NordDesign 2020
August 11-14, 2020
Kgs. Lyngby, Denmark

On the Treatment of Equality Constraints in Mechanical Systems Design Subject to Uncertainty

Julian Stumpf[1,2], Thomas Naumann[2], Marc Eric Vogt[1,3], Fabian Duddeck[1], Markus Zimmermann[1]

[1]Technical University of Munich, Germany
[2]Mercedes-Benz AG, Stuttgart, Germany
[3]BMW Group, Munich, Germany
julian.stumpf@tum.de
thomas.t.naumann@daimler.com
marc-eric.vogt@bmw.de
duddeck@tum.de
zimmermann@tum.de

Abstract

In the early phase of product development, uncertainty may be treated by a set-based design method relying on so-called solution spaces. In this approach, all relevant system requirements are used to generate interval-type component requirements that are sufficient for reaching the overall design goal while allowing for design flexibility in a distributed component development process. The product of these interval-type requirements on component level is defined as solution box. The aim is not to find an optimal design, but to maximize the solution box that guarantees compliance with the system requirements. Unfortunately, with this approach only inequality-type requirements on the system level can be treated. However, in mechanical systems design it is often necessary to consider equality-type requirements, for example when mechanisms need to satisfy certain kinematic specifications. This paper proposes several approaches to incorporate equality-based requirements into the framework that calculates solution spaces. First, a relaxed problem statement is presented that can be solved by existing algorithms to maximize solution boxes. Furthermore, design variables are split into two groups: early-decision variables that are subject to large uncertainty and late-decision variables that are controllable and adjustable in a later stage of the development process. An existing projection technique and a novel approach that both project solution spaces to increase permissible intervals for early-decision variables are introduced. The relaxed problem statement and an inversion-based projection technique are applied to the design of a tailgate system of a passenger car, where the chassis and tailgate are designed in a distributed development process. It is shown that the resulting permissible intervals on the component level are significantly wider when projecting the solution space onto a subset of design variables than by solving the relaxed problem statement without applying a projection operator.

Keywords: computational design methods, solution space, uncertainty, robust design, early design phase

1 Introduction

Complex systems development is subject to uncertainties. With an increasing number of functional interrelationships between design variables and requirements to be met, complexity grows and the entire system becomes uncertain during the development process (Suh, 2005). In order to minimize the number of iterations in the development process and to deal with uncertainties, set-based design approaches, in which permissible intervals are sought for each design variable, have a large potential in comparison to classic point-based design methods, which look for one optimal value for each design variable (Qureshi et al., 2014). Usually set-based design approaches pursue the aim of finding large sets of good solutions in a given design space. Therefore, it is necessary to explore the design space in an efficient way. In the development of software architecture, this is a well-known procedure prior to the implementation to find design alternatives that satisfy all global constraints (Palesi and Givargis, 2002 and Kang et al., 2011).

For design space exploration in an early stage of product development under uncertainties Yannou & Harmel (2004) present a Constraint Programming (CP) method, in which a design space, described by several multi-dimensional boxes is calculated. These boxes enclose a set of good designs that fulfill every requirement on the system but contain bad designs as well. Therefore, a further step to calculate a solution is needed and design variables are not decoupled. Decoupling means that each design variable is assigned a valid solution interval in which its value can be modified independently of all other design variables.

Further research on Generative Design Approach (GDA) is presented in Li and Lachmayer (2019). The idea of this method is to transform the given design problem into a configuration problem. In an iterative process, the design space can be efficiently explored by configuring variable design elements. Thus, GDA helps the designer to find larger regions of feasible designs, compared to conventional modelling methods. However, GDA is only applied on designing single parts and does not consider complexity in the system development processes. Abi Akle et al. (2017) focus on the visualization of the design space and compare different possibilities of representing systems behavior. To get a deeper understanding, of how requirements on a system and solution spaces can be related, Salado et al. (2017) present a comprehensive summery of dependencies and mathematical formulations.

In this paper, the focus stays on the so-called Solution Space Engineering, a set-based design method, proposed by Zimmermann and Von Hössle (2013), in which a set of good designs that meet all considered requirements is defined as a solution space. The aim of this method is to find one large multi-dimensional box-shaped solution space in order to achieve a decoupling of the design variables and thus maximum flexibility in design. A box-shaped solution space can be represented as the product of permissible intervals. Each interval may be treated as a component design goal and since the intervals are independent, this is a huge advantage in a distributed component design (Zimmermann and Von Hössle, 2013). Unfortunately, complete decoupling of design variables is in conflict with design feasibility, because the permissible intervals are often not sufficiently large. The following three approaches to increase the resulting permissible intervals are found in literature.

Erschen et al. (2017) propose an approach that optimizes a set of permissible two-dimensional regions (2D-spaces) for pairs of design variables, represented by polygons. The complete solution space is the Cartesian product of all 2D-spaces. Another approach, in which design variables are pair-wisely coupled to increase a box-shaped solution space, is presented by

Harbrecht et al. (2019). Vogt et al. (2018a) propose a method, where design variables are split into two groups in order to reduce the dimensions of the overlaying optimization problem and calculate solution-compensation spaces: early-decision variables have a strong influence on the systems performance and need to be associated with permissible intervals on which they may assume any value; whereas late-decision variables can be adjusted in a later stage of the development process and thus they are associated with intervals on which they have to be able to assume any value. Consequently, this leads to larger permissible intervals for early-decision variables due to additional conditions for late-decision variables (Vogt et al., 2018a).

However, none of the mentioned approaches offers the treatment of design problems with equality constraints. Nevertheless, the design of mechanical systems often deals with geometrical and functional requirements simultaneously. Strong geometrical requirements, e.g. a mechanism that must reach a given coordinate point on its movement, often lead to equality constraints in a mathematical form. This study aims to extend the classical problem formulation of Solution Space Engineering in order to consider equality constraints. Furthermore, it discusses different opportunities to solve this extended problem.

2 Solution Spaces

To describe the behavior of the considered system, the performance function $f: \mathbb{R}^n \to \mathbb{R}^m$ is defined as

$$z = f(x), \tag{2.1}$$

where $x = [x_1, x_2, \dots, x_n] \in \Omega_{\mathrm{ds}}$ is a single *design point* and $z = [z_1, z_2, \dots, z_m] \in \mathbb{R}^{\mathrm{m}}$ the response of the system behavior. $\Omega_{\mathrm{ds}} \subset \mathbb{R}^n$ is called *design space*.

Conventional optimization problems have a formulation that the performance function f, or at least a subset of it, is defined as objective function. Marler and Arora (2004) show a comprehensive collection of approaches to find solutions of such multi-objective optimization problems. In an early stage of the development process, however, the definition of a set of interdisciplinary objectives may be very subjective, which is why an exploration of a solution space, containing only feasible design points, may be preferred according to Yannou et al. (2009). Therefore, several methods exist to deduce an optimization problem by defining the performance of a system as constraints only, in order to maximize the robustness of a design. Hendrix et al. (1996) defines a couple of robustness measurements and presents an optimization formulation and the solution of this problem statement for linear constraints. Nevertheless, Hendrix´ formulation is point-based and the decoupling of variables is not guaranteed. In the following, the optimization problem referring to Solution Space Engineering is introduced in detail. In this approach, the performance function f is defined as constraints and the aim is to maximize the box-shaped solution space.

The requirements on the system are defined as inequality constraints

$$f(x) \le f_{\mathrm{c}}, \tag{2.2}$$

with $f_{\mathrm{c}} \in \mathbb{R}^m$ as the given threshold values. A design point x is called a *good design*, if the constraints (2.2) are fulfilled. Zimmermann and Von Hössle (2013) define the *complete solution space* as the set of all good designs. To decouple the design variables, a box-shaped solution space (*solution box*) is defined as

$$\Omega = I_1 \times I_2 \times \dots \times I_n \subset \Omega_{\mathrm{ds}}, \tag{2.3}$$

where $I_i = [x_i^{\mathrm{l}}, x_i^{\mathrm{u}}]$ is a permissible interval of the i-th design variable with the lower and the upper boundary x_i^{l} and x_i^{u} respectively.
Figure 1 shows an example design space with two design variables x_i and x_j and three requirements. The complete solution space is colored green. Designs in the yellow, red and blue areas do not fulfill the requirements. As a subset of the complete solution space, a solution box is drawn in dark green.

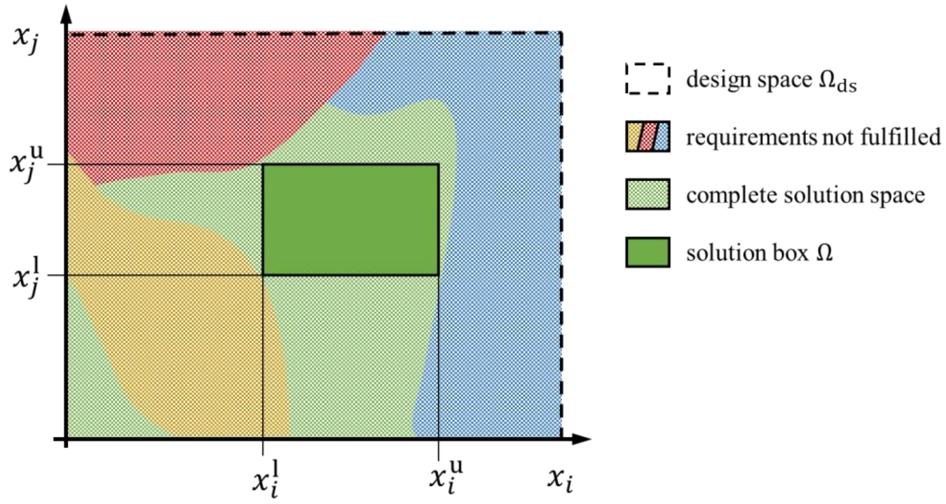

Figure 1: Design space, solution space and solution box

To maximize the size $\mu(\Omega) \in \mathbb{R}$ of the solution box Ω, Zimmermann and Von Hössle (2013) define the optimization problem

$$\max_{\Omega \subset \Omega_{\mathrm{ds}}} \mu(\Omega) \tag{2.4}$$
$$s.t. \quad \forall x \in \Omega, \qquad f(x) \leq f_{\mathrm{c}}.$$

As a possible size measure $\mu(\Omega)$ of the solution box Ω, Zimmermann and Von Hössle (2013) propose the volume

$$\mu(\Omega) = \prod_{i=1}^{n} (x_i^{\mathrm{u}} - x_i^{\mathrm{l}}). \tag{2.5}$$

3 Problem Formulation

In the previous paragraph, the Solution Space Engineering approach was presented in detail. This section demonstrates a simple example problem, where this approach cannot be applied, because the shape of the complete solution space does not allow a maximization of the solution box. Subsequently, the resulting general problem statement is defined.

3.1 Simple Example Problem

Figure 2a shows a mechanism with one degree of freedom. In A, a rigid body is mounted pivotably. It can be turned about the angle φ. Between B and C a gas spring is attached in order to ensure that the rigid body stays safe in both positions, closed and open. The maximum angle $\varphi_{\max}$ depends on the distance Δx between A and B as well as the length l of the gas spring when completely extended.

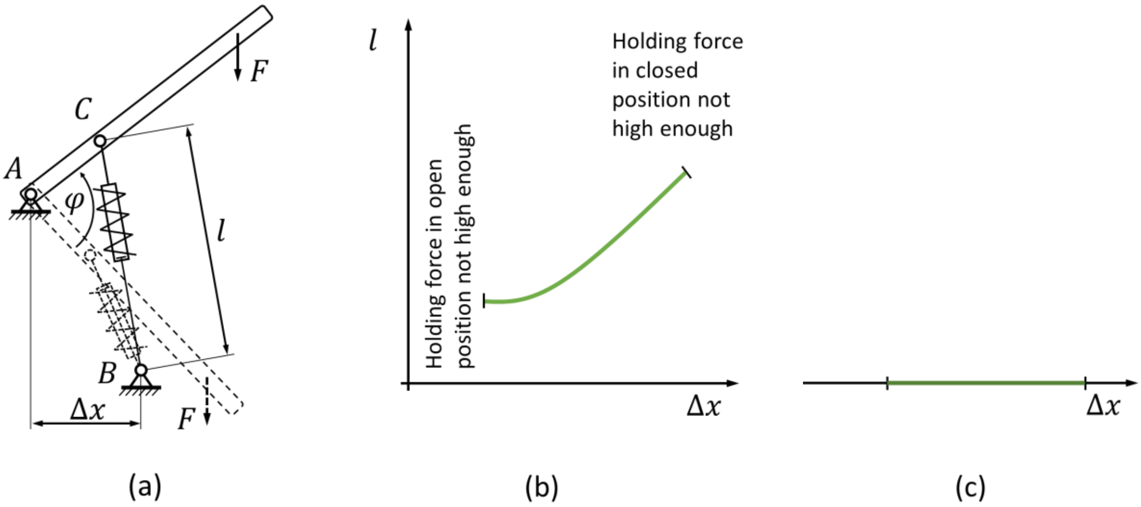

Figure 2: Simple example problem with two design variables: a) Mechanism resembling a vehicle tailgate b) Complete solution space c) Projected solution space

The requirements of the system are to fulfil an equality restriction on $\varphi_{\max}$ and a set of inequality constraints in order to ensure that the resulting forces are within a permissible range. The solution space of this design problem is a thin and curved line with the volume equal to zero, shown in Figure 2b. How can an efficient projection of this solution space onto a subset of design variables, as depicted in Figure 2c, be achieved? To discuss this question, a general problem statement is pointed out in the following.

3.2 General Problem Statement

In order to take equality constraints into account, the optimization problem (2.4) is extended and becomes

$$\max_{\Omega\subset\Omega_{\mathrm{ds}}} \mu(\Omega) \tag{3.1}$$
$$s.t. \quad \forall x \in \Omega, \qquad f(x) \leq f_{\mathrm{c}}, \qquad h(x) = h_0,$$

with the threshold values $f_{\mathrm{c}} \in \mathbb{R}^m$ and the nominal values $h_0 \in \mathbb{R}^s$. The performance functions are defined as $f: \mathbb{R}^n \to \mathbb{R}^m$ and $h: \mathbb{R}^n \to \mathbb{R}^s$ subject to inequality and equality constraints respectively. This problem formulation leads to a solution space, whose volume is equal to zero, if at least one equality constraint (depending on all design variables) is defined (see example in Figure 2b). In this case, the optimization problem is not well defined, because the size of the maximum solution box is equal to zero at any point of the solution space. To resolve this problem, different formulations are discussed in the following.

4 Solution Approaches

Conventional algorithms of Solution Space Engineering are based on Monte Carlo sampling techniques (Zimmermann et al., 2017). These methods fail when solving the optimization problem (3.1), because the volume of the solution space is equal to zero and a good design point will not be found by random sampling.

4.1 Relaxed Problem Statement

One possible modification to achieve a feasible form of the optimization problem (3.1) is to define a small permissible range $\kappa \in \mathbb{R}^s_+$ around the nominal values h_0 of the equality constraints, so that the relaxed constraints of the system reads

$$\tilde{f}(x) = \begin{pmatrix} f(x) \\ h(x) \\ -h(x) \end{pmatrix} \leq \begin{pmatrix} f_\mathrm{c} \\ h_0 + \kappa \\ -h_0 + \kappa \end{pmatrix} = \tilde{f}_\mathrm{c}, \tag{4.1}$$

with the performance function $\tilde{f}: \mathbb{R}^n \rightarrow \mathbb{R}^{m+2s}$ and the relaxed threshold values $\tilde{f}_\mathrm{c} \in \mathbb{R}^{m+2s}$. Sampling-based Solution Space Engineering algorithms can achieve the maximization of the solution box according to the optimization problem (2.4) with the constraints defined in equation (4.1). However, the resulting solution space is not well suited to decouple the design variables, because it is still thin and thus the maximum solution box is often too small.

4.2 Solution-Compensation Spaces

To increase the solution intervals for crucial design variables of the relaxed problem statement Vogt et al. (2018a) define an optimization problem to compute so-called solution-compensation spaces as

$$\max_{\Omega_a \subset \Omega_{\mathrm{ds},a}} \mu(\Omega_a) \tag{4.2}$$

$$s.t. \quad \forall x_a \in \Omega_a, \qquad \exists x_b \in \Omega_{\mathrm{ds},b}, \qquad f(x_a, x_b) \leq f_\mathrm{c},$$

where a design point $x = [x_a, x_b]$ is represented by early- and late-decision variables $x_a \in \Omega_{\mathrm{ds},a} \subset \mathbb{R}^p$ and $x_b \in \Omega_{\mathrm{ds},b} \subset \mathbb{R}^q$ respectively. Early-decision variables are design variables that are subject to large uncertainty. They need to be associated with large permissible intervals on which they may assume any value in order to increase flexibility for component design. In contrast, late-decision variables are design variables that are adjustable in a later stage of the development process. The solution of the optimization problem (4.2) leads to a projection of the complete solution space onto the early-decision variables with larger permissible ranges. Solution-compensation spaces are the Cartesian product of early-decision variables and late-decision variables (Vogt et al., 2018a).

To solve the optimization problem (4.2), sampling-based algorithms to maximize solution boxes can be used, whereby each evaluation of a sample point x_a requires solving a further optimization problem that reads

$$z_\mathrm{opt} = \min_{x_b \in \Omega_{\mathrm{ds},b}} f(x_a, x_b). \tag{4.3}$$

Thus, the number of performance function evaluations increases significantly and calculation costs are often not reasonable. An efficient algorithm to calculate solution-compensation spaces for linear problems is presented by Vogt et al. (2018b). However, the development of efficient methods to compute solution-compensation spaces for general problems is part of future research.

4.3 Equality Constrained Projection

The relaxed problem statement does not ensure that a design meets the equality constraints exactly. This paper presents a formulation, in which a direct projection of the solution space onto the early-decision variables using the equality constraints is achieved. Therefore, the optimization problem (4.2) to calculate solution-compensation spaces is extended to consider equality constraints and becomes

$$\max_{\Omega_a \subset \Omega_{\mathrm{ds},a}} \mu(\Omega_a) \tag{4.4}$$

$$s.t. \quad \forall x_a \in \Omega_a, \qquad \exists x_b \in \Omega_{\mathrm{ds},b}, \qquad f(x_a, x_b) \le f_{\mathrm{c}}, \qquad h(x_a, x_b) = h_0.$$

The solution of the optimization problem (4.4) can be found by a sampling-based algorithm to maximize solution boxes. When evaluating a sample point x_a, the following steps must be taken:

1) Find $x_b \in \Omega_{\mathrm{ds},b}$, so that the equality constraints

   $$h(x_a, x_b) = h_0 \tag{4.5}$$

   are fulfilled.

2) Evaluate the inequality performance function:

   $$z = f(x_a, x_b). \tag{4.6}$$

3) Check if the inequality constraints are fulfilled:

   $$\mathit{if}\ \mathrm{z} \le f_{\mathrm{c}} \implies x_a \in \mathit{good\ designs}. \tag{4.7}$$

To clarify the idea of the approach described by the equations (4.5)-(4.7), Figure 3 illustrates the evaluation of a sample point with and without projection in the form of a dependency graph, with z and z_{eq} as the responses of the performance functions $f(x)$ and $h(x)$ respectively.

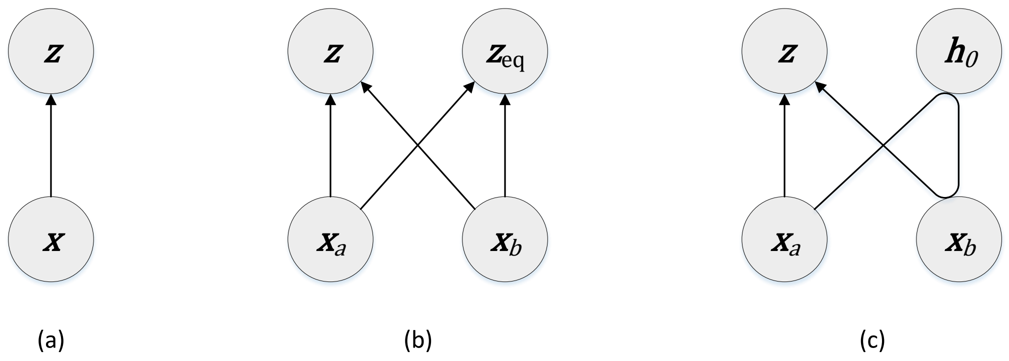

Figure 3: Evaluation of a sample point: a) Conventional Solution Space Engineering formulation b) Formulation with early- and late-decision variables as well as inequality and equality constraints c) Equality constrained projection

The conventional formulation (Figure 3a) refers to the optimization problem (2.4), where a single set of inequality constraints is given. The differentiation between early- and late-decision variables as well as equality and inequality constraints are shown in Figure 3b. This formulation relates to the constraints of the optimization problem (4.4). The calculation of the projected solution space by the equality constraints (including the nominal values h_0) is shown in Figure 3c. The corresponding calculation procedure is characterized by the equations (4.5)-(4.7).

The projection of the complete solution space by the equality constraints onto early-design variables requires the solution of the system of nonlinear equations (4.5) for every evaluation of a sample point x_a. In comparison to the calculation of solution-compensation spaces (solving optimization problem (4.3)), this might be numerically less expensive, because the number of equations is less. However, there are several prerequisites on equation (4.5) to ensure a numerically stable solution. Additionally, the numerical effort is difficult to estimate and might be too expensive considering that this system has to be solved for each sample point evaluation. Therefore, the definition of efficient methods and algorithms is part of future research. However, in certain cases it is possible to build the inverse of the performance function referring to the equality constraints, which is defined as

$$x_b = g(x_a, h_0). \tag{4.8}$$

In this case, the first step of the presented approach above (equation (4.5)) is numerically inexpensive and easy to calculate.

5 Application to the Design of a Tailgate System

In order to compare the methods presented, the design of a tailgate system of a passenger car is considered in the following.

5.1 Problem Description

In Figure 4 the tailgate system is shown in closed position. Part of the design problem is the chassis in grey, the trunk lid in yellow and the gas spring (gs) in blue color. The attachment points of the gas spring are colored red. The design of the chassis and the tailgate takes place in the early phase of vehicle development. Therefore, it is necessary to know the permissible area for the attachment points of the gas spring. The example shows how these permissible ranges for the attachment points can be calculated, to ensure that all requirements on the tailgate system level are met.

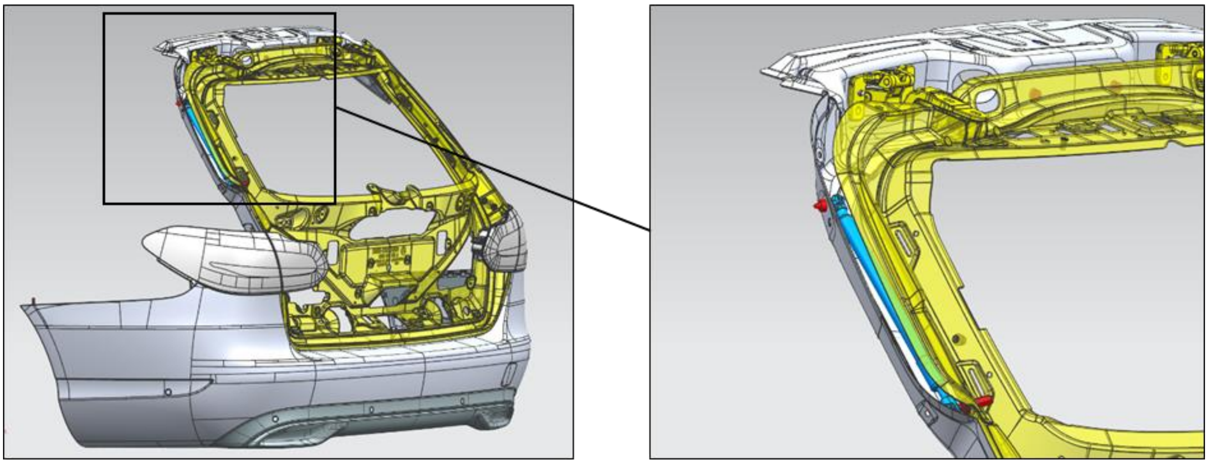

Figure 4: Tailgate system of a passenger car

The dependency graph of the tailgate system is shown in figure 5. The critical design variables are the coordinates in vehicle x and z direction of the attachment points of the gas spring on the chassis and on the tailgate. Those variables have a strong influence onto the functional requirements of the trunk lid, e.g. the holding forces and the closing behavior of the tailgate system and depend on the component design of the chassis and the tailgate.

Since the design of the components takes place in a distributed development environment it is a huge advantage to get the largest permissible intervals (solution box) for the critical design variables, which fulfill all functional requirements, to increase the flexibility of the component design for the chassis and the tailgate. For this reason, the attachment points are allocated to the early-decision variables. The parameters of the gas spring are the nominal force $F_{n,gs}$ and the length $l_{max,gs}$ (length when gas spring is fully extended). These design variables are not very critical, because their value can be chosen in a later stage of the development process. Thus, it is not necessary to maximize the solution box in these dimensions and they are assigned to the late-decision variables.

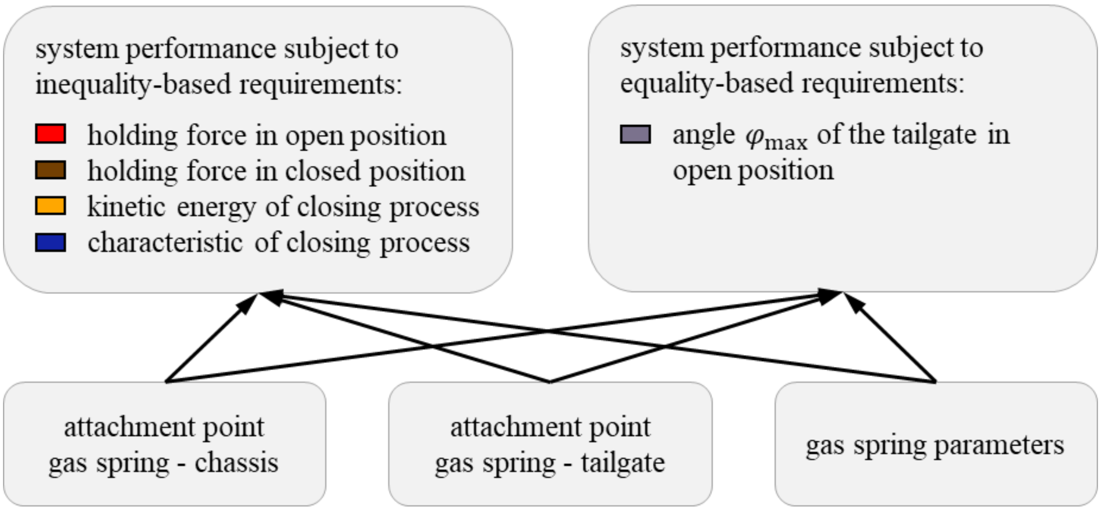

Figure 5: Dependency graph of the tailgate system with the system performance on top and the design variables at the bottom

Since the given design problem involves the equality constraint on the tailgate angle φ_{max} in open position, it is not possible to solve it with the conventional Solution Space Engineering approach. Therefore, we derive and solve the relaxed problem statement as presented in section 4.1 and apply the projection by the equality constraint, introduced in section 4.3. The model to evaluate the performance functions is implemented as a multi-body simulation.

5.2 Results of the Relaxed Problem Statement

Figure 6 shows the results, calculated by solving the optimization problem (2.4) with the relaxed constrains (4.1). The relaxation constant is chosen to $\kappa = 1°$ for the tailgate angle φ_{max} in open position. The green area is the complete solution space. The other colors indicate the constraints, relating to the colors of the system performance in the dependency graph (Figure 5). As a result, the complete solution space of this problem is a thin layer and the size of the maximum solution box to decouple the design variables is not sufficiently large for the distributed development process of the chassis and the tailgate.

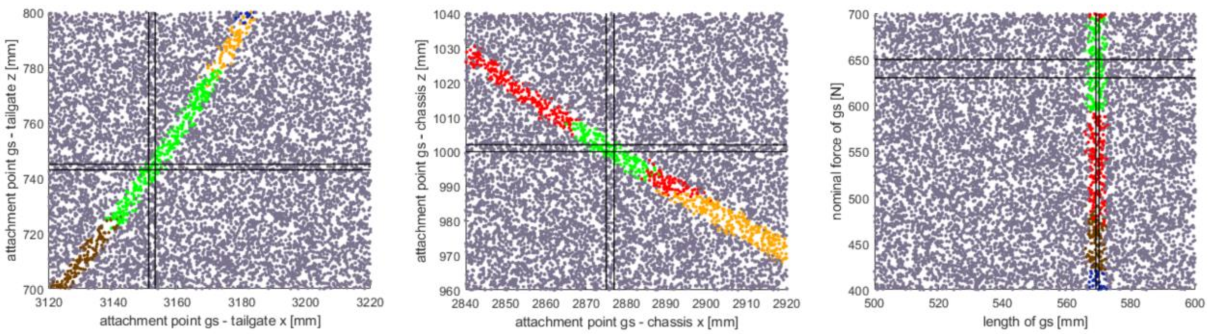

Figure 6: Solution space and solution box of the relaxed problem statement (colors of requirements refer to Figure 5)

5.3 Results with Equality Constrained Projection

The results depicted in Figure 7 are the solution of the optimization problem (4.4). As well as in Figure 6, the green area is the complete solution space and the colors of the constraints are related to the system performance, shown in the dependency graph (Figure 5). Although the equality constraint subject to $\varphi_{\max}$ (grey) is used as the projection operator, there are areas where this requirement is not fulfilled (see Figure 7). This happens, when the projection leads to an invalid value of the late-decision variable $l_{\max,\mathrm{gs}}$.

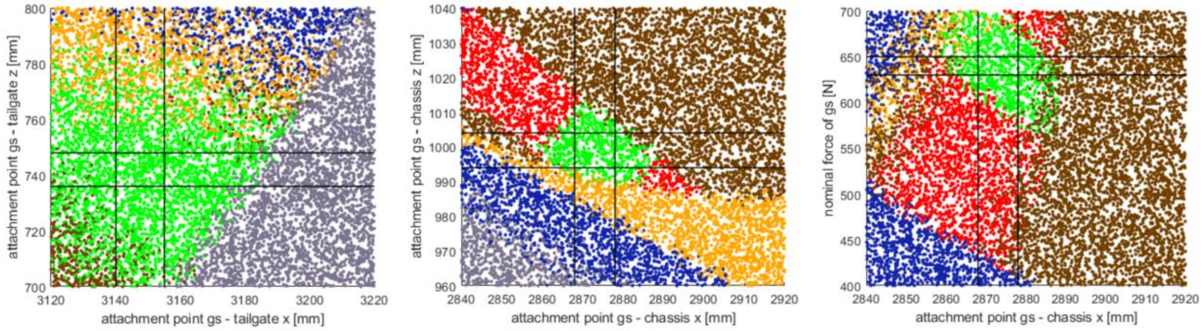

Figure 7: Solution space and solution box with projected equality constraint $\varphi_{\max}$ (colors of requirements refer to Figure 5)

The projection by the equality constraint $\varphi_{\max}$ is inversion-based and thus easy and fast to calculate. The design space is reduced by the late-decision variable length of the gas spring $l_{\max,\mathrm{gs}}$. This leads to a significant wider solution space and solution box with enough flexibility for the component development.

5.4 Comparison of the Solution Boxes

In Table 1, the calculated solution boxes respectively the widths of the permissible intervals of each design variable are shown. When projecting the solution space by the equality constraint, the widths of the intervals are up to seven times of the solution without projection.

Table 1: Width of the permissible intervals of the design variables

design variable	unit	solution intervals of relaxed problem statement	solution intervals with equality constrained projection
attachment point gs-tailgate x	mm	2	15
attachment point gs-tailgate z	mm	2	12
attachment point gs-chassis x	mm	2	10
attachment point gs-chassis z	mm	2	10
nominal force of gs	N	20	20
length of gs	mm	1	-

6 Conclusion

This paper shows two extensions of the Solution Space Engineering approach, which was introduced by Zimmermann and Von Hössle (2013), to enable the integration of equality constraints. First, a relaxed problem statement was presented, where a relaxation constant is defined and each equality constraint is replaced by two inequality constraints. The resulting solution space is very thin and the maximum solution box is often not sufficiently large. One possibility to obtain a wider solution box is the projection of the complete solution space onto a subset of design variables, called early-decision variables, by computing solution-compensation spaces. Since this is numerically expensive, the idea of the second approach is to project the solution space by the equality constraints onto the early-decision variables. In certain cases it is possible to define this projection inversion-based in a closed form, thus it is numerically favorable.

Both approaches were applied on the design of a tailgate system with six design variables, four inequality-based requirements and one equality-based requirement on the system level. The results of this example show that the first approach, without any projection, leads to very small solution boxes. Additionally, since there is a relaxation constant defined, the equality-based requirement is never exactly fulfilled. The projection of the solution space by the equality constraint leads to much larger permissible intervals for the early-decision variables. In the distributed development process of the chassis and the tailgate, this leads to more flexibility in component design and less communication effort.

In this case, the projection was performed by an inversion-based closed equation, which is easy to evaluate. Unfortunately, it is not possible to define such a numerically inexpensive projection in general. Therefore, future work will focus on the development of efficient formulations to project a solution space with equality-based requirements onto early-design variables.

References

Abi Akle, A., Yannou, B. & Minel, S. (2018). Design space visualization for efficiency in knowledge discovery leading to an informed decision. Proceedings of the 21st International Conference on Engineering Design (ICED 17), Vol 6, pp. 61-70.

Erschen, S., Duddeck, F., Gerdts, M. & Zimmermann, M. (2017). On the Optimal Decomposition of High-Dimensional Solution Spaces of Complex Systems. ASME Journal of Risk and Uncertain in Engineering Systems, Part B: Mechanical Engineering, Vol. 4 No. 2: 021008.

Harbrecht, H., Tröndle, D. & Zimmermann, M. (2019). A sampling-based optimization algorithm for solution spaces with pair-wise-coupled design variables. Structural and Multidisciplinary Optimization, Vol. 60 No. 2, pp. 501-512.

Hendrix, E.M.T., Mecking, C.J., & Hendriks, T.H.B. (1996). Finding robust solutions for product design problems. European Journal of Operational Research, Vol. 92 No. 1, pp. 28-36.

Kang, E., Jackson, E. & Schulte, W. (2011). An Approach for Effective Design Space Exploration. Foundations of Computer Software. Modeling, Development, and Verification of Adaptive Systems, pp. 33-54.

Li, H. & Lachmayer, R. (2019). Automated Exploration of Design Solution Space Applying the Generative Design Approach. Proceedings of the 22nd International Conference on Engineering Design (ICED19), pp. 1085-1094.

Marler, R. & Arora, J. (2004). Survey of multi-objective optimization methods for engineering. Structural and Multidisciplinary Optimization, Vol. 26, pp. 369-395.

Palesi, M. & Givargis, T. (2002). Multi-objective design space exploration using genetic algorithms. Proceedings of the tenth international symposium on hardware/software codesign (CODES '02), pp. 67–72.

Qureshi, A. J., Dantan, J.-Y., Bruyère, J. & Bigot, R. (2014). Set-based design of mechanical systems with design robustness integrated. International Journal of Product Development, Vol. 19 No. 1/2/3, pp. 64-89.

Suh, N. P. (2005). Complexity in Engineering. CIRP Annals - Manufacturing Technology, Vol. 54 No. 2, pp. 46-63.

Salado, A., Nilchiani, R. & Verma, D. (2017). A contribution to the scientific foundations of systems engineering: Solution spaces and requirements. Journal of Systems Science and Systems Engineering, Vol. 26, pp. 549-589.

Vogt, M. E., Duddeck, F., Wahle, M. & Zimmermann, M. (2018a). Optimizing tolerance to uncertainty in systems design with early- and late-decision variables. IMA Journal of Management Mathematics, Vol. 30 No. 3, pp. 269-280.

Vogt, M. E., Duddeck, F., Harbrecht, H., Stutz, F., Wahle, M. & Zimmermann, M. (2018b). Computing solution-compensation spaces using an enhanced Fourier-Motzkin algorithm. Proceedings in Applied Mathematics and Mechanics, Vol. 18 No. 1.

Yannou, B. & Harmel, G. (2004). A Comparative Study of Constraint Programming Techniques Over Intervals in Preliminary Design. Proceedings of the ASME 2004 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Vol. 1: pp. 189-198.

Yannou, B., Troussier, N., Chateauneuf, A., Boudaoud, N. & Scaravetti D., (2009). Dimensioning a product in preliminary design through different exploration techniques. International Journal of Product Development (IJPD), Vol. 9 No. 1/2/3, pp. 140-163.

Zimmermann, M. & Von Hoessle, J. E. (2013). Computing solution spaces for robust design. International Journal for Numerical Methods in Engineering, Vol. 94 No. 3, pp. 290-307.

Zimmermann, M., Königs, S., Niemeyer, C., Fender J., Zeherbauer, C., et al. (2017). On the design of large systems subject to uncertainty. Journal of Engineering Design, Vol. 28, pp. 233–254.