The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3–7 June 2024, Lisboa, Portugal

DESIGN DOMAIN DISTRIBUTION FOR TOPOLOGY OPTIMIZATION USING MACHINE LEARNING ECCOMAS CONGRESS 2024

Felix Endress[1], Sergi Pagés i Diaz[1] and Markus Zimmermann[1]

[1] Technical University of Munich, TUM School of Engineering and Design, Department of Mechanical Engineering, Laboratory for Product Development and Lightweight Design, Boltzmannstraße 15, 85748 Garching near Munich, felix.endress@tum.de

Key words: Topology optimization, systems design, lightweight design, surrogate models, machine learning

Summary.

The design domain has significant influence on the result of topology optimization. Thus, when several components are to be topology optimized separately, the allocation of the available space to each of them will be crucial for the resulting mechanical performance of the system [1, 2, 3]. Unfortunately, the optimal allocation depends on the results of component optimizations which are not known initially. This chicken-or-egg problem was solved in [2] by solving several component topology optimization problems for varying design domains as preparation and making the relationship between resulting component mass and design space available as sample data. Based on this sample data, good design domains could be chosen manually. Unfortunately, this technique is limited to very simple design problems. To extend it to more complex design problems and to automate the approach, this paper proposes the following alternative: For several samples with varying design domains, topology optimizations are performed for each component. Then, based on the produced dataset, meta models are trained to estimate (1) the physical feasibility and (2) the mass of individual components, as a function of the dimensions of the allocated design domain. The actual design domains are then allocated by numerical optimization using these meta models. Final geometries are determined by detailed topology optimizations based on the previous optimization results. The method is applied to the steering mechanism of a glider plane. A reduction in mass by 10.1%, compared to a manual distribution of design domains could be achieved.

1 INTRODUCTION

There are various approaches for topology optimization, all aiming for an optimal distribution of material in a given design domain with respect to certain objectives [4]. This makes topology optimization a versatile tool for generating structures when boundary conditions and loads are known. However, the results strongly depend on the optimization settings and problem formulations chosen [5, 1]. Further, in a generic development process, as shown in Fig. 1, system-level design is usually performed at an earlier stage, therefore influencing the detail design. For example, topology optimization design domains are usually allocated to components in the phase of systems design. For simple structures, such as beams or tubes, analytical solutions, e.g., on the deflections for a given load, are analytically available. So, combining

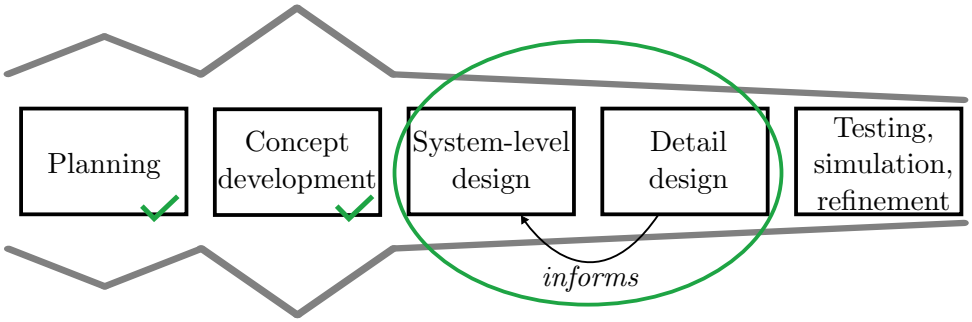

Figure 1: Modified design process from [6]. The scope of the introduced approach is highlighted by the green circle.

components into a system is straight forward, and the resulting quantities of interest can be calculated in the system's design phase. Unfortunately, when combining multiple topology-optimized components, relevant mappings between design variables and quantities of interest are not known a priori. There exists a chicken-or-egg problem, since for optimally distributing design domains, information from the detail design is required. The goal of this paper is to inform system-level design about feasible, topology optimized structures and their corresponding structural performance using machine learning.

This paper is organized as follows: After the Introduction in Section 1, the State of the Art has been reviewed in Section 2. Section 3 describes a use case, before in Section 4 the automated approach is illustrated and problem formulations are introduced. Then, in Section 5, the approach is applied to the use case, where a shared design domain inside the wing of a glider is divided and allocated to three individual brackets. The paper ends with a discussion of the results in Section 6 and a conclusion in Section 7.

2 STATE OF THE ART

The use of machine learning (ML) to improve the conventional framework for topology optimization is a growing research branch showing numerous approaches and applications [7]. Considering multi-component systems, one can find a few works exploiting ML methods. In [3, 8] and summarized in [9], an approach was used for system decomposition involving topology optimization, where stiffness requirements were distributed for connected components to satisfy system requirements and minimize system mass simultaneously. Information from the detail-level design was obtained through trained meta models that could predict structural responses for the optimization of various robotic arms. Design domains, defining the position of the interface between components, remained unchanged throughout the process. In [8] the same meta model, considering different design domain sizes, was used for all components in the system, to reduce training effort. However, again, design domain sizes were not treated as design variables in the system optimization. Only mass and stiffness budgets were distributed, enabling the decomposition of system requirements into decoupled requirements for individual components. This way, the simultaneous design of different components was enabled [10]. However, when topology optimization is applied, the system architecture, i.e. sizes of components or design domains, is usually prescribed and not part of an optimization problem [11].

Modifying the design domain in a topology optimization problem has been addressed in other studies, mainly with the goal to overcome performance limitations imposed by a fixed design patch [12, 13]. These efforts have been directed towards the optimization of single components, where better topologies were found by allowing the design domain to grow or adapt to the specific problem. In Maute and Ramm [12], Bézier splines were used to fit and re-mesh the design domain to the material distribution in the ongoing iteration, increasing the smoothness and performance of the final result by reducing the number of design variables. In Kim and Kwak [13], the design domain was incorporated into the optimization algorithm as an additional variable, obtaining better minima as a result. However, the scope in these publications was the design of isolated components, thus interactions and compromises occurring in system-level design were not considered. In some applications such as robotics, there exists extensive literature on the design of component lengths and linkage locations, e.g. topology synthesis, usually with the goal of an optimal trajectory for a specific task. Therefore, the system architecture is set fixed, and the potentials for lightweight design may be lost. Nevertheless, a few studies on mechanism design have been conducted where component topology optimization and linkage locations are computed simultaneously. Swartz et al. [14, 15] proposed an approach for multi-body planar systems where component topology optimization is performed, together with finding the best location of rotational joints. Still, this approach follows a layered mesh scheme where the design domain of each component remains unchanged, meaning there exists no interaction nor dependencies between the design domains of components on the system-level. In the work of Sun et al. [16, 17] an approach is presented for variable-length mechanisms based on Moving-Morphable-Components. Since design domains are affected by changes in component length, virtual design domains are introduced to transform the dynamic response of the problem into a static one, reanalyzing the sets of equivalent static loads. Still, in their work the topology optimization is posed as a minimum-compliance problem. An optimization of design domain on system-level is not performed with information from topology optimization. Similar to distributing design domains, in [18] joint locations and topologies of components are optimized simultaneously. However, whilst joint locations are continuously movable, the initial design domains remain unchanged throughout the optimization, and, individual stiffness requirements on component-level were not considered.

3 USE CASE

A system of three brackets is considered, as described in [2]. A common design domain is shared by three components, as illustrated by the dashed lines in Fig. 2. The individual component's function is to maintain a bearing in position. On system-level, a maximal height of $h_{req} = 60$ mm must not be exceeded and the length of both components must equal the distance between the attachment areas, which measures to $l_{req} = 80$ mm. The maximal width of the system is $w_{req} = 60$ mm. Two different bracket types are considered in this study. As shown in Fig. 2, the concept of bracket A is that the bearing is located at the bottom of the design domain, whereas for bracket B it is located at central height. The bearing was simplified and modelled as solid elements, spanning a height of $h_b = 3$ mm (dark grey in Fig. 2). The stiffness of these elements can be modified. Above (and for bracket B also below) the bearing, a cylindrical void region was modelled to ensure the accessibility of the machine element. For each component, two load cases were given from the application as $\mathbf{F}_A = [-450, -900, \pm 600]$ N

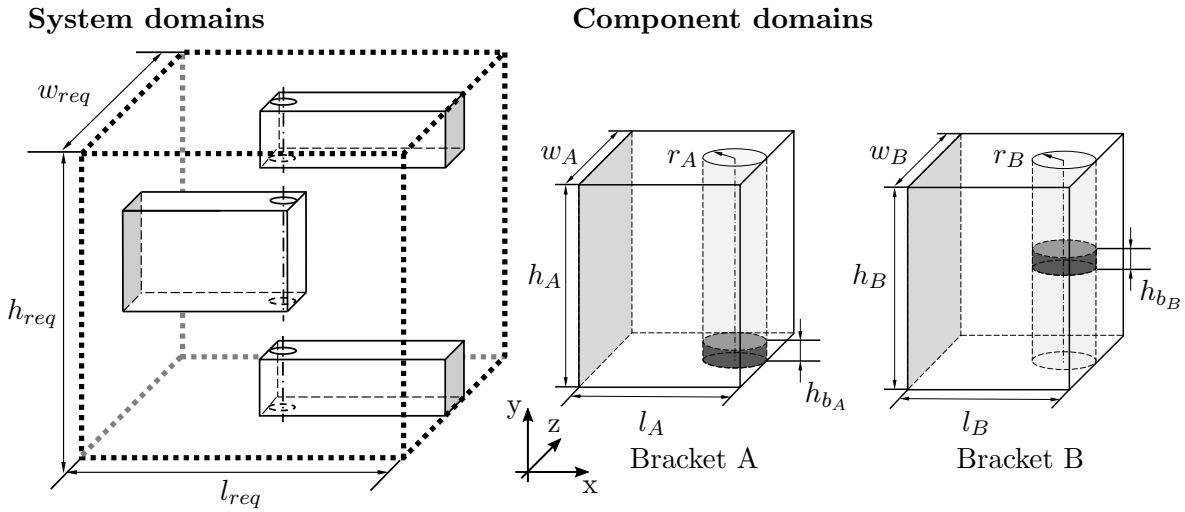

Figure 2: Sketches of the complete design domain and the different types of brackets.

for bracket A and $\mathbf{F}_B = [-330, -1100, \pm 770]$ N for bracket B. For all load cases a maximum deflection of ± 0.1 mm was prescribed to all directions (see coordinate system in Fig. 2).

4 METHODOLOGY

In this paper, the system optimization of Endress et al. [2] is performed using surrogate models. The approach is based on two trained estimators as ML models, for each component's response, and two optimization problem formulations: one at the component-, the other one at the system-level.

4.1 MASS AND PHYSICAL FEASIBILITY ESTIMATORS

The method to distribute design domains requires two meta models per component to perform the system optimization. Thus, in a first step, a bottom-up mapping is created using a sampling strategy (see [10]) by running multiple topology optimizations and assessing the respective performance, with varying design domains. This provides a database for the model training. As it is shown in Fig. 3, the models enable decoupling of individual geometries, found by applying topology optimization, from the higher levels in the graph.

Firstly, a mass estimator is trained, predicting the component's mass for a given design domain. Secondly, since some components may not fulfill requirements on the displacement, even when the volume-fraction reaches 100%, a classifier is used to predict physical feasibility. This becomes especially relevant for components with high lengths. During sampling, to reduce time to create a database, topology optimizations were aborted when the volume fraction reached 85%. For the surrogate modelling two different types of models are used: black-box models and an interpretable glass-box models. The black-box models chosen follow [3], where artificial neural networks (ANNs) were used. The models were combined with Bayesian Optimization to optimize the number of hidden layers, neurons per layer, and activation functions. For the classification of physical feasibility, a two-class Support Vector Classifier was trained with kernel functions of polynomial order $p = 3$. The models were trained and used in MATLAB using the

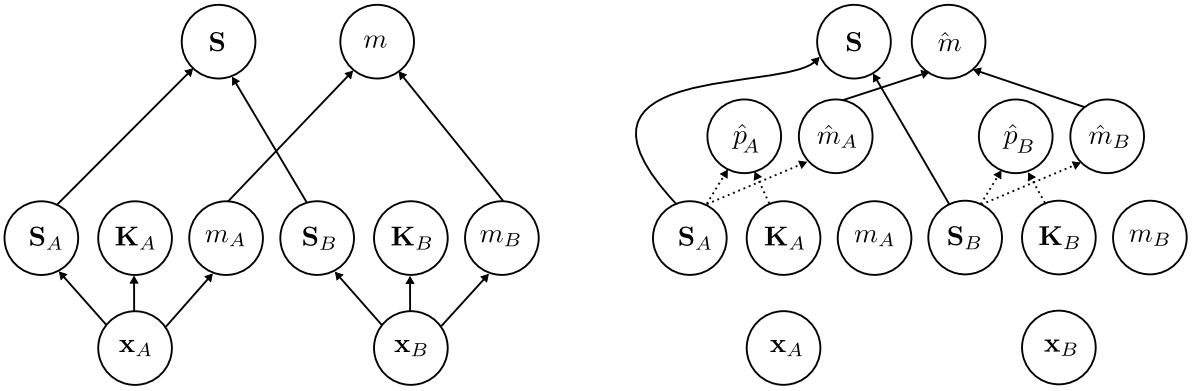

Figure 3: Dependencies in the system before (left) and after the introduction of meta models (right). Dashed lines indicate predictions made using the introduced meta models.

Machine Learning Toolbox. A 80% / 20% split for training and validation was used for both, the mass and physical feasibility estimators. For the mass predictor, only physically feasible samples were used to train the model.

For the interpretable model a Generalized Additive Model (GAM) is used [19]. The Explainable Boosting Machine (EBM) was implemented in python using the *interpret* package [20]. For the regressor, the dataset was split in 80% training data and 20% test data. Again, only feasible data points were used for training the mass estimator. For the feasibility estimator a split of only 10% training data was used, as there were only a few infeasible designs.

4.2 OPTIMIZATION PROBLEM STATEMENTS

The presented approach consists of system and component optimization. In the **system optimization** the total mass of the system is to be minimized, such that geometric constraints are fulfilled. The geometric constraints come from the use case considered, but could be of any form that can be parametrized (e.g. other shapes than boxes). Further, physical feasibility is required, which is also a constraint in the problem statement. For both, see in Equation 1, component mass m_i and feasibility p_i are realized using the machine learning estimators $\hat{m}_i$ and $\hat{p}_i$ introduced in Section 4.1:

$$\begin{aligned} &\min_{\mathbf{S}_i} \hat{m} = \sum_i \hat{m}_i(\mathbf{S}_i) \quad \text{with} \quad \mathbf{S}_i = [l_i, h_i, w_i] \\ &\text{s.t. } l = \sum_i l_i = l_{req}, \\ &\quad h = \sum_i h_i \leq h_{req}, \\ &\quad w_i \leq w_{req}, \\ &\quad \hat{p}_i(\mathbf{S}_i) \leq 0. \end{aligned} \tag{1}$$

For detailed design, i.e. **component optimization**, a topology optimization problem is solved. For a prescribed design domain, minimization of mass is realized while constraining de-

flections at the load introduction in all three directions. Formally, the optimization formulation at the component-level reads for each component i:

$$\begin{aligned} \min_{\rho_e} \; & m = \sum_{e=1}^{N_e} \rho_e \\ \text{s.t.} \; & u_x \leq u_x^c, \\ & u_y \leq u_y^c, \\ & u_z \leq u_z^c, \\ & \mathbf{KU} = \mathbf{F}, \\ & 0 < \rho_{min} \leq \rho_e \leq 1, \end{aligned} \tag{2}$$

where m is the component's mass, $\mathbf{K}$ is the stiffness matrix, $\mathbf{F}$ and $\mathbf{U}$ are the nodal load and displacement vectors respectively, ρ_e are the element densities, and N_e is the number of elements. A continuation scheme was applied by increasing the penalization from 1 to 4 in steps of 0.1 every 10 iterations. A density based filtering was used with a radius of 1.2 mm. Young's modulus was set to $E = 105,000$ MPa and Poisson's ratio to $\nu = 0.3$. The solver used is the MMA by [21], basis for the implementation in code was [22].

5 RESULTS

5.1 Bottom-Up Mapping: Sampling and Training of Models

After parametrizing the design domain of the brackets as boxes with individual heights, lengths and widths, shown in Fig. 4, bounds for the design domain dimensions (in mm) are chosen as

$$\begin{aligned} l_A \in [20, 70], h_A \in [15, 30], w_A \in [30, 60], \\ l_B \in [20, 80], h_B \in [15, 40], w_B \in [30, 60]. \end{aligned} \tag{3}$$

A latin hypercube sampling was performed with $n_1 = 50$ data points, for which topology optimizations were performed. Each sampling point is characterized by a different design domain. Results of the topology optimizations are the masses of components that are required to fulfil the displacement constraints, given the corresponding design domains (see Fig. 5). In a second

Concept Parametrized, box-shaped design domains

Figure 4: Concept for the brackets (blue for bracket A/ left, green for bracket B/ right) and parametrization of design domains as boxes (from [2]).

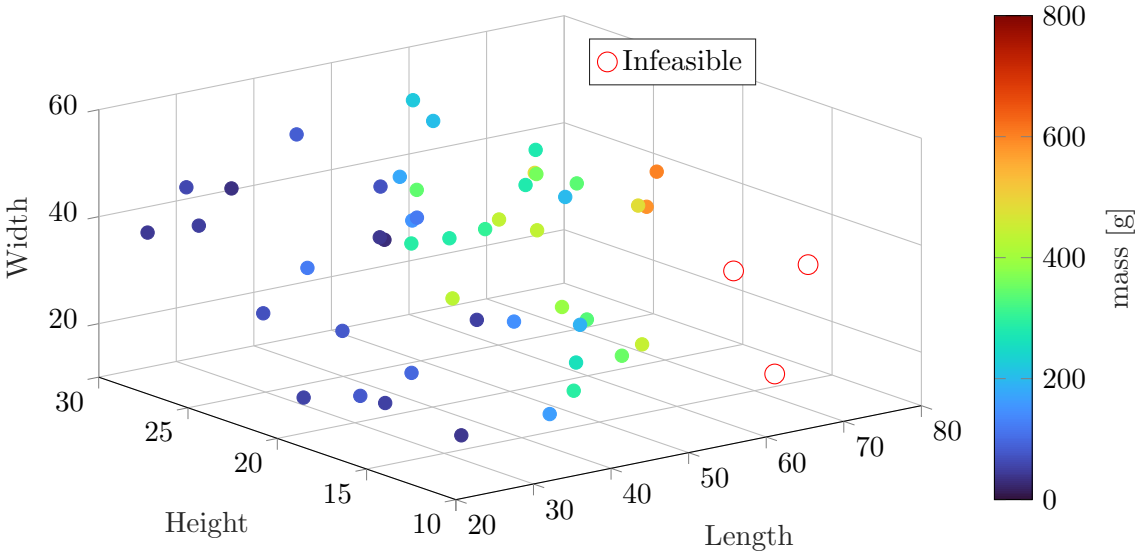

Figure 5: Sampling data for bracket A with three physically infeasible designs

step, the database for bracket B was increased by another $n_2 = 10$ data points in the area of high lengths. The same sampling was repeated for a different bearing model, where its stiffness was set to quasi rigid.

In the next step, ML models were trained, as described in Section 4, for different components (bracket A and B) and for different bearing models (*elastic* and *quasi rigid*). For training the mass estimators infeasible solutions were excluded. For the elastic bearing model, using the ANN, a R^2 score for bracket A resulted in 0.991 and for bracket B in 0.989. For the GAM for bracket A, $R^2 = 0.890$, and for bracket B, $R^2 = 0.970$. For the rigid bearing model the trained ANN led to an $R^2 = 0.976$ for bracket A and $R^2 = 0.968$ for bracket B. For visualizations of data points using the elastic bearing model, represented in Fig. 6, the width of components is set to its maximum ($w_i = 60$ mm). For the physical feasibility estimator, by manually selecting the training and validation sets, a reasonable training and test database could be defined. For both models, all designs were correctly classified (F1-score and accuracy of 1.00). For the interpretable model, error bounds could also be used to identify regions where further data would be favorable for training.

5.2 Top-Down Design

5.2.1 System-level Design

The constrained non-linear multivariable optimization problem is solved for the ANNs and interpretable models with different functions. For system optimization with ANNs, in MATLAB, an interior-point algorithm was used to search for the global optimum (fmincon). In the python implementation of the interpretable models, differential evolution was used (SciPy, optimize). The starting point was always $\mathbf{S_0} = [40, 30, 60, 40, 15, 60]$ mm as $\mathbf{S_0} = [l_A, h_A, w_A, l_B, h_B, w_B]$ for both optimizations. For the differential evolution algorithm a maximum of 10,000 iterations was chosen. The system optimization produces design domains which are shown in Tab. 1.

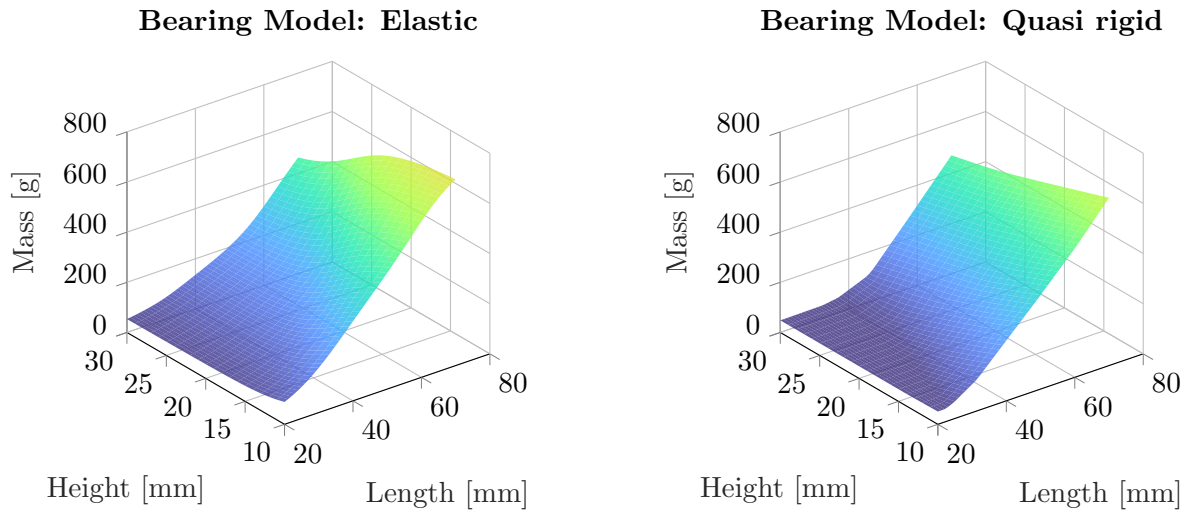

Figure 6: Meta models of bracket A at $w_i = 60mm$ with the bearing modelled as elastic elements (left) and as quasi rigid (right)

5.2.2 Detail-level Design

The results from system optimization, i.e. optimal design domains, are used for the component problem formulations. To find the actual mass of the system using the optimized domains, topology optimizations on both brackets are performed. Resulting topologies are shown in Fig. 7. In Tab. 1 the corresponding masses are shown. Since bracket A is mounted twice in the assembly (see Fig. 4), the total system mass is computed as $m = 2m_A + m_B$. Also in Tab. 1, comparisons to the rigid modelling of the conventional domain distribution are marked with $*$.

Table 1: Results of system optimization with realized masses for the single components and the corresponding system.

Nr.	Bearing model	m_A	m_B	m	Δ	l_A	h_A	w_A	l_B	h_B	w_B
		g	g	g	%	mm	mm	mm	mm	mm	mm
Artificial Neuronal Network											
1	Elastic	90	194	374	-10.1	33	15	30	47	30	60
2	Quasi rigid	98	154	350	-4.6*	38	22	30	42	15	30
3	Elastic	112	172	396	-4.8	38	22	30	42	15	30
Interpretable Machine Learning EBM											
4	Elastic	36	327	399	-4,1	23	16	58	57	28	51
Benchmark - Intuitive Distribution											
5	Elastic	141	134	416	-	40	20	60	40	20	60
6	Quasi rigid (*)	125	117	367	-	40	20	60	40	20	60

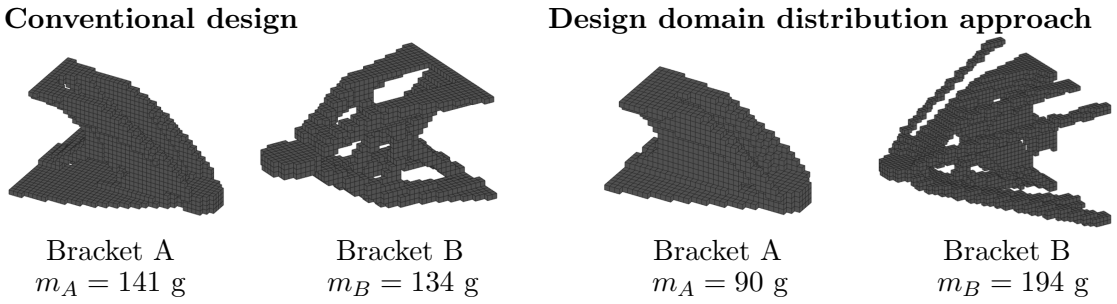

Figure 7: Topologies for an intuitive (left) and optimized (right, using the ANN) distribution of design domains. Bracket A is twice in the system, leading to a total mass of 374 g for the optimized and 416 g for the conventional design domain distribution.

6 DISCUSSION

6.1 Performance

Using ANNs for surrogate modelling, a mass reduction of -10,1% was realized, compared to an initial intuitive distribution of design domains. This indicates a great weight-saving potential when using the presented approach for systems design with topology optimization.

A loss of accuracy in the surrogate models might reduce lightweight potentials (compare mass reduction of -10.1% using the ANN with -4.1% using the EBM). Many data points are required to train the models to obtain highly accurate predictions. However, this comes with high computational effort to produce the database, and a compromise between computational effort and dataset quality must be achieved.

6.2 Decoupling Detail Design from System-level Design

It was shown that an engineer's choice in the detail design phase affects system-level design. In the presented use case, it was demonstrated that the specific structural properties of the bearing must be known in order to establish reasonable bottom-up mappings. The sampling requires topology optimizations depending on modelling details, such as solid and void elements for accessibility, as well as boundary conditions. Design domains, as well as resulting system mass changed when different bearing models were used. With the presented approach, these dependencies are resolved. System-level design obtains information from the detail-level thanks to the machine learning models, resulting into a decoupled optimization problem.

7 CONCLUSION

With the presented problem formulation, the response of the detail designs is modeled to optimize the system performance. In order to optimally distribute the design domains, the models must sufficiently represent reality, e.g., through appropriate modelling of the bearings and component attachments. Thus, information from the detail design is required in terms of component performance (here physical feasibility and mass), but also in terms of boundary conditions (connection, load introduction, solid and void areas). Besides missing information, the availability of data is an important factor in carrying out systems optimization based on surrogate models. In the case study presented, the database should be expanded to improve

quality of fit, especially for the GAM.

Future research will focus on systems where stiffness requirements are formulated at the system-level. For example, when components are stacked, i.e. attached to each other. In this case, in addition to design domains, stiffness budgets should also be distributed optimally within components.

8 FUNDING

This research was conducted as part of the PROVING research project in the national aeronautical research program VI-1 funded by the Bundesministerium für Wirtschaft und Klimaschutz. The authors declare no conflict of interest.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used Chat GPT and DeepL Write in order to improve language and readability. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

REFERENCES

[1] E. Tyflopoulos and M. Steinert, Messing with boundaries - quantifying the potential loss by pre-set parameters in topology optimization, *Procedia CIRP*, Vol. **84**, 979-985, 2019. https://doi.org/10.1016/j.procir.2019.04.307

[2] F. Endress, T. Kipouros, and M. Zimmermann, Distributing Design Domains for Topology Optimization in Systems Design, *Proceedings of the ASME 2023 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference. Volume 2: 43rd Computers and Information in Engineering Conference (CIE)*. Boston, Massachusetts, USA. August 20–23, 2023. ASME. https://doi.org/10.1115/DETC2023-114883

[3] L. Krischer and M. Zimmermann, Decomposition and optimization of linear structures using meta models, *Struct Multidisc Optim*, Vol. **64**, 2393–2407, 2021. https://doi.org/10.1007/s00158-021-02993-1

[4] O. Sigmund and K. Maute, Topology optimization approaches, *Structural and Multidisciplinary Optimization*, Vol. **48**, No. 6, 1031-1055, 2013. https://doi.org/10.1007/s00158-013-0978-6

[5] O. Sigmund, On benchmarking and good scientific practice in topology optimization, *Structural and Multidisciplinary Optimization*, Vol. **65**, No. 11, 2022. https://doi.org/10.1007/s00158-022-03427-2

[6] K. T. Ulrich and S. D. Eppinger, *Product Design and Development*, 6th ed., McGraw-Hill Education, New York, 2016. ISBN 978-0-07-802906-6.

[7] R. V. Woldseth, N. Aage, J. A. Bærentzen, and O. Sigmund, On the use of artificial neural networks in topology optimisation, *Structural and Multidisciplinary Optimization*, Vol. **65**, No. 10, 294, 2022. https://doi.org/10.1007/s00158-022-03347-1

[8] L. Krischer, A. V. Sureshbabu, and M. Zimmermann, Active-Learning Combined with Topology Optimization for Top-Down Design of Multi-Component Systems, *Proceedings of the Design Society*, Vol. **2**, Cambridge University Press, 1629–1638, 2022. https://doi.org/10.1017/pds.2022.165

[9] L. Krischer, F. Endress, T. Wanninger, and M. Zimmermann, Distributed design optimization of multi-component systems using meta models and topology optimization, *Structural and Multidisciplinary Optimization*, Vol. **28**, No. 4, 233–254, 2024. https://doi.org/10.1007/s00158-024-03836-5 (accepted)

[10] M. Zimmermann, S. Königs, C. Niemeyer, J. Fender, C. Zeherbauer, R. Vitale, and M. Wahle, On the design of large systems subject to uncertainty, *Journal of Engineering Design*, Vol. **28**, No. 4, 233–254, 2017. Taylor & Francis. https://doi.org/10.1080/09544828.2017.1303664

[11] F. Endress, J. Rieser, and M. Zimmermann, On the Treatments of Requirements in DfAM: Three Industrial Use Cases, *Proceedings of the Design Society*, Vol. **3**, 2815-2824, 2023. https://doi.org/10.1017/pds.2023.282

[12] K. Maute and E. Ramm, Adaptive topology optimization, *Structural Optimization*, Vol. **10**, No. 2, 100–112, 1995. https://doi.org/10.1007/BF01743537

[13] I. Y. Kim and B. M. Kwak, Design space optimization using a numerical design continuation method, *International Journal for Numerical Methods in Engineering*, Vol. **53**, No. 8, 1979–2002, 2002. https://doi.org/10.1002/nme.369

[14] K. Swartz, D. A. Tortorelli, and K. A. James, Optimal Design of Multi-Body Mechanisms Using Layered Connectivity Parameterization (LCP), *2018 Multidisciplinary Analysis and Optimization Conference*, Atlanta, Georgia, USA, June 2018. American Institute of Aeronautics and Astronautics. https://doi.org/10.2514/6.2018-4057

[15] K. E. Swartz and K. A. James, Gaussian Layer Connectivity Parameterization: A New Approach to Topology Optimization of Multi-Body Mechanisms, *Computer-Aided Design*, Vol. **115**, 42–51, 2019. https://doi.org/10.1016/j.cad.2019.05.008

[16] J. Sun, Q. Tian, H. Hu, and N. L. Pedersen, Topology optimization of a flexible multibody system with variable-length bodies described by ALE–ANCF, *Nonlinear Dynamics*, Vol. **93**, No. 2, 413–441, 2018. https://doi.org/10.1007/s11071-018-4201-6

[17] J. Sun, Q. Tian, H. Hu, and N. L. Pedersen, Simultaneous topology and size optimization of a 3D variable-length structure described by the ALE–ANCF, *Mechanism and Machine Theory*, Vol. **129**, 80–105, 2018. https://doi.org/10.1016/j.mechmachtheory.2018.07.013

[18] O. Ambrozkiewicz and B. Kriegesmann, Simultaneous topology and fastener layout optimization of assemblies considering joint failure, *International Journal for Numerical Methods in Engineering*, Vol. **122**, No. 1, 294-319, 2021. https://doi.org/10.1002/nme.6538

[19] H. Nori, S. Jenkins, P. Koch, and R. Caruana, Interpretml: A unified framework for machine learning interpretability, *arXiv preprint arXiv:1909.09223*, 2019. https://arxiv.org/abs/1909.09223

[20] InterpretML: A unified framework for machine learning interpretability, *InterpretML Documentation*, available at: https://interpret.ml/docs/, accessed on August 29, 2024.

[21] K. Svanberg, The method of moving asymptotes—a new method for structural optimization, *International Journal for Numerical Methods in Engineering*, Vol. **24**, No. 2, 359-373, 1987. https://doi.org/10.1002/nme.1620240207

[22] K. Liu and A. Tovar, An efficient 3D topology optimization code written in Matlab, *Structural and Multidisciplinary Optimization*, Vol. **50**, No. 6, 1175-1196, 2014. https://doi.org/10.1007/s00158-014-1107-x