimensional volumes of $\Omega^k(f_c^k)$, $k = 1,\dots,n$ are known, the total volume is obtained by multiplication. The single volumes can be approximated or calculated analytically:

- *Volume approximation*: Monte Carlo integration can be used for an efficient approximation of the volume of $\Omega^k(f_c^k)$, see (Evans and Swartz 2000). Therefore, N_t independent, uniformly distributed sample points are generated in Ω_{ds}^k, where f^k is evaluated at each sample point. If $f^k(x^k)\le f_c^k$ holds, the sample point x^k is considered as permissible. Now, the volume of $\Omega^k(f_c^k)$ can be calculated by dividing the number of permissible sample points, denoted by N_p, by N_t, and multiplying with the volume of Ω_{ds}^k, i.e.,

$$\text{vol}(\Omega^k(f_c^k))=\frac{N_p}{N_t}\prod_{i=1}^{d^k}(x_{ds,i}^{u,k}-x_{ds,i}^{l,k}). \tag{12}$$

- *Exact volume computation*: For linear performance functions, $\Omega^k(b^k)$, i.e.,

$$\Omega^k(b^k)=\{x^k\in\Omega_{ds}^k:\ A^kx^k\le b^k\} \tag{13}$$

with $b^k = f_c^k$ is a polytope. Its volume can be computed exactly by using the method proposed in Lasserre (1983). Here, the volume of a polytope which is described by $Ax\le b$, $A\in\mathbb{R}^{m\times d}$, $b\in\mathbb{R}^m$, is computed via a recursion scheme. Let $V(d,A,b)$ be the volume of this polytope, then:

$$V(d,A,b)=\frac{1}{d}\sum_{j=1}^{m}\frac{b_j}{|a_{j,i(j)}|}V_j(d-1,\tilde{A}_{i(j)},\tilde{b}), \tag{14}$$

where $V_j(d-1,\tilde{A}_{i(j)},\tilde{b})$ is the volume of the jth face, obtained by eliminating $x_{i(j)}$ in $Ax\le b$. For this purpose, $a_{i(j)}^Tx=b_{i(j)}$ is used where $a_{i(j)}^T$ is the ith row of A which yields $\tilde{A}_{i(j)}x\le\tilde{b}$, $\tilde{A}_{i(j)}\in\mathbb{R}^{(m-1)\times(d-1)}$, $\tilde{b}\in\mathbb{R}^{m-1}$. Other methods to calculate the volume of a polytope require the calculating of its corner points, see Büeler et al. (2000) for an overview. For arbitrary functions f^k, analytical formulae to calculate the volume of $\Omega^k(f_c^k)$, $k = 1,\dots,n$ are either not available or more complex.

In general, the objective function of problem (11) is not differentiable, see Erschen (2018) for an example. Therefore, non-gradient-based optimization algorithms such as direct search methods are preferred.

Suitable initial values $(f_{\mathrm{c},0}^1, \ldots, f_{\mathrm{c},0}^n)$ for problem (11) can be obtained by solving the minimax problem as follows:

$$\begin{aligned} &\underset{x^1,\ldots,x^n}{\text{minimize}} \ \max\{f_j(x^1,\ldots,x^n) - f_{\mathrm{c},j} : \ j = 1,\ldots,m\} \\ &\text{subject to } x_{\mathrm{ds}}^{\mathrm{l},k} \le x^k \le x_{\mathrm{ds}}^{\mathrm{u},k}, \ k = 1,\ldots,n. \end{aligned} \tag{15}$$

and setting $f_{\mathrm{c},0}^k = f^k(x^k) + \epsilon^k, \epsilon^k \in \mathbb{R}_+^m, k = 1,\ldots,n$ with $\sum_{k=1}^{n} (f^k(x^k) + \epsilon^k) \le f_{\mathrm{c}}$ for an optimal solution of problem (15). Note that problem (15) can be also used for choosing initial values for problem (5). Moreover, if a solution of problem (5) is already available, i.e., box-shaped component solution spaces, initial values for problem (11) can be also obtained by setting $f_{\mathrm{c},j,0}^k = \max\{f_j^k(x^k) : \ x^{\mathrm{l},k} \le x^k \le x^{\mathrm{u},k}\}, \ j = 1,\ldots,m, k = 1,\ldots,n$.

In the following two sections, optimal box-shaped and optimal arbitrarily shaped component are computed and compared for crash design problems. In doing so, the numerical approaches presented in this paper are used.

3 Application to a simple crash design problem

3.1 System specifications

The problem of designing a vehicle front structure consisting of two sections modeled as components is considered, compare (Zimmermann and von Hoessle 2013). The problem is visualized in Fig. 4. The vehicle mass m is lumped at the end of the two front end components. Together with the deformation length of the components $\bar{s}^1$ and $\bar{s}^2$, they are assumed to be uncontrollable parameters. They are assumed without deeper knowledge for the early design stage and define the frame of the first decisions with $m = 1500\,\mathrm{kg}$, $\bar{s}^1 = \bar{s}^2 = 300\,\mathrm{mm}$. Further detailed properties of the components still need to be chosen such that the overall vehicle meets crash requirements. These are formulated with respect to quantities measured in crash tests. For example, in the US-NCAP crash load case, a vehicle is driven against a rigid barrier with a velocity of $v_0 = 15.6\,\frac{\mathrm{mm}}{\mathrm{ms}}$. Here, there are requirements on the system performances concerning the absorbed energy, the maximum acceleration of the vehicle, and the progressive order of deformation, see Fender (2013). All of these responses can be computed from force–deformation characteristics which are functional properties of the two components and represent the force levels required for plastic deformation. Design variables are used to define the force–deformation characteristics $F^1(s)$ for $s \in [0\,\mathrm{mm}, \bar{s}^1)$ and $F^2(s)$ for $s \in [0\,\mathrm{mm}, \bar{s}^2)$. Note that in this simple example, the mass m.

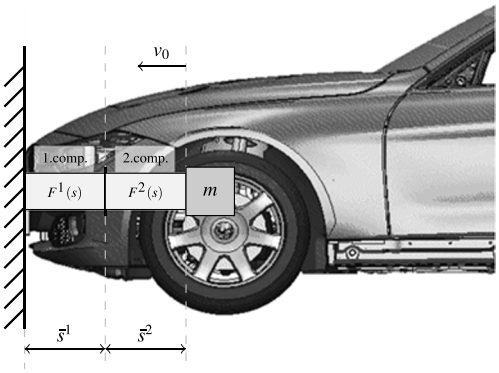


Fig. 4 Vehicle front structure for the simple crash design problem with two components and force-deformation characteristics $F^1(s)$, $s \in [0\,\mathrm{mm}, \bar{s}^1)$ and $F^2(s)$, $s \in [0\,\mathrm{mm}, \bar{s}^2)$

Due to technical limitations in this crash design problem, it is not possible to build force–deformation characteristics $F^k(s)$ that go below 0 kN and above 500 kN for $s \in [0\,\mathrm{mm}, \bar{s}^k), k = 1, 2$. The requirements on the responses can be formulated mathematically as follows:

- *Energy absorption*: The impact energy $\frac{1}{2}mv_0^2$ must be completely absorbed, meaning

$$-\int_{0\,\mathrm{mm}}^{\bar{s}^1} F^1(s)\,\mathrm{d}s - \int_{0\,\mathrm{mm}}^{\bar{s}^2} F^2(s)\,\mathrm{d}s \le -\frac{1}{2}mv_0^2. \tag{16}$$

- *Maximum acceleration*: The deceleration must be smaller than a critical threshold value a_{c}, i.e.,

$$F^1(s) \le ma_{\mathrm{c}} \tag{17}$$

for all $s \in [0\,\mathrm{mm}, \bar{s}^1)$ and

$$F^2(s) \le ma_{\mathrm{c}} \tag{18}$$

for all $s \in [0\,\mathrm{mm}, \bar{s}^2)$. Here, $a_{\mathrm{c}} = 0.3\,\frac{\mathrm{mm}}{\mathrm{ms}^2}$ is used.

- *Progressive order of deformation*: The ordered deformation of the vehicle must start at the front, meaning

$$F^1(s) \le F^2(0\,\mathrm{mm}) \tag{19}$$

for all $s \in [0\,\mathrm{mm}, \bar{s}^1)$.

For computing optimal component solution spaces with the approaches presented in this work, force–deformation characteristics with a finite number of design variables are considered. In the following, the crash design problem is investigated under a various number of design variables. Both approaches from the previous chapter are used for this purpose and are compared.

3.2 Components with one degree of freedom

First, constant force–deformation characteristics are considered for the simple crash design problem. Each of them has one degree of freedom which means $F^k(s) = F_1^k$ for $s \in [0\,\text{mm}, \bar{s}^k)$. Here, $F_1^k, k = 1, 2$, are the design variables. In Fig. 7, examples of such characteristics are shown.

The design space of the system for constant force–deformation characteristics is $\Omega_{\text{ds}} = [0\,\text{kN}, 500\,\text{kN}]^2$, and the system performance functions become linear of the form $f(x^1, x^2) = A^1x^1 + A^2x^2$. Hence, the performance requirements represent a system of linear inequalities, i.e.,

$$\underbrace{\begin{pmatrix} -\bar{s}^1 \\ 1 \\ 0 \\ 1 \end{pmatrix}}_{=A^1} \begin{pmatrix} F_1^1 \end{pmatrix} + \underbrace{\begin{pmatrix} -\bar{s}^2 \\ 0 \\ 1 \\ -1 \end{pmatrix}}_{=A^2} \begin{pmatrix} F_1^2 \end{pmatrix} \leq \underbrace{\begin{pmatrix} -\frac{1}{2}mv_0^2 \\ ma_{\text{c}} \\ ma_{\text{c}} \\ 0 \end{pmatrix}}_{=b} \tag{20}$$

where the first row belongs to inequality (16), the second row to (17), the third row to (18), and the last row to (19).

Now, optimal component solution spaces Ω^k can be computed for F_1^k, $k = 1, 2$. Box-shaped component solution spaces Ω_{bs}^k are accomplished by solving problem (7) and arbitrarily shaped component solution spaces Ω_{as}^k by using the exact method from Section 2.5. In Fig. 5, the corresponding component solution spaces are visualized.

The component solution spaces for both approaches coincide, i.e., are identical intervals. This is due to the consistence of the optimization problems (5) and (11) for convex system performance functions and $d^k = 1$, compare (Daub 2017).

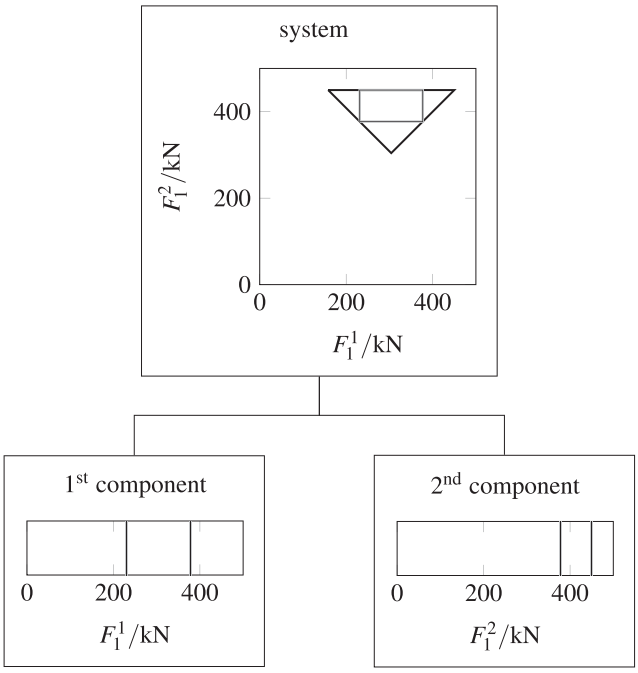


Fig. 5 Decomposition of the complete system solution space Ω_{c} (thick black) for the simple crash design problem with $d^k = 1$ into Ω_{bs}^1, Ω_{bs}^2 (thin black) and Ω_{as}^1, Ω_{as}^2 (gray)

3.3 Components with two degrees of freedom

Second, the simple crash design problem is considered where each force–deformation characteristic is modeled as a linear function. In detail, it is as follows:

$$F^k(s) = F_1^k \underbrace{\frac{\bar{s}^k - s}{\bar{s}^k}}_{=\phi_1^k(s)} + F_2^k \underbrace{\frac{s}{\bar{s}^k}}_{=\phi_2^k(s)} \tag{21}$$

for $s \in [0\,\text{mm}, \bar{s}^k)$, $k = 1, 2$. This means that every force–deformation characteristic has two degrees of freedom. The functions ϕ_1^k and ϕ_2^k are similar to linear basis functions in finite element theory. Examples of such characteristics are shown in Fig. 7.

Here, the design space of the system is $\Omega_{\text{ds}} = [0\,\text{kN}, 500\,\text{kN}]^4$ and the system performance functions are linear again. Thus, the performance requirements can be written as a system of linear inequalities, i.e.,

$$\underbrace{\begin{pmatrix} -\frac{\bar{s}^1}{2} & -\frac{\bar{s}^1}{2} \\ 1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \end{pmatrix}}_{=A^1} \begin{pmatrix} F_1^1 \\ F_2^1 \end{pmatrix} + \underbrace{\begin{pmatrix} -\frac{\bar{s}^2}{2} & -\frac{\bar{s}^2}{2} \\ 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & 1 \\ -1 & 0 \\ -1 & 0 \end{pmatrix}}_{=A^2} \begin{pmatrix} F_1^2 \\ F_2^2 \end{pmatrix} \leq \underbrace{\begin{pmatrix} -\frac{1}{2}mv_0^2 \\ ma_{\text{c}} \\ ma_{\text{c}} \\ ma_{\text{c}} \\ ma_{\text{c}} \\ 0 \\ 0 \end{pmatrix}}_{=b} \tag{22}$$

where the first row belongs to inequality (16), the second and third row to (17), the following two rows to (18), and the last two rows to (19). Note that other force–deformation characteristics with two degrees of freedom may be also reasonable, for example, piece-wise constant functions $F^k(s) = F_1^k$, $s \in [0, \frac{\bar{s}^k}{2})$ and $F^k(s) = F_2^k$, $s \in [\frac{\bar{s}^k}{2}, \bar{s}^k)$, $k = 1, 2$.

Now, optimal component solution spaces Ω_{as}^k and Ω_{bs}^k are computed for (F_1^k, F_2^k), $k = 1, 2$, like in the previous case and visualized in Fig. 6.

The volume of $\Omega_{\text{as}}^1 \times \Omega_{\text{as}}^2$, denoted by V_{as}, is approximately 2.40 times bigger than the volume of $\Omega_{\text{bs}}^1 \times \Omega_{\text{bs}}^2$, denoted by V_{bs}. This increase in volume makes system design more flexible and more uncertainty can be tolerated.

3.4 Components with arbitrary degrees of freedom

This procedure can be extended to force–deformation characteristics with arbitrary, but finitely many degrees of freedom. Here, piece-wise linear and continuous

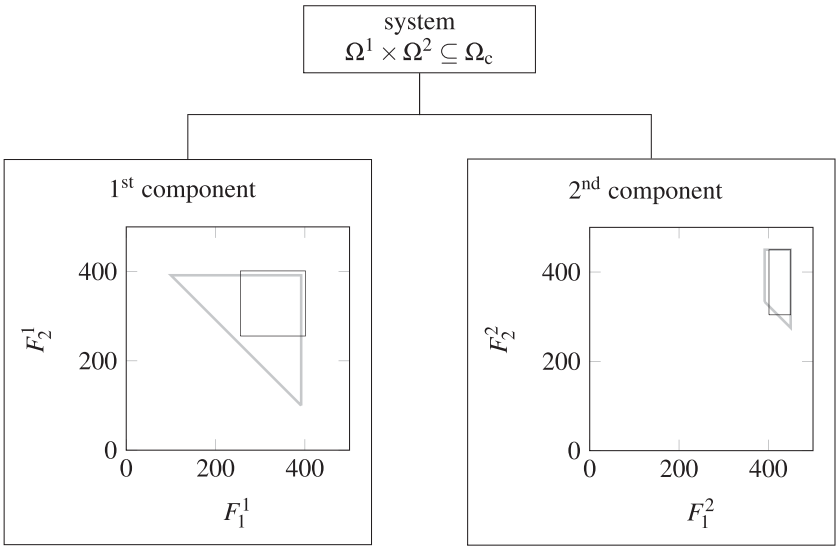


Fig. 6 Decomposition of the complete system solution space Ω_c for the simple crash design problem with $d^k = 2$ into Ω^1_{bs}, Ω^2_{bs} (black) and Ω^1_{as}, Ω^2_{as} (gray)

characteristics are as follows:

$$F^k(s) = \sum_{i=1}^{d^k} F_i^k \phi_i(s) \tag{23}$$

for $s \in [0\,\text{mm}, \bar{s}^k)$ and $d^k > 1$ are considered. The functions ϕ_i^k, $i = 1, \ldots, d^k$ are similar to linear basis functions from finite element theory. They are piece-wise linear, continuous, and $\phi_i((j-1)\frac{\bar{s}^k}{d^k-1}) = 0$ for $i \neq j$, $\phi_i((j-1)\frac{\bar{s}^k}{d^k-1}) = 1$ for $i = j$, $i, j = 1, \ldots, d^k$. In Fig. 7, examples of such characteristics are shown. As in the previous cases, the system performance functions are linear and the performance requirements given by the inequalities (16)–(19) can be formulated as a system of linear inequalities. The optimal component solution spaces Ω^k_{as} and Ω^k_{bs}, $k = 1, 2$, can then be computed as done previously. However, the visualization of these sets as geometric shapes is difficult for $d^k > 3$.

To overcome this problem, regions of possibly permissible characteristics can be used which are related to parallel coordinate plots. A force–deformation characteristic given by (23) is said to be permissible if $F^k \in \Omega^k$ holds where $F^k = (F_1^k, \ldots, F_{d^k}^k)$. The region of possibly permissible characteristics is then defined as the region between the two bounding characteristics defined by $F^{l,k}_{\text{out},i} = \min\{F_i^k : F^k \in \Omega^k\}$ and $F^{u,k}_{\text{out},i} = \max\{F_i^k : F^k \in \Omega^k\}$ for $i = 1, \ldots, d^k$. The d^k-dimensional interval $[F^{l,k}_{\text{out}}, F^{u,k}_{\text{out}}]$ forms the minimal outer box of the component solution space Ω^k. However, any force–deformation inside this region is not guaranteed to be permissible. In order to get an idea about the quantity of permissible force–deformation characteristics within the region of possibly permissible ones, the average edge lengths of Ω^k and $[F^{l,k}_{\text{out}}, F^{u,k}_{\text{out}}]$ can be calculated and compared. For its ratio r^k, it holds as follows:

$$r^k = \sqrt[d^k]{\frac{\text{vol}(\Omega^k)}{\prod_{i=1}^{d^k}(F^{u,k}_{\text{out},i} - F^{l,k}_{\text{out},i})}} \tag{24}$$

In Fig. 7, Ω^k is visualized for different degrees of freedom together with the region of possibly permissible characteristics.

3.5 Box-shaped vs. arbitrarily-shaped component solution spaces

An advantage of using box-shaped component solution spaces is that the regions of possibly permissible characteristics do not contain non-permissible force–deformation characteristics as $\Omega^k_{bs} = [F^{l,k}_{\text{out}}, F^{u,k}_{\text{out}}]$, and therefore $r^k = 1$ always holds. However, the volume V_{bs} and therefore the regions of possibly permissible characteristics, become rapidly smaller with growing d^k compared to the volume of V_{as}. This difference can also be assessed in terms of average edge length of $\Omega^1_{bs} \times \Omega^2_{bs}$, $l_{bs} = \sqrt[2d^k]{V_{bs}}$, and of $\Omega^1_{as} \times \Omega^2_{as}$, $l_{as} = \sqrt[2d^k]{V_{as}}$, see Fig. 8.

This means, the greater d^k, the more permissible force–deformation characteristics can be found within the regions of possibly permissible characteristics for arbitrarily shaped compared to box-shaped component solution spaces. However, with increasing d^k, calculating arbitrarily shaped component solution spaces with exact volume computation requires much more computing time than the calculation of box-shaped solution spaces, compare Fig. 9.

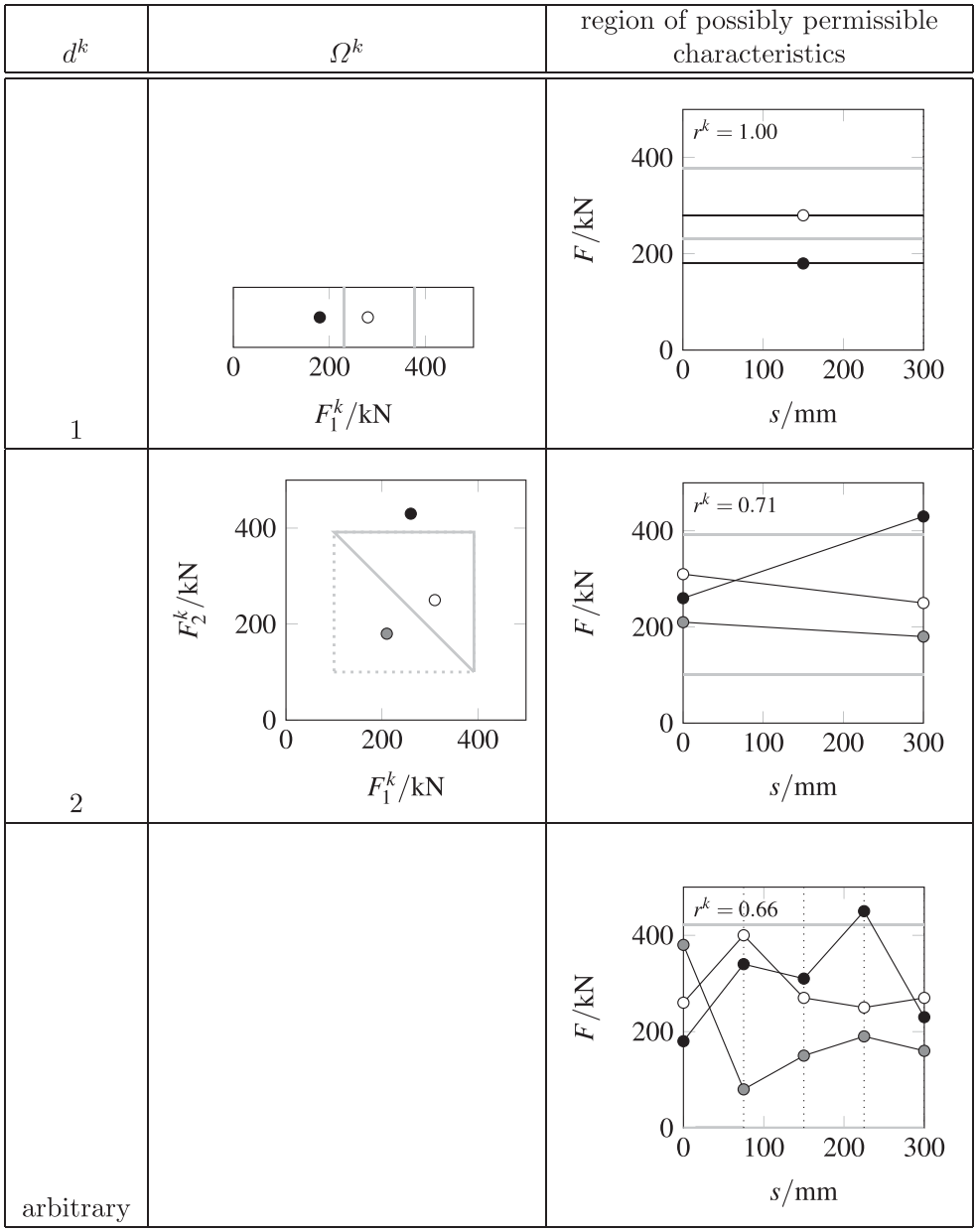


Fig. 7 Component solution spaces Ω^k and their corresponding region of possibly permissible characteristics both bounded by solid gray lines for different d^k. Inside, there are permissible (white dots) or non-permissible (gray dots) force–deformation characteristics. Outside, they are always non-permissible (black dots)

4 Application to a realistic crash design problem

4.1 System specifications

A realistic crash design problem is considered where the frontal structure of the vehicle is composed of 8 crash-relevant components, i.e., $n = 8$, which are distributed to three different load paths, denoted by $n_{\text{lp}} = 3$. In contrast to Section 3, these components are not fully deformable in the event of a crash. They are characterized by a deformation length in horizontal direction, denoted by $\bar{s}^k$, $k = 1, \ldots, 8$, before they behave as if they were rigid. From knowledge about the crash behavior of similar components, their deformation lengths are given in Table 1. The sum of the deformation length and the non-deformable length yield the real length of the component. Furthermore, the mass of the vehicle can be distributed to 5 points, denoted by m_i, $i = 1, \ldots, 5$. The values of these discrete mass points are also given in Table 2. They add up to the total mass of the vehicle. The masses m_i, $i = 1, \ldots, 4$, are each part of one load path, the mass m_5 acts on all load paths as it represents the rear end of the vehicle. All in all, a model in *geometry space* is obtained which is visualized together with the underlying vehicle structure in Fig. 10a.

From geometry space, the parts of the structure that deform simultaneously during a vehicle crash can be deduced. For that reason, the non-deformable lengths of

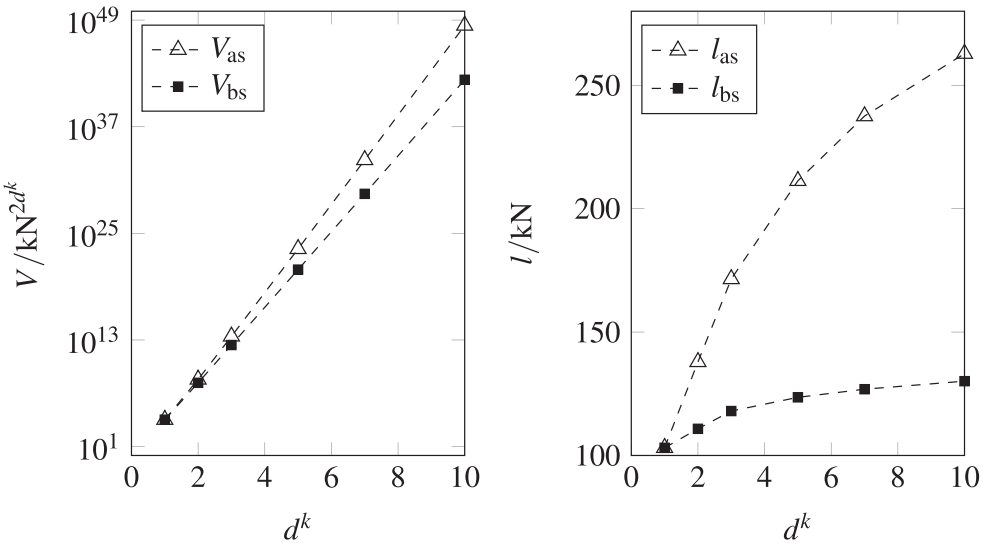


Fig. 8 Total volume (left) and average edge length (right) of box-shaped (bs) and arbitrarily shaped (as) component solution spaces

the components are subtracted from the real component length. Hence, components which are shortened to their deformation lengths $\bar{s}^k$, $k = 1, \ldots, 8$ are considered. They are stapled on the right. Now, the parts of the vehicle that deform simultaneously are aligned vertically. This yields a model in *deformation space* (see Fig. 10b).

For more information about this modeling approach see, e.g., Fender (2013). Note that any interaction between the different load paths due to structural connection is assumed to be negligible here. This is, for example, the case for the connection between the first and forth component and the connection between the second and sixth component (see Fig. 10a). An extension of this approach for non-negligible interactions is proposed in Lange et al. (2019).

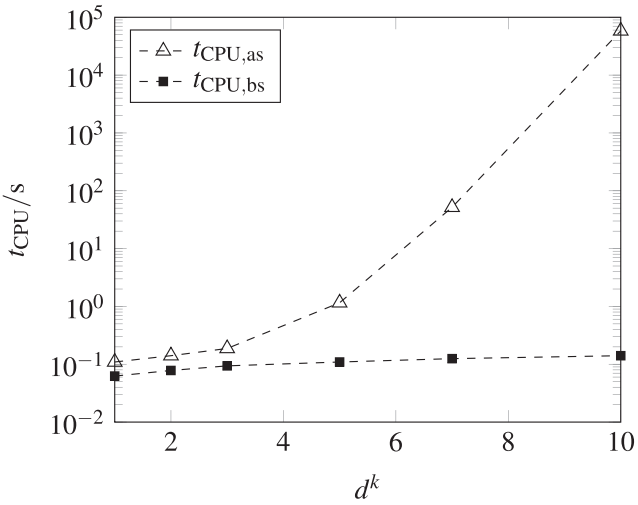


Fig. 9 CPU time for Intel(R) Xeon(R) CPU E5-1660 v4 @ 3.20 GHz to calculate box-shaped (bs) and arbitrarily shaped (as) component solution spaces with analytically calculated volume

Like for the simple crash design problem from Section 3, requirements from crash tests are considered. These are the energy absorption, the maximal acceleration, and the order of deformation of the components. Again, they are formulated with respect to the force–deformation characteristics of the components. Due to technical limitation, the force–deformation characteristics of the kth component are bounded by a lower limit $F_{\mathrm{ds}}^{\mathrm{l},k}$ and an upper limit $F_{\mathrm{ds}}^{\mathrm{u},k}$. The corresponding values are given in Table 1. Compared to Section 3 where local positions of deformation are used as input for the force–deformation characteristics of the components, positions of deformation are globally defined here. The deformation of the vehicle starts at $s_0 = 0\,\mathrm{mm}$ and ends at $s_{\mathrm{end}} = 750\,\mathrm{mm}$. The deformation of the kth starts at $s_0^k \in [s_0, s_{\mathrm{end}})$ and ends at $s_{\mathrm{end}}^k \in (s_0, s_{\mathrm{end}}]$ with $s_0^k < s_{\mathrm{end}}^k$, compare Fig. 4. Moreover, the mass of the vehicle that is active at position $s \in [s_0, s_{\mathrm{end}})$ is denoted by $m^*(s)$, which is the mass in the interval $(s, s_{\mathrm{end}}]$. Every component can be identified by the deformation position $s \in [s_0, s_{\mathrm{end}})$ and the load path $j \in \{1, \ldots, n_{\mathrm{lp}}\}$ which is denoted by $k(j, s)$. Thus, the requirements can be mathematically formulated as follows:

– *Energy absorption*:

$$-\int_{s_0}^{s_{\mathrm{end}}} \frac{1}{m^*(s)} \sum_{j=1}^{n_{\mathrm{lp}}} F^{k(j,s)}(s)\,\mathrm{ds} \leq -\frac{1}{2} v_0^2 \tag{25}$$

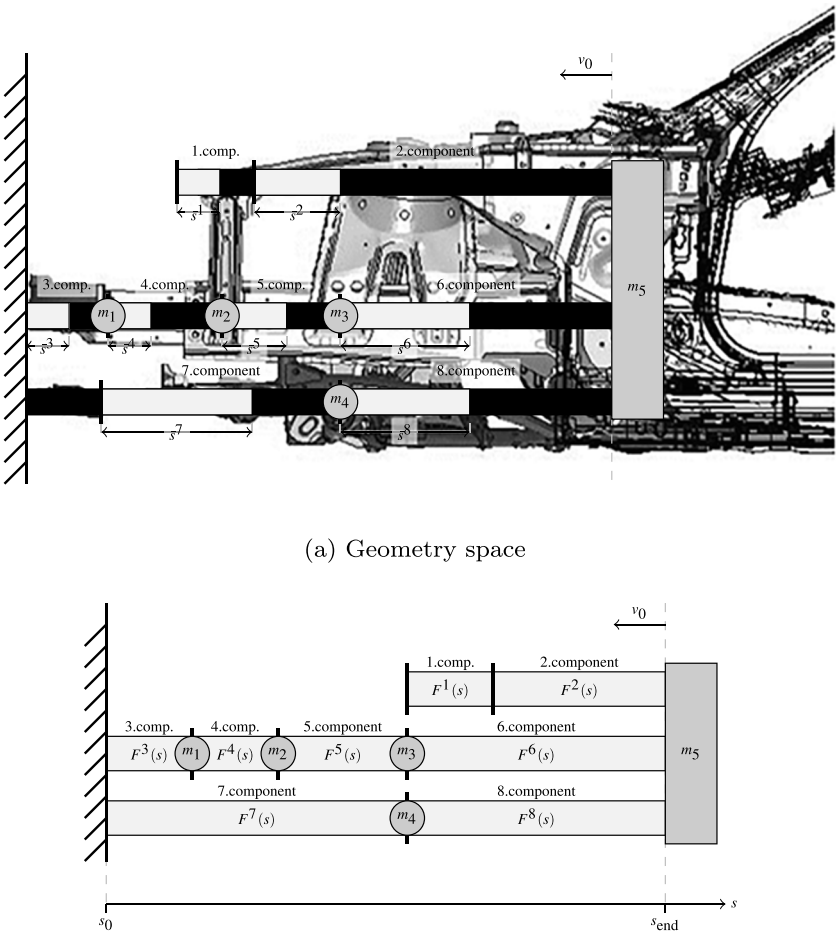
where v_0 is given in Table 2.

Table 1 Values of deformation lengths and bounds of force-deformation characteristics for the realistic crash problem

Component	1	2	3	4	5	6	7	8
Value of $\bar{s}^k$ in mm	100	200	100	100	150	300	350	300
Value of $F_{\mathrm{ds}}^{\mathrm{l},1}$ in kN	0	0	0	0	0	0	0	0
Value of $F_{\mathrm{ds}}^{\mathrm{u},k}$ in kN	200	200	350	350	350	350	150	150

Table 2 Values of masses, initial velocity, and critical acceleration for the realistic crash problem.

Parameter	m_1	m_2	m_3	m_4	m_5	v_0	a_{c}
Value	20 kg	20 kg	200 kg	200 kg	1100 kg	15.6 $\frac{\mathrm{mm}}{\mathrm{ms}}$	0.3 $\frac{\mathrm{mm}}{\mathrm{ms}^2}$



(a) Geometry space

(b) Deformation space (stretched)

Fig. 10 Vehicle front structure for the realistic crash design problem with eight components. The gray parts indicate deformable structure and the black parts non-deformable structure

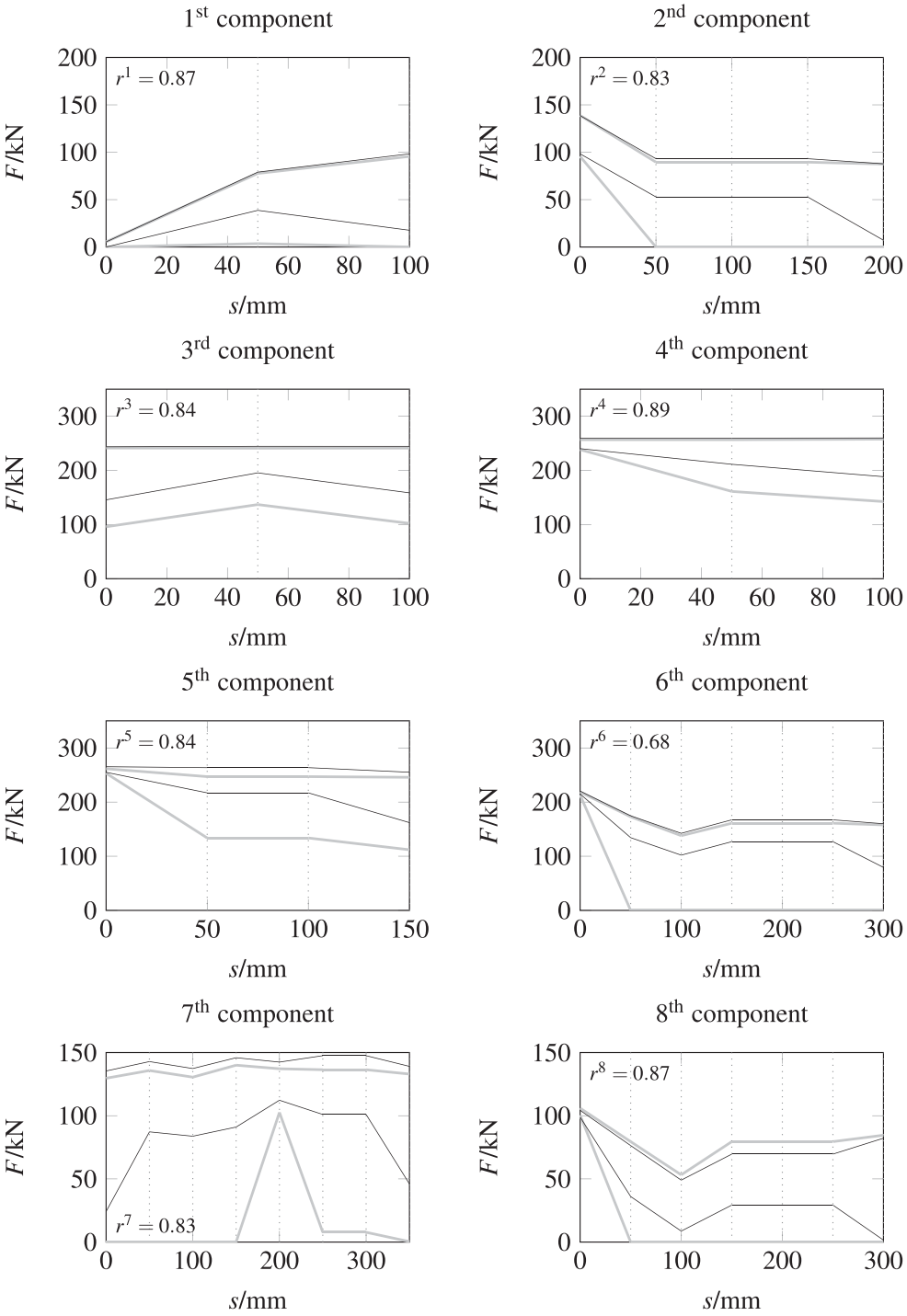


Fig. 11 Regions of possibly permissible characteristics from optimal component solution spaces for optimal arbitrarily shaped component solution spaces (gray) and lower and upper bounds of the optimal box-shaped component solution spaces (black)

– *Maximum acceleration*:

$$\frac{1}{m^*(s)} \sum_{j=1}^{n_{\mathrm{lp}}} F^{k(j,s)}(s) \leq a_{\mathrm{c}} \tag{26}$$

for all $s \in [s_0, s_{\mathrm{end}})$ where a_{c} is given in Table 2.

– *Progressive order of deformation*:

$$F^k(s) - \frac{m^{k,*}(s)}{m^*(s)} \sum_{j=1}^{n_{\mathrm{lp}}} F^{k(j,s)}(s) \leq F^{k+1}(s_0^{k+1}) \tag{27}$$

for all $s \in [s_0^k, s_{\mathrm{end}}^k)$, $k = 1, \ldots, (n-1)$ where the kth and $(k+1)^{\mathrm{th}}$ component share the same load path and

Table 3 Comparison of volume V, average edge length l, and CPU time (Intel(R) Xeon(R) CPU E5-1660 v4 @ 3.20 GHz) for different approaches to compute component solution spaces

Approach	V/kN^{40}	l/kN	CPU time/s
bs	5.06×10^{64}	41.48	1.17
as: initial values	4.37×10^{68}	52.00	1.17
as: Monte Carlo 10^3	4.59×10^{69}	55.15	3.94×10^2
as: Monte Carlo 10^4	3.35×10^{70}	57.96	5.46×10^3
as: Monte Carlo 10^5	7.92×10^{70}	59.22	6.44×10^4
as: Monte Carlo 10^6	1.43×10^{71}	60.10	8.86×10^5
as: exact	4.79×10^{71}	61.95	3.74×10^5

$m^{k,*}(s)$ is the mass in the interval $(s, s^k_{\text{end}}]$ in the load path of the kth component.

The requirements (16)–(19) from Section 3 are special cases of the requirements (25)–(27) where there is only one load path and the active mass m^* does not change.

The values given in Tables 1 and 2 are uncontrollable parameters. The controllable variables, which can be targeted by component designers, are the degrees of freedom of the force–deformation characteristics of the components. Again, piece-wise linear force–deformation characteristics with d^k degrees of freedom are considered which are given by (23) for $k = 1, \ldots, 8$. According to the deformation length of the components, the degrees of freedom of the components, i.e., the number of their design variables are chosen as $d^1 = d^3 = d^4 = 3$, $d^5 = 4$, $d^2 = 5$, $d^6 = d^8 = 7$, and $d^7 = 8$.

Therefore, the design space has 40 dimensions and the requirements on the system performance reduce to a system of linear inequalities. Now the optimal component solution spaces Ω^k, $k = 1, \ldots, 8$, for the design variables of the force–deformation characteristics can be computed like before. The results are visualized by using the lower and upper bounds of box-shaped component solution spaces and the regions of possibly permissible characteristics for arbitrarily shaped component solution spaces (see Fig. 11).

Here, the findings for the simple crash design problem of Section 3 are validated. Again, the volume of arbitrarily shaped component solution spaces is significantly greater than the volume of box-shaped ones, as is their average edge length. Details are stated in Table 3. However, the design variables of any force–deformation characteristic within the region of possibly permissible characteristics need to be tested if they are permissible, meaning to be in Ω^k, $k = 1, \ldots, 8$. In the case of box-shaped component solution spaces, this is not necessary. Furthermore, the computation time for arbitrarily shaped component solution spaces is much greater than the computation time for box-shaped component solution spaces.

To determine the regions of possibly permissible characteristics in Fig. 11, the exact algorithm for the computation of the volume of Ω^k, $k = 1, \ldots, 8$, from Section 3 is used. In Table 3, this is extended by also using the Monto Carlo approach with 10^3, 10^4, 10^5, and 10^6 local sample points per component to compute the volume of each Ω^k, $k = 1, \ldots, 8$. Their computation time is shown together with the volume of $\Omega^1 \times \cdots \times \Omega^8$ and its average edge length. The table is complemented by the results for the initial values for computing arbitrarily shaped component solution spaces from box-shaped solution spaces.

Table 3 demonstrates that arbitrarily shaped component solution spaces computed by using the exact algorithm for volume computation comprise the largest volume V. Although the same optimization problem is solved by using the Monte Carlo method for volume computation, the volume V is smaller. The less sample points to compute the volume of Ω^k, $k = 1, \ldots, 8$, the less computational effort is required to solve problem (11). However, the optimal volume found is also less. If 10^6 sample points are used, the volume gets close to the volume which is obtained by using the exact approach. If 10^3 sample points are used, the volume is still close to the volume for the initial values.

As the initial values for the computation of arbitrarily shaped component solution spaces are deducted directly from box-shaped component solution spaces, their computation time is the same. Furthermore, the volume V for the initial values is approximately 10^4 times greater than the volume for box-shaped solution spaces. In contrast to the other approaches which solve problem (11), the computational effort to obtain component solution spaces for the initial values is very small. Overall, it is always a trade-off between the volume V and the CPU time to decide which approach to choose when computing arbitrarily shaped component solution spaces. In case the volume for box-shaped solution spaces is sufficient, such a decision is not required.

5 Conclusion

In this paper, a new approach for optimal decomposition of system requirements into component requirements is presented, yielding component solution spaces. Hence, a more appropriate and more flexible decoupling-based

development of components is enabled. This decoupling is necessary to break down complexity and to allow for independent design of components. Previous work relied on strong decoupling of all design variables, which often leads to an unnecessary loss of solution space associated with the loss of feasible designs. Hence, this work aims at a better balance between decoupling and flexibility. For this, a new hybrid approach is proposed where engineering knowledge is combined with mathematical procedures such that the decoupling can follow the engineering definition of components, which are motivated by the organization of the different stakeholders contributing to the overall system development.

In contrast to existing box-shaped solution space approaches by Zimmermann and von Hoessle (2013) and Fender et al. (2016), the proposed decomposition scheme for computing component solution spaces does not decouple all considered design variables. By not decoupling design variables of one component, larger solution spaces with potentially more feasible designs, and thus higher design flexibility, are achieved. The originally box-shaped solution spaces for the fine-grain decoupling are now replaced by the Cartesian product of arbitrarily shaped sets as a subset of the complete solution space following the engineering definitions of meaningful components.

The effectiveness of the new approach is demonstrated by small examples and the full potential is illustrated by a complex structural design problem for crashworthiness. The proposed method can be transferred to other problems in systems engineering where a clear decomposition into components is meaningful.

Acknowledgments This work was supported by the SPP 1886 "Polymorphic uncertainty modeling for the numerical design of structures" of the German Research Foundation, DFG.

Compliance with ethical standards

Conflict of interest The authors declare that they have no conflict of interest.

Replication of results All results can be reproduced by using the equations and values presented in this work.

References

Avriel M, Rijckaert MJ, Wilde DJ (1973) Optimization and design: International summer school on the impact of optimization theory on technological design, Katholieke Universiteit te Leuven, 1971. Prentice-Hall, Englewood Cliffs

Bemporad A, Filippi C, Torrisi FD (2004) Inner and outer approximations of polytopes using boxes. Comput Geom 27(2):151–178. https://doi.org/10.1016/S0925-7721(03)00048-8

Beyer H-G, Sendhoff B (2007) Robust optimization – a comprehensive survey. Comput Methods Appl Mech Eng 196(33-34):3190–3218. https://doi.org/10.1016/j.cma.2007.03.003

Büeler B, Enge A, Fukuda K (2000) Exact volume computation for polytopes: a practical study. In: Kalai G, Ziegler GM (eds) Polytopes, DMV Seminar. Birkhäuser, Basel, pp 131–154. https://doi.org/10.1007/978-3-0348-8438-9_6

Daub M (2017) Konvexe Optimierung am Beispiel volumenmaximaler einbeschriebener Rechtecksmengen (Convex optimization at the example of maximum-volume rectangle sets). Master thesis, Universität Konstanz, Germany, http://nbn-resolving.de/urn:nbn:de:bsz:352-0-420923

Erschen S (2018) Optimal decomposition of high-dimensional solution spaces for chassis design. Ph.D. thesis, Technische Universität München, Germany

Evans M, Swartz T (2000) Approximating integrals via Monte Carlo and deterministic methods, Oxford statistical science series, vol 20. Oxford University Press, Oxford

Fender J (2013) Solution spaces for vehicle crash design. Ph.D. thesis, Technische Universität München, Germany

Fender J, Duddeck F, Zimmermann M (2016) Direct computation of solution spaces. Struct Multidiscip Optim 55(5):1787–1796. https://doi.org/10.1007/s00158-016-1615-y

Götz M, Liebscher M, Graf W (2012) Efficient detection of permissible design spaces in an early design stage. Proceedings of the 11th LS-DYNA Forum, Ulm, Germany. https://doi.org/10.13140/2.1.2057.9202

Graf W, Götz M, Kaliske M (2018) Computing permissible design spaces under consideration of functional responses. Adv Eng Softw 117:95–106. https://doi.org/10.1016/j.advengsoft.2017.05.015

Graff L, Harbrecht H, Zimmermann M (2016) On the computation of solution spaces in high dimensions. Struct Multidiscip Optim 54(4):811–829. https://doi.org/10.1007/s00158-016-1454-x

Harwood SM, Barton PI (2017) How to solve a design centering problem. Math Methods Oper Res 86(1):215–254. https://doi.org/10.1007/s00186-017-0591-3

Hendrix EM, Mecking CJ, Hendriks TH (1996) Finding robust solutions for product design problems. Eur J Oper Res 92(1):28–36. https://doi.org/10.1016/0377-2217(95)00082-8

Lange VA, Fender J, Song L, Duddeck F (2019) Early phase modeling of frontal impacts for crashworthiness: from lumped mass–spring models to deformation space models. Proc Inst Mech Eng Part D: J Autom Eng 233(12):3000–3015. https://doi.org/10.1177/0954407018814034

Lasserre JB (1983) An analytical expression and an algorithm for the volume of a convex polyhedron inr n. J Optim Theory Appl 39(3):363–377. https://doi.org/10.1007/BF00934543

Milanese M, Norton J, Piet-Lahanier H, Walter E. (eds) (1996) Bounding approaches for system identification. Springer Science+Business Media, New York. https://doi.org/10.1007/978-1-4757-9545-5

Parkinson A, Sorensen C, Pourhassan N (1993) A general approach for robust optimal design. J Mech Des 115(1):74. https://doi.org/10.1115/1.2919328

Zimmermann M, von Hoessle JE (2013) Computing solution spaces for robust design. Int J Numer Methods Eng 94(3):290–307. https://doi.org/10.1002/nme.4450

Zimmermann M, Königs S., Niemeyer C, Fender J, Zeherbauer C, Vitale R, Wahle M (2017) On the design of large systems subject to uncertainty. J Eng Des 28(4):233–254. https://doi.org/10.1080/09544828.2017.1303664