25th INTERNATIONAL DEPENDENCY AND STRUCTURE MODELING CONFERENCE, DSM 2023
GOTHENBURG, SWEDEN, 03 – 05 October 2023

Flexibility Despite Dependencies And Constraints: Product Family Design With Solution-Compensation Spaces

Nicola Barthelmes[1], Stefan Sicklinger[2], Markus Zimmermann[1].

[1]Laboratory for Product Development and Lightweight Design,
TUM School of Engineering and Design, Technical University of Munich
[2]CARIAD SE.

Abstract: Optimization of components and commonality in a product family can be prohibitively expensive due to combinatorics. In a previously presented approach, numerical effort is significantly reduced by computing permissible intervals for design variables of each product variant. Components can be shared where intervals overlap. However, these intervals tend to be small and have little overlap. This paper introduces solution-compensation spaces into product family design. Permissible intervals are determined only for so-called early-decision variables that are relevant for commonality within the product family. All other variables are treated as so-called late-decision variables. They are adjusted for each component once a commonality pattern is established. They compensate for the choice of early-decision variables. This increases interval size and ensures greater flexibility to simultaneously design the components of the product family, while the dependency between the design variables is fully considered. The approach is applied to sensor design and positioning for automated vehicles.

Keywords: Product Family Design, Solution Spaces, Complexity Management

1 Introduction

Automated vehicles use perception sensors to detect relevant objects in their surrounding (Jackson et al., 2021; Maurer et al., 2016). In numerous papers, the performance of perception sensors is analysed to compute challenging test scenarios (Gogri et al., 2020; Ponn et al., 2020), which allows the conclusion that a robust choice of characteristics and positions of perception sensors is safety critical. For a series production of automated vehicles, the costs of state-of-the art perception sensors play an increasingly larger role (Broggi et al., 2013; Khatab et al., 2022; Lambert et al., 2020). To decrease overall costs and minimize the introduction of model-specific faults, an entire product family of automated vehicles is considered. Using identical sensors, i.e., with identical properties and from the same manufacturer for different product variants, comes with an additional safety benefit. Rare hardware or integration faults of this sensor model (Bock et al., 2018; Koopman and Wagner, 2016) are more likely to be detected if it is used more often and subsequently the error can be remedied once for the entire fleet.

A product family consists of several vehicle variants, which can have smaller differences, e.g., different suspension systems, but it may also include models where shape and dynamic behaviour of the vehicles vary considerably. The design of a perception sensor within this paper consists of its characteristics, e.g., the opening angle or resolution of a camera, and its positioning. The term “positioning” is used in this paper to describe the sensor mounting location and its orientation on the surface of the vehicle. Several perception sensors and several vehicle variants lead to a large number of different combinations in the product family optimization. While evaluating the cost of the product family might be comparatively cheap (Ehrlenspiel et al., 2020), simulating different traffic scenarios, combined with various environmental conditions, leads overall to a high-dimensional parameter space that has to be explored to find the optimal performance of the perception system. A method based on solution-spaces was introduced to reduce the possible number of combinations (Eichstetter et al., 2015).

Permissible intervals for the different design variables, where requirements are fulfilled, are computed for each variant individually. Independent intervals for the different design variables reduce the combinatorics of the product family design problem. However, instead of finding one globally optimal solution, the solution-space approach is developed to determine requirements for different components (permissible intervals for the design variables) from system-level requirements. By maximizing the volume of a box of good designs, the different components can be optimized independently (Zimmermann and von Hoessle, 2013), e.g., in a distributed development process. The decoupling, however, comes at the price of neglecting the good designs outside of the box. For product family optimization, components can be shared where the intervals overlap. This means that the flexibility to share components may be lost, possibly also the area where the overall cost has its global minimum. The original method that maximizes commonality is extended to incorporate a more realistic cost model that also accounts for oversizing of individual components by adding a subsequent cost optimization algorithm (Rötzer et al., 2020).

Instead of a design approach that allows to select the design variables completely independently, but retaining some advantages of the coupling, i.e., access to the complete solution space, a combined design decision approach (Daub et al., 2020) is selected in this work. Some design variable values are determined first, and the remaining set of design variable values is determined based on this selection. One possibility to implement this is the solution-compensation space approach (Funk et al., 2019; Vogt et al., 2018b), where variables that can be adjusted in a later design phase are used to compensate for limited controllability in early-decision variables.

This paper addresses the question how to include solution-compensation spaces into product family design to increase the flexibility to optimize the commonality pattern and component designs. At the same time, the advantages of using the solution space approach to decrease the combinatorics is retained. The paper is organized as follows. In the next section, first the general product family design optimization problem is introduced. Existing approaches to optimize product families, classify design decisions and compute solution-compensation spaces are revised. In the following Section 3, a new algorithm is introduced to design product families using solution-compensation spaces, which is demonstrated in an example in Section 4. Finally, the findings are summarized in the last section and possible areas to explore further are given.

2 Related Work

2.1 Product family design

Product family design is the simultaneous design of multiple product variants (Fujita and Yoshida, 2004). This work focuses on scale-based product families (Simpson et al., 2001), which means that between different variants, components are described using the same design variables, only their magnitude, i.e., scaling, varies. In contrast to this, module-based product families allow to add or exchange entire modules that might be described by different design variables. A compact problem statement is given as

$$\min_{x,\xi} C_{PF}(x,\xi,c) \qquad (1)$$

$$\text{subject to: } g(e,x) = v_{lb} - v(x,e) \leq 0 \qquad (2)$$

$$x_{lb} \leq x \leq x_{ub}\,. \qquad (3)$$

The design variables x contain the sensor properties. To simplify the explanations, within this paper, each component is described using one design variable $x_i, i = 1..n_C$, where n_C is the total number of components and also describes the dimensionality of the optimization problem. Goal of the product family design is to determine optimal values for each design variable x_i for each variant $k = 1..n_p$ of the total number of product variants n_p. The assignment scheme $\xi_{ki} \in 1..n_{CV_i}$ assigns each product variant k and component i one component variant, where n_{CV_i} is the number of variants of component i. The total cost of the product family C_{PF} has to be minimized, see Eq. (1), which also depends on the parameters of the cost model c and the assignment scheme ξ. The commonality pattern ζ determines whether components are different or shared between the variants of the product family. For two product variants P_1 and P_2, the commonality pattern ζ_i for component i reads

$$\zeta_i^{P_1 \leftrightarrow P_2} = \begin{cases} 1, & \text{if } x_{i,P_1} = x_{i,P_2}. \\ 0, & \text{else.} \end{cases} \qquad (4)$$

The visibility v of the relevant objects in the surrounding, which are described using the environment parameters e, has to exceed a lower boundary v_{lb}, see Eq. (2). Finally, the design space is restricted by lower and upper bounds in Eq. (3). This problem statement is also given in graphical form in Figure 1, where the design variables and design parameters are given in the bottom and the quantities of interest at the top. Links between the nodes are used to indicate the hierarchical dependencies in the modeling. If values of the design parameters (environmental parameters e and cost parameters c) are known and the design variables selected, the quantities of interest are determined.

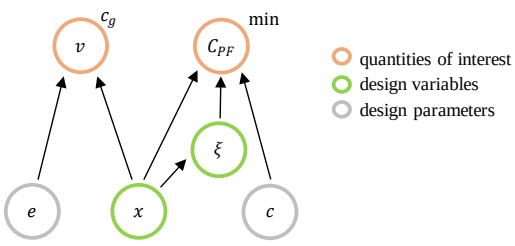


Figure 1: Graphical representation of the problem statement given in Eqs. (1)-(3).

As mentioned in the introduction, there are different possibilities to solve this optimization problem. Since the visibility is generally a nonlinear function and for complex scenarios even non-convex (e.g., a vehicle has to be detected and monitored that turns left or right, each with a certain probability), global optimization algorithms like Genetic Algorithms should be used to determine a global optimum (Fujita and Yoshida, 2004). Another approach using the computation of

solution spaces (Rötzer et al., 2020) is sketched in Figure 2(b)-2(d) for an example of two design variables and two product variants.

The basic idea of solution space engineering is sketched in Figure 2(a). Instead of determining one design that minimizes an objective function and satisfies the constraints, a set of good designs is identified, which constitutes a solution space. A solution space is part of the design space that includes only good designs, i.e., designs that satisfy all requirements. In Figure 2(a), the complete solution space for an abstract design problem is blue. Regions of the design space where at least one requirement is not fulfilled are red. To decouple the selection of design variables, a (n_C-dimensional) box is determined within the complete solution space, such that its volume is maximised and it only contains good designs. Each "side" of the box is a permissible interval Ω_i of design variable x_i. The box-shape guarantees that as long as each design variable is selected from within the respective permissible interval, the overall design will fulfil all requirements. This offers significant flexibility for design. Design variables do not have to assume a specified target value anymore, instead they may vary within their intervals to account for additional requirements or constraints, e.g., due to product family design. Various algorithms exist to find the permissible intervals for the design variables (Zimmermann and von Hoessle, 2013).

For an arbitrary example of a product family of two product variants and two design variables, permissible intervals of the design variables (box-shaped solution spaces) are computed for each product variant in the first step in Figure 2(b). For each design variable, the permissible intervals of both product variants are compared in Figure 2(c) and where they overlap, standardization of this component is technically feasible. In this example, the intervals of component 1 overlap and the assignment scheme is determined by comparing the costs if x_1 is shared or not in Figure 2(d). Comparing the intervals of both product variants of component 2, they do not overlap and the costs of shared or different selections do not have to be evaluated. However, looking at Figure 2(e), where the complete solution spaces of both variants are sketched, it can clearly be seen that if the dependencies between the two design variables are considered in the optimization, component 2 could also be shared.

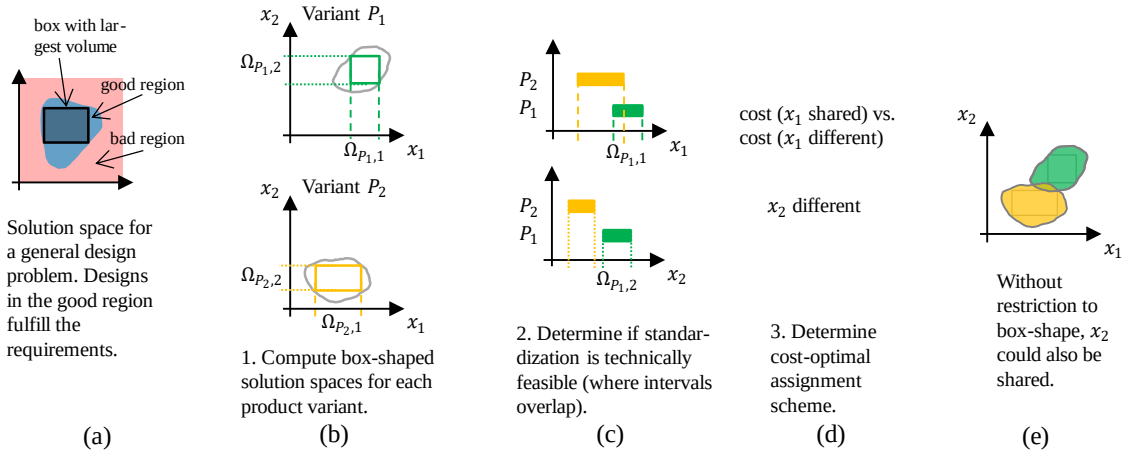


Figure 2: Two-level optimization for product family design using solution-space optimization.

2.2 Interdependent, dependent, independent design decisions

To amend this shortcoming, we look at the analysis of different design decisions for systems engineering (Daub et al., 2020). Three fundamentally different ways to reach distributed design decisions are compared, which is sketched in Figure 3 for two design variables x_1 and x_2. They are analysed with respect to costs, which is described as the necessary amount of information exchange, and flexibility, which is measured as the volume of the set of possible choices of the design variables that fulfil the constraint $g(x_1, x_2)$.

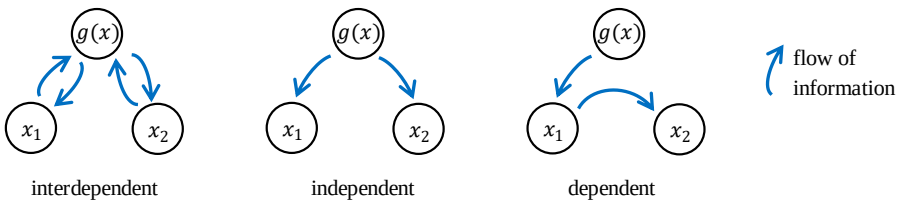
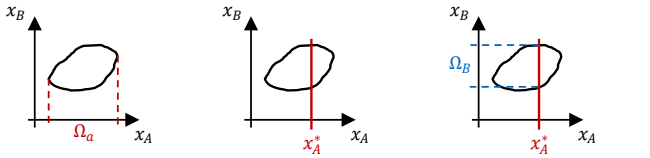


Figure 3: Flow of information for interdependent, independent and dependent design decisions, adapted from (Daub et al., 2020).

In interdependent design decisions, a repeated information exchange is necessary to find permissible values for the design variables x_1 and x_2. Since the full dependency between the design variables is considered, it allows maximal flexibility but comes at high costs. For independent designs, the constraint is decomposed first to reach permissible intervals for both design variables. While the decomposition also requires effort and reduces flexibility, the design variables can be selected in parallel independently from each other once the requirement is decomposed, which is very inexpensive. This principle is used for example when box-shaped solution spaces are selected. For the dependent design decision, first the value of one design variable is selected and based on this, the second is computed. Compared to the independent design approach, it increases the flexibility, since the dependency between the variables is considered, but it also leads to larger costs. The dependent design principle is used in the solution-compensation space approach.

2.3 Solution-compensation spaces

Solution-compensation spaces are introduced to allow greater flexibility to select design variables when they are decoupled using boxed-shaped solution spaces (Vogt et al., 2018b). When the design problem consists of many dimensions and constraints, the boxes tend to become very small. The key idea is to divide the design variables into two sets, the early decision variables x_A and the late decision variables x_B. Early decision variables are selected from experience or sensitivity analyses, such that they have the largest influence on the overall system performance and therefore have to be selected early in the design process. Late-decision variables have to be selected such that the designer can adjust them accurately in a later development phase. The method is sketched in Figure 4 for two design variables.



1. Determine interval for x_A, s.t. a good x_B exists.

2. Select x_A^* in early development phase.

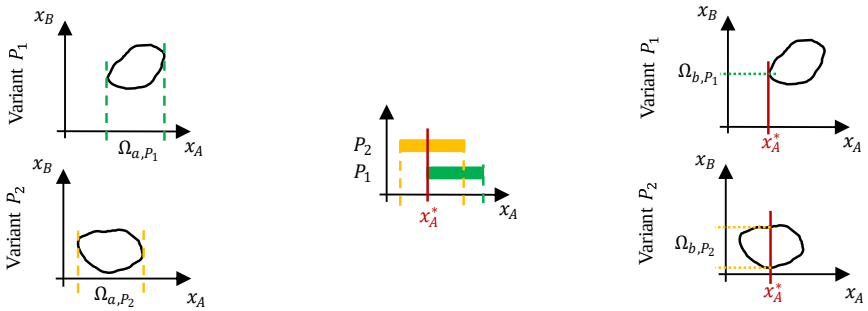
3. Select x_B from emerging permissible interval.

Figure 4: Solution-compensation space approach.

Permissible intervals for early decision variables become significantly larger by allowing the late decision variables to compensate them in a later phase of the development process, after the early decision variables have been selected as x_A^*. The asterisk-subscript is used to indicate that the solution is optimal. Increased flexibility in the selection of early decision variables comes at the cost of only being able to determine late decision variables once the early decision variables are selected. This makes the design process dependent according to the terminology introduced in the previous section.

3 Product family design with solution-compensation spaces

To combine the advantages of product family design using solution-spaces with the advantages of dependent design decisions, this paper introduces solution-compensation spaces into the product family optimization framework. While (Daub et al., 2020) measures the flexibility as the magnitude of the set of possible design choices, for product family design optimization, it is also relevant how many possibilities exist to standardize components between different variants. An overview of the method is given in Figure 5 for an arbitrary two-dimensional design problem.



1. Compute permissible intervals for early-decision variables x_A (sensor characteristics β).

(a)

2. Identify cost-optimal commonality and design variable values x_A^*.

(b)

3. Determine solution-spaces of late-decision variables (sensor positionings p).

(c)

Figure 5: Three-level optimization for product family design with solution-compensation spaces.

The overall problem statement from Eqs. (1)-(3) is divided into three subsequent optimization problems. In the first optimization, sketched in Figure 5(a), for each product variant k permissible intervals $\Omega_{A,k}$ are computed for the early decision variables.

$$\min_{\Omega_{A,k}} -\mu(\Omega_{A,k}) \qquad (5)$$

$$\text{subject to: } \forall x_{A,k} \in \Omega_{A,k}\ \exists\ x_{B,k} \text{ such that } g(e, x_{A,k}, x_{B,k}) \leq 0. \qquad (6)$$

In Eq. (5), the intervals for early-decision variables are selected such that the volume of the space containing good designs $\mu(\Omega_{A,k})$ is maximized. For x_A to be permissible, there has to exist at least one x_B for which the constraints are fulfilled, see Eq. (6). For linear constraints, this can be achieved with reasonable effort by eliminating the x_B variables successively through projecting them into the Ω_A space (Vogt et al., 2018a). For nonlinear constraint functions as in the described design problem, it becomes more complicated. The simplest approach is brute-force and consists of a nested sampling where for each sample of the early decision variables, a separate sampling of the late decision variables is performed. The constraints are evaluated for this one x_A-sample and all x_B-samples which can be aborted if the constraint is fulfilled for one of the x_B designs, since this suffices to fulfill Eq. (6).

The result of the first optimization are permissible intervals $\Omega_{A,k}$ for the early decision variables for each product variant. This serves as input to the second optimization in Eqs. (7)-(8), and Figure 5(b), to determine optimal design variable values x_A^* and assignment scheme ξ, such that the overall cost of the product family C_{PF} is minimized. The cost function in Eq. (7) also depends on cost parameters c.

$$\min_{x_{A,k}^*, \xi} C_{PF}(x_{A,k}, \xi, c) \qquad (7)$$

$$\text{subject to: } x_{A,k} \in \Omega_{A,k} \qquad (8)$$

Similar to the approach described in Section 2.1, only those commonality possibilities have to be considered that are technically feasible, so this problem can be parallelized for each component in x_A. However, only the early decision variables are considered in this optimization and their optimal commonality is determined. Therefore, this approach only works when the early decision variables are selected to have the only (or at least the dominating) effect on the overall cost of the product family. Figure 6 is modified from the basic problem statement sketched in Figure 1 by adding sensor characteristics β and sensor positionings p separately. Together, they constitute the design variables $x = [p, \beta]^{\mathrm{T}}$. From the description of the problem of perception for automated driving, it can be followed that while the sensor positionings have an influence on the perception performance, the characteristics of the sensors drive the costs. Accordingly, the sensor characteristics β are selected as early decision variables.

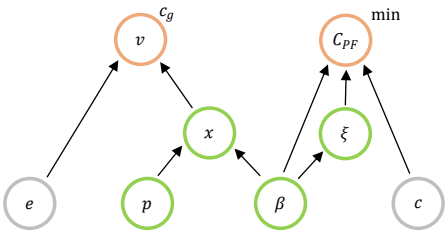


Figure 6: Graphical representation of the problem statement where sensor positionings p and characteristics β are chosen sequentially.

After selecting optimal values and commonality of the sensor characteristics in the second optimization, the last optimization in Figure 5(c) is used to determine optimal values of the dependent parameters or late-decision variables x_B. Similar to Eq. (5) and for each product variant independently, a set of good designs $\Omega_{B,k}$ is computed from Eqs. (9)-(10).

$$\min_{\Omega_{B,k}} -\mu(\Omega_{B,k}) \qquad (9)$$

$$\text{subject to: } g(e, x_{A,k}^*, x_{B,k}) \leq 0 \qquad (10)$$

Eq. (6) guarantees that the set of good designs $\Omega_{B,k}$ is non-empty. If all design variables are selected as early decision variables, this approach condenses to the existing approach in Figure 2. If only one variant is considered and the optimal value for early decision variables is computed by minimizing a cost function, the basic solution-compensation space approach in Figure 4 appears.

By fully considering the dependency between sensor characteristics and positions, the flexibility to find the optimal commonality pattern in the product family is increased. From a computational point of view, the first and third optimization problems require constraint evaluations in the computation of the permissible intervals, which might be expensive. However, since commonality is not considered here, the computations for each product variant are completely independent and can be parallelized. Combinatorics of the product family is only relevant in the second optimization. Computing independent intervals in the first optimization reduces the computational effort, since only feasible combinations have to be analyzed for cost-optimality. If the cost model does not contain any terms that depend on more than one component simultaneously, this optimization can be done in parallel for the different components. The introduced approach is applied to a simple design example in the next section.

4 Application to an example

Automated vehicles use various algorithms to fuse information from different sensors and determine locations and classifications of relevant objects in their surrounding environment. These algorithms are mostly based on machine-learning (Bhatt et al., 2020; Wirges et al., 2018) and have to be validated by testing them on a large driven distance and number of scenarios (Amersbach and Winner, 2019; Koopman and Wagner, 2018). Furthermore, number, type and position of perception sensors have to be decided in an early development phase, when neither the detection algorithms nor a detailed vehicle layout exists (Hartstern et al., 2020). Currently, for research and proof-of-concept vehicles, the focus lies on ensuring a robust environment perception (Kocic et al., 2018), which leads to a large number of costly perception sensors like multi-purpose cameras and lidars (Buchholz et al., 2020). While current research mainly focuses either on the performance of sensor fusion and object detection algorithms (Yeong et al., 2021) or on the optimal positioning of specific sensors (Hu et al., 2022; Roos et al., 2021) a framework that simultaneously determines the required sensor characteristics and sensor positionings for automated vehicles does not exist to the authors' knowledge. Furthermore, the resulting optimization problem to design the perception sensor setup is characterized by a complex and generally non-linear evaluation metric, the visibility of relevant objects in the surrounding.

Consider the example sketched in Figure 7. For two variants of the Ego vehicle P_1 and P_2, a camera is designed, which is described by the horizontal opening angle of its field of view β_h as well as the mounting position p_y and orientation α_z. For a camera setup to be considered good, an oncoming vehicle, sketched in grey, has to be detected. In this paper, it is simply checked whether the relevant object is visible to the sensor, i.e., within the field of view of the camera. Visibility of the object is a prerequisite for any algorithm to detect the object robustly.

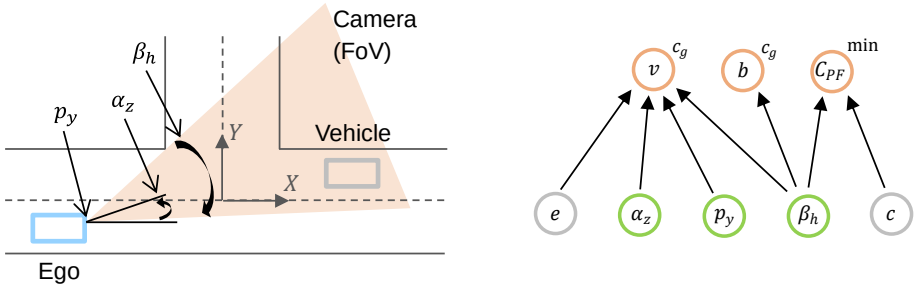


Figure 7: Sketch and dependency graph of the example.

The scenario description is collected in the design parameters e, here it contains the relative position and dimensions of both vehicles. The two product variants are members of different product segments. While product variant P_1 is a small, low-cost vehicle that moves at moderate speeds, variant P_2 is grand and expensive and required to handle larger accelerations. The variants are characterized by vehicle dimensions, which are the length from the center of gravity towards the front l_E and the width w_E. To account for the different budgets b, an additional requirement is added in Figure 7 that restricts the horizontal opening angle to some upper bound $\beta_{h,ub}$, see Eq. (11). It would also be possible to account for this with variant-specific design space boundaries.

$$b(\beta_h) = \beta_h - \beta_{h,ub} \leq 0 \qquad (11)$$

Finally, since variant P_1 is only required to move at moderate speeds, the lower bound of the visibility constraint v_{lb} in Eq. (2) can be relaxed compared to the second variant. Values for the parameters describing the product variants are given in Table 1.

Table 1. Design parameter values and constraint boundaries to describe the product variants.

	length of Ego vehicle, l_E	width of Ego vehicle, w_E	upper bound for opening angle, $\beta_{h,ub}$	lower bound for visibility, v_{lb}
product variant P_1	1.0 m	1.9 m	40°	0.4
product variant P_2	1.2 m	2.2 m	75°	0.99

The design space for the three design variables is given in Table 2. The variable describing the sensor characteristics is the horizontal opening angle β_h of the field of view and this is selected as early decision variable. As mentioned in the previous section, the choice of early and late decision variables from the design variables depends on the problem specifics. The early decision variables are selected such that they contribute significantly to the optimization objective of the product family. For this application, the sensor characteristic, i.e., the horizontal opening angle β_h of the field of view, dominates the costs ($x_A = \beta_h$). Subsequently, the commonality of the late decision variables does not, or hardly, influence the overall cost of the product family. Since all design variables influence the constraint functions, the late decision variables, here the camera position and orientation ($x_B = p = [\alpha_z, p_y]$), can be used to increase the flexibility to share components described by early decision design variables and this might decrease the overall costs.

Table 2. Design space boundaries and early/late decision for the design variables.

	camera orientation α_z	camera position p_y	horizontal opening angle β_h
early or late decision	late ($\alpha_z = x_{B,1}$)	late ($p_y = x_{B,2}$)	early ($\beta_h = x_A$)
lower bound x_{lb}	$-\pi/4$	$-w_E/2$ m	0°
lower bound x_{ub}	$\pi/4$	$w_E/2$ m	80°

Figure 8(a) shows the results of the first optimization step for both vehicle variants, the solution-compensation space according to Eqs. (5)-(6). Results of a classical box-shaped-solution space with no differentiation between early and late decision variables is plotted in Figure 8(b) for comparison. Each design consists of three design variables and a colour property indicates whether this combination of design variables is considered good (green), the visibility constraint (red) or the budget constraint (blue) are violated. To improve clarity, the four-dimensional results are projected onto three-dimensional plots (two design variables and colour property). This is achieved by sampling over the complete design space for the two design variables of the plot, while the other design variables are sampled within their permissible intervals (not over the entire design space). The permissible intervals for the first variant are sketched in black, for the second variant in blue, and both solutions are pictured together in the bottom row of Figure 8. For the presented solution-compensation space approach, the result, to be very specific, is only the plot of the two intervals over the early decision variable β_h in the bottom row of Figure 8(a). Permissible, independent intervals of the positioning variables (late decision, $p = [p_y, \alpha_z]$) are computed later in the third optimization. However, to be able to compare the results to the classical solution-space approach in the projections, the intervals of the late-decision variables are selected as maximal and minimal values that fulfil the compensation constraints, Eq. (6), to achieve the plots in the top and middle row in Figure 8(a).

While this simple example leads to large solution spaces for both approaches, the solution-compensation space approach leads to overlapping intervals for β_h in Figure 8(a), which means a shared sensor can be selected. For each permissible opening angle in those intervals, Eq. (6) ensured that at least one combination for the sensor positionings (late decision variables $p = [p_y, \alpha_z]$) exists where the requirements are fulfilled. However, the selection of the late decision variables now depends on the specific selection of β_h. A larger flexibility in the selection of the opening angle, that drives the costs of the product family, is compensated by decreased flexibility in the selection of the positionings. This can also be seen in the first two rows of Figure 8(a). While the interval over β_h (early decision) is maximized, the box also contains designs where the requirements are not fulfilled for certain positionings. The box-shaped solution spaces in Figure 8(b), combined for both product variants in the bottom row, ensures that each of the three design variables ($x = [\beta_h, p_y, \alpha_z]$) can be selected independently within its permissible interval and the constraints will be fulfilled. However, by trying to find the solution-space box with the overall largest volume in the good designs, the classical algorithm leads to two boxes that do not overlap for β_h and a common camera for both product variants cannot be found without compensation.

In the second optimization, the optimal assignment scheme is computed that minimizes the cost of the product family following Eqs. (7)-(8). A simple cost model is assumed for the cost of the product family (Ehrlenspiel et al., 2020), that consists of two parts. The first part, design specific costs, increases with larger values for the opening angles and the second part increases with the number of component variants. Since the lower boundaries for both variants, resulting from the first optimization, are close, a common sensor is selected with an opening angle of $\beta^*_{h,P_1} = \beta^*_{h,P_2} = 16°$, which is the lower boundary of the permissible interval of P_2 in Figure 8(a). The commonality pattern reads $\zeta_{\beta_h}^{P_1 \leftrightarrow P_2} = 1$ (component is shared) and the assignment scheme $\xi_{P_1\beta_h} = \xi_{P_2\beta_h} = 1$ (both share the same component, with index 1).

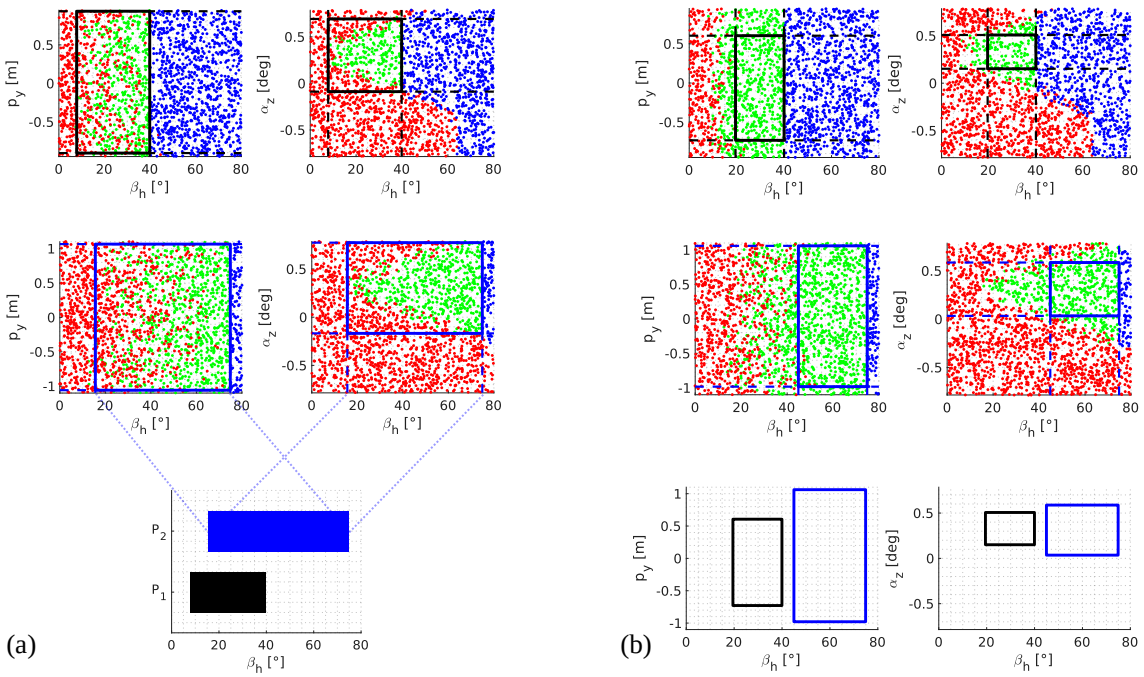


Figure 8: (a) Solution-compensation spaces and (b) box-shaped solution spaces for variants P_1 (top row, black boxes) and P_2 (middle row, blue boxes). Permissible intervals for both variants (bottom row).

After selecting the optimal sensor characteristics, permissible intervals for the positioning design variables $p = [p_y, \alpha_z]^T$ are computed from Eqs. (9)-(10) for both variants individually. Resulting solution spaces are given in Figure 9, where, similarly to Figure 8, green dots indicate good designs and red dots visibility constraint violations. The optimal values can be selected in a later development phase according to secondary criteria, separately for both variables if a box-shaped solution space is computed to decouple them.

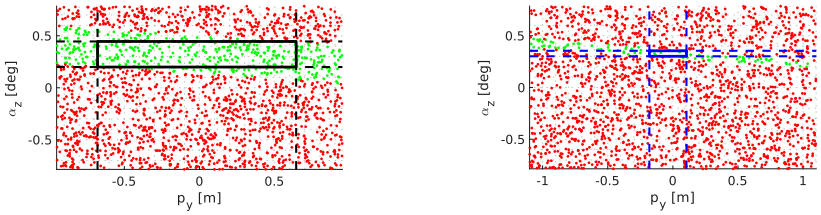


Figure 9: Solution spaces for the late decision variables for variant P_1 (left) and P_2 (right).

Even though the selected example is very simple, only two constraints are considered, and the product variants are only slightly different, the advantages of the proposed method can be seen. Larger permissible intervals for the sensor characteristics in the first optimization leads to greater flexibility to select a cost-optimal product family in the second step. To reduce the computational effort, only combinations are considered where the constraints are fulfilled. Permissible values for sensor positions are computed in the third optimization step, where the constraint formulation in the first optimization guarantees that at least one good position exists.

5 Conclusion and Outlook

The motivating problem was described as finding the setup of perception sensors for a product family of automated vehicles, where the cost is minimized and the visibility of the relevant objects in the surrounding is guaranteed. This is characterized by a large number of design variables and constraints. Nonlinear constraint functions and combinatorics how components are shared between product variants can make the optimization problem prohibitively expensive. This paper introduced an approach to seek the optimal design of a product family utilizing solution-compensation spaces. A previously presented approach based on classical solution spaces computes permissible intervals for all design variable of each product variant. Components can be shared between product variants where the intervals overlap. The disadvantage of this approach is that for high-dimensional problems, the intervals become very small, which in turn restricts the

combination possibilities and optimality in the product family. By (re-)introducing a dependency between sensor characteristics and positions, the flexibility to select the former is increased. In the solution-compensation space approach, larger permissible intervals for early decision variables are achieved by allowing late decision variables to compensate different choices. However, this comes at additional costs in the first optimization step to find the permissible intervals for the design variables describing the sensor characteristics, such that at least one permissible solution exists for the (later) positionings.

An important next step is to analyse the performance of the proposed method by applying it to a more complex example. Flexibility when selecting the components, but also the quality of the overall result has to be analysed and contrasted with computational effort. An efficient computation of the solution-compensation spaces will be crucial, which requires further investigation. The basic idea to utilise the dependency between sensor characteristics and sensor positionings to increase the flexibility in the product family could also be beneficial in a product family optimization method that is not based on solution spaces. Finally, for the method to be applicable to a larger number of design problems, it should be extended to module-based product families.

References

Amersbach, C., Winner, H., 2019. Defining Required and Feasible Test Coverage for Scenario-Based Validation of Highly Automated Vehicles, in: 2019 IEEE Intelligent Transportation Systems Conference (ITSC). IEEE, Auckland, New Zealand, pp. 425–430. https://doi.org/10.1109/ITSC.2019.8917534

Bhatt, D., Bansal, D., Gupta, G., Lee, H., Jatavallabhula, K.M., Paull, L., 2020. Probabilistic Object Detection: Strengths, Weaknesses, Opportunities, in: Workshop on AI for Autonomous Driving, the 37 Th International Conference on Machine Learning. Vienna, Austria, p. 7.

Bock, F., Siegl, S., Bazan, P., Buchholz, P., German, R., 2018. Reliability and test effort analysis of multi-sensor driver assistance systems. Journal of Systems Architecture 85–86, 1–13. https://doi.org/10.1016/j.sysarc.2018.01.006

Broggi, A., Buzzoni, M., Debattisti, S., Grisleri, P., Laghi, M.C., Medici, P., Versari, P., 2013. Extensive Tests of Autonomous Driving Technologies. IEEE Trans. Intell. Transport. Syst. 14, 1403–1415. https://doi.org/10.1109/TITS.2013.2262331

Buchholz, M., Gies, F., Danzer, A., Henning, M., Hermann, C., Herzog, M., Horn, M., Schoen, M., Rexin, N., Dietmayer, K., 2020. Automation of the UNICARagil vehicles, in: 29th Aachen Colloquium Sustainable Mobility. pp. 1531--1560.

Daub, M., W, F., Zimmermann, M., 2020. Optimizing Distributed Design Processes for Flexibility and Cost, in: DS 103: Proceedings of the 22nd International DSM Conference (DSM 2020). The Design Society, MIT, Cambridge, Massachusetts, pp. 10–10. https://doi.org/10.35199/dsm2020.2

Ehrlenspiel, K., Kiewert, A., Lindemann, U., Mörtl, M., 2020. Kostengünstig Entwickeln und Konstruieren: Kostenmanagement bei der integrierten Produktentwicklung. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-62591-0

Eichstetter, M., Müller, S., Zimmermann, M., 2015. Product Family Design with Solution Spaces. Journal of Mechanical Design 137, 121401. https://doi.org/10.1115/1.4031637

Fujita, K., Yoshida, H., 2004. Product Variety Optimization Simultaneously Designing Module Combination and Module Attributes. Concurrent Engineering 12, 105–118. https://doi.org/10.1177/1063293X04044758

Funk, M., Jautze, M., Strohe, M., Zimmermann, M., 2019. Sequential Updating of Quantitative Requirements for Increased Flexibility in Robust Systems Design. Proc. Int. Conf. Eng. Des. 1, 3531–3540. https://doi.org/10.1017/dsi.2019.360

Gogri, M., Hartstern, M., Stork, W., Winsel, T., 2020. A Methodology to Determine Test Scenarios for Sensor Constellation Evaluations, in: 2020 IEEE 3rd Connected and Automated Vehicles Symposium (CAVS). IEEE, Victoria, BC, Canada, pp. 1–5. https://doi.org/10.1109/CAVS51000.2020.9334603

Hartstern, M., Rack, V., Stork, W., 2020. Conceptual Design of Automotive Sensor Systems: Analyzing the impact of different sensor positions on surround-view coverage, in: 2020 IEEE SENSORS. IEEE, Rotterdam, Netherlands, pp. 1–4. https://doi.org/10.1109/SENSORS47125.2020.9278638

Hu, H., Liu, Z., Chitlangia, S., Agnihotri, A., Zhao, D., 2022. Investigating the Impact of Multi-LiDAR Placement on Object Detection for Autonomous Driving, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 2550--2559.

Jackson, D., Richmond, V., Wang, M., Chow, J., Guajardo, U., Kong, S., Campos, S., Litt, G., Arechiga, N., 2021. Certified Control: An Architecture for Verifiable Safety of Autonomous Vehicles. arXiv:2104.06178 [cs, eess].

Khatab, E., Onsy, A., Abouelfarag, A., 2022. Evaluation of 3D Vulnerable Objects' Detection Using a Multi-Sensors System for Autonomous Vehicles. Sensors 22, 1663. https://doi.org/10.3390/s22041663

Kocic, J., Jovicic, N., Drndarevic, V., 2018. Sensors and Sensor Fusion in Autonomous Vehicles, in: 2018 26th Telecommunications Forum (TELFOR). IEEE, Belgrade, pp. 420–425. https://doi.org/10.1109/TELFOR.2018.8612054

Koopman, P., Wagner, M., 2018. Toward a Framework for Highly Automated Vehicle Safety Validation. Presented at the WCX World Congress Experience, pp. 2018-01–1071. https://doi.org/10.4271/2018-01-1071

Koopman, P., Wagner, M., 2016. Challenges in Autonomous Vehicle Testing and Validation. SAE Int. J. Trans. Safety 4, 15–24. https://doi.org/10.4271/2016-01-0128

Lambert, J., Carballo, A., Cano, A.M., Narksri, P., Wong, D., Takeuchi, E., Takeda, K., 2020. Performance Analysis of 10 Models of 3D LiDARs for Automated Driving. IEEE Access 8, 131699–131722. https://doi.org/10.1109/ACCESS.2020.3009680

Maurer, M., Gerdes, J.C., Lenz, B., Winner, H. (Eds.), 2016. Autonomous Driving. Springer Berlin Heidelberg, Berlin, Heidelberg. https://doi.org/10.1007/978-3-662-48847-8

Ponn, T., Kroeger, T., Diermeyer, F., 2020. Identification and Explanation of Challenging Conditions for Camera-Based Object Detection of Automated Vehicles. Sensors 20, 3699. https://doi.org/10.3390/s20133699

Roos, S., Volkel, T., Schmidt, J., Ewecker, L., Stork, W., 2021. A Framework for Simulative Evaluation and Optimization of Point Cloud-Based Automotive Sensor Sets, in: 2021 IEEE International Intelligent Transportation Systems Conference (ITSC). IEEE, Indianapolis, IN, USA, pp. 3231–3237. https://doi.org/10.1109/ITSC48978.2021.9564871

Rötzer, S., Thoma, D., Zimmermann, M., 2020. Cost Optimization of Product Families using Solution Spaces. Proc. Des. Soc.: Des. Conf. 1, 1087–1094. https://doi.org/10.1017/dsd.2020.178

Simpson, T.W., Maier, J.R., Mistree, F., 2001. Product platform design: method and application. Res Eng Design 13, 2–22. https://doi.org/10.1007/s001630100002

Vogt, M.E., Duddeck, F., Harbrecht, H., Stutz, F., Wahle, M., Zimmermann, M., 2018a. Computing solution-compensation spaces using an enhanced Fourier-Motzkin algorithm. Proc. Appl. Math. Mech. 18. https://doi.org/10.1002/pamm.201800103

Vogt, M.E., Duddeck, F., Wahle, M., Zimmermann, M., 2018b. Optimizing tolerance to uncertainty in systems design with early- and late-decision variables. IMA Journal of Management Mathematics 30, 269–280. https://doi.org/10.1093/imaman/dpy003

Wirges, S., Fischer, T., Stiller, C., Frias, J.B., 2018. Object Detection and Classification in Occupancy Grid Maps Using Deep Convolutional Networks, in: 2018 21st International Conference on Intelligent Transportation Systems (ITSC). IEEE, Maui, HI, pp. 3530–3535. https://doi.org/10.1109/ITSC.2018.8569433

Yeong, D.J., Velasco-Hernandez, G., Barry, J., Walsh, J., 2021. Sensor and Sensor Fusion Technology in Autonomous Vehicles: A Review. Sensors 21, 2140. https://doi.org/10.3390/s21062140

Zimmermann, M., von Hoessle, J.E., 2013. Computing solution spaces for robust design. Int. J. Numer. Meth. Engng 94, 290–307. https://doi.org/10.1002/nme.4450

Contact: N. Barthelmes, Laboratory for Product Development and Lightweight Design, TUM School of Engineering and Design, Faculty of Mechanical Engineering, Technical University of Munich, Boltzmannstr. 15, 85748 Garching, Germany, nicola.barthelmes@tum.de, zimmermann@tum.de.