Workflow for Statistical Optimization of 3D-Printed Continuous Carbon Fiber Composite Lay-Ups

M. Akermann, 9T Labs AG, michael@9tlabs.com.

1. Abstract

Continuous carbon fiber composite strands can be laid down with a 3D-printer to realize complex fiber paths, which can also alternate for each individual layer of the part. Therefore, the question arises which layer configuration and which combination of certain layers best fulfils the mechanical requirements of the application. To make efficient use of carbon fibers, they need to be oriented in the directions of the forces that are applied to the structure. Finite element simulation allows to calculate stiffness and strengths of such parts with consideration of fiber orientation, layer configuration and layer sequence.

This work shows the potential of using statistical optimization in combination with finite element models to find the best lay-up configurations and lay-up sequence of specific fiber paths in a carbon fiber composite part. The different layers with possible fiber paths are drawn by an engineer based on experience and/or previously performed isotropic topology optimization. The combination of different layers, stacking sequence as well as the number of each individual layer is extremely difficult to assess by an engineer. The mechanical response of highly anisotropic materials for a multi latitude load case, including mechanical coupling terms known in fiber composites, are beyond the imagination of pure intuition and experience, therefore new methodologies to support engineers need to be found. This work employs evolutionary algorithms to close this gap, guaranteeing certain boundary conditions as well as minimizing an objective. The objective can be the lightest design, highest part stiffness, highest part strength or a combination. To demonstrate this advanced approach, a mountain bike rocker arm which connects the seat stays with the frame has been chosen as a demonstrator part, see figure 1. The part is made out of PA12 with 60% carbon fiber volume content in the composite areas as well as with neat PA12 in the plastic areas. *OptiSLang* was used to parametrize a python script which then adjusted the fiber lay-up sequence in *ANSYS Composite Pre* before the model was calculated by *ANSYS Mechanical*. With help of a sensitivity study the most promising permutation of the four different layer designs could be quickly evaluated. This was followed up by an optimization from an Evolutionary Algorithm (EA). The algorithm came up with solutions which fulfilled all the different requirements. This approach proved to be viable and beneficial, especially for parts with a consistent cross-section in printing direction.

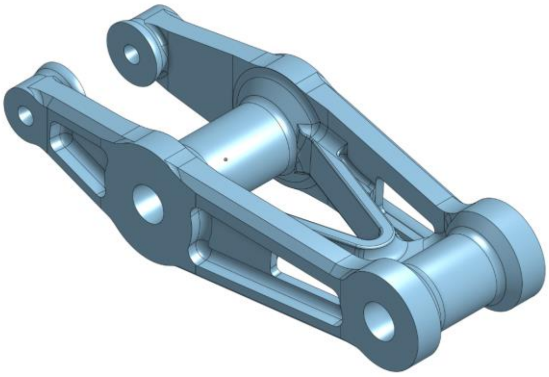


Figure 1: Original Aluminum SCOTT bike rocker

2. Introduction

Additive manufacturing of carbon fiber reinforced polymers (CFRP) using the novel printing technology from *9T Labs* allows to place fibers with an extraordinary high design freedom. This increases the design space of composite parts. Particularly, it enables variable fiber angles and designs that comprise plastic regions exempt of fiber reinforcements. Since almost any fiber path (0.8mm layer width) can be realized within a very thin printed layer (0.216 mm layer height) the question arises, which combination of certain layers best fulfils the mechanical requirements.

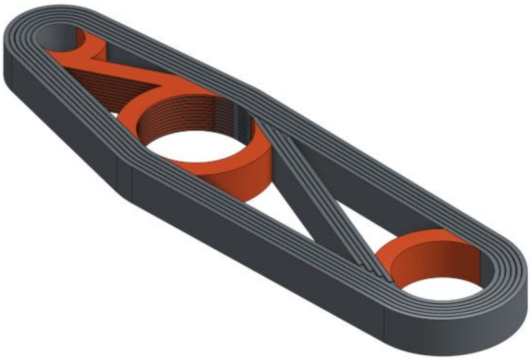


Figure 2: Half of the composite mountain bike rocker with its different composite paths and plastic areas

In order to address this question statistical optimization of a parametrized numerical model is used to analyze the influence of different lay-ups to find ideal lay-up configurations and the layup sequence which fulfill best the boundary conditions according to multiple load cases and optimization objective.

3. Method and Parametrization

Carbon fiber composites consist of fibers embedded in a polymeric matrix material. The fibers are very thin, with diameters of only seven thousandths of a millimeter. The fibers are extremely strong if pulled from the ends, but like a piece of string they don’t hold any load on their own if you try to compress them. Similarly, the resistance is very low if pulled in radial direction. To use fibers in an efficient way there are two main things of importance:

- Fibers should be oriented locally along the loading directions.
- A strong, but also resilient matrix should be used to embed the fibers.

These circumstances need to be considered when the numerical model is set up. Fiber orientations and the associated material anisotropy need to be modelled in the numerical model. Therefore, the local coordinate system of the element needs to be oriented according to the fiber direction, including the defined material model representing the orthotropic material properties. Since a single layer height is only around two tenths of a millimeter one could consider using shell elements. But since many layers are stacked on each other the assumption of neglecting stresses in out of plane direction is not valid anymore. For this reason, layered solid elements are used. This special finite element can represent several layers in thickness direction by adjusting the stiffness matrix according to the fiber direction of the represented layers. The definition of these elements is simplified by *ANSYS Composite Pre (ACP)* which defines all the element properties. Identically to the 3-axis printing process the setup of the composite model in *ACP* is based on a base surface from where the model is built. This surface is split by projections of the composite paths of the different layers in order to represent every single layer design based on one surface, see figure 3. After meshing the split surface selections of the elements are available and the lay-up can be built within *ACP*.

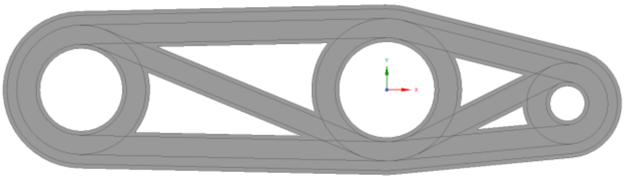


Figure 3: Split surface representing every composite path of the four layers

The *9T Labs* additive manufacturing technology for fiber composite materials consists of a build module to gradually lay down fiber strands according to predefined trjectories. The head is capable of manufacturing hybrid parts with fiber composite and mere plastic sections by using the Omnidirectional Filament Placement (OFP) technology and conventional plastic printing technology.

One of the major differences of the OFP system compared to conventional FDM-printing is that the continuous carbon fiber filament needs to be cut after a strand has been laid down.

The mountain bike rocker arm application consists of a constant cross-section in thickness direction. In order to allow to interchange the different layers randomly every layer needs to fulfil the projected area of the bike rocker. Every layer consists of predefined composite paths and plastics areas. For the definition in *ANSYS ACP* this means a combination of multiple plies based on the element selection sets for each layer. For the parametrization the plies need to be addressed layer-wise. By default, *ANSYS ACP* allows the parametrization of single plies but does not have the functionality to address a group. For this reason, a python script sets up the whole layer sequence and orders the layers according to a permutation vector, see figure 4. Another vector defines the immediate repetition of a single layer whereas the variable repetition repeats the whole layer sequence defined by the two mentioned vectors, see figure 5.

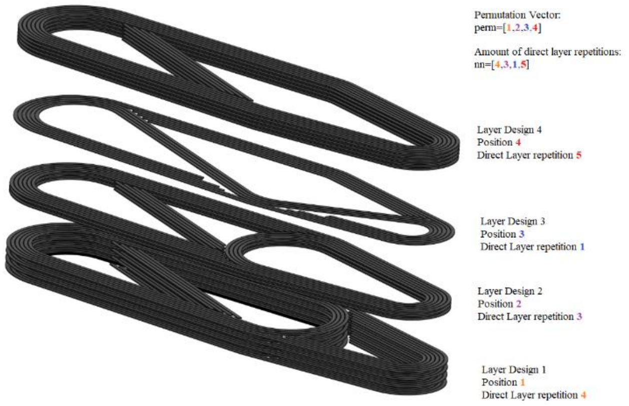


Figure 4: Exemplary layer sequence for perm[1,2,3,4] and nn=[4,3,1,5]

The vector $perm$ defines the order of a four-layer design sequence. The numbers in the vector sequentially define the position of each layer design inside the sequence. In addition, the vector nn accordingly defines the direct repetitions within the four-layer design sequence. The variable r defines the amount of repetitions of the four layer design sequence defined by the vectors $perm$ and nn. In order to make this approach available to other parts with uniform cross-sections in printing direction a unified nomenclature for the script is used. The finite element selections which are based on the split surface as shown in figure 3, are called plies within *ACP*. A layer consists of multiple plies according to its geometry. Plies are made out of oriented selection sets where the element set are adresses and the fiber orientation defined. These sets are named by its accordance to the design layer e.g. L1 followed by 'cf' or 'pl' for plastic or composite and end with a unique number e.g. 'L2_cf_1'. The python script builds up the modelling group according to the input vectors by defining plies for each layer by their corresponding oriented selection. Members which occur at every layer are identified by oriented selection sets starting with 'cf' or 'pl' followed by every layer e.g. 'cf_everylayer_1'. They are considered and put in each ply group for every layer. The input variables for *ACP* are parametrized by *OptiSLang* by manipulating the values directly within the python script which is executed every single run. Figure 5 shows a defined a defined layer sequence and a beginning repetition within *ANSYS ACP*. Except from the part weight and part thickness the responses for the optimization are coming from *ANSYS Mechanical* and are results from a finite element calculation. The part thickness is calculated within *OptiSLang* based on the input vector nn and variable r and the defined layer thickness. The weight is handed over from *ACP* as a response variable and considers the different weight and the occurrence of the design layers. The optimization was run within *OptiSLang* and *ANSYS* was called in batch mode to solve the model. For this reason, a second python script was set up which feeds in the script for *ACP* with the varying parameter and actualizes the *ANSYS Workbench* project. A challenge to define the range for the input variables for the vector $perm$ was to ensure that the optimizer does not choose identical numbers within the vector, which consist of four variables. A position number needs to be unique within the vector $perm$ if not *ACP* would then simply search the next free space for those plies which have an already forgiven number.

For the implementation, rather than making the optimization too complex, the optimization is split into two parts:

1. Sensitivity study: Analyzing the influence of permutations.
2. Optimization: Adjusting occurrence and repetition of certain design layers by a given permutation.

For analyzing the influence of the permutation in a sensitivity study, a nested system is used within *OptiSLang*, see figure 6. The inner system, called "Replace constant parameter" shown in figure 6, defines the start design configurations by defining the vector $perm$ which are then calculated for every design given from the outer system which defines the vector nn, direct repetitions of layers within a layer sequence and the vector r, which defines repetions of the whole layer sequence.

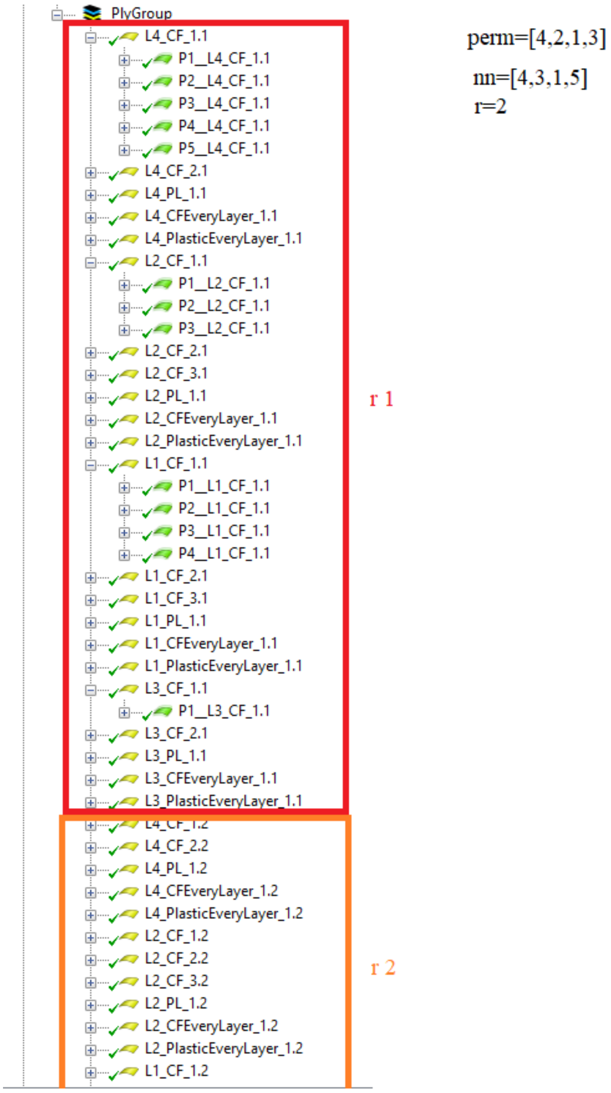


Figure 5: Modelling group within ACP with its plies after execution of the python script with perm=[4,2,1,3], nn=[4,3,1,5] and r=2

In order to address every possible permutation each one is defined as a start design, in the inner system. In case of the rocker this is factorial of four which leads to twenty-four permutations. The result of the outer system will then always be a vector of twenty-four values. In order to identify promising permutations according to defined criteria further post processing is done by the function blocks *Data Mining* and the *Python Script Block* from *OptiSLang*.

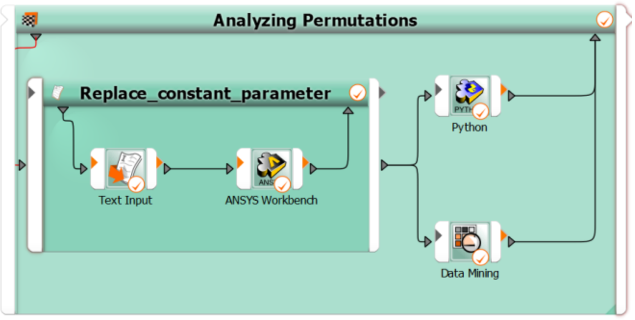


Figure 6: Nested system within OptiSLang for analyzing influence of permutations

The block *Data Mining* gives the possibility to filter out certain values from the vector, like the maxima. With help of python scripting further criteria are defined and checked with respect to the design IDs from the permutations. The idea of this pre-study, where all the possible permutation have been analyzed at certain thicknesses, is to understand the behavior of the model in a better way before an optimization is started based on the results from this sensitivity study. The permutation which was identified as the most promising according to defined criteria like stiffness is taken as a fixed permutation for the optimization by the evolutionary algorithm. The parameters for the optimization are vector nn , direct layer repetitions within a layer sequence and r which are the repetitions of the whole layer sequence. The range of the parameters within the vector nn is defined in a way that the algorithm has the possibility to deactivate a layer by defining zero as values as well as making a single layer sequence by only one layer by defining the amount of direct repetitions equal to the number of plies. In case of the rocker this would be four. The variable r is set to a range that the thickness ends in the max tolerable thickness of the part by multiplication the amount of design layer and layer thickness. This leads to $r_{max} = \frac{t_{part\,max}}{h_{layer}*n_{design\,layer}}$ where r_{max} is the maximal value for repetitions, h_{layer} the constant layer thickness, $t_{part\,max}$ the maximal defined part thickness and $n_{design\,layer}$ the amount of design layers. In the dimensioning of composite parts most often more than one load case needs to be considered. In order to avoid a multi-objective optimization a single objective is made from responses of all the load cases. Additionally, the individual load cases are addressed by boundary conditions.

4. Load Case and Parameter Definition

Half of the rocker (symmetry condition) is analyzed for three different load cases based on the standard *safety requirements for bicycles (ISO 4210).* The first two load cases called *Pedal 1* and *Pedal 2* represent the load when the bicycle is loaded on one single pedal. The difference of the load cases *Pedal 1 and Pedal 2* is the force direction..

The third load case called *Jump* represents the drop of the bike on the back wheel. In the optimization only half of the rocker is calculated as the composite rocker will consist of two identical parts connected by composite pipe. The loads assumed for Pedal 1 and 2 is 500 N out plane at hole B (Figure 8). Since only half of the rocker is calculated half of the load is applied. The rocker is fixed at location A and the load at hole C is adjusted to put the momentum at A in equilibrium. For the load case jump a Force of 2000 N is assumed. With respect of the half of the rocker 1000 N is applied and 591 N to compensate the momentum at A (Figure 7).

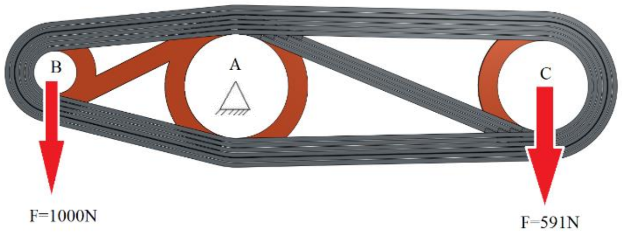


Figure 7: Load case 3 Jump

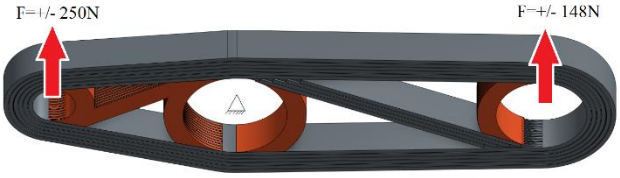


Figure 8: Load case Pedal 1 and 2

As responses from the finite element calculation the following results are considered for the optimization:

- Deformation for each load case according to the force direction.
- Max principal stress for each load case.
- Puck inverse reserve factor for each load case.

The load case *Jump* is considered as the critical and major loading. For this reason the deformation is minimized for both holes B and C:

obj = min (abs(dir_deformation_B+dir_deformation_C))

As strength criteria the Puck hypothesis is chosen. For each load case a specific safety factor can be defined:

$$InverseReserveFactor_{Jump} \leq 0.8$$

$$InverseReserveFactor_{Pedal1/2} \leq 0.9$$

In order to avoid that the part thickness in unnecessarily increased minimal and maximal principal stresses for the load case *Jump* are defined.

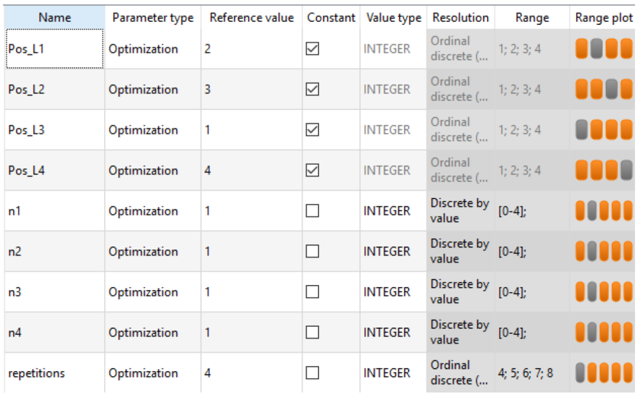
$$Max\ Principal\ Stress_{Jump} \geq 120\ MPa$$

In order to further reduce the design space, the minimal thickness of the part is defined to a minimum of 3 mm and maximal thickness of 7 mm

$$(n1 + n2 + n3 + n4) * 0.216 \leq 7\ mm$$

$$(n1 + n2 + n3 + n4) * 0.216 \geq 3\ mm$$

The input parameters for the vector nn are defined based on the approach as described previously at the number of design layers and maximal thickness of the part. In order to allow the algorithm to simulate a part of only one single design layer all the layers need at least a range from 0-4 and the variable repetitions needs to multiply 4 times 8 to get a maximal part thickness of 7 mm (figure 9).



Name	Parameter type	Reference value	Constant	Value type	Resolution	Range	Range plot
Pos_L1	Optimization	2	☑	INTEGER	Ordinal discrete (...	1; 2; 3; 4	🟧⬜🟧🟧
Pos_L2	Optimization	3	☑	INTEGER	Ordinal discrete (...	1; 2; 3; 4	🟧🟧⬜🟧
Pos_L3	Optimization	1	☑	INTEGER	Ordinal discrete (...	1; 2; 3; 4	⬜🟧🟧🟧
Pos_L4	Optimization	4	☑	INTEGER	Ordinal discrete (...	1; 2; 3; 4	🟧🟧🟧⬜
n1	Optimization	1	□	INTEGER	Discrete by value	[0-4];	🟧⬜🟧🟧🟧
n2	Optimization	1	□	INTEGER	Discrete by value	[0-4];	🟧⬜🟧🟧🟧
n3	Optimization	1	□	INTEGER	Discrete by value	[0-4];	🟧⬜🟧🟧🟧
n4	Optimization	1	□	INTEGER	Discrete by value	[0-4];	🟧⬜🟧🟧🟧
repetitions	Optimization	4	□	INTEGER	Ordinal discrete (...	4; 5; 6; 7; 8	⬜🟧🟧🟧🟧

Figure 9: Parameter range for Evolutionary Algorithm with direct solver calls

5. Results and Outlook

The constant values for the layer positions are based on the sensitivity study where all the permutations have been calculated through. For the sensitivity study the amount of direct layer repetitions was constant so every layer occurs the same amount of times within the lay-up. After first trials with varying direct repetitions this ended up with an enormous amount of solver calls. Since this optimization problem is very discrete generated Metamodels are only limited usable for optimizations. For this reason, all the permutations were simply calculated through at one thickness based on this one permutation was chosen for the optimization.

At a constant thickness it could be shown that different permutations vary the stiffness in direction of the load case *Jump* at around 5% and the Max Principal Stress almost 10% (figure 10).

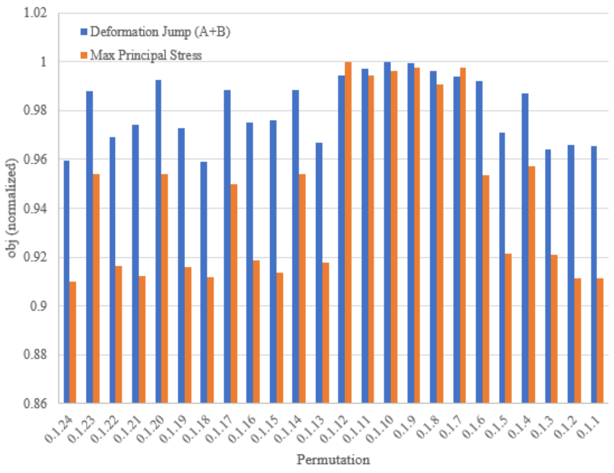
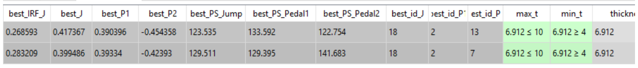


Figure 10: Deformation and max principal stress for every possible permutation with constant immediate repetitions(vector nn) and fixed repetitions (variable r)

Based on this small analysis permutation 18 was chosen in order to analyze the amount of direct layer repetitions and stack up repetitions. Similarly, the python script identified the permutation IDs based on programmed criteria for each load case without any additional data handling, shown in figure 11.



best_IRF_J	best_J	best_P1	best_P2	best_PS_Jump	best_PS_Pedal1	best_PS_Pedal2	best_id_J	›est_id_P1	est_id_P	max_t	min_t	thickn
0.268593	0.417367	0.390396	-0.454358	123.535	133.592	122.754	18	2	13	6.912 ≤ 10	6.912 ≥ 4	6.912
0.283209	0.399486	0.39334	-0.42393	129.511	129.395	141.683	18	2	7	6.912 ≤ 10	6.912 ≥ 4	6.912

Figure 11: Results from permutation analysis based on post process PYTHON scripted criteria

The evolutionary algorithm converged after 218 direct solver calls. The figure below with the objective history shows that the criteria leaves a large window open between 0.4 mm and 0.2 mm. The algorithm chose, due to the criteria for minimizing the objective, the stiffest and therefore also one of the heavier configurations. Here, this is optimization design 122, shown in figure 12.

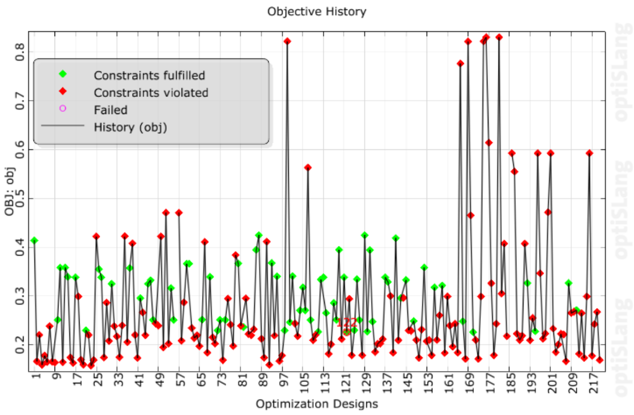


Figure 12: Objective history (deformation at load case Jump) of converged Evolutionary Algorithm

The *Criteria Data* in figure 13 show that the criteria which increases the usage factor of the part prevents the algorithm to simply define the thickest version.

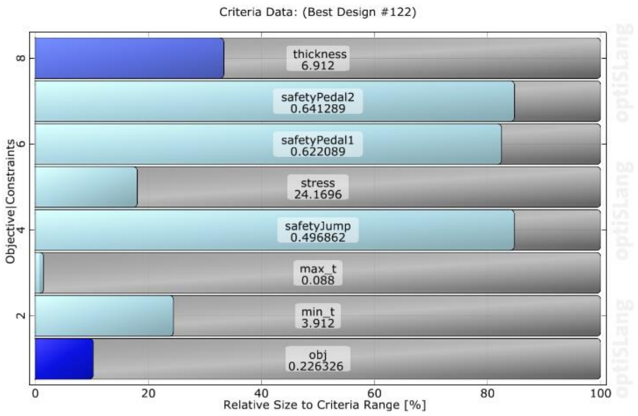


Figure 13: Criterial data of best design according to defined constraints and objective

The minimal and maximal principal stress in the load case Jump is chosen rather low compared to the inverse reserve factors. If we look at the objective with the highest value, but still valid constraint, we see that 3.456 mm part thickness seems to be the version with lowest number of layers which still fulfils the constraints, see figure 14.

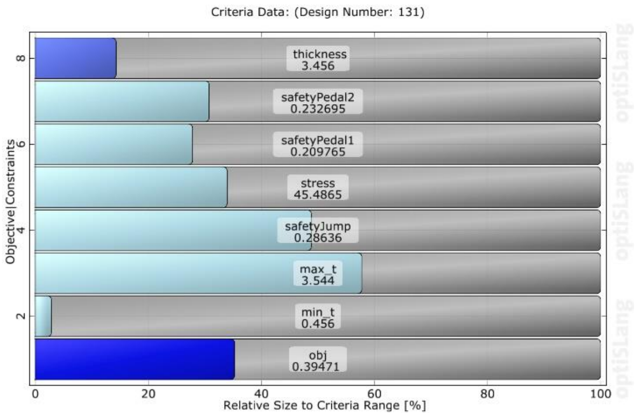


Figure 14: Criteria data from design with valid constraint and largest objective (highest deformation)

The results give a good understanding of the part stiffness and strength for all three different load cases. However, in order to find a more specific solution it is beneficial to unify the objective with the response weight or use the inverse reserve factor to utilize the capable load potential of the part more. Nevertheless, the workflow clearly shows how a vast number of layers can be optimized in a systematic way for a part with defined boundary conditions. This workflow has already been applied successfully on an industrial part with six design layers and more than two hundred layers. This year in late summer 9T Labs will release its Fiber Design Suite within ANSYS Space Claim which will give further possibilities to enhance this optimization approach by coupling the direct modeler of the geometry to this workflow.

6. Acknowledgements

The author would like to thank *9T Labs* for supporting this project. Additionally, Mr. Andreas Grosche from *Dynardo* who supported the development of this workflow.

About the Author

Michael Akermann is Lead Mechanical Engineer at *9T Labs* and in charge of finite element simulations. He pursued his M.eng. in Applied Computational Mechanics in the esocaet program at Technische Hoschule Ingolstadt