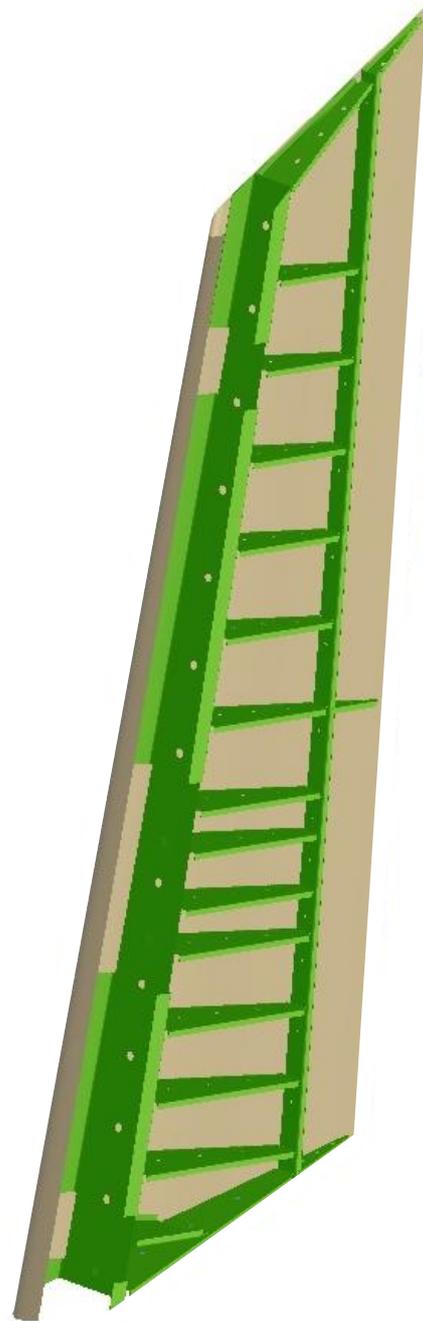


Assessment of design optimization strategies on a composite rudder using a knowledge-based engineering tool

K.B. van Zomeren, B.Sc.

Delft University of Technology





Confidential



Assessment of design optimization strategies on a composite rudder using a knowledge-based engineering tool

By

B.Sc. K.B. van Zomeren

in partial fulfilment of the requirements for the degree of

Master of Science

in Flight Performance and Propulsion
at the Delft University of Technology,
to be defended publicly on Monday June 26, 2017 at 9:30 AM.

Supervisors:	ir. I. van Gent	TU delft
	ir. J. Baan	Fokker Aerostructures
Thesis committee:	dr. A. Gangoli Rao	TU Delft
	dr. ir. G. La Rocca	TU Delft
	dr. ir. J van Campen	TU Delft

Thesis number: **135#17#MT#FPP.**

This thesis is confidential and cannot be made public until June, 2022.



Preface

Before you lies the final report of the master thesis conducted at the faculty of Aerospace Engineering at Delft University of Technology. The research was performed in collaboration with Fokker Aerostructure, located in Papendrecht. The objective was to increase product quality of a rudder while reducing the development lead time, by automating the design process such that it supports optimization. The objective of ITEA2 project *‘Integrated and Distributed Engineering Services framework for MDO’* (IDEalisM) proved to be in line with the goals of this thesis. Hence, part of the work of this thesis has been performed in the scope of this project.

*K. B. van Zomeren
Delft, June 2017*



Acknowledgements

A long time has passed since I started my bachelor at the TU Delft. Getting to the point where I am today would not have been possible without the support of many people. I would like to use this opportunity to express my gratitude to everyone who has supported and guided me throughout my studies and this research.

First and foremost I would like to thank my supervisor at the TU Delft, Imco van Gent, for his continuous support and patience to guide me through this research, reviewing my work, and keeping me on the right track. My sincere thanks also go to my supervisors at Fokker Aerostructures, Jan Baan and Tobie van den Berg, for providing me with this opportunity and allowing me to follow my own path but guiding me when necessary in between their busy schedules.

I would like to thank the initiator of my thesis, Reinier van Dijk, for presenting me with the opportunity to perform this thesis, as well as, sharing his Python-based KBE system ParaPy and the support on the workings of it. Many thanks go out to my colleagues at Fokker Aerostructures, with whom I spent numerous hours in interviews in which they shared their knowledge on rudder design. I would like to thank them for their patience, availability, and willingness to share knowledge which made performing this thesis possible.

I want to show my gratitude to my friends which supported me throughout the research. A special thanks goes out Fons van der Beek for providing me with valuable feedback on my English writing, helping to increase the readability of this report. Last, but certainly not least, I would like to thank my family, my parents, Mieke and Johan, and my siblings for their love, support and motivation.

My gratitude goes out to all of you, thanks!

*Kailey
Delft, June 2017*



Summary

Tier 1 suppliers of aircraft components, such as Fokker Aerostructures, have to distinguish themselves to keep up with the increasing competitiveness by improving the design lead of a rudder while increasing the product performance. An investigation has shown that, at the moment, this is not possible in the current design process. The time required to design one rudder structure is around seven to eight months. This is a problem in itself, which is exacerbated by the fact that it prevents engineer from checking many and perhaps uncommon designs. This means that the current approach does not support optimization and only three to five designs are analysed within two to three years' time. Additionally, as a side effect of the long design lead time, the rudder design starts with many uncertainties of the design requirements. Therefore, engineers are forced to make assumptions for these uncertainties, which often lead to less optimal rudder product variants. Hence, to gain a competitive edge, a radical rethinking of the design process is required.

This research proposes a novel methodology for the design of composite rudders, with the goal to design better rudder structures faster. This methodology is inspired on the front-loading strategy supported by automated design methods. Front-loading is defined as a strategy to enhance the product development process by reducing design lead time and increasing product performance. For this research, front-loading means developing engineering knowledge in the earliest stage, before the actual design process starts. It is expected that this can be achieved by capturing engineering knowledge and implementing this in Knowledge-Based Engineering (KBE) applications to rapidly evaluate product variants. However, the rapid evaluation of the product variants alone is not sufficient to gain a competitive edge in the market. The design space of a rudder is immense, as a result, one cannot simply analyse all the possibilities, meaning a more strategic manner is required to search the design space.

To this purpose, the Knowledge-based Rudder and Interface design Optimization System (KRIOS) was developed, which automates and optimizes the composite rudder design. KRIOS can quickly and automatically perform the detailed synthesis of a composite rudder. Hence, it allows for quick and accurate analysis of the weight and cost. Additionally, KRIOS automatically analyses the rudder for different load cases. Therefore, it can quickly reflect the effect of changes made in the rudder structure. As a result, it can design a high fidelity rudder variant within a few minutes, which supports the implementation of optimization.

Different optimization strategies were wrapped around KRIOS, to satisfy both goals of reducing design lead time and increasing product performance. The optimizations strategies are divided into four main building blocks: Parameterization, Integration, Post-process, and Algorithm, called the PIPA. KRIOS uses the knowledge from engineers and previous rudder designs to determine the PIPA setting, in order to give the tool as much freedom and try out a large variety of product variants while still being able to find an optimum within a limited amount of time. Hence, it ensures that the scope of KRIOS is not limited to few common

design variants. Using different PIPA settings, one of the first strategies implemented in KRIOS was able to find an optimum within five days by analysing around 1950 rudder products, whereas subsequent strategies showed an even further reduction to 1150 evaluations within 2.5 days. This research has shown the benefits of KRIOS and how it provides a first step into the assessment of higher fidelity product models while increasing the scope and helps to design better rudders faster.



Table of contents

List of figures.....	x
List of tables.....	xii
Nomenclature	xiii
1 Introduction.....	1
1.1 Front-loading	1
1.2 Front-loading approach at Fokker Aerostructures	3
1.3 Research objectives	6
1.4 Document structure.....	7
2 The need for a new methodology	8
2.1 The current design methodology and design requirements	8
2.1.1 Step 1: Project definition	9
2.1.2 Step 2: Rudder synthesis	9
2.1.3 Step 3: Rudder structural analysis.....	10
2.1.4 Step 4: Margin of safety convergence loop	10
2.1.5 Step 5: New rudder configuration or accept final design	11
2.2 Limitations of the current design methodology.....	12
2.3 Material selection procedure at Fokker Aerostructures.....	12
2.4 Challenge of full external loads analysis	14
2.5 Relevant work in open literature.....	15
2.6 The challenges that lie ahead.....	18
2.7 The scope of this research.....	19
Part I – Automated rudder design method.....	20
3 Introduction to rudder design	21
3.1 The fin.....	22
3.2 The fin-rudder-interface.....	23
3.3 The rudder.....	24
3.3.1 The rudder skin	24
3.3.2 The ribs of a rudder.....	27
3.3.3 The spars of a rudder.....	28
4 The Knowledge-based Rudder & Interface design Optimization System (KRIOS) ...	29
4.1 Methodology behind the model.....	29
4.2 The main structure of the Rudder Generator	30
4.2.1 Input handler module	31
4.2.2 Hinge system module.....	35
4.2.3 Load cases generation module.....	36
4.2.4 Rudder structure generation module.....	37
4.2.5 Structural analysis module.....	42
4.2.6 Weight and cost estimation module.....	43
4.2.7 Output collection module.....	46
4.3 Validation	47
Part II – Rudder optimization methodology	51
5 The developed methodology for optimization	52
5.1 Parameterization	52
5.2 Integration.....	53
5.3 Post-processing	55
5.4 Algorithm selection	56

5.5 Criteria for the different strategies.....	57
6 The optimization strategies	59
6.1 Strategy A: Response of the model using directive search.....	59
6.1.1 Implementation	59
6.1.2 Results.....	60
6.2 Strategy B: Higher fidelity optimization with a differential evolution algorithm	62
6.2.1 Implementation	63
6.2.2 Results.....	66
6.3 Strategy C: Validation of optimum and algorithm settings	67
6.3.1 Implementation	67
6.3.2 Results.....	68
6.4 Strategy D: Tailored skin parameterization	72
6.4.1 Implementation	73
6.4.2 Results.....	74
6.5 Strategy E: Quadratic skin distribution parameterization.....	75
6.5.1 Implementation	76
6.5.2 Results.....	78
6.6 Strategy F: Improved quadratic skin distribution parameterization	78
6.6.1 Implementation	79
6.6.2 Results.....	80
6.7 Comparison of the different strategies.....	81
7 Conclusion and recommendations.....	84
7.1 KRIOS, the automated design method	84
7.2 Assessment of rudder optimization strategies using KRIOS.....	85
Appendix A. Overview of contribution to the RG	90
Appendix B. Detailed explanation of limitations of the FE model and their effects.....	92
Appendix C. Detailed explanation of the face-grouping functionality	95
Appendix D. Additional margin of safety visualization for increasing Safety factor	98
Appendix F. Detailed explanation of the average stopping length calculation	99

List of figures

Figure 1: Effects of implementation of KBE (adapted from Stokes [4]).....	3
Figure 2: Traditional, current, and proposed future product development process with front-loading implemented [8].	4
Figure 3: Activity diagram of the Knowledge-based Rudder & Interface design Optimization System (KRIOS)	5
Figure 4: Overview of the chapters of this thesis	7
Figure 5: Workflow of the rudder design process at Fokker	8
Figure 6: Stacking sequence visualization of the "8_spar" laminate	13
Figure 7: Distribution of scope versus fidelity of previous optimization approaches and the goal of this thesis	18
Figure 8: Partial structural breakdown of the vertical tail	21
Figure 9: Schematic overview of the fin, interface system and rudder.	22
Figure 10: Schematic overview of a partial structural breakdown of a rudder.....	25
Figure 11: Definition of skin sub-sections. (A: skin-rib-interface, B: skin-spar-interface, C: combined case).....	25
Figure 12: Criteria for material selection in skin pockets.....	26
Figure 13: Criteria for material selection of spar.....	28
Figure 14: Overview of the main modules of the Rudder Generator	30
Figure 15: Standardization of different inputs formats and the required output.....	31
Figure 16: Results of example input geometry from different input files	32
Figure 17: Example of rib distribution function from zero to three additional ribs	34
Figure 18: Visualization of constraints, forces and pressure distribution for two load cases..	37
Figure 19: Placing of the front and rear spar datums and the corresponding EoP datums	38
Figure 20: Effects of the End of Part (EoP) planes on the actual rib size.....	39
Figure 21: Overview of the required steps for a high fidelity rib geometry	40
Figure 22: Example of allowable envelope	41
Figure 23: Example of optimal shape position when on area used and area wasted when	44
Figure 24: Example laminate distribution on a skin and the corresponding ply shapes.....	45
Figure 25: Margin of safety visualization of the two load cases from Figure 18	47
Figure 26: Deviation of the true and estimate OMLs for different views at different intersections	48
Figure 27: Rudder kick case due to crosswind during approach	49
Figure 28: Displacements (white mesh) of a rudder in PATRAN for a rudder kick load case	50
Figure 29: Different skin parameterizations and their number of possibilities.	53
Figure 30: Different ways of tools integrations	54
Figure 31: Flow diagram of the workflow used in Optimus.....	55
Figure 32: Results from the manual directive search.....	61
Figure 33: The discrete steps in the margin of safety of the additional ribs for increasing number of ribs	61
Figure 34: Definitions for the new parameterization	62
Figure 35: Example of new parameterization of the skin material allocation.	65
Figure 36: Normalized weight per iteration for the higher fidelity optimization	67
Figure 37: Success rate and number of evaluations required for DE's with different population sizes. Here N is the number of variables	69
Figure 38: Number of evaluations required for different DOE methods.....	71
Figure 39: Skin parameterization strategy B (left) versus new parameterization (right)	73

Figure 40: Skin thickness distribution for found optimum with tailored skin parameterization	74
Figure 41: Expected moment distribution on the rudder	75
Figure 42: Expected thickness distribution for the skin according to engineers	75
Figure 43: Control points for quadratic skin thickness distribution lines	77
Figure 44: Discretization of the quadratic skin distribution	77
Figure 45: Visualization of the same skin thickness distribution for different design points (CP = 7.5 and CP = 7.1)	78
Figure 46: Effect of numbers of steps on the number of skin thickness distributions	79
Figure 47: Normalized weight per iteration for the quadratic skin thickness distribution.	80
Figure 48: Normalized weight and number of function evaluations per strategy	81
Figure 49: Final scope versus fidelity level of KRIOS and its aim, adjust from Figure 7.	85
Figure 50: Unstructured (left) versus structured mesh (right)	92
Figure 51: Mesh without corner cutout (left) and with corner cutout (right)	93
Figure 52: Effects of a simplified bracket FE model	94
Figure 53: First four steps and explanation of face grouping	95
Figure 54: Steps five to eight of face grouping	96
Figure 55: Steps nine and ten of face grouping	97
Figure 56: Resulting different sections of face grouping	97
Figure 57: Margin of safety visualization for increasing aerodynamic loads	98

List of tables

Table 1: Time requirement for FE model generation or adjustments at Fokker Aerostructures [10].....	10
Table 2: Topology requirements of the rudder	11
Table 3: Example ply library for skin (left) and spar (right)	13
Table 4: Possible failure cases for two hinge failures.....	15
Table 5: Summary of the analysis requirements of the rudder	15
Table 6: Visualization of scope and fidelity	16
Table 7: List of selected load cases.....	19
Table 8: Effect of adding ribs on the rib pitch per bay	33
Table 9: Resulting rib material dictionary as a result of different design vectors	35
Table 10: Effect on computation time for rib mass for different level of details	44
Table 11: Weight comparison between the simulate Rudder X and the actual Rudder X.....	48
Table 12: Trade-off table template for the different optimization strategies.....	58
Table 13: Selection of the new parameterization.....	65
Table 14: Selected variables for the skin distribution line.....	77
Table 15: Number of plies per section for different CP values	78
Table 16: Trade-off table of the optimization strategies.....	82
Table 17: Contribution of the author to the Rudder Generator.....	90
Table 18: Normalized delta between different design points and the most optimal point	100

Nomenclature

List of abbreviations

AI	=	Artificial Intelligence
BB	=	Box Behnken
CAD	=	Computer Aided Design
CAE	=	Computer Aided Engineering
CotS	=	Commercial of the Shelf
DE	=	Differential Evolution
DoE	=	Design of Experiments
DV	=	Design Vector
EoP	=	End of Part
ES	=	Exhaustive Search
FEA	=	Finite Element Analysis
FEM	=	Finite Element Method
FF	=	Full Factorial
GA	=	Genetic Algorithm
GB	=	Gradient Based
HDOT	=	Hinge-system Design and Optimization Tool
HL	=	Heavily Loaded
IML	=	Inner Mold Line
KBE	=	Knowledge-Based Engineering
KRIOS	=	Knowledge-based Rudder & Interface design Optimization System
LE	=	Leading Edge
LHC	=	Latin Hyper Cube
MDO	=	Multidisciplinary Design Optimization
MMG	=	Multi-Model Generator
MS	=	Margin of Safety
NASTRAN	=	NASA STRucture Analysis
NL	=	Normally Loaded
OEM	=	Original Equipment Manufacturer
OML	=	Outer Mold Line
PID	=	Ply Identifier

PIDO	=	Process Integrations and Design Optimization
PIPA	=	Parameterization, Integration, Post-processing & Algorithm
RiaM	=	Rudder in a Month
RG	=	Rudder Generator
STEP	=	Standard for the Exchange of Product model data.
TE	=	Trailing Edge
VT	=	Vertical Tail

List of symbols

λ	=	Population size
h	=	Height
N_{ribs}	=	Number of ribs
N	=	Number of plies
t	=	Thickness
W	=	Weight

1

Introduction

Tier 1 suppliers of aircraft components are challenged by an increasing competitiveness due to market globalization. This is driving companies and their engineers to come up with more cost-effective and shorter development processes [1]. While this is a challenge in itself, other requirements concerning safety and sustainability are also becoming more and more important, leading to increased complexity of aircraft systems [2]. Addressing all these problems would mean that engineers need to take into account more details in less time. However, the current design approach is characterized by a long design lead time and does not support automated design methods very well. As a result, it is not well-suited to cope with these challenges.

Accompanying these challenges is the brain drain phenomenon. Brain drain is the loss of “tribal knowledge” gained by engineers over decades of researching or designing a specific topic or aircraft component. This loss comes from the aging of these highly educated engineers which are about to retire. Currently, the average age in the aerospace industry is 57 and as a result, e.g. at Boeing about 20% of the work force (and their knowledge) is eligible for retirement. Although experts say many companies are ignoring the threat, others are aggressively seeking solutions (Velocci, Aviation week magazine 2013).

The research presented in this thesis was a combined effort between Fokker Aerostructures and the TU Delft to propose a novel methodology for the design of composite rudders. This methodology should increase Fokker's competitiveness and reduce the impact of the brain drain effect. This methodology is inspired on the front-loading strategy supported by automated design methods.

1.1 Front-loading

Front-loading is a strategy to enhance the product development process by reducing design lead time and increasing product performance. Here, the development process is the transformation of a market opportunity (the customer's request) to an actual sellable product. Front-loading is defined, by Thomke and Fujimoto [3], as: *"a strategy that seeks to improve development performance by shifting the identification and solving of (design) problems to earlier phases of a product development process."* The main focus of this strategy lies upon identifying and solving problems in an earlier stage. According to Stokes [4], there are two main approaches to enable front-loading:

- **Project-to-project knowledge transfer:** Storing of problems and their solutions for similar design processes or projects, to avoid solving the same problem twice;

- **Rapid problem-solving:** Usage of advanced technologies to quickly (and automatically) develop and analyse virtual prototypes instead of real life physical testing.

The transfer of knowledge between projects is necessary to access the full potential of rapid problem solving. Usage of advanced technologies, e.g. a Finite Element Method solver such as NASTRAN, has been replacing the real life testing during the current design approach and reduced the time required for analysis tremendously. However, this only presents a small part of design process, pre-processing and analysis of the results still consume a large amount of time and manual labour. Therefore, if possible, a full automation of the design process is necessary to reduce evaluation times even further. Automation of the design requires that all the design requirements, steps, and decisions made during the design process are known. This can be achieved by capturing the knowledge of previous projects and formalising this knowledge such that it can be used to automate a design process.

Capturing of knowledge can be done in many ways, but already introduces some difficulties. The knowledge from previous projects can be obtained by, e.g. interviewing engineers or studying project documentations, e-mails, or even forums. These can contain knowledge about the design requirements that should be taken into account and the steps taken along the way and decisions. According to Milton [5] capturing this knowledge already proposes difficulties:

- It could be difficult for an engineer to express his expertise;
- At the same time, the knowledge engineer has to understand, document, and keep the interview going.

The next step is the translation of the captured knowledge into formal knowledge, by structuring it in a consistent, neutral format. This ensures the knowledge can be used to create, e.g. Knowledge-Based Engineering (KBE) applications, which can (partly) automate the design processes. The concept of KBE is discussed in more detail in Section 4.1.

Implementation of rapid problem solving, using KBE, has both strengths and weaknesses. The rapid development and analysis of product variants (designs), allows the engineers to consider many designs, including uncommon design which are usually avoided due to time pressure. In addition, automating the non-creative tasks gives the engineers more time to spend on the creative-innovative processes while still reducing the overall design lead time, as is illustrated in Figure 1. However, according to Stokes [4] and La Rocca [6] KBE cannot be used all the time or is even to be avoided at times, due to the following reasons:

- The task is too straightforward and non-repetitive, and can be performed using less resources;
- It consist of many creative design processes and the design process is highly subjected to change;
- Organizations tend to use existing Commercial Off The Shell (COTS) tools, as they are a cheaper alternative cost and carry less risk.

Often it is known whether the design process is too straightforward, non-repetitive or contains mostly creative steps, so the main challenge lies in identifying if developing such a tool is worth it. This is because building a KBE application can be very expensive, as it can

cost more resources compared to doing one design using COTS tools. Additionally, it introduces risks, e.g. whether it is even possible to build such an application and how to ensure that the results are valid. As it is difficult to predict these cost and risks of developing such a tool, it can discourage companies to invest in KBE tools.

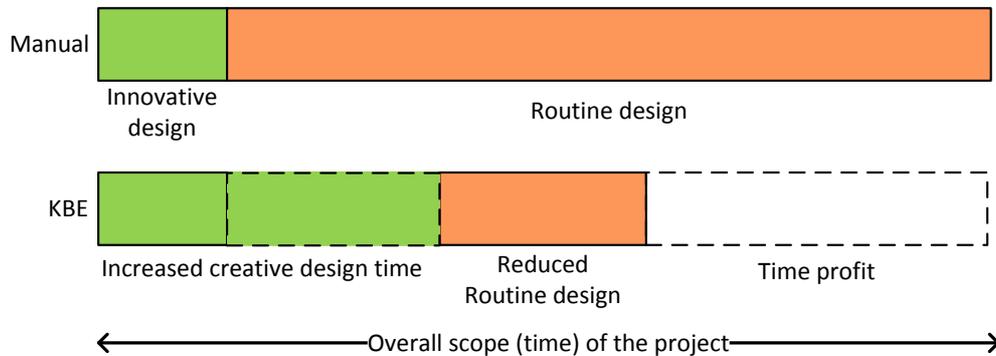


Figure 1: Effects of implementation of KBE (adapted from Stokes [4])

Fokker Aerostructures, a Tier 1 supplier of aircraft components, e.g. empennages, movables, and fuselage panels, believes that developing a KBE tool for one of their design processes can help them to gain a competitive edge. In one of their research projects, called Rudder in a Month (RiaM), Fokker wants to reduce the development time of a composite rudder from two to three years to a month, while increasing the product performance. According to Fokker, this can be achieved by enabling frontloading and using KBE. The research in this thesis assesses whether if, and how, this can be achieved for the automation and optimization of composite rudder structures.

1.2 Front-loading approach at Fokker Aerostructures

As the current rudder design approach at Fokker shows many limitations, a radical rethinking of the design process is required [7]. The current practice is to apply concurrent engineering, as is shown in Figure 2, which means that the process takes place concurrently. The design process is completed when the Critical Design Review (CDR) is achieved, meaning that the design is stable and expected to fulfil the design requirements. Although concurrent engineering reduces lead time compared to sequential product development, the design lead time of a rudder is still too long and does not support optimization.

The reason for the long lead time can be found in the many manual, repetitive steps which take place between engineers from different disciplines. Currently, many involve such step, e.g. simple geometry manipulations like selecting edges of a shape, or inserting data in and executing Excel files or other separate tools and software. Additionally, these steps are repeated multiple times during the design process. These steps can be automated. However, the lack of automation in the current design process results in a long design lead time.

The long design lead time results in additional problems, as it does not support the implementation of parametric studies, let alone the possibility of performing optimization. Currently, only 3-5 designs are analysed during a period of 2-3 years, thus an average of 7-8

months per design option. Performing parametric studies or optimization requires the analysis of a larger amount of designs, which would take too long.

Currently, most of the rudder design knowledge is stored within the engineers, meaning that each of the different steps often requires the contribution of a specific engineer. This introduces additional waiting times: the time required when one engineer is done and the next engineer actually starts. Next to that, it also introduces the problem of the brain drain phenomena when these engineers leave the company.

Additionally, assumptions often have to be made, for concurrent engineering, as requirements are still unknown. If this is done incorrectly, re-design is required and if time is limited, the ad-hoc results in these re-designs lead to additional weight and cost of the final product. A more detailed explanation about the origin of these limitations is discussed in Section 2. Finding a solution to these limitations requires a new approach, which is presented in this thesis.

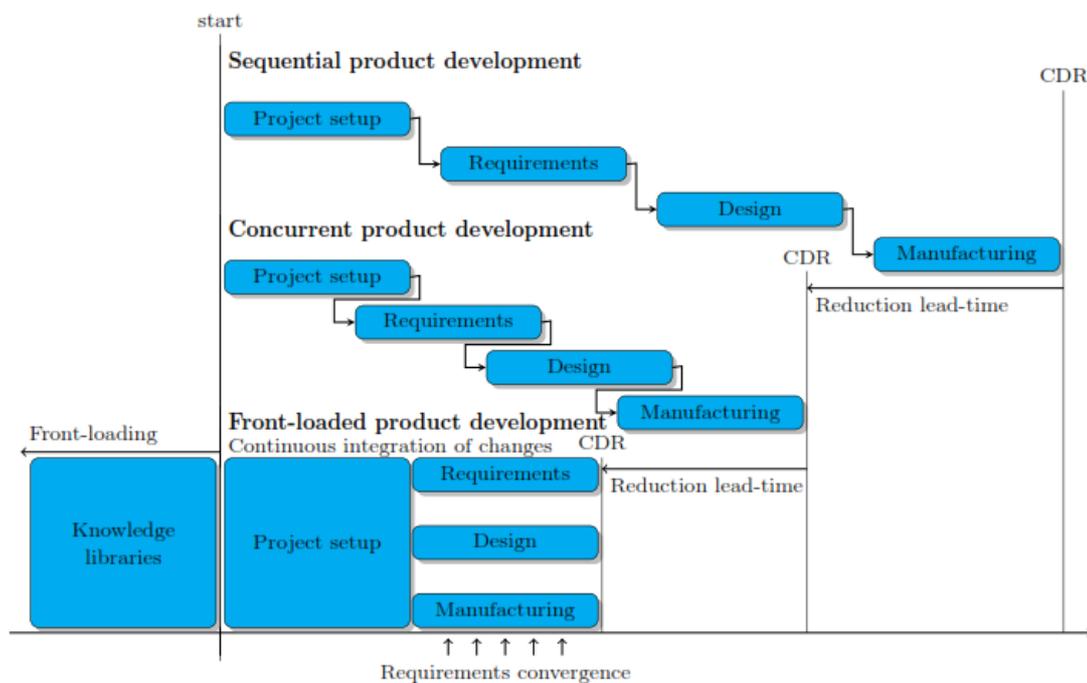


Figure 2: Traditional, current, and proposed future product development process with front-loading implemented [8].

For Fokker, front-loading means developing engineering knowledge libraries in the earliest stage, before the actual design process starts, as is depicted in Figure 2. It is expected that this can be achieved by capturing engineering knowledge and implementing this in KBE applications to rapidly evaluate product variants.

Fokker started the development of the KBE applications to automate the design, though until now these applications have their limitations or are incomplete. One of these applications, the Hinge Design & Optimization Tool (HDOT), developed by Kulkarni [9], focuses on the hinge design optimization using a simplified rudder. A simplified activity diagram of HDOT can be seen in Figure 3. Here, the hinges are the interface points between the fin and rudder. However, Fokker as a Tier 1 supplier of movables has to perform the complete rudder design.

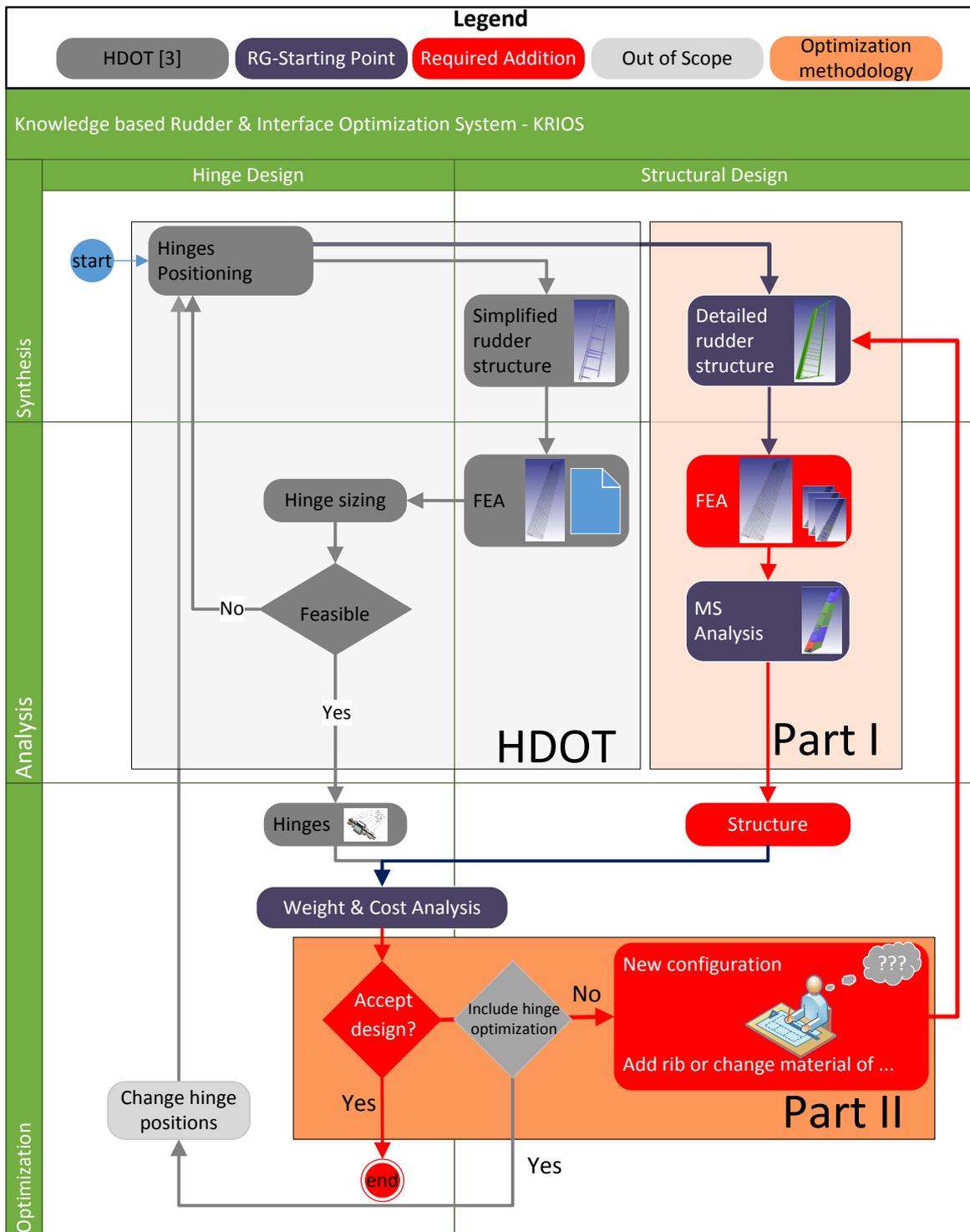


Figure 3: Activity diagram of the Knowledge-based Rudder & Interface design Optimization System (KRIOS)

In a pilot project, Fokker started with the development of a KBE tool to automate the rudder design, the Rudder Generator (RG). However, this tool was incomplete. A simplified activity diagram of the RG is also shown in Figure 3 where one can see that it was only capable of performing the synthesis of the rudder, while the analysis was missing. As a result, the tool could not yet be used for optimization.

Even if a tool could be fully developed (Part I in Figure 3 could be completed), and all the limitations would be solved, there is no clear solution on how to perform rudder optimization after that. Since the amount of possible rudder configurations is endless, one cannot simply try all possibilities and select the best option. A more strategic manner is required to use such a tool to optimize the rudder, which is represented by Part II in Figure 3. Thus, some questions remain, e.g. on how such a tool can be integrated into an optimization framework, on when to accept a design, on what to do when a design is rejected, and on what output is required to answer these questions.

A lot of domain-specific information on optimization is required to answer these questions, which often is not available. Since the current design process does not support optimization, there is limited experience with performing optimization. As a result, non-expert users do not know how to exactly address the above mentioned questions. Therefore, the additional challenge that arose is how to implement the KBE tool in an optimization framework while remaining accessible to non-expert users. This requires accessible optimization strategies. Overall, these strategies will help the engineer in future projects to perform rudder optimization and help Fokker achieve their goal of gaining a competitive edge by designing better rudders faster.

1.3 Research objectives

Considering the above, the objective was the development of these optimization strategies, which should overcome the gap between the developed KBE application and the optimal design. This includes the development of the tool itself, the selection of numerous strategies and the implementation of these strategies. Ultimately, the results inform engineers what the best strategies are and when they should be used, so that an increase in product quality can be achieved in less time. Therefore, the research question is formulated as:

“What methodology can be used to automate the detailed design and analysis process of a composite rudder torsion box, such that it can support the implementation of multidisciplinary design optimization strategies and what are the associated benefits?”

This leads to the following sub questions:

- What are the design requirements of a rudder design at an industrial level?
- How can these design requirements be implemented in a tool, such that it supports automation?
- Why is the current design methodology unable to support optimization strategies?
- Why are the existing solutions unable to support optimization strategies on an industrial level?
- How can the tool be used for optimization strategies and what are these possible strategies?
- How can the benefits of different strategies be obtained and what are these benefits?

The Knowledge-based Rudder and Interface design Optimization System (KRIOS) was developed to automate and optimize the rudder design. A general overview of KRIOS can be found in Figure 3. KRIOS combines two existing tools: the aforementioned HDOT and the Rudder Generator (RG). After extending the RG, the combined tool was wrapped in an Optimization System, to support automation. Hence, KRIOS automates the rudder design and at the same time optimizes the rudder.

1.4 Document structure

This thesis is written in two parts, which comply with the parts illustrated in Figure 3. First, Part I focuses on the developments of a KBE application capable of automating the rudder design. Then, Part II discusses the implementation of the optimization strategies. Therefore, the chapters are structured as follows: Chapter 2 first discusses the current state of the design process and the design requirements of a rudder at Fokker, then the methodologies found in literature, and the challenges that should be overcome. Next, Chapter 3 will introduce some common definitions of a composite rudder to make sure that the content of every chapter can be put in the right perspective. Chapter 4 discusses the development of the model, while Chapter 5 focuses on the theory of developed methodology. The implementation and results of the optimization strategies are discussed in Chapter 6. Finally, Chapter 7 presents the conclusions and recommendations.

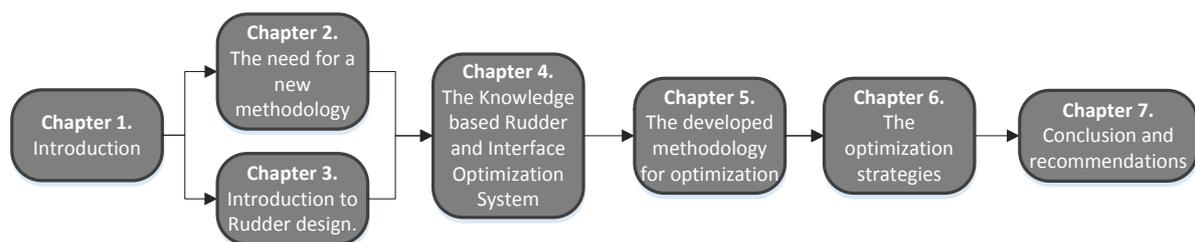


Figure 4: Overview of the chapters of this thesis

2

The need for a new methodology

This section presents the state of the art regarding composite (rudder) structural design. The aim is to emphasize the gap between the current existing methodologies (both at Fokker as well as in open literature) and the required methodology to ensure the increase of product performance within less time. First, it discusses the design process of a rudder at Fokker and its limitations. Then, the proposed solution for both rudder design and composite structure optimization are discussed. From this, the challenges can be identified and the scope of this research is stated.

2.1 The current design methodology and design requirements

The design process of a rudder was studied to identify the limitations of the current industrial approach and proposed solutions to similar problems. As this research took place at Fokker, their design process is taken as a baseline. First, Fokker's design process is introduced, the workflow of the design process illustrated Figure 5. A step by step description is given below. Note that the numbering of these steps matches the numbering in Figure 5. From this design process the design requirements are extracted, of which a summary can be found in Table 2 and Table 5.

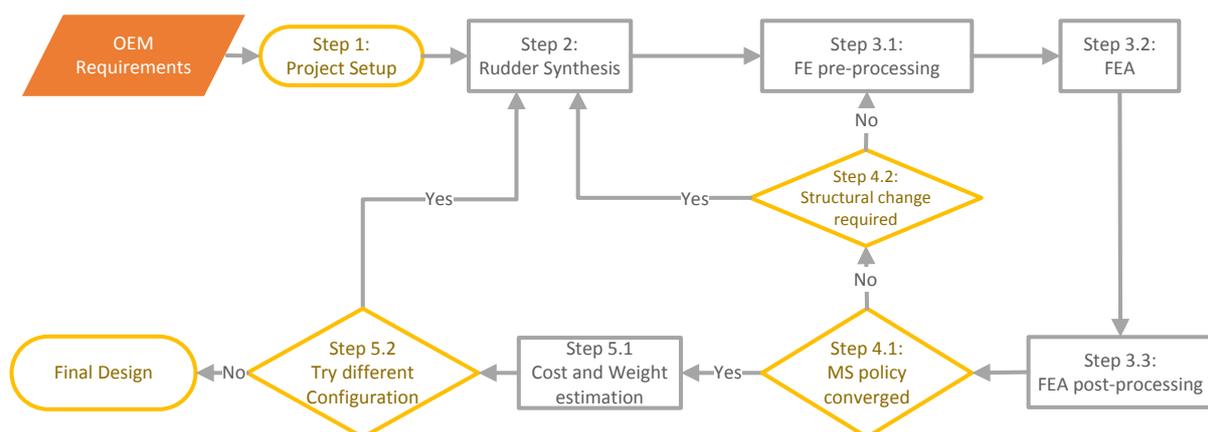


Figure 5: Workflow of the rudder design process at Fokker

2.1.1 Step 1: Project definition

The first step is summarized as the project setup and can be split up in obtaining the request and structuring the requirements. For a rudder this request comes from the Original Equipment Manufacturer (OEM). The request includes a set of requirements, which can differ depending on the OEM. The requirements contain information about the outer dimensions of the rudder, called the Outer Mold Line (OML). Next to that, two sets of planes are given: the first gives the possible planes in which the rotating hinges must lie, the second set gives the planes in which actuator hinges must lie, more information on this can be found in Section 3.2.

In addition to the topology requirements, the different loads working on the rudder can be given. These can be as simple as the total lift the rudder generates, or as detailed pressure distributions as results from wind tunnel testing. The former is usually known in the beginning of the design phase. It often occurs that this is a rough estimate and will change later on. After the requirements are analyzed and set to a standard format, the first step is completed and the synthesis starts.

2.1.2 Step 2: Rudder synthesis

The main objective of the second stage is the synthesis of the rudder, which is mostly done manually and requires a lot of time. For the synthesis, the **design engineer** has to first determine the position of all ribs and spars, which is done with a planar model. The planar model consist of the OML and planes (often referred to as datums), which represent the spars and ribs. Although the OML and some of the rib locations are fixed, since there have to be ribs behind the hinges as is explained in Section 3.2, other datum locations and orientations have to be manually determined and created. This is not the most time consuming process, but it has to be done every time changes are made.

The next step is the material allocation, which is the process of assigning materials to all the spars, ribs, and sections of the skin, this contains many, simple repetitive steps. For the initial model, this is based on knowledge from previous projects. During this process, the engineer has to manually insert what composite material all the components are made of. More on the selection of these composite materials can be found in Section 2.2. The materials allocation is adjusted often throughout the design process, since it is done manually, this consumes a lot of time for a simple process.

After that the geometry generation of each of the components starts, which contains many, non-creative, manual steps. Generating the detailed geometry requires a lot of geometry manipulations. However, these are very simple for an engineer. For example, the generation of extra planes with an offset or selecting the corner points and edges of a square or other shape. More on this can be found in Section 4.2.4. The process behind this is simple and known yet they have to be performed for each and every component, before the full Computer-Aided Design (CAD) model is generated.

2.1.3 Step 3: Rudder structural analysis

Once the synthesis is completed, the analysis process starts. This contains many manual and repetitive steps, but also includes many interactions between engineers. The structural analysis can be divided into three steps: the pre-processing, the analysis itself, and the post-process. During the preprocessing phase, **step 3.1**, a Finite Element Method (**FEM**) **expert** generates a Finite Element (FE) model. At the same time, a **stress engineer** creates and applies load cases to the rudder. Both these processes contain a lot of manual interactions and are therefore time expensive, due to the requirements placed on the mesh and the high number of load cases that should be considered for a rudder. More detail on the latter can be found in Section 2.4. An overview of the time required for the post-processing can be found in Table 1. These times were obtained from interviews with Fokker engineers [10].

Table 1: Time requirement for FE model generation or adjustments at Fokker Aerostructures [10]

Task	Time
Meshing of a newly generated loft	10 days
Change a spar - rib position	1 - 2 day(s)
Applying of external loads	2 days
Assigning different materials	2 days

Although the FEA is performed automatically, a lot of time is required for the pre- and post-processing. With the load cases defined and the FE model available, the **stress engineer** starts the Finite Element Analysis process which is performed with NASTRAN (**step 3.2**). NASTRAN is a FEA software that can automatically analyze the model for the given load cases, hence not many manual interactions are required here. However, there is a lot of time required for setting up the above mentioned FE model and load cases. Additionally, in **step 3.3**, the results are analyzed to determine the critical Margin of Safeties (MS). Here, the MS are calculated using Equation 1. This process is semi-automated using Excel, though still requires some manual interactions, e.g. setting up the Excel input files which include linking the FEA results back to each structural element and its material properties.

$$MS = \frac{Applied\ Load}{Designed\ Load} - 1 \quad (1)$$

2.1.4 Step 4: Margin of safety convergence loop

The MS convergence, is the inner sizing loop which contains many, repetitive steps and a lot of interactions between the different engineers. With the MS obtained, the stress and design engineer will determine what components are over- or under designed and changes can be made accordingly. For the inner sizing loop only minor changes are allowed, e.g. changing the material allocation or rib positions. This is done until the MS are converged, meaning that no large improvement is expected. It is important to note, that each of this loops requires the **design engineer** to change the planar model, who then gives this to the **FEM expert** to change the FE model and reapply the external loads, and then letting the **stress engineer** perform the analysis.

2.1.5 Step 5: New rudder configuration or accept final design

After a design is accepted, weight and cost are estimated, where the latter is a time consuming process. The weight estimation can be done rather quickly, as a lot is known due to the availability of the CAD model. The cost estimation does, however, require many manual, non-creative task and is usually performed by a **costing expert**. Once the weight and cost are known, there is the possibility to make a large topology change with a bigger impact or accept the design. Note that if a large change is made, most of the above mentioned steps have to be repeated. Hence, this decision is usually a trade-off between the expected improvement and the time remaining as one of these iterations can often take a few months.

Table 2: Topology requirements of the rudder

Objects	Object definition	Default value	Comment
Outer Mold Line (OML)	Outer surface	OEM Input	The OML contains the outer boundaries which the rudder cannot exceed, typical OEM input
Hinge line	Position	OEM Input	Two positions perpendicular to front spar, typical OEM input
Actuator line	Position	OEM Input	One position perpendicular to front spar and hinge line, typical OEM input
Hinges	Number	4	Typical OEM input
	Position	Behind fin ribs	Placed behind fin ribs, typical OEM input
Actuators	Number	2	Typical OEM input
	Position	On actuator datums	Placed on actuator datums, typical OEM input
Spars	Number	2	Fixed parameter due to limitations of tool
	Position	As far forward/aft as possible	Determined using manufacturing constraints, can be placed by user by giving a chordwise fraction at the root and tip
	Material	4 – 10 ply composite	Currently one, continuous composite for the whole spar, taken from a ply library
	Flange direction Flange size	Rule-based Depends on fasteners type	In flight direction for front spar, opposite for rear spar Width of the flanges depends on the selected fasteners, thickness is same as main component
Ribs	Number	9	Is controlled by number of ribs
	Position	Equally spaced	Can be done based on an equal pitch between ribs or user can define acceptable position
	Material	4 or 6 ply composite	Ribs can only be made from two predefined composites
	Flange direction Flange size	Rule-based Depends on fasteners type	Width of the flanges depend on the selected fasteners, thickness is same as main component
LE Ribs	Number	-	Number of ribs in the LE.
	Position	-	
Splice plates	Number	1	Number of ribs in the TE, usually only to keep TE skin in place but has no effect in the current optimizations
	Position	Behind the middle rib	
Skin	Material	5 – 16 ply composite	Number of plies can vary, however, they are taken from a predefined ply library
Joint method	Number	Rule-based	Follows from the joint strength analysis Method used join the main components like ribs and skin. One could use fasteners or welds
	Type	Fasteners	

2.2 Limitations of the current design methodology

The current design process at Fokker does not support optimization. This is due to the limitations and bottlenecks that exist. The former can be found in the repetitive steps which are performed manually. For example, the generation of the Finite Element (FE) model (**step 3.1**), which, for every iteration, has to be manually remade after ribs are added or different materials are used. Next to that, bottlenecks occur since an engineer cannot start the analysis before the FE model is generated (**step 3.1 – 3.2**). Furthermore, the engineer performing the analysis is sometimes not directly available. Overall, this results in a long design lead time and does not allow for optimization.

The long design lead time results in two other challenges, which combined propose a major problem; firstly, the rudder design has to start while many of the requirements are estimated, and, secondly, changes in requirements can sometimes not be efficiently integrated in the model. The design starts in a relatively early phase of the aircraft design, because of the time required to design a rudder, while many of the requirements are still unknown and are estimated. This in itself would not have tremendous effects, if it was easy to incorporate changes of requirements into the design. However, integrating these changes is sometimes impossible as most big design decisions are already taken and re-design is not feasible due to time constraints. This means that sometimes material has to be added instead of redistributed, e.g. adding a rib instead of repositioning ribs. Overall, this problem will result in a loss of product performance.

To summarize, the current design process of a rudder is unable to design better rudders faster and, thus, a different approach is required. As the long lead time can be found in the many repetitive manual tasks, automation is required. At the same time, there are numerous of design requirements that should be kept in mind, these cannot be ignored as they have a negative impact on the product quality or lead to non-viable solutions. Therefore, a new methodology is required that supports optimization by automating the design process, while including all of the design requirements.

2.3 Material selection procedure at Fokker Aerostructures

The material selection at Fokker is done using existing ply libraries. While this results in heavier designs, it is required to obtain valid and less expensive rudders. When designing a composite structure, there are many possibilities for the stacking sequence for all the composite components. Finding the most optimal (lightest) is an optimization problem in itself. However, there is one problem when using this approach on an industrial level, which is the validation of new laminates. This validation process is expensive in both cost and time. In case of a small aircraft component, like a rudder, the obtained weight reduction is not enough to outweigh the increase in cost and time. Hence, Fokker selects the laminates from existing ply libraries.

Each structural element has its own ply library and only requires the selected number of plies to obtain the element material properties. An example library is shown in Table 3. Note that this is not the real library and is shown for illustrative purpose only. From these tables, the following information can be obtained:

- **PID:** A unique Ply Identifier;
- **Material:** Name of the material used in the ply, refers to a material database that contains information about material properties, e.g. Elastic Modulus, Poisson Ratio, thickness, density;
- **Laminate names:** In the same row of the PID and Material, one can see the laminate names (highlighted with orange in Table 3 on the left). These names are a combination of the structural component names (skin, spar and rib) and the number of plies of this laminate. These are the possible laminates that can be selected for the individual components;
- **Ply orientation:** The orientation of fibers in each ply, highlighted with purple in Table 3 on the left, can be found in the column below the laminate's name. Note that “ - ” means the ply does not exist in the laminate.

If, for example, the 8_spar laminate is chosen, which is highlighted with green in Table 3 on the right, this will result in an 8 ply thick laminate. An illustration of this laminate is shown in Figure 6.

Table 3: Example ply library for skin (left) and spar (right)

Skin						Spar					
PID	Material	8_skin	9_skin	10_skin	11_skin	PID	Material	6_spar	7_spar	8_spar	9_spar
1.01	mat1.234	-45	-45	-45	-45	2.01	mat2.567	0	0	0	0
1.02	mat1.567	0	0	0	0	2.02	mat2.234	90	90	90	90
1.03	mat1.234	90	90	90	90	2.03	mat2.567	45	45	45	45
1.04	mat1.567	45	45	45	45	2.04	mat2.234	-	-45	-45	-45
1.05	mat1.234	-	90	90	90	2.05	mat2.567	-	-	-	0
1.06	mat1.567	-	-	-	45	2.06	mat2.234	-	-	-45	-45
1.07	mat1.234	-	-	90	90	2.07	mat2.567	45	45	45	45
1.08	mat1.567	45	45	45	45	2.08	mat2.234	90	90	90	90
1.09	mat1.234	90	90	90	90	2.09	mat2.567	0	0	0	0
1.10	mat1.567	0	0	0	0						
1.11	mat1.234	-45	-45	-45	-45						

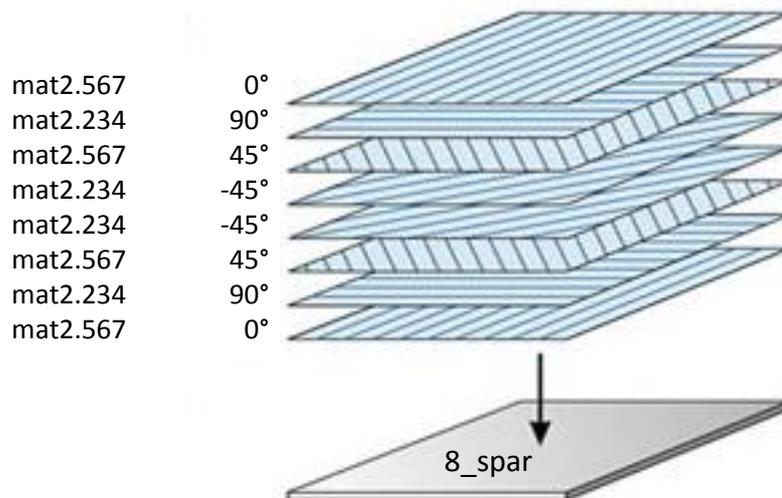


Figure 6: Stacking sequence visualization of the "8_spar" laminate

The predefined ply libraries contain both valid and validated laminates. Here, validated laminates means that they were carefully tested during a previous project, hence require no further validation. Whereas valid laminate means that the laminate complies with a set of requirement and guidelines. According to Kassapoglou [11] and Fokker Aerostructures [10], the following requirements and guidelines have to be met:

- **Symmetry:** Ensure symmetry around the mid-plane;
- **Continuity:** All plies in the thinnest part of the structure should cover the whole structure;
- **Drop-off & ramp ratio:** The amount of drop-offs (reduction of plies) should be done gradually;
- **Grouping:** Avoid grouping of plies with similar orientation;
- **Orientation difference:** Prevent large (90°+) orientation difference in neighboring plies;
- **%-rule:** The amount of $\pm 45^\circ$ plies should be between 40-80% of the total amount of plies;
- **Outer-ply:** Avoid plies with 0° orientation on the outside and should be continuous over the whole structure;

So using predefined ply libraries will both reduce the complexity of the optimization and will result in more cost efficient solutions. Hence, this approach is often used for smaller aircraft components.

2.4 Challenge of full external loads analysis

Even for a small aircraft component like the rudder it is impossible to perform a complete structural analysis during the beginning of the design phase. Performing the complete analysis would require the analysis for over a 1000 load cases that can occur. These cases are a combination of the following:

- **Intact or failure case:** the structural intactness of the rudder;
- **Unit Load Case:** the aerodynamic loads working on the rudder and enforced hinge displacement introduced by bending of the fin;
- **Failure mode:** the failure mode that is analyzed, e.g. material failure, buckling.

As the different combinations can result in high stresses on different locations inside the structure, they should all be considered for the final design. As this would take too long in the early stages, it is common to select the most critical cases during the design process.

The large amount of load cases is due to the combinations that are possible. To illustrate this, consider the following limited possibilities of a rudder which contains four hinges. Each of these hinges can fail in two ways: they can jam or disconnect. For now, assume only one hinge can fail at a time. This would result in the following:

- **Intact or failure case:** One intact case, and four hinges that can fail in two ways

$$1 \text{ intact} + (4 \text{ hinges} * 2 \text{ failure case}) = 9 \text{ failure cases}$$

- **Unit Load Cases:** Maximum aerodynamic loads on rudder to the left and right

2 unit load cases

- **Failure mode:** Material strength failure 1 failure mode.

1 failure mode

The number of possibilities can be found multiplying each of them, resulting in $9 * 2 * 1 = 18$ cases. Though for the real rudder, multiple hinges can fail at the same time. In addition, actuator hinges can fail in three different ways. Table 4 shows how this rapidly increases the number of possible failure cases. Similarly, multiple unit load cases exist and different failure modes have to be checked. An overview of the most critical failure modes, unit load cases and failure cases is given in Table 5.

Table 4: Possible failure cases for two hinge failures

Hinge failing sequence	Possible failure cases
Rotating – rotating	8
Rotating – actuator	24
Actuator – actuator	18
Total	50

Table 5: Summary of the analysis requirements of the rudder

Analysis requirement	Definition	Comments
Failure modes	Material Strength	Check if applied stresses are lower than the design stress of a component
	Buckling Analysis	Check if buckling criteria are satisfied, usually done for load case with highest aerodynamic loads
	Joint Strength	Usually checked in the later stage of the design
	Rib crushing	Usually checked in the later stage of the design
Unit Load cases	Maximum deflection	Minimum and maximum lift, also used for the buckling analysis
	Rudder kick cases	Minimum and maximum hinge moments
	Maximum fin bending	Minimum and maximum hinge displacements
Failure load cases	Hinge disconnection	Hinge breaks and carries no more loads
	Hinge jamming	Hinge jams and does not allow rotations and translations
	Actuator disconnection	Same as hinge disconnection
	Actuator jamming	Same as hinge jamming
	Actuator runaway	Actuator malfunctions and applies full force.
	Rudder jamming	All hinges jam, usually checked in later stage

2.5 Relevant work in open literature

Looking at the challenges in broader context, it can be seen that conventional solutions cannot be used here. Numerous articles can be found in open literature, both on the automation and optimization of (composite) structural design. When selecting a few

examples, the following was found: the solution proposed in the articles either achieve a high fidelity or a large scope. However, to apply a design methodology on an industrial level, a solution that combines both is required.

An overview is given in Figure 7, which shows the distribution of the fidelity level and the scope of selected examples, Table 6 visualizes the definitions of these two. According to Gross [12], the fidelity level is defined as how close the model represent the real world object, or how well the model can represent the **state** and **behavior** of the actual object. The scope can be seen as the **level of integration** of the complete process combined with the **level of MDO** which takes place. According to Griesing and Bently [13] full MDO is defined as “optimization takes place at vehicle level taking into account most critical disciplines” and a low level of MDO contains “a trade studies of generated design points without formal optimization”

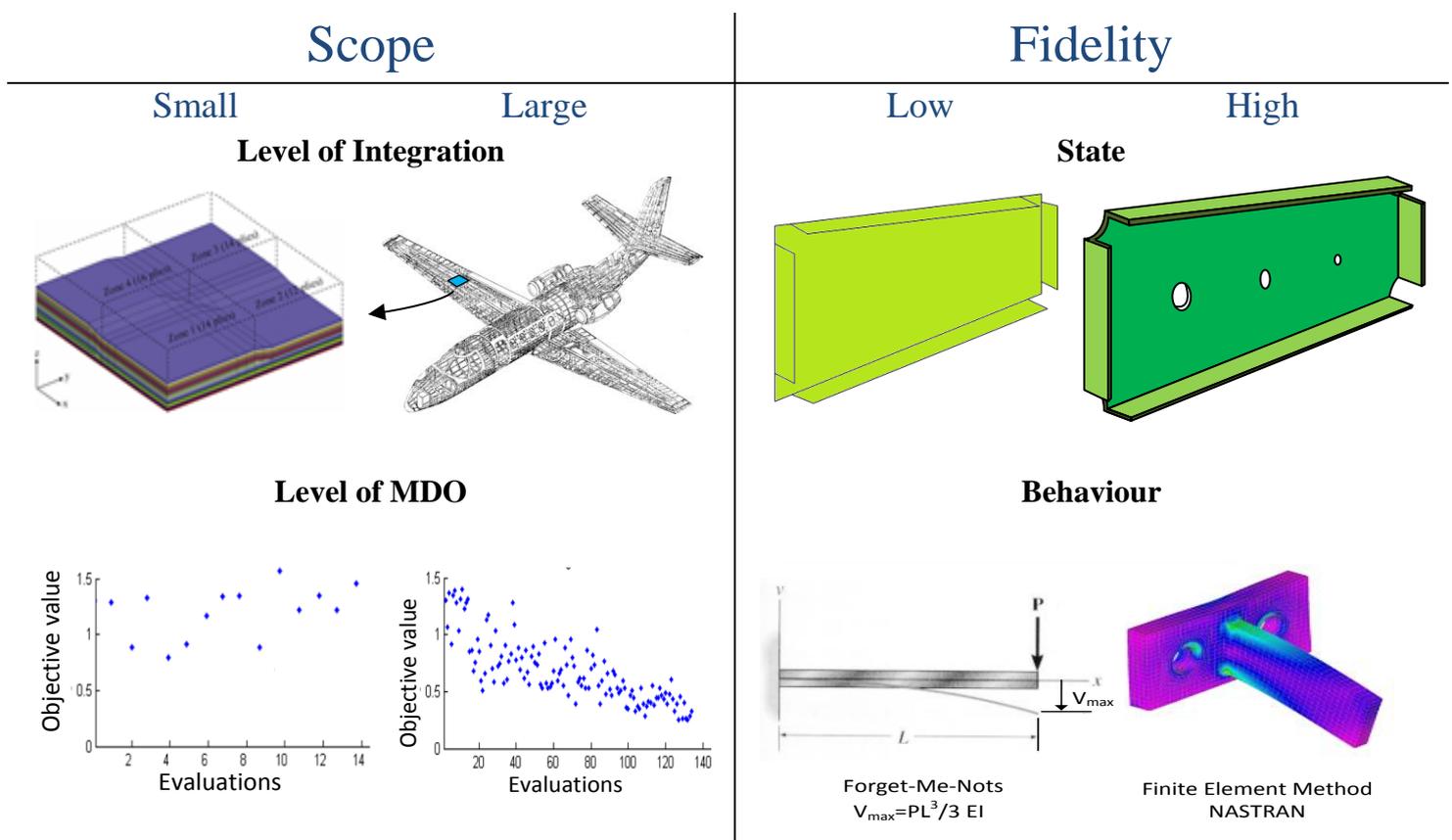


Table 6: Visualization of scope and fidelity

The examples were taken from different studies, and placed on the authors’ impression of scope and fidelity of the project. The selected examples follow the trend, displayed with the blue dashed line, when their fidelity increases, the scope reduces. Hence, current techniques are unable to reach the combination of a large scope and high fidelity.

On the bottom right of Figure 7, one can see the Hinge-system Design and Optimization Tool [9], placed in that location due to its small scope but high fidelity level. As was mentioned, HDOT focuses on the hinge design and optimization, as these are just a small part of the vertical tail, the level of integrations can be seen as low. When looking at the detail of

the hinges and how they are analyzed, it can be said that this gives a good representation of the actual object. Meaning that the level of fidelity for the hinges is very high, hence placing HDOT on the bottom right.

Two other examples can be found at the bottom right; these are two composite structure optimizations. These approaches were taken from Ghaisi and Pasini [14] and [15], which gives a broad overview on composite structure optimizations. Many of the proposed solution in this paper show the implementation of high detail, such as, the stacking and even layup optimization for a plate taking into account different load cases. However, their scope is often limited to plate optimization, whereas this could be used to design small parts of the skin of the rudder, there are many more design requirements that should be considered. Hence, for these it can be said that they have a high fidelity and small scope.

On the other side of the figure, on can see the implementations of proposed rudder optimizations, for which it can be said that they have a higher scope and lower fidelity. Van der Laan [16] supports the optimization of main components of aircraft movables, not only including the design of the torsion box, but also the LE and Trailing Edge (TE). Hence the scope of this project is large. Though when looking at the fidelity level of this approach, it can be seen as relatively low as only the larger components are taken into account. Next to that, this optimization considers a metallic rudder. van Dijk [17] does consider the implementation of composites, however here the scope is limited to the rudder and no MDO has taken place. The approach opted here, shows a large comparison with the design methodology at Fokker as it was a combined research. Hence the same approach was taken regarding the sizing of the materials of the rudder. One could say that this also has a negative effect on the level of MDO as these techniques are often used for weight optimization, but cannot be used when cost becomes an important factor. Overall, these examples have a large scope but low level of fidelity and are therefore placed in the top left.

Lastly, there are two approaches that both involve the composite skin optimization, they try to extend the plate optimization to a more complex structure [18], [19]. However, they do not combine the two at the wanted level. First of all, a less detailed structure analysis is performed compared to the two examples mentioned about plate optimization. In addition, they are for fixed topologies and their objective is weight reduction, hence losing a large part of the scope. So once again, it can be seen achieving the combination of the two aspects seems challenging and the aimed combination is not yet reached

From Figure 7, it can be observed that current approaches are unable to reach the desired combination. The provided, detailed, optimizations are usually limited to small structural components or even plates, whereas the larger models only focus on the main components of the rudder, ignoring the smaller design requirements of the rudder. However, for the implementation of a methodology on an industrial level a combination of large scope and high fidelity is required. Thus, the goal of this thesis is achieving this required combination.

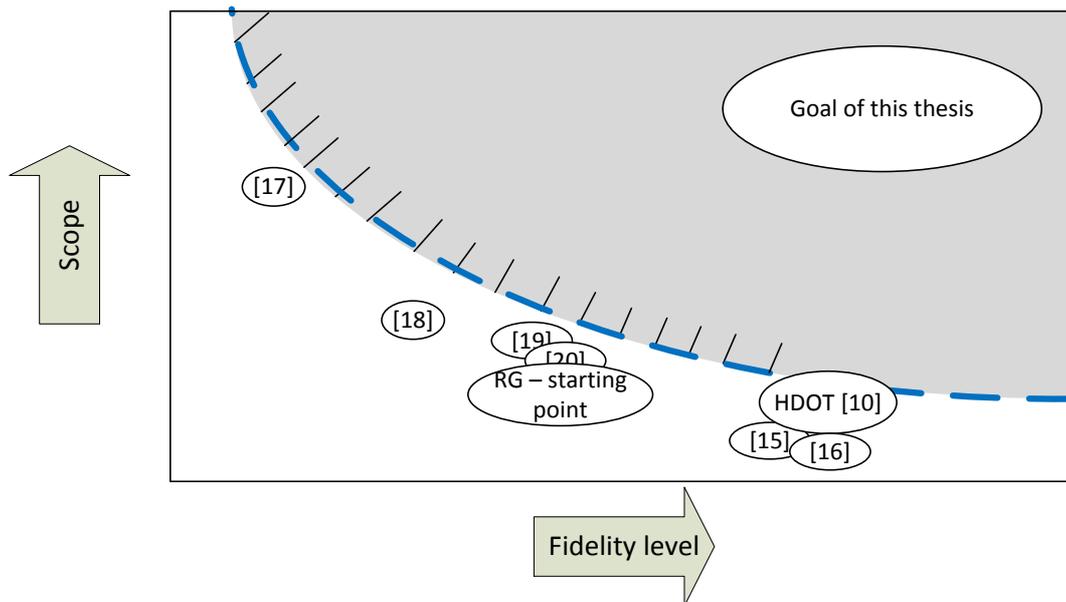


Figure 7: Distribution of scope versus fidelity of previous optimization approaches and the goal of this thesis

2.6 The challenges that lie ahead

When looking at the design requirements, the design process at Fokker, and previous work applied to similar problems, there are many challenges that need to be overcome to apply rudder optimization on an industrial level. The current approach shows a lot of bottlenecks and drawbacks that first need to be solved, before one can even start thinking about optimizing. Even if all these problems are solved, performing a rudder optimization will present its own set of challenges to overcome. Hence, the proposed methodology for the rudder design has to solve all of the following challenges.

- 1) **Automation of the design process:** Summarizing the challenges that occur in the current design process, one can see that there is a necessity for automation to design rudders faster. As the approach shows that the design of a rudder requires many repetitive manual steps between multiple engineers from different disciplines. This results in a **long design lead time**, which introduces two major problems: it does not allow for **optimization of the rudder**, and changes in requirements impact the product performance negatively since they consist out of workarounds as **re-design of the rudder is not possible**. Hence, the first challenge was to automate the process, such that it could be used for optimization, as illustrated in Figure 3, Part I.
- 2) **Integration of the tool into an optimization framework:** Even if a tool capable of implementing all design requirements was provided, there is still no clear strategy on how to continue after that. As the possibilities of rudder configurations are endless, not all the possibilities can be considered. Therefore, the challenge becomes how such a tool can be used such that it achieves a high fidelity design, while taking into account all the design requirements of rudder, within limited amount time. Strictly speaking, how can the tool be used for optimization, as indicated with Part II in Figure 3.

- 3) **Accessibility of both the tool and optimization strategies;** Solving the aforementioned mentioned challenge will require a lot of domain specific information on MDO as well as on the tool, which can result in a prohibitive amount of knowledge required to make use of the tool. Therefore an additional challenge is to ensure that the provided methodology is accessible to non-expert users.

2.7 The scope of this research

Since solving all the challenges would be too much for one Thesis, the scope is limited to the following:

- The fin-rudder interface is assumed to be an OEM input and hence remains constant during this design, since Kulkarni [9] already focuses on the fin-rudder interface design optimization. This means that for this research the weight of the hinges is not computed and the hinge locations are fixed.
- The focus will be on the center torsion box, since the LE box is not yet implemented in the model, and the TE box requires no optimization as it carries no critical loads.
- Fixed number of spars, as it is most common to have at least two spars and multi-spar (2+) concept is not included in the model.
- Rib positioning will be done based on the selected number of ribs.
- Only the material strength analysis and a simplified buckling analysis are taken into account. As implementation of the other failure modes would be time consuming process, but was not seen as a challenge and was expected to have a limited effect on the selected optimization strategies
- Only ten different load cases were used during the optimization, these were selected based on past experience within Fokker with previous rudder designs, see Table 7.
- The objective will be weight optimization, since the cost model was still under development.

Table 7: List of selected load cases

Unit load case	Failure case	Failure mode
Max aerodynamic load left	Intact	Material Strength
Max aerodynamic load right	Intact	Material Strength
Max hinge moment left	Intact	Material Strength
Max hinge moment right	Intact	Material Strength
Max hinge moment right	Hinge 02 disconnect	Material Strength
Max hinge moment right	Hinge 03 disconnect	Material Strength
Max hinge moment right	Actuator 01 jam	Material Strength
Max hinge moment right	Actuator 02 jam	Material Strength
Max aerodynamic load left, no hinge displacements	Intact	Buckling analysis
Max aerodynamic load right, no hinge displacements	Intact	Buckling analysis

Part I – Automated rudder design method



3

Introduction to rudder design

Before diving into the details on how this research provides a solution to the existing problems of existing methodologies, the definitions of a composite rudder will be described to make sure that the content of every chapter can be put in the right perspective. First, a structural breakdown of the vertical tail (VT) is given, followed by a detailed overview of the rudder. Then, existing definitions within in the rudder are explained, as these definitions are used throughout the report.

The main functionality of the vertical tail is to provide lateral stability and controllability. Therefore the vertical tail usually consists of the following three systems, which are shown in Figure 8 and listed below.

1. **Fin:** A fixed, larger part of the vertical tail;
2. **Fin-rudder-interface:** An interface that connects the fin and rudder, these connection points are referred to as hinges;
3. **Rudder:** A moveable, smaller part of the vertical tail.

Each of these systems will experience different sets of forces necessary to achieve the required stability and controllability the vertical tail should provide. Hence, they need to be designed accordingly. The forces depend on the individual sizing of each of the systems, meaning that the design of each system affects the other systems in one way or another, thus making it a multidisciplinary problem. During this section, most of the intertwined effects will be highlight. However, note that for this research the design of the fin is assumed to be a customer input and the fin-rudder-interface design was already done by Kulkarni [3]. Therefore, the focus will lie on the rudder structure design.

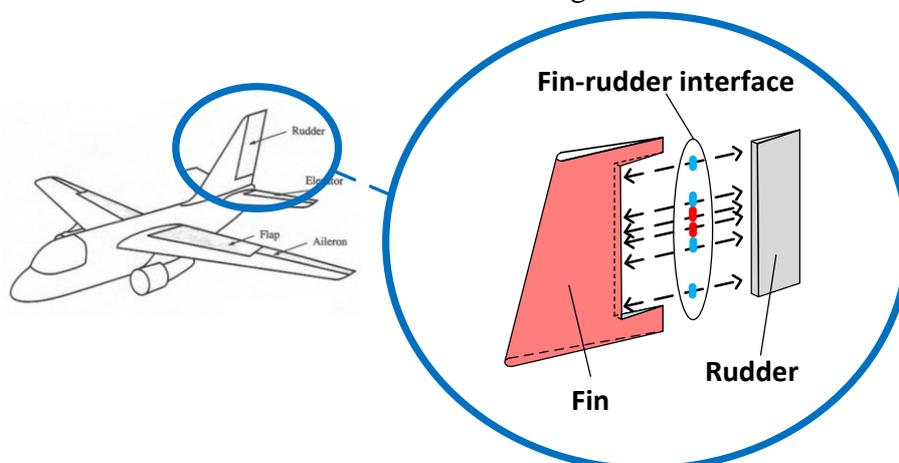


Figure 8: Partial structural breakdown of the vertical tail

3.1 The fin

The fin is the fixed, larger part of the vertical tail which affects the interface and rudder design. The fin has to ensure lateral stability, meaning that it has to generate lift and being able to withstand this lift. Therefore, the fin contains structural elements. Those that affect the rudder can be seen in Figure 9, where each of the structural elements affects the interface and or rudder design in the following way:

1. **Skin:** Aerodynamic outer shape of the fin, illustrated with the red surface in Figure 9. Affects the rudder as the rudder skin must follow the airfoil contour as the fin.
2. **Rear spar:** Spanwise element in the fin, illustrated with the blue solid line in Figure 9. It determines where the fin-rudder interface is positioned, which indirectly affects the size of the rudder.
3. **Ribs:** Chordwise elements in the fin, illustrated with the black dashed lines in Figure 9. Affects the hinge position, which indirectly affects the placement of required ribs in the rudder, as will be explained in Section 3.2;
4. **Actuators:** Actuators can generate a moment allowing for an angle of rotation between the the fin and rudder. It affects the rudder as it determines the position of actuator hinges, hence determining the position of required ribs. Additionally, it introduces large forces on the rudder.

As a result, the location, orientation, and size of these elements will affect the interface and rudder design.

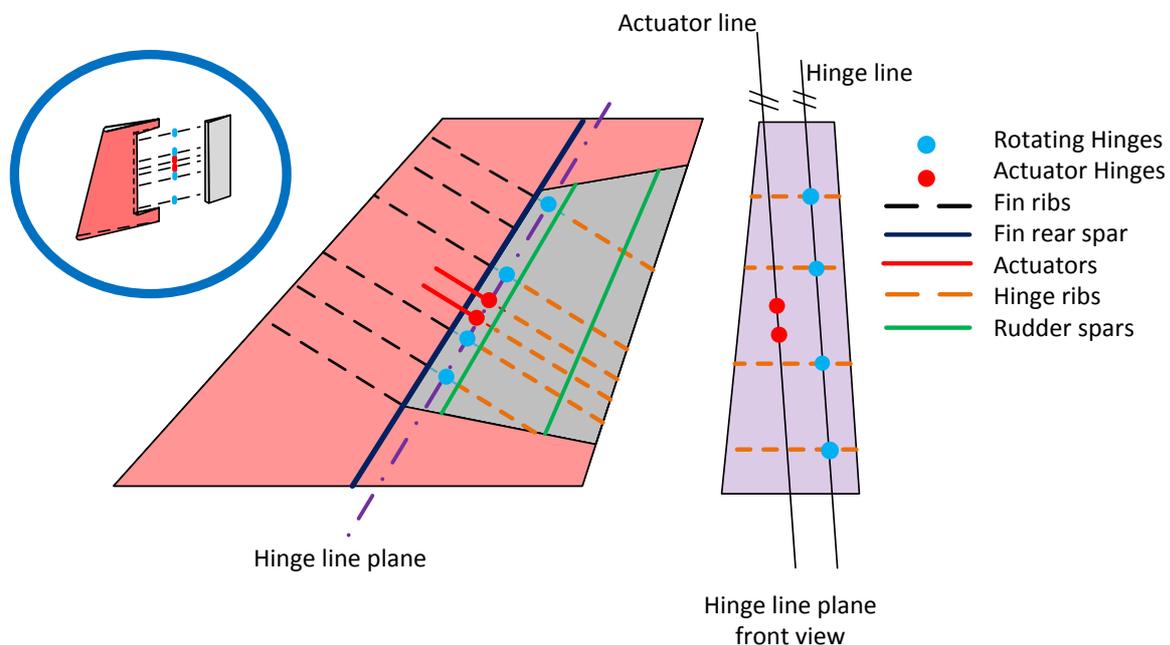


Figure 9: Schematic overview of the fin, interface system and rudder.

Due to the dependencies between the fin, interface and rudder, the vertical tail design starts with the design of the fin. This can be done by the same supplier of the rudder or a different one. For this research, it is assumed that the fin structural layout comes from a

customer, the Original Equipment Manufacturer (OEM). Hence, the design process of the fin is not discussed and the next step is the design of the fin-rudder interface and the rudder itself.

3.2 The fin-rudder-interface

The function fin-rudder interface is to connect the fin and rudder, while allowing an angle of rotation between the two. Therefore, the interface is composed out of rotation hinges and actuator hinges, which are visualized with blue and red in Figure 9 respectively. The rotating hinges connect the fin and the rudder, the latter connects the rudder with the actuators. These actuators can generate a moment allowing for an angle of rotation between the fin and rudder.

The hinge design is a multidisciplinary problem. Previously, it was mentioned that the structural layout of the fin affects the hinge design as it affects the positioning of the hinges. The positioning also affects the rudder structure. This is, however, not a one way street, as the rudder design will determine the forces working on the hinges, which will affect the hinge sizing. Additionally, these forces will affect the fin structure. Making the fin, hinges and rudder design a multidisciplinary problem.

The hinge design can be broadly broken down into four steps, according to Kulkarni [9]. First, there is the design of the structural elements of the fin which affect the hinge and rudder design (see Section 3.1). Second is the hinges and selecting the type of hinges, which is explained below. Then there is the design of the structural components of the rudder, see Section 3.3, which affect the hinge sizing. Lastly, there is the sizing of the hinges, which is also explained below.

As the fin design is taken as a customer input, the first step is the determination of the hinge position which has to comply with many requirements. It was mentioned that the main function of the interface was to connect the fin and rudder, and allow an angle of rotation between the two. Therefore, the placement of the hinges has to be done taking into account the following requirements, as is depicted in Figure 9:

- The rotating hinges must lie in plane with fin ribs, to ensure proper loading paths between fin and rudder. For the same reason, ribs must be placed behind the hinges in the rudder.
- The actuator hinges must lie in the actuator plane to ensure proper loading paths between actuator and rudder. For the same reason, ribs must be placed behind the hinges in the rudder.
- All hinges must lie in one plane, the hinge plane on the right, to allow rotation between the rudder and fin.
- All rotating hinges must be on one line, the hinge line, to ensure it can rotate around the hinge line
- All actuator hinges must be on one line parallel to the hinge line, the actuator line, to allow for rotation between fin and rudder

The hinge design methodology developed by Kulkarni [9], ensures that these requirements are met by using the following parameterization for the hinge positioning;

- Selecting the fin ribs after which hinges will be placed;

- The hinge line is constructed by two points, based on a thickness percentages;
- The actuator line is constructed using one point, based on a thickness percentage and being parallel to the hinge line.

Before the FE model of the rudder can be made and analyzed to obtain the forces working on the hinges, the type of hinges have to be selected. The different types of hinges affect the FEA since they have different degrees of freedom. There are two types of, namely:

- **Sliding hinges:** Allow rotation around all axis, and only one translation along hinge line;
- **Clamped hinges:** Allow rotation around all axis, but no translations.

For the rudder design it is common that the actuators are clamped as well as one of the rotating hinges. This rotating hinge is usually the one closest to the actuators located on the inboard side of the rudder. When this is done, the structural analysis can be performed, the forces working on the hinges can be obtained and each hinge can be sized.

The sizing of the hinges is done by sizing the individual hinge components, to ensure that each of them can withstand the forces. Since the research of Kulkarni [9] focuses on the hinge design, the hinge sizing is not included in this research, meaning that the weight of the hinges is not computed. Next to that, the same hinge positioning method will be used but the positions remains fixed throughout this research.

3.3 The rudder

The rudder is the movable part of the vertical tail which has to generate and withstand loads to provide the controllability of an aircraft. Therefore it contains the following structural elements, illustrated in Figure 10:

- **Skin:** Outer aerodynamic shape of the rudder that carry loads.
- **Ribs:** Chordwise elements that carry loads.
- **Spars:** Spanwise elements that carry loads.

As the rudder structure is the focus of this research, a detailed description of all the structural elements are given below. Including the different types and sections that can occur, and how they influence the design.

3.3.1 The rudder skin

Although the skin of a rudder contains multiple separated sections, the main focus during the design lies on two of these. The skin is divided into a Leading Edge (LE) cap, center and a Trailing Edge (TE) section for the both the Left Hand Side (LHS) as well as the Right Hand Side (RHS) of the rudder, an overview of the sections is illustrated in Figure 10 at the right. The centre sections, of both the LHS and RHS are the point of interest during the design process, as the LE and TE sections often experience low forces and are therefore designed based on manufacturability requirement: the minimum number of plies.

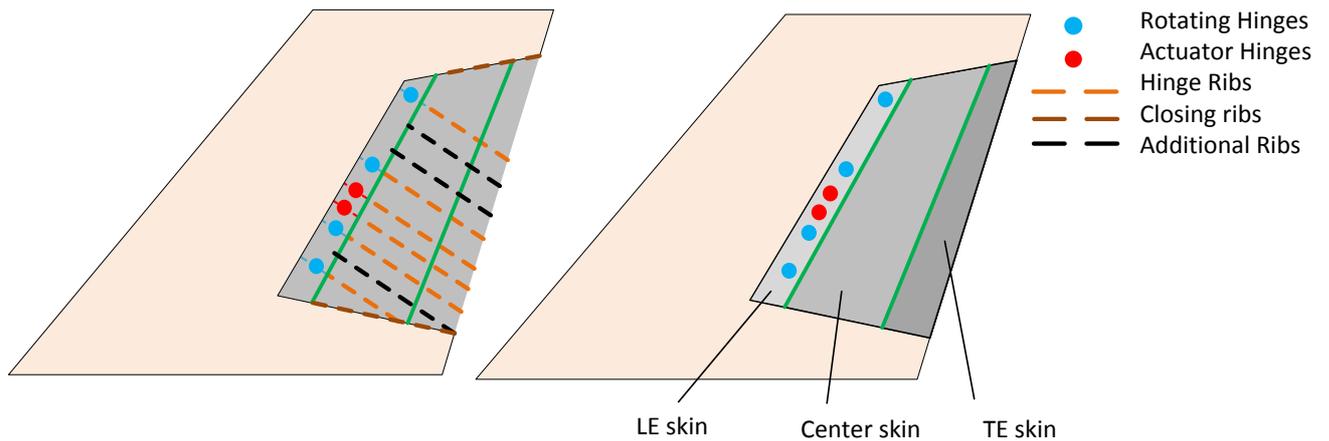


Figure 10: Schematic overview of a partial structural breakdown of a rudder

The centre skin usually consists out of one panel, though different sub-sections are located in this panel. There are different requirements for skin sections located directly above the ribs and spars. Namely, that no drop-offs are allowed in these sections and there is different minimal number of plies due to assembly constraints. Therefore it is important to identify the different sub-sections. Looking at the LHS skin, the following sub-sections are present, these are illustrated in Figure 11:

1. **Interface:** The interface sections are skin strips located above the ribs and spars, see Figure 11 A and B respectively.
2. **Pockets:** The skin pockets are the sections located between the interface sections (two skin-rib-interfaces and two skin-spar-interfaces), see Figure 11 C

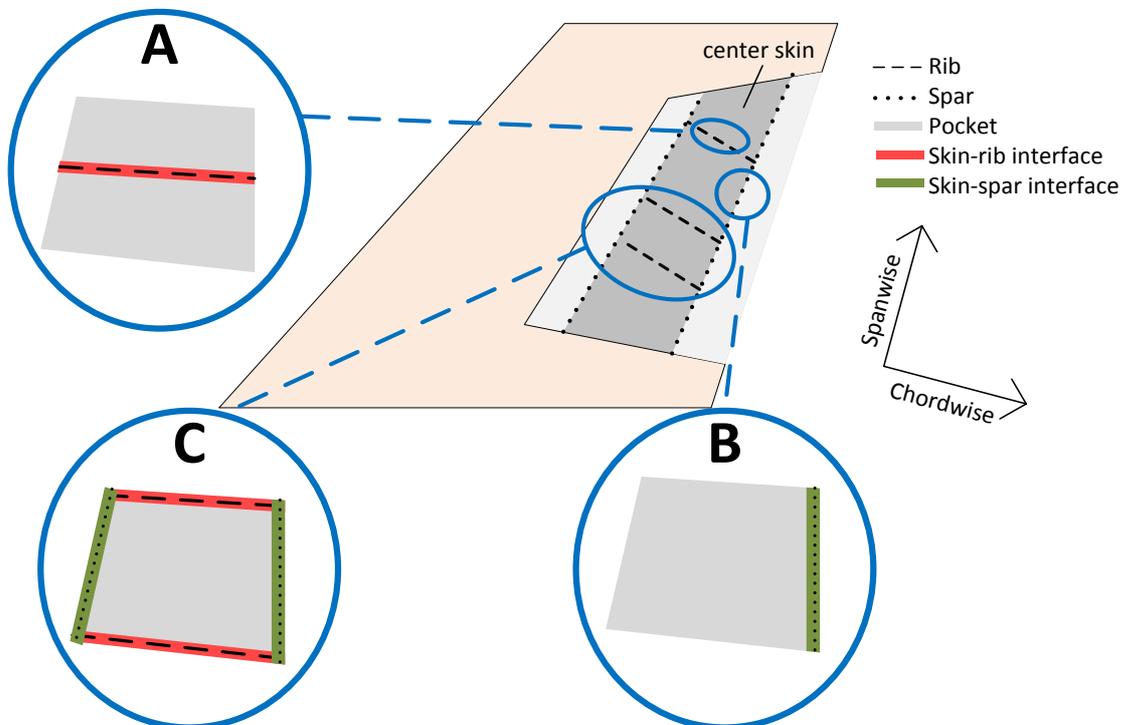


Figure 11: Definition of skin sub-sections. (A: skin-rib-interface, B: skin-spar-interface, C: combined case)

The skin is the outer most shape of the rudder and, therefore, must follow the contours of the predefined aerodynamic surface. Hence, there is a strict boundary condition for the skin. This surface is called the Outer Mold Line and is given by the Original Equipment Manufacturer (OEM). Although the skin contour is fixed by this OML, the material selection of the skin introduces many design options.

The numerous amounts of possibilities is evident due to the existence of many skin sections (interface and pockets) and the number of possible materials. When looking at skin-rib-interfaces, it is common to have just one material in this section, since this reduces the complexity of the shape of the rib. However, for the skin-pockets, as is depicted in Figure 12, there is the possibility to select multiple materials, though this can only occur in chordwise direction. Overall, this can easily result in 40 different sections where, e.g. ten different materials can be selected for each of them. This results in a design space of 10^{40} possibilities. Note that this is just a small part of the design space, as it only considers the possible options for the skin. The full design space contains the combinations of the skin multiplied with the possibilities of other design variables, e.g. the number of ribs and material per rib and material per spar.

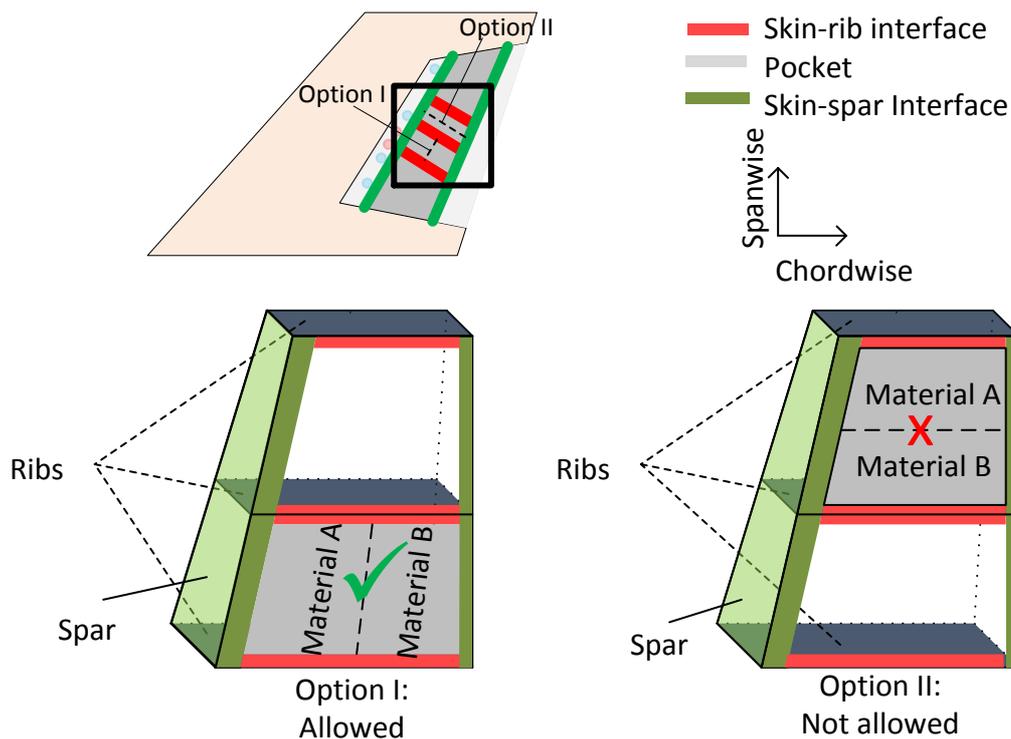


Figure 12: Criteria for material selection in skin pockets

In practice it is chosen not to assign a material to each individual section, but rather to group multiple sections and then sign a material to each of these groups. Though, how to exactly the group the sections remains unknown. For example, one way of doing this is by

categorizing the sections based on their location relative to the actuators, as this is directly connected to the expected loads in the sections of the rudder. These categories are:

- **Heavily Loaded sections:** These are skin section behind and directly next to the actuators.
- **Normally Loaded Sections:** The remaining skin sections.

This summarized the common definitions that arise during the design of a skin, as well as how they affect the design process and how they can be selected. Leaving the description of the ribs and spars of a rudder, of which the former is discussed next.

3.3.2 The ribs of a rudder

There are three types of ribs, during the design the focus lies on one of these types. The different types of ribs are the LE-ribs, Center-ribs, and TE-ribs. Just as for the skin, the ribs located in the LE and TE are often not carrying high loads, but are placed to keep the LE and TE skins in place. Therefore, they often fixed and the main focus lies on the design of the center-ribs. When looking at the center ribs, different types can be identified. These are listed below, an overview can be found in Figure 10:

- **Fixed ribs:** Ribs which are always required in a rudder at a fixed location:
 - **Hinge ribs:** These are the required ribs located behind the hinges, as mentioned in Section 3.1. Illustrated in Figure 10 with the orange dashed lines;
 - **Closing ribs:** Required ribs to close the torsion box, located at root and tip. Illustrated in Figure 10 with the brown dashed lines.
- **Additional ribs:** Additional ribs, which are located between the fixed ribs. Illustrated in Figure 10 with the black dashed lines

Overall, the design decisions for each rib depend on whether they are fixed or additional. For the fixed ribs, number, location and orientation are fixed, only allowing the selection of the material of these ribs. For the additional ribs, the engineer can vary the location, material and amount of ribs. The maximum amount of ribs is limited by the manufacturing constraint of minimum pitch.

Similar to the skin material section, the material selection for the ribs can be done individually or by identifying different groups. Here, one could choose to separate the ribs into two categories and assign the material accordingly. Similar to the skin categorizing, these two groups depend on the location and the expected load path around that location. Though, for the ribs these do not only depend on the actuator hinge, but also the rotating hinges and are defined as:

- **Heavily Loaded ribs:** Ribs that are expected to have to carry high loads, these are usually the **hinge ribs**.
- **Normally Loaded ribs:** Ribs that are expected to have to carry low loads, these are usually the **additional** and **closing ribs**.

This concludes the common definitions regarding the design of ribs as well as how they can be controlled during the design process. With a detailed overview of the ribs and skin presented, only the spars of the rudder remain.

3.3.3 The spars of a rudder

The spars of the rudder are components located in spanwise direction of the rudder. Once again, the types depend on the locations as depicted in Figure 10. The different types are the:

1. **Front spar:** Spar located at the intersection of the LE and center.
2. **Rear spar:** Spar located at the intersection of the center and TE.
3. **Intermediate spars:** Spars located in the center box (between LE and TE spars)

For the rudder design, it is common to only have a front and rear spar, as intermediate spars are often non-existing due to the small size of a rudder.

The location of the front and rear spar is often fixed, since it is most optimal to place them as much forward and aft, respectively. Since this increases the size of the torsion box, thereby making it more resistant to torsion. Therefore, the only decision regarding the spars is the material selection. For the spars it is possible to define a material distribution in spanwise direction. However, as is depicted in Figure 13, there is no drop-off allowed between two ribs.

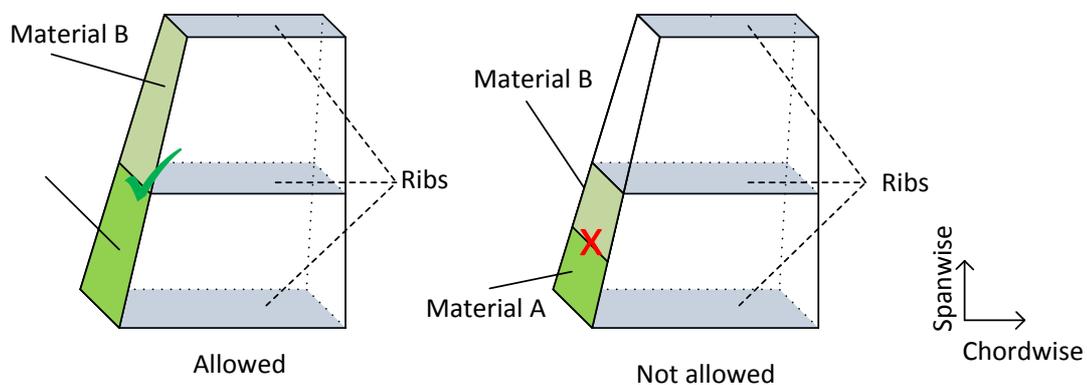


Figure 13: Criteria for material selection of spar

This chapter presented a global overview of the design of a vertical tail. Although many, smaller components can be found in the VT, the most important one regarding the rudder design are discussed here. Additionally, some of the possible design decisions were discussed as well as how they increase the size design space, e.g. the large amount of skin variants. As a result, automating the design process, finishing Part I of Figure 3 is just a start. Find good solution inside the immense design space can propose an even bigger challenge. Though, the first step remains the development of a tool capable of doing this, which is discussed next.

4

The Knowledge-based Rudder & Interface design Optimization System (KRIOS)

The **K**nowledge-based **R**udder & **I**nterface design **O**ptimization **S**ystem (**KRIOS**) is a Knowledge-Based Engineering (KBE) tool developed to automate and optimize the rudder and fin-rudder interface system. KRIOS combines the existing **H**inge-system **D**esign and **O**ptimization **T**ool (**HDOT**), developed by Kulkarni [9], with the **R**udder **G**enerator (**RG**). As the names suggest, HDOT focuses on the automation of the interface design, whereas the RG automates the rudder structural design.

Although HDOT includes its own rudder design, it has major limitations. To automate the fin-rudder interface design, HDOT generates a very simple metallic rudder structure. In addition, this structure is never analysed for its structural strength, cost and weight. However, Fokker Aerostructures is a Tier 1 supplier of composite rudders, so the rudder design of HDOT was not sufficient and the RG was developed.

The RG was developed during this research to automate the design of a detailed composite rudder. However, it should be noted that the RG is a combined effort by a team existing of engineers from Fokker Aerostructures, ParaPy b.v. and the author itself. An overview of the author's contributions can be found in Appendix A. The RG was partly developed during a pilot project. However, it was not yet complete at the end of the pilot, meaning that the RG had to be further developed during this research before it could be used for optimization.

This section will focus on the development of the RG. First, the methodology behind the RG is explained. Then, an overview of the main structure of the RG is given and the individual modules are shortly explained. Lastly, the validation of the tool is discussed.

4.1 Methodology behind the model

The methodology used for the developed tool is based on the concept of front-loading. Front-loading can be achieved in multiple ways, where the most common ones are Rapid Problem Solving and Project to Project Transfer. The former can be achieved by using advanced technologies, e.g. Computer-Aided Engineering (CAE) and Computer-Aided Design (CAD) to develop applications and tools and thereby reduce problem-solving time, according to La Rocca [6]. The latter, Project-to-Project transfer, can be achieved by retrieving knowledge about problems and their solutions from previous designs and implementing them in the current design process. For the development of the tool, both concepts are used to enable front-loading.

Implementation of the two concepts can be found in KBE, where KBE systems allow for both the storage of knowledge and automation. According to La Rocca [20] this is defined as: *"Knowledge-Based Engineering is a technology based on the use of dedicated software tools called KBE systems, which are able to capture and systemically reuse product and process engineering knowledge, with the final goal of reducing time and costs of product development by means of the following:*

- *Automation of repetitive and non-creative design tasks.*
- *Support of MDO in all the phases of the design process."*

The development of the tool was performed in ParaPy, which is a KBE system that combines the power of CAD and Artificial Intelligence (AI). ParaPy is a Python based objected oriented programming framework which can be used to build parametric, rule-based applications that can automatically design engineering processes [21]. Hence, it can capture both concepts of rapid problems solving and project to project knowledge transfer.

4.2 The main structure of the Rudder Generator

KBE offers the possibility to automate the synthesis and analysis of a composite rudder, because the many, repetitive and rule based steps in the design process. Therefore, the Rudder Generator was developed in a KBE system, called ParaPy. In this section, the main structure of the developed tool is discussed, this includes the functionalities of the tool and how they automate the manual task. A global overview of the RG can be found in Figure 3, Part I, whereas a more detailed overview of main modules and how they are connected is shown in Figure 14. The processes can be found in the seven main modules of the RG. Since part of the tool was already developed during a pilot project, the focus will be on the added functionalities. An overview of how the work was divided can be found in Appendix A.

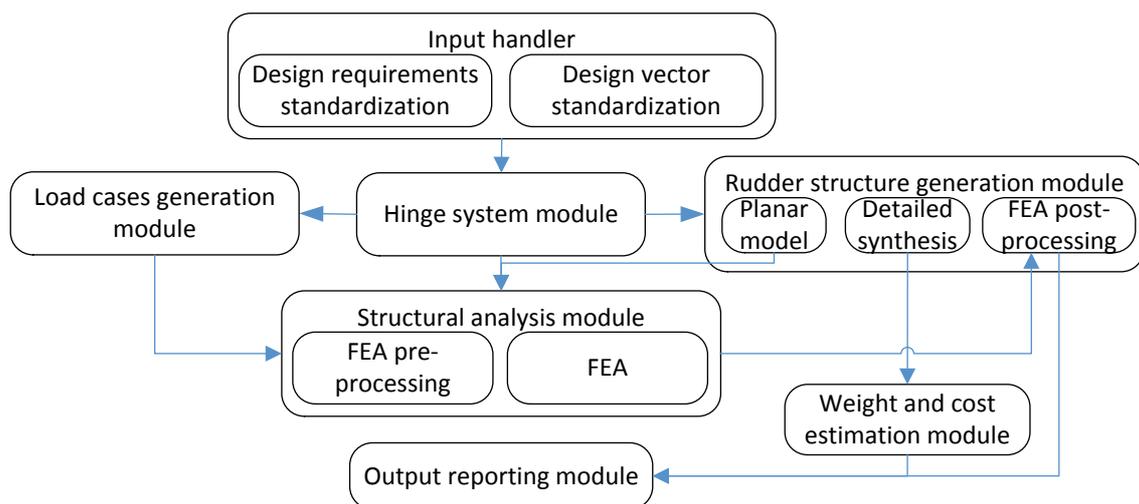


Figure 14: Overview of the main modules of the Rudder Generator

4.2.1 Input handler module

Before starting the process of the rudder design, the first step is to convert the different possible input formats to a format the tool can work with, as depicted in Figure 15 a). Depending on the customer and the amount of information available, required design inputs can come in different formats. Fokker has to adapt to these different geometry input types, to a format the tool can work with, this is illustrated in Figure 15 b). Additionally, it is currently unknown what the best way is to control the tool, meaning that different parameterizations can occur during the optimization strategies. Therefore, as depicted in Figure 15 c) the input handler has to ensure that the new parameterizations are also converted to an input the tool can work with.

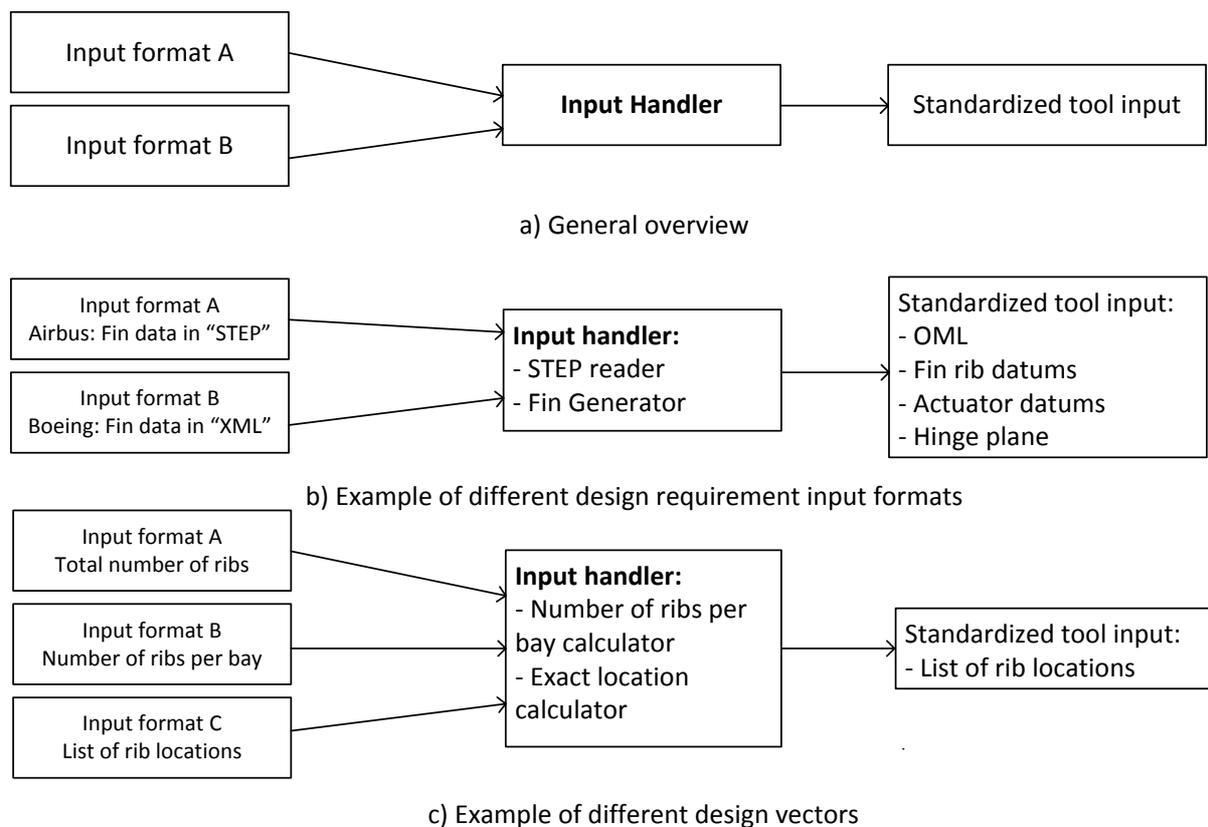


Figure 15: Standardization of different inputs formats and the required output

Design requirements standardization

A minimum amount of input is required to design a rudder, since this input can come in different formats, it has to be standardized. The required geometry inputs are a strict boundary condition for the rudder and fix the position of some structural elements, e.g. OML, locations of fin ribs and actuator datums. These inputs are fixed during the rudder design and can be provided in multiple ways, depending on the customer and the stage of the design project. For example, the customer can provide the exact geometry in a STEP or IGES file, where the tool has to read these files and provide the required input geometry in a standard format. Additionally, since according to Fokker front-loading means that the rudder design

starts before the customer proposal, it could be the case that this exact geometry is unknown due to confidentiality. As a result, the geometry has to be estimated and generated based on a set of parameters concerning the topology of the vertical tail, e.g. root- and tip chord, span, and fin rib positions and once again should result in the same geometry format.

At this point, it is expected that there are two different input formats. These are a STEP file or a XML file, the input handler can convert both formats to the standard format. Additionally, due to the existence of this module, if another input format is provided they can be simply integrated here and does not require the engineer to go into the core structure of the tool. This is beneficial since making changes in the core of the tool requires that the engineer has a good understand of the tool. The resulting input geometry from the STEP and XML file is illustrated in Figure 16.

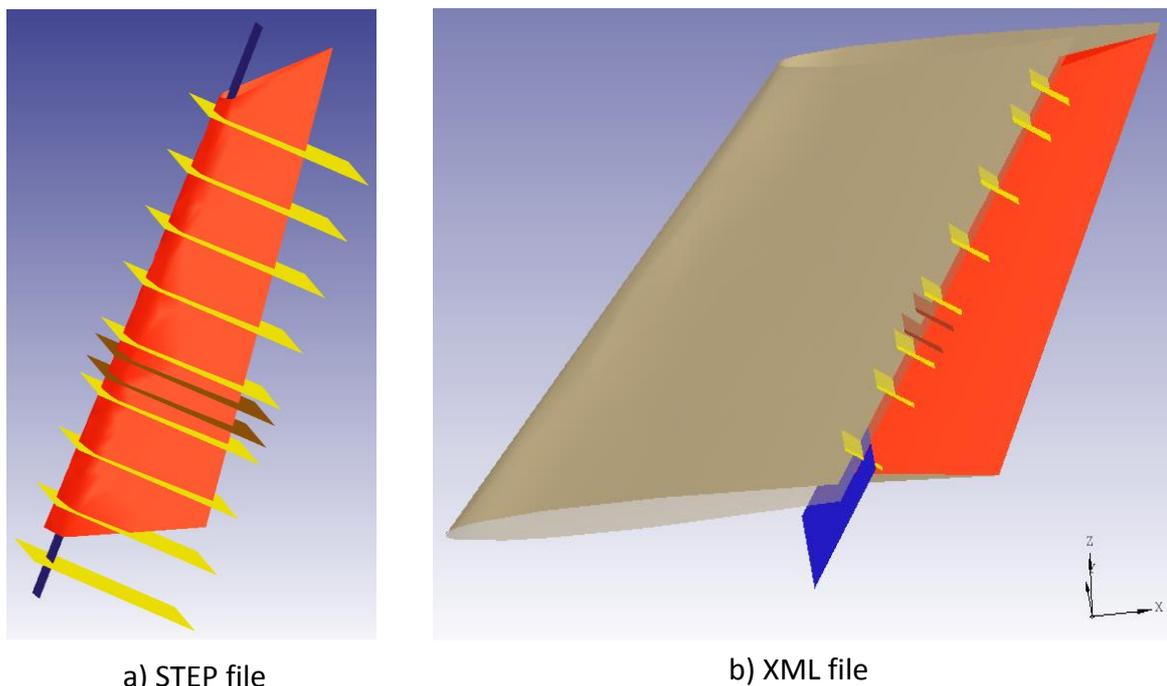


Figure 16: Results of example input geometry from different input files

Design vector standardization

Additionally to the design requirement standardization, a similar process occurs for the design vector to increase the accessibility of the tool. The tool has to be capable of dealing with the different parameterizations, to support the different optimization strategies. Since there is no clear method on how to control the tool, e.g. how to determine the number of ribs and their location, and how to assign the materials to all the components, different parameterizations were analysed throughout this research. This module converts the different parameterizations to an input the tool can work with. Two examples of how different design variables are standardized are discussed below. It is important that this module can be easily extended or adjusted to support future optimizations, without having the engineer to make changes in the core of the tool as this would require the engineer to have a good understanding of the tool.

Rib positioning

The first example, the number of ribs in a rudder, is one of the main design variables and can be controlled by numerous variables. First of all, as depicted in Figure 15 c), it is possible to just define the total number of ribs and let the tool determine where the ribs are placed. Next, it is possible to say how many ribs there are available in every bay, where a bay was defined as the area between two **fixed ribs** as defined in 3.3.2 and let the tool determine where in the bays the ribs are placed. One could go even further and give the exact locations of the ribs.

This module contains the functionality to account for all of these inputs. If, for example, the total amount of ribs is defined, the tool determines the exact location such that they maintain an equal distance with each other. However, it is not possible to equally distance all the ribs as the rudder contains some fixed ribs.

Looking at the example depicted in Figure 17, there are two bays, Bay 1 has an initial pitch of 40, Bay 2 of 108. If one additional rib should be placed, this will be done in the bay where the rib pitch is the largest, which is Bay 2 in this case. Then the new rib pitch is computed and the process starts over again. A simplified computer function is shown below, which converts the total number of ribs to number of ribs per bay, the results of selecting 6 additional ribs is shown in Table 8.

Additionally, one can just define the number of ribs per bay, avoiding this whole process, while giving the engineer a bit more freedom, but adding more variables. Lastly, one could give the exact locations of the ribs itself. Doing this gives the engineer even more freedom in designing a rudder as ribs are not placed at an equal distance. However, it has many downsides which complicate the optimization, e.g. it adds a continuous variable, and it results in a variable amount of design variables (number of locations depends on number of ribs). Even though the latter is unlikely to be used, the tool accepts all the different input variables to select and place the number of ribs.

```
>> Loop until number of additional ribs is reached
    >> Find bay with maximum rib pitch
    >> Add rib to this bay
    >> Compute new rib pitch of this bay
```

Table 8: Effect of adding ribs on the rib pitch per bay

Additional ribs	Rib pitch bay 1	Rib pitch bay 2	Find maximum rib pitch	Ribs bay 1	Ribs bay 2	Add next rib in
0	40	108	Bay 1 < Bay 2	0	0	Bay 2
1	40	108/2 = 54	Bay 1 < Bay 2	0	1	Bay 2
2	40	108/3 = 36	Bay 1 > Bay 2	0	2	Bay 1
3	40/2 = 20	108/3 = 36	Bay 1 < Bay 2	1	2	Bay 2
4	40/2 = 20	108/4 = 27	Bay 1 < Bay 2	1	3	Bay 2
5	40/2 = 20	108/5 = 21.8	Bay 1 < Bay 2	1	4	Bay 2
6	40/2 = 20	108/6 = 18	Bay 1 < Bay 2	1	5	Bay 1

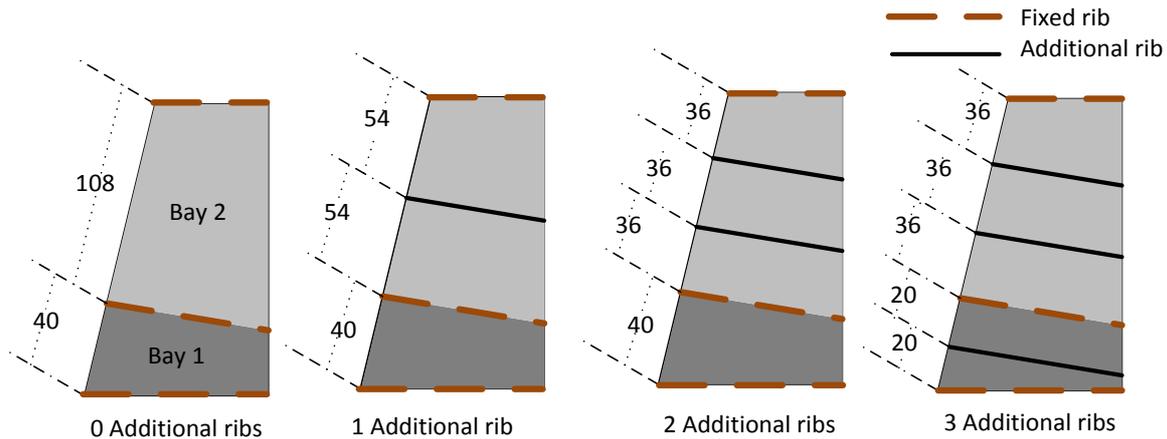


Figure 17: Example of rib distribution function from zero to three additional ribs

Material allocation

Secondly, the material allocation is an important step of the rudder design and can also be done using different design variables. Allocating materials will determine what laminate the components are made of. Just as with the variable for the number of ribs, this can be done in many ways and once again they have to be transformed to a standard format. For example, one can select one material for all the ribs, one can split the ribs in different groups and provide different materials for these groups, and one can give the material for each rib individually. This module is capable of dealing with the different inputs and transforms it into a standard material dictionary, for the ribs but also for the skins and spars. An example of how this is done for the ribs and how it supports future optimizations strategies follows below, note that a similar process is done for the spars and skins.

The standardization for the rib material allocations allows the tool to run with the different input variables and support future extensions of the optimization strategies. Take for example a rudder with four ribs (a root rib, a tip rib and two intermediate ribs) the resulting material dictionary for the ribs will have the following format, shown in Table 8. It will be a list of lists, meaning that the materials selected for each rib can be one or multiple materials. Although, only one material for each rib is selected at this point, for future optimization one might want to try different materials for one rib. Therefore, already assuming that list will be used, even though it only contains one material, the engineer can easily implement this at a top level at a later stage. As the engineer only has to create the list and does not have to implement changes inside the tool.

Now that the format is explain, lets look at how the three, aforementioned, inputs are standardized. The results are shown in Table 8. First of all, if one material (**material a**) for all the ribs are given, it makes the dictionary shown in the second column. Second, if two groups are defined, e.g. the intermediate ribs and the inside and ribs on the outside, where the groups are given **material a** and **material b**, respectively. This results in the third column. Lastly, if a material is assigned for all the ribs individually (**material a - d**) one can easily create the list shown in the fourth column. So the tool is capable of handling the different input formats and allows for a more detailed material allocation in the future.

Table 9: Resulting rib material dictionary as a result of different design vectors

General rib materials library	One material for all ribs	Inside and outside ribs	One material per rib
[material 11, ..., material 1x]	[material a]	[material b]	[material a]
[material 21, ..., material 2x]	[material a]	[material a]	[material b]
[material 31, ..., material 3x]	[material a]	[material a]	[material c]
[material 41, ..., material 4x]	[material a]	[material b]	[material d]

Just as for the number of ribs, the different material allocations allow for more freedom while increasing the design space. Although there are only two materials of which a rib can be made of, when assigning the material for each rib individually, it will result in many variants. For example, it is not unlikely to have a rudder with 15 ribs, which result in $2^{15} = 32.678$ possibilities. Whereas defining two groups and assigning the material accordingly, it results in $2^2 = 4$ possibilities. Once again it is a trade-off between the freedom and the complexity of the optimization.

This concludes the main functionalities of the input handler, which is an important step before the rudder design can start. As Fokker has to ensue the demands of the customer, different input formats have to be accepted. Additionally, there are different ways on how the design variables of a rudder can be controlled. It is important that the different inputs are standardized at a top level, meaning that the engineer can adjust or extend the functionalities without going in the core structure of the tool, which is very complex. It should be noted that not all of the implemented functionalities are used, but were added as place holders for recommended strategies. Now that the input is converted to a standard format, the design of the rudder can start. Here, the first step is to determine the exact position of the hinges.

4.2.2 Hinge system module

The hinge system module places, generates, and sizes the hinges. The first step is to determine the exact location of the hinges. The exact locations of the hinges can come from the STEP file delivered by the costumer, or are based on a set of parameters. As this approach was taking from HDOT [9], more information behind this process can be found there.

Having placed the hinges, the forces acting on them need to be determined before they can be sized. Therefore, the structural analysis module generates a finite element (FE) model and performs the finite element analysis in NASTRAN. From this, the maximum forces working on the hinges can be obtained. Note that the sizing of the hinges was outside of the scope of this research, so the weight and the cost of the hinges is not analysed. The position of the hinges also remains the same throughout this research, meaning that the effect of changing the hinge locations is not studied. Although this reduces the scope for now, first there is a need for a methodology that can optimize the rudder before the hinge design optimization can be included.

The hinge positioning influences the rudder structure and the load cases. In Section 3.3.2 it was already mentioned that **fixed ribs** are required behind the hinges, hence they influence the structural layout. Next to this, they also determine the boundary conditions of the rudder,

as they restrict the movement of the rudder in some directions, as was mentioned in Section 3.2. These boundary conditions are important for the generation of the different load case. Since the hinge positions are now determined, the generations of the load cases and rudder structure can start.

4.2.3 Load cases generation module

The rudder structure has to be able to withstand different load cases, as mention in 2.4, which can be easily generated and selected in the Rudder Generator. There are many different load cases, not all of them are analysed, and hence this module allows the engineer to select a certain combination of load cases and generates the required load case information for the FEA. The load cases can be easily selected by switching them on and off, allowing for quick feedback of the effects of each load cases. In addition, engineers can choose to increase the sizes of certain loads, once again allowing them to study the effects.

Information regarding the unit load case (the aerodynamic forces) can come in four levels, just like in Section 4.2.1, the tool has to be capable of dealing with different levels. Luckily the latter three are rather easy to implement whereas the main challenge lies in the first one. The different levels are as follows:

- **Level 0:** Minimum sizing data is available and past experience is used
- **Level 1:** Initial wind tunnel data is available
- **Level 2:** Data is reliable enough to fulfil load case certification
- **Level 3:** Actual flight test data is available

Currently it is most common that only Level 0 data is available at the start of the rudder design. The main difference between level 0 and level 1 - 3 is that for a Level 0 the data available is just the total lift the rudder generates. The latter can consist out of an actual pressure distribution. For the Finite Element Analysis, the pressure needs to be known over the whole skin. Therefore, the different levels of input are transformed to pressure distribution function. This function computes the pressure force based on the x (chordwise) and z (spanwise) coordinate and can determine the pressure all over the skin. For the Level 1 – 3, this function can be easily derived. In case of Level 0 data, first a triangular pressure distribution over the skin is assumed and then the function is derived accordingly.

Hinges may fail in two or three ways, as was discussed in Section 2.4, which will affect the load paths. This module can recognize the different failure cases and provide the input for the FEA. Take for example a normal working hinge, for which the translations can be restrict in x and y direction or x, y and z depending on the hinge type. Though, when a hinge disconnects, these constraints are no longer there and if a hinge jams, no translations or rotations are allowed. To illustrate, Figure 18 visualizes two of the used load cases mentioned in Section 2.7, an intact case with maximum rudder deflection to the right and the same case with a hinge 2 failure. In this figure the blue (triple) arrow show the translation constraints of the hinges, note that the constrains of the second hinge are not there for the failure case (circled in red), the yellow arrows show the actuator forces and at last, the left and right hand side pressure distributions can be seen by the larger trapezoidal areas (RHS = brown and LHS = red). Note that this only requires the engineer to select a hinge and the

failure case of this hinge, whereas the process of creating the FEA input file, running the FEA and analysis of the MS is done automatically. An overview on how this affects the MS of the rudder is shown in Section 4.2.7, Figure 25.

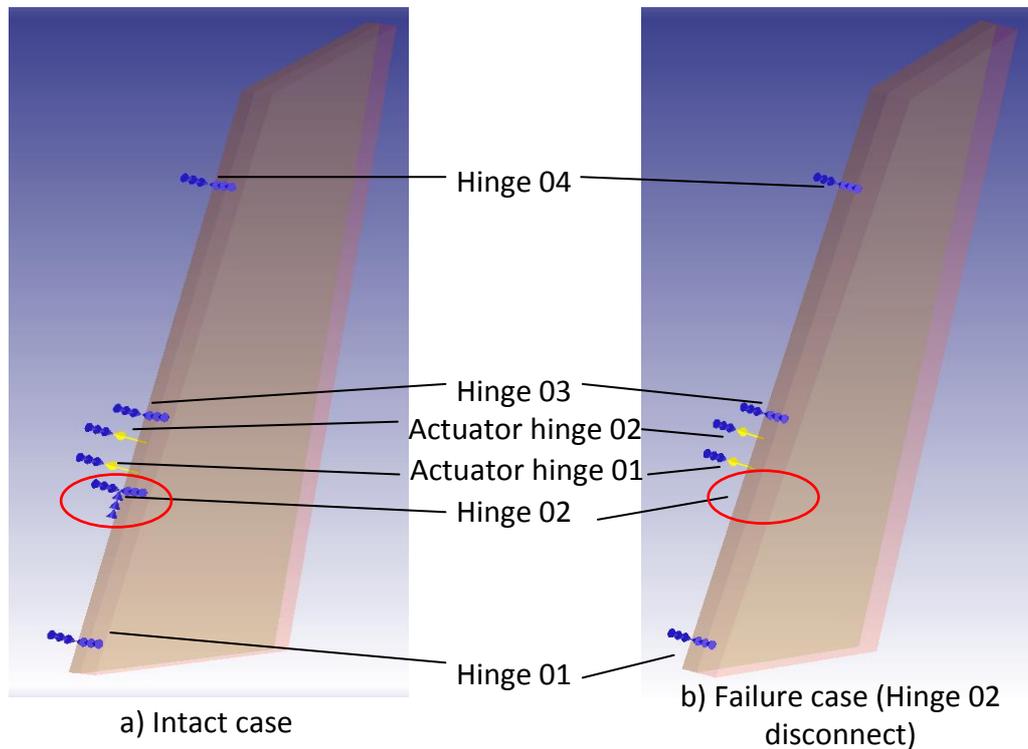


Figure 18: Visualization of constraints, forces and pressure distribution for two load cases

4.2.4 Rudder structure generation module

This module performs the detailed synthesis of the rudder components and can be seen as the core of the Rudder Generator. Each of the separate structural elements (skin, ribs, spars and joints) are generated here, as many of these elements have a dependency with each other, this is done concurrently. Luckily, with a KBE software like ParaPy, these dependencies can be easily maintained. Additionally, the generation of these CAD models require many geometry manipulations (selecting edges as is explained later on in this section), which are currently performed manually and take a lot of time. During this section, the generation of a detailed rib is considered. A step-by-step approach is discussed below, a visualization of the steps taken are illustrated in Figure 20 and Figure 21.

Multiple steps are required to generate the complete model, where the first one is the generation of a **planar model**. This model contains the OML and the datums of the ribs and spars. The location of the spar and ribs will affect the actual size of these components. The placement of the rib datums was already explained in 4.2.1, leaving the placements of the spars. This can be done by giving a chordwise location of each spar, however, as was explained in Section 3.3.3, it is more optimal to place the front spar as much forward as possible and the rear spar as much aft. For the front spar, the tool computes the minimum required distance from the hinges, as is shown in Figure 19. The rear spar is placed as much aft until a critical manufacturing height is found, which can also be seen in Figure 19. Once

the locations of the ribs and spars are known, one can start looking at the true size of the spars and ribs.

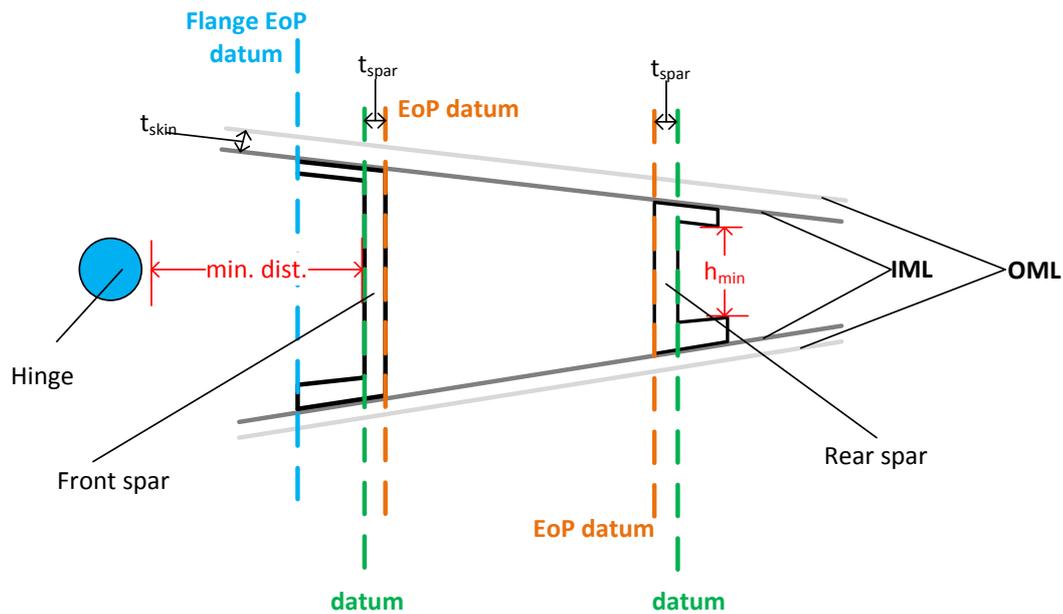


Figure 19: Placing of the front and rear spar datums and the corresponding EoP datums

The next step is to determine the actual size of the spar and rib webs. The datums in the planar model only represent a 2D plane while a skin, rib or a spar is a 3D shape. This can affect the size of, e.g. the web of a rib, as is depicted in Figure 20, a step-by-step explanation follows below. For the generation of the 3D geometry, as is illustrated in Figure 19 with the orange lines, additional planes are required that identify the End of Part (EoP). These will determine the true outer dimensions of the rib. For the spars, these planes are placed at a distance t_{spar} (spar thickness) inwards. The skin thickness has the same effect, but this process is a bit more complicated, since the skin has different thickness at different locations. Therefore each of skin sections of the OML is placed inwards with its **local** thickness, this results in the so called Inner Mold Line (IML). The whole process on how the spar and skin influence the rib can be seen in Figure 20. Note that this is an exaggerated view. The actual difference between the two rib webs is a lot smaller. However, it is required to increase the fidelity of the model. From this, the actual size of the webs can be determined.

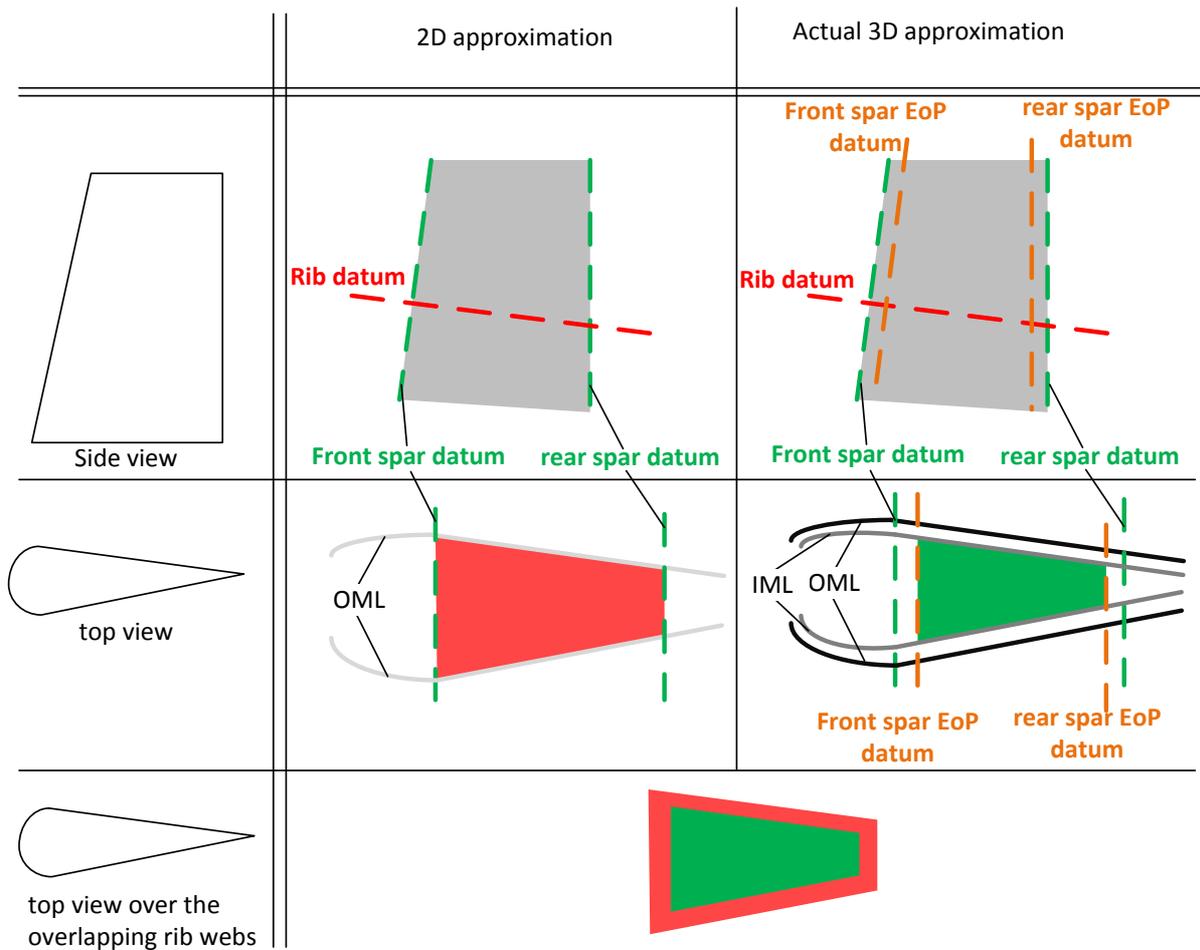


Figure 20: Effects of the End of Part (EoP) planes on the actual rib size

One last step is required to determine the outer dimensions of the web. Since the ribs (the web and flanges) are formed out of one continuous laminate, the web corners are adjusted to comply with manufacturability. Hence, as depicted in Figure 21 **step B**, corner cut-outs are made in the web.

The next step is the generations of the planar flanges, **step C.1**. First, the EoP of these flanges are determined, as can be seen in Figure 19 with the blue planes. These flanges are placed at a distance x_{flange} from the datum, where this distance depends on two aspects. One is the fastener type that is used and the second is the material of the flange. However, the dependency is not completely integrated and therefore constant sizes for the flanges are used.

From this, the exact outer sizes of the rib web and rib-flange can be obtained, The final step is performing the same process for the inner side of the rib, seen in **step C.2**. Note that the actual distance between the inner and outer rib (**C.1.** and **C.2.**) is actually really small (less than 2 mm). This can be done by placing the EoP of the spars and the skin IML even further inwards ($t_{\text{spar}} + t_{\text{rib}}$ or $t_{\text{skin}} + t_{\text{rib}}$). From this, a solid rib can be generated as shown in **step E**. Finally, edge radii are applied, see **step F**, and tooling holes are added, **step G**. This can be done easily using ParaPy as it has built in functions (FillettedSolid, and Holes).

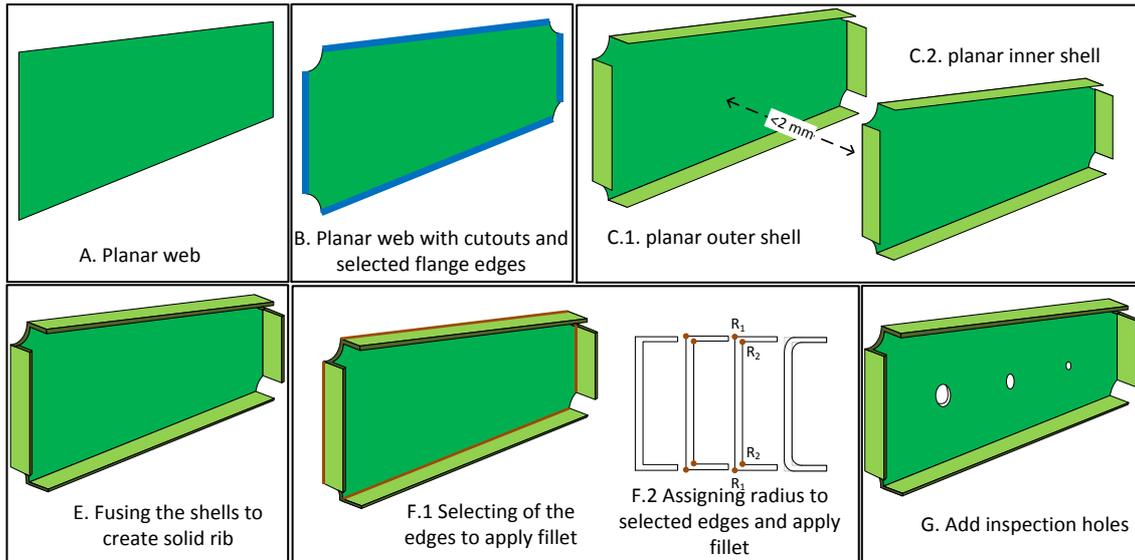


Figure 21: Overview of the required steps for a high fidelity rib geometry

One of the last functionalities of this module is the pre-processing for the generation of Finite Element Mode (FEM) that takes place. Here, a dictionary is created containing all the faces of the structural elements, using the face evolution, which is introduced by Kulkarni [9]. With the concept of face evolution it is possible to keep track of the origin of the faces after they have been split or modified during other geometry manipulations. Knowing the origin of each face allows the sub module Laminate Property Zone to obtain the following data for each of the faces;

- It will find the correct laminate from the ply library.
- It will calculate the material properties of this laminate.
- It calculates the weight of this (sub-)element.
- If FEA is performed, it can retrieve the stresses, strains and normal vectors for the current property zone and can calculate the MS per element and/or node in the component, a summary on how the MS are calculated for the material strength and buckling analysis can be found below.
- Visualization of the MS to allow for quick analysis, see Section 4.2.7

Material strength analysis

The following steps are taken to obtain the MS for the material strength analysis. It starts with retrieving the strains per mesh elements from the Finite Element Analysis. This is done using the PUNCH reader and creating the FEA result dictionary. Then, for each mesh element, and per selected load cases the applied principal strains are calculated using equation 2.

$$\epsilon_{max}, \epsilon_{min} = \frac{\epsilon_{xx} + \epsilon_{yy}}{2} \pm \sqrt{\left(\frac{\epsilon_{xx} - \epsilon_{yy}}{2}\right)^2 + \left(\frac{\gamma_{xy}}{2}\right)^2} \quad (2)$$

Next, per property property region, the selected material properties are determined. If, e.g. the spar_8 laminate is chosen, the first step is finding this laminate in the ply library, then

obtaining all the plies that are used and the individual ply properties. Using this stacking sequence definition and the material properties per ply, the cumulative laminate properties and allowables are determined. From this the allowable envelope is constructed, using the four strain allowables and laminate poisson's ratio. An example of this envelope, and how this results in a allowable strain can be seen in Figure 22. The MS can then be found by dividing the vector length of the allowable by the applied strains minus one, see Equation 3.

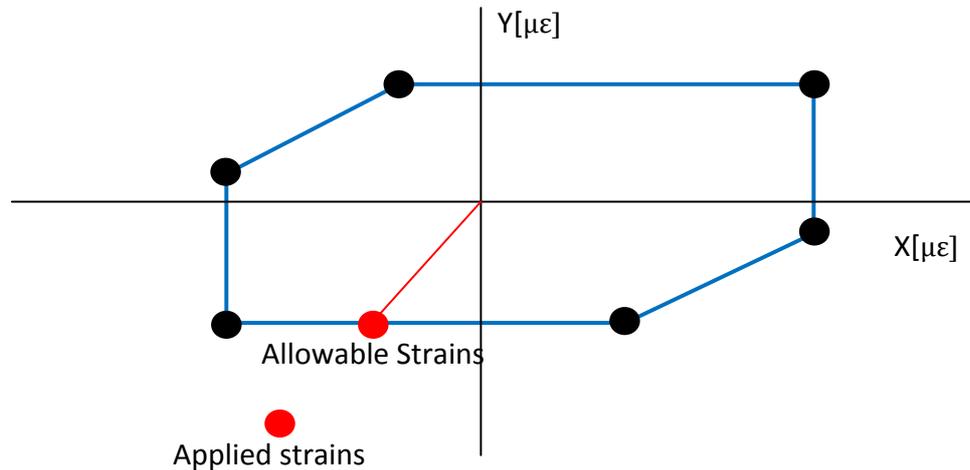


Figure 22: Example of allowable envelope

$$\text{Margin of Safety} = \text{Reserve Factor} - 1 = \frac{\text{Allowables strains}}{\text{Applied Strains}} - 1 \quad (3)$$

Simplified buckling analysis

The following steps are taken for the simplified buckling analysis. Note that this analysis runs a **linear** buckling analysis, NASTRAN sol 105. This is a faster and less sensitive solver, but is less accurate. After the FEA is completed, the critical eigenvalues per node are retrieved from the PUNCH file, using the PUNCH reader. Next, the allowable of the property zone is determined; these can be different for different structural components. Their value is confidential, but a list of the different components can be found below.

- C₁ for ribs and spars
- C₂ for normal loaded skin sections
- C₃ for heavily loaded skin sections (stricter requirements for buckling around actuator hinges)

Then the MS can be calculated by dividing the eigenvalue by the allowable minus one, see equation 4

$$\text{Margin of Safety} = \text{Reserve Factor} - 1 = \frac{\text{Eigenvalue}}{\text{Allowable}} - 1 \quad (4)$$

In this section the core of the rudder generator was discussed. One of the main strengths of developing a KBE application can be seen in the synthesis of the components. Currently,

similar steps are performed manually. As a result they are repeated for all the ribs and spars which can take hours, whereas the rudder generator can generate the same geometry within 2 minutes. Additionally, using the concept of face evolution and using the Laminate Property Zone sub-modules, the FEA results can be automatically analysed, whereas the current design approach required many manual interactions of coupling the different Excel tools, see Section 2.1.3. Note that this module only contains the post-processing of the FEA, the actual FEA is done the structural analysis module, which is explained next.

4.2.5 Structural analysis module

The Rudder Generator can generate the Finite Element (FE) model automatically. Just as for the detailed geometry generation, many manual, repetitive geometry manipulations are required in the current design process. As a result, this process can take up to 10 days, whereas the Rudder Generator performs this process in less than a minute. The methodology on how to automate the FE model generation can be found in the research of Kulkarni [9], as the same meshing tool wrapped in ParaPy was used and similar steps were taken. Note that there are some differences between the FE model generated with HDOT [9] and the rudder generator, where some of these increase the fidelity level of the tool and others are limitations, hence lowering the fidelity level.

Automation of FE meshing has been done in many cases, e.g. van Tooren [22] and Kulkarni [9]. Until now, however, they do not account for additional requirements regarding composites. The RG is capable of doing this. It does have some limitations, e.g.; the mesh is still unstructured, as a result the stresses in triangular are overestimated; the hinge brackets are still represented by a few rigid bars, hence the forces acting in the bracket are introduced in a small area of the front spar; and corner cut-outs are not yet accounted for, which causes high stress peaks at the intersection of the spar, rib and skin mesh. More information on how this affects the model can be found in Appendix C.

The mesh is used for the FEA, which is performed with NASTRAN. The mesh, load cases and the material properties are used to generate a Bulk Data File (BDF). This is a required input for NASTRAN. These individual load cases in the BDF contain information about their required output format, e.g. whether it should retrieve the forces working on the hinges (required for hinge sizing), retrieve the stresses and strains of each element (required for material failure analysis) or retrieve the eigenvalues, eigenvector and eigenmodes for each of the nodes (required for the simplified buckling analysis).

After the FEA is performed, this module contains the functionalities of reading the PUNCH files, this is one of the output files from NASTRAN. It can link the data from the punch files back to the mesh elements in ParaPy, hence keeping the dependency. The punch file is transformed into a dictionary, with the same layout of the face dictionary, created in the previous module. For each of the separate surfaces, e.g. a skin pocket, it will contain a list of load cases, where each of these load cases will contain the stresses and strains for all the elements in the skin pocket or the eigenvalue and eigenmodes of all the nodes in this skin pocket. The Margin of Safety calculations itself take place in the Laminate Property Zone, the aforementioned sub module in the rudder structure generation module (Section 4.2.4).

It is required that the rudder can withstand all the load cases, meaning that the MS should be above zero. However, a rudder where all the MS are exactly zero does not mean that it results in the best design. The best design depends on the cost and weight, these are calculated in the following module, the weight and cost estimation model.

4.2.6 Weight and cost estimation module

During the extent of this research, both a weight and cost estimation module were developed. The modules account for standard parts and non-standard parts. Here standard parts are parts that are taken off-the-shell, meaning that they are not developed in house, e.g. rivets. For these parts the weight and cost can be found using the standard part libraries. Non-standard-parts are parts that are not fixed in size, e.g. a rib or a spar, as a result calculating the weight and the cost proposes a bigger challenge.

Weight estimation module

The weight of the rudder non-standard parts can be calculated on three different levels of detail.

- Level 0: Component is represented by a planar web only, weight is found by multiplying smeared thickness by area. (depicted in Figure 21 A)
- Level 1: Component is represented by a planar web; with tooling holes and cut outs and planar flanges if applicable. Weight is found by multiplying smeared thickness by area (depicted in Figure 21 C.1)
- Level 2: Detailed 3D represented is generated according to design and manufacturing rules. Weight is found by multiplying volume by density (depicted in Figure 21 G)

For the optimization, the highest level of detail was chosen as the best possibility. Table 8 shows the total time for the experiment (weight and structural analysis), for a rudder with 15 ribs, and amount of time required computing the weight of one rib, **after** the necessary interactions for the structural analysis are performed. Therefore, the time required for the weight calculation of a rib can be correctly analyzed as they are not influenced by actions that have to take place for the structural analysis. From this, it can be seen that the weight analysis for a high level of detail rib takes about 4 seconds longer, which results in about 30% extra total time. However, due to the large error margin in the lower detailed weight estimations, this extra time was accepted to achieve the higher fidelity. Next to that, it was observed that the concept of reusing knowledge can reduce the run time while keeping the highest level of detail. Since the detailed geometry of the ribs and spars is controlled by the following four variables:

- Spar thickness: thicker spars reduce the chordwise length slightly.
- Location of the rib in spanwise direction (OML get thinner towards the outside), though due to the equal distance rib placement, the possible locations are limited.
- Skin thickness around the rib, most often equals to seven plies thick laminate.
- Material of the rib itself, which can be a four or six plies laminate

Overall, there are somewhat limited combinations for the shape of the ribs. Therefore, a functionality was added that accesses a database and tries to obtain the mass of the rib depending on the aforementioned four variables. If it can find the rib in the database, it will take that mass, otherwise it has to create the detailed geometry (as was explained in Section 4.2.4), computes the mass, and then stores it in the database. This can reduce the time required for a Level 2 detail rib to less than 0.01 second if the rib has already been computed. Hence, reusing the knowledge from previous, analyzed designs can reduce the amount of time required while obtaining the same level of accuracy.

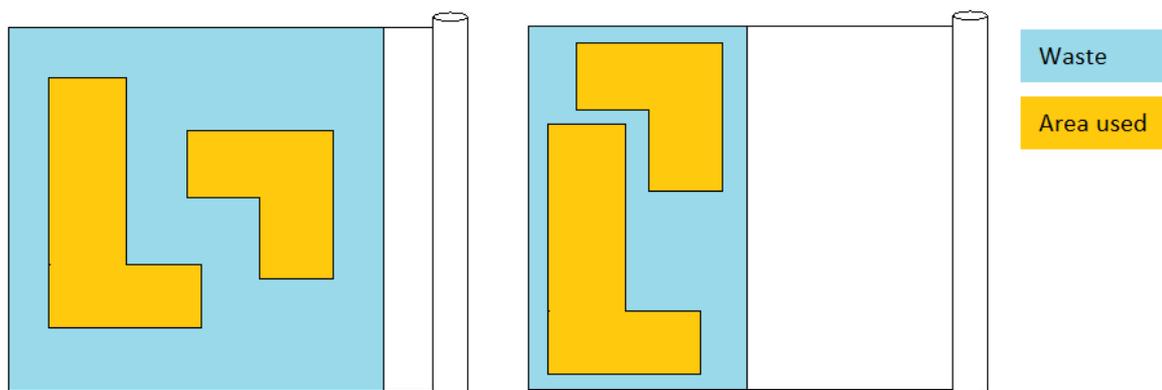
Table 10: Effect on computation time for rib mass for different level of details

Level of detail	Total time	Time per rib (s)	Error margin
0	159	0.1	15%
1	189	2	7%
2	219	4	0%
2 knowledge reuse	158	0.01	0%

Pre-processing of the cost estimation module

As the cost module was under development during this research, the methodology behind the cost estimation module can be found in research from J. Page [8]. This module requires a specific set of inputs. Therefore, the functionality of **face grouping** was added, which shows how a typical usage of KBE, and is discussed below.

To determine the cost of the materials and optimize for this, the outer dimensions of each piece of ply has to be known before the optimal placement can be determined. An example on how these outer dimensions can help to reduce cost is shown in Figure 23, note that the selected shapes are a result of the example presented below. If sizes are unknown it has to be estimated with an extra safety margin, hence they are usually overestimated. However, to obtain the exact shape of each ply can be a tedious process. For the skin, an engineer would have to determine which plies exist in which skin sections, and whether these plies are continuing in other sections. Overall, this contains many manual and repetitive steps thus restricting the possibility designing rudders faster.



a) Exact outer dimensions are unknown

b) Exact outer dimension are known

Figure 23: Example of optimal shape position when on area used and area wasted when

For example, when considering a rectangular plate consisting out of 16 different sections, where for each section two different laminates can be selected, as is illustrated in Figure 24.

It can either be a two plies laminate or a one ply laminate with their unique Ply Identifiers (PID): **1.01** and **1.02**. To estimate the cost, one would need to know the outer dimensions of each of the plies, which can be obtained quickly using face grouping. The desired result is shown in Figure 24 at the bottom, where the results of the 1.01 ply is continuous all over the skin and the 1.02 ply exist out of 2 pieces. Doing this process manually is rather straightforward, but requires a lot of time. In addition, this process has to be done multiple times for more than 16 sections, thus rapidly increasing the time required for this process. Using face grouping, this process is done within less than a few seconds, and allows the cost module to determine the optimal placement.

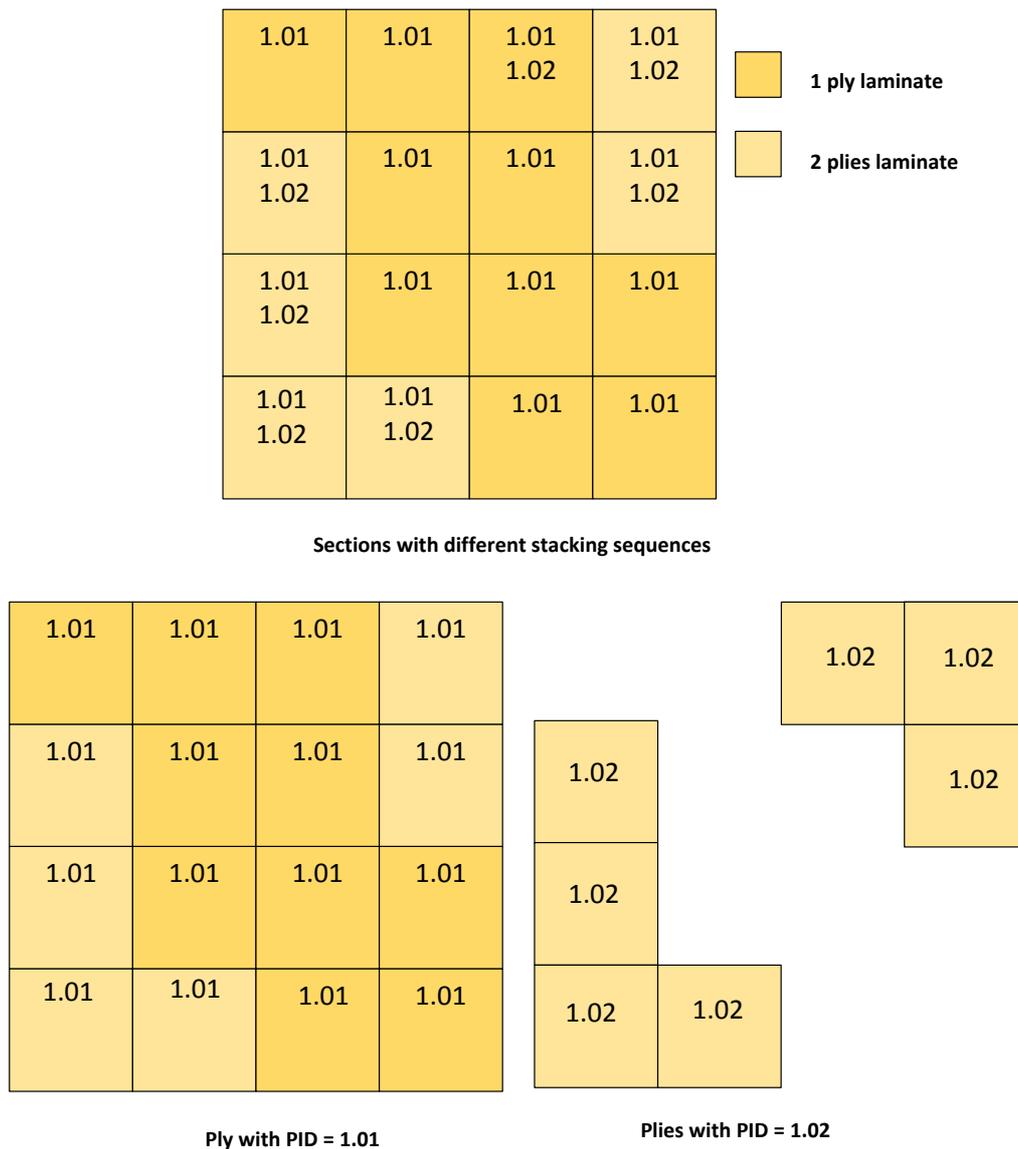


Figure 24: Example laminate distribution on a skin and the corresponding ply shapes

A quick summary of how this functionality works is present below to give the reader an idea how some of the steps that take place, a more elaborate explanation can be found in Appendix D. The functionality starts with a random **section A** and random **ply X**, then

checks the adjacent sections (**section B** and **section C**) if they contain the **ply X**. Lets say that, **section B** contain **Ply X** and **section C** does not. This function will realize that **Ply X** is continuing in **Section B** and will check the adjacent sections to **Section B** and ignores the adjacent sections of **Section C**. It continues this process until none of the adjacent sections contain **Ply X**. Then it restarts the process of selecting a random section and ply, which has not been visited before, until all the plies in all the sections are considered.

This functionality ensures that the cost module can estimate the cost of the rudder, as can be seen in the research of in J. Page [8]. Though, since the module was under development extend of this research, the cost analysis is not included. It is, however, important that the rudder generator can use the cost estimation module to perform multi-objective optimization in the future, which would have had a lower fidelity without face grouping.

Although the synthesis and analysis of the rudder is now complete, one last step remains. Where the input handler module has to deal with different parameterization, the tool should also provide the required output. As this information can be scattered throughout the tool, the last module collects this information. This step is required to perform optimization, but also makes the tool more accessible.

4.2.7 Output collection module

KBE is a relatively new concept, therefore engineers in the current industry are often sceptical about the usages of KBE tools. Often the word black box is named in these discussions. The output collection module was added to meet the engineering demands on this level. Although, this module is mostly unnecessary during the optimizations, it provides the engineer with a more detailed and familiar output.

The black box concept comes from the idea that the KBE tools can only show whether the rudder is strong enough, what the total weight and total cost is and do not indicate how these values are obtained. Although one can choose to only return these three outputs, much more information is available. This module shows that in three different ways:

- Possibility to generate the weight status reports in specific costumer formats. These contain information about all standard and non-standard parts, Hence, they show how the total mass is obtained.
- Connections to current tools, such as PATRAN, although PARTRAN which is currently used to generate a mesh is removed from the loop. The FE model and the FEA results can still be visualized in this familiar software for Fokker engineers, as shown in Figure 28.
- Margin of Safety visualization, this can quickly show what areas of the current rudder structure are most critical and or overdesigned. An example of the MS visualization of the load cases visualized in Figure 18 can be found in Figure 25.

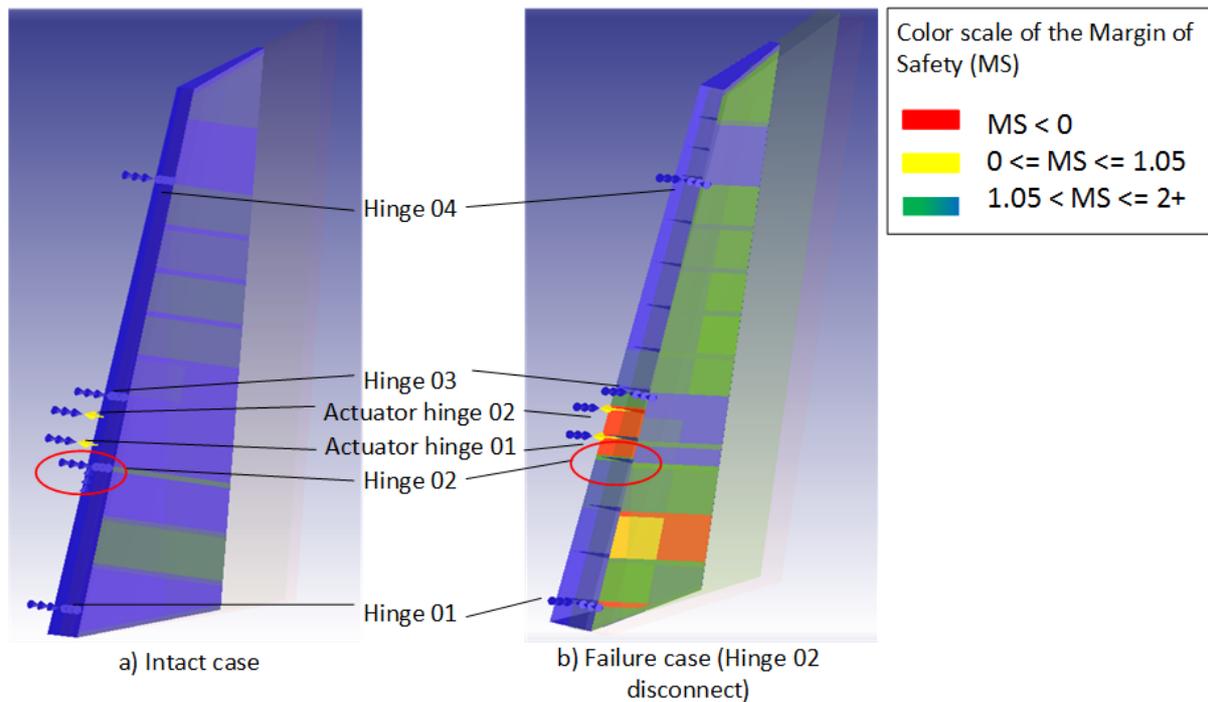


Figure 25: Margin of safety visualization of the two load cases from Figure 18

Additional functionalities are available in this module, which are required for the different optimizations. Since different optimization strategies can require different outputs, part of the module is capable of collecting these outputs. For example, to determine whether the rudder structure is capable of withstanding the loads, one can retrieve the following information.

- True or False (Pass or Fail)
- Most critical (lowest) MS in the whole rudder structure
- Most critical (lowest) MS of different functional element (One for the spars, one for the ribs and one for the skins)
- Most critical (lowest) MS of each functional element (One per rib)
- All the critical (lowest) MS per mesh element

Now that the seven main modules of the Rudder Generator are shortly introduced and for each it is shown how they can automate the design process. The manual, repetitive and non-creative steps can be avoided due to smart geometry manipulations and other build-in functions. As a result, the tool can perform a detailed rudder design within 3 to 4 minutes. It is, however, important to note that there still are some limitations. This reduces the fidelity of the tool. Though, the tool was validated with an existing rudder, referred to as **Rudder X**, to show that these limitations do not affect the proper workings of the tool.

4.3 Validation

The Rudder X was simulated in the Rudder Generator to perform the validation. For the simulation the exact positions of the rib and spar datums were obtained, though an estimated OML was used. The simulated rudder was analyzed for its weight and structural behavior, where the cost validation was performed by J. Page [8].

It was found that the weight computation capability of KRIOS has a high degree of accuracy. For the weight estimation an error margin of 0.25 – 5.3 % was found for the ribs and spars; an overview can be found in Table 11. Note that the values are given in percentages as the true weights are confidential. A positive percentage indicates an overestimation in KRIOS.

Table 11: Weight comparison between the simulate Rudder X and the actual Rudder X.

Component	Normalized Delta Weight (Compared to Rudder X)	Comment
Ribs on inboard	+ 0.25 – 2 %	From Root cap rib to rib behind actuator 1
Ribs on outboard	+ 2 – 4.8 %	From rib behind actuator 2 to tip cap rib
Front Spar	+ 5.3 %	Spar is wider due to larger OML Difference in fastener and tooling holes
Rear Spar	+ 2.2 %	Material allocation not an exact match Spar is wider due to larger OML Difference in fastener and tooling holes
Skin (left hand side)	- 8.8 %	Skin above spar flanges not accounted for
Skin (right hand side)	- 4.3 %	Skin above spar flanges estimated
Skin (right hand side)	- 8.6 %	Skin above spar flanges not accounted for
	- 4.1 %	Skin above spar flanges estimated

The difference can be explained due to the difference in OML that is used by KRIOS, as this is an estimated surface of the true OML. Looking at a top view at the root (right bottom) and tip (right top) in Figure 26, one can see the difference between the used OML and the actual Rudder X OML. The used OML is larger towards the tip, hence explaining the increasing error margin for the rib weights towards the tip, but also the overestimations of the spar weights. The difference in OML was found in a later stage and due to time constraints it was chosen to address the difference instead of repeating the validations, it is however recommended that the model is validated with the true OML. This should reduce the error margin between the actual and simulated rudder even further.

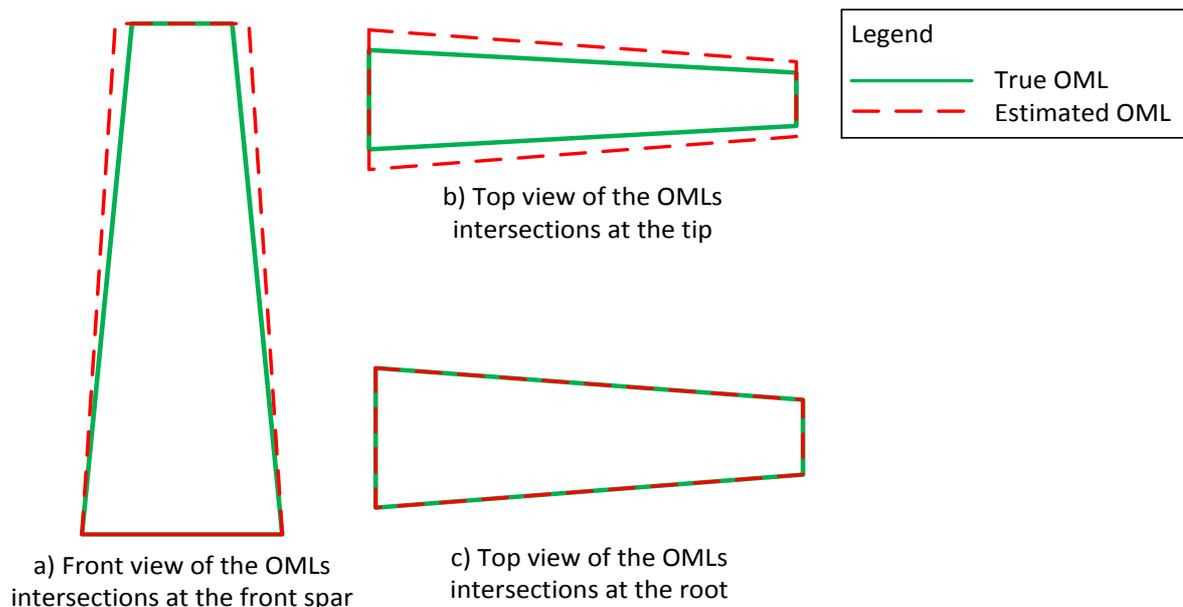


Figure 26: Deviation of the true and estimate OMLs for different views at different intersections

Weight estimation of the skin has a bigger difference at -8.8%, which can be explained as the exact weight of the Rudder X skin includes a proportion of the skin located above the front spar. Whereas the detailed design of this skin-spar interface, which is a very complex shape, is not correctly accounted for in KRIOS. Estimating the mass of these skins reduces the error to about -4.3%. So, taking into account the limitations, the tool gives a highly accurate weight compared to current weight estimations.

Even though the validations of the structural analysis could not be performed at the required level, it was shown that the analysis results are as expected. The model was compared with the stress reports of the Rudder X. Here, it was found that the model estimates the behavior of the actual model in the same way. However, the exact comparisons between the values cannot yet take place. Since not all of the failure modes are implemented, e.g. non-linear buckling and joint analysis and the FE model still has some limitations, e.g. missing corner cut-outs and simplified bracket modeling (see Appendix C). Though when looking, e.g. at the deformation of the rudder it behaves like one would expect for each of the load cases (an example for the rudder kick load case is discussed below), hence the current state was accepted for this research.

The deformation of a rudder for multiple load case was visualized with PATRAN to ensure that the rudder is responding as expected. An example of the deformation of the rudder can be seen in Figure 28, which is for a rudder kick case. The definition of this load case will be discussed first to indicate what the expected deformation would look like. During this case, an aircraft can be approaching the runway with an angle to the runway. Just before touchdown, a full rudder deflection in opposite direction is given. This means that during phase 2, the vertical tail has to provide a lifting force towards the right, causing the fin to bend to the right. Just before landing, the rudder is fully deflected to the right. At his point, the fin is still bending towards the right and the lift on the rudder is towards the left. Now looking at the displacements, the white mesh in Figure 28, one can see the effect of the fin bending, is bending the rudder structure as well, while the lift towards the left becomes visible on the skin panels which are displaced towards the left. Note that the displacements are exaggerated, the largest displacement is actually 13.6 mm (1.36+001)

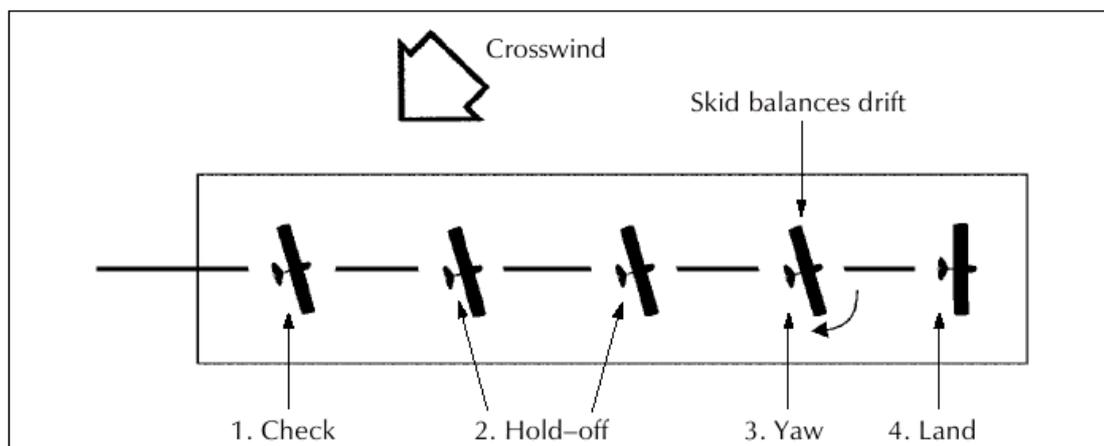


Figure 27: Rudder kick case due to crosswind during approach

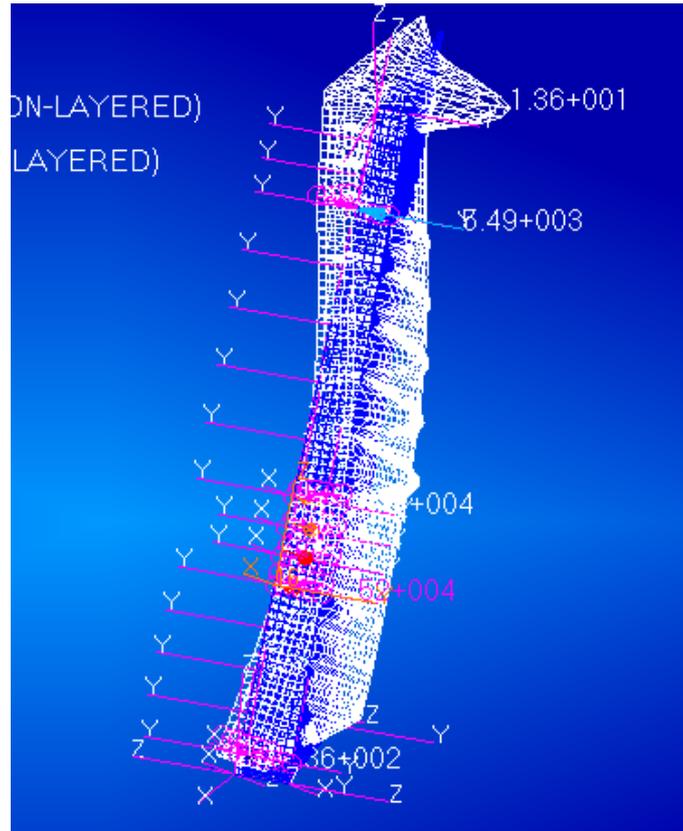


Figure 28: Displacements (white mesh) of a rudder in PATRAN for a rudder kick load case

Additionally, one can verify the response of the structural analysis by looking at the MS visualizations in Figure 25. Here the MS visualizations of an intact rudder and a hinge failure load case for the same aerodynamic force are illustrated. Once again, the position of the disconnected (failed) hinge is indicated with the red circle. Due to the disconnected hinge, one can expect higher loads at the surround hinges (Hinge 01 and actuator hinges 01 and 02). As a result, the MS around these areas is expected to drop, which can also be seen in Figure 25. Appendix E shows the effects of increasing the safety factors on the MS of the structure. The safety factor is a factor to ensure the rudder does not fail when the operating loads are reached, for intact cases these are set to 1.5 times the operating loads. The increasing safety factor in Appendix E is for visualization purpose only.

During this section it was shown how the Rudder Generator works and that can rapidly perform the detailed synthesis and analysis of a rudder. First, it was shown how using KBE can be useful for a process like the rudder design, by explained the processes that take place inside the tool. It was found that the rudder synthesis and analysis can be performed within three to four minutes, compared to the 7-8 months for the current design process. Within these minutes, the tool is capable of estimating the weight of the rudder with a 0.25-5.3 % error margin. It has some limitations on the structural analysis, e.g. the simplified buckling analysis, missing failure modes and limitations to the FE model. However, it was shown that the structural analysis responds correctly and can be used for optimization.

Part II – Rudder optimization methodology



5

The developed methodology for optimization

While the goal of this research is to design better rudders faster, having a tool capable of designing a rudder within a few minutes is not sufficient. In Section 3.3.1 it was shown that there can easily be 10^{40} different skin material allocations, whereas Section 4.2.1 showed that the material allocation for the ribs can be done in 2^{15} ways. Even with a tool capable of designing rudders in 3 – 4 minutes, it would still take around 2.25 trillion ($2.25 * 10^{12}$) years. Note that this is not including the variable number of ribs and material allocation of the spars. Hence, trying out all the possibilities is impossible and a more strategic manner is required. This can be achieved by setting up different optimization strategies.

The developed optimization methodology uses the implementation of these optimization strategies. An optimization strategy is the approach taken to obtain a good solution for an optimization problem. Hence, it includes information about parameters that can be varied, the **design variables**. A selected combination of these variables, for an experiment, form the **design vector**. All the possible design vectors, which do not violate any **constraints** mark the **design space**. The search to the best solution is guided by an **objective**, i.e., a figure of merit, depending on one or more disciplines which need to be minimized or maximized. Selection of these definitions partly solves the proposed challenges in Section 2.6, which stated how such a tool can be integrated in an optimization framework. Thus the use of optimization strategies is a good starting point.

The extra challenge that remains is ensuring accessibility of the tool and strategies for engineers. Defining the optimization strategies can require a lot of domain specific knowledge, which is often not available. So solving this problem and allowing engineers to continuously use and adjust the strategies means that the optimization strategies have to be accessible and do not require the full knowledge of the workings of the tool. Therefore, they should contain information on the Parameterization, Integration, Post-process, and Algorithm (PIPA). The first four sections, first, gives the definitions of the used aspects and after it discusses the approach on how they were selected for the different strategies. Lastly, Section 5.5 describes the criteria used to grade the different strategies.

5.1 Parameterization

The parameterization can be done in numerous ways, however, if not done correctly, they can have a huge impact on the fidelity, scope, and required run time. The parameterization contains information on **design variables** for the current optimization strategy. There are

multiple ways that they can be chosen, Figure 29 shows three different parameterizations for the skin: the first contains one variable for the whole skin, the second one uses knowledge from previous projects where Heavily Loaded (HL), Normal Loaded (NL) and Lightly Loaded (LL) sections occur, and the third generates a mesh of the skin so that each of the sections can be sized individually. In the top left of the figure, one can see the size of the design space when there are ten different materials. Here, it can be observed that the size of the design space can rapidly increase, which usually corresponds to an increase in required time to find the optimum. Generally speaking, the trade-off is made between the amount of parameters and the expected optimality.

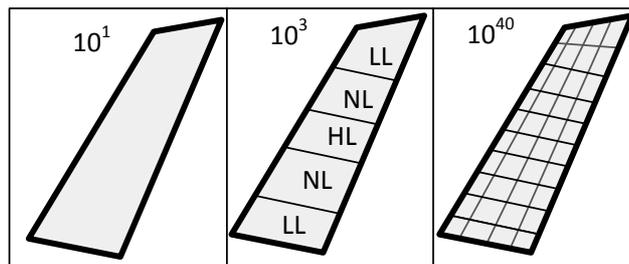


Figure 29: Different skin parameterizations and their number of possibilities.

Currently, there is no clear method on how to exactly control the tool, meaning that it is unknown what the most suitable parameterization is. Although the parameterization is not clearly defined, they can still be derived taking into account the following:

1. **Project – to – Project transfer:** by studying previous rudder designs, variables can be defined in such a way that they can simulate the same model.
2. **Knowledge capture from engineers:** based on experience from engineers, multiple trends are often seen in the design process. Formalizing these trends into parameterizations can reduce the amount of variables while keeping a required level of fidelity.
3. **Post-processing of previous strategies:** the post-processing can show what areas are overdesigned and how these are related to the parameterization. Hence, if possible they can show how a different parameterization can result in lighter design.

5.2 Integration

There are two types of integrations considered during the optimization strategies, the **design vector integration** and **tool integration**. First, the two ways of parameterization integration are discussed, and then the different ways of tool integration are explained.

There are different levels at which a parameterization can be integrated: at top level or at the tool level, where the latter can result in a tool with a lower accessibility. Since it is not always possible to directly implement the selected parameterization into the tool, some integration has to take place. As explained in Section 4.2.1, this can occur at the top level by the input handler module of KRIOS. However, it can be the case that the parameterization requires that changes have to be made in the core of the tool. These integrations at tool level

can have a negative impact on the accessibility of a strategy as it would mean that an engineer requires having a good understanding of the tool.

Additionally, there are the tools integrations, which can improve accessibility at the cost of some freedom. Integration of the tool will determine what tools are used and how they will be used for optimization. There are many possibilities on how this can be done, as is depicted in Figure 30, e.g. one can choose to manually set up and execute different experiments or use designated Process Integrations and Design Optimization (PIDO) tools. The main advantages between performing the optimization manually compared to using a PIDO tool is that the engineer is not restricted to the available optimization algorithms of the PIDO tool. Though, doing this process manually requires the engineer to have a good understanding about optimization, meaning that it reduces the accessibility of strategy.

When using a PIDO tool, as is depicted in Figure 30, one can consider the option of integrating the tool as a whole, or dividing the different disciplines, e.g. separating the torsion box into the ribs, spars, and skins. Here the main advantages of splitting up the tool can be a faster convergence of the individual structural components, however, this approach can fail due to the dependency between the components, e.g. changing the number and materials of the ribs influences the forces acting in the skin.

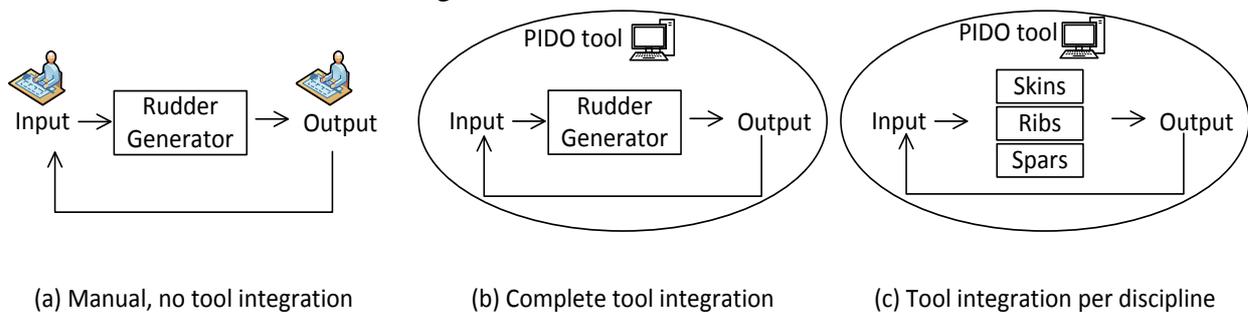


Figure 30: Different ways of tools integrations

The tools integration remains the same throughout the different strategies. Automating the rudder design was just the first part of designing better rudders faster. Part II, mentioned in Figure 3, shows that a clear approach on how to optimize a rudder is desired. As this would require the analysis of many, different designs, it is unlikely that an engineer can try a different design every four minutes. Additionally, setting up the optimization manually would require the engineer to have a lot of knowledge on optimization. Hence, a designated Process Integrations and Design Optimization (PIDO) tool, called OPTIMUS, was used for each strategy. Optimus can automate and manage the set-up and execution of simulations and analysis of engineering tasks by creating simulation workflows, according to D’Auria and D’Ippolito [23]. The Rudder Generator was integrated into this PIDO tool, which led to the complete development the **KRIOS**. A description of the workflow in Optimus is shown in Figure 31. It was chosen to not split KRIOS into different modules and allow for the separate optimization of these modules, e.g. the ribs, spars, and skin due to the high dependency between these components.

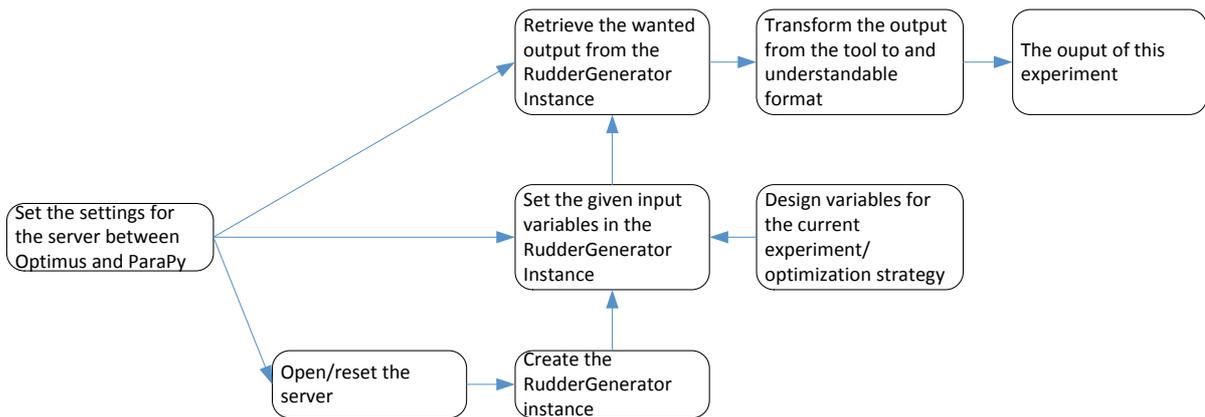


Figure 31: Flow diagram of the workflow used in Optimus

The parameterization integration changes to account for the selected parameterization. Not all the parameterizations could be directly implemented into the tool, some integration has to take place. For the parameterization integrations, it was made sure that:

1. Whenever possible, changes are made at the top level, such that this is more accessible for engineers. The integration module, discussed in Section 4.2.1 was developed to convert the parameterization to a standard input format, instead of accounting for the different input formats inside the tool.
2. For all changes that are made, both on the top level or inside the tool, they should be easily extendable to future strategies. One example of this is done rib material allocation is already discussed in Section 4.2.1. Another example is the skin material allocation, currently the left and right side are controlled by the same variables. However, it is expected that different materials can be used for both sides. Hence, a placeholder was added that checks if different inputs for the left and right skin are given.

Overall these type integrations ensure that the strategies remain accessible, meaning that they are easy to use for future optimization strategies or in different projects.

5.3 Post-processing

The third aspect, post-processing, depends on the **objective**, **constraints**, the selected **algorithm**, but also the **preference** of an engineer. The post-processing is defined as the output that is expected from the tool. In case of a rudder, this could be the weight, cost and the critical MS. However, it is possible that an algorithm requires more feedback about the MS of different components or that this could contain useful information to engineers regarding risks or points of interest. However, retrieving unnecessary information will result in a data overflow and hides important aspects of a strategy, e.g. if all the MS of all the 10.000+ mesh elements are retrieved, it is difficult to identify the most critical structural component. Thus, it is important that the correct information becomes available, to support both the algorithm and the engineer during optimization.

The post-processing is an important aspect, to perform the trade studies on the strategies. Three different aspects are considered to acquire the correct information for post-processing:

1. The objective and goal of the strategy should become clear. The objective remained the same for each strategy, which is weight optimization.
2. To study the effect of the selected parameterization, the critical MS were grouped such that they represent one of the parameters as closely as possible, e.g. if one material is given for the whole skin, the critical MS of the whole skin is retrieved, if different materials are selected for different skin sections, the critical MS of all these sections are retrieved. These MS policies could then be studied to identify points of interest for new parameterization in future strategies,
3. One last step, which does not affect the optimization, is the manual validation that takes place after each optimization. Here, the tool was run manually with the optimal design and changes in the parameterization were made to check whether a more optimal solution was found easily. In addition, the critical components were also studied to check for possibilities for future optimizations.

Using these three aspects should help to quickly identify the required outputs of the tool. In addition, the last point resulted in a clear overview of the MS of each of the components as was discussed in Section 4.2.7.

5.4 Algorithm selection

The algorithm selection can have a huge impact on the effectiveness of a strategy, in terms of time and objective value. However, the possible options are limited. The optimization of composite rudder structure is rather complex. This is because it can contain both discrete and continuous variables and gradient information is often unavailable. Hence, it is common to select non-gradient based algorithms as a Gradient Based (GB) algorithm will often get stuck at a local optimum or might not work at all. Other criteria on the algorithm selection can be found in the post-processing, e.g. the difference between single and multi-objective optimization. Finally, the selection can be limited to the available algorithms in the PIDO tool.

The algorithm selection is limited due to the previously set PIPA aspects. Firstly, the selected PIDO has limited algorithms available. For the post-processing it was said that the objective is weight optimization, meaning that it is a single-objective optimization problem and does not require the usage of multi-objective algorithms. Looking at the selected parameterization, it can be found that it is a discrete problem and hence the algorithm should be capable of handling this. Finally, gradient information cannot easily be obtained, meaning that Gradient Based (GB) algorithms will not suffice. Hence, the algorithm chosen has to be available in Optimus, is single objective, non-gradient based and can handle discrete variables.

5.5 Criteria for the different strategies

Comparison of the different strategies requires a clear set of criteria, which should be straightforward and provide advice on when to use a strategy. Some of these criteria can be easily derived from the objective, others are less obvious and should account for the changes made in one of the PIPA settings. The following criteria were selected, a summary of the criteria and their weight factors can be found in Table 12:

- **Objective value:** The final object value of the strategy, this will show how effective every strategy is in order to find a lighter design. Note that due to confidentiality the weight is normalized by simulating the Rudder X in KRIOS, a previous design at Fokker. Since this is the object of the strategy, which will say how much better a rudder is, a weight factor of 15 will be assigned to this.
- **Run time:** The time it takes and the amount of experiments required to obtain an optimum. Both time and number of experiments can be taken as a measurement, but they should be taken into account carefully. Time for all or one of the experiments can depend on the computer's computation speed. This research performs optimization on different computers; hence the time required per strategy is affected by this. Additionally, when considering the number of experiments, it can occur that some experiments are considered infeasible, hence they take about 5 seconds compared to the 3+ minutes. Luckily, these experiments can be differentiated. As the main goal was to design a better rudder **faster**, the weight factor assigned to this criterion is 5.
- **Accessibility:** Ease of using a strategy. If, in the future, an engineer wants to use this strategy for a different design, how many changes are required by the engineer and where do they have to occur? It is important that no major changes have to be performed inside the tool, although this can sometimes not be avoided, the weight factor assigned to this criterion is 2.
- **Fidelity:** In how far can the selected strategy represent the real object? This can both say something about the validity of the found optimum and the maximum detail of the object. The former can depend on whether a strategy is abusing missing requirements or constraints in the tool. The maximum level of detail can for example occur in the skin, allowing just one material over the skin is not a good representation of the actual skin. The weight factor assigned to this criterion is 2.
- **Risk:** Chances of the strategy finding the global or local optimum. First of all, while one can never be sure whether the global optimum is found, there is still a possibility that an engineer can quickly find a better solution if the strategy performing rather poorly. Additionally, an engineer might already know a better solution. The weight factor assigned to this criterion is 2.

In this section both the theory and approach behind developed methodology were discussed. Here the solution can be found in defining optimization strategies. These should be split up into the four different PIPA aspects. The implementation of this can be found in the next section, where multiple strategies are defined, explained and implemented. Then they

are compared using the trade-off table shown below, to obtain the associated benefits of each strategy.

Table 12: Trade-off table template for the different optimization strategies

	Objective	Run Time	Risk	Accessibility	Fidelity	Total Score
Weight Factor	15	5	2	2	2	
Optimization Strategy						
A: Response of the model						
B: Higher fidelity optimization						
C: Validation of the optimum and algorithm settings						
D: Tailored skin parameterization						
E: Quadratic skin distribution						
E: Improved quadratic skin distribution						

6

The optimization strategies

The goals of designing **better** rudders **faster** can both be achieved by implementing the right optimization strategy. The selected Parameterization, Integration, Post-processing and Algorithm (PIPA) settings for each of the strategies can have an influence on the final objective value found and the time required to reach this objective. If done incorrectly, the strategy will not be able to get to the “best” solution or it takes unnecessarily long to get there.

This section presents the practical implementation of different optimization strategies. This includes how the different PIPA settings were selected and implemented, followed by the analysis of the results and how they can affect future strategies. The main idea of each step is to increase the effectiveness of the strategies. However, it should be noted that the goal of some of the strategies was the validations of the results of the tool and the found optimum. Lastly, the overall performances of each of the strategies are compared and a trade-off is made.

6.1 Strategy A: Response of the model using directive search

The first strategy is a simplistic one and was used to assess the response of the model and can help to reduce the design space. The goal of this strategy was to check the response of the model inside the framework. Therefore, only a few variables were selected and a directive search algorithm was chosen, meaning that the variables were optimized one by one. For example, one can start with maximum values for all the variables and start with, e.g. removing ribs until the structure fails, then the next variable is reduced, for example, the skin thickness, until the critical MS is below zero. This is a rather simple way of optimizing a rudder, but is unlikely to result in an optimum because after optimizing the first variable, part of the structure contains a critical area with a MS close to zero. As a result, there is already critical area in the rudder which can fail when reducing the other variables slightly. Thus, only the first variable is optimized, while the other variables cannot really vary. So this approach can quickly show the responds of the model, but is not the best approach for rudder optimization.

6.1.1 Implementation

- P** The parameterization of the first strategy entails four variables, where each have their own boundaries. The first three are the number of plies (indicated with N) of the laminate of the spars, ribs and skins. The laminates are taken from the

corresponding ply libraries, as was explained in Section 2.3. The last variable is the number of ribs (N_{ribs}), which has to remain within certain boundaries, since there is a minimum required rib pitch.

- I** For the integration of this strategy, no changes had to be made to the model. As this input was already implemented in the tool.
- P** To obtain the correct response of the model, four different MSs are obtained for post-processing. These four were selected such that they reflect the change of variables. They are the critical MS of both spars, left and right skin, fixed ribs, and the additional ribs.
- A** As the main objective is to check the response of the model and the number of variables is limited, a manual directive search is performed. This allows one to quickly identify the changes in MS and weight when one of the variables is changed, e.g. adding a ply to the skin overall leads to an increase in weight and the Margin of Safety (MS) of the skins.

This leads to the following optimization problem description:

Objective:	Minimize	$\frac{W_{rudder}}{W_{rudder_x}}$
W.r.t.:	N_{ribs}	$\in \{x \in \mathbb{Z} \mid 8 \leq x \leq 18\}$
	N_{ribs}	$\in [4, 6]$
	N_{skins}	$\in \{x \in \mathbb{Z} \mid 5 \leq x \leq 16\}$
	N_{spars}	$\in \{x \in \mathbb{Z} \mid 4 \leq x \leq 12\}$
S.t.:	MS_{min}	≥ 0

Here N_x is the number of plies of the composite component x , the MS_{min} the minimum MS in the whole rudder. Although multiple MS are retrieved, as long as the MS_{min} is above zero the design is valid. Hence it is only subjected to (s.t) the constraint of $MS_{min} \geq 0$. With the optimization problem set up, the variables were optimized one by one, and the responds of the model was analyzed.

6.1.2 Results

The implementation of this strategy validates the correct working of KRIOS. Figure 32 shows the effect when only changing the number of ribs. As expected, the weight increases when the number of ribs increases, see Figure 32 a). When looking at the critical MS, it is expected that, overall, the critical MS values should increase when adding more ribs. The increasing MS of the spars and additional ribs can be seen in Figure 32 c) and Figure 33, respectively. Although this increase occurs in steps, which can be explained due to the parameterization, it is still as expected. The MS of the fixed ribs, shown in Figure 32 b), is oscillating. However, this oscillates between the values of 0.14 to 0.22, which is a relatively small difference. Here, the oscillation can be explained due to the complex nature of rudder structure, e.g. increasing the stiffness of a structure at Point A, can increase or reduce the

stresses in the adjacent section Point B. This is also one of the main reasons why Gradient Based (GB) algorithms might not work for these problems. So, keeping in mind the effects of the current parameterization and the complex nature of the problem, it can be said that the model responds as expected.

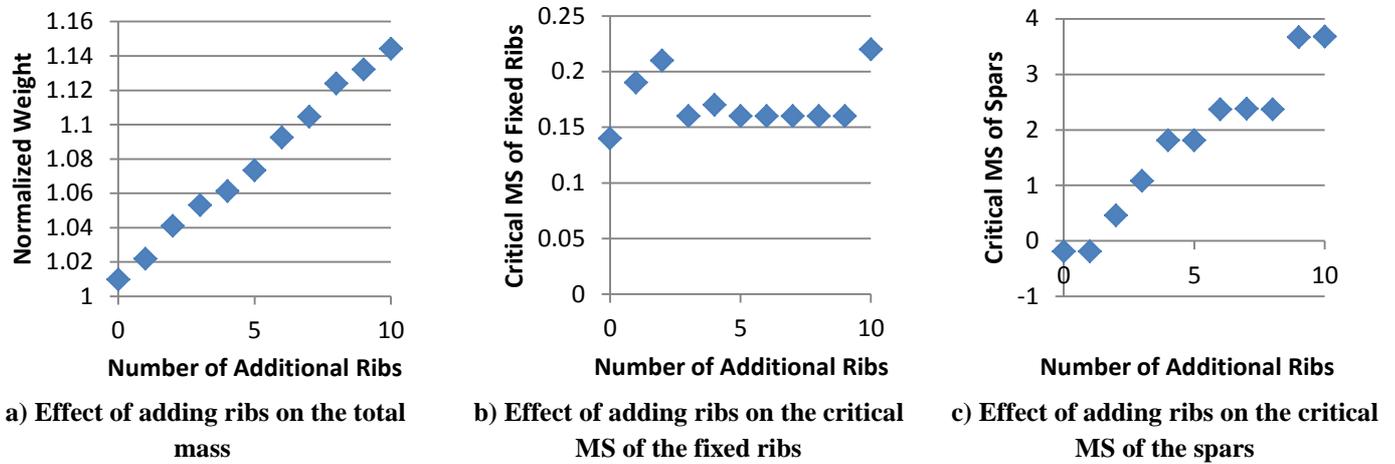


Figure 32: Results from the manual directive search.

The discrete steps in the critical MS are a result of the suboptimal parameterization for the number of ribs and indicate how a more optimal solution can be obtained. The current parameterization only allows changing the total number ribs. The tool itself will determine where these ribs are placed based, as was explained in Section 4.2.1. As a result, it is possible that ribs are placed in an uncritical area. When looking at Figure 33, this effect can be seen for the critical MS of the added ribs. To illustrate this, imagine that the rudder is divided into a top and bottom part and the critical MS occurs in the top. Adding a rib in the bottom is unlikely to have a (big) effect on the critical MS. Hence, changes are only visible when ribs are added at the correct locations. Therefore, the parameterization needs to be changed to account for this.

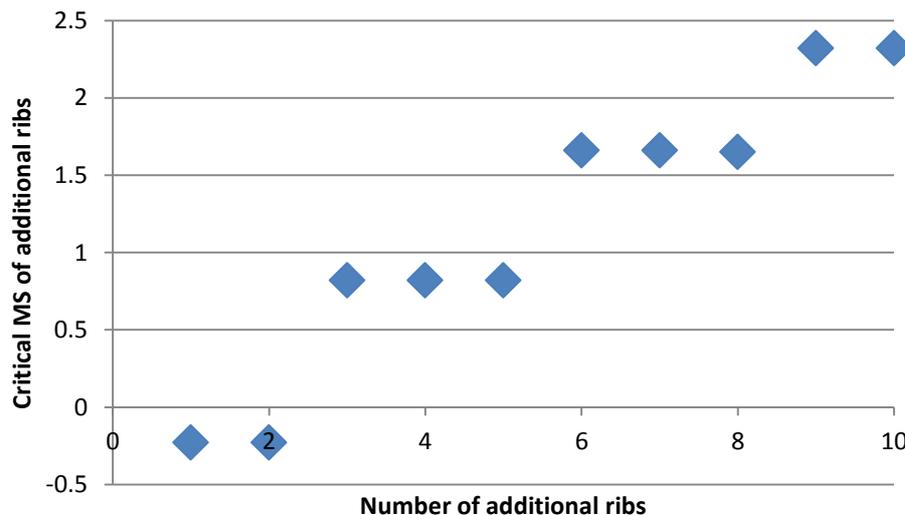


Figure 33: The discrete steps in the margin of safety of the additional ribs for increasing number of ribs

Although the directive search (DS) can quickly show the response of the model, the downside of using this approach can also be seen. After just after 35 experiments, roughly four hours, an optimum point can be found. The found optimum depends on the order in which the variables are optimized. A most optimal objective value of 0.983 can be found when starting with the skin thickness. Note that the weight is normalized due to confidentiality. The reference weight was found by simulating the previous rudder design, Rudder X, in KRIOS. It is not surprising that the found optimum depends on the order, as it is advised to only use the DS for single variable problems. Hence, Strategy B uses a different algorithm, the selections of this algorithm is explained in the PIPA of next section.

6.2 Strategy B: Higher fidelity optimization with a differential evolution algorithm

The next strategy involves a more realistic design optimization of the rudder, which comes at the cost of a higher complexity and an increase in time required to find the optimum. As the parameterization of Strategy A is limited, a new parameterization is introduced to achieve a more realistic design, this increases the number of design variables and the design space and makes it more difficult to find the optimum. Additionally, the DS algorithm is replaced by the differential evolution (DE) algorithm more suitable for multi-variable and non-gradient based optimization problems [24], but the implementation of this algorithm is less straightforward compared to a DS.

Before the PIPA settings are given, some definitions for the parameterization are introduced. Figure 34 illustrates these definitions, where the left illustration shows how bays are defined and the right illustration depicts the corresponding skin sections. A bay is defined as the area between the root-cap rib and hinge rib, two hinge ribs, or hinge rib and tip-cap rib. This results in seven bays when a total of six hinge ribs are required.

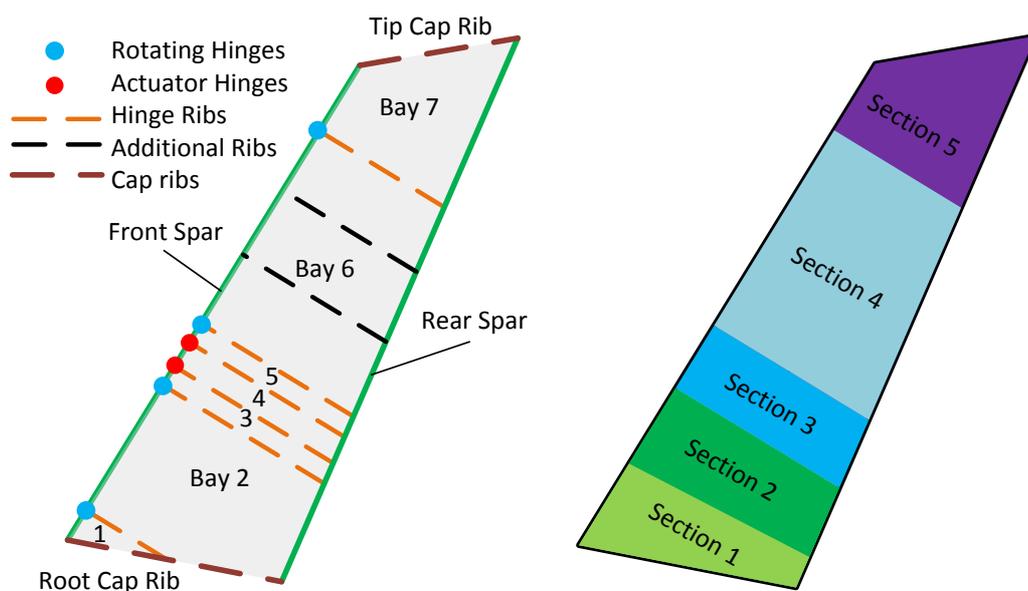


Figure 34: Definitions for the new parameterization

6.2.1 Implementation

P The number of variables for this parameterization is increased from four to sixteen these changes are based on results from the previous strategy, knowledge from previous project, and logical decisions. The old variables and how they are changed can be seen in Table 13, whereas reasoning behind it follows.

A new parameterization for the number of ribs is required. During the post-processing of the **previous strategy** (see Section 6.1.2), a limitation of the parameterization for the number of ribs was found. Namely, that ribs were being added in non-critical bays, meaning that material was being added without improving the design. Therefore, the new parameterization prevents this by giving the tool more control on the placement of these ribs, by adding the variables of number of ribs per bay, instead of the total number of ribs. Since there are seven bays, this will add six extra variables, luckily four of do not have to be considered. This is because, as can be seen in Figure 34, the pitches of bay 1, 3, 4 and 5 are small and do not allow ribs to be placed here due to manufacturing constraints. Although the new parameterization adds more variables (from 1 variable to 3), it is required to avoid ribs being placed in non-critical area.

Additionally, more flexibility in the material allocation was added. For the skin this was done by splitting the skin into five different sections, to mimic the skin of a **previous project**, adding a total of 4 variables. The material selection for the front and rear spar is now done separately. This is a **logical decision**, since there is no requirement for both spars being the same material and they were only grouped to check the responds of the model. Lastly, the material for the ribs is selected for two groups, the heavily and normally loaded ribs, the definition of these ribs were explained in Section 3.3.2. According to Fokker engineers [10], higher loads are expected is ribs located behind hinges. Hence, this was done based on **experience of engineers**.

I As the skin was divided into five sections, to mimic the skin of the **Rudder X**, a new implementation was required. It is likely that the boundaries of the predefined sections are located between two ribs. However, for the area between two ribs, which are defined as skin pockets it is required that they only contain one composite material in spanwise direction, as was explained in Section 3.3.1. Therefore, the sections have to be adjusted, as is shown in Figure 35. Note that this is a simplified version, as only two different sections are shown. The definition of the sections is illustrated on the left, whereas in the middle illustration two ribs are added. Here, it can be seen that the middle skin pocket is located in the two different sections, when different materials are assigned to both section, a spanwise drop-off would occur between two ribs. This is not allowed. Hence, the added functionality enforces that the skin pocket will be part of either section one or two, e.g. Figure 35 this pocket is added to section two

P The post-processing was changed such that the critical MS that are obtained are split in the same way as the variables, e.g. from one for the skin to one per skin section. This enhances the implementation of parametric studies.

A As the goal was to perform a first optimization, for a single objective discrete problem, the differential evolution (DE) algorithm is most suitable. The DE is very similar to the better known Genetic Algorithm (GA). Both have their strengths and weaknesses. The DE and the GA are metaheuristic and do not ensure that the global optimum will be found. Though they are very effective on large scaled problems which do not have gradient information. The DE is easily applied to a wide variety of real valued problems, whereas the GA uses chromosomes. As a result, the DE is easier to use and does not require the same fine tuning of the algorithm settings compared to the GA, e.g. the cross-over rate and mutation value. The DE was selected since it is easier to use, requires less fine tuning and is available in Optimus. A description of the DE can be described as follows:

1. Initialize the starting population of design vectors, if no starting population is given λ amount of random design points in the design space are selected. Here λ is the population size.
2. Evaluate the fitness of the initial starting population
3. Loop for i from 0 to λ

3.1 Select three randomly chosen design vectors from the previous population

$x_{r1}^{(g)}, x_{r2}^{(g)}, x_{r3}^{(g)}$ these must be distinct from each other and the target vector $x_i^{(g)}$

3.2 Build the weighted difference vector, v using:

$$v_i = x_{r1}^{(g)} + WF * (x_{r2}^{(g)} - x_{r3}^{(g)}) \quad (5)$$

Here WF is the weight factor, g is the iteration index

3.3 Generate trial vector by repeating for n from 1 to D

$$t_{n,i} \begin{cases} v_{n,i} & \text{if RND} < \text{CR or } n = \text{rand}(D) + 1 \\ x_{n,i}^{(g)} & \text{otherwise} \end{cases} \quad (6)$$

Here RND generates a random number with $[0,1]$, CR is the crossover rate, D the dimension of the vector and $\text{rand}(D)$ generates an integer number within $[0, D-1]$

3.4 Selection between trial vector and target vector, the fitter one (lowest objective value survives)

$$x_i^{(g+1)} \begin{cases} v_i & \text{if } f(v_i) < f(x_i^{(g)}) \\ x_i^{(g)} & \text{otherwise} \end{cases} \quad (7)$$

4. Check if stopping criteria are reached. Which are the average stopping step width and maximum iterations otherwise return to step 3 and start the next iteration

The settings of the DE, which are the Weight Factor (WF), population size (λ), CRrossover (CR), maximum number of iteration and average stopping length criteria are discussed in Section 6.3 as for now the default values are taken.

Table 13: Selection of the new parameterization.

Old variable	New Variables	Selection criteria	Comment
N_{ribs}	$N_{ribs_bay_i}$ for i in $[1,2, \dots,6,7]$	Previous strategy	During the post-processing of the previous strategy it was found that ribs were being placed in none critical areas. This new parameterizations prevents by giving more control on the exact location of the ribs.
N_{skin}	$N_{skin_section_i}$ in $[1,2,3,4,5]$	Knowledge-based from previous project	A previous rudder design at Fokker has a skin containing five sections. See the right part of Figure 34.
N_{spars}	N_{front_spars} & N_{rear_spars}	Logical decision	There is no requirement that front and rear spar should be the same material, they were only grouped to quick check the responds of the model
N_{ribs}	N_{HL_ribs} & N_{NL_ribs}	Experience of Engineers	Heavily Loaded (HL) ribs are the ribs located behind the hinges, see the left part of Figure 8. Normally Loaded (NL) ribs are the added ribs, e.g. the additional ribs in Figure 8 on left.

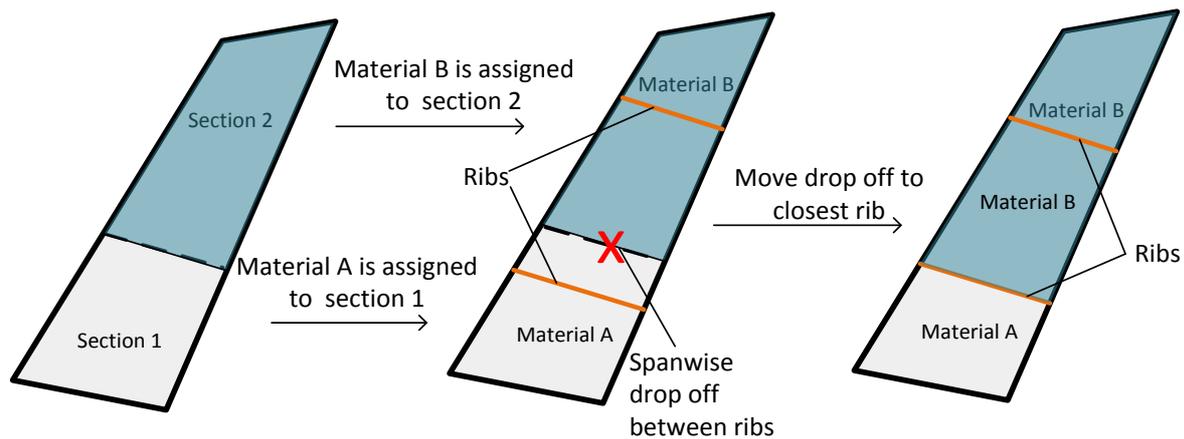


Figure 35: Example of new parameterization of the skin material allocation.

Accordingly, the following optimization problem description can be stated:

$$\begin{aligned}
 \text{Objective: Minimize} \quad & \frac{W_{\text{rudder}}}{W_{\text{rudder}_x}} \\
 \text{W.r.t.:} \quad & N_{\text{ribs_bay}_2} \in \{x \in \mathbb{Z} \mid 0 \leq x \leq 4\} \\
 & N_{\text{ribs_bay}_6} \in \{x \in \mathbb{Z} \mid 0 \leq x \leq 7\} \\
 & N_{\text{ribs_bay}_7} \in \{x \in \mathbb{Z} \mid 0 \leq x \leq 2\} \\
 & N_{\text{HL_ribs}}, N_{\text{NL_ribs}} \in [4,6] \\
 & N_{\text{skin_section}_k} \in \{x \in \mathbb{Z} \mid 5 \leq x \leq 16\} \quad \text{for } k \text{ in } [1,2,3,4,5] \\
 & N_{\text{front_spars}}, N_{\text{rear_spars}} \in \{x \in \mathbb{Z} \mid 4 \leq x \leq 12\} \\
 & N_{\text{HL_interface}}, N_{\text{NL_interface}} \in \{x \in \mathbb{Z} \mid 7 \leq x \leq 16\} \\
 \text{S.t.:} \quad & MS_{\min} \geq 0
 \end{aligned}$$

Here, k is the skin section index. This results in a total amount of 12 variables which will be optimized. Once again the only constraint is the MS_{\min} . Now that the strategy is defined, the results of the optimization are discussed next.

6.2.2 Results

As a result of the more detailed parameterization and selection of a suitable algorithm, the obtained optimum is better compared to the previous approach. However, this comes at a cost. Since a DE optimization takes place, one can see a large increase in the number of experiments. Figure 36 shows the normalized weight, for the different iterations. A total of 30 iterations took place, each containing 65 experiments, which results in a total of 1950 experiments and took approximately 4.5 days. This is a large increase compared to the four hours required in Strategy 6.1. Although the time required to reach an optimum is much longer, it results in a better objective value of 0.827 compared to the previously found normalized weight of 0.983.

The effect of using the DE with its default settings has a negative effect. In Figure 36 one can see that the optimum was found after 20 iterations, whereas the optimization continued till 30th iteration, which is the maximum number of iterations stopping criteria of the DE. Usually, this should be avoided as it means that the required convergence criteria is not yet met. Though, since the default settings were taken, this convergence criteria was too low for this optimization problem. Hence, the convergence criteria should be studied and adapted. If done correctly, the optimization could stop after less iterations and thereby reduce the time required.

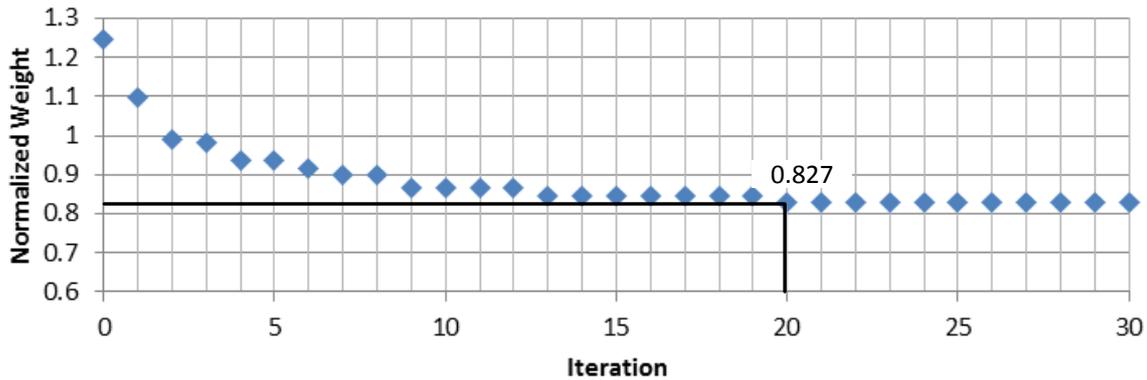


Figure 36: Normalized weight per iteration for the higher fidelity optimization

The new parameterization for the number of ribs per bay has a beneficial effect on the found optimum. When looking at the number of ribs per bay, an optimal point is found which cannot occur when only the total number of ribs is controlled. Hence, the same optimum cannot be found using the rib parameterization of Strategy A.

6.3 Strategy C: Validation of optimum and algorithm settings

It is difficult to prove whether an optimization reached a global or a local optimum, so the next step was to check the optimum found in Section 6.2. Two different approaches were taken: an Exhaustive Search (ES) on a limited design space close to the found optimum and a manual search with a design engineer. Next to the validation, some fine tuning of the algorithm setting was also performed. This was performed on the limited design space. The reason why a limited design space has to be used is explained below.

To illustrate how large the design space becomes, consider the options of the parameterization from Section 6.2. With this parameterization there are 120 possibilities for the selection number of ribs and for each of these the material selection for the ribs can be done in four ways, leading to 480 possibilities. Next to that, there are eight options for the material selection of the front and rear spar, resulting in a total of 64 combinations for the material selection of the spars. This means that each of the 480 number of rib possibilities can be combined with any of the 64 spar material selection combinations, resulting in 30720 possibilities. Lastly, each of the five sections has twelve possibilities for skin material, resulting in 12^5 , or 248.832, combinations for the skin material allocation. Combining the selection of number of ribs and the material allocation would result in $7.6 * 10^9$ possibilities. With each evaluation taking 3.5 minutes on average, this comes down to 50.902 years when using one computer. Hence, doing the ES on the full design space would not be feasible.

6.3.1 Implementation

Almost all of the same PIPA settings from the previous strategy were used, only the boundaries of the variables had to be adjusted. The design space was reduced to perform the ES and fine tuning of the DE settings, as was explained above. Therefore, the boundaries of the variables were set ± 1 of the found optimum. The resulting definition of the optimization problem can be found below. The ES will consider all the possibilities in the design space,

which are 2592 experiments. The fine tuning of the DE settings will be done on the same design space.

The manual search, which was performed in combination with an engineer, was done considering the full design space. The manual approach was performed using the MS visualization explained in Section 4.2.7. Different designs were tried based on the engineers' experience, to try to reduce weight. The optimization problem is as follows:

Objective:	Minimize	$\frac{W_{\text{rudder}}}{W_{\text{rudder}_x}}$	
W.r.t.:	$N_{\text{ribs_bay_2}}$	$\in [1, 2, 3]$	
	$N_{\text{ribs_bay_6}}$	$\in [3, 4, 5]$	
	$N_{\text{ribs_bay_7}}$	$\in [1, 2]$	
	$N_{\text{HL_ribs}}, N_{\text{NL_ribs}}$	$\in [4, 6]$	
	$N_{\text{skin_section}_i}$	$\in [5, 6, 7, 8]$	for i in $[1, 2, 3, 4, 5]$
	$N_{\text{front_spars}}$	$\in [4, 5, 6]$	
	$N_{\text{rear_spars}}$	$\in [4, 5]$	
S.t.:	$MS_{\text{min}} \geq 0$		

This results in a total of twelve variables, Note that the design space is reduced significantly to support the implementation of multiple DEs with different settings.

6.3.2 Results

The optimum found here was the same optimum as in Strategy B. While this does not confirm it is the global optimum, it gives more certainty about the found optimum. The exhaustive search around the found optimum - which contains about 2592 experiments - results in the same optimum. Secondly, the manual search approach - which uses the found optimum as a starting point - did not result in a better optimum either. Hence, while in the limited design space the most optimum point is found, this cannot be said for the full design space.

Although the found settings strongly depend on the optimization problem at hand, they still provided useful information for following strategies. The settings are often related to the optimization problem. Hence, one can derive different settings for other strategies. For example, Storn and Price [25] recommend that the population size is at least 4 times the number of variables. From running multiple DEs, the following settings were obtained.

Population size

The first study that was performed focuses on finding the optimal population size. Researchers usually argue about the optimal size, as a “small” population size could guide the algorithm to poor solutions [26], [27], and that a “large” population size could make the algorithm expend more computation time in finding a solution [28], [29]. Hence, a trade-off has to be made. This was done keeping in mind that finding the optimal population size is a difficult problem in itself [30] and a general rule cannot be applied to every type of problem

or function to be evaluated [31]. The trade-off for this is limited to the analysis of different population sizes, their success rate, and the required number of evaluations.

Each population size was analyzed five times, only changing the random seeds. These only affect the randomly made choices within an algorithm. Figure 37 shows the result of different population sizes, which includes how many of the five DEs managed to find the optimum and how many experiments it took to find the optimum. Here, one can see that for a population size of 60 ($N * 5$) and 72 ($N * 6$), where N is the number of variables, both of these DEs found the optimum 100% of the time. Lowering the population size can result in lower success rates. Note that there is never a guarantee that the optimum will be found, though a clear increase in the success rate can be seen with increasing population sizes. Looking at the number of evaluations, no trend can be identified. Both 5 and 6 times the number of variables seem suitable, but since the success rate seems to increase with increasing population size, the latter was chosen.

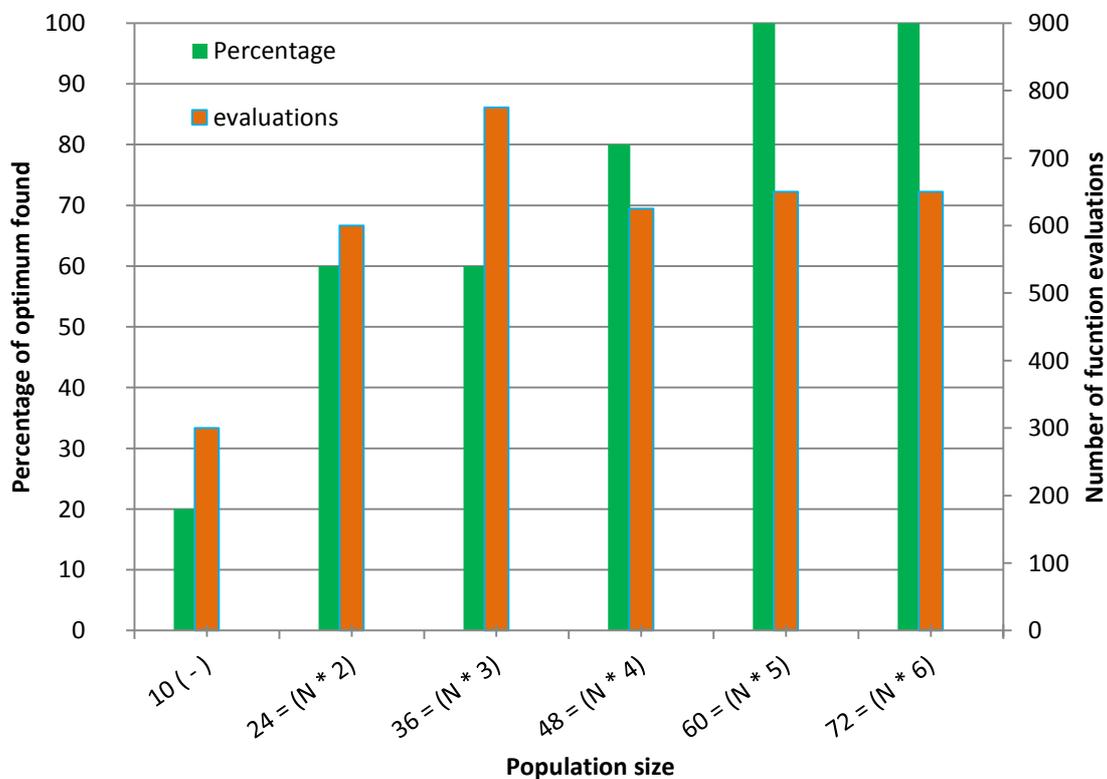


Figure 37: Success rate and number of evaluations required for DE's with different population sizes. Here N is the number of variables

Average step size stopping length

The variation in the design vectors of a population is used as the stopping criterion for the differential evolution. It is assumed that a small variation of all design vectors in a population cannot produce a large variation in the quality. Meaning that if (all) the design points of the current populations are located close to each other, no large change in the objective value is expected and the optimization can be terminated. Whether the deviation in the population is

small enough, is checked by seeing if the average step size of all the design vectors in the population is below the average step size stopping length. The average step size can be found with, Equation (8):

$$\text{Average step size} = \frac{1}{\lambda} \sum_{k=0}^{\lambda} \text{Average } \Delta_k \quad (8)$$

Here, Average Δ_k is the normalized difference of the design vector k compared to the most optimal design vector (DV_{opt}) in the starting population and can be found with Equation (9):

$$\text{Average } \Delta_k = \frac{1}{N_{variables}} \sum_{j=0}^{N_{variables}} \frac{DV_{j,k} - DV_{j,opt}}{DV_{j,upper} - DV_{j,lower}} \quad (9)$$

Here, $DV_{j,upper}$ and $DV_{j,lower}$ are the upper and lower bounds of j^{th} design variable. An example calculation of the average step size can be found in Appendix F.

The average stopping criterion per strategy is found by assuming that within the population the optimum is available, twelve design vectors are close enough and only differ minimally for one of the variables in the design vector, and the remaining design vectors only differ minimally for two variables. For this parameterization, this results in a value of 0.15, though it changes throughout the strategies.

Starting population

It has been recognized that if the initial population of a Genetic Algorithm (GA) is good, then the algorithm has a better possibility of finding a good solution [32] and that, if the initial supply of building blocks is not large enough or good enough, it would be difficult for the algorithm to find a good solution [28]. Similar to finding the optimal population size, finding the optimal starting population is difficult and, once again, no general rule can be applied [30], [31]. Multiple Design of Experiments (DOE) were selected as starting populations. These were generated using different methods and sizes, to assess the correct starting populations. Three common DOE methods were selected, the fractional factorial, the Box-Behnken (BB) and the Latin Hyper Cube (LHC). Once again, a trade-off was made between the success rate and number of function evaluations. It was found that all of the different starting populations were capable of finding the optimum, hence Figure 38 shows the number of required evaluations to find an optimum only where a, b and c show:

- Case 1-3 are three DE's with three different random seeds using the **Fractional Factorial** starting population;
- Case 1-3 are three DE's with three different random seeds using the **Box Behnken** starting population;
- Case 1-10, where each case represents the average of three DE's with three different random seeds for the LHC starting population of different sizes [72, 100, 150, 200, 250, 300, 400, 500].

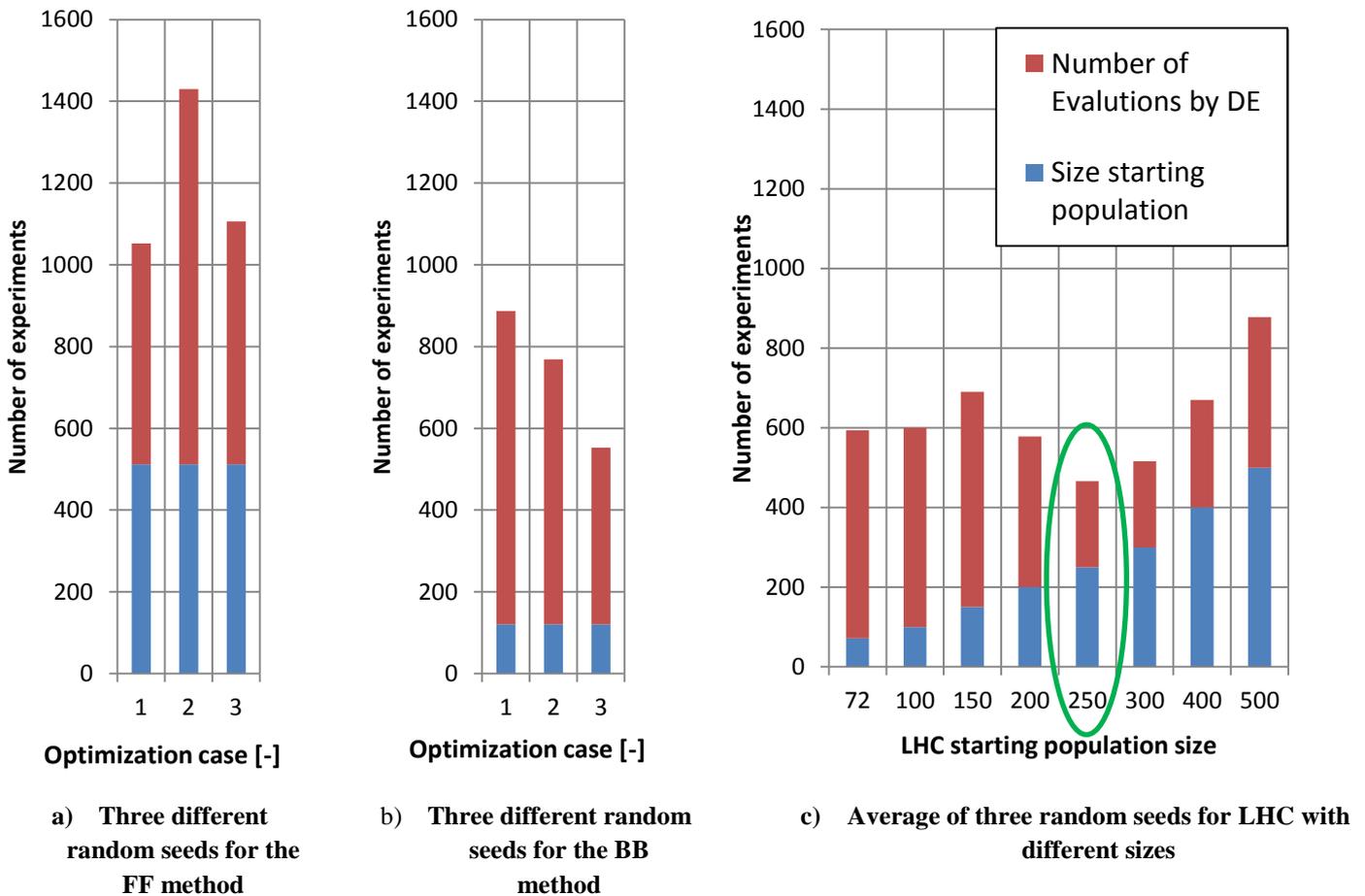


Figure 38: Number of evaluations required for different DOE methods

It was found that all starting populations can find the optimum. However, a major difference can be seen in the required amount of evaluations to find the optimum. The Full Factorial performs the worst at 1175 evaluation on average, whereas the BB performs better with 700 evaluation on average, and LHC performs the best at 850 – 475 evaluations on average. Note that limited design space has an effect on the FF and the BB. Since the design space was set **around** the optimum point (± 1 boundaries), the optimum point lies in the middle for most variables. The BB, which searches in the middle of the design space, will look closer to the optimum in this case, whereas the FF looks at the corner points, hence looking in the wrong space. Keeping this in mind, selecting FF or BB can affect the strategy negatively when used in the full design space. Hence, the LHC was selected, where a LHC of 250 evaluations, encircled in Figure 38 c) seems to perform best.

Weighting Factor and Inverse crossover probability

Lastly, the Weight Factor (WF) and inverse Crossover (CR) were studied. Here, the WF is used to generate a new, potential design point by adding the weighted difference between two design points to a third, see Equation 5. The CR will determine if a design variable is selected from the previously created potential design vector or from a design vector in the existing population, see Equation 7.

The two settings were studied keeping in mind their dependency on the optimization problem at hand. A WF of 0.8 was found to be most optimal. Higher values result in

premature convergence, which lowers the probability of finding the global optimum. It should be noted that this strategy only considers a small part of the design space, meaning that with a WF of 0.8 on the full design space the convergence rate can be too high. Once again, this lowers the possibility of finding the global optimum. Therefore, it was accepted to keep the default value of 0.7, which slows down the optimization process but provides a higher probability of finding the global optimum. Similar things can be observed for the crossover probability. Here, lowering the value ensures that good designs survive longer, on a small design space this is less critical, as the design points are relatively close to the optimum. On the full design space this is not the case and can result in the DE getting stuck in a local optimum, hence it was also kept at the default value of 0.85.

Summary of the settings

The main goal of this study was to assess the settings of the algorithm which can help to obtain a better solution faster. Once again, it should be noted that for DE or GA strategies the settings can be quite arbitrary. Even throughout the different optimization strategies these can vary. However, recommendations for some of the settings were found. It is advised to revise the settings in a later stage, though that can be a thesis project in itself. As performing the DEs on the full design space can take up to three to five days per DE, it could take months before meaningful conclusion can be drawn. As the WF and CR require a more detailed analysis, they are kept at their default values. The LHC of 250 experiments was used as a starting population, where the population size was based on the number of design variables multiplied by six. Lastly, the average step length stopping criterion was computed during each of the strategies as it depends on the selected parameterization.

Although it is not possible to prove that the found optimum is the global optimum, this section gives more certainties that this is the case. Additionally, this section enriched the author on the effects of the different settings and how they can be manipulated to increase the efficiency of optimization. Lastly, the manual search over the whole design space did not result in a more optimal result, though an opportunity for a more optimal parameterization was observed. This led to the following strategy, the tailored skin parameterization.

6.4 Strategy D: Tailored skin parameterization

During the post-processing of the previous strategy, it was observed that the skin-rib interfaces, discussed in Section 3.3.1, around the actuator were most critical. Due to the current parameterization, large skin sections around these critical areas were assigned the same material, causing it to be over designed. Hence, additional variables were added to the optimization to differentiate these sections resulting in a tailored skin parameterization strategy.

6.4.1 Implementation

- P** The previous skin parameterization did not provide enough freedom and caused excessive material to be added to some skin section. Therefore, two variables were introduced to prevent this, as visualized in Figure 39. The variables are as follows:
- Number of plies for the Heavily Loaded (HL) skin-rib interfaces, yellow stripes in Figure 39 on the right;
 - Number of plies for the Normally Loaded (NL) skin-rib interfaces, red stripes in Figure 39 on the right.
- I** Some minor changes were required inside the tool to use the selected parameterization for the HL and NL skin-interfaces. This was done by checking if the interface sections are located behind or adjacent to the actuators and assign the materials accordingly.
- P** For the post-processing, two additional MS_{crit} are retrieved. These correspond to the critical MS of the NL and HL skin-interfaces.
- A** The same algorithm as in Section 6.2, the DE, was selected. The DE settings were chosen as described in Section 6.3. Hence, for the starting population, a LHC of 250 experiments; the average step size stopping length criteria, set at 0.14; and the population size was set to 84.

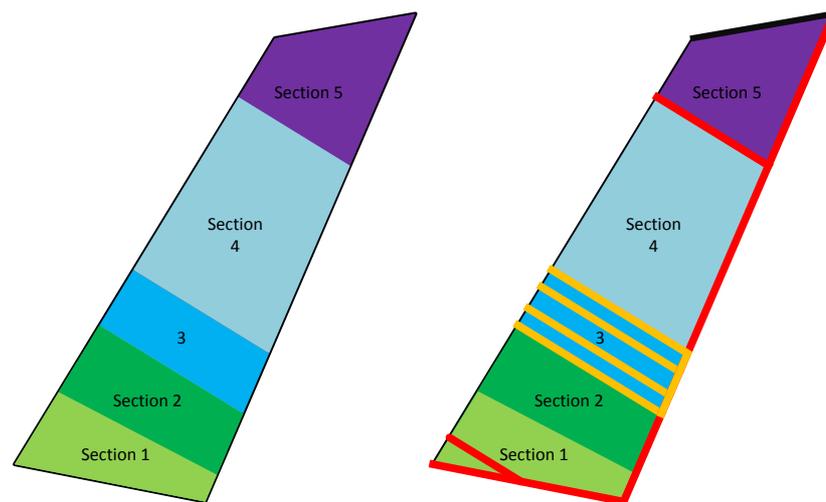


Figure 39: Skin parameterization strategy B (left) versus new parameterization (right)

Objective: Minimize	$\frac{W_{rudder}}{W_{rudder_x}}$	
W.r.t:	$N_{ribs_bay_2}$	$\in \{x \in \mathbb{Z} \mid 0 \leq x \leq 4\}$
	$N_{ribs_bay_6}$	$\in \{x \in \mathbb{Z} \mid 0 \leq x \leq 7\}$
	$N_{ribs_bay_7}$	$\in \{x \in \mathbb{Z} \mid 0 \leq x \leq 2\}$
	N_{HL_ribs}, N_{NL_ribs}	$\in [4,6]$
	$N_{skin_section_k}$	$\in \{x \in \mathbb{Z} \mid 5 \leq x \leq 16\}$ for k in [1,2,3,4,5]
	$N_{front_spars}, N_{rear_spars}$	$\in \{x \in \mathbb{Z} \mid 4 \leq x \leq 12\}$
	$N_{HL_interface}, N_{NL_interface}$	$\in \{x \in \mathbb{Z} \mid 7 \leq x \leq 16\}$
S.t.:	$MS_{min} \geq 0$	

Here, $N_{HL_interface}$, $N_{NL_interface}$ are the number of plies of the HL and NL skin-rib interfaces. This results in a total of fourteen variables. Note that the minimum number of plies of the skin-rib interfaces is higher compared to the skin pockets, which is due to assembly requirements. Using the more tailored skin parameterization gives the skin more variation and could result in a better optimum.

6.4.2 Results

A better optimum was found, however it was found to be infeasible. This was due to KRIOS's limitations. The found optimum for the skin can be seen in Figure 40, where the skin thickness, which is directly linked to the number of plies, is aggregated. One can see that the heavily loaded skin interfaces are way thicker than the skin pockets (14 plies versus 5 plies). Although it is possible to manufacture such a skin, a more evenly distributed thickness distribution is expected according to Fokker engineers. The large deviation between the sections can be explained due limitations of the mesh in KRIOS, as was mentioned in Section 4.2.5. The hinge brackets in the FE model are modeled using a few rigid bar elements. These introduce high loads at small sections of the front spar, while the real bracket has a larger contact area with the front spar and distributes the load more evenly. Additionally, the corner cut outs are missing in the rib mesh, meaning that in the FE model the spar, rib, and skin are connected in the corners, causing higher peak stresses in the skin. Both these limitations occur at the location of the highly loaded skin interfaces. Thus, the parameterization abuses these limitations and therefore the solution was not accepted.

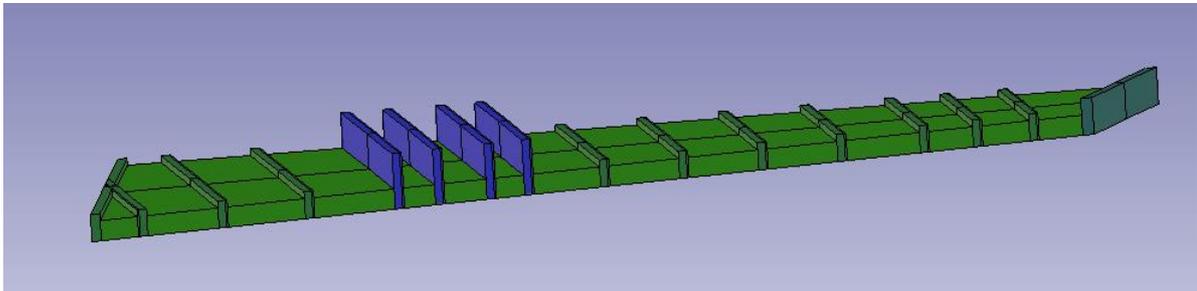


Figure 40: Skin thickness distribution for found optimum with tailored skin parameterization

Due to the limitations, it was chosen to remove the parameterization for now. The large deviations between the number of plies of these sections can be avoided in multiple ways. Since the number of plies between the two skin panels was too extreme there is the possibility to add a constraint, e.g. only allowing a difference of one or two plies. However, this means that the tool can still abuse the same limitations in the FE model. In addition, it is expected that when the limitations in KRIOS are fixed, this will automatically result in a more evenly distributed skin. Hence, this parameterization is removed for now and a different approach for the skin distribution will be selected, though it is recommended to revise this strategy when the limitations of KRIOS are fixed.

6.5 Strategy E: Quadratic skin distribution parameterization

The previous strategies from Section 6.2 and 6.4 have their limitations or abuse the missing requirements of the tool. Hence, a different approach was required. The limitations of the strategy described in Section 6.2, is that there are predefined sections which might prevent the tool from finding the optimal solution. By giving the tool more freedom, e.g. increasing the number of predefined sections, the design space will increase problematically. Additionally the high difference between number of plies of neighboring sections, as occurred in Section 6.4, can occur here as well. Taking this into account, a new approach has to be used that gives the tool more freedom, while also ensuring that an optimum can be found within a limited amount of time.

The new approach was found using the experience from the previous strategies, experience from engineers, and the expected moment distribution of a rudder, thereby circumventing the limitations of Section 6.2 and 6.4. When looking at the moment distribution over the rudder, a quadratic distribution is expected. This is shown in Figure 41, where the black line represents the front view of the rudder, while the blue and red dots are the rotating and actuator hinges, respectively. The green line represents the expected moment distribution. It is expected that this results into a skin thickness distribution as shown in Figure 42. Hence, the new parameterization for the skin uses this knowledge and creates these distribution lines, which should automatically result in a more gradual skin distribution. Note that this line can be divided into three separate lines, two quadratic lines: these are on the inboard and outboard and a constant line is present in the middle of the rudder.

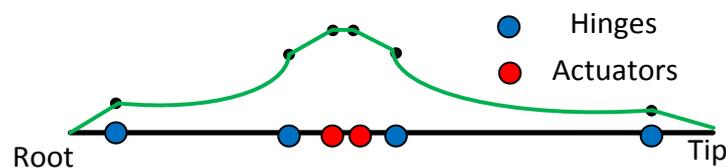


Figure 41: Expected moment distribution on the rudder

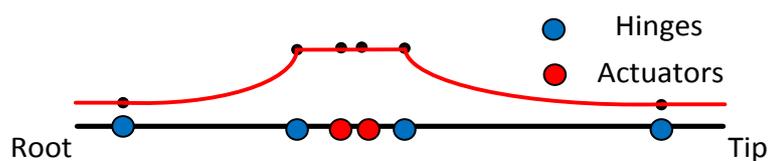


Figure 42: Expected thickness distribution for the skin according to engineers

Using these expected thickness distribution line seems promising, though the question remains on how the shape of the line can be controlled effectively and how it result in a feasible skin thickness distribution. Here, the trade-off should be made between having an as small as possible set of points with less control over the shape of the line, or having many points leading to a large increase in design space thus requiring more time. Additionally, a composite material can only contain discrete steps and drop-offs in skin pockets are not allowed. Thus, the expected quadratic distribution cannot be used directly, an intermediate step was required, which will be explained in the integration of the PIPA. An example of how the parabolic line is used to obtain a discretely stepped skin thickness distribution is

illustrated in Figure 44, the selected parameterization for the distribution lines is explained below.

6.5.1 Implementation

P The number of variables for the parameterization remains the same as strategy discussed in Section 6.2, however, the skin variables are changed to implement the aforementioned approach. Due to the expected shape of the line, it was decided to use five variables to allow for a more flexible skin distribution, these can be found in Table 14. Note that the values for control points are discrete with a very small and normalized. Hence, they give the skin distribution more freedom, using the same amount of variables in B. They were normalized to ensure the following constraint:

- $N_{\text{skin_root}} \leq N_{\text{skin_CP1}} \leq N_{\text{skin_mid}}$
- $N_{\text{skin_tip}} \leq N_{\text{skin_CP2}} \leq N_{\text{skin_mid}}$

Note the difference between the two; CP_{skin_2} , the normalized number of plies at the second control point; $N_{\text{skin_CP}_2}$, the number of plies at the second control point. Hence, they are related as follows:

$$N_{\text{skin_CP2}} = N_{\text{skin_tip}} + CP_{\text{skin}_2} * (N_{\text{skin_mid}} - N_{\text{skin_tip}})$$

I The new approach required some integration. Luckily, all of these could be implemented on the top level. Since the tool expects a list of material for all the sections, only the functionality that changes the polynomial line into a discretely stepped list was added. This is done by deriving the polynomial function $ax^2 + bx + c$, where x is the skin section index, a , b and c are constants which can be derived based on the material selected at the root, mid and tip. The outcome for each of the section is rounded up, since only discrete values for the number of plies can be selected.

Additionally, two constraints were added in the tool to support the new approach, which assume that the thickness as the root and tip is lower or equal to the thickness in the middle.

- $N_{\text{skin_root}} \leq N_{\text{skin_mid}}$
- $N_{\text{skin_tip}} \leq N_{\text{skin_mid}}$

P The post-processing was kept the same as in section 6.2.

A The same algorithm as in Section 6.4, the DE, was selected. Two settings of the DE were changed; the average step size stopping length criteria was changed and set to 0.05; and the population size was set to back to 72.

Table 14: Selected variables for the skin distribution line

Variable	Description	Index in Figure 43
N_{skin_root}	Defines the number of plies at the root and hinge closest to root	1
CP_{skin_1}	Normalized value for the number of plies at the control point 1 (0 results in N_{root} , 1 results in N_{mid})	2
N_{skin_mid}	Defines the number of plies between the hinges closest to the actuator hinges	3
CP_{skin_2}	Normalized value for the number of plies at the control point 2 (0 results in N_{tip} , 1 results in N_{mid})	4
N_{skin_tip}	Defines the number of plies at the tip and hinge closest to tip	5

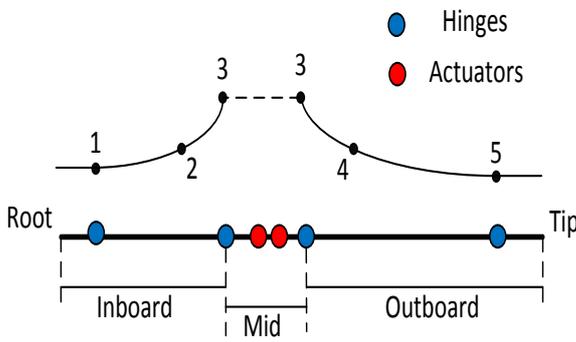


Figure 43: Control points for quadratic skin thickness distribution lines

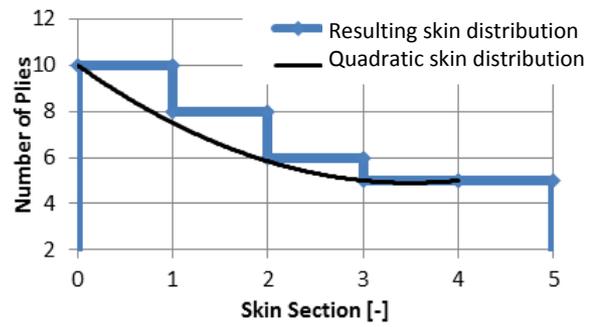


Figure 44: Discretization of the quadratic skin distribution

Objective: Minimize $\frac{W_{rudder}}{W_{rudder_x}}$

W.r.t:

- $N_{ribs_bay_2} \in \{x \in \mathbb{Z} \mid 0 \leq x \leq 4\}$
- $N_{ribs_bay_6} \in \{x \in \mathbb{Z} \mid 0 \leq x \leq 7\}$
- $N_{ribs_bay_7} \in \{x \in \mathbb{Z} \mid 0 \leq x \leq 2\}$
- $N_{HL_ribs}, N_{NL_ribs} \in [4,6]$
- $N_{skin_root}, N_{skin_mid}, N_{skin_tip} \in \{x \in \mathbb{Z} \mid 5 \leq x \leq 16\}$
- $N_{front_spars}, N_{rear_spars} \in \{x \in \mathbb{Z} \mid 4 \leq x \leq 12\}$
- $CP_{skin_1}, CP_{skin_2} \in \{\frac{x}{10^4} \mid x \in \mathbb{Z}, 0 \leq x \leq 10^4\}$

S.t:

- $MS_{min} \geq 0$
- $N_{skin_root} \leq N_{skin_mid}$
- $N_{skin_tip} \leq N_{skin_mid}$

Here, CP_{skin_1} , CP_{skin_2} are the normalized number of plies at the control points 1 and 2. They can vary between zero and one with a step size of $\frac{1}{10^4}$. Also note that the addition constraints that are added, if these are not satisfied, the function analysis will not compute the weight nor

the MSs of rudder. As a result, these experiments only take a couple of seconds compared to the 3 – 4 minutes of the other experiments.

6.5.2 Results

Due to the new skin parameterization the optimization was unable to find an optimum. Although the control points are discrete, very small steps is allowed for this variable (0.0001). However, changing the control point slightly has almost no effect on the quadratic line. Figure 45 shows how this effect, where only the CP_{skin_2} is varied from (0.405 to 0.5) which can be compared in a number of plies value of 7.1 - 7.5 at this point, and Table 15 show the corresponding values for each section. Although the values slightly differ, when rounded up they result in the same thickness distribution. Therefore, the optimizer can obtain duplicate results regarding weight and critical MS for different design points. As the step size is too small, many experiments with different CP values will yield the same result. If this result contains best objective value of the current population, the DE might think that this is the optimum and can cause the DE to get stuck.

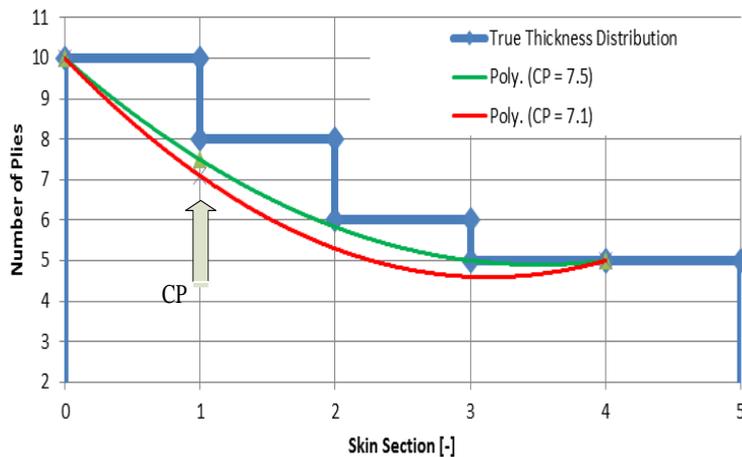


Table 15: Number of plies per section for different CP values

Section	Section value for CP = 7.5	Section value for CP = 7.1
0	10	10
1	8	8
2	6	6
3	5	5
4	5	5
5	5	5

Figure 45: Visualization of the same skin thickness distribution for different design points (CP = 7.5 and CP = 7.1)

6.6 Strategy F: Improved quadratic skin distribution parameterization

Two solutions are suggested for a new strategy, both have a negative side: one can reduce the number of duplicated distributions or add a penalty function. Reducing the number of duplicates can be done by increasing the minimum step size of the control points. This solution comes at the cost as it is not including all possibilities if a step size is chosen that is too large. Adding a penalty function can be done by checking, e.g. the difference between the thickness distribution line and the true thickness distribution and try to minimize the difference. The downside to this solution is that it requires the analysis of all the duplicates, which results in a long run time. It is expected that, compared to the reduction in freedom by increasing the minimum step size, it will have a more negative influence on the optimization run time. Hence, it was chosen to increase the minimum step size.

The step size has to be chosen carefully, therefore a study was performed to obtain the optimal step size for both control points. By increasing the step size a lower amount of duplicated experiments occur, though this comes at the cost that not all the possibilities are include. So a trade-off between the number of duplicates and the amount of possibilities was performed. Figure 46 illustrates how the trade-off was made, here the number of steps versus the number of skin thickness distributions is plotted. Note that the number of steps is directly related to the step size ($= (N_{\text{steps}})^{-1}$), also note that Figure 46 is a simple illustration for explanation purposes only.

The step size was chosen such that it includes at least 90% of the possible solution, displayed by the blue dashed in Figure 46. The unique amount of skin distribution is indicated with the red line, whereas the solid blue line shows the amount of duplicated skin distributions. One can see that when the number of steps is equal to four, all skin distributions will be unique. However, if five steps are allowed one of the skin distribution will occur twice and there will be two design points with the same result.

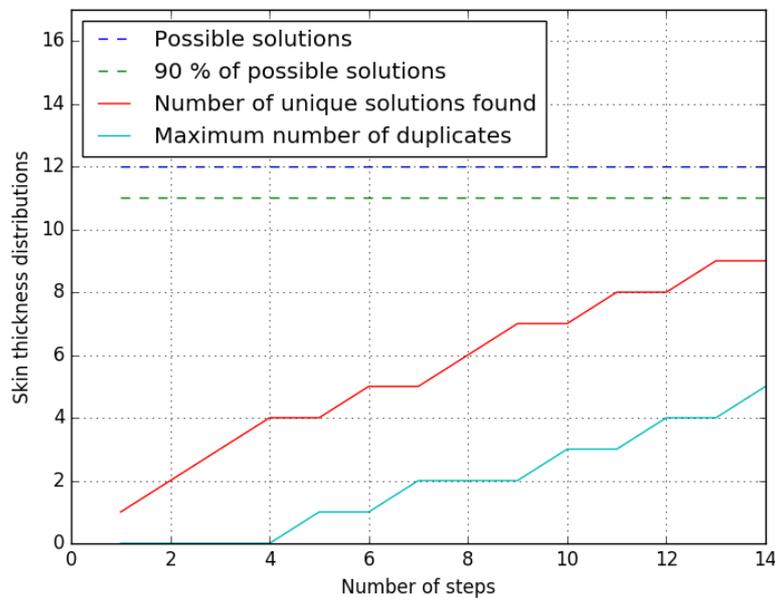


Figure 46: Effect of numbers of steps on the number of skin thickness distributions

6.6.1 Implementation

- P** The same variables are used as in strategy from Section 6.5, however, the number of allowed steps for the CP_{skin_1} and CP_{skin_2} were added and set at 55 and 70, respectively.
- I** The new approach required no further integration.
- P** The post-processing was kept the same as in Section 6.5.
- A** The same algorithm (DE) and settings from Section 6.5 were used

Objective: Minimize $\frac{W_{\text{rudder}}}{W_{\text{rudder}_x}}$

W.r.t:

$N_{\text{ribs_bay}_2}$	$\in \{x \in \mathbb{Z} \mid 0 \leq x \leq 4\}$
$N_{\text{ribs_bay}_6}$	$\in \{x \in \mathbb{Z} \mid 0 \leq x \leq 7\}$
$N_{\text{ribs_bay}_7}$	$\in \{x \in \mathbb{Z} \mid 0 \leq x \leq 2\}$
$N_{\text{HL_ribs}}, N_{\text{NL_ribs}}$	$\in [4,6]$
$N_{\text{skin_root}}, N_{\text{skin_mid}}, N_{\text{skin_tip}}$	$\in \{x \in \mathbb{Z} \mid 5 \leq x \leq 16\}$
$N_{\text{front_spars}}, N_{\text{rear_spars}}$	$\in \{x \in \mathbb{Z} \mid 4 \leq x \leq 12\}$
CP_{skin_1}	$\in \{\frac{x}{54} \mid x \in \mathbb{Z}, 0 \leq x \leq 54\}$
CP_{skin_2}	$\in \{\frac{x}{69} \mid x \in \mathbb{Z}, 0 \leq x \leq 69\}$

S.t:

$MS_{\text{min}} \geq 0$
$N_{\text{skin_root}} \leq N_{\text{skin_mid}}$
$N_{\text{skin_tip}} \leq N_{\text{skin_mid}}$

6.6.2 Results

The implementation of the step size helps the optimizer find an optimum faster. The optimizer is capable of finding the same optimum as found in Strategy B while reducing the number of experiments significantly. As is illustrated in Figure 47, the optimum was found after 14 iterations (1053 experiments). Compared to strategy B, this is a reduction of 897 experiments (46 %). It should be noted that of the 1053 experiments of this strategy, 17% did not satisfy the added constraint of number of plies at the root and tip being equal or lower than the number of plies in the middle. If this constraint is violated, no structural nor weight analysis takes place and these cases only take a few seconds to analyze, compared to the 3 - 4 minutes of the other experiments. Hence, only 874 (reduction of 55%) function evaluations are actually required compared to Strategy B. Though, since the same constraint can be added to Strategy B and the effect of adding this constraint is unknown, the conservative reduction of 46% is taken.

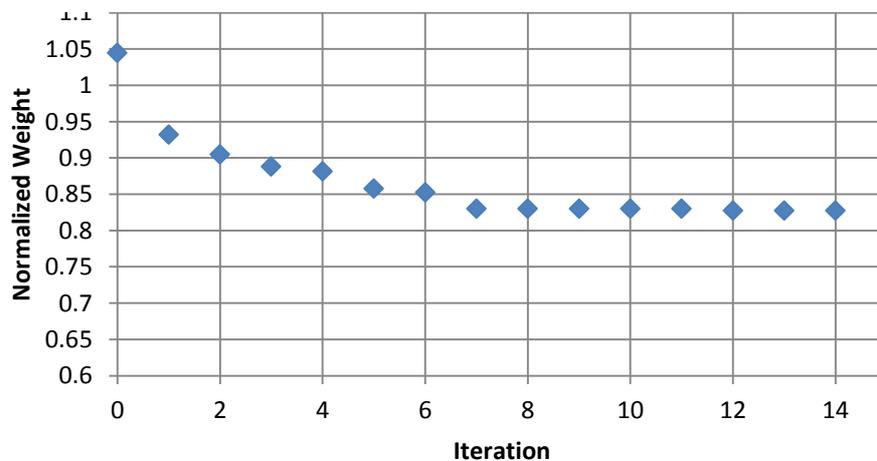


Figure 47: Normalized weight per iteration for the quadratic skin thickness distribution.

6.7 Comparison of the different strategies

In this section multiple strategies were assessed and it was shown how a rudder can be optimized. The strategies were created based on the PIPA aspects, which give a clear overview of the optimization set up and will support the engineers for using the strategies in future projects. During these strategies, the design requirements of the Rudder X were used. Although KRIOS managed to find a weight reduction of 17%, it should be treated with caution as reduction can be partly caused by a limitation in KRIOS. This limitation can be found in the failure modes, as currently a simplified buckling analysis takes place and some failure modes, e.g. joint strength analysis, are missing. As a result, the found optimum might not be satisfactory if these failure modes are fixed/added. Overall, this only affects the final objective value, which will be less optimal, but the same optimization strategies can be implemented.

Although the lightest design was found in strategy B, subsequent strategies show major reduction in the required function evaluations. Figure 48 shows the normalized weights for the aforementioned strategies, as well as the total number of function evaluations that were required. Here, it can be seen that the last strategy (Strategy F) finds the same objective value as Strategy B. However, the number of function evaluations went from over 1950 to 1053, achieving a reduction of 46% which is equivalent to a reduction of almost 2.5 days.

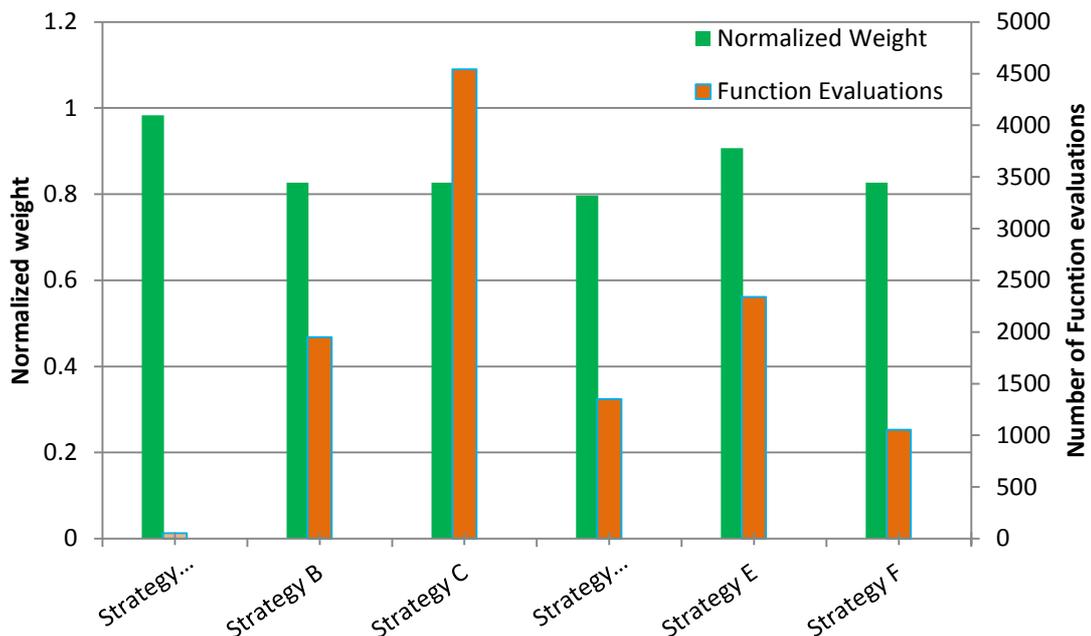


Figure 48: Normalized weight and number of function evaluations per strategy

A trade-off was performed for the different strategies to assess the associated benefits. A summary can be found in Table 16, where a higher value represents a better score. An elaborate explanation follows. Although Strategy B found the same optimum as Strategy F, it was observed that the predefined sections of strategy B can result into a suboptimal result for different design requirements. So even though this strategy was capable of finding the same optimum as Strategy F now, there is a high risk this is not the case for other rudder designs.

The strategies D, E, and F come with their own risks. First of all, Strategy D resulted in an infeasible solution, because its abuse of the limitations of the tool. Next, Strategy E was unable to find the optimum, resulting in a bad score on the found optimum and the risk. Lastly, strategy F was capable of finding an optimum, though it still contains duplicated skin distribution which introduces a small risk of the optimizer getting stuck.

The accessibility of each strategy was obtained taking into account three subcriteria: The difficulty of the parameter integration on the top level, the comprehensibility of the strategy, and the changes required inside the tool. As the strategies A, B, and C contained a parameterization that was already implemented, no large changes were required at either top level or inside the tool. Additionally, the defined variables of these strategies are more self-explanatory, e.g. Strategy A contains one variable for the whole skin material, while Strategy B and C contains a slightly less obvious material selection for different sections, compared to Strategy F which contains a normalized value between zero and one and uses this to create the quadratic skin distribution. Here, the skin material selection for Strategy F and its effect on the optimization strategy is more difficult to explain to an engineer. Lastly, the strategies D, E and F, required a large integration at top level and the addition of a small functionality inside the tool.

The last criteria, the fidelity of the model, was obtained by checking to what degree the real object can be modeled. In Strategy A, a rough parameterization was used, e.g. one variable for the skin, whereas the actual skin contains more variety. Hence, this parameterization cannot results in a lower fidelity. Strategy B, allows more variety in the skin, ribs and spars. During the last three strategies this freedom on the skin distribution increases even further, increasing the fidelity. However, Strategy D abuses the limitations and can therefore not represent the actual object correctly, hence the lower fidelity.

Table 16: Trade-off table of the optimization strategies

	Objective	Run Time	Risk	Accessibility	Fidelity	Total Score
Weight Factor	15	5	2	2	2	
Optimization Strategy						
A: Response of the model	0.017	0.99	0.5	0.7	0.25	8.72
B: Higher fidelity optimization	0.173	0.56	0.75	0.5	0.50	9.78
C: Validation of the optimum and algorithm settings	0.173	0.00	0.75	0.4	0.50	6.68
D: Tailored skin parameterization	0.203	0.70	0	0.3	0.00	6.43
E: Quadratic skin distribution	0.093	0.33	0.25	0.3	0.75	6.48
E: Improved quadratic skin distribution	0.173	0.82	0.75	0.2	0.75	10.93

Strategy E, the improved quadratic skin distribution with an increased step size, is performing the best. However, it should be noted that both strategy A and B can be used as well to optimize a rudder. Even though Strategy A will result in a less optimal result, it can help to quickly analyze the design space and give a rough estimate of the weight. One could choose to use Strategy B, though Strategy F obtains the same optimum with fewer evaluations. Strategy D can be promising, though it requires a more thorough examination once a higher fidelity model is available. Lastly, Strategy E should be avoided as it cannot find an optimum.

7

Conclusion and recommendations

The question addressed in the thesis was whether the detailed design and analysis process of a composite rudder torsion box can be automated such that it supports the implementation of design optimization and to investigate the associated benefits. It was found that this required a new approach that combines both a large scope and high level of fidelity, as was illustrated in Figure 7. The thesis was divided into two parts to achieve this combination. First, Part I of this research was the development of KRIOS, which automates the detailed design process such that it supports optimization. Second, Part II was to determine if, and how, KRIOS could then be used to optimize the rudder. This research provides a first step into the assessment of higher fidelity product models while increasing the scope of optimization. Future improvements, which can be seen as Part III, can be made at different levels to access the full power of KRIOS.

7.1 KRIOS, the automated design method

The first part of this research, the development of KRIOS, showed how development of a KBE tool in ParaPy made it possible to automate the detailed design process of a composite rudder, taking into account industrial requirements for the rudder design. KRIOS increases both scope and fidelity by extending the RG and combining this with HDOT, as illustrated in Figure 49, while being fully automated. As a result, the time to design one rudder configuration with a comparable scope, but lower fidelity level, is reduced from between seven and eight months to three and four minutes. The lower fidelity arises from the limitations in the FE model and the missing failure modes. However, the Finite Element Analysis that takes place represents the behavior to a high enough degree to perform optimization, as was validated in Section 4.3. Here, it was also shown that KRIOS can estimate the weight to within a 5% error margin.

KBE is a relatively new concept, meaning that engineers in the current industry are often sceptical about the usages of KBE tools. With KRIOS, engineers have the possibility to access different and familiar output types, as discussed in Section 4.2.7. Hence, these functionalities allow the engineer to use the tool, but also inspect the output in different stages. This helps to solve the common problem of KBE tools, which is the lack of trust engineers have in these tools when it looks like a black box.

The fidelity level of KRIOS should be increased, to achieve the aim of KRIOS as depicted in Figure 49. To this end, the following areas are especially of interest:

- Improving the finite element model by including a structured mesh, rib corner cutouts, and implement a detailed bracket FE model;

- Adding missing failure modes, e.g. rib crushing and joint strength analysis.

Additionally, the scope of KRIOS should be increased by implementing the following:

- Adding multi-spar configurations;
- Including the LE-box and TE-box design.

The rapid synthesis and analysis of the rudder opened up the possibility to perform optimization. Due to the long design lead time of the manual design process, a trade-off between three to five different designs was possible. However, KRIOS allows for a rapid rudder design. It is capable of evaluating up to 2000 designs within five days. However, due to the immense design space of a rudder, this by itself is not sufficient to find an optimal rudder. As such, different optimization strategies were implemented and compared.

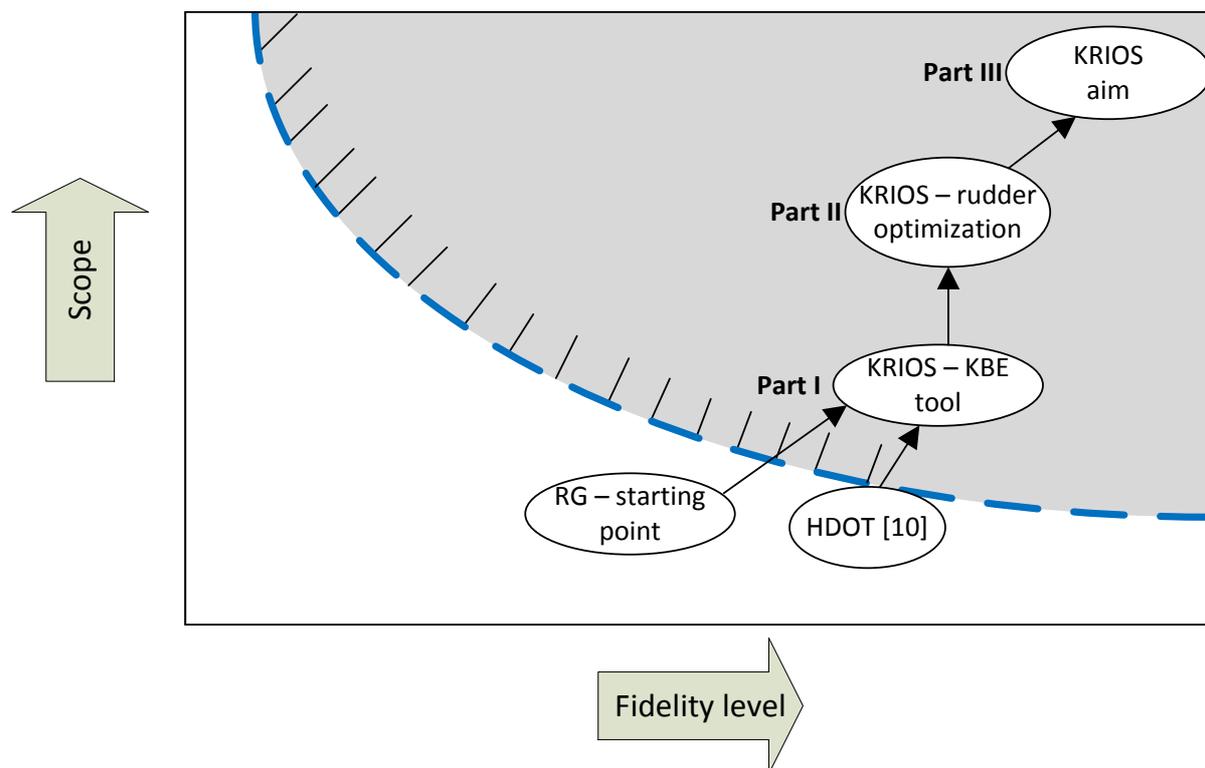


Figure 49: Final scope versus fidelity level of KRIOS and its aim, adjust from Figure 7.

7.2 Assessment of rudder optimization strategies using KRIOS

In the second part of this research different optimization strategies were assessed to check if and how KRIOS could be used to optimize a composite rudder. Thanks to the implementation of optimization strategies a large increase in scope is achieved, illustrated in Figure 49. The strategies were divided into four main building blocks; these are the Parameterization, Integration, Post-process, and Algorithm, also called the PIPA. This simplifies the implementation and adjustment of the strategies in future projects. Currently, six strategies were assessed, allowing engineers to select one based on the performance of the selected criteria, discussed in Section 5.5.

In this research, the parameterization was the one of the main focuses. To study the effects of the parameterizations, the integration, post-process and algorithm had to be adjusted

accordingly. Throughout the research it was shown that the possible design variants are immense if all the parameters of a rudder can be changed individually, which makes it difficult to optimize the ruder. Generally speaking, the trade-off is made between the amount of variables and level of optimality. In this thesis, different solutions for the parameterization were proposed, which reduces the amount of variables while allowing a large amount of the design possibilities, and thus ensuring that a large scope and high fidelity can be achieved.

For the ribs, the selected parameterization was applied by determining the number of ribs in different bays, whereas the materials of the ribs were selected based on the expected loads acting on the rib. As for the front and rear spar, the locations are determined by placing them as far forward and aft as possible, until critical assembly constraints are reached, hence they are not variables. Currently, one material was selected for each spar. The biggest challenge lies in the correct parameterization for the skin material, due to the large variety possible, as was discussed in Sections 3.3.1 and 5.1. For one of the first optimizations, the skin was divided into five sections and a material could be assigned to each of these sections. This strategy was capable of finding an optimum after analyzing 1950 different designs, in around five days. Later on, a quadratic skin distribution was introduced, as the moment distribution on the rudder is expected to follow the same shape, while ensuring compliance with the requirements regarding composite materials and allowed locations for drop-offs. With this parameterization, the same optimum was found, though only 875 evaluations were required, in only 2.3 days, which is a reduction of about 55%.

It is recommended to reassess, extend, or implement new optimization strategies to increase the fidelity level and scope even further and reduce the optimization time. Regarding the **fidelity**, the following key are points identified:

- Reassessment of the tailored skin parameterization when the FE limitations are fixed.
- Improved material allocation for skin, e.g. allow a chordwise drop-off in the skin pockets, by using two quadratic skin distribution or adding a chordwise drop-off value.
- Improved material allocation for the spars, by adding a similar material distribution line for the front spar.

The **scope** can be increased taking into account the following:

- Multi-objective optimization, with the recently developed cost module both cost and weight can be optimized.
- Multidisciplinary optimization, performing both hinge and rudder optimization. As the tools are already combined, this can currently be done, though HDOT only accounts for one load case and takes 90 minutes for one hinge location. Hence, the following is advised here:
 - Find the maximum forces and moments working on the hinge out of all the load cases and use this for the hinge sizing.
 - Knowledge reuse, by creating a database (or a response surface) of the hinge size, cost, and weight with their associated force and moment.
 - Remove the exhaustive search from HDOT and use a different optimization strategy.

Lastly, reducing the **time** required to find an optimum can be achieved by:

- Reducing the design space. Current strategies allow unlikely designs, e.g. zero ribs or a very thick skin of 16 plies. It is possible to reduce the boundaries, though this might remove one of the main advantages of rapid problem solving, which is the possibility to analyse uncommon designs.
- Parallel processing allows multiple experiments to run in parallel, which can be easily implemented when using a DE algorithm, as the different experiments in a population can be analysed individually

Concluding, this thesis provides the first steps on how KRIOS enables composite rudder structure design on an industrial level and allows companies like Fokker to quickly respond to changes in customer demand. Hopefully, further research in the different optimization strategies is performed to investigate the unexplored areas with high potential benefits, such that the full power of KRIOS can be exploited, which can help Fokker to design even better rudders even faster.

Bibliography

- [1] T. van den Berg, "Harnessing the potential of Knowledge Based Engineering in manufacturing design," PhD Thesis,, TU Delft, 2013.
- [2] W. J. C. Verhagen, P. Bermell-Garcia and R. E. van Dijk, "A critical review of knowledge-based engineering an identification of," *Advanced Engineering Informatics*, vol. 26, pp. 5-15, 2012.
- [3] T. Fujimoto and S. Thomke, "The effect of front-loading problem-solving on product development performance.," *AIAA, Journal of Product Innovation Management*, vol. 17, no. 2, pp. 128-142, 2000.
- [4] M. Stokes, "Managing Engineering Knowledge: MOKA methodology for Knowledge Based Engineering Applications," John Wiley and Sons, London, 2001.
- [5] N. R. Miltion, "Knowledge acquisition in practice: a step by step guide," Springer, London, 2007.
- [6] G. La Rocca, "Knowledge based engineering: Between ai and cad. review of a language based," *AIAA, Advanced Engineering Informatics*, vol. 26, no. 2, pp. 159-176, 2012.
- [7] M. van Rij, T. van den Berg and A. van der Laan, "Aerospace solution optimization through," *ACE 15*, 2015.
- [8] J. Page Risueño, "Development and implementation of a cost assessment methodology for multidisciplinary design optimization of aircraft components," Msc Thesis, TU Delft, 2017.
- [9] A. Raju Kulkarni, "Development of a knowledge based engineering tool to support fin-rudder interface design and optimization," MSc. Thesis, TU Delft, 2015.
- [10] F. Aerostructures, Interviewee, *knowledge acquisition*. [Interview]. 2016-2017.
- [11] C. Kassapoglou, *Design and Analysis of Composite Structures: With Applications to Aerospace Structures*, Delft: John Wiley & Sons Ltd, 2013.
- [12] D. Gross, "Report from the fidelity implemntation study group," in *Simulation Interoperability workshop*, Florida, 1999.
- [13] J. Griesing and J. Bently, "A summary of industrial mdo applications and needs," American Institute of Aeronautics and Astronautics, Inc., Hampton, 1998.
- [14] H. Ghiasi, D. Pasini and L. Lessard, "Optimum stacking sequence design of composite materials Part 1: Constant stiffness design," *Composite Structures*, vol. 90, no. 1, pp. 1-11, 2009.
- [15] H. Ghiasi, K. Fayazbakhsh and D. Pasini, "Optimum stacking sequence design of composite materials Part II: Variable stiffness design," *Composite Structures*, vol. 93, no. 1, pp. 1-13, 2010.
- [16] T. van der Laan, "Knowledge based engineering support for aircraft components design," PhD Thesis, TU delft, 2007.
- [17] R. van Dijk and X. Zhao, "Multidisciplinary design and optimization framework for aircraft box structures," in *Proceedings of the 3rd Aircraft Structural Design Conference*, Delft, 2012.
- [18] F. Farzan Nasaba, H. Geijselaers and I. Baran, "A multi-level set gradient based algorithm for buckling optimization of blended composite structures," in *5th Aircraft Structural Design Conference*, Manchester, 2016.
- [19] M. van den Bosch, "Integrated Approach to Optimize the Skin Layup of a Horizontal

- Stabilizer,” MSc. Thesis, TU Twente, 2015.
- [20] G. La Rocca, “Knowledge based engineering techniques to suport aircraft design and optimization,” PhD Thesis, TU delft, 2011.
- [21] R. van Dijk, “ParaPy,” ParaPy B.V., 2017. [Online]. Available: <https://www.parapy.nl/>. [Accessed 2017].
- [22] M. J. L. van Tooren and G. La Rocca, “A knowledge based engineering approach to support autmatic generation of FE models in aircraft design,” in *45th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, 2007.
- [23] M. D’Auria and R. and D’Ippolito, “Process Integration and Design Optimization Ontologies,” in *OTM Confederated International Conferences*, Graz, 2013.
- [24] R. Storn and K. Price, “Differential Evolution - a simple and efficient adaptive scheme for global optimization over continuous spaces,” International Computer Science Institute, Munchen, 1995.
- [25] R. Storn and K. Price, “Minimizing the real functions of ICEC’96 contest by Differential Evolution,” in *IEEE*, Nagoya, 1996.
- [26] A. Piszcz and T. Soule, “Genetic programming:Optimal population sizes for varying complexity problems,” in *Proceedings of the Genetic and Evolutionary Computation Conference*, pp. 953-954, 2006.
- [27] V. K. Koumoussis and C. P. Katsaras, “A sawtooth genetic algorithm combining the effects of variable population size and reinitialization to enhance performance,” ” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 1, pp. 19-28, 2006.
- [28] F. G. Lobo and C. F. Lima, “A review of adaptive population sizing schemes in genetic algorithms,” in *The Genetic and Evolutionary Computation Conference*, pp. 228-234, 2005.
- [29] F. G. Lobo and D. E. Goldberg, “The parameterless genetic algorithm in practice,” *Information Sciences—Informatics and Computer Science*, vol. 167, no. 4, pp. 217-232, 2004.
- [30] A. E. Eiben, R. Hinterding and Z. Michalewicz, “Parameter control in evolutionary algorithms,” *IEEE Transactions on Evolutionary Computation*, vol. 3, no. 2, pp. 124-141, 1999.
- [31] M. Lunacek and D. Whitley, “The dispersion metric and the CMA evolution strategy,” in *The Genetic and Evolutionary Computation Conference*, pp. 447-484, 2006.
- [32] E. K. Burke, S. Gustafson and G. Kendall, “Diversity in genetic programming: An analysis of measures and correlation with fitness,” *IEEE Transactions on Evolutionary Computation*, vol. 8, no. 1, pp. 47-62, 2004.
- [33] M. Tyre and E. von Hippel, “Sticky infomration and the locus of problem solving: implication for innovation,” *Management science*, vol. 40, no. 4, pp. 429-439, 1994.
- [34] B. Noesis Solution, Optimus Theoretical Background, Leuven, 2015.
- [35] C. Chapman and M. Pinfold, “Design engineering- a need to rethink the solution using knowledge based engineering,” *Knowledge-based systems*, vol. 12, no. 5, pp. 257-267, 1999.

Appendix A. Overview of contribution to the RG

Table 17: Contribution of the author to the Rudder Generator

Module	Sub-task	Involvement	Phase	Comment
Input Handler	STEP reader	10%	Internship	Taken from [8], only required integration and small adjustments
	Fin-generator	100%	Internship/ Thesis	Though not used as the tool requires validations first. Hence required the STEP with the OML of the rudder X
	Parameterization integration	100%	Internship/ Thesis	Initial set up during internship, though many adjustments required during integrations of different optimization strategies
Load cases	Data collection and possible input format	100%	Internship	
	Doghouse pressure	50%	Internship	Partly taken from [3], but simplified doghouse pressure was added to reduce time and addition pressure distribution function was added
	Extra	50%	Internship/ Thesis	Load case selection, failure cases integrations, NASTRAN solver selection. Boundary conditions
Hinge System	Hinge positioning	5%	Internship	Integration of HDOT position in the RG
	Hinge sizing	-	-	-
	Detailed Bracket generation	-	Pilot	-
Rudder	Planar model	90%	Internship	
	Material allocation	75%	Thesis	Integrating new parameterization for skins, ribs and spars.
	Detailed ribs	50%	Thesis	Mainly fixing dependencies between structural component
	Detailed spars	50%	Thesis	Mainly fixing dependencies between structural component
	Detailed skin	50%	Internship/ Thesis	Different skin material allocations
	Property zone	50%	Thesis	Connecting to FEA results and

				MS visualizations, additional weight estimations
	Joints	25%	Thesis	
Structural analysis	Finite element model	5%	Thesis	Adding flanges to the FEM model
	BDF writing	5%	Thesis	Fixing minor bugs in BDF writing, Adding required axis systems for hinges. Adding SOL 105 writer
	PUNCH reader	40%	Thesis	
	FEA result dictionary	50%	Thesis	Rearranging the dependencies between the punch file and the ParaPy instance.
	Material Strength Analysis	15%	Thesis	Connected the FEA results to the Material strength analysis module
	Buckling Analysis	100%	Thesis	Added simplified buckling analysis
Weight module	Weight estimation	25%	Pilot	Base was set up by another Fokker intern, but additional weight estimations methods were added for this research.
Cost module	Pre-processing	25%	Thesis	Only face grouping was done by the author additional pre-processing was done by Page [8]
	Cost estimation	0%	-	Not used but implemented, done by Page [8]
Output collector	Different post-processing	100%	Thesis	Different ways of collecting weight and MS depending on selected post-processing
	MS visualization view	100%	Thesis	Allow to see the MS of each individual component or even mesh element
	Weight breakdown tables	0%	Pilot	Printing out familiar weight status tables, done by another Fokker intern

Appendix B. Detailed explanation of limitations of the FE model and their effects

The three main limitations of the FE model and their effect on the analysis can be found in the table below. Each of the limitations is visualized in Figure 50 to Figure 52.

Limitation	Description	Effect on Analysis
Unstructured mesh	See Figure 50, the mesh is contains triangular elements	These elements should be avoided as strains in these elements are often over predicted
Corner cut-outs of spar and ribs	See Figure 51, in the actual rudder, there are corner cutouts which avoid peak stresses at the corner points	High peak stresses will occur in the corner, since the stresses from the skin, spar, and ribs are combined here.
Simple bracket mesh model	See Figure 52, the FE model of the bracket is represented by a few rigid bars, whereas the true bracket is far more complex	Causing higher load paths around the connections of the rigid beams with front spar. Introduces high forces in a small area instead of medium forces over a bigger area.

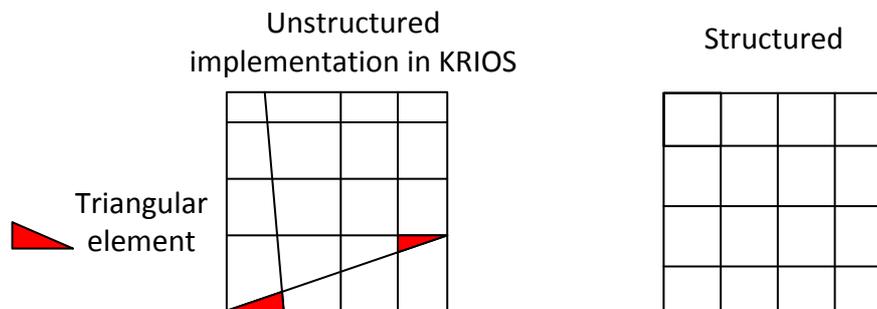


Figure 50: Unstructured (left) versus structured mesh (right)

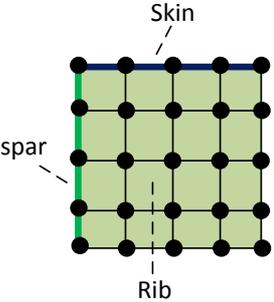
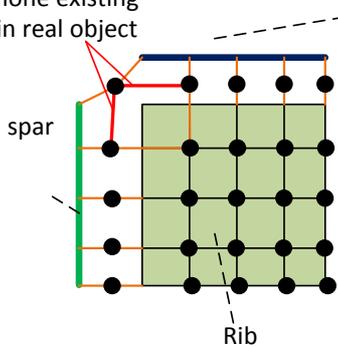
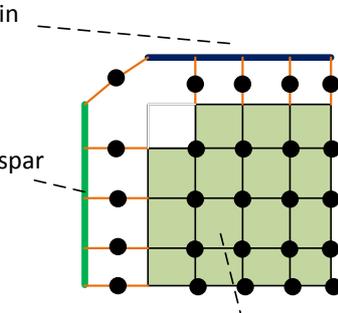
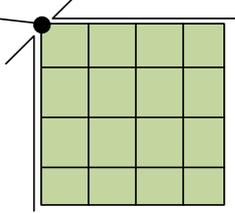
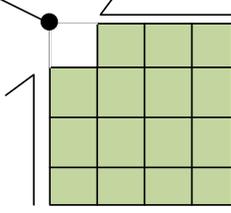
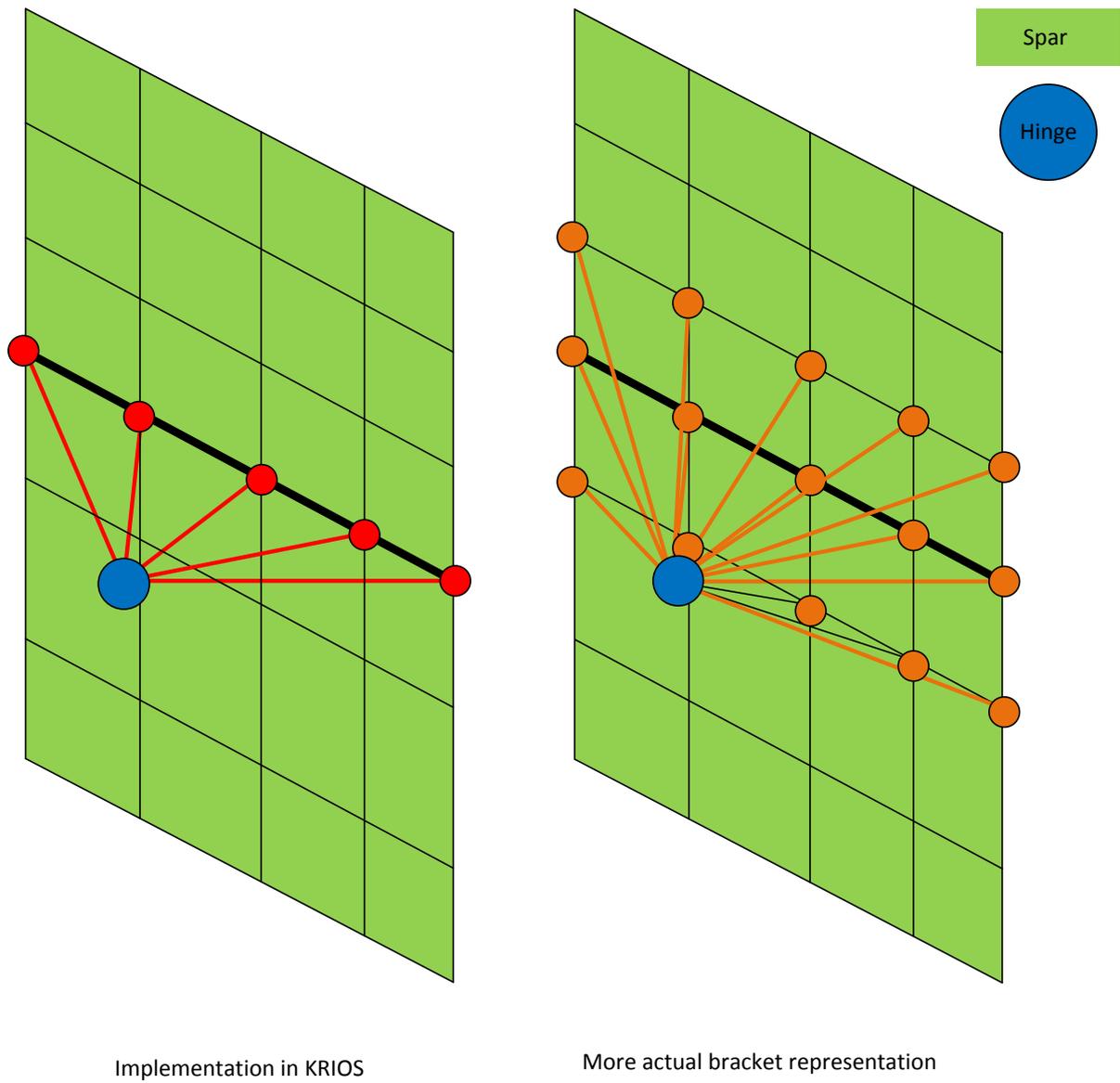
Normal mesh view	Exploded mesh view	
	Implementation in KRIOS	Actual
	<p data-bbox="624 344 778 436">Connection is none existing in real object</p> 	
Resulting Stress in node	Implementation in KRIOS	Actual
	<p data-bbox="619 862 746 929">Stress peak in node</p> 	<p data-bbox="1050 840 1209 907">No stress peak in node</p> 

Figure 51: Mesh without corner cutout (left) and with corner cutout (right)



Implementation in KRIOS

More actual bracket representation

Figure 52: Effects of a simplified bracket FE model

Appendix C. Detailed explanation of the face-grouping functionality

This appendix contains a more elaborate explanation of the face grouping functionality. Only the first loop contains additional comments, the last few loops are done on visualization.

5 plies	5 plies	5 plies	4 plies
4 plies	5 plies	4 plies	4 plies
4 plies	5 plies	6 plies	6 plies
4 plies	4 plies	5 plies	5 plies

Select a random section (blue) and check its neighbours. Here it find two neighbours with the same amount of plies (green). These section are checked next

5 plies	5 plies	5 plies	4 plies
4 plies	5 plies	4 plies	4 plies
4 plies	5 plies	6 plies	6 plies
4 plies	4 plies	5 plies	5 plies

Repeat the process for the previously found neighbouring section(s). Now there are two sections which contain 5 plies (red) and none with the same amount.

5 plies	5 plies	5 plies	4 plies
4 plies	5 plies	4 plies	4 plies
4 plies	5 plies	6 plies	6 plies
4 plies	4 plies	5 plies	5 plies

Repeat the process for the previously found neighbouring sections. Now there are two sections which contain 5 plies (red) and just one with the same amount (green). This section is checked next, the red sections are ignored

5 plies	5 plies	5 plies	4 plies
4 plies	5 plies	4 plies	4 plies
4 plies	5 plies	6 plies	6 plies
4 plies	4 plies	5 plies	5 plies

As no new section were found, one of the areas is done. The algorithm starts over with a new random section which has not be analysed yet

Figure 53: First four steps and explanation of face grouping

5 plies	5 plies	5 plies	4 plies
	5 plies	4 plies	4 plies
	5 plies	6 plies	6 plies
		5 plies	5 plies

5 plies	5 plies	5 plies	4 plies
	5 plies	4 plies	4 plies
	5 plies	6 plies	6 plies
		5 plies	5 plies

5 plies	5 plies	5 plies	4 plies
	5 plies	4 plies	4 plies
	5 plies	6 plies	6 plies
		5 plies	5 plies

5 plies	5 plies	5 plies	4 plies
	5 plies	4 plies	4 plies
	5 plies	6 plies	6 plies
		5 plies	5 plies

Figure 54: Steps five to eight of face grouping

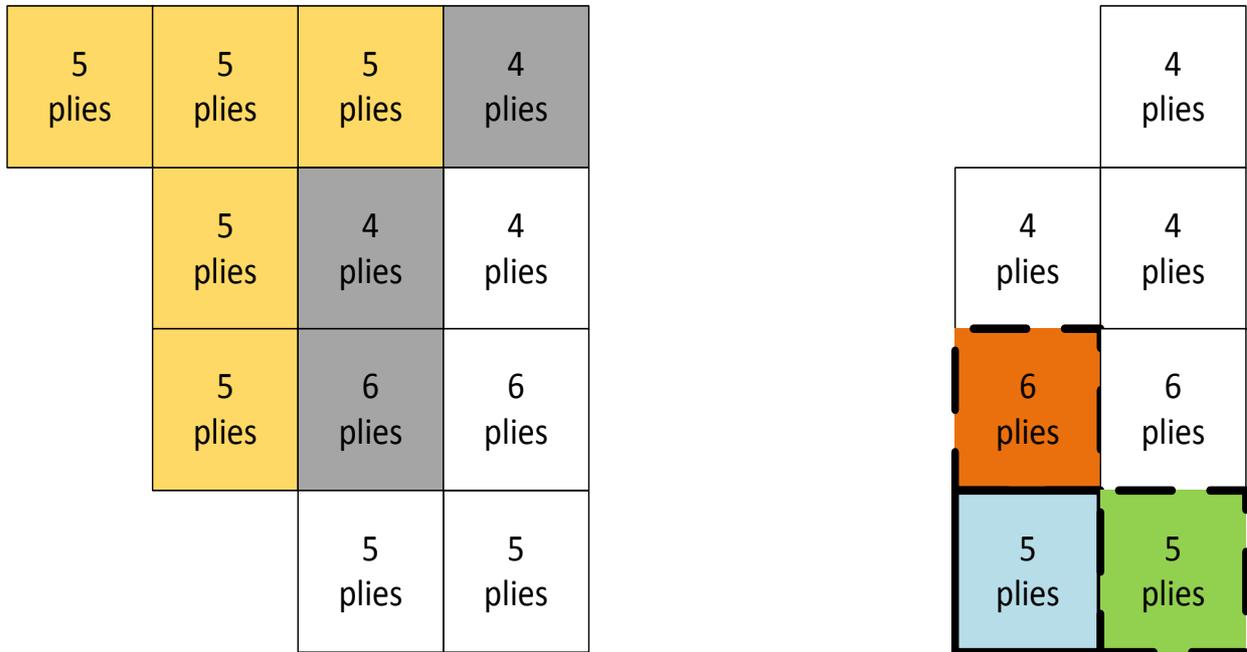


Figure 55: Steps nine and ten of face grouping

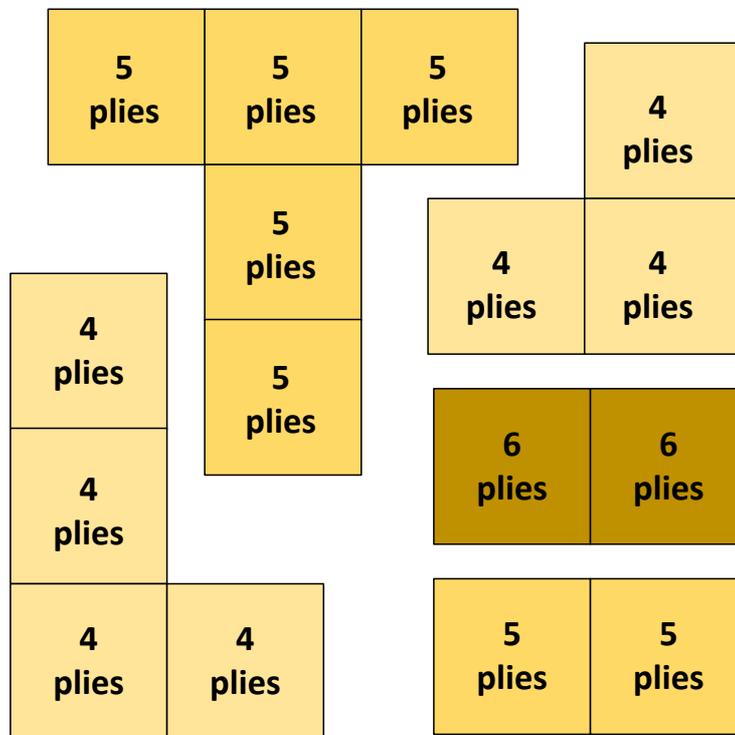


Figure 56: Resulting different sections of face grouping

Appendix D. Additional margin of safety visualization for increasing Safety factor

As part of the validation it was test how the model responds for increasing safety factor. Note that here the required lift is multiplied with Safety Factor (SF) to ensure that the rudder does not fail when flight loads are reached. The SF is increased with 10 % each time, mean that if the original lift the rudder generates is 10 newton, this because 15 N for the first cases, 16.5N of the etc. This is not the correct way to scale the load cases but can help to quickly analyze the responds of KRIOS

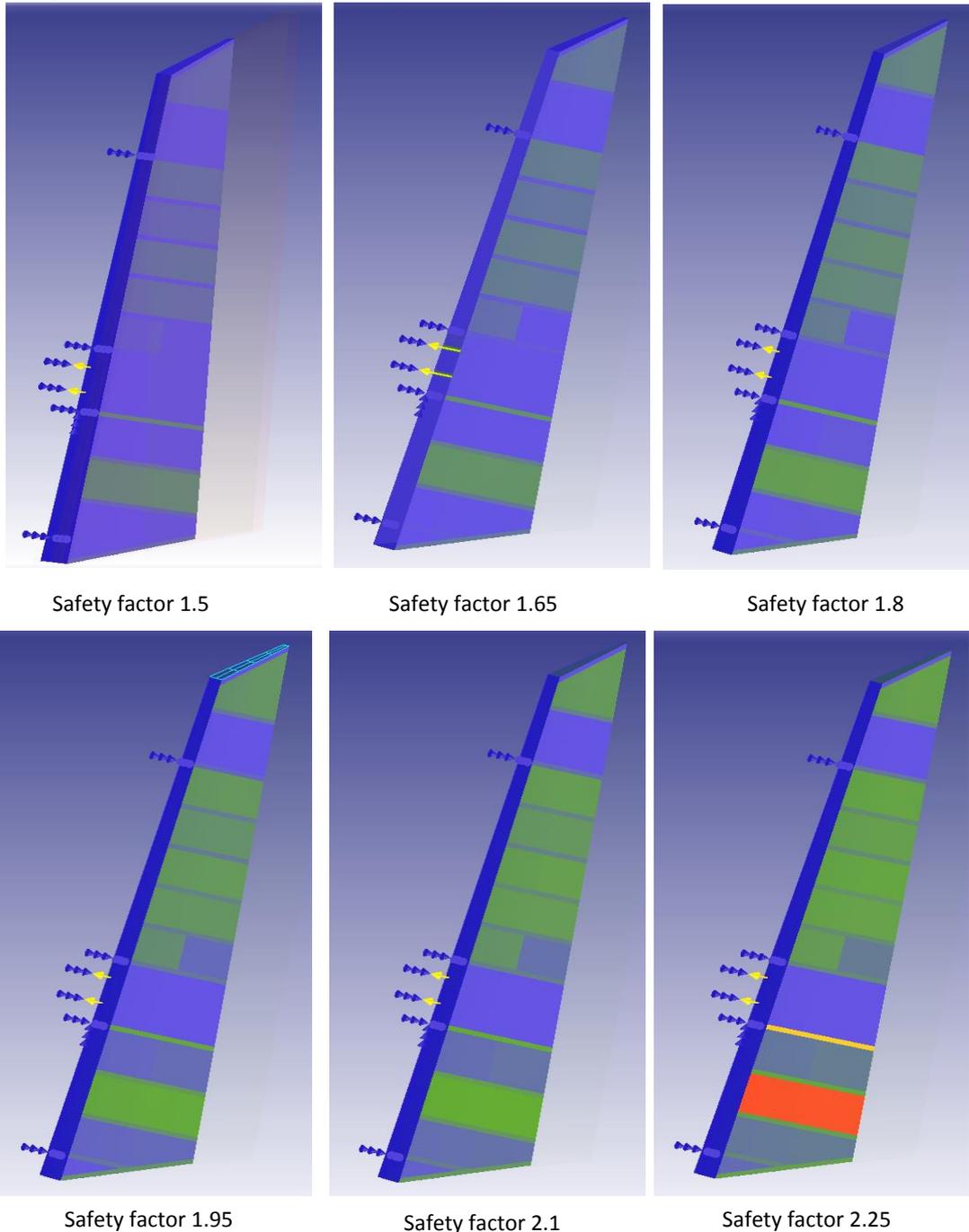


Figure 57: Margin of safety visualization for increasing aerodynamic loads

Appendix F. Detailed explanation of the average stopping length calculation

To explain how the average stopping step size was chosen, consider the following example.

Objective:	Minimize	$F(A,B,C) = A + B - C$
W.r.t.:	A	$\in [0, 1]$
	B	$\in [1, 2]$
	C	$\in [0, 1, 2]$

Consider the following population, which evidently contains the optimum.

$$\begin{aligned} F(A,B,C) &= \\ F(0,1,2) &= -1 \quad (\text{Optimum}) \\ F(1,1,2) &= 0 \\ F(0,2,2) &= 0 \\ F(0,1,1) &= 0 \end{aligned}$$

The average step size can be found with, Equation (10):

$$\text{Average stepsize} = \frac{1}{\text{Population Size}} \sum_{k=0}^{\lambda} \text{Average } \Delta_k \quad (10)$$

Here, Average Δ_k is the normalized difference of the design vector k compared to the most optimal design vector (DV_{opt}) in the starting population and can be found with Equation (11):

$$\text{Average } \Delta_k(A_k, B_k, C_k) = \frac{1}{N_{variables}} \sum_{j=0}^{N_{variables}} \frac{DV_{j,k} - DV_{j,opt}}{DV_{j,upper} - DV_{j,lower}} \quad (11)$$

Here, $DV_{j,upper}$ and $DV_{j,lower}$ are the boundaries of design variable j .

For this example, this results in:

$$\text{Average } \Delta_k(A, B, C) = \frac{1}{N_{variables}} * \left(\frac{A_k - A_{opt}}{A_{upper} - A_{lower}} + \frac{B_k - B_{opt}}{B_{upper} - B_{lower}} + \frac{C_k - C_{opt}}{C_{upper} - C_{lower}} \right) \quad (12)$$

Looking at the average difference between the optimum $F(0,1,2)$ and the selected $F(0,1,1)$, one can obtain the following step size for the selected experiment. The average step size can then be easily computed by taking the average of all experiments. If the found average is lower than the stopping step size length, the optimization is ended.

$$\text{Average } \Delta_k(0,1,1) = \frac{1}{3} * \left(\frac{0 - 0}{1 - 0} + \frac{1 - 1}{2 - 1} + \frac{2 - 1}{2 - 0} \right) = \frac{1}{3} * \left(0 + 0 + \frac{1}{2} \right) = \frac{1}{6} \quad (13)$$

Table 18: Normalized delta between different design points and the most optimal point

	Objective value	Normalized delta variable A	Normalized delta variable B	Normalized delta variable B	Average delta all variables
F(0,1,2)	-1	0	0	0	-
F(1,1,2)	0	1	0	0	1/3
F(0,2,2)	0	0	1	0	1/3
F(0,1,1)	0	0	0	1/2	1/6
F(0,1,2)	1	1	1	0	2/3