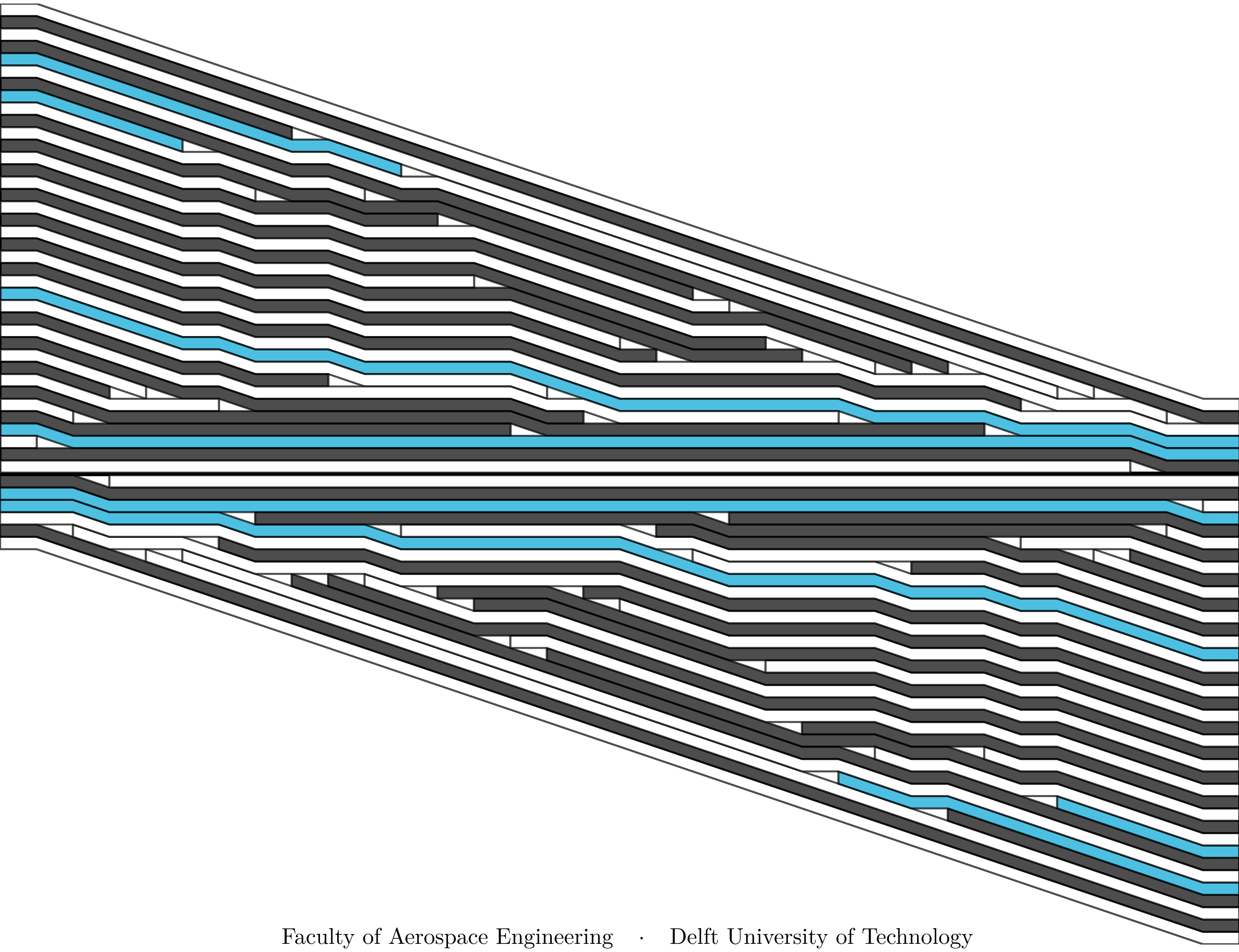
Master of Science Thesis

Stacking Sequence Retrieval of Large Composite Structures in Bi-step Optimization Strategies Using Mechanical Constraints

Frederico Vicente



Faculty of Aerospace Engineering · Delft University of Technology

TU Delft Delft University of Technology

Stacking Sequence Retrieval of Large Composite Structures in Bi-step Optimization Strategies Using Mechanical Constraints

Master of Science Thesis

For obtaining the degree of Master of Science in Aerospace Engineering
at Delft University of Technology

Frederico Vicente

03/09/2019

Faculty of Aerospace Engineering · Delft University of Technology

Delft University of Technology
Faculty of Aerospace Engineering
Department of Aerospace Structures and Materials

GRADUATION COMMITTEE

Dated: 03/09/2019

Supervisor:

Dr.ir. Roeland De Breuker, ASCM

Readers:

Dr. Julie Teuwen, AMT

Dr.ir. Julien van Campen, ASCM

Paul Lancelot, ASCM

Marco Tito Bordogna, ASCM

Summary

The optimization and design of large composite structures remains a difficult task due to the size and discrete nature of the design space of anisotropic materials. Formulation of the stiffness distribution in composite structures with lamination parameters alleviates this problem since the resulting design space is continuous and convex, in which case gradient optimizers are well suited to obtain optimal solutions at a low computational cost. This strategy is part of a class of composite design and optimization strategies known as multi-step optimization where a second optimization step is needed to translate the optimal lamination parameters to stacking sequences with an equivalent stiffness distribution. In this step, it's typical to impose manufacturing constraints in order to ensure that the retrieved designs are resilient and manufacturable.

These constraints lead to a mismatch between the feasible design spaces of the continuous and discrete optimization steps, often causing difficulty in the retrieval of equivalent stacking sequences. In large composite structures, spatial variation in stiffness can be achieved either through locally changing the number of plies or by fiber steering. In the former, ensuring fiber continuity throughout the structure (blending) further reduces the design space which in turn leads to a substantial increase in mass of feasible designs retrieved in the discrete step.

A method to include blending constraints in the continuous optimization proposed by Macquart et al. [1] has been shown to yield continuous optima with a more realistic stiffness distribution at a modest mass increase. In doing so, the mismatch in the feasible design regions between the continuous and discrete optimization steps is reduced and exploration of the design space in the vicinity of the continuous optima with lamination parameter matching objectives becomes a viable option.

To this end, an open-source stacking sequence retrieval toolbox coded in Python named **pyTLO** was developed for this thesis. This application relies on a genetic algorithm dedicated to stacking sequence tables (SST) encoding [2] which enables a straightforward implementation of various manufacturing constraints in the discrete optimization step and guarantees that all retrieved stacking sequence designs are blended. Other core criteria of the pyTLO application include computational efficiency and the ability to carry out simultaneous optimization of multiple composite components, each with an individual SST.

The work presented here focuses on the stacking sequence retrieval of large composite structures in multi-step optimization schemes through lamination parameter matching of a continuous LP optimum. The aim was to explore the vicinity of the optima obtained in the continuous optimization and retrieve stacking sequences with a similar stiffness distribution and mechanical behavior without a substantial increase in mass. This was achieved by using pyTLO and subjecting simple lamination parameter objective functions to mechanical constraint penalties. By penalizing solutions with failed constraints, the genetic algorithm is steered towards designs with improved mechanical performance in the vicinity of the continuous optima.

A novel constraint approximation method based on Kreisselmeier-Steinhauser (KS) envelope functions which agglomerates multiple local linear approximations into a single conservative approximation was used. It proved effective in the retrieval of feasible stacking sequences of an ONERA regional wing model described by 396 design variables without requiring refined sampling of reference points. Even in circumstances where the evaluated discrete design points had a large error with respect to the lamination parameters of the continuous optimum, the KS approximations where sufficiently accurate to steer the discrete optimization towards feasible designs.

A total of three different feasible stacking sequence solutions for the ONERA regional wing model are presented in the report. Two solutions were obtained for a continuous optimum including blending constraints, whereas the third solution was obtained for an optimum with no blending constraints. It was demonstrated that although the use of blending constraints in the continuous optimization imparted a 5.7% mass increase over the continuous result with no blending constraints, stacking sequence retrieval yielded feasible designs that were up to 18% lighter than those retrieved from the continuous optimum with no blending constraints.

Table of Contents

List of Figures

List of Tables

Acknowledgements

I would like to thank Marco Tito Bordogna for his support, feedback and ideas throughout the project that made sure I didn't go around in circles for too long. Another round of appreciation goes for Paul Lancelot for his continuous support, availability and his vast database of scientific papers neatly stored somewhere in his head. I would also like to thank Roeland De Breuker for his supervision throughout the project, as well as Julie Teuwen and Julien van Campen for being part of the graduation committee and taking the time to read this report which I hope is worthwhile.

Finally, I express my gratitude to ONERA for providing access to their Finite Element Model of the ONERA regional wing model which was invaluable in its contribution to this project.

Chapter 1

Introduction

Composite materials are well known for superior mechanical properties such as stiffness- and strength-to-weight ratio when compared to metals. Coupled with their anisotropic nature, a higher structural efficiency can be achieved through the ability of tailoring the strength and stiffness direction of designs for an optimal load distribution. Where aircraft design is concerned, the increased tailoring capabilities open up additional possibilities in the design of lighter aircraft wings by enabling designers to carefully control its shape under aerodynamic loads for optimal performance and stability in a range of flight phases.

These advantages are especially valuable to the aerospace industry, where minimizing weight is one of the core criteria in the design of aircraft. The use of composite materials in the secondary and primary structures of commercial aircraft has been increasing over the past decades, with both Boeing and Airbus manufacturing aircraft containing composite structures accounting for over 50% of their weight. This rise can be attributed to rising fuel costs, tighter environmental regulations which pressure manufacturers to design lighter and more efficient airplanes. Innovation in industry automation an manufacturing of composite structures also contributed by making them more cost-effective.

The potential gains in weight and performance derived from using composite materials in aerospace structures is entirely dependent on the level of our understanding of various factors surrounding the design, manufacturing and operation of composite structures. As it stands, the design of composite structures in aerospace relies on several knock-down factors on material properties that ensure that the gaps in that knowledge never become a problem during operation, but this compromises performance gains over metals [4]. Defining robust optimization strategies that factor in manufacturing constraints and are able to consistently retrieve optimal designs can help to fill in some of these gaps and lead to significant gains in structural efficiency.

Even though composite structures are widely applied in the industry, their design and optimization remains a challenge. Anisotropy is a 'double edged sword'; although tailoring the stiffness distribution can enhance structural design and with it lower weight can be achieved, it comes at the cost of a much larger number of design variables. Take for instance composite laminate structures, which are the most frequently used type of composite materials in aerospace. These laminates are built with thin layers of fibers embedded in a plastic matrix that are stacked in various fiber orientations to obtain the desired directional stiffness and strength. For a single stiffness region, the number of design variables needed to describe a laminate becomes 2N, where N is the total number of layers used and each is described by its fiber orientation and thickness. In highly-loaded design problems where hundreds of plies can be necessary, the number of design variables quickly explodes.

Another disadvantage in the design of stacking sequences is that the design space is discrete since the number of layers is defined by an integer and fiber orientation is restricted to specific orientations due to manufacturing constraints, which rules out gradient based optimization as a means to determine design optimums. Instead, direct search methods such as evolutionary algorithms are used. These methods are good at exploring the design space since there's a low risk of being trapped in local optima, but become less effective at searching it entirely as the design problems become larger and more complex. As a consequence, retrieved optima are never guaranteed to be global and these algorithms are plagued by long run times if finite element analysis of the designs is involved.

In order to mitigate some of these issues, it is possible to represent the stiffness of laminates not in terms of their stacking sequence but with lamination parameters (LP) formulation. Proposed by Tsai and Pagano [5], LP make it possible to describe the mechanical properties of any laminate with 12 parameters in addition to its thickness, independently of the number of layers and orientations it contains. The 12 parameters are divided in three sets of four variables - V^A_{1-4} are used to describe the membrane stiffness, V^D_{1-4} the out-of-plane stiffness and V^B_{1-4} the coupling between the two former sets. In addition to the reduced number of design variables, LP-space is continuous and convex. Thus, gradient-based optimization can be used to explore the design space effectively and at a low computational cost [6].

In this context, the most common type of optimization is a multi-step strategy where a first step is carried out in the continuous design space step followed by a second step in the discrete space that translates the results of the first step to equivalent stacking sequences. An exact match between the two is possible provided that there is no restriction on layer orientation and thickness, but that is not the case due to manufacturing constraints.

In structures where load varies spatially, it is typical to define multiple regions with changing mechanical properties in order to improve structural efficiency. In composite laminates, the changes in properties are achieved by adding or dropping plies between adjacent regions which introduces stress concentrations and material discontinuities. In order to avoid premature failure, fiber continuity and the severity of the stress concentrations due to ply drops are considered in the design process by a method referred to as blending of variable stiffness composite structures. The additional blending requirements constrain the design space further and the mismatch in feasible design regions between the two steps becomes too large for the discrete step to succeed in finding equivalent stacking sequences. Results obtained through stiffness or LP-matching are often inadequate and unable to meet strength requirements without a substantial increase in weight.

A proposal by Terence et al. [7] to include blending constraints in the continuous design space has been shown to yield optimal LP-space designs with a smoother and hence more realistic stiffness and thickness distribution in variable stiffness composites. By reducing the mismatch between the continuous and discrete feasible regions, stacking sequence retrieval through LP-matching is more likely to succeed in retrieving equivalent stacking sequences. In the discrete step, improvements on blending algorithms such as Irisarri's Stacking Sequence Tables (SST) [2] have made it possible for large composite design problems to be represented with a relatively low number of design variables while providing inherent blending and a straightforward framework in the application of several manufacturing constraints in stacking sequence design.

1.1 Project Goals

The aim of the thesis project was to revisit stacking sequence retrieval of large composite structures with a GA by LP-matching of continuous optima subjected to blending constraints. By subjecting the LP-matching objective to mechanical constraints such as strength and buckling in the discrete step of the optimization, it is expected that the GA should be steered towards designs in the vicinity of the continuous optimum with improved mechanical behavior. For this purpose, two main research questions were devised to guide the project:

1. Can LP-matching fitness strategies subjected to mechanical constraints improve the mechanical performance of Variable Stiffness (VS) composite laminates retrieved by a genetic algorithm?
2. What is the increase in mass of an aircraft wing design over its continuous LP optimum with a LP-matching fitness function subjected to mechanical constraints?
   (a) What is the change in the mass increase of wing designs obtained from a continuous target with blending constraints vs. a target without blending constraints?

A regional aircraft wing model is considered in this project which was provided by ONERA[1] together with a ready-to-use FE model. Finally, an open-source python based GA for stacking sequence retrieval using SST encoding was developed in the course of the project in order test various objective functions and to evaluate the effects of blending constraints on the retrieval stacking sequences.

1.2 Report Outline

In chapter 2 a more in-depth introduction is given to the optimization of large composite structures with focus on multi-step strategies. In chapter 3 a description of the python tapered laminate optimizer (pyTLO) developed for this project is given where its computational performance is also evaluated. The 18-panel horseshoe benchmark is used to evaluate the effectiveness of LP-matching objectives subjected to mechanical constraints in chapter 4, by comparing retrieved results to an example available in the literature. Then, in chapter 5 a comparison of stacking sequence retrieval of a 4-component aircraft wing known as the ONERA regional wing model is carried out with different fitness functions under a constant mass condition.

Finally, the mass increase necessary to retrieve feasible stacking sequences for the ONERA regional wing model is discussed in chapter 6 followed by a summary of the report's conclusions and general recommendations for future research and development in chapter 7.

[1]The French Aerospace Lab

Chapter 2

An Introduction to Composite Structural Optimization

The design and optimization of composite structures, especially where variable stiffness laminates are considered, is a challenging task due to the anisotropic nature of composite materials. Several strategies have been proposed in the literature for which references [8, 9] provide a thorough overview. In this report, the scope is narrowed to a class of composite design methods known as multi-step optimization.

These strategies comprise of an initial optimization step carried out using continuous intermediate design variables, generally LP, followed by a second step in discrete space where stacking sequences that best comply with the structural properties derived in the first step are retrieved. Successfully translating the results of the continuous step to feasible [1] laminates remains a difficult task due to additional design criteria that are not imposed in the first step. The focus of this work relies on combining traditional LP Root Mean Square (RMS) error objective functions with response constraints in the discrete step in order to steer a GA towards feasible stacking sequence designs.

The different classes of composite optimization are expanded in section 2.1 where concepts such as variable stiffness (VS) and Constant Stiffness (CS) are explained, followed by an introduction to lamination parameters in section 2.2. In section 2.3 an introduction to multi-step optimization techniques based on lamination parameters and the challenges related to stacking sequence retrieval are discussed. A number of design guidelines aimed at ensuring manufacturability and robustness of composite laminates such as blending is presented in section 2.4. The encoding method used in this thesis to retrieve blended stacking sequences for VS composite structures known as SST is covered in section 2.5. Finally, response approximation schemes are presented in section 2.6 in addition to the relevance of these topics in the goals of the project discussed in this report.

[1] Meeting the design criteria and being manufacturable.

2.1 Variable Stiffness Laminates

The design and optimization of composite laminate structures can be classified in two different categories as proposed by Setoodeh et al. [8, 10]:

1. Constant stiffness CS: Both the stacking sequence and thickness distribution are uniform throughout the entire composite structure.

2. Variable Stiffness VS: The composite structure contains multiple regions where the mechanical properties are allowed to vary as a function of the loads it's subjected to. Hence, both the thickness distribution and stacking sequences can vary throughout the structure.

Designing VS composite laminates implies discretising it into multiple regions of different stiffness according to load distribution and\or geometric constraints. The stacking sequences of these regions can then be optimized for the best performance subjected to a number of manufacturing constraints such as "blending" [11] to ensure that structural integrity is not compromised. Conventional VS laminate stacking sequences make use of straight fiber layers which can be placed at different orientations with respect to one another, and stiffness changes are obtained by adding or removing layers in the different design zones. This is referred to as patch design. Alternatively, non-conventional laminates employ fiber or tow steering to gradually change the fiber orientation throughout the layers themselves.

Although fiber steering enables a higher potential in the tailoring of composite structures due to the additional design freedom, the number of design variables to consider is also larger than straight-fiber stacking sequence optimization. Furthermore, fiber-steered large composite structures are not yet a viable option in the aerospace industry due to added complexity and costs in design and manufacturing in addition to the current certification framework being aimed at conventional laminates [12]. For these reasons, fiber-steered large composite aerospace structures is considered to be at a lower Technological Readiness Level (TRL) than conventional straight-fiber layers and in the work presented here it is the only method considered in the design of VS laminates.

The optimization of composite structures poses a difficult challenge due to the anisotropic nature of the material which provides a higher potential in tailoring the design, but a larger number of design variables need to be considered. For instance, in the design of constant stiffness laminates the number of variables grows with $2N$, where N is the number of layers for which the material and orientation need to be selected. As a result, hundreds of variables need to be defined in problems concerning large composite structures where laminates contain a large number of plies. This problem is further exacerbated in the design of variable stiffness composite structures since multiple different stacking sequences need to be defined. Another issue with optimizing composite laminates in terms of ply material and orientation is that the design space is discrete and difficult to explore since the optimization relies on direct search methods such as GA that do not guarantee global optimums are obtained [8]. The root cause is that GA are not deterministic, so various runs yield different results and only with an infinite number of runs would one be able to confirm whether a result is a global optimum or not.

2.2 Lamination Parameters

The stiffness distribution of composite structures can be represented in terms of a set of intermediate design variables known as lamination parameters. This method is independent of the laminate's stacking sequence and thus reduces the number of variables in composite design problems. LP were proposed by Tsai and Pagano [5] and are obtained by integrating the layer angles of a given laminate through thickness h as given by equation 2.1.

$$\begin{aligned}
\left(V_1^A, V_2^A, V_3^A, V_4^A\right) &= \frac{1}{h}\int_{-h/2}^{h/2} (cos2\theta, sin2\theta, cos4\theta, sin4\theta)\, dz \\
\left(V_1^B, V_2^B, V_3^B, V_4^B\right) &= \frac{4}{h^2}\int_{-h/2}^{h/2} z\,(cos2\theta, sin2\theta, cos4\theta, sin4\theta)\, dz \\
\left(V_1^D, V_2^D, V_3^D, V_4^D\right) &= \frac{12}{h^3}\int_{-h/2}^{h/2} z^2\,(cos2\theta, sin2\theta, cos4\theta, sin4\theta)\, dz
\end{aligned} \tag{2.1}$$

The three sets of lamination parameters have a similar representation to that of the $[ABD]$ matrix in the sense that V_n^A set represents the membrane stiffness, V_n^D the bending or out-of-plane stiffness and the V_n^B set represents the response couplings between the membrane and bending stiffness. The $[ABD]$ matrices can be derived from the LP, the material invariant matrices and laminate thickness as shown in equation 2.2. The material invariants depend only on ply material and are not considered as variables during optimization.

This method makes it possible to describe the stiffness of a laminate independently of its stacking sequence, which can considerably reduce the number of design variables. For any number of plies a laminate contains, the 12 LP in addition to the thickness are sufficient to determine the laminate stiffness. This number is further reduced when only symmetric laminates are considered, since the in- and out-of-plane response coupling terms $V_n^B = 0$ and $[B] = 0$. Constraining the search to balanced designs leads to only 4 LP in addition to thickness being needed, since (A_{16}, A_{26}) and bending-twist (D_{16}, D_{26}) response couplings equal zero. The same applies to lamination parameters, i.e., $(V_2^A, V_4^A) = 0$ and $(V_2^D, V_4^D) = 0$. This is a clear advantage over stiffness as a function of stacking sequences, especially where thicker laminates are concerned.

$$\begin{aligned}
A &= \frac{1}{h}\left(\Gamma_0 + \Gamma_1 \cdot V_1^A + \Gamma_2 \cdot V_2^A + \Gamma_3 \cdot V_3^A + \Gamma_4 \cdot V_4^A\right) \\
B &= \frac{4}{h^2}\left(\Gamma_0 + \Gamma_1 \cdot V_1^B + \Gamma_2 \cdot V_2^B + \Gamma_3 \cdot V_3^B + \Gamma_4 \cdot V_4^B\right) \\
D &= \frac{12}{h^3}\left(\Gamma_0 + \Gamma_1 \cdot V_1^D + \Gamma_2 \cdot V_2^D + \Gamma_3 \cdot V_3^D + \Gamma_4 \cdot V_4^D\right)
\end{aligned} \tag{2.2}$$

Other advantages in using LP to describe the stiffness properties of the laminates lie in the fact that the design problem can be formulated by continuous design variables alone and that the resulting design space is convex [6]. This enables the use of fast converging gradient-based optimization strategies which are efficient in finding optimal design points at a relatively low computational cost even for complex VS composite design problems [9]. Lamination parameter optimization is generally used as a multi-step optimization strategy since a second step is necessary to retrieve stacking sequences that best match the design variables obtained in the first optimization step.

2.3 Multi-Step Optimization strategies

Structural optimization is employed with the aim of maximizing a design's performance subjected to a number of constraints such as buckling or strength. A general formulation for optimization problems can therefore be expressed by equation 2.3 where f_{obj} is an objective function of a given property and x represents a set of design variables bounded by a lower and upper limit x^l, x^u. The terms g_i and K express design constraints and their threshold, respectively.

$$\begin{gathered} min\,(f_{obj}(x)) \\ g_i \leq K \;\; i = 1, ..., N \\ x^l \leq x \leq x^u \end{gathered} \tag{2.3}$$

This principle applies whether the optimization is carried out in a continuous or a discrete design space.

Multi-step composite optimization relies on two or more steps to retrieve feasible designs. The first step concerns a structural optimization in a continuous design space using LP, in which gradient-based optimizers are used to derive optimal design points using sensitivities of the constraints with respect to LP and thickness to guide the optimization towards feasible solutions. Although these methods display high convergence rates, they are highly dependent on the starting position in the design space and tend to converge to local optima in nonconvex problems. However, since the stiffness of composite laminates depends linearly on LP as seen in equation 2.2, problems formulated in terms of lamination parameters result in a convex design space [13] which can be efficiently explored with gradient-based solvers. Nevertheless, the optimization of composite structures based on LP formulation is not without limitations.

Firstly, LP are coupled so an independent variation of the parameters can lead to unfeasible design points that don't yield a positive-definite stiffness matrix. The definition of feasible regions in LP formulation is therefore important but not entirely solved as no analytical method is available to define the feasible envelope of in- and out-of-plane lamination parameters simultaneously. A numerical method has been proposed by Setoodeh et al. [14] to approximate the feasible envelope and represent it as a set of linear inequality constraints which can be used with gradient-based solvers.

Secondly, design optima derived in the continuous step need to be translated to stacking sequences as LP by themselves only describe the stiffness distribution and hold no bearing on whether the solution is manufacturable or not. Furthermore, an exact match between LP and retrieved stacking sequences is only guaranteed provided that the ply orientations and thickness are taken as continuous variables, which is not the case in the design of composite laminates.

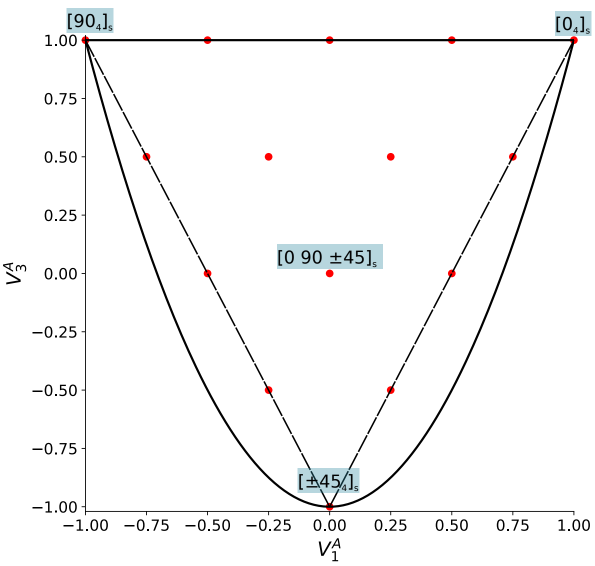


Figure 2.1: Continuous and discrete design space of the in-plane parameters V_1^A and V_3^A.

Generally, plies are limited to a set of angles and thickness to ensure manufacturability and this results in a reduction of the feasible design space between the continuous and discrete optimizations. As a consequence, stiffness and load distribution can be entirely different in retrieved stacking sequences resulting in poor performance or even premature failure under design loads. This can be visualized in figure 2.1 where the feasible region of the in-plane lamination parameters V_1^A and V_3^A is plotted for balanced-symmetric laminates [15,16]. Whereas the continuous design space is the entire area enclosed by the quadratic function, by restricting the permissible ply angles to combinations of [0, 90, ±45] layers in the discrete step, the feasible space is reduced to the triangular area within the dashed lines assuming an infinite number of plies. As the number of plies is restricted, the design space becomes discrete. For an 8-ply laminate under balance and symmetry constraints, the only possible combinations are represented by the red dots.

Since the second step is carried out in a discrete design space, gradient solvers are no longer an option and direct search methods are used instead. GA optimizers are especially popular in the design of composite structures due to their ability to search the design space without becoming trapped in local optima. GA can be compared to a simplified version of the theory of evolution where a “population” made up of encoded design points as chromosomes goes through the process of selection, mating and mutation over a number of generations to induce semi-random variations in the designs and guide it towards feasible solutions.

As the problem dimensionality grows, the effectiveness of GA is diminished due to the high number of evaluations required. Furthermore, GA tend to incur a high computational cost due to the large number of design points evaluated in a single optimization run, especially if Finite Element Analysis (FEA) is employed to calculate the structural response. For this reason, approximations of the response are used extensively over Finite Element Modelling (FEM) to verify that structural constraints are satisfied during the optimization.

2.4 Design Criteria of Composite Laminates

The design of composite laminates is challenging not only due to the many design variables that need to be considered, but also due to weaknesses related to anisotropy such as interlaminar stresses and damage tolerance. When compared to metals, additional care in the design phase of composite laminates is needed to ensure that these weaknesses are patched up to avoid premature failure of the designs. To this end, a number of best practices has been compiled by the industry pertaining the design of composite structures [17,18] covering a large range of criteria in both micro and macro design details. Given that the optimization and stacking sequence retrieval is generally done at a preliminary design level, the guidelines more commonly applied at this stage in the design of stacking sequences are as follows [2]:

1. **Symmetry:** Ensures that the stacking sequence of laminates is symmetric about the mid-ply, with the goal of decoupling membrane and bending responses to loading. As a result, the B-matrix and the 4 V_n^B terms equal zero and can be neglected in the design process.

2. **Balance:** For every ply angle $\theta \neq \{0°, 90°\}$, a $-\theta$ ply is present in the stacking sequence. As a consequence, tension-shear and bending-twist response couplings equal zero. Although this is beneficial in most designs, it has been shown that unbalanced designs are advantageous in the aeroelastic tailoring of aircraft wings since the shear-twist couplings of the skins can be used as a passive load relief mechanism and improve divergence characteristics [19].

3. **Damage Tolerance:** Outer layers of the laminate should be comprised of ±45° plies which provide better impact resistance by protecting the main load carrying layers . A positive side-effect of this design rule is that buckling resistance is improved by placing ±45°layers away from the mid-ply.

4. **The 10% rule:** The stacking sequence should contain at least 10% of plies in the primary directions (0°, 90°, ±45°) to prevent premature failure due to unexpected secondary loads and avoid matrix-dominated directions in the laminate. Since this guideline can constrain the ply angle selection considerably, a guideline based on minimum in-plane stiffness has been proposed by Abdalla et al. [20].

5. **Disorientation:** In order to avoid high interlaminar stresses which can lead to delamination between layers, the angle difference between adjacent plies should be a maximum of 45°.

6. **Ply Contiguity:** Clusters of plies with equal orientation should not exceed 2 to 4 [2,17] equal plies to limit the growth of matrix cracks across multiple layers.

Where the design of VS composite laminates is concerned, additional criteria is considered to mitigate the effects of ply drops between adjacent patches which introduce out-of-plane loads locally and to ensure fiber continuity throughout the structure:

1. **Covering:** A continuous layer should cover the inner and outer surfaces of the laminate to avoid exposed dropped plies.

2. **Internal Continuity:** There should be a continuous layer for every three dropped plies.

3. **Maximum Ply Drop:** A limit on the number of dropped plies between adjacent panels should be used to minimize stress concentrations and to ensure a more gradual load distribution. This guideline is known as the Δn rule.

4. **Ply Continuity:** Plies present in the thinnest patch of the structure should be continuous throughout the entire structure. Furthermore, discontinuities in fiber path between adjacent patches must be avoided to maintain structural integrity as well as to guarantee manufacturability of the stacking sequence design. This requirement is also known as blending and is covered in more detail in section 2.4.1.

2.4.1 Blending

Efficient design of light-weight aerospace structures calls for a equally loaded material mindset which is achieved by defining regions of different stiffness throughout the structure to accommodate a varying load distribution. This mesh of different stiffness regions can be more or less refined depending on the problem complexity and computational constraints as the mesh size dictates the number of design variables that need to considered.

Each of these regions can be represented by a stacking sequence of straight fiber ‘patches’ in the design of composite laminate structures which can be modified by adding or removing plies to accommodate the stiffness layout of the different regions. However, the variation of these properties between adjacent patches lead to discontinuities in the fiber path, leading to stress concentrations and introduces a taper between adjacent regions. This taper results in a local strength reduction and out-of-plane stresses that contribute to delamination effects [21, 22]. Since these discontinuities are unavoidable in the quest for structural efficiency, it is necessary to consider strategies to optimize layer continuity and ply drops in VS composite laminates and this process is known as ‘blending’.

A number of different approaches to blending in the discrete optimization step have been developed. One of the first methods was proposed by Liu and Haftka [23] where a continuity measure is used between two adjacent laminates that controlled both the fiber continuity at the global level [2] and the stacking sequence at the local level. Minimum continuity constraints

[2] Properties affecting all or a majority of the sub-regions of a structure

were then used in a two-step optimization with a GA for stacking sequence retrieval where it was shown that relatively high fiber continuity is possible throughout the structure without considerable impact on weight.

Kristinsdottir et al. [11] introduced a 'greater-than-or-equal-to' blending strategy where plies could be progressively dropped from the most loaded towards the least loaded patches. The achievable structural efficiency was limited since removed plies could not be re-added, resulting in low load regions to be over-designed. A two-step strategy based on sub-laminates was proposed by Soremekun et al. [24] in 2002, where sub-laminates covering the entire structure are defined by the thinnest region in the first step, followed a second step where remaining sub-laminates are defined for the remaining unassigned thickness regions. Finally, these sub-laminates are optimized by a GA until a satisfactory solution is retrieved. In the same paper a benchmark problem to gauge the effectiveness of different blending strategies known as the '18-panel Horseshoe' problem was introduced.

A guide-based strategy was proposed by Adams et al. [3] where VS composite laminates are described by a guide stacking sequence matching the thickest region in the structure and subsequent ply drops. The number of design variables is considerably reduced with respect to other methods since all stacking sequences are a derivative of the guide in addition to the ply drops. Another advantage of this method is that all solutions are inherently blended which removes the need for blending constraints in the GA. Two types of guide-based blending were introduced: Outer- and Inner-blending, which depends on whether the outermost or innermost plies can be dropped, as shown in figure 2.2.

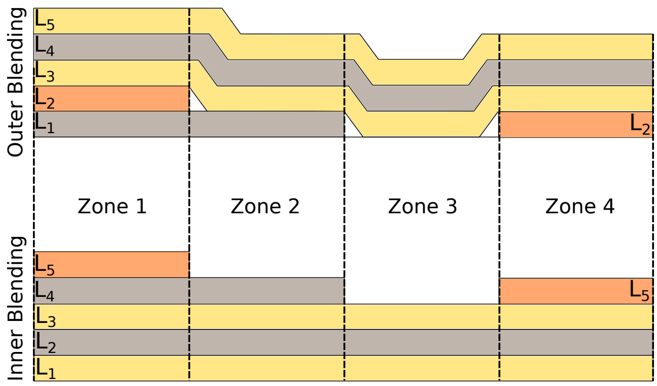


Figure 2.2: An Outward and Inward blending example.

Although the guide and sub-laminate strategies can both return blended designs, the latter would not be considered blended under the terms of the guide-based approach. This indicates that some blending definitions are too limited over what constitutes a blended VS laminate. Accordingly, Van Campen et al. [25] proposed two blending definitions that cover any potential blended designs, namely generalized blending, which is similar to the guide-based approach proposed by Adams et al. [3], and a relaxed blending definition that is the least constrained blending type, allowing for patches of the same thickness to have different stacking sequences and still be blended at the global level. Any stacking sequence retrieval algorithm for tapered laminate structures invariably falls into one of these categories.

The *generalized blending definition* ensures the continuity of the layers of the thinnest stacking sequence throughout the whole structure independently of its position in the stacking sequence unlike Adams' guide-based method where plies are only dropped at the outer or inner surface of the laminate as shown in figure 2.3 where the two definitions are compared. The generalized blending definition is succinctly described by van Campen et. al:

> "*Generalized blending definition: We consider two adjacent panels completely blended if all the layers from the thinner panel continue in the thicker one regardless of their position along the thickness of the laminate.*" [25]

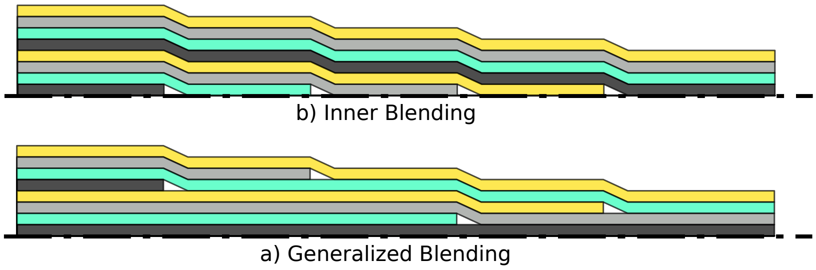


Figure 2.3: Two blended designs. Design (a) would not be considered blended according to inner or outer definitions defined by Adams et al. [3].

The encoding of tapered laminates used in this thesis is based on a method known as SST which falls under the generalized blending category. Some constraints over blending designs still remain in this definition, such as the inability to contain different stacking sequences for different regions of the same thickness or to combine multiple individually blended designs into a single blended structure. Hence, the relaxed blending definition relies on a single criteria: A tapered composite laminate design is blended if there are no layer edges in contact in any of the interfaces between different design regions as shown in 2.4.

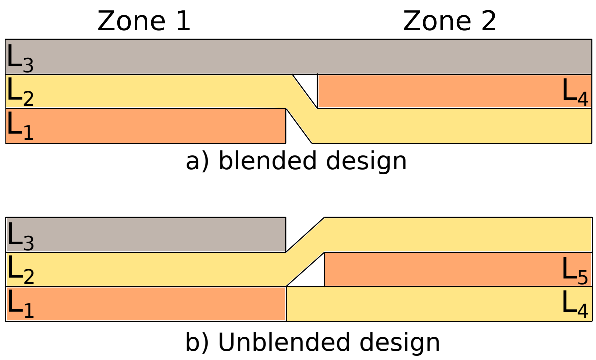


Figure 2.4: Example of a blended and unblended design using the relaxed definition. Design b) is not blended due to layers 1 and 4 having their edges in contact.

2.4.2 Continuous Design Space Blending Constraints

The blending requirement introduces constraints in the design space which reduces its feasible region in comparison to LP-space. This is in part due to the fact that in the continuous step there is no restriction on the variation of lamination parameter values between adjacent regions since they're optimized independently, whereas in the discrete design space, blending of stacking sequences between adjacent regions implies similarity to one another and thus the change in mechanical properties is limited.

As a result, the retrieval of stacking sequences in the context of multi-step optimization schemes rarely returns feasible solutions as the constraints make it impossible to match the variation in LP and stiffness direction throughout the laminate. In order to improve stacking sequence retrieval from a continuous optimum, Macquart et al. [7] proposed a set of blending constraints applied in the continuous design space that restrict the variation of the LP between adjacent regions as well as the number of allowed ply drops.

Although this leads to an increase in mass on the optimal design points obtained in LP-space, a smoother and more realistic stiffness distribution is achieved and stacking sequence retrieval in the discrete step is more effective in finding feasible solutions similar to the optima obtained in the continuous optimization. Their effectiveness in reducing the LP error of retrieved stacking sequences in subsequent discrete optimizations has been demonstrated in a number of studies; Bordogna et al. [26] achieved a 45% reduction in the LP RMS error between the continuous optimum and retrieved stacking sequences, as well as a lower number of failed elements and failure index of mechanical constraints.

More recently, Silva et al. [27] included blending constraints as well as ply percentage constraints limited to standard [±45°, 0°, 90°] ply orientations in the continuous design space and were able to retrieve feasible designs of a regional aircraft wing with only a 1% weight increase over the continuous optimum. Other manufacturing constraints have a similar effect on constraining the feasible design space of the discrete step, but so far no corresponding constraints have been developed in order to achieve a more realistic representation of continuous design spaces [28].

2.5 Stacking Sequence Tables

Stacking sequence tables are based on lay-up tables used in the manufacturing of composite laminates by the aerospace industry [29].SST have also been used by Meldrum et al. [30], on the behalf of Dassault Aviation, to carry out aero-structural optimization in an automatic process to guide the initial structural sizing and weight of aircraft structures with promising results. SST are a guide-based method following a generalized blending approach that represents the stacking sequences and ply-drops of variable stiffness composite laminates in a table format. Each table contains all stacking sequences between an upper and a lower bound of the number of plies [N_{max}, N_{min}] in the composite laminate with ply-drop locations starting from a guide laminate as shown in figure 2.5. This representation has a number of advantages; all SST solutions are inherently blended and the table layout simplifies the implementation of manufacturing guidelines in the design process.

Stacking Sequence Table (SST)

	N_{max}=16					N_{min}=8
	-45	-45	-45	-45	-45	-45
	0					
	+45	+45				
	90	90	90	90	90	90
	-45	-45	-45			
	0	0	0	0		
	+45	+45	+45	+45	+45	+45
Symm.	0	0	0	0	0	0

Figure 2.5: A Stacking Sequence Table example.

Irisarri et. al [2] proposed an encoding of the SST which is suitable for stacking sequence retrieval using GA. In order to describe the stacking sequence of any design point representing a multi-patch composite laminate, only three one-dimensional vectors or chromosomes are needed:

- **$\mathbf{SST_{lam}}$** : Represents the stacking sequence of the thickest laminate ($N_{ply} = N_{max}$). The vector length is equal to the number of plies in that patch, or half that length if symmetric laminates are considered.

- **$\mathbf{SST_{ins}}$**: Represents the rank of insertion of each ply w.r.t. the thinnest laminate. Zero entries denote plies of the thin guide and thus are present in all patches. Non-zero entries represent the lowest thickness in which the ply is present. Same vector length as SST_{lam}. Unlike the SST method proposed by Irisarri et al., the SST_{ins} can have plies with the same rank of insertion where balanced laminates are concerned since non-balanced columns are not encoded.

- $\mathbf{N_{str}}$: Contains the number of plies per patch. The vector length is equal to the number of patches in the structure.

This encoding considerably decreases the number of design variables needed to describe a multi-patch problem. Additionally, since the decoding of the SST results in inherently blended designs, it is not necessary to subject the GA to continuity constraints which improves computational performance and the exploration of the design space.

The encoding of the SST is exemplified in figure 2.7 where the resulting SST vectors of a four patch laminate given in figure 2.6 are displayed for a balanced and symmetric laminate. It also represents a compressed form of the table shown in 2.5 where all non-balanced stacking sequences are removed.

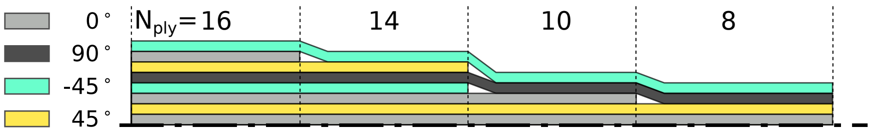


Figure 2.6: A 4-patch stacking sequence (symmetric).

Stacking Sequence Table (SST)

SST_{lam}	SST_{ins}	N_{ply}	16	14	10	8
-45	0		-45	-45	-45	-45
0	16		0			
+45	14		+45	+45		
90	0		90	90	90	90
-45	14		-45	-45		
0	10		0	0	0	
+45	0		+45	+45	+45	+45
0	0		0	0	0	0

Figure 2.7: A Stacking Sequence Table of a 4-patch problem (symmetric).

The encoded solutions are then used in a GA where the three vectors are used to fully describe a design point. These vectors can be seen as chromosomes that are subjected to evolutionary operators in order to induce variations on the design points to explore the design space.

Although there is a gain in computational performance as a result of the compact encoding, this comes at a cost of a constrained design space typical of guide-based blending strategies. Patches of the same thickness are forced to have the same stacking sequence even with different load distributions which restricts local optimization in favor of inherent blending. Other types of blending encoding are available and provide more design freedom, such as a global shared layer blending method developed by Jing et al. [31–33] and a genetic algorithm coupled cellular automaton method developed by van den Oord [34]. However, unlike guide-based methods, these do not always guarantee that the retrieved designs are fully blended and hence are not considered here.

2.6 Response Approximations and Constraints

The optima obtained in continuous LP design space can be considered as the theoretical design performance ceiling for a given composite structure. It stands to reason that approaching the discrete optimization step with a RMS error problem of the continuous and discrete set of lamination parameters (eq. 2.4) should provide feasible VS stacking sequences. However, due to the reduction of the feasible design space imposed by blending and manufacturing constraints, RMS error solutions fail to guarantee valid designs as the error between LP is too large to draw any parallel between the solutions of the two steps.

$$LP_{rmse} = min\left(\frac{1}{N_p}\sum_{n=1}^{N_p}\sum_{j=1}^{J}\sqrt{\frac{(LP_{j,GA}(x_i) - LP_{j,opt})^2}{J}}\right) \tag{2.4}$$

Here, LP_j represents a single lamination parameter of a design retrieved in the GA and $LP_{j,opt}$ is the equivalent target parameter obtained in the continuous optimization. N_p represents the total number of patches in the structure. For symmetric laminates, $J = 8$ and for balanced symmetric laminates, $J = 4$. Additionally, the formulation of the objective function given by equation 2.4 acts under the assumption that all patches and LP have equal weight on performance. Indeed, two solutions with the same RMS error can be completely different and unable to satisfy constraints.

For feasible solutions to be retrieved additional information on the design requirements needs to be added to the objective in the form of constraints. These can be either **direct** or **indirect**. Direct constraints basically act as a filter that removes non-compliant designs from the population, whereas the latter is applied as a penalty in the fitness of the design but does not remove it from the population. In the present work, both types of constraints are enforced. Manufacturing constraints are applied as direct constraints, whereas active mechanical constraints [3] are applied indirectly in the objective function as shown in equation 2.5.

$$f(x_i)_{GA} = min\left(LP_{rmse} + \sum_{j=1}^{J}\delta(\tilde{R}_k(x_i) - R^t)\right) \tag{2.5}$$

Given that the approximation is built on continuous design points, the introduction of indirect constraints in the objective function acts as a weight on the importance of each design variable by penalizing solutions with active constraints and push the GA to search for specific RMS error solutions that lower their number and/or magnitude.

2.6.1 Response Approximations

Response approximations are an effective tool in evaluating constraints at a fraction of the cost of FEM calculations. The approximations are a function of the design variables and sometimes the derivatives of reference design points. Several approximation methods are available which can be broadly classified under two categories [35].

1. **Local Approximations:** Responses are approximated with respect to a reference design point such as optima retrieved in previous runs and are only accurate near the reference. Linear approximations based on a Taylor expansion of the response are the most common type used in structural approximations due to their low computational cost and have been covered extensively in the literature [35, 36].

[3] Constraints such as buckling or strength that are above its failure threshold for a given design.

2. **Global Approximations:** Valid in the entire design space, these methods rely on building a surface response approximation through interpolation based on multiple design points obtained by sampling. For complex design problems with a large number of design variables, extensive sampling is required for accurate approximation and thus the computational costs of building such models can become prohibitive.

The creation of a surface response approximation is outside the scope of this project and only local approximations are considered in the remainder of this report. More specifically, linear approximations around reference design points are used to evaluate buckling, strength and displacement responses in the discrete optimization step.

Linear approximations of a response can be obtained by a Taylor's expansion restricted to the first order terms as shown in eq. 2.6. R_0 is a response evaluated at a reference design point such as the optima obtained in the continuous design point described by the set of design variables x_0. The sensitivities of the response $\frac{\partial R}{\partial x_i}$ are obtained in the continuous optimization and used as reference to build the approximation. Higher order terms can be considered to improve accuracy but incur a higher computational expense.

$$\tilde{R}(x_i) = R(x_0) + \sum_{i=1}^{N} \left.\frac{\partial R}{\partial x_i}\right|_0 \cdot (x_i - x_{0i}) \tag{2.6}$$

Expressing the linear approximation directly in terms of the optimization design variables, i.e., LP and patch thickness t_p yields equation 2.7 for symmetric laminates ($V^B_{1,2,3,4} = 0$). The additional sum accounts for load redistribution by adding the contribution of the cross-sensitivities to the response.

$$\tilde{R}(V,t) = R_0 + \sum_{p=1}^{N_p} \left[\sum_{i=1}^{4} \left.\frac{\partial R}{\partial V^{A,D}_{i,p}}\right|_0 \cdot \left(V^{A,D}_{i,p} - V^{A,D}_{0i,p} \right) + \frac{\partial R}{\partial t_p} \cdot (t_p - t_{0,p}) \right] \tag{2.7}$$

There is however a disadvantage of using simple linear approximations of the response in order to evaluate the fitness of design points (eq. 2.5). It provides an incentive fora GA to retrieve solutions where the approximation accuracy is diminished non-conservatively by maximizing the error between the design variables that affect the response positively. This is problematic since the approximations no longer hold true and the retrieved designs are in fact poor in performance.

This issue is especially evident when local approximations based on a single design point are used since the validity of the approximation diminishes if the GA moves away from it in any direction in the stiffness space. A solution to this problem is to use linear approximations of multiple reference design points and select the most conservative value of the set of approximations for a given response. Doing so guarantees that non-conservative approximations are discarded as moving away from one reference design point is likely to lead the optimization towards the vicinity of another reference point, improving the accuracy of the approximation and steering the discrete optimization towards feasible solutions.

2.6.2 Kreisselmeier-Steinhauser envelope functions

The selection of the appropriate approximation during the discrete optimization process can be made by using a type of envelope functions known as the Kreisselmeier-Steinhauser (KS) function [37] shown in equation 2.8. Normally, these functions are used to aggregate multiple constraints in order to simplify the optimization. For a number of constraints $g_j(x)$, the function evaluates each one and takes the value of the most critical constraint, neglecting the others. This process can be made more or less conservative depending on which ρ value is used. Lowering the ρ factor translates to a conservative output whereas increasing it pushes the output towards $max(g_j)$, i.e., $lim_{\rho\to\infty} f_{ks} = max(g_j)$ as shown in figure 2.8.

$$f_{ks}\left(g_j(x)\right) = g_{\max}(x) + \frac{1}{\rho}\ln\left[\sum_{j}^{n_g} \mathrm{e}^{\rho(g_j(x)-g_{\max}(x))}\right] \tag{2.8}$$

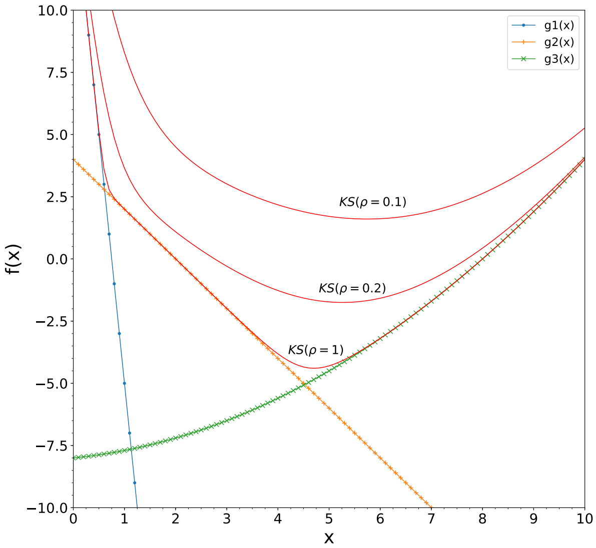


Figure 2.8: g_i constraint aggregation example with three different ρ values.

Alternatively, instead of aggregating constraints this method can be used to aggregate a number of linear approximations based on multiple reference points for a response R_i and return a conservative approximation with respect to a simple linear approximation relying on a single point. For this process, it's necessary to first obtain a number of design points in the vicinity of the continuous optimum including FEM analysis and stored as a reference for subsequent discrete optimization runs. This reference data is then used to build **N** linear approximations for each response and stored in a list from which the KS approximation can be obtained as shown in figure 2.9.

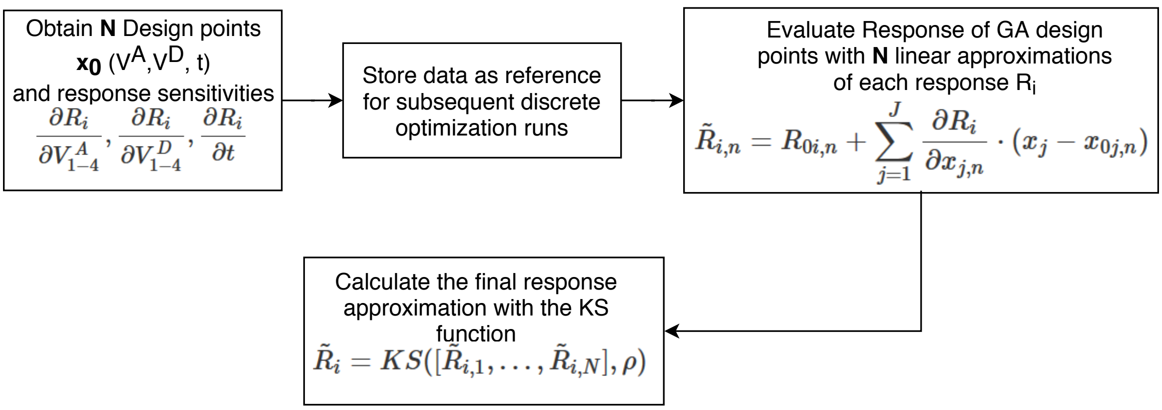


Figure 2.9: Flowchart of the process to obtain the KS of a number of linear approximations.

Having defined key topics in the multi-step optimization of composite structures such as the stacking sequence retrieval challenges and approximation schemes, the goals of this project can be detailed more clearly. A detailed optimization framework flowchart is exemplified in figure 2.10 which factors in the approximations of the response and the objective function provided by equation 2.5.

The focus of this project is on the second discrete step. More specifically, the aim is to retrieve feasible stacking sequences starting from optimal design points previously retrieved in the continuous step that are also solutions to the RMS error problem given by equation 2.4. To do so, mechanical constraints are added as penalty to the RMS objective function to determine the fitness of the design points evaluated by a GA as given by equation 2.5.

These constraints include strength criteria, buckling, displacement and mass as will be shown in later chapters using ONERA's regional wing model as a case study. The mechanical constraints are approximated using the KS method described in section 2.6.2. As blending constraints have been used to retrieve the continuous reference points, it is expected that LP matching fitness functions yield improved results with a lower LP RMS error and further gains can be made by exploring the vicinity of the continuous optimum by considering mechanical constraints in the discrete step. Accordingly, a GA based on SST encoding has been developed and is introduced in the next chapter.

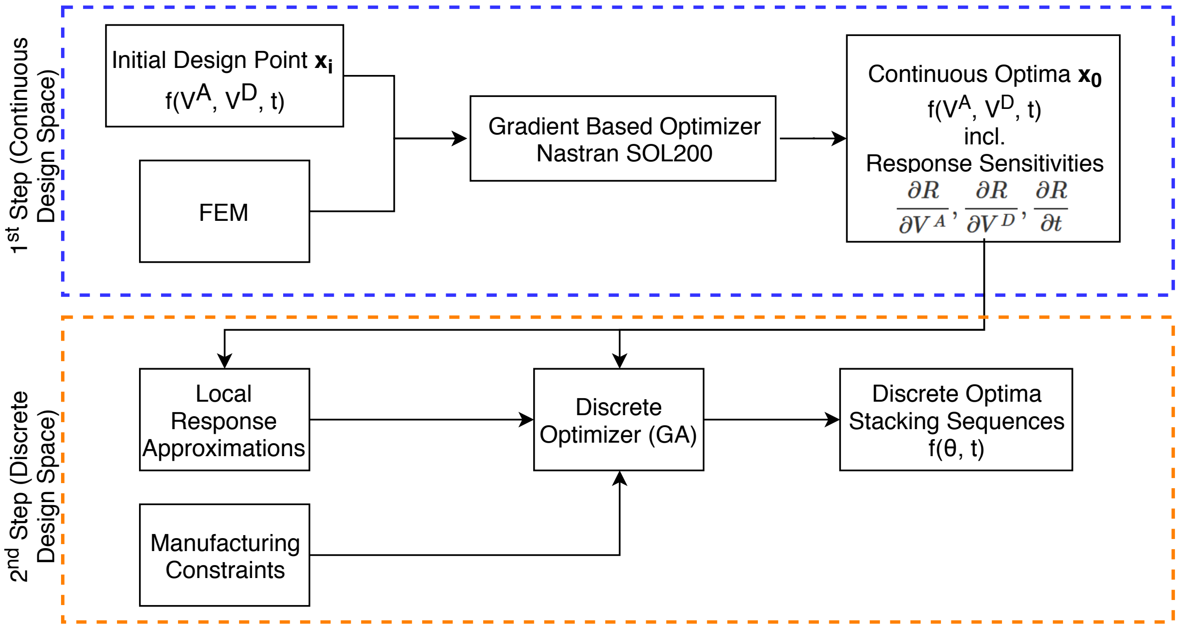


Figure 2.10: Multi-step optimization framework.

Chapter 3

pyTLO: A Python Tapered Laminate Optimizer

During this thesis, a new GA toolbox was designed. Its name is **pyTLO**, which stands for **py**thon-based **T**apered **L**aminate **O**ptimizer. One of its core criteria was to take advantage of the SST encoding and provide a GA with dedicated crossover and mutation operators that use different methods depending on which manufacturing constraints are considered. A general version of **pyTLO** has been made available Open-Source on Gitlab at `https://gitlab.com/FGS_SSR/pytlo`.

An overview of its capabilities and processes is provided in this chapter in section 3.1, which covers its general framework and code structure, followed by a detailed description of pyTLO's workflow in section 3.2. In section 3.3, the different crossover and mutation operations used in pyTLO's GA are described in detail. Finally, the process of verifying the application is covered in section 3.4 where its effectiveness in retrieving known solutions of a set of Multi-Panel Assembly (MPA) with varying number of patches and thicknesses is measured. Additionally, the 18 panel horseshoe benchmark problem is used to compare the tool's performance with literature results using the same SST encoding method.

3.1 pyTLO Framework and Capabilities

The encoding of stacking sequence designs using the SST method provides an efficient means of obtaining fully blended designs for VS laminate structures using a GA. This is a result of its compact form that reduces the number of design variables needed to describe the problem and guaranteed blending. Although there are a number of GA toolboxes available, these are often kept generic to comply with different optimization problems.

In the context of composite optimization subjected to manufacturing constraints, considerable computational costs are incurred when there is no restriction on what constitutes a feasible modification of a stacking sequence design. Hence, the probability of retrieving valid designs from generic GA operators is extremely low, meaning computational resources are wasted on repeated evolution-related operations until a sufficient number of valid outputs is generated.

The development of pyTLO was bound by a core set of requirements to ensure its re-usability in other composite optimization projects. It was mainly devised as a toolbox that could be used to test different GA objective functions in the context of large blended composite structures. For that reason, it was built on Python and libraries which are open-source. Furthermore, the code framework follows an Object-Oriented Programming (OOP) approach which helps to define a modular code structure.

This makes it easy for future users to adapt the application to their needs by modifying existing or adding extra modules as is explained later in this chapter. Overall, it follows a typical GA structure - firstly a population is generated and ranked according to objective criteria, followed by selection of the fittest designs, crossover and mutation for a designated number of generations or until an objective threshold is reached.

The top-level framework of pyTLO is comprised of eight modules shown in bold in figure 3.1. Since code readability is a key property to ensure the code is re-usable by third parties, most functions and methods are self-contained, i.e., exist in separate files to restrict the maximum number of lines per file. In total, the generic[1] pyTLO version contains roughly 80 files.

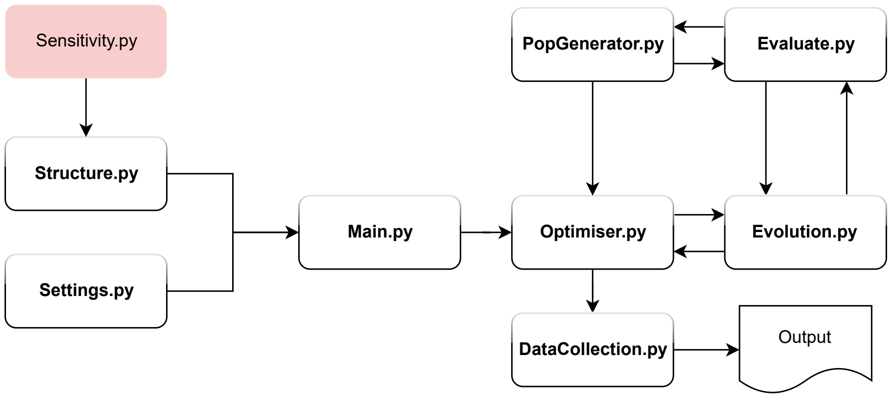


Figure 3.1: pyTLO's main module tree and relationship.

Additionally, since the design of large composite structures is rarely restricted to a single component, the ability to factor in the effects of changing the stiffness of a component onto the entire structure has been made an integral part of pyTLO. Simultaneous optimization of multiple components within a structure is possible, provided that a response approximation model containing cross-component sensitivities is included in the problem setup. This brings advantages in the design of aerospace structures where final shape for a given load case is important, such as in the aeroelastic tailoring of aircraft wings.

The ability to carry out simultaneous optimization is made possible by using an OOP organization for each individual design as shown in figure 3.2 that contains information of the different component SST, as well as fitness, design variables and responses. Take for instance an aircraft wing where it is desirable to consider the lower & upper skins, as well as rear & front spars; the attributes under `individual.Components` would appear 4 times and can be accessed with `individual.components['component_ID'].attribute_name`. In the context of a GA, the population of designs is a list containing N individual objects. The evolution process within the GA is applied independently to each component, A more detailed description of pyTLO and how to access, add or modify attributes within the different classes shown in figure 3.1 is provided in a user manual available in pyTLO's repository.

In addition to the multi-component optimization, pyTLO was tailored to take advantage of the SST encoding discussed in chapter 2. Considerable computational cost reduction was achieved by using dedicated functions in both the initial population initialization and evolution operations when manufacturing constraints are considered. Instead of randomly generating designs and filtering out those that don't comply with criteria, specific operations are carried out with respect to constraints. For instance, if damage tolerance is part of the design criteria the top ply orientation is automatically set to $\pm 45°$; if only balanced laminates are desired, every time a non-$0°$ or $90°$ ply is added the algorithm keeps track of it and

[1] Not containing any project specific modules such as Sensitivity.py.

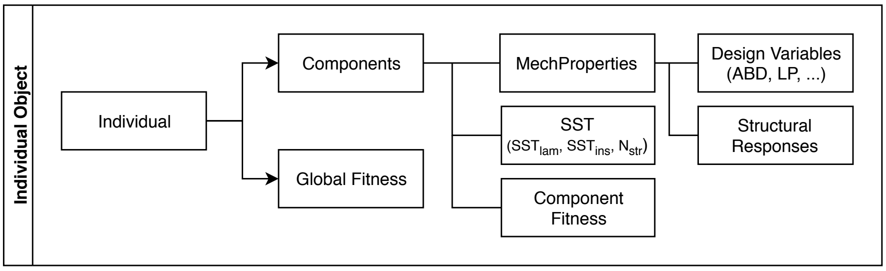


Figure 3.2: Individual object tree.

attempts to add a balancing ply in a randomly selected stacking position in the following operation(s). Other manufacturing constraints listed in section 2.4 are implemented in a similar manner. Randomness in ply angle selection and positioning ensures that the design space is properly explored[2].

3.2 pyTLO's Workflow

The steps needed to initialize pyTLO are described in this section followed by the work-flow sequence of pyTLO's optimization framework. Firstly, the guidelines for setting up the design problem are provided, followed by a work-flow diagram and explanation of the optimization process with pyTLO. Overall, there are three files that should be edited with information pertaining the design problem: `Settings.py`, `Structure.py` and `Objective.py`. The `Settings.py` file is where material properties, GA settings and manufacturing constraints are defined. The complete list of options within this file in the default version of pyTLO is as follows:

- Ply Properties
- Manufacturing Constraints
- GA Settings
- General options - Objective type, constraint weights, etc.
- Data Settings - Options related to results organization such as time-stamp or batch ID.

The `Structure.py` file contains aspects of the design problem such as patch layout, neighboring panels and load distribution. The target optimum data retrieved in LP space such as thickness distribution and lamination parameters is defined in the `objective.py` file. In any of these files, extra attributes can be added by the user on the fly if needed by the objective function guiding the optimization. Once the design problem is defined in these files, pyTLO can be initialized by executing `pytlo.py` in which the total number of runs can be defined. Every time a run is completed, general fitness plots and best individuals are stored in a dictionary that can be accessed by the user once all runs are completed. Otherwise, this information is also stored in a 'results' folder.

The work-flow of the optimization process is displayed in figure 3.3. After initializing the structure and settings files, these are used as input for the main optimizer. At this point, several classes are initialized with information that only needs to be calculated once. For instance, the `CLT_properties.py` calculates all material invariants, reduced stiffness matrix and rotated stiffness matrices of the permissible ply orientations. This information is stored in hash tables and can be accessed at any point during the optimization process.

[2]Within the inherent constraints of the SST method.

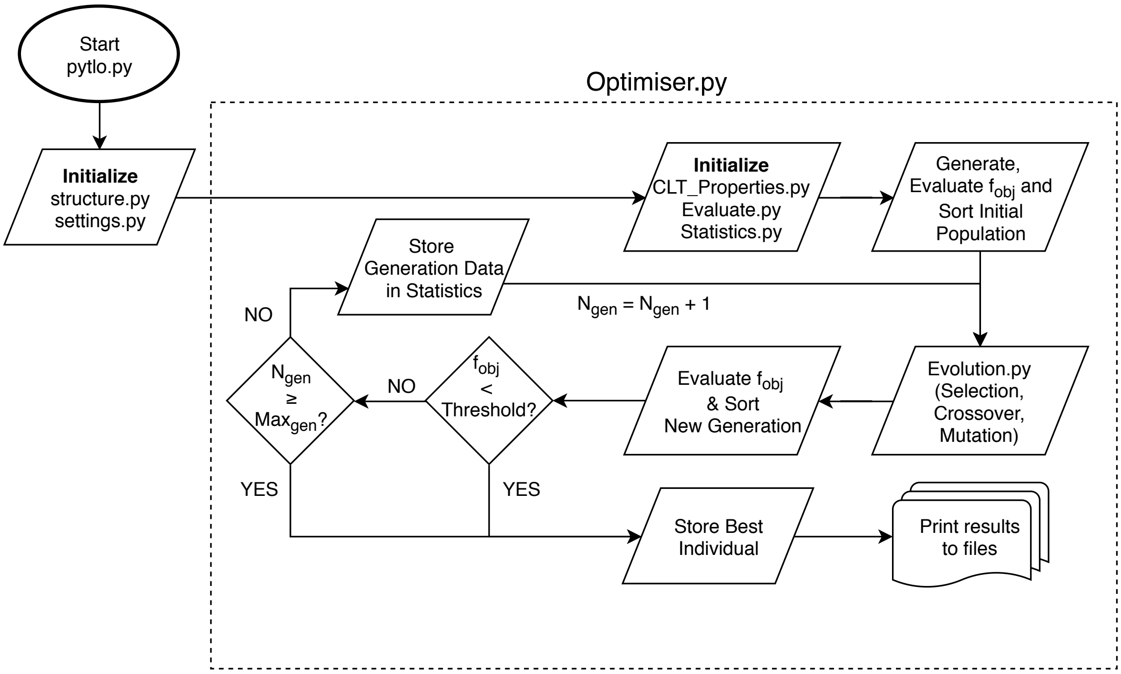


Figure 3.3: pyTLO's simplified workflow diagram

After each new generation of designs has been evaluated, data such as best, worst and mean fitness are stored in the `statistics` class. A hall of fame of the best design at each generation is also assembled. The code terminates once either the maximum number of generations or the fitness threshold has been reached. The run statistics and the best design properties are saved in `.csv` format. In the latter, the following files are generated: `component_name.csv` which includes its SST_{lam}, SST_{ins} and N_{str} as well as the list of manufacturing constraints applied. `LP_error.csv` and `LP_retrieved.csv` contain the LP error for each patch and the retrieved values, respectively. Finally, `patch_properties.csv` contains the stacking sequence of each patch, thickness distribution and total number of plies.

3.3 The Genetic Algorithm

The genetic algorithm implemented in pyTLO was designed with the SST encoding in mind. As a result, its crossover and mutation operations are geared towards handling the SST_{lam}, SST_{ins} and N_{str} lists. The advantage is that the process can be guided by the manufacturing constraints, which in turn improves computational efficiency considerably since the chance of a feasible design being retrieved from these operations is much higher than completely random operations with a direct constraint filter. The algorithm behavior can be controlled to a set of options available in `settings.ga_settings` which is listed in table 3.1.

3.3.1 Population Creator

The population generation builds the SST from the thinnest to the thickest stacking sequence in the structure. Firstly, the guide is built step by step taking into account the manufacturing constraints until it contains N_{min} plies. For instance, if damage tolerance is active, the first ply will have a ± 45 orientation. Subsequent additions consider the allowable orientations and filter them depending on which stack position is being considered.

Table 3.1: GA options in pyTLO

Attribute	Description
`elitism`	Fraction of population kept unchanged through next generation. `elitism` $\in [0, 1]$
`initial_pop`	Size of generated designs from which the best `pop_size` individuals are selected
`pop_size`	Population size per generation
`cx_fraction`	Fraction of population to be used for crossover, ranked according to fitness. `cx_fraction` $\in [0, 1]$
`mut_pb`	Mutation probability for each individual. `mut_pb` $\in [0, 1]$
`max_gen`	Maximum number of iterations for the GA.
`rf_freq`	Refresh population frequency every N generations.
`fit_tol`	Fitness threshold necessary to conclude GA
`update_cycle`	Frequency with which fitness statistics are collected & progress printed on screen.
`fit_weights`	Fitness weight(s). Selection always based on first weight. If negative, minimization objective.
`delete_duplicates`	Delete crossover results that are duplicates of existing designs in the population pool.
`refresh_population`	Refresh non elite population every `rf_freq` generations.

This operation takes into account manufacturing constraints such as disorientation and contiguity by checking the neighboring plies. After filtering out the orientations that wouldn't comply, a random selection with uniform probability of the remaining orientations is made. For balanced designs, a wait list is used to keep track of all θ plies added. If the wait list isn't empty, each new add operation first considers the plies needed to balance the guide before randomly selecting plies from the list of all possible options.

Once the guide is assembled, the SST genotype described by the SST_{lam}, SST_{ins} and N_{str} vectors is created in similar fashion by ply-add operations from the thinnest to the thickest stacking sequence in the laminate. The SST_{lam} is built by extending the initial guide stacking sequence with each new ply added according to manufacturing constraints. In this step, the probability of a given position in the stacking sequence to be chosen for a ply add is inversely proportional to the distance to the position of the previous add operation. This is done to ensure that dropped plies are well distributed through-thickness.

Considering now the symmetric example given in figure 3.4 where a ply is added to the guide laminate in position 4 and the possible angles $\in [-75°, 90]$ in steps of 15°. With a maximum disorientation between plies of ±45°, the resulting possible orientations are $[0, -15, -30, -45, -60]$ from which the ply orientation is randomly selected.

If balance does not need to be restored the SST_{ins} is updated with the rank of insertion of the new ply. Note here that the rank of insertion of a given ply in the SST_{ins} vector represents the minimum laminate thickness at which a given ply is present and the zero values denote the guide plies. The advantage of this change in the SST_{ins} encoding is that modifications to the SST resulting in the removal of columns do not affect SST_{ins} and no additional repair operation is necessary. Furthermore, the decoding of the stacking sequences with respect to both SST_{ins} and N_{str} is more straightforward.

The balancing of the SST is done by an extra operation which is triggered whenever an orientation not in $[0°, 90°]$ is added. In this case, two plies or four if symmetry is enforced, are added in a single operation. At the end of each add operation, the SST_{ins} vector is

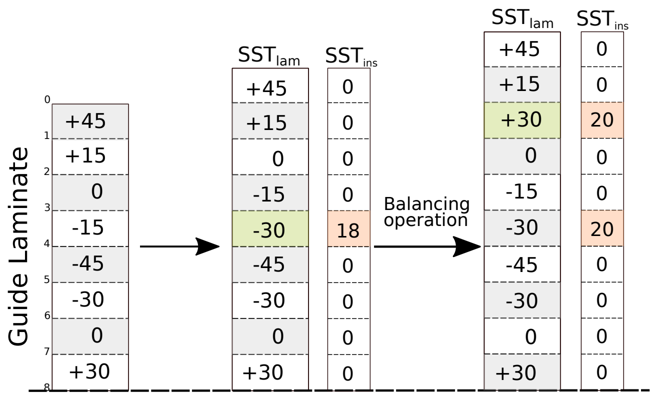


Figure 3.4: Assembly of a Stacking Sequence Table

modified with the rank of insertion of the added plies as shown in figure 3.4. Since both plies were added in the same add operation, the SST_{ins} entry is the same for both plies.

Finally, the N_{str} list is built according to predefined thickness distributions[3] and maximum allowed ply-drop (Δn) between adjacent panels. For this step, a connectivity matrix detailing the neighbors of a patch is needed. For each patch, the list of possible thicknesses derived from the SST_{ins} is filtered according to the target thickness range of the patch and the maximum Δn between adjacent patches. The N_{str} entry is randomly selected from the resulting list. There are two cases where the filtered list is empty and the resulting N_{str} can differ from the predefined thickness distribution:

1. The SST does not contain a stacking sequence thickness entry that falls within the thickness range of a given patch. This can be the case with balanced laminates since some stacking sequences are skipped for not being balanced.

2. Due to the Δn constraints. Neighboring patches can restrict the permissible patch thickness outside the prescribed range.

In both cases, the selected N_{str} entry will be the nearest possible integer.

3.3.2 Selection

In the context of GA, selection refers to the process with which the population of designs is filtered for different purposes. Generally, selection is influenced by the fitness attributed to an individual design. In pyTLO, three different kinds of selection are available: by best fitness, tournament and random selection.

Best fitness selection simply takes the population pool as input and sorts it by best fitness in descending order, returning N individuals, where $0 < N \leq populationsize$. Tournament selection takes the population pool and tournament size as inputs; a random selection of P individuals is made, where $P =$ tournament size. From this set of individuals, the two with the best fitness are selected for crossover. The probability of lower fitness designs taking part in the crossover operations depends on P, where larger values lead to a lower chance poor fitness designs participate. Random selection is, as the name implies, a random selection of N individuals from a population pool independent of fitness values.

[3] Each patch is assigned a $[N_{max}, N_{min}]$ thickness range

3.3.3 Crossover

The crossover operation is applied once every generation. Firstly a "crossover population" of size `pop_size` · `cx_fraction` is built using the best fitness selection method. Then two designs are selected through a tournament selection followed by the crossover of either the SST_{lam} or N_{str} which is randomly selected. If the operation is successful according to design constraints, the new designs (offspring) are added to the new population as shown in figure 3.5. This process is repeated until the non-elite portion of the population is re-filled. Once the population size equals `pop_size` the GA progresses to the mutation phase.

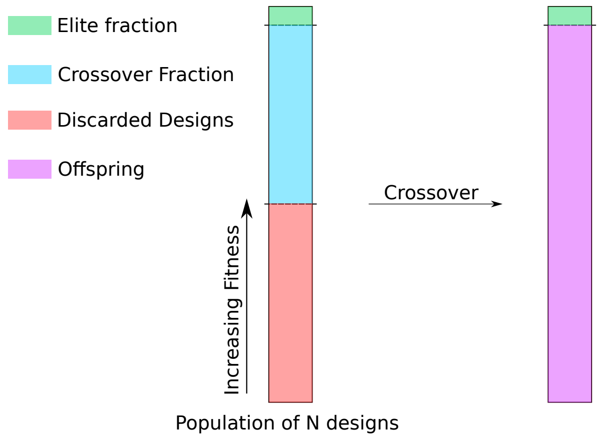


Figure 3.5: Population crossover process

The go-to crossover operation applied by pyTLO for general laminates is a simple two-point crossover. This operation is only applied to the SST_{lam} or N_{str} lists, with the latter being skipped in case thickness distribution is fixed. An example of a two-point crossover is shown in figure 3.6 on an N_{str} list; two slicing points, $CX1$ and $CX2$ are randomly selected, followed by an exchange of the slice between the parents. In all cases, the result of a crossover operation is checked against manufacturing constraints and if it's deemed unfeasible, new parent designs are selected and a new attempt is made.

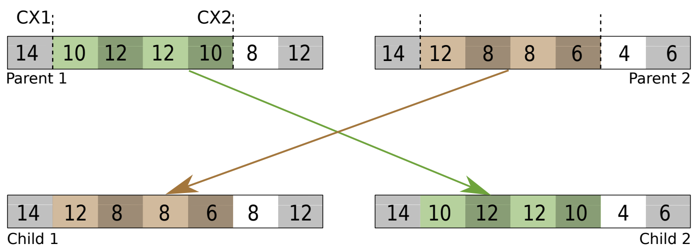


Figure 3.6: Crossover operation of the N_{str} vector for non-balanced laminates.

Balanced laminates

In order to increase the success rate of the crossover operations for balanced laminates, a dedicated operation is applied which is shown in detail in figure 3.7. After parent selection, a ply-pair list and their positions is built for each SST_{lam}. Finally, a ply-pair position is randomly selected and exchanged between the two designs if deemed possible under predefined manufacturing constraints.

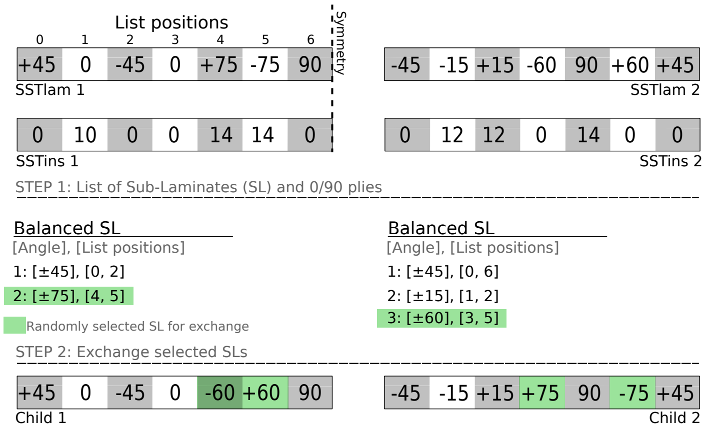


Figure 3.7: Crossover operation for balanced laminates.

Given that balanced stacking sequence tables are likely to have different possible thickness distributions, the crossover of the N_{str} list can lead to cases where a stacking sequence does not exist for all thickness distributions. In such cases, the resulting N_{str} is repaired by rounding-off the non-existing thickness distribution to the nearest available integer in the SST as shown in figure 3.9.

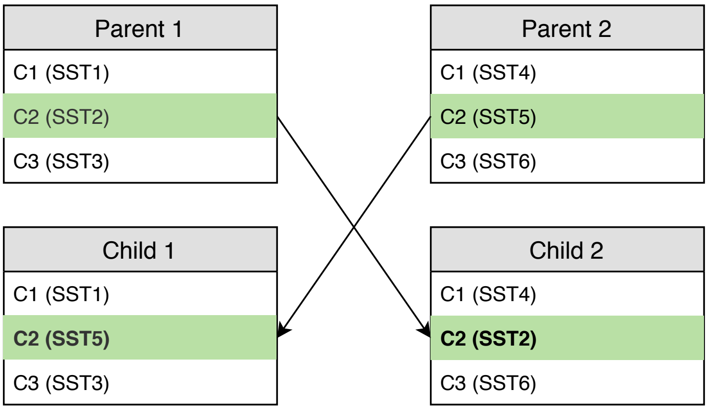


Figure 3.8: Component crossover operation.

Finally, if the optimization is carried out for structures containing at least two components, a crossover operation to exchange component designs between different individuals is applied. In this case, the selected component for crossover exchanges its entire SST as shown in figure 3.8.

3.3.4 Mutation

When all crossover operations are completed and the new population is assembled, a mutation cycle is applied. Mutations introduce variations in the designs which lower the risk of the GA becoming trapped in local optima and thus improving the exploration of the design space.

The probability that a non-elite individual of the population undergoes mutation is defined by the `settings.ga_settings.mut_pb` attribute. In total, four different mutations are possible: Change of orientation of N plies in the SST_{lam} vector, permutation of the ply rank of insertion values in the SST_{ins}, change in ply stacking sequence position and change of patch thickness in the N_{str} vector. As with the crossover operations, different methods are used

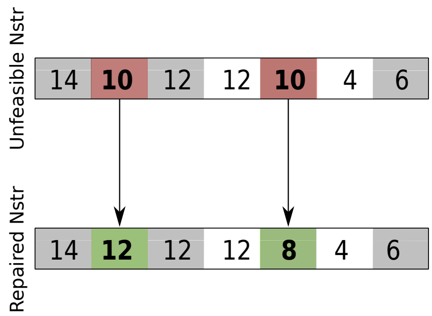


Figure 3.9: Repair operation of the N_{str} list for balanced laminates.

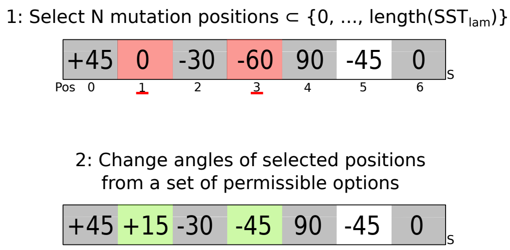


Figure 3.10: Mutation of the SST_{lam}

depending on the balance constraint. Only one mutation is applied for a selected individual and it is randomly selected from the list of available mutations.

$\mathbf{SST_{lam}}$ Mutation

The SST_{lam} mutation of non-balanced designs involves changing ply orientations of N randomly selected stacking positions as shown in figure 3.10. N can be set either as an integer number or as a function of the patch thickness to control how large the mutation is with respect to the number of plies in the laminate; mutating a single ply angle on a 200-ply laminate has a much smaller effect on the design when compared to a single ply mutation in a 10-ply laminate.

$\mathbf{SST_{ins}}$ Mutation

Two different mutations of the SST_{ins} vector are possible. One applies a permutation to the non-zero entries of the vector which correspond to the layers that are not part of the guide laminate. This translates to a change in which layers are dropped at a given laminate thickness as shown in figure 3.11. The goal of the second mutation is to change the stacking sequence of the laminate and its effect over the design depends on the selected layer's rank of insertion, i.e., changing the position of a layer of the guide laminate has a larger effect than if the selected layer exists only in thicker laminates. This mutation impacts both the SST_{ins} and SST_{lam} vectors as shown in figure 3.12. The two layers exchange position, but their rank of insertion remains the same. In the example provided, if the disorientation guideline is active, the resulting mutation would be discarded due to the adjacent $\pm 45°$ laminates and a new mutation attempt is carried out.

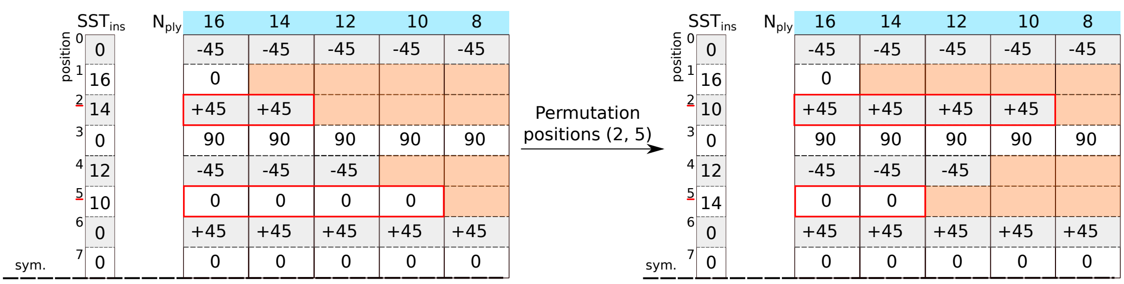


Figure 3.11: SST_{ins} permutation mutation

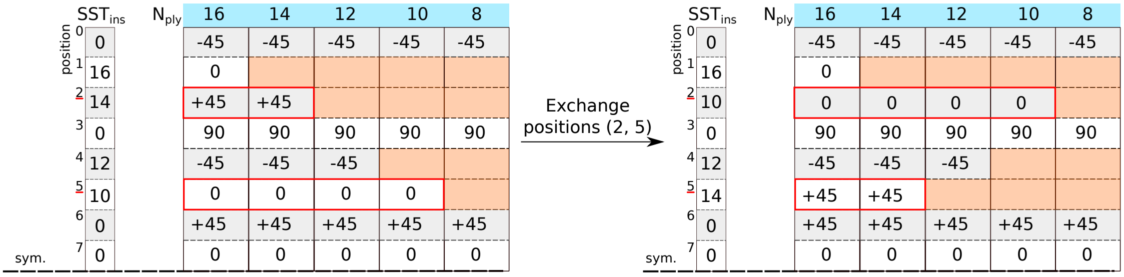


Figure 3.12: SST_{ins} stacking sequence mutation

N_{str} Mutation

The mutation of the N_{str} vector simply changes the thickness of a randomly selected patch from a list of possible options. These options are obtained from the list of available thicknesses in the stacking sequence table and filtered according to the maximum allowed ply drop Δn between neighboring panels.

Balanced laminates

When balanced laminates are considered, a different set of mutation algorithms is used to ensure that balance is maintained throughout the retrieval process. Of the four possible mutations, the stacking sequence and N_{str} mutations are not affected by the balance guideline and no separate operation is needed. For the SST_{lam} mutation, the applied approach is similar to the crossover of balanced designs shown in figure 3.7 where the ply-pairs are listed followed by changing the orientation of a randomly selected pair for the list of allowed orientations. After each operation, the result is checked against the manufacturing constraints and if the outcome is invalid a new attempt is made. For each individual that's been selected for mutation, up to 30 attempts are made for a valid mutation. If no feasible mutation is retrieved, the algorithm skips to the next individual.

The permutation of ply ranks of insertion of the SST_{ins} vector also operates in terms of ply pairs where balanced designs are considered. Hence, the algorithm starts by assembling a list of ply pairs, followed by an exchange in the SST_{ins} entries of two randomly selected pairs. Additionally, if the laminate contains more than one 0 or 90° ply there's a 50% chance of making a permutation of the rank of insertion of three plies; the initially selected ply pair in addition to a 0 or 90°.

3.4 Algorithm Verification

The effectiveness of the GA provided in pyTLO was measured by testing its ability to retrieve known stacking sequence tables. For this purpose, a total of 18 different multi-panel problems are considered which include variation in number of patches as well as maximum number of plies N_{max}.

Finally, an optimization test is made using the 18 panel horseshoe problem in order to validate the algorithm by showing that the lightest solutions retrieved are in agreement with published results in the literature using the same blending strategy. This optimization is carried out twice, one with symmetry as the only manufacturing constraint and another with all constraints listed in section 2.4.

3.4.1 Performance Verification With a Set of Multi-Panel Problems

The 18 multi-panel problems considered have 2, 4, 8, 16, 32 and 64 patches. For each of these cases, three different N_{max}= $[20, 40, 80]$ are applied with $N_{min} = 14$ for all cases. The GA settings are given in table 3.2. Where manufacturing constraints are concerned, ply orientations in steps of 15° are used and symmetry is enforced. Given that the ply-drop rule Δn is not enforced and neither loads nor responses are relevant to this verification process, the layout and dimension of the patches are not relevant, only their thickness distribution for the purpose of calculating the LP.

The genotype (SST_{lam}, SST_{ins} and N_{str}) of the target design points were randomly retrieved using pyTLO's population creator and their lamination parameters used as a target in the verification runs. This way it is ensured that the targets are part of the discrete feasible design space and can be retrieved by the GA. The experiment is set up to run 50 times per case with a simple LP-matching objective function given by equations 3.1 and 3.2.

$$f_{obj} = min(LP_{rmse}) \tag{3.1}$$

$$LP_{rmse} = \frac{1}{P}\sum_{p=1}^{P}\sqrt{mean\left((LP_{GA} - LP_{obj})^2\right)} \tag{3.2}$$

Where P is the total number of patches. The ply material properties used are given in table 3.3.

Table 3.2: GA settings used in the verification runs

attribute	**Value**
`elitism`	0.04
`initial_pop`	100
`pop_size`	100
`cx_fraction`	0.75
`mut_pb`	0.15
`max_gen`	500
`rf_freq`	-
`fit_tol`	0.001
`update_cycle`	10
`fit_weights`	-1
`delete_duplicates`	False
`refresh_population`	False

Table 3.3: Ply material properties

Property	**Value**
E_1 $[GPa]$	177
E_2 $[GPa]$	10.8
G_{12} $[GPa]$	7.6
μ_{12} $[-]$	0.27
$Thickness$ $[mm]$	0.20

The effectiveness of the algorithm is determined by how well the design space is explored, i.e., that design points similar to the target are reliably retrieved and how long a run of 500 generations lasts. The results are collected in figures 3.14-3.15. The time taken per run displayed in figure 3.13 is obtained by calculating the mean of the time taken for all runs of each MPA.

It can be seen that the number of patches does not have a large impact on the run time when the maximum $N_{ply} = [20, 40]$, whereas the increase in maximum N_{ply} leads to a clear increase which is exponential. This is expected as a consequence of the encoding used in the SST; the size of the three chromosomes SST_{lam}, SST_{ins} and N_{str} vary differently depending on which parameter is modified. The length of both the SST_{lam} and SST_{ins} depends only on $N_{ply,max}$, whereas the number of patches affects only N_{str}.

The increase in run-time associated with the number of patches is mainly related to the number of different stacking sequences that need to be evaluated for each design point in the GA. Given that only symmetric laminates are being considered, the maximum number of different stacking sequences resulting from dropped plies is equal to $1 + \frac{N_{max}-N_{min}}{2}$. Since $N_{max} = [20, 40, 80]$ and $N_{min} = 14$, this means that for N_{max} the maximum number of stacking sequences that need to be evaluated is equal to 4, 14 and 34 respectively.

This also explains why after a certain number of patches, the computational time is no longer affected by further increases in patch number as a consequence of the number of different stacking sequences reaching its peak when $N_{patches} \geq N_{laminates}$. If the number of patches had been further increased from 64 for the $N_{max} = 80$ case, the (blue) line of the run-time would be nearly vertical starting from $N_{patches} = 32$ similar to the other two cases. On average, evaluation of the LP of the population accounts for $20 - 25\%$ of the total run-time.

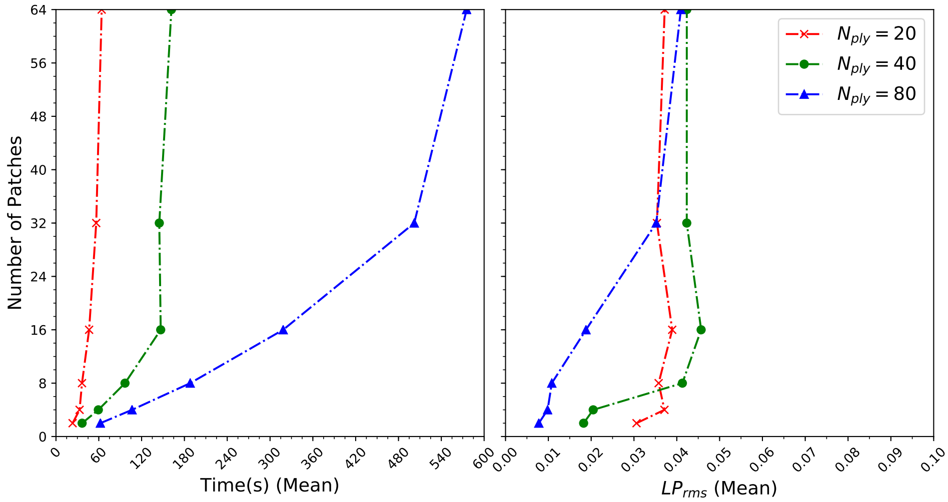
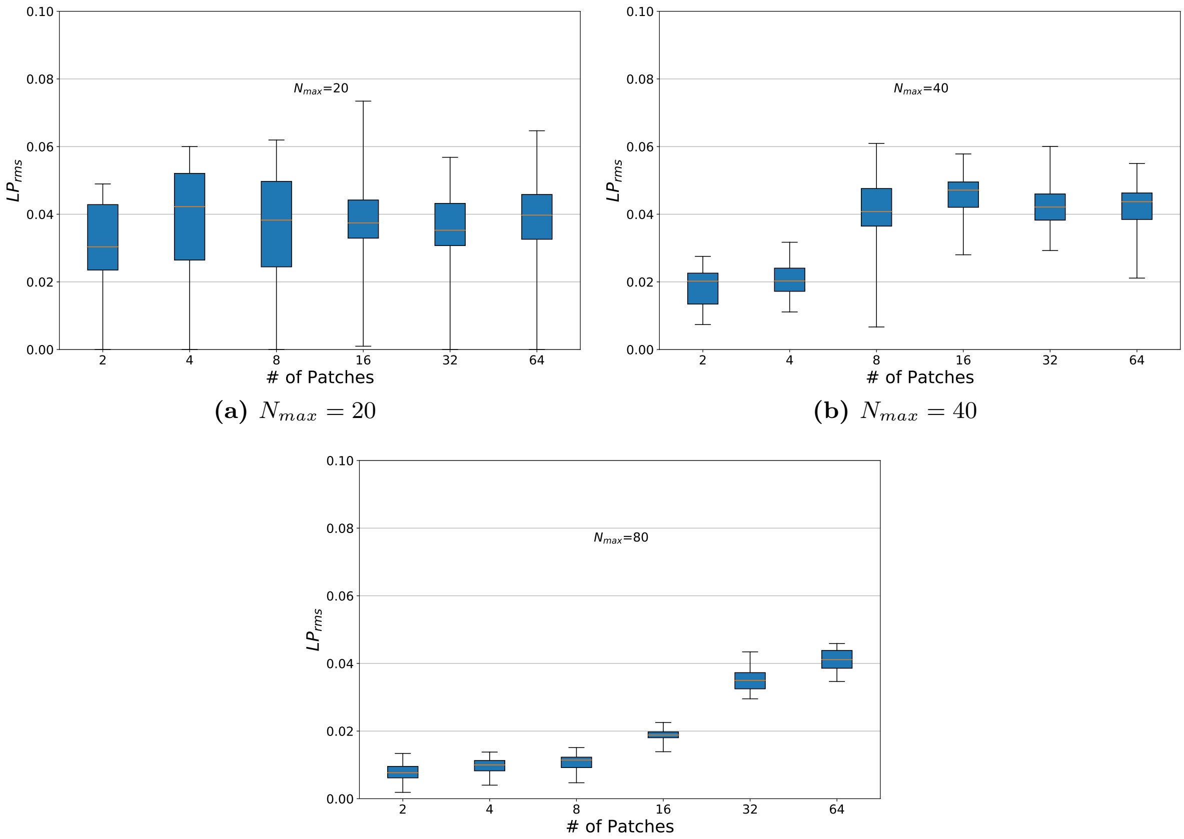


Figure 3.13: Mean time taken per run (left) and Mean of best LP_{rmse} (right)

The mean of LP_{rmse} of the best design point retrieved for each run in 3.13 shows a similar behavior, namely that once the peak for the maximum number of different laminates is reached, the retrieved LP_{rmse} does not vary with increasing patch numbers. Furthermore, it can be seen that runs with thicker stacking sequences return lower average LP_{rmse} for cases with 2 or 4 patches. This is a combination of thicker laminates providing a less constrained design space since more stacking combinations are available and the number of different laminates in the SST is small enough that 500 generations are sufficient to explore the design space. For the MPA cases where $N_{max} = 20$, it can be seen in 3.14 a) that the algorithm was able to retrieve the exact solution for most MPA which was no longer the case for $N_{max} = 40$ and $N_{max} = 80$ due to the increased dimensionality of the problem and relatively low number of generations per run.

Overall, the algorithm consistently retrieves design points with a relatively low LP_{rmse} when compared to the average error and rapidly converges at the start of the optimization as seen in figure 3.15.



(a) $N_{max} = 20$

(b) $N_{max} = 40$

(c) $N_{max} = 80$

Figure 3.14: LP_{rmse} of the best design point retrieved for 50 runs per case.

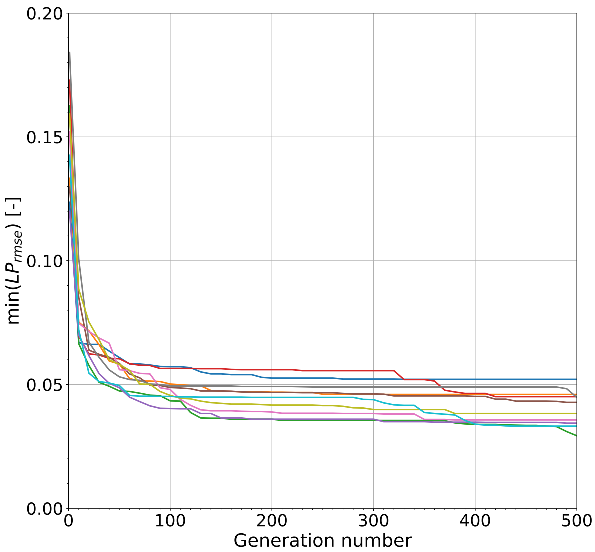


Figure 3.15: Best fitness progression of 10 randomly selected runs with 32 patches, $N_{ply} = 40$

3.4.2 Verification with the 18-panel Horseshoe Problem

Finally, the algorithm is used to optimize the horseshoe problem proposed by Soremekun et al. [24] shown in figure 3.16 which is comprised of 18 patches subjected to bi-axial compression loads. This benchmark was created with the purpose of measuring the effectiveness of different blending strategies in the context of weight optimization subjected to buckling constraints. The goal is then to retrieve the lightest solution that is fully blended and complies with buckling constraints as summarized by equation 3.3.

$$f_{obj} = min(\rho \sum_{i=1}^{P} A_i \cdot t_i) \quad st. \ \ \lambda_i \geq 1 \tag{3.3}$$

Where P is the total number of patches, A_i and t_i are the area and thickness of the i^{th} patch, respectively. The buckling reserve factor of each patch λ_i is obtained using equation 3.4.

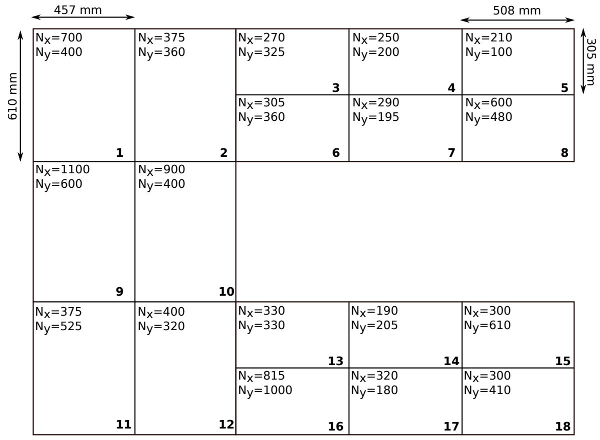


Figure 3.16: The 18-panel horseshoe problem incl. loading distribution in lb/in^2

$$\lambda_i = \pi^2 \frac{D_{11,i}(m/a_i)^4 + 2\left(D_{12,i} + 2D_{66,i}\right)(m/a_i)^2(n/b_i)^2 + D_{22,i}(n/b_i)^4}{(m/a_i)^2 N_{x,i} + (n/b_i)^2 N_{y,i}} \tag{3.4}$$

This benchmark problem is critical in buckling mode $m, n = 1$ so no other modes were considered during the optimization but the final retrieved results were verified against all buckling mode combinations up to $m, n = 3$. The benchmark relies on two assumptions: 1) The patches are assumed to be under a simply supported boundary condition and 2) The applied loads remain constant with load redistribution due to stiffness changes not being considered in the optimization. The material properties used in the optimization runs represent Graphite/Epoxy IM7/8552 laminates with $E_1 = 141$ GPa, $E_2 = 9.03$ GPa, $G_{12} = 4.27$ GPa, $u_{12} = 0.32$ and $t_{ply} = 0.191$ mm. The permissible ply orientations are given in steps of 15 degrees.

The lightest solution obtained in ten runs with only symmetry enforced weighs 28.81 Kg and each run took 5 minutes to complete, whereas the solution obtained by Irrisarri et al. [2] using SST blending weighs 28.55 Kg with a run time of 40 minutes. This difference can be explained by the fact that pyTLO laminates are always even-numbered, whereas the literature example allows odd-numbered symmetric laminates. If the same decoding is applied in both cases, the resulting weight is identical. With all manufacturing constraints enforced except for maximum Δn, the lightest solution obtained weighs 29.46 Kg which is slightly above literature results of 29.29 Kg [38]. The run-time with all manufacturing constraints increased to 12 minutes per run due to the additional operations and checks required to ensure feasible laminates.

Chapter 4

Discrete Optimization Subjected to Mechanical Constraints

In this chapter, the methodology used in this report to retrieve stacking sequences of multi-panel blended laminates is described in detail. The core of the strategy is to subject LP RMS error objective functions to mechanical constraints such as buckling in order to search the vicinity of the target continuous optimum for feasible designs. A continuous optimization of the 18-panel horseshoe benchmark that employs blending constraints [7] is used as a target for the discrete optimization discussed in this chapter.

In section 4.1 the two main objective functions used with pyTLO are introduced together with their limitations, followed by a comparison of the functions and their effectiveness in retrieving solutions for the horseshoe benchmark problem in section 4.2 where the methodology, results and conclusions are covered.

4.1 LP matching fitness functions

In this work, focus is given to a class of fitness functions known as LP matching, which act by guiding the discrete optimization step towards stacking sequences that best match the parameters obtained in the continuous step. These objectives are known to result in poor solutions due to the mismatch between the continuous and discrete design spaces, but with the application of blending constraints in the continuous design space, it is possible to retrieve solutions with lower lamination parameter error [26].

4.1.1 Lamination Parameter Matching Objective Functions

Lamination parameter matching objective functions are based on a RMS error minimization problem where the objective is to achieve the lowest RMS error between the target LP of the continuous step and the retrieved stacking sequences. The lower the error, the more the retrieved stacking sequences match the stiffness distribution of the continuous optimum. As the design space of the discrete space is more constrained, the lowest achievable error between LP is often still too large which causes the mechanical properties of the retrieved stacking sequences to be far too different from those derived from the continuous optimization step [27].

A common LP matching objective function used in the retrieval of tapered laminate stacking sequences is given by equation 4.1 where N_p is the number of patches in the laminate, J is the number of LP being considered per patch and x represents the design variables, which in the discrete optimization step are the stacking sequence orientations $[\theta_1, ..., \theta_n]$ and thickness of each patch for the i^{th} design being evaluated by the GA.

$$LP_{rmse} = min\left(\frac{1}{N_p}\sum_{n=1}^{N_p}\sum_{j=1}^{J}\sqrt{\frac{(LP_{j,GA}(x_i) - LP_{j,opt})^2}{J}}\right) \tag{4.1}$$

Since the objective function is an average of the error contribution from all panels and LP, all these variables are considered to have the same weight in the optimization, which is not necessarily the case. This can be clearly demonstrated with figure 4.1 assuming a single patch buckling problem under a simply-supported assumption.

This implies that the only relevant lamination parameters are V_1^D and V_3^D. After retrieving a continuous design optimum in terms of LP, a stacking sequence retrieval is applied and due to additional constraints in the discrete optimization a residual error remains in all retrieved solutions "+". Although they all portray an equal fitness according to equation 4.1 since they lie on the dashed circle centered on the target design point, the difference in the LP can lead to some of these designs having a lower than required buckling reserve factor.

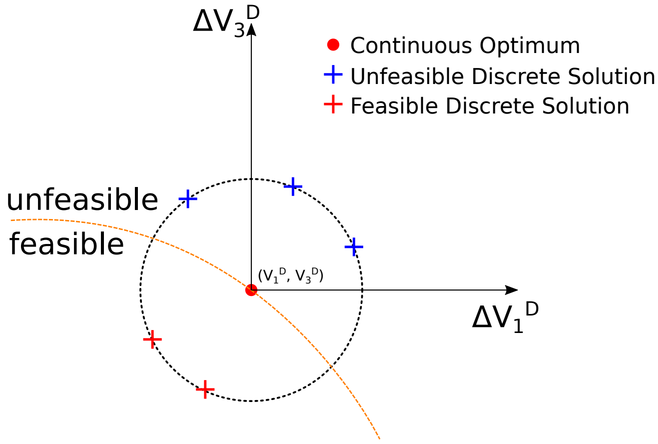


Figure 4.1: Example of a number of discrete solutions with the same LP_{rmse} but different stiffness properties.

This forms the basis of enhancing the discrete optimization step with mechanical constraints in this project. As gradients of the different responses can be obtained from FEA, this information can be used to approximate the response in the evaluation of the designs by the GA and add a penalty to solutions which don't comply with constraints. In effect, using the gradients to weight the different design variables indirectly by their impact on the mechanical responses.

4.1.2 Lamination Parameter Matching Subjected to Response Constraints

The mechanical responses of the design points can be added to the objective function 4.1 as indirect constraints, which ensures that potential solutions are not prematurely removed from the population pool without first going through the evolutionary processes. The resulting function is given by equation 4.2 where $\tilde{R}_k(x)$ is an approximated mechanical response j of a GA individual and R^t is a threshold that determines whether the constraint is active or not. The penalty is only added when $\tilde{R}_k(x) - R^t > 0$ as defined by the unit step function δ.

$$f(x)_{GA} = min\left(\frac{1}{N_p}\sum_{n=1}^{N_p}\sum_{j=1}^{J}\sqrt{\frac{(LP_{j,GA}(x_i) - LP_{j,opt})^2}{J}} + \sum_{k=1}^{K}\delta(\tilde{R}_k(x) - R^t)\right) \tag{4.2}$$

$$\begin{cases}\delta = 1 & \tilde{R}_k(x) > R^t \\ \delta = 0 & \tilde{R}_k(x) \leq R^t\end{cases}$$

The equation above is only valid for cases where the thickness distribution, and hence the mass, are fixed throughout the optimization. Otherwise, it becomes unbounded since increasing patch thickness translates to a lower LP_{rms} error as a larger number of plies relaxes the design space and encourages a mass increase for improved fitness. Additionally, it's effectiveness is directly tied to the approximation scheme. If the approximation or surrogate carries a large error over the actual FEM response it can cause retrieved results to be assumed feasible by the GA if the approximation is non-conservative when in fact a number of failed elements are present. The opposite happens with over-conservative approximations, leading to a compromised structural efficiency.

4.2 Horseshoe Benchmark Test

The 18 panel horseshoe benchmark is again used here to compare the two objective functions given by equations 4.1 and 4.2. The assumptions remain the same, i.e., the loads are fixed and each panel is subjected to a simply supported boundary condition. These assumptions simplify the problem considerably since the panels are effectively decoupled from one another except for the blending constraint.

Additionally, with the simply supported and symmetric laminates assumptions, only two lamination parameters are relevant in the optimization: V_1^D and V_3^D. The aim of this exercise is to gauge the improvement on the retrieved stacking sequences with respect to the reserve buckling factor of the different patches without allowing for a mass increase over a reference continuous optimum obtained from an article by Macquart et al. [7] which employs blending constraints in the continuous optimization.

The buckling reserve factor is calculated for each patch using equation 3.4 which is repeated again here for convenience. The threshold for buckling failure is equal to 1, hence a design is only feasible if all patches have a buckling reserve factor $\lambda_i \geq 1$.

$$\lambda_i = \pi^2 \frac{D_{11,i}(m/a_i)^4 + 2\left(D_{12,i} + 2D_{66,i}\right)(m/a_i)^2(n/b_i)^2 + D_{22,i}(n/b_i)^4}{(m/a_i)^2 N_{x,i} + (n/b_i)^2 N_{y,i}}$$

The derivatives of the buckling response λ_i in terms of the V_1^D and V_3^D parameters can be obtained from equation 3.4 by replacing the $[D]$ stiffness terms with the LP equivalent in terms of material invariants as given by equation 2.2. Since thickness is taken as a constant and the loads are fixed in this exercise, the resulting derivatives are a constant given by the expressions 4.3 and 4.4.

$$\frac{\mathrm{d}\lambda_i}{\mathrm{d}V_1^D} = \frac{h^3\pi^2 U_2}{12\left[N_{x,i}\left(\frac{m}{a_i}\right)^2 + N_{y,i}\left(\frac{n}{b_i}\right)^2\right]} \cdot \left[\left(\frac{m}{a_i}\right)^4 - \left(\frac{n}{b_i}\right)^4\right] \tag{4.3}$$

$$\frac{\mathrm{d}\lambda_i}{\mathrm{d}V_3^D} = \frac{h^3\pi^2 U_3}{12\left[N_{x,i}\left(\frac{m}{a_i}\right)^2 + N_{y,i}\left(\frac{n}{b_i}\right)^2\right]} \cdot \left[\left(\frac{m}{a_i}\right)^4 + \left(\frac{n}{b_i}\right)^4 - \left(\frac{m}{a_i}\right)^2 \cdot \left(\frac{n}{b_i}\right)^2\right] \tag{4.4}$$

The buckling reserve factor can then be calculated solely in terms of LP and the sensitivities through a linear approximation given by equation 4.5.

$$\tilde{\lambda}_i = \lambda_{0i} + \left.\frac{\mathrm{d}\lambda_i}{\mathrm{d}V_1^D}\right|_0 \cdot \Delta V_1^D + \left.\frac{\mathrm{d}\lambda_i}{\mathrm{d}V_3^D}\right|_0 \cdot \Delta V_3^D \tag{4.5}$$

Where $\Delta V_n^D = V_{n,i}^D - V_{n,0i}^D$, the difference between retrieved and target LP. Since the derivatives are constant, the approximation is exact.

4.2.1 Optimization Methodology

In order to measure the performance of the different fitness functions, pyTLO is set up with the horseshoe problem including the sensitivities which are calculated a priori. A total of three different runs are executed:

1. LP_{rmse}: Simple LP matching with respect to a reference continuous optimum (eq. 4.1).
2. $LP_{rmse} + \lambda_c$: LP matching subjected to buckling penalty whenever $\lambda_i < 1$ as given by equation 4.2.
3. λ_c: Fitness is measured only by the penalty of active constraints, i.e., only the second half of equation 4.2.

Twenty GA runs were performed for each case. The settings of the runs are the same as given in table 3.2 and the material reference is Graphite/Epoxy IM7/8552. The only manufacturing constraint being considered is symmetry which is in line with the reference case.

4.2.2 Results

The achieved buckling reserve factors are provided in table 4.1 for the best results obtained, selected according to the lowest LP_{rmse} of each batch. The first column pertains to the continuous design optimum and thus all panels are above the failure threshold. This is not the case for the retrieved designs in the discrete design space where the algorithm fails to retrieve designs with less than 8 failed panels (marked in bold) or 12 in the case of the reference discrete optimum.

It is in figure 4.2 that the effects of subjecting the LP matching objective to constraint penalties become clear. The top diagram shows the number of failed panels per run for each case and the dashed red line represents the reference discrete result. Although all methods were able to retrieve designs with a minimum of 8 failed panels only those with constraint penalties were able to do so consistently across all runs. It can be noted that for all the cases the retrieved results were better than those of the reference where the best results had 12 failed panels. This is a consequence of the SST encoding used in pyTLO, which follows a generalized blending definition and thus is less restrictive on the design space than the method used in the reference paper which follows an inner blending definition.

The results indicate that adding constraints to guide the GA towards the feasible regions of the design space (see figure 4.1) in the vicinity of the continuous optimum leads to an improvement in the mechanical properties of the retrieved solutions. The impact on the discrete optimization is expected to be larger where the load redistribution resulting from stiffness changes in tapered laminates is considered since each set of LP pertaining to a given patch will also be weighted according to its effect on surrounding patches due to cross-sensitivities, i.e., the effects of its design variables on the response of surrounding patches.

Table 4.1: Reserve Buckling Factors(λ) for each objective, best of 20 runs.

Patch No.	Reference Results [7]		Discrete f_{obj}		
	Continuous Optimum	Discrete Optimum	LP_{rmse}	$LP_{rmse} + \lambda_c$	λ_c
1	1.006	**0.884**	**0.917**	**0.925**	**0.941**
2	1.024	**0.930**	**0.937**	**0.951**	**0.975**
3	1.002	**0.940**	**0.985**	**0.985**	**0.981**
4	1.039	**0.985**	1.039	1.047	1.042
5	1.156	1.138	1.206	1.212	1.208
6	1.133	1.095	1.136	1.144	1.133
7	1.006	**0.954**	1.006	1.014	1.009
8	1.035	**0.941**	**0.970**	**0.975**	**0.968**
9	1.044	1.007	1.034	1.040	1.034
10	1.005	**0.891**	**0.922**	**0.929**	**0.942**
11	1.011	1.010	1.017	1.029	1.051
12	1.019	**0.926**	**0.932**	**0.947**	**0.971**
13	1.186	1.146	1.189	1.197	1.186
14	1.102	1.045	1.102	1.110	1.105
15	1.003	**0.912**	**0.941**	**0.946**	**0.939**
16	1.005	**0.958**	**0.976**	**0.979**	**0.975**
17	1.020	**0.968**	1.020	1.028	1.023
18	1.027	**0.993**	1.031	1.037	1.028
LP_{rmse}	-	0.186	0.099	0.124	0.212

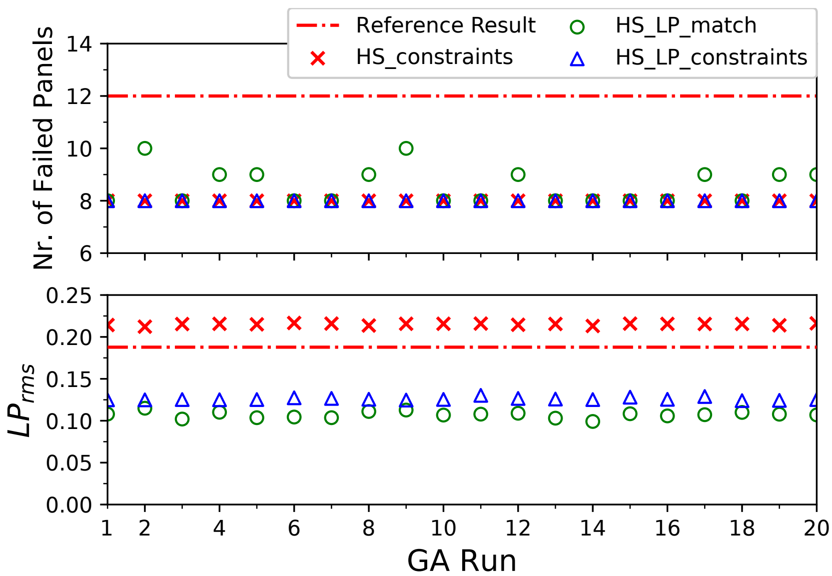


Figure 4.2: Number of failed panels and LP_{rmse} per GA run for different objective functions.

Finally, although both the $LP_{rmse} + \lambda_c$ and λ_c cases consistently retrieved a reduced number of failed panels, the former did so which a much lower LP_{rmse} of 0.124 versus 0.212. The LP_{rmse} of the reference retrieved design amounts to 0.186. If load redistribution had been taken into account, it's possible that the larger error of the λ_c and the subsequent difference in stiffness distribution compared to the continuous optimum could result in a larger number of panels to fail [39].

In the benchmark of the 18-panel horseshoe problem there's no real need to restrict the discrete optimization to the vicinity of the continuous optimum since the mechanical behavior of each patch is mostly independent of one another except for the blending requirement. However, in other design problems such as aircraft wings, restricting the direct search to the vicinity of optima retrieved in the continuous design space brings real value as that guides the second step towards designs with similar stiffness distribution which translates to similarities in the wing deformation under load.

This is important in the aeroelastic tailoring of aircraft wings since the wing shape is necessary in the optimization to ensure that aerodynamic performance is maintained between the continuous and discrete steps. If the wing shape is not maintained, the aerodynamic behavior of the wing defined in the first step becomes invalidated and the retrieved discrete design is unlikely to meet performance criteria even if buckling and strength constraints are satisfied.

Chapter 5

Aircraft Wing Discrete Optimization: Fixed Thickness

A more complex problem is used to validate the use of mechanical constraint approximations in a GA to retrieve stacking sequences. To this end, a discrete optimization using response approximations is used to retrieve stacking sequences of an aircraft wing comprised of four components that best matches a previously retrieved LP continuous optimum. This reference wing has been provided by ONERA together with an FE model and is identified as "ONERA regional wing model" henceforth. Unlike the horseshoe benchmark problem, load redistribution due to stiffness changes is considered by building response approximations using cross-patch and cross-component sensitivities. The approximations are obtained by a novel method which aggregates a number of linear approximations built on multiple reference design points using a KS envelope function. This method helps to ensure that the approximation is conservative and that retrieved stacking sequences satisfy the mechanical constraints. In this step, optimization is carried out under a fixed thickness condition, i.e., the patch thickness distribution derived in the continuous optimum is maintained in the discrete step.

In section 5.1 a general description of the ONERA regional wing model and its components is provided, in addition to load criteria, design variables and reference data. A selection method to reduce the number of mechanical constraints used in the optimization is discussed in section 5.2, followed by a description of the KS approximation method in section 5.3 where its validation is also discussed. Finally, the results of the discrete optimization of the reference wing are presented in section 5.4 under a fixed thickness condition.

5.1 ONERA Regional Wing Model: General Description

The ONERA regional wing model is comprised of four separate components as seen in figure 5.1: Upper Wing (UW) and Lower Wing (LW) skins, Front Spar (FS) and Rear Spar (RS). Each component is split into a number of different stiffness regions (patches) where the stacking sequence is constant.

The structure is divided in a total of 44 different patches, 14 for the upper and lower skins, 8 for each spar. The properties of the ply material used in both optimization steps are given in table 5.4.

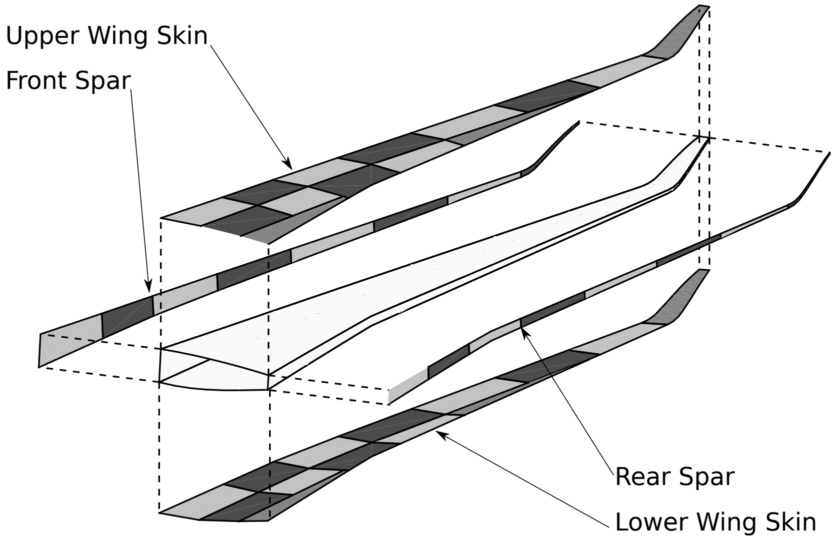


Figure 5.1: ONERA regional wing model layout with the various patches.

5.1.1 Reference Data

The continuous optimization of the ONERA regional wing model was carried out externally and is only used here as a reference for subsequent stacking sequence retrieval in the discrete step of the optimization. Two different cases are considered, one included blending constraints [7] in the continuous design space in order to reduce the mismatch between the feasible regions of the two optimization steps, whereas the other did not. From here on, these cases are referred to as the blended and unblended target, respectively. Both cases were optimized for the eight load cases provided in table 5.1.

The final mass of the retrieved continuous designs is 610.25 Kg with blending constraints and 577.49 Kg otherwise. The 5.7% increase in mass is a direct result of the blending constraints "smoothing" out the thickness and stiffness distribution throughout the structure. In both cases, the thickness of all patches was rounded to multiples of $.2mm$ in order to ensure an equivalent ply count integer. Only the blended target is considered in this chapter since the retrieval of stacking sequences is more likely to succeed if the difference between the feasible design space of the two step optimization is reduced. In chapter 6, a comparison is made between retrieved stacking sequences and total mass starting from the blended and unblended continuous optimum.

Included in the reference data for all patches is the thickness distribution as shown in figure 5.2, target lamination parameters for each patch, responses and their sensitivities in terms of the design variables, i.e., LP and thickness. Only symmetric laminates are considered in the optimization, hence the coupling terms $V_n^B = 0$ and the stiffness C_i of each patch is a function of $C_i = f(V_n^A, V_n^D, t)$. With 44 different patches, a total of 396 design variables are needed to describe the wing in the continuous design space; 8 LP per patch in addition to their thickness.

Additionally, four responses are considered here:

1. Strength(σ): These responses are evaluated according to a failure envelope developed by Ijsselmuiden [36] which is based on the Tsai-Wu criteria to define conservative feasible design regions where strain failure does not occur in any orientation for a given stacking sequence.

2. Buckling reserve factor (λ): The buckling responses are obtained under the assumption that each patch is simply supported for a conservative estimate.

3. Displacement (u): Linear modeling of the wing displacement tracked at 90 locations in the wing, 45 points at the Leading Edge (LE) and the other 45 at the Trailing Edge (TE).

4. Mass: Single value for the entire structure.

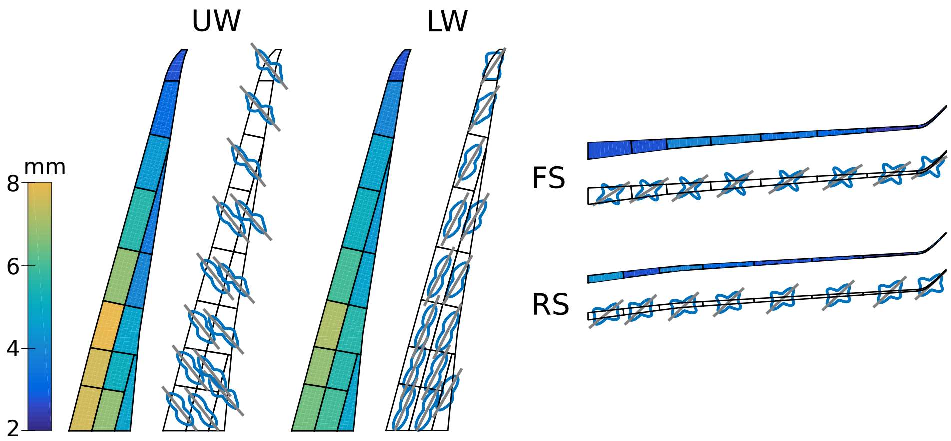


Figure 5.2: Thickness distribution and polar stiffness $E_{11}(\theta)$ of the blended continuous optimum.

Finally, the wing is optimized according to the eight static loads given in table 5.1. The sensitivities of the mechanical constraints and displacement are available for each load case.

Table 5.1: List of Static Load Cases

Load Case	Flight Load	Weight	Dynamic Pressure
LC1	2.5G	MTOW	Sea Level
LC2	-1G	MTOW	Sea Level
LC3	2.5G	MTOW	Cruise
LC4	-1G	MTOW	Cruise
LC5	2.5G	MLW	Sea Level
LC6	-1G	MLW	Sea Level
LC7	2.5G	MLW	Cruise
LC8	-1G	MLW	Cruise

5.1.2 Mechanical Constraints

Where the strength and buckling constraints are considered, the structure is further divided into smaller elements. The buckling regions represent the plate elements delimited by stringer and rib placement throughout the wing, whereas all FEM CQUAD4 elements are considered for the strength constraints.

Buckling constraints are only considered in the UW and LW components. Each wing skin contains over 200 buckling regions where 15 different buckling modes are considered. The strength constraints are applied to all four components and their individual count is given in table 5.2. A total of 3987 constraints per load case are included in the model.

The sensitivities of these responses are obtained from the FEM in the continuous optimization for each design variable and load case. Hence each mechanical constraint is accompanied by

Table 5.2: Mechanical Constraints Breakdown

Component	Strength	Buckling
Upper Wing (UW)	1576	222, 15 modes
Lower Wing (LW)	1586	223, 15 modes
Front Spar (FS)	196	-
Rear Spar (RS)	184	-

396 sensitivities per load case given in equation 5.1 where x_p is a vector containing the design variables of the patch $p = [1, ..., 44]$.

$$\frac{\partial R_i}{\partial x_p} = \left(\frac{\partial R_i}{\partial V_{1-4}^{A,p}}, \frac{\partial R_i}{\partial V_{1-4}^{D,p}}, \frac{\partial R_i}{\partial t_p} \right) \tag{5.1}$$

As stated in chapter 3, pyTLO is capable of handling multiple components at once. This becomes an advantage when response cross-patch and cross-component sensitivities are available. The stacking sequence retrieval can be carried out simultaneously for the four components so that load redistribution due to stiffness changes in one component are taken into account in the optimization of the remaining components. This is especially important with respect to buckling, since although the buckling ratio equation (3.4) is a function of the out-of-plane stiffness terms only, the in-plane loads N_y and N_x are a function of the in-plane stiffness terms. Otherwise, the individual optimization of multiple components cannot guarantee that the design as a whole satisfies the constraints once load redistribution takes place. As a workaround, margins of safety can be applied to ensure that all constraints are satisfied even if the actual load distribution differs, but this comes at the cost of a higher structural weight.

5.2 Constraint Selection

Due to the large number of constraints being used in the model, a selection method was developed to filter out buckling and strength responses that are not at risk of failure without a substantial change in the design variables. This helps to reduce the computational time considerably, since fewer linear approximations are needed in the discrete optimization. Basically, a simplified linear approximation of each constraint is built with respect to its sensitivities and a uniform ΔLP to determine what change in LP is causes it to reach a failure threshold.

If the value of ΔLP is below a pre-defined threshold, the constraint R_i is added to a list of constraints used in the fitness evaluation in the discrete optimization step. This process is summarized in the flowchart given in figure 5.3. This task is only carried out once and the selection list is then stored for subsequent optimization runs.

The ΔLP threshold is defined in the first step by using a number of reference GA runs with a simple LP matching fitness function. The largest LP error found in these runs is multiplied by a safety margin of 1.5 to ensure that the constraint selection is conservative. A total of 10 reference runs N_{GA} were used to obtain the threshold for the aircraft wing. Since a constant Δ LP is considered, the sensitivities of a given response R_i are simply summed together for the approximation. A couple of additional guidelines are applied to the process regarding the selection across multiple load cases and modes where buckling is concerned. If constraint R_i is critical in multiple load cases, only the most critical one (lowest ΔLP_i) is added to the selection list.

Buckling constraints require an additional step; since there are 15 modes per response, the most critical one needs to be selected. Two different modes of the same response are added to the selection list if there are critical responses with opposing ΔLP_i sign. If the slope of

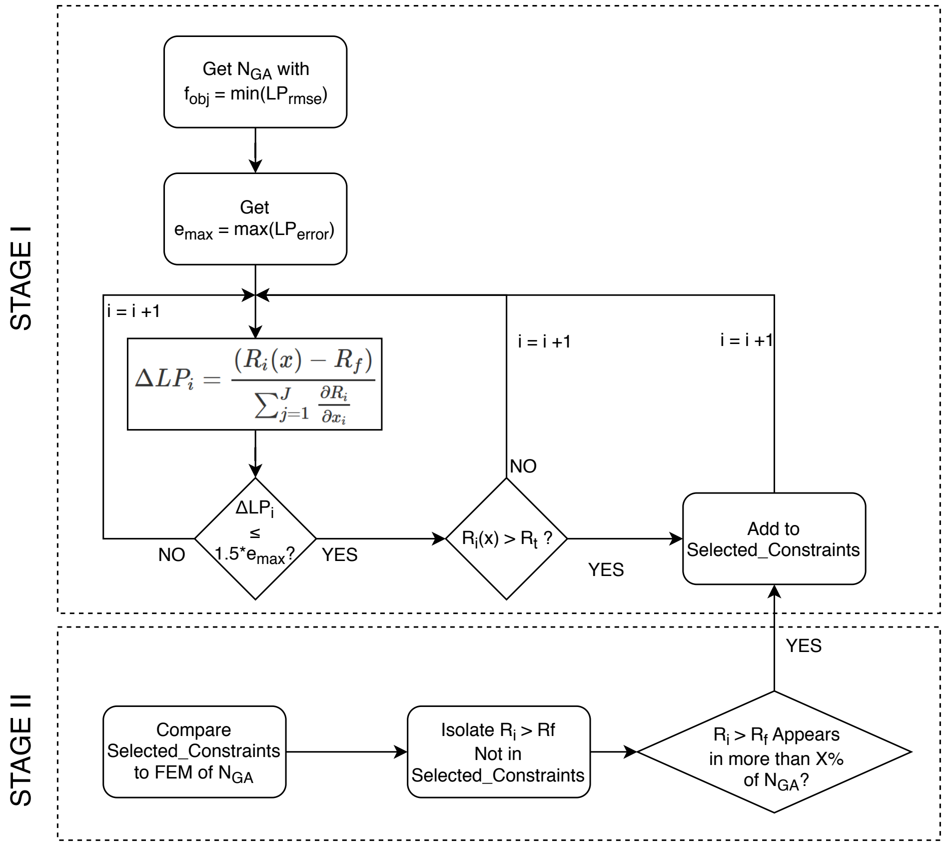


Figure 5.3: Constraint Selection Process

the approximation is negative for one mode, and positive for the other, both modes should be considered.

Although this is a rather simplistic method that doesn't take into account LP feasible regions nor how their values don't vary independently, the selection is thorough and the number of constraints added in the second stage is small. For instance, out of 1783 selected constraints for all four components, only 18 were added in the second stage.

If this selection is not carried out, the additional number of calculations would greatly increase the computational time. At 3987 constraints and 8 load cases, that's 127.6×10^9 linear approximations required for a GA run with 200 individuals and 2000 generations. With the constraint selection, a total of 7.1×10^9 approximations are required, which is a reduction of 94.4%.

5.3 Response Approximation Method

Approximations of the response are used in the discrete optimization step in order to avoid the high computational cost of using an FEM evaluation for each design point generated by the GA. Local approximations based on a Taylor's expansion around a reference design point are used in this project and combined with KS envelope functions to provide a conservative approximation.

5.3.1 Linear Approximations

Buckling and strength responses are approximated linearly from a reference design point such as a continuous optimum using the sensitivities of the response in terms of the design variables, namely LP and patch thickness. The contribution of cross-patch sensitivities is also factored in as shown in equation 5.2.

$$\tilde{R}(V,t) = R_0 + \sum_{p=1}^{N_p}\left[\sum_{i=1}^{4} \frac{\partial R}{\partial V_{i,p}^{A,D}}\bigg|_0 \cdot \left(V_{i,p}^{A,D} - V_{0i,p}^{A,D}\right) + \frac{\partial R}{\partial t_p}\cdot(t_p - t_{0,p})\right] \tag{5.2}$$

These approximations are only accurate in the immediate vicinity of the reference point and perform poorly as the LP error increases. As a consequence, they are not reliable in the context of this project due to LP varying non-linearly in addition to the relatively large LP errors between the target and retrieved stacking sequences. For this reason, these approximations are instead used here to create a more conservative approach based on KS envelope functions [37].

5.3.2 Response Approximations with Kreisselmeier-Steinhauser Function

The response approximations used in the discrete optimization step are obtained by building $N_{ref}+1$ linear approximations of the response R_i, where N_{ref} is the number of reference design points. These reference design points are comprised of the continuous target as well as subsequent discrete optima obtained with the GA. In principle, the distance in terms of LP to some of these reference points x_0 to the individuals x_i evaluated during a GA run is smaller than to the continuous target and linear approximations. As a consequence, linear approximations built on the nearest reference point are more accurate than those built on the continuous target. This concept can be visualized with a simplified representation of the design space shown in figure 5.4 where $LP_{rms2} << LP_{rms1}$.

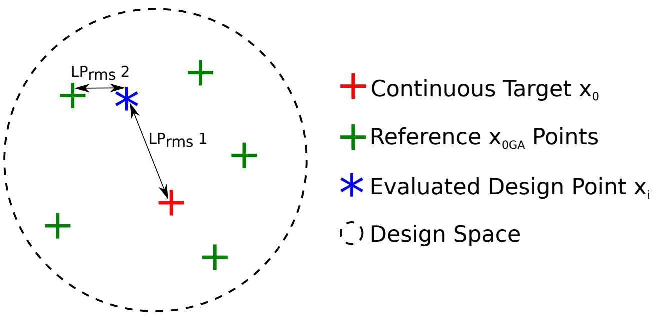


Figure 5.4: Representation of a distribution of reference points in the design space

After building the linear approximations over all the reference points for a given constraint, the final response value R_i for a constraint is obtained by using the KS function given in equation 5.3 which selects the largest (most conservative) linear approximation $\tilde{R}_{i,j}$ from a set of R_i approximations, i.e., $R_{max} = max([\tilde{R}_{i,1}, ..., \tilde{R}_{i,N_{ref}}])$ in addition to the ρ contribution that controls how conservative the KS output is.

$$\tilde{R}_i\left(\tilde{R}_{i,j}(x)\right) = R_{\max}(x) + \frac{1}{\rho}\ln\left[\sum_{j=1}^{N_{ref}+1} \mathrm{e}^{\rho\left(\tilde{R}_{i,j}(x) - R_{\max}(x)\right)}\right] \tag{5.3}$$

In equation 5.3, i refers to the constraint ID, j is the linear approximation ID and x is the vector containing the design variables.

5.3.3 Validation of the KS Approximation Strategy

The approximation method based on the KS envelope functions was validated by carrying out a large number of discrete optimizations using pyTLO. All the optima retrieved in these run are evaluated using the FEM and then stored in a “GA” pool. The validation process is comprised of the following steps:

1. N_{ref} reference design points are selected from the GA pool.
2. Linear and KS approximations of the remaining design points using the selected reference design points are built.
3. Calculate approximation error, median and RMS error with respect to FEM evaluations.
4. Repeat steps 1-3 10 times to ensure multiple combinations of reference points and approximated optima.
5. Plot the results.

A total of 80 GA optima were used to validate the approximation method for different numbers of reference runs and ρ coefficients. The median of the approximation error is shown in figure 5.5 for both simple linear approximations based on the continuous target and KS approximations.

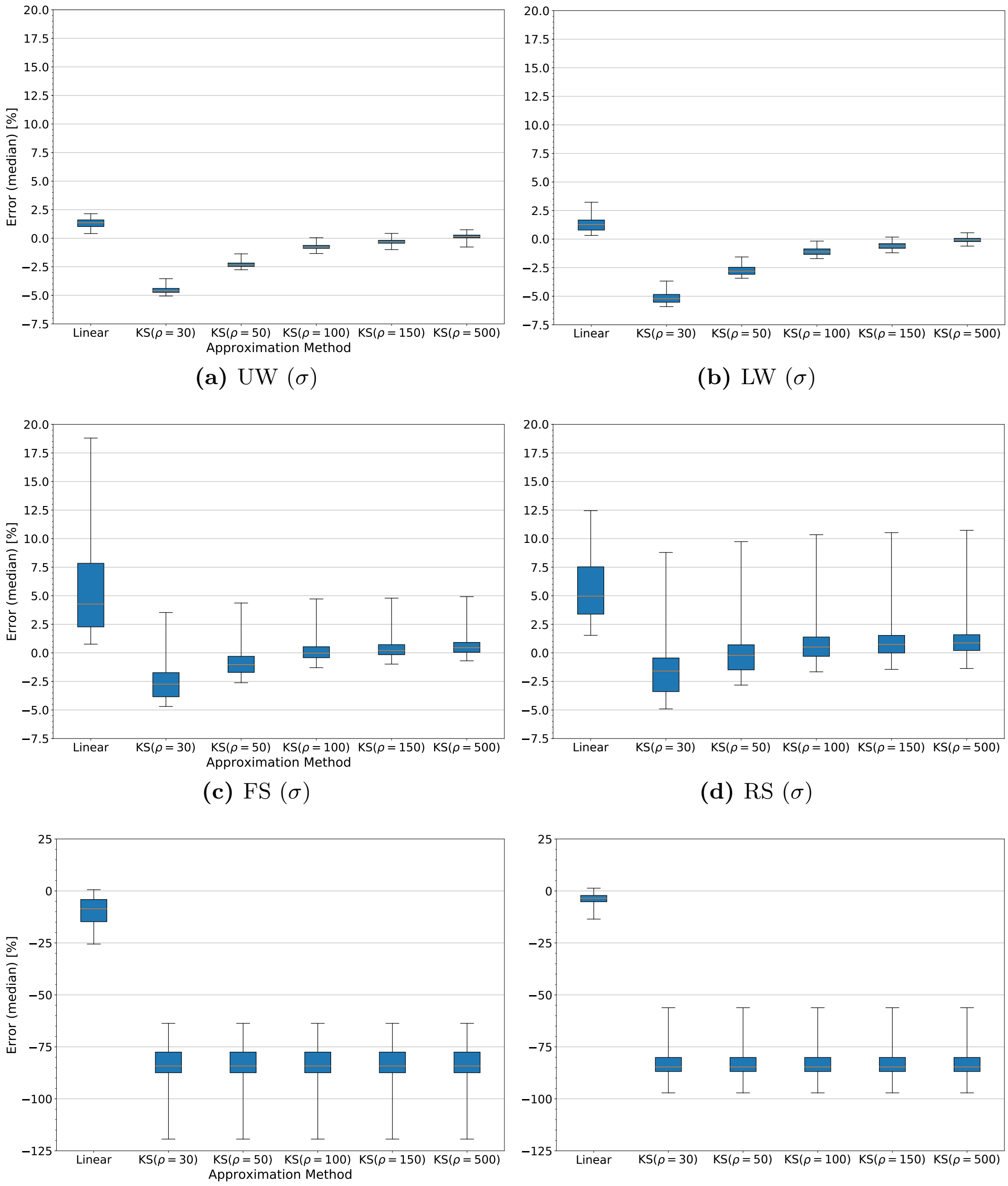
Where the linear strength(σ) approximations are concerned, not only is the error large for the FS and RS components, the approximation is non conservative in all cases. The consequence is that retrieving feasible designs on linear approximations alone is not possible as the constraints remain below the failure threshold instead of being added as a penalty to a design’s fitness. Although these designs are deemed feasible by the GA evaluation, that is not the case when the results are finally verified with the FEM.

On the other hand, there is a marked improvement in using the KS method. Except for larger values of ρ, the approximation is conservative in all cases and the average error is under 5% for the majority of the constraints even though no sampling was used in the selection of the reference runs. The spars are affected by a larger error; stacking sequence retrieval of the front spar is affected by a larger LP_{rmse} which in turn affects the approximation, whereas the rear spar constraints are rarely near failure, meaning that the constraints are not guiding its optimization process, only the LP_{rmse} is.

It can be seen in figures 5.5 e) and f) that the approximation of the buckling responses using the KS method is extremely poor in comparison to a simple linear approximation. This is a consequence of some of the reference runs containing outliers in the sensitivities which are orders of magnitude larger than those in the continuous optimum, leading to a large variation in the approximation from small changes in the design variables. Evidently, it’s impossible to match the LP of two different design points and if both display large outlier sensitivities in the response, one of them will compromise the KS of the response since it selects the larger linear approximation.

The source of these outlier sensitivities provided by the FEM could not be identified and it sometimes occurs in the strength sensitivities. These reference design points are filtered out of the N_{ref} runs to ensure that the strength approximation is not compromised. Unlike the linear strength approximations, the buckling counterparts are conservative (negative error %) and can be used in the GA instead of the KS to retrieve feasible results.

Finally, the best values of the ρ parameter were defined from these results. Subsequent optimizations used $\rho = 150$ for the UW, $\rho = 500$ for the LW, $\rho = 100$ for the FS and RS. The number of reference linear approximations per constraint is set to $N_{ref} = 10$ as increasing further incurs a higher computational cost and increases the approximation error as seen in figure 5.6. Only the UW diagram is provided here since the trend is similar for the four components.



(a) UW (σ) **(b)** LW (σ)

(c) FS (σ) **(d)** RS (σ)

(e) UW (λ) **(f)** LW (λ)

Figure 5.5: Median of the approximation error for all components. $N_{ref} = 10$

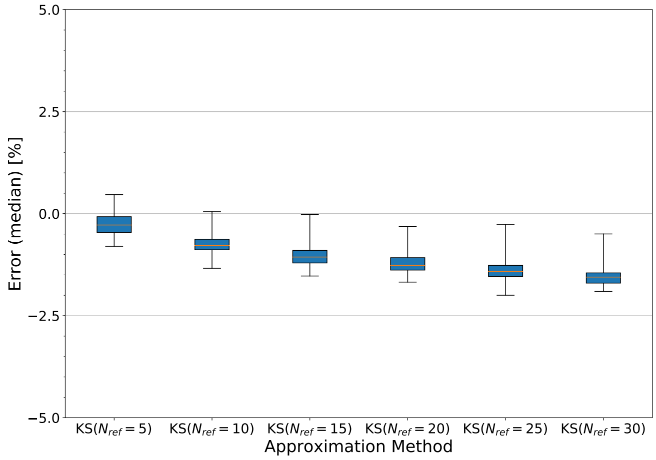


Figure 5.6: Median of the UW strength approximation error for an increasing N_{ref}. $\rho = 100$.

5.4 Results - Fixed Thickness

The first set of discrete optimization runs of the reference aircraft wing are done under a fixed thickness condition, i.e., no additional plies are added over the optimal thickness obtained in the continuous optimization. The aim is to gauge the improvement over LP-matching fitness functions if they are subjected to indirect mechanical constraints. For this exercise the only manufacturing constraint considered is symmetry. The remaining pyTLO GA settings are given in table 5.3. The run-time of each optimization using the KS approximation for 1783 mechanical constraints is around 1 hour for fixed thickness. A simple LP_{rmse} run of all components takes 15 minutes.

Table 5.3: Fixed thickness GA settings

attribute	Value
`elitism`	0.02
`initial_pop`	500
`pop_size`	200
`cx_fraction`	0.75
`mut_pb`	0.25
`max_gen`	2000
`rf_freq`	-
`fit_tol`	0.001
`update_cycle`	10
`fit_weights`	-1
`delete_duplicates`	False
`refresh_population`	False

Table 5.4: Ply material properties

Property	Value
E_1 $[GPa]$	177
E_2 $[GPa]$	10.8
G_{12} $[GPa]$	7.6
μ_{12} $[-]$	0.27
$Thickness$ $[mm]$	0.20

5.4.1 LP Matching Objective

Simple LP-matching fitness does not take into account constraints and hence no load redistribution. The discrete optimization is carried out individually for each component with the aim of reducing the LP_{rmse} between the target continuous optimum and retrieved stacking sequences as given by equation 5.4.

$$f_{obj,1} = min\left(\frac{1}{N_p}\sum_{n=1}^{N_p}\sum_{j=1}^{8}\sqrt{\frac{(LP_{j,GA}(x_i) - LP_{j,opt})^2}{8}}\right) \tag{5.4}$$

After the discrete optimization is complete, retrieved stacking sequences are re-evaluated with an FEM. The buckling and strength results are given in figures 6.9 and 6.10. The surface plots pertain to the responses subjected to load case 3 (LC3), whereas the scatter plots (figures 5.11 and 5.12) show the largest value of each response across all load cases.

It can be seen that the retrieved stacking sequences aren't able to satisfy a number of constraints and buckling performance is especially poor. This is expected as a consequence of the load redistribution due to changes in the in-plane V_{1-4}^{A} terms not having any weight on the fitness of the individuals. The obtained LP_{rmse} for each component is provided in table 5.5.

Table 5.5: LP_{rmse} of the retrieved stacking sequences.

Component	$f_{obj} = min(LP_{rmse})$	$f_{obj} = min(LP_{rmse} + C(\lambda, \sigma))$
UW	0.138	0.189
LW	0.134	0.174
FS	0.170	0.291
RS	0.105	0.115

5.4.2 LP Matching Subjected to Mechanical Constraints

Subsequent discrete optimizations are made using a penalty based objective function where active response constraints [1] are added to an individual's fitness on top of the LP_{rmse} penalty as shown in equation 5.5. In doing so, the objective is still to match the LP parameters of the continuous target but with an incentive to explore the vicinity of the reference point and rank designs according to the number and severity of unsatisfied constraints.

$$f_{obj,2} = min\left(LP_{rmse}\right) + C(\sigma, \lambda) \tag{5.5}$$

The constraint penalty $C(\sigma, \lambda)$ is obtained by approximating the selected responses and summing the portion over the failure threshold as shown in equation 5.6.

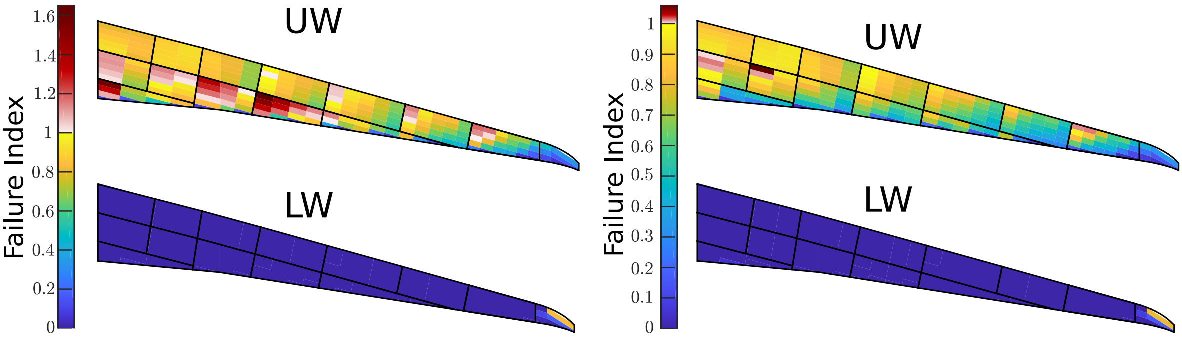
$$C(\sigma, \lambda) = \sum_{i=1}^{N_\sigma}\delta(\tilde{\sigma}_i - \sigma_f) + \sum_{i=j}^{N_\lambda}\gamma(\tilde{\lambda}_i - \lambda_f) \tag{5.6}$$

$$\begin{cases}\delta = 1 & \tilde{\sigma} > \sigma_f\\ \delta = 0 & Otherwise\end{cases} \qquad \begin{cases}\gamma = 1 & \tilde{\lambda} > \lambda_f\\ \gamma = 0 & Otherwise\end{cases}$$

Where $\tilde{\sigma}_i$ and σ_f are the strength constraint approximations and their respective failure threshold. Similarly, $\tilde{\lambda}_i$ and λ_f correspond to the buckling constraint approximations and failure threshold. Except for the fitness function, all other optimization settings remain the same.

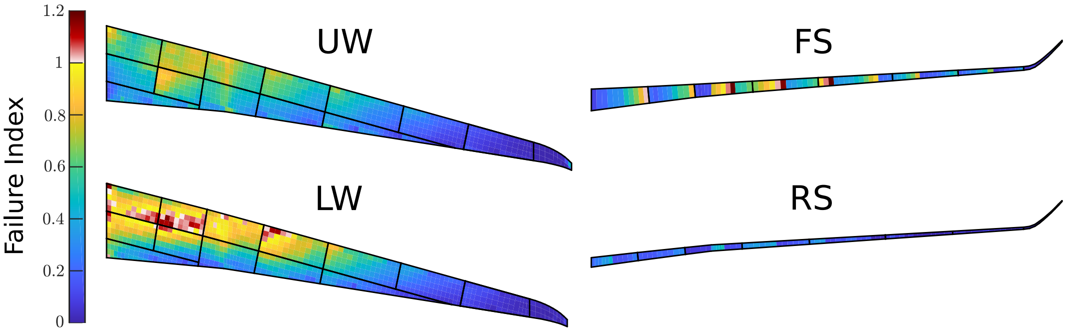
[1] A response constraint is deemed "active" if its value is above the failure threshold.

Even though no additional plies were added, it can be seen that considerable improvements were obtained using $f_{obj,2}$. By comparing the response values plotted for each fitness function (fig. 5.11 and fig. 5.12), it can be seen that both the number of active constraints and their magnitude is lower. Where buckling is concerned, a 23% reduction in the failure index is achieved. The retrieved LP_{rmse} for each component is given in table 5.5 and can be compared to the results obtained through $f_{obj,1}$.

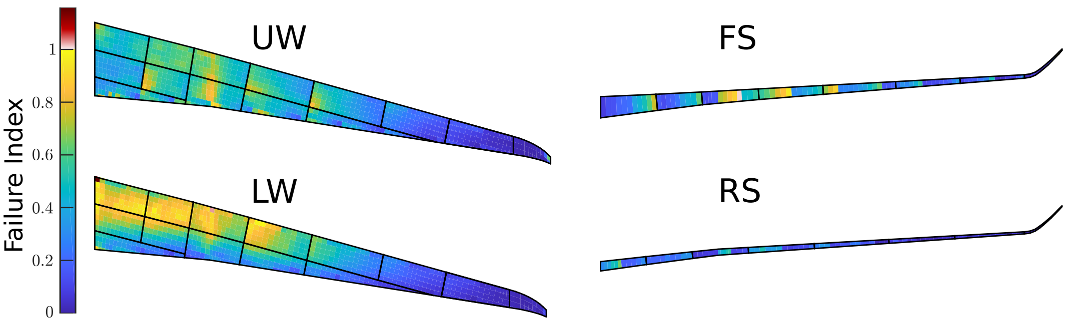


(a) $f_{obj,1} = min(LP_{rmse})$ **(b)** $f_{obj,2} = min(LP_{rmse} + C(\lambda, \sigma))$

Figure 5.7: LC3 Buckling Failure Index (λ) for two different fitness functions.



(a) $f_{obj,1} = min(LP_{rmse})$



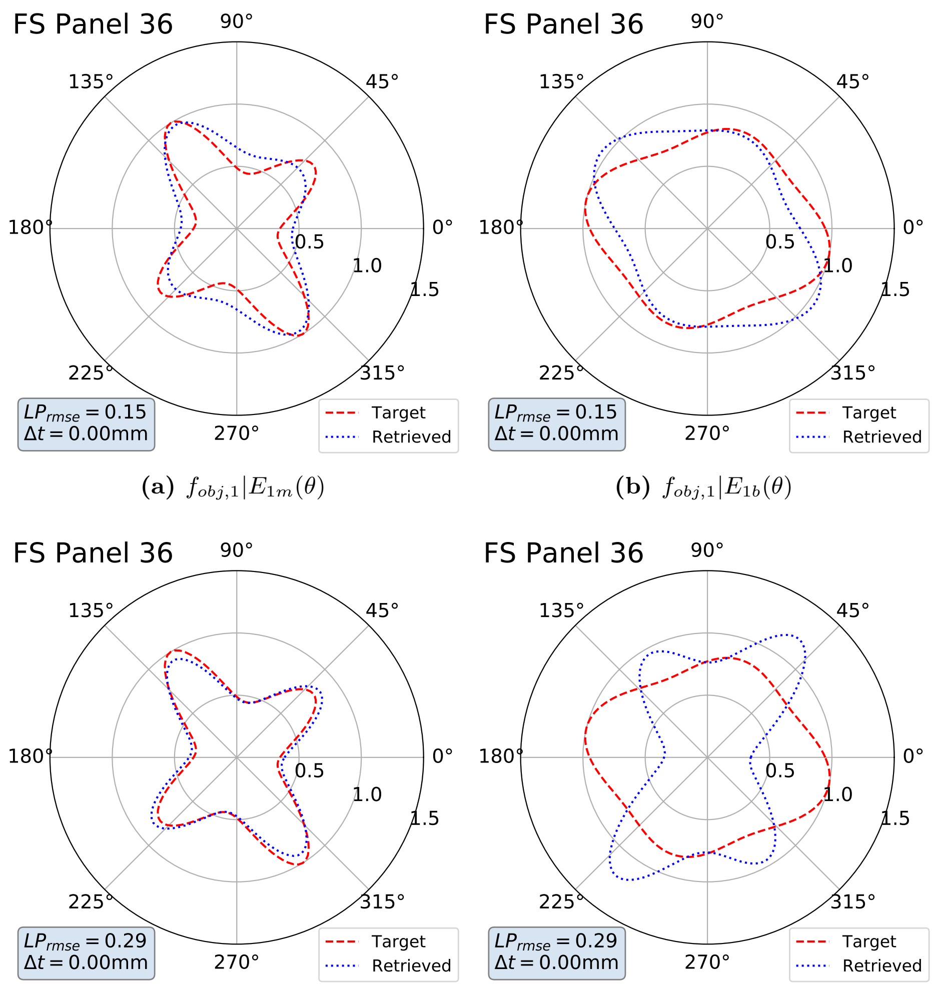
(b) $f_{obj,2} = min(LP_{rmse} + C(\lambda, \sigma))$

Figure 5.8: LC3 Strength Failure Index (σ) for two different fitness functions.

The optimum obtained with $f_{obj,2}$ incurred a larger LP_{rmse} as a trade-off to a lower constraint penalty. This is especially evident with the FS, which is only subjected to strength constraints and its out-of-plane stiffness has no impact on the buckling performance of the wing skins. As a result, although the LP_{rmse} is 71% larger, most of that error originates in the V_{1-4}^D

terms as shown in figure 5.10 where the $rmse(V_{1-4}^{A})$ and $rmse(V_{1-4}^{D})$ distribution per patch are plotted for each component.

This is also visible in the polar plots of the membrane stiffness E_{1m} and bending stiffness E_{1b}stiffness plots for a single FS patch is given in figure A.12. A compromise in matching the E_{1m} and E_{1b} stiffness distribution is seen in the design resulting from a simple LP matching objective ($f_{obj,1}$), whereas a clear preference towards matching the E_{1m} distribution is seen in the design retrieved with $f_{obj,2}$.

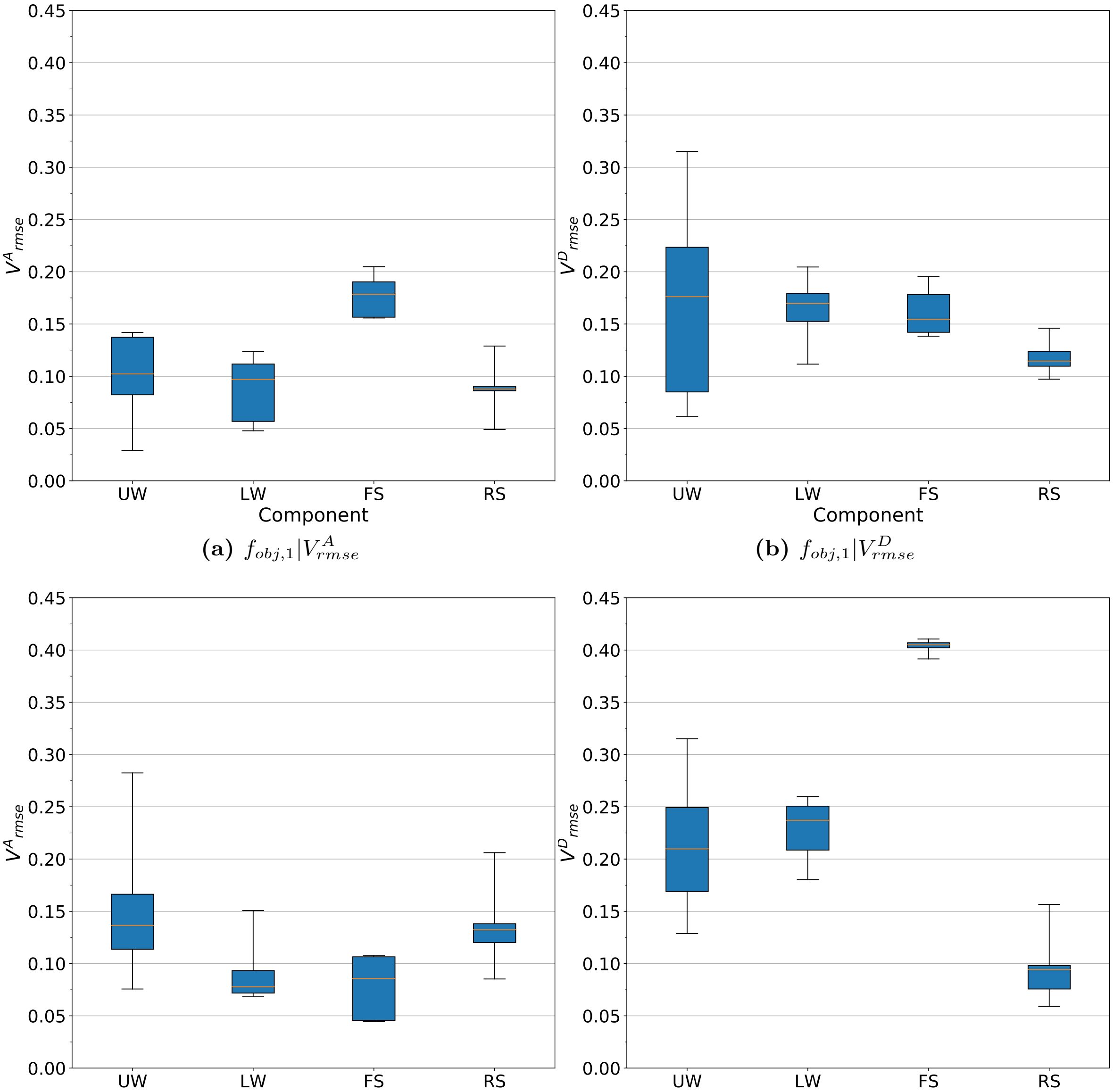


(a) $f_{obj,1}|E_{1m}(\theta)$ **(b)** $f_{obj,1}|E_{1b}(\theta)$

(c) $f_{obj,2}|E_{1m}(\theta)$ **(d)** $f_{obj,2}|E_{1b}(\theta)$

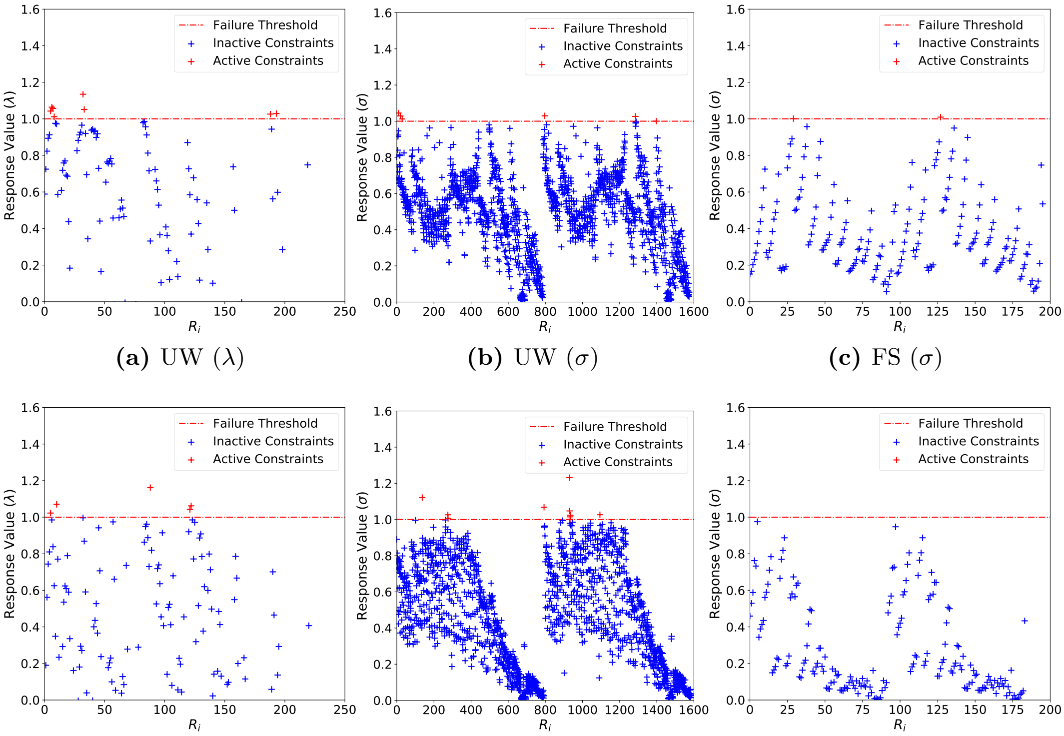
Figure 5.9: Polar stiffness E_{1m} and $E_{1}b$ of FS patch 36 for $f_{obj,1}$ and $f_{obj,2}$.

After a total of 30 discrete optimization runs using the $f_{obj,2}$ objective, no stacking sequences with zero mechanical constraints were retrieved. This implies that the SST method to retrieve blended stacking sequences constrains the design space significantly and that additional plies have to be added in the discrete design in order to achieve feasible stacking sequences with no failed elements under the prescribed design loads.



(a) $f_{obj,1}|V^A_{rmse}$ (b) $f_{obj,1}|V^D_{rmse}$ (c) $f_{obj,2}|V^A_{rmse}$ (d) $f_{obj,2}|V^D_{rmse}$

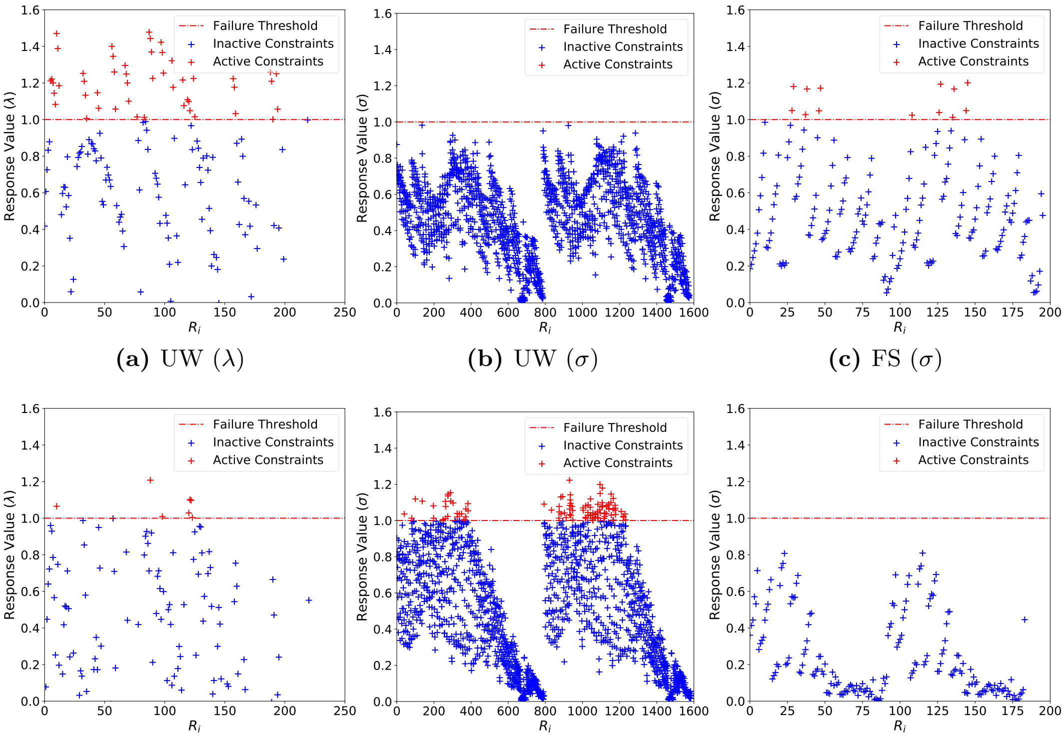
Figure 5.10: V^A_{rmse} and V^D_{rmse} distribution for all panels.



(a) UW (λ) **(b)** UW (σ) **(c)** FS (σ)

(d) LW (λ) **(e)** LW (σ) **(f)** RS (σ)

Figure 5.11: Maximum response values across all load cases ($f_{obj,2} = min[LP_{rmse} + C(\lambda, \sigma)]$).



(a) UW (λ) **(b)** UW (σ) **(c)** FS (σ)

(d) LW (λ) **(e)** LW (σ) **(f)** RS (σ)

Figure 5.12: Maximum response values across all load cases ($f_{obj,1} = min(LP_{rmse})$).

Chapter 6

Aircraft Wing Discrete Optimization: Variable Mass

In this chapter the process to retrieve stacking sequences for the ONERA regional wing model that are feasible, e.g., having no failed elements under prescribed design loads is described. In this optimization scheme, patch thickness and wing mass is no longer restricted to the distribution retrieved in the continuous optimization and is allowed to increase until feasible designs are obtained. By allowing an increase in mass, a multi-objective optimization is necessary and additional steps are required to ensure that 1) the response approximations remain valid and 2), that the optimization objective yields optimal results in terms of both LP-matching and mass increase over the reference continuous optima. Furthermore, an additional optimization objective is considered to match the wing shape due to aerodynamic loading obtained in the continuous optimization to ensure that cruise performance the solutions obtained in the discrete design space is equivalent to the target.

In section 6.1 the changes to the discrete optimization method for feasible designs are discussed. Here, the ϵ-constraint method for multi-objective optimization is introduced. The implementation of the wing shape objective is discussed in section 6.2, followed by an update to the process of building KS-approximations in section 6.3 that factors in changing thickness distribution. Finally, results of the discrete optimization are discussed in section 6.4 where three different feasible designs are compared, two for the blended case and one for the unblended continuous reference.

6.1 Variable Mass Discrete Optimization

In previous discrete optimizations, the thickness distribution throughout the wing components was fixed with respect to the continuous optimum. Since no feasible designs were retrieved, the increase in thickness is allowed in subsequent optimization runs.

In doing so, total wing mass becomes an additional objective in the optimization since it is desirable to obtain the lightest designs that comply with all other design criteria. Using the same objective function (eq. 5.5) as used in the stacking sequence retrieval method described in chapter 5 is no longer possible as there is an inverse relation between increasing mass and achievable LP_{rmse}; increasing the patch thickness results in a relaxation of the design space since more stacking sequence options are available and a lower LP_{rmse} is possible, and unless the mass increase is restricted, the GA selection process favors heavier designs.

On the other hand, multi-objective functions to minimize both the LP_{rmse} and mass increase also pose some challenges. The interaction between the various objectives needs to be considered since these can conflict with one another as is the case with mass and LP_{rmse} objectives. Additionally, weighting of the objectives and their impact on the design fitness needs to be carefully selected to avoid solutions being dominated by a single objective. A large number of runs is needed in order to assemble a set of solutions and weights that best satisfy the objectives and support the decision making of which solution best fits design criteria [40].

$$f(x) = min(f_1(x), f_2(x), ..., f_n(x)) \tag{6.1}$$

The two most common strategies in handling multi-objective optimization problems rely on either using a single fitness function where all objectives are aggregated (eq. 6.1) or using a single objective function where all other objectives are considered as constraints (eq. 6.2). In the former, optimized weighting of each objective's contribution to the fitness is needed to ensure non-dominated solutions are retrieved whereas in the latter it is necessary to know the objective bounds in order to set them up as constraints.

$$\begin{aligned} f(x) &= min(f_1(x)) \\ st.\ f_2(x) &\leq \epsilon_2,\ ...,\ f_n(x) \leq \epsilon_n \end{aligned} \tag{6.2}$$

In the current problem, only two objectives are considered, mass and LP_{rmse}. Scaling these objectives and selecting the appropriate weights in the fitness function requires a more thorough search of the design space and subsequently a large number of discrete optimizations and FEM evaluations if compared to the second approach. Setting one of these objectives as a constraint requires little additional action since their bounds are easy to define. The LP_{rmse} bounds can be determined from the discrete runs done for chapter 5, or mass can be added as a constraint in a single objective function and relaxed whenever the discrete optimization is unable to retrieve feasible designs. This approach is known as the ϵ-constraint method. By changing the mass constraint upper bound incrementally, each set of solutions is Pareto optimal [41].

The change in mass in pyTLO's GA is controlled by an upper limit of thickness on each patch. Each patch is assigned lower and upper bounds $[N_{min},\ N_{max}]$ for their thickness; the lower bound equals the target continuous optimum. The upper limit sets the maximum number of plies in the SST but no additional plies are added over this limit during the optimization. Since the SST can have non-coding columns, i.e. stacking sequences that are not used in the retrieved design, the optimizer is free to change between higher or lower thicknesses so long as these are within N_{min}, N_{max}. For a mass increase of up to 10%, 10 plies were added as the upper limit of each patch.

Mass Constraints

In order to restrict wing mass increase in the discrete optimization, an additional indirect constraint ϵ_m is added to the LP-matching fitness function as shown in equation 6.3, where $\beta\Delta m$ is the penalty given to a design whose mass exceeds the constraint is the thickness distribution throughout the structure.

$$f_{obj,3} = min\left(LP_{rmse}\right) + C(\sigma, \lambda) + \beta \cdot \Delta m \tag{6.3}$$

The penalty $\beta \cdot \Delta m$ is simply the total added mass over the continuous target multiplied by a factor β (eq. 6.5) large enough to discourage the GA to select such designs. The design mass is calculated from sensitivities $\frac{\partial m}{\partial t_p}$ obtained from the FEM and is given by equation 6.4, where N_p is the total number of patches in the structure and Δt_p is the change in thickness of patch n versus the target continuous optimum.

$$\Delta m = \sum_{p=1}^{N_p} \frac{\partial m}{\partial t_p} \cdot \Delta t_p \tag{6.4}$$

$$\begin{cases} \beta = 0 & \frac{\Delta m}{m_{opt}} \leq \epsilon_m \\ \beta = P & \frac{\Delta m}{m_{opt}} > \epsilon_m \end{cases} \tag{6.5}$$

The constraint threshold ϵ_m is set as a $x\%$ increase in mass over the target and is changed incrementally by 1% until a feasible design is found as shown in figure 6.1.

6.2 Wing Shape Matching

The resulting wing deformation from aerodynamic loading determines its performance during cruise. The desired shape for optimal performance is obtained through aeroelastic tailoring of the wing for which composites are especially suited due to the possibility of controlling stiffness direction. This is taken into account in the continuous optimization of the reference aircraft wing and should be propagated to the discrete step to ensure that retrieved stacking sequences match the optimal shape so that cruise performance is maintained in addition to matching initial conditions for gust loading analysis.

This is factored into the fitness equation by adding the RMS error of the LE and TE vertical displacements u_k to $f_{obj,3}$. The displacement is tracked at 45 points in both LE and TE so a total of $K = 90$ displacements are considered. The error is calculated by multiplying the displacement sensitivities with the changes to the design variables in comparison to the reference optimum.

$$f_{obj,3} = min\left(LP_{rmse} + u_{rmse}\right) + C(\sigma, \lambda) + \beta \cdot \Delta m \tag{6.6}$$

$$u_{rmse} = \sqrt{\frac{1}{K}\sum_{k=1}^{K} \Delta u_k^2}$$

$$\Delta u_k = \sum_{p=1}^{N_p} \left(\sum_{i=1}^{4} \frac{\partial u_k}{\partial V_{i,p}^{A,D}}\bigg|_0 \cdot \Delta V_{i,p}^{A,D} + \frac{\partial u_k}{\partial t_p} \cdot \Delta t_p \right)$$

Unlike mass, the displacement error objective does not conflict with the LP_{rmse} objective as both are minimized by reducing the error between retrieved and target LP (eq. 6.7).

$$\lim_{\Delta LP \to 0} LP_{rmse} = \lim_{\Delta LP \to 0} u_{rmse} = 0 \tag{6.7}$$

However, it does act as a weight on which LP to prioritize in order to match the target displacement and a check is made using only the displacement as an objective (eq. 6.8) to determine if there is any impact in retrieved solutions. The chosen reference load case is LC3 since that is when wing displacement is largest and any mismatch between retrieved and target solutions is more evident. As the displacement model of the reference wing is linear, matching the displacement of one of the load cases is sufficient and the 1g displacement can be obtained simply by dividing the LC3 displacement by its aerodynamic load, e.g. $u_{k,1g} = u_{k,2.5g}/2.5$.

$$f_{obj,4} = min\left(u_{rmse}\right) + C(\sigma, \lambda) + \beta \cdot \Delta m \tag{6.8}$$

6.3 KS Process for Changing Mass

By adding additional plies to the design, changes to the stiffness distribution in the wing can be significant, especially where out of plane stiffness is concerned since the effect of increasing thickness is more pronounced. In order to ensure that the reference design points used to build the KS approximations remain accurate, an update cycle is added to replace reference runs of designs having thickness distributions no longer relevant to the optimization with more up-to-date references as shown in figure 6.1 where m_{ga}/m_{opt} is the ratio between target optimal mass and retrieved mass of the best individual of the discrete optimization.

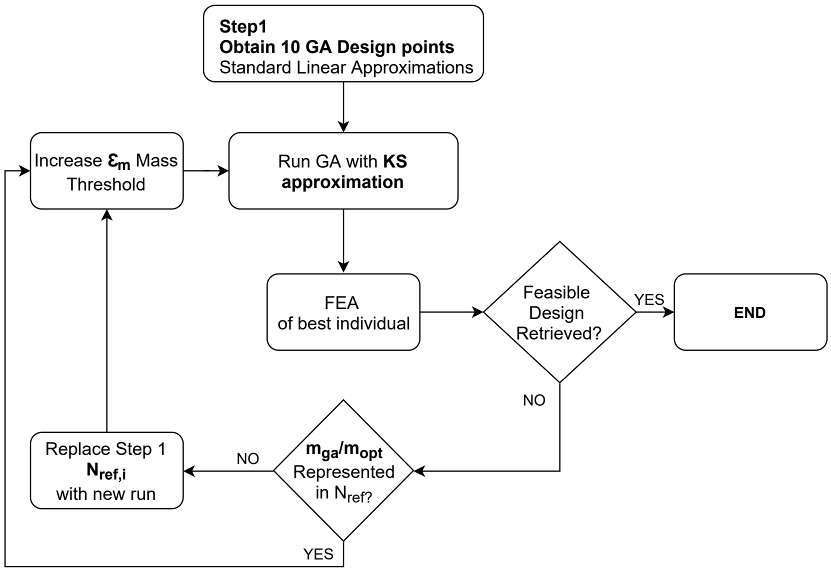


Figure 6.1: Discrete optimization flowchart with varying thickness distribution

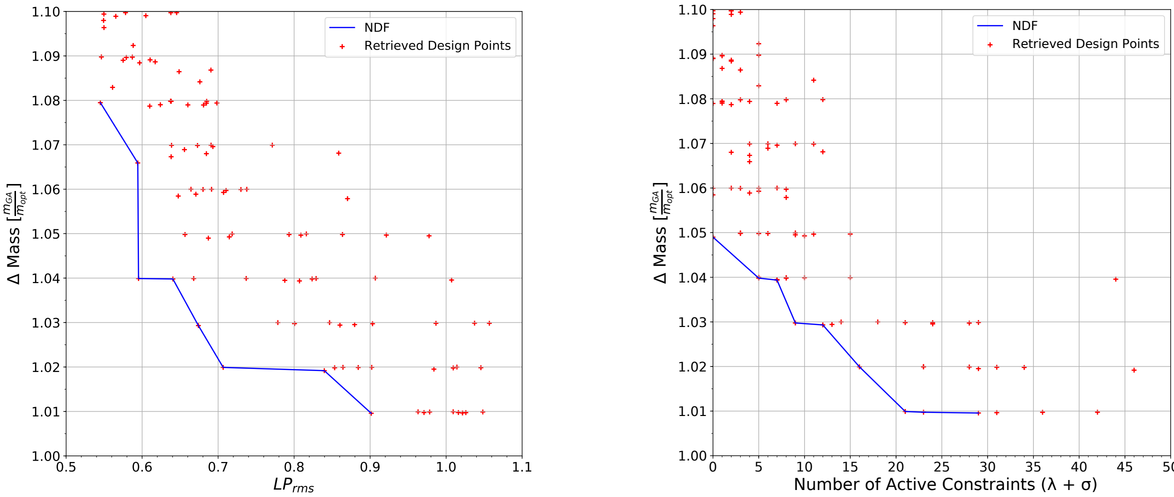
6.4 Results

In this section, the results of the discrete optimization subjected to mass constraints are discussed. Two different continuous optimum targets are considered; the target obtained with blending constraints is discussed first, followed by the unblended target. Two solutions for the blended target are provided using the $f_{obj,3}$ and $f_{obj,4}$ objective functions and one solution for the unblended target based on the $f_{obj,3}$ objective. Finally, blended and unblended solutions are compared in terms of mass increase, displacement and LP_{rmse} matching.

6.4.1 Blended Continuous Optimum

For this optimization a total of 100 runs using the fitness function given in equation 6.6 were carried out in order to cover a mass constraint ϵ_m increase of 10% in steps of 1%. By plotting the resulting number of active mechanical constraints versus mass increase and LP_{rmse} versus mass, a conclusion can be made on the optimality of the feasible solutions. The optimization settings are unchanged according to table 5.3, the same ply material properties and manufacturing constraints (symmetry) apply.

The lightest design retrieved (solution 1) incurred a 4.9% (29.9 Kg) mass increase over the target blended optimum with $\epsilon_m = 1.05$, The mass, number of active constraints and LP_{rmse} of best individuals of each of the 100 runs are plotted in figures 6.2 a) and b). The genotype



(a) LP_{rmse}-Mass (b) Active Constraints - Mass

Figure 6.2: Effect of Mass increase on active constraints and LP_{rmse}. Both quantities are a sum of the four wing components. NDF: Non-Dominated-Front

of the SST obtained for each component is given in table 6.1 and a graphical representation of the SST is given in figure 6.3.

The inverse relation between the mass change and achievable LP_{rmse} discussed earlier is clearly seen in figure 6.2a. The GA solutions are all clustered at the upper limit of each ϵ_m constraint at least until $\epsilon_m \geq 1.05$, where solutions with no mechanical constraint penalty are possible. At this stage, it's possible for the GA to favor those solutions at the cost of a higher LP_{rmse}.

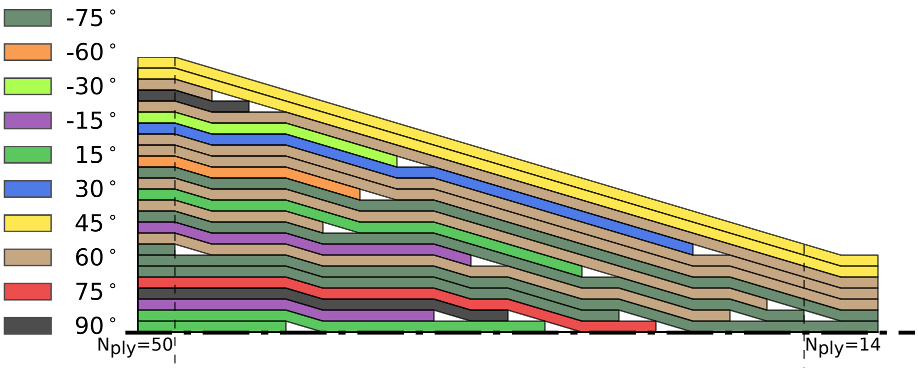
In terms of mass increase, it can be argued that lightest solution retrieved is near optimal, since no feasible solutions were retrieved at $\epsilon_m \leq 1.04$ and the best solution in this case still had 5 active constraints. On the other hand, it is not the most efficient solution in terms of LP_{rmse} achieved, since lower results were obtained for lower mass. Regardless, having no active constraints takes precedence over an increase of 4% per component in LP_{rmse}.

Overall, it can be seen that even at higher $\epsilon_m \geq 1.05$, unfeasible designs are still frequent which implies that not all GA runs are successful in retrieving designs without active constraints. This can be improved by increasing the penalty associated with the mechanical constraints or by increasing the number of generations per GA run. More accurate approximation schemes can also improve the success rate.

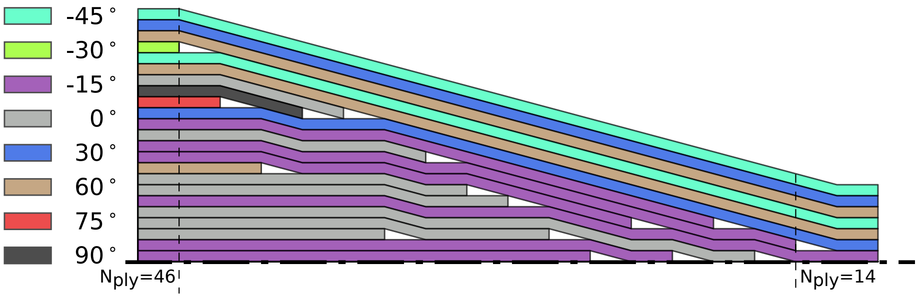
A single run using the $f_{obj,4}$ was necessary to retrieve a second solution with a $\frac{\Delta m}{m_{opt}}$ of 1.05. subsequently, 10 runs using the same displacement objective were done to explore the $\epsilon_m \leq 1.045$ range, but no feasible designs were retrieved, further enforcing the argument that the two retrieved solutions are indeed near-optimal and that the lowest mass feasible design is within the range of $1.045 < \frac{\Delta_m}{m_opt} \leq 1.049$.

6.4.2 Feasible Stacking Sequences for an Unblended Target

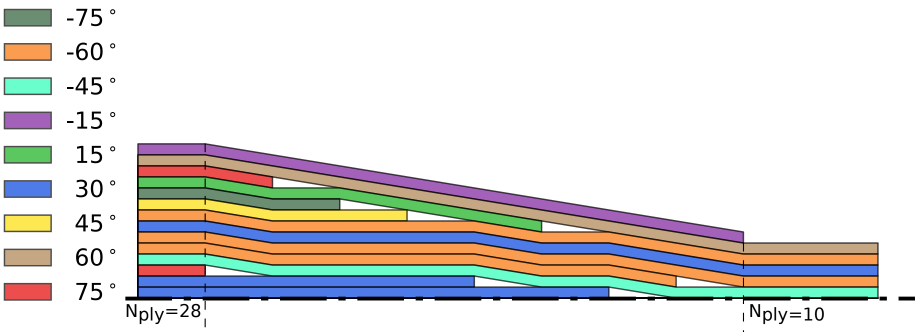
The same discrete optimization method is now used to retrieve stacking sequences of the same aircraft wing structure with respect to a reference continuous optimum obtained without constraining the design space with blending constraints. This is considerably more challenging than the blended case due to the fact patches are individually optimized for the best global performance. As a result, severe changes in stiffness direction and thickness can occur between



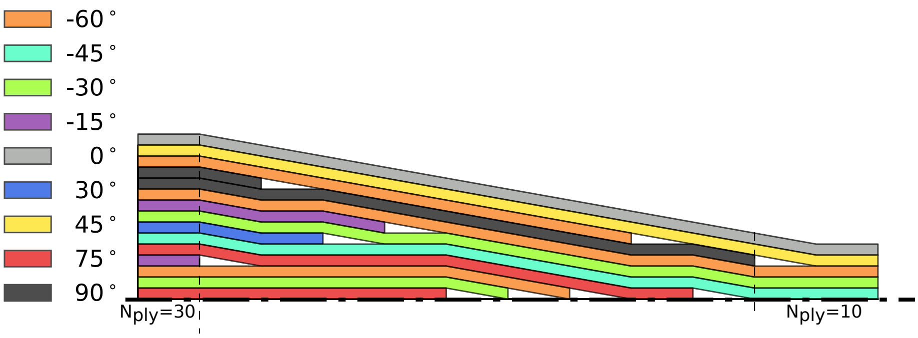
(a) UW



(b) LW



(c) FS



(d) RS

Figure 6.3: Feasible stacking sequence tables of Solution 1. Note that some stacks are non-coding and thus not included in the retrieved design.

Table 6.1: SST Genotype of a feasible design (Solution 1, $m = 1.049 m_{opt}$). The (+x) numbers in the N_{str} chromosome represent the additional number of plies in a patch over the target optimum.

Upper Wing Skin

SST_{lam}	[45	45	60	90	60	-30	30	60	60	-60	-75	60	15	60	-75	-15	60	-75	-75	-75	75	90	-15	15	$15]_s$
SST_{ins}	[0	0	48	46	0	38	22	0	0	40	0	18	28	42	16	34	20	50	0	26	24	32	36	30	$44]_s$
N_{str}	38	34	28(+**4**)	40(+**2**)	26	40	26(+**2**)	34	24(+**4**)	28	20(+**2**)	22	18(+**2**)	18(+**4**)											

Lower Wing Skin

SST_{lam}	[-45	30	60	-30	-45	60	0	90	75	30	-15	0	-15	-15	60	0	0	-15	0	0	0	-15	$-15]_s$
SST_{ins}	[0	0	0	46	0	0	38	40	44	0	20	34	16	24	42	32	30	0	18	28	36	22	$26]_s$
N_{str}	34(+**2**)	36(+**6**)	24	36(+**2**)	30(+**2**)	38(+**2**)	28	32(+**2**)	26(+**2**)	26	24(+**2**)	24	20	16(+**2**)									

Front Spar

SST_{lam}	[-15	60	75	15	-75	45	-60	30	-60	-60	-45	75	30	$30]_s$
SST_{ins}	[12	0	26	18	24	22	0	0	0	14	0	28	20	$16]_s$
N_{str}	14(+**2**)	14(+**2**)	22(+**4**)	22(+**4**)	18(+**2**)	18(+**4**)	18(+**8**)	16(+**4**)						

Rear Spar

SST_{lam}	[0	45	-60	90	90	-60	-15	-30	30	-45	75	-15	-60	-30	$75]_s$
SST_{ins}	[0	0	16	28	12	0	24	0	26	0	14	30	18	20	$22]_s$
N_{str}	22(+**2**)	16(+**4**)	28(+**10**)	14	20(+**8**)	16(+**4**)	16(+**6**)	16							

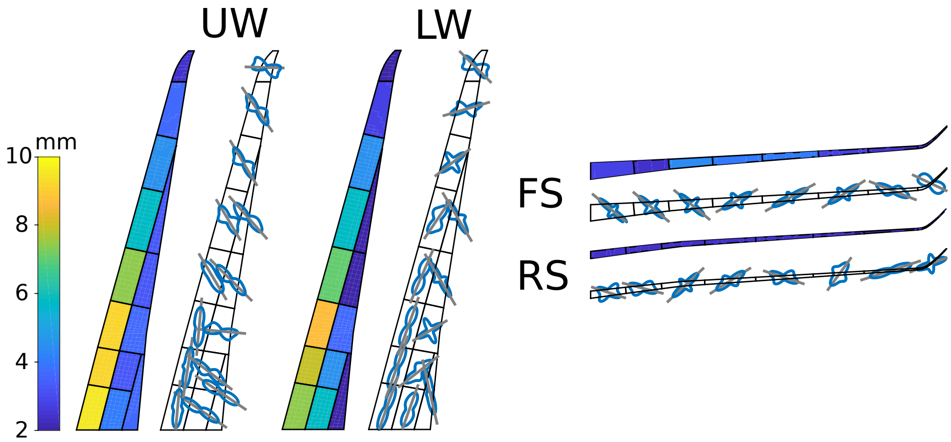


Figure 6.4: Thickness distribution and polar stiffness $E_{11}(\theta)$ of the unblended continuous optimum.

adjacent patches as is seen in the thickness layout and direction provided in figure 6.4. In the UW skin, changes in the main stiffness direction can be as high as 90° between adjacent patches.

In the discrete step, these changes in stiffness direction cannot be captured by the type of guide-based blending used here as changing stiffness direction is restricted by how many plies can be added or removed between adjacent regions. This is exemplified with two E_{1m} polar plots of adjacent UW patches showing a feasible retrieved design and the target optimum, figure 6.5, where even a 36-ply difference is not sufficient to capture the change in direction.

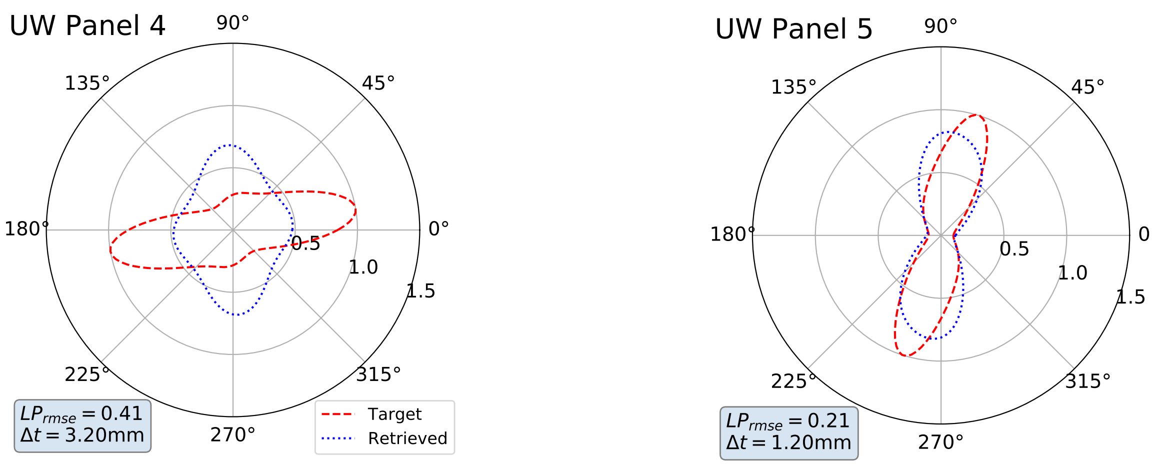


Figure 6.5: E_{1m} polar plots of two adjacent patches in the upper wing. Note the change in target stiffness direction.

For these reasons, attempting to retrieve feasible stacking sequences by matching the LP of a continuous optimum where no manufacturing constraints are enforced is unlikely to yield efficient or even desirable results in terms of extra mass, adherence to design guidelines and mechanical behavior.

Using the same settings as in the previous runs, a feasible design (solution 3) was obtained with a mass increase of 202.7 Kg over the target optimum of 577.49 Kg, an increase of 35%. This is comparable to results obtained by Dillinger [19] where a 42% increase from an unblended continuous target was registered.

6.4.3 Comparison of the Feasible Discrete Solutions

A total of three feasible stacking sequence designs representing the lowest weight gain results are selected for discussion. Solutions 1 and 2 represent the feasible designs retrieved for the blended continuous optimum with $f_{obj,3}$ and $f_{obj,4}$, respectively. Solution 3 represents the retrieved design for the unblended continuous target with $f_{obj,3}$ as the optimization objective. The performance of these designs is evaluated in terms of mass gain over the continuous target, LP_{rmse}, margin of safety of the λ, σ responses and vertical displacement error Δu as summarized in table 6.2. The given margin of safety is the lowest found in the four components for each case and the Δu error is the maximum value found for a given solution.

Table 6.2: Summary of retrieved feasible designs performance.

Solution	Safety Margin	LP_{rmse}				Δu_{max} [cm]	Wing Mass [Kg]
		UW	LW	FS	RS		
1	0.1%	0.191	0.166	0.177	0.153	5.17 (2.4%)	640.15 (▲4.9%)
2	1.9%	0.223	0.263	0.421	0.507	1.35 (0.6%)	640.75 (▲5.0%)
3	0.3%	0.330	0.358	0.355	0.307	7.21 (6.5%)	780.19 (▲35%)

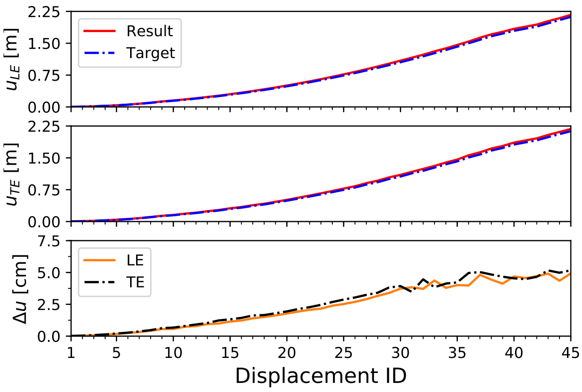
As it turns out, the effect of the displacement error objective $\min(u_{rmse})$ is not as similar to $\min(LP_{rmse})$ as initially assumed. The optimization based on $f_{obj,4}$ (solution 2) resulted in a design of similar mass as $f_{obj,3}$ (solution 1) but with an increase of 1.8% in the margin of safety as well as a lower displacement error of 0.6%. This is not only applicable to the maximum displacement error; the retrieved stacking sequence for solution 2 is a good match with the target optimum along the entire wingspan as seen in figure 6.6 b).

On the other hand, the retrieved LP_{rmse} of solution 2 is considerably higher than solution 1. Since the in-plane membrane stiffness distribution has the most effect on the vertical displacement of the wing, $f_{obj,4}$ steers the GA towards minimizing the error between the V^A parameters over the V^D terms. The RMS error of the UW skin V^A parameters for solution 2 stands at 0.101, whereas its V^D RMS is higher at 0.287. This is further supported by comparison of the V^A and V^D sensitivities of the displacement response where the former is between 2 and 4 orders of magnitude larger. In order to determine whether the retrieved displacement errors impact cruise performance of the wing, further analysis is needed to calculate the resulting wing twist. Due to a lack of data on the wing, the twist verification has not been done in this work.

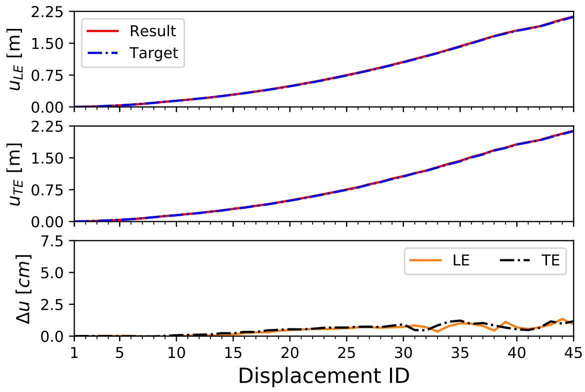
Between solutions 1 and 3, the retrieved stacking sequences for the unblended continuous optimum perform poorly in terms of mass gain, LP_{rmse} and displacement error (figures 6.7, 6.8 and 6.6). By comparing the failure index of the buckling and strength responses given in figures 6.9 and 6.10, it can also be seen that solution 3 is very inefficient, with most of the elements in the wing components being far from failure save for a few strength elements in the lower wing skin. This is especially evident in buckling due to the large increase in patch thickness t_p which affects the out-of-plane stiffness with $t_p^3/12$.

6.4.4 KS Approximation Accuracy

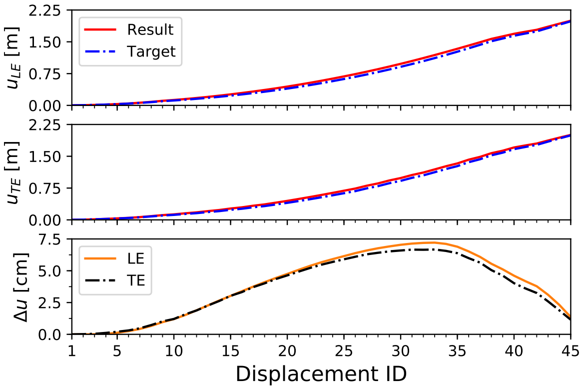
Finally, the KS approximation scheme is evaluated in terms of the error present in the approximations of the retrieved feasible solutions. The error distribution given in figure 6.11 is calculated for all the mechanical constraints selected by the method described in chapter 5, section 5.2. Overall, the strength response approximation error is under 5% and conservative with the exception of the RS, which rarely contains elements near failure. Lowering the ρ factor for the rear spar can push the approximation to be more conservative if necessary.



(a) Solution 1 - min(LP_{rmse}+u_{rmse})



(b) Solution 2 - min(u_{rmse})



(c) Solution 3 - (Unblended target) min(LP_{rmse}+u_{rmse})

Figure 6.6: LE and TE displacement of retrieved feasible solutions.

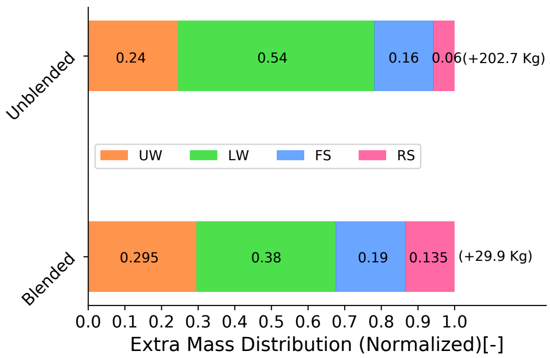


Figure 6.7: Added mass per component in the discrete optimization.

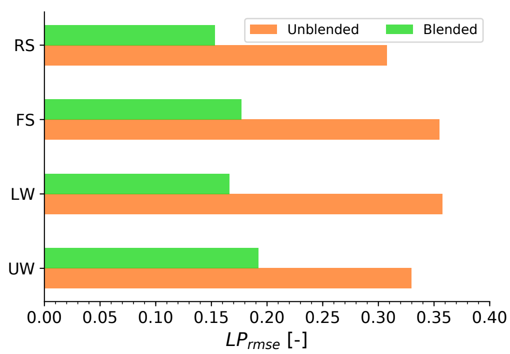
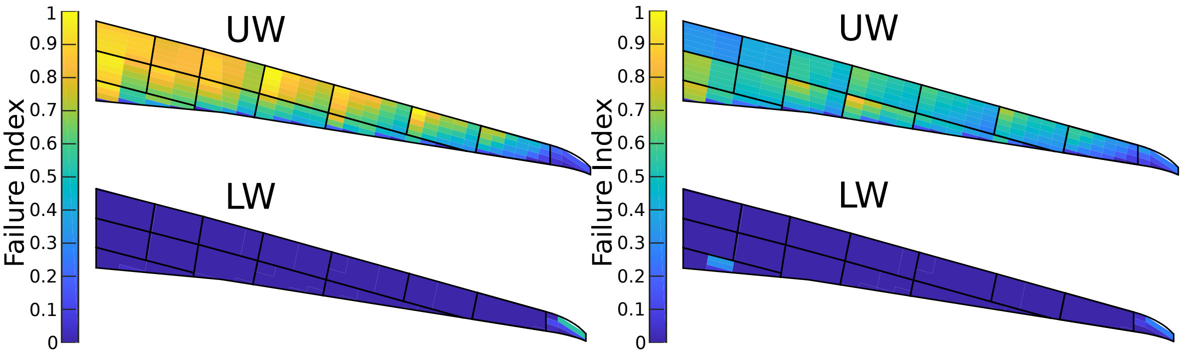
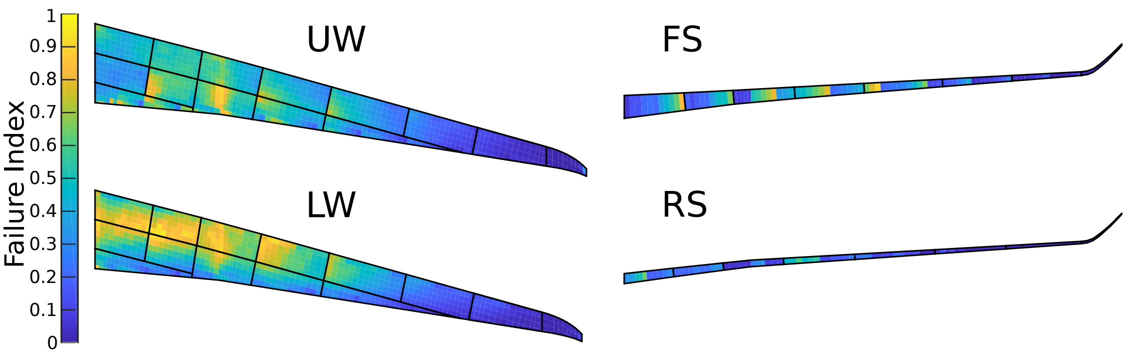


Figure 6.8: LP_{rmse} of all components for the blended (Solution 1) and unblended target.

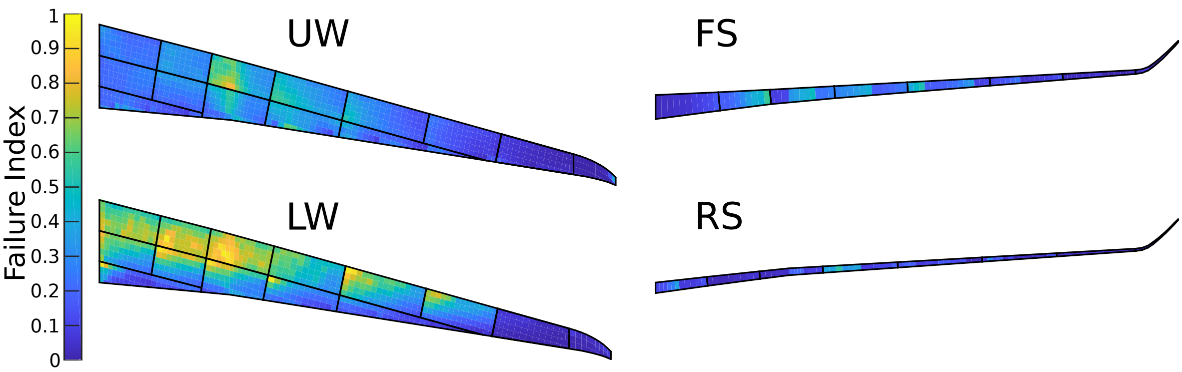


(a) Solution 1 (Blended Target) (b) Solution 3 (Unblended Target)

Figure 6.9: LC3 Buckling Failure Index of Retrieved Stacking Sequences for the blended and unblended continuous optimum.



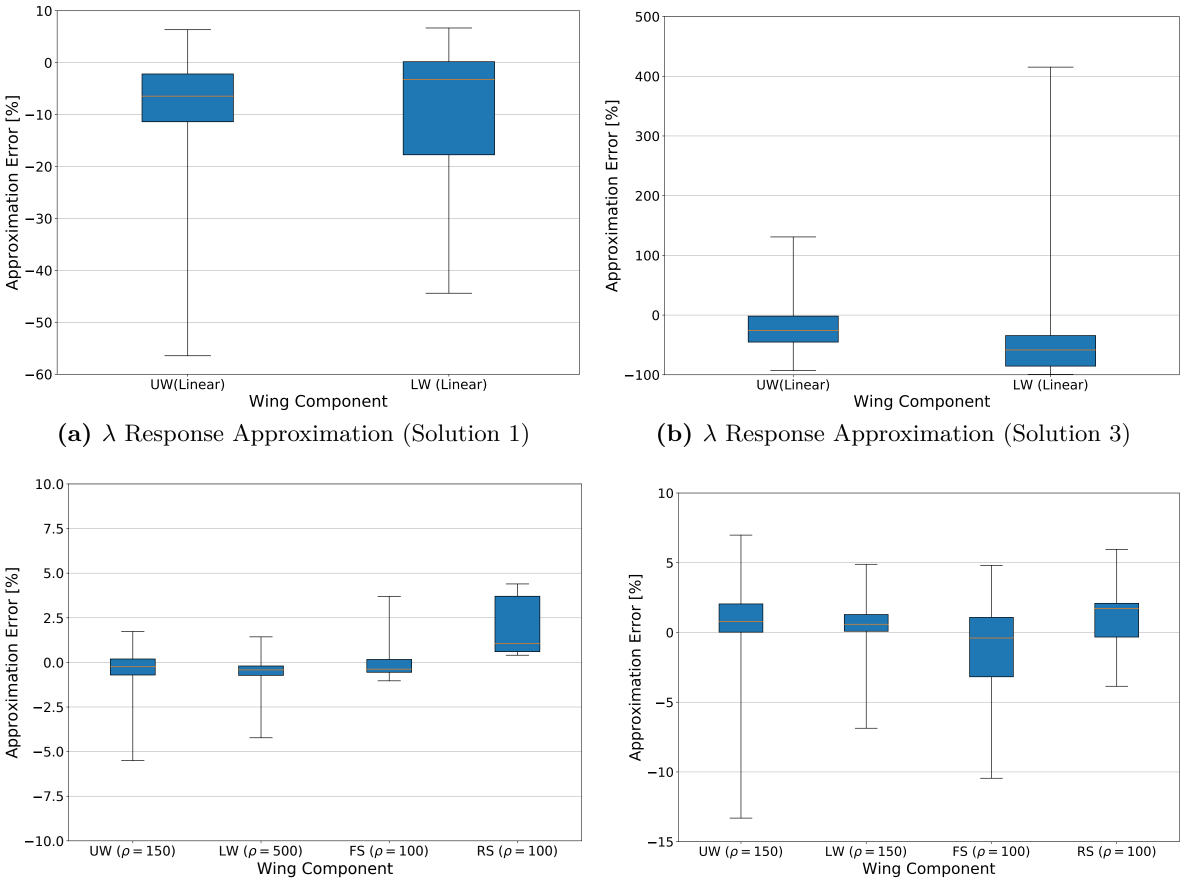
(a) Solution 1 (Blended Target)



(b) Solution 3 (Unblended Target)

Figure 6.10: LC3 Strength Failure Index of Retrieved Stacking Sequences for the blended and unblended continuous optimum.

As expected, the strength approximation error is considerably larger for solution 3, which is based on the unblended continuous optimum and thus suffers from a nearly twice larger LP_{rmse} error as shown in figure 6.8.



(a) λ Response Approximation (Solution 1)
(b) λ Response Approximation (Solution 3)
(c) σ Response Approximation (Solution 1)
(d) σ Response Approximation (Solution 3)

Figure 6.11: Approximation Error of the two solutions.

Although the maximum error of the buckling approximation is very large, this does not actually pose a problem in the optimization because the approximation error for buckling responses near the failure threshold is much more accurate. The error tends to increase only for responses far from failure. This is likely to be related to the fact that the constraints near failure play an active role in the fitness of the designs evaluated by the GA, which in turn leads to a reduction in the error of the relevant LP for that response and an improvement in the approximation since it is being built nearer to its reference point.

For solution 1, the buckling approximation error for responses at or above a failure index of 0.9 is between -2% and 1% for the UW and LW skins and -5% to -0.1% for solution 2. For solution 3, buckling responses at or above a failure index of 0.6 are approximated with an error of $\approx -0.2\%$. A lower failure index threshold is used here due to the fact that no buckling responses had a failure index ≥ 0.9 in solution 3.

An additional set of discrete runs were made with a single objective optimization $min(\Delta m)$ without trying to match the LP of the continuous optimum. No feasible solutions were retrieved due to the GA being steered to regions in the design space where the mechanical constraints are not approximated accurately.

This was verified by using a distance measure in stiffness space proposed by Irisarri et al. [42] which provides a measure of the distance between the [A] and [D] matrices of two laminates referred to as d_{AD}. Solutions obtained from the blended target with a min(LP_{rmse}) objective displayed a two to three times lower stiffness distance from the continuous target than those obtained with a min(Δm) objective. Between the three solutions discussed here, solution 1 has the lowest stiffness distance $d_{AD} = 0.054$, whereas solution 3 has the highest value of $d_{AD} = 0.290$.

Where the runs subjected solely to $min(\Delta m)$ are concerned, discrete results were obtained with 0 active constraints according to the response evaluation in the GA, but subsequent FEA showed widespread failure in strength, implying that the KS approximation method relies heavily on being in the vicinity of the reference points. Another weakness in this method is that there is no built-in convergence to ensure that with a growing number of reference runs the approximation error improves and that the response values don't need to be evaluated by subsequent FE analysis.

Chapter 7

Conclusion and Recommendations

Three core goals guided the work described in this report. Firstly, an open-source discrete optimization toolbox for composite structures named pyTLO was created in order to test objective functions based on LP-matching. This tool is based on the SST method which scales well for large composite structures due to its compact encoding and practical laminate layout that makes the implementation of manufacturing constraints in the stacking sequence design trivial. The computational performance of pyTLO has been improved over similar toolboxes, with a 18 panel horseshoe optimization problem taking as little as 5 to 15 minutes depending on which manufacturing constraints are enforced. Other improvements include the ability to carry out multi-component optimization that factors in response sensitivities due to changes in stiffness distribution across the components.

Secondly, improvements in the retrieval of feasible stacking sequences for the ONERA regional wing model were made by using lamination parameter matching objective functions subjected to strength and buckling constraints. Here, approximations of the structural responses were used to keep the computational costs of evaluation thousands of designs at a minimum. A novel local approximation method which relies on building a KS envelope function of linear approximations built on multiple reference design points was used. The approximation is sufficiently accurate for the strength response, with median errors around 1% for the UW and LW skins and up to 10% for the spars. More importantly, the approximations became conservative when compared to simple linear approximations.

Subjecting a LP-matching objective to mechanical constraints resulted in a clear improvement of the retrieved stacking sequences. At a small increase in the LP error over the continuous target, the number and intensity of active constraints were reduced considerably and it can be determined that the use of constraints had a clear effect on weighting the LP during the optimization according to their impact on the response. The use of blending constraints in the continuous optimization played an important role in the effectiveness of matching LP as an objective; the more realistic stiffness distribution of the continuous optimum made it possible for relatively low LP_{rmse} discrete optima to be obtained, which in turn improves the response approximation.

Finally, feasible stacking sequences were retrieved by allowing thickness to increase over the distribution determined in the continuous optimization. An ϵ- constraint method was used to handle the multi-objective optimization where the increase in mass Δm is used as a constraint and LP_{rmse} is minimized. By incrementally increasing the Δm constraint, feasible stacking sequences were obtained for the reference aircraft wing with a mass increase of 4.9% over the continuous optimum of 610.25 Kg. Another feasible design was obtained using a vertical displacement matching objective (u_{rmse}) instead of LP_{rmse} which resulted in a 5% increase in mass but with a higher margin of safety in the structural responses and a better match of the wing shape under aerodynamic loading with only a 0.6% error in the maximum displacement. A third feasible design was retrieved using a continuous optimum that did not include blending constraints with a mass increase of 35% over the continuous optimum of 577.49 Kg. It can be concluded that using blending constraints in the continuous optimization step led to a considerable improvement in the retrieval of stacking sequence in subsequent discrete optimization by providing a continuous target with a more streamlined stiffness and thickness distribution throughout the wing. In all, an 18 % reduction in weight of retrieved feasible stacking sequences was achieved when blending constraints are used in the continuous optimization.

In order to bring out the full potential of multi-step optimization strategies in the design of composite structures, a number of items in both the discrete and continuous steps require additional attention. These are left here as recommendations for future reference. In what concerns the continuous step, the two recommendations are targeted at increasing the similarity between the feasible design regions of the two steps so that retrieval of equivalent laminates is improved.

1. Additional research on the selection of blending constraint coefficients would be beneficial in determining recommended constraint bounds that lead to an improvement in stacking sequence matching without compromising the search of the continuous design space.
2. In addition to blending constraints, formulating other manufacturing constraints in LP-space could further mitigate the mismatch between the continuous and discrete design space. This would be especially beneficial in the design of more damage tolerant structures if guidelines such as the 10% rule are included in the continuous step.

Finally, four recommendations are made with regards to the discrete step of the optimization that could lead to improvements in computational performance and the retrieval of equivalent stacking sequences.

1. Better blending algorithms: Although SST is among the best guide-based blending methods available, it still constrains the discrete space considerably by requiring that equal thickness patches have the same stacking sequence. Improving the flexibility of blending algorithms without compromising computational performance or manufacturability of retrieved solutions can considerably improve stacking sequence retrieval through better matching of parametrized optima.
2. Constraint Selection: Thousands of mechanical constraints are considered in the design and optimization of large composite structures. The evaluation of these constraints even when they are not design critical can become expensive in the discrete step due to the large number of designs that are evaluated by evolutionary algorithms. Sampling the vicinity of the continuous optimum and building local approximations to determine the critical constraints can translate to considerable computational cost reduction without impacting the optimization process.
3. As a follow-up of 3), additional constraints can be considered in the discrete optimization of aircraft wings to account for aeroelastic responses and ensure that retrieved designs don't require extensive repair or further optimization in the search of feasible stacking sequences.
4. Approximations more accurate than the KS method are needed to improve the success rate of the discrete step. Defining a sampling framework to cover the vicinity of a continuous optimum that can be used to build accurate local approximations based on selection of the nearest sampled reference point would improve the search of the design space without constraining it.

References

[1] Terence Macquart, Noud Werter, and Roeland De Breuker. Aeroelastic design of blended composite structures using lamination parameters. *Journal of Aircraft*, 54(2):561–571, 2016.

[2] F-X Irisarri, Alexis Lasseigne, F-H Leroy, and Rodolphe Le Riche. Optimal design of laminated composite structures with ply drops using stacking sequence tables. *Composite Structures*, 107:559–569, 2014.

[3] David B Adams, Layne T Watson, Zafer Gürdal, and Christine M Anderson-Cook. Genetic algorithm optimization and blending of composite laminates by locally reducing laminate thickness. *Advances in Engineering Software*, 35(1):35–43, 2004.

[4] Kuen Y Lin and Andrey V Styuart. Probabilistic approach to damage tolerance design of aircraft composite structures. *Journal of Aircraft*, 44(4):1309–1317, 2007.

[5] Stephen W Tsai and Nicholas J Pagano. Invariant properties of composite materials. Technical report, Air Force Materials Lab Wright-Patterson AFB Ohio, 1968.

[6] JL Grenestedt and Peter Gudmundson. Layup optimization of composite material structures. *Optimal design with advanced materials*, pages 311–336, 1993.

[7] Terence Macquart, Marco T Bordogna, Paul Lancelot, and Roeland De Breuker. Derivation and application of blending constraints in lamination parameter space for composite optimisation. *Composite Structures*, 135:224–235, 2016.

[8] Hossein Ghiasi, Damiano Pasini, and Larry Lessard. Optimum stacking sequence design of composite materials part i: Constant stiffness design. *Composite Structures*, 90(1):1 – 11, 2009.

[9] Hossein Ghiasi, Kazem Fayazbakhsh, Damiano Pasini, and Larry Lessard. Optimum stacking sequence design of composite materials part ii: Variable stiffness design. *Composite Structures*, 93(1):1–13, 2010.

[10] Shahriar Setoodeh, Mostafa M Abdalla, and Zafer Gürdal. Design of variable–stiffness laminates using lamination parameters. *Composites Part B: Engineering*, 37(4-5):301–309, 2006.

[11] Birna P Kristinsdottir, Zelda B Zabinsky, Mark E Tuttle, and Sudipto Neogi. Optimal design of large composite panels with varying loads. *Composite Structures*, 51(1):93–102, 2001.

[12] Timothy R Brooks, Joaquim RRA Martins, and Graeme J Kennedy. High-fidelity aerostructural optimization of tow-steered composite wings. *Journal of Fluids and Structures*, 88:122–147, 2019.

[13] Jacob Foldager, JS Hansen, and Niels Olhoff. A general approach forcing convexity of ply angle optimization in composite laminates. *Structural Optimization*, 16(2-3):201–211, 1998.

[14] Shahriar Setoodeh, Mostafa Abdalla, and Zafer Gurdal. Approximate feasible regions for lamination parameters. In *11th AIAA/ISSMO multidisciplinary analysis and optimization conference*, page 6973, 2006.

[15] Mitsunori Miki and Yoshihiko Sugiyama. Optimum design of laminated composite plates using lamination parameters. In *32nd Structures, Structural Dynamics, and Materials Conference*, page 971, 1991.

[16] Akira Todoroki and Masafumi Sasai. Stacking sequence optimizations using ga with zoomed response surface on lamination parameters. *Advanced Composite Materials*, 11(3):299–318, 2002.

[17] JA Bailie, RP Ley, and A Pasricha. A summary and review of composite laminate design guidelines. *Langley RC, Hampton*, 1997.

[18] Michael Chun-Yung Niu. *Composite airframe structures: practical design information and data.* Adaso Adastra Engineering Center, 1992.

[19] Johannes Karl Stefan Dillinger. *Static aeroelastic optimization of composite wings with variable stiffness laminates.* PhD thesis, Delft University of Technology, 2014.

[20] Mostafa Abdalla, Zafer Gurdal, and Christos Kassapoglu. Formulation of composite laminate robustness constraint in lamination parameters space. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 17th AIAA/ASME/AHS Adaptive Structures Conference 11th AIAA No*, page 2478, 2009.

[21] CY Poon, C Ruiz, and CB Allen. Finite element analysis of a tapered composite. *Composites science and technology*, 51(3):429–440, 1994.

[22] James M Curry, Eric R Ohnson, and James H Tarnes. Effect of dropped plies on the strength of graphite-epoxy laminates. *AIAA journal*, 30(2):449–456, 1992.

[23] Boyang Liu and Raphael Haftka. Composite wing structural design optimization with continuity constraints. In *19th AIAA Applied Aerodynamics Conference*, page 1205, 2001.

[24] Grant Soremekun, Zafer Gürdal, Christos Kassapoglou, and Darryl Toni. Stacking sequence blending of multiple composite laminates using genetic algorithms. *Composite Structures*, 56(1):53–62, 2002.

[25] Julien van Campen, Omprakash Seresta, Mostafa Abdalla, and Zafer Gürdal. General blending definitions for stacking sequence design of composite laminate structures. In *49th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, 16th AIAA/ASME/AHS Adaptive Structures Conference, 10th AIAA Non-Deterministic Approaches Conference, 9th AIAA Gossamer Spacecraft Forum, 4th AIAA Multidisciplinary Design Optimization Specialists Conference*, page 1798, 2008.

[26] Marco Bordogna, Paul Lancelot, Dimitri Bettebghor, and Roeland De Breuker. Aeroelastic tailoring for static and dynamic loads with blending constraints. In *Proceedings from International Forum on Aeroelasticity and Structural Dynamics, Como, Italy*. International Forum on Aeroelasticity and Structural Dynamics (IFASD), 2017.

[27] Gustavo HC Silva, Alex Pereira do Prado, Pedro Higino Cabral, Roeland De Breuker, and Johannes KS Dillinger. Tailoring of a composite regional jet wing using the slice and swap method. *Journal of Aircraft*, pages 1–15, 2018.

[28] Terence Macquart. Future research in multi-step composite optimisation. In *2016 International Conference Multidisciplinary Engineering Design Optimization (MEDO)*, pages 1–5. IEEE, 2016.

[29] Alban Carpentier, Laurent Michel, Stéphane Grihon, and Jean-Jacques Barrau. Optimization methodology of composite panels. In *12th European Conference on Composite Materials (ECCM 12)*, pages 1–8, September 2006.

[30] S Meldrum, Ludovic Colo, Gabriel Broux, and Eric Garrigues. A new dassault industrial approach for aero-structural optimization of composite structures with stacking table constraints. In *Proceedings from International Forum on Aeroelasticity and Structural Dynamics, Como, Italy*, 2017.

[31] Zhao Jing, Xueling Fan, and Qin Sun. Global shared-layer blending method for stacking sequence optimization design and blending of composite structures. *Composites Part B: Engineering*, 69:181–190, 2015.

[32] Zhao Jing, Qin Sun, and Vadim V Silberschmidt. A framework for design and optimization of tapered composite structures. part i: From individual panel to global blending structure. *Composite Structures*, 154:106–128, 2016.

[33] Zhao Jing, Qin Sun, Jian Qiao Chen, and Vadim V Silberschmidt. A framework for design and optimization of tapered composite structures. part ii: Enhanced design framework with a global blending model. *Composite Structures*, 188:531–552, 2018.

[34] Ellen van den Oord. Overcoming the curse of dimensionality in composite laminate blending. Master's thesis, Delft University of Technology, The Netherlands, 2018.

[35] J-FM Barthelemy and Raphael T Haftka. Approximation concepts for optimum structural design—a review. *Structural optimization*, 5(3):129–144, 1993.

[36] ST IJsselmuiden. *Optimal Design of Variable Stiffness Composite Structures Using lamination Parameters*. PhD thesis, Delft University of Technology, 2011.

[37] G. Kreisselmeier and R. Steinhauser. Systematic control design by optimizing a vector performance index. *IFAC Proceedings Volumes*, 12(7):113 – 117, 1979. IFAC Symposium on computer Aided Design of Control Systems, Zurich, Switzerland, 29-31 August.

[38] Alexis Lasseigne. *Optimization of variable-thickness composite structures. Application to a CROR blade.* PhD thesis, Lyon, 2016.

[39] Samuel T IJsselmuiden, Mostafa M Abdalla, Omprakash Seresta, and Zafer Gürdal. Multi-step blended stacking sequence design of panel assemblies with buckling constraints. *Composites part b: engineering*, 40(4):329–336, 2009.

[40] Abdullah Konak, David W Coit, and Alice E Smith. Multi-objective optimization using genetic algorithms: A tutorial. *Reliability Engineering & System Safety*, 91(9):992–1007, 2006.

[41] George Mavrotas. Effective implementation of the ε-constraint method in multi-objective mathematical programming problems. *Applied mathematics and computation*, 213(2):455–465, 2009.

[42] Francois-Xavier Irisarri, Mostafa M Abdalla, and Zafer Gürdal. Improved shepard's method for the optimization of composite structures. *AIAA journal*, 49(12):2726–2736, 2011.

Appendix A

Appendix: Polar Plots ($E_{1m}(\theta)$) of Feasible Stacking Sequences

In this appendix the membrane stiffness E_{11} polar plots of the three solutions discussed in chapter 6 are provided. Section A.1 contains the polar plots of Solution 1 which was retrieved from the blended target of the ONERA regional wing model with the following objective function: $f_{obj} = min(LP_{rmse} + u_{rmse} + C(\sigma, \lambda, m))$.

In section A.2 the polar plots of Solution 2 are given. This solution was retrieved from the blended target with a displacement matching objective, i.e., $f_{obj} = min(u_{rmse} + C(\sigma, \lambda, m))$.

Finally, in section A.3 the polar plots of Solution 3 are provided. This solution was retrieved from the unblended target with the following objective function $f_{obj} = min(LP_{rmse} + u_{rmse} + C(\sigma, \lambda, m))$. In all plots, the straight lines represent the main stiffness direction of the target (red) and the retrieved discrete stacking sequences (blue).

A.1 Polar plots of Solution 1

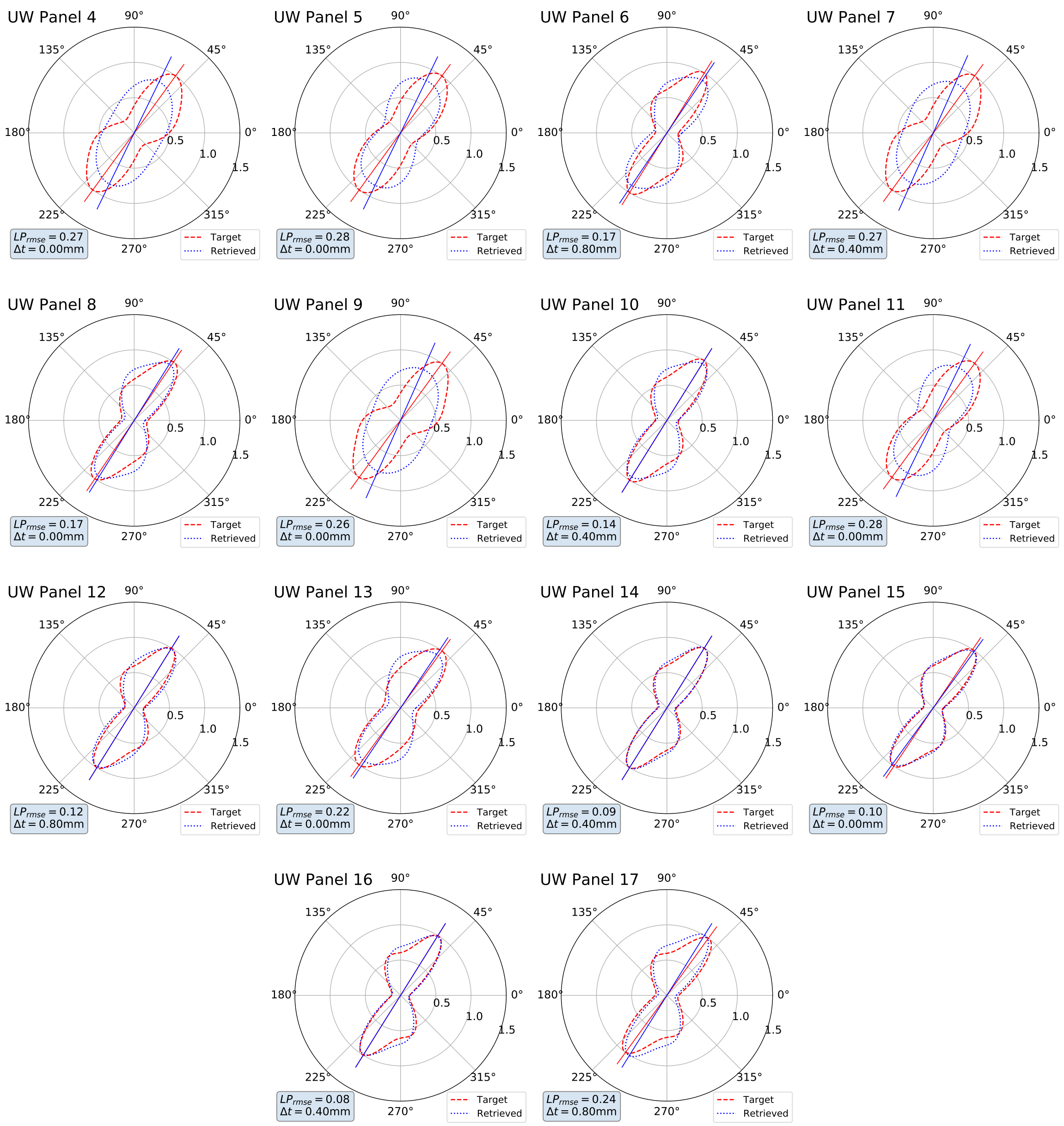


Figure A.1: Retrieved UW Polar stiffness E_{1m} of Solution 1 (blended target). Normalized according to target polar stiffness.

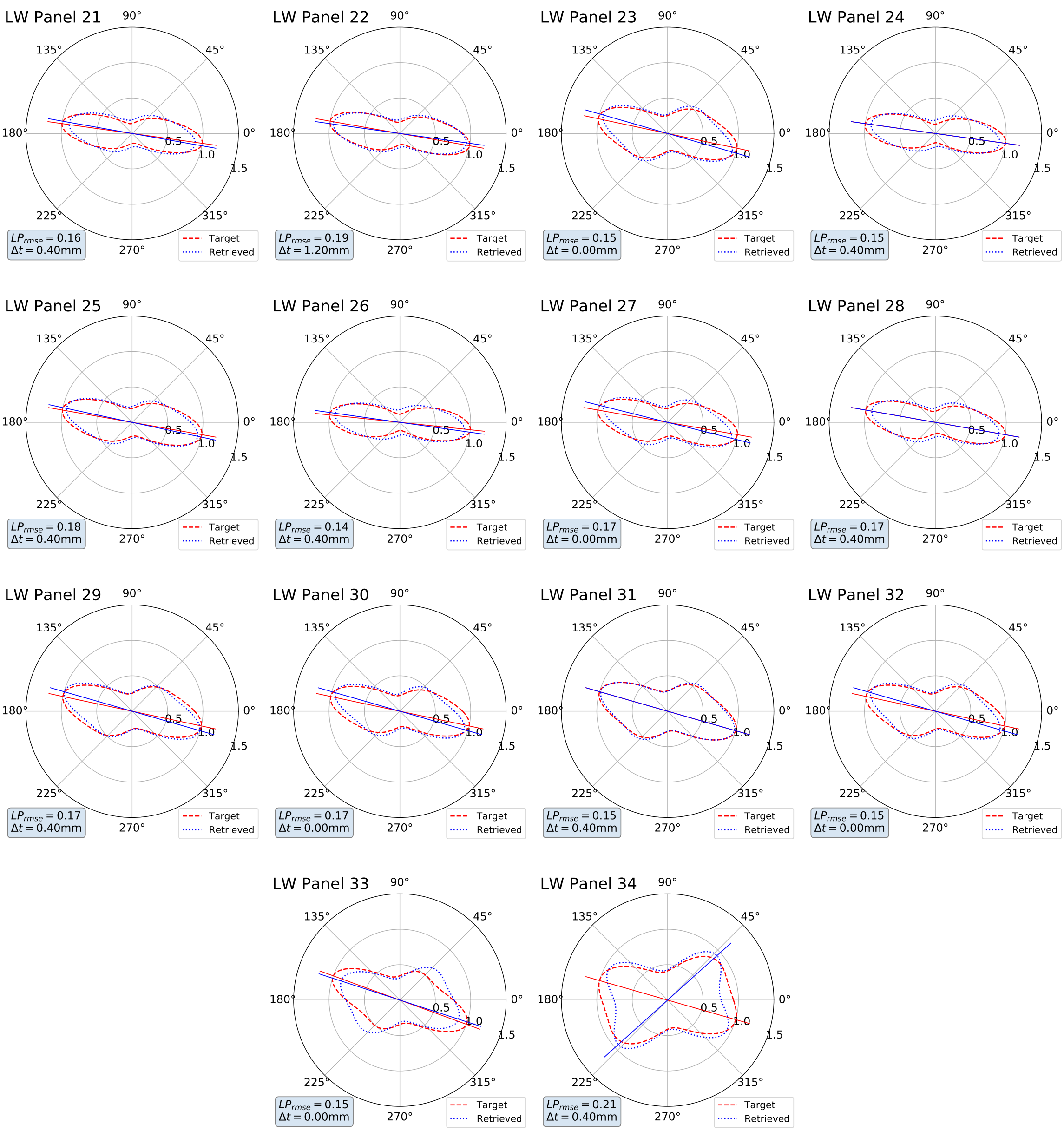


Figure A.2: Retrieved LW Polar stiffness E_{1m} of Solution 1 (blended target). Normalized according to target polar stiffness.

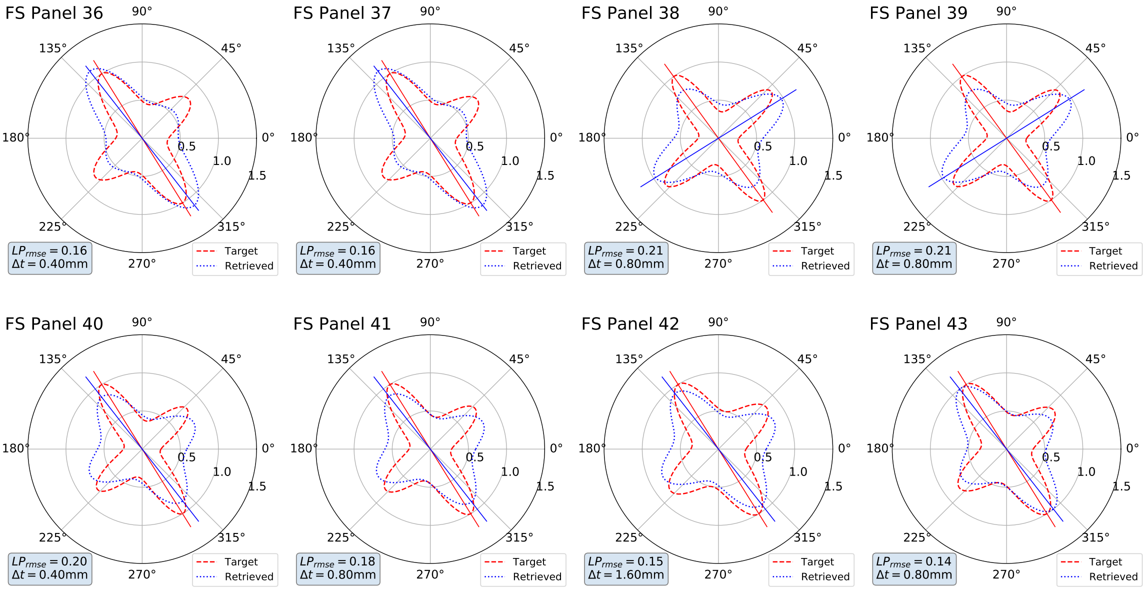


Figure A.3: Retrieved FS Polar stiffness E_{1m} of Solution 1 (blended target). Normalized according to target polar stiffness.

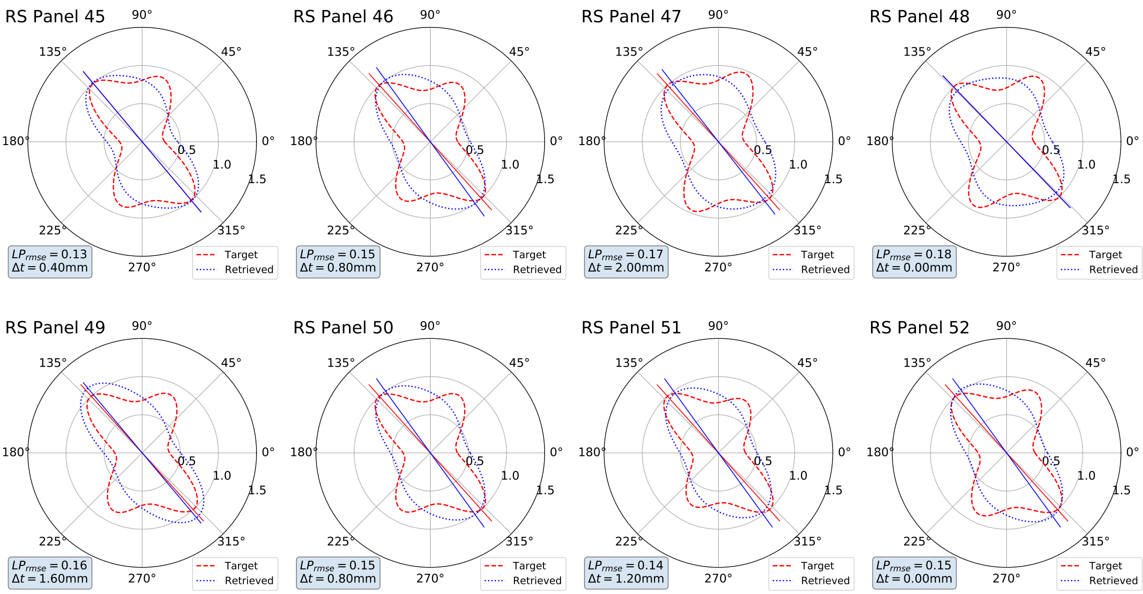


Figure A.4: Retrieved FS Polar stiffness E_{1m} of Solution 1 (blended target). Normalized according to target polar stiffness.

A.2 Polar plots of Solution 2

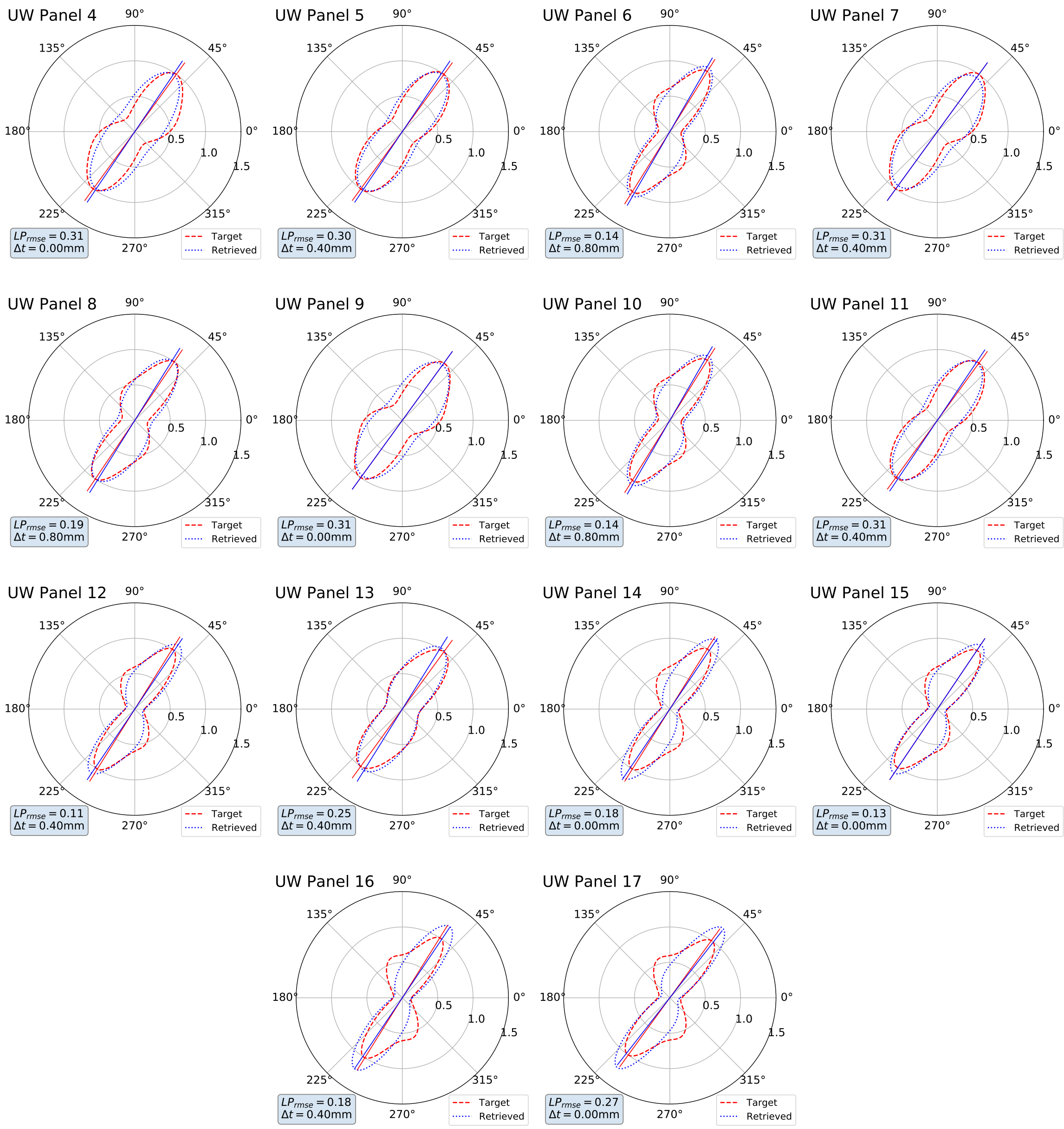


Figure A.5: Retrieved UW Polar stiffness E_{1m} of Solution 2 (blended target, Displacement Matching). Normalized according to target polar stiffness.

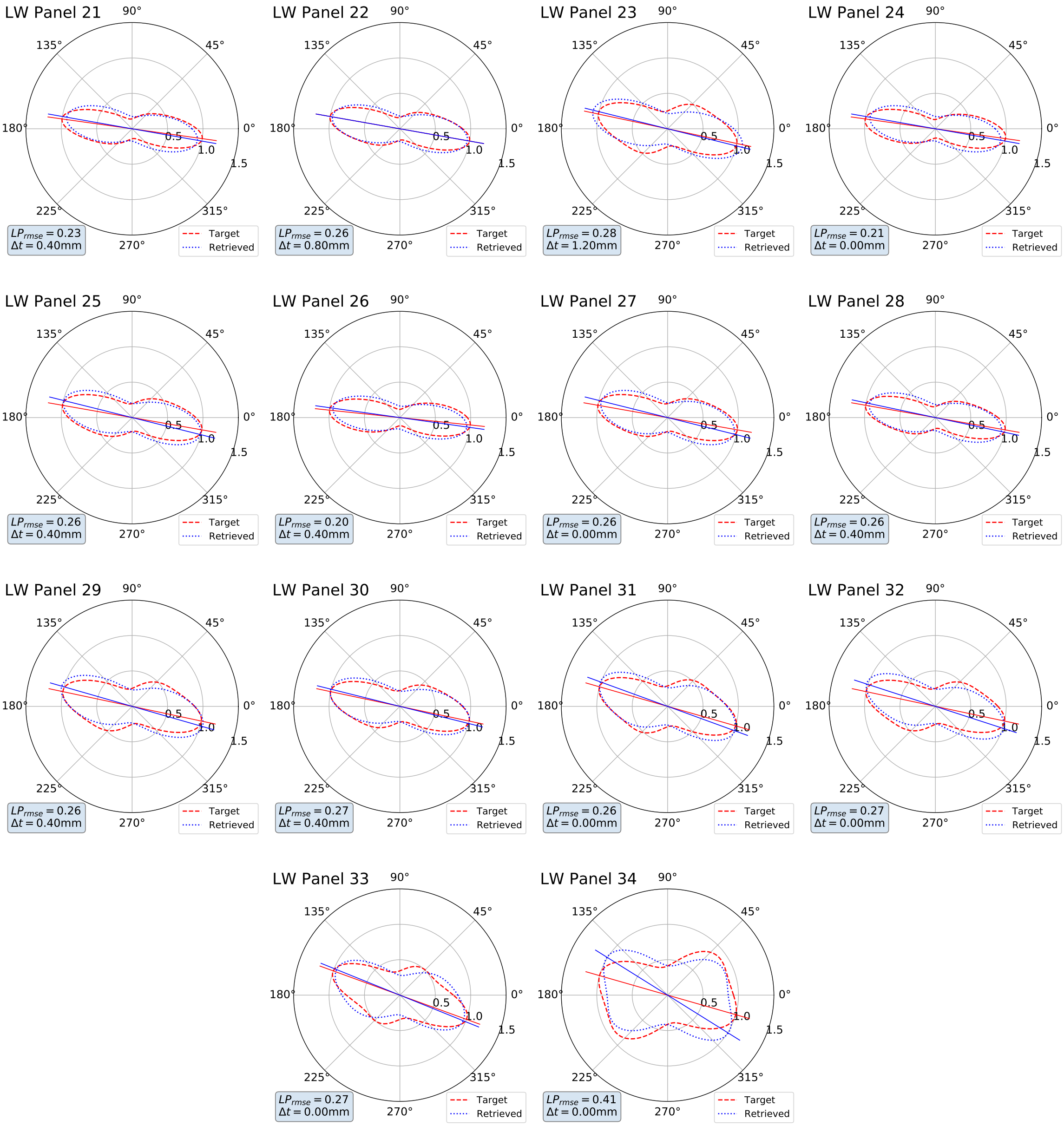


Figure A.6: Retrieved LW Polar stiffness E_{1m} of Solution 2 (blended target, Displacement Matching). Normalized according to target polar stiffness.

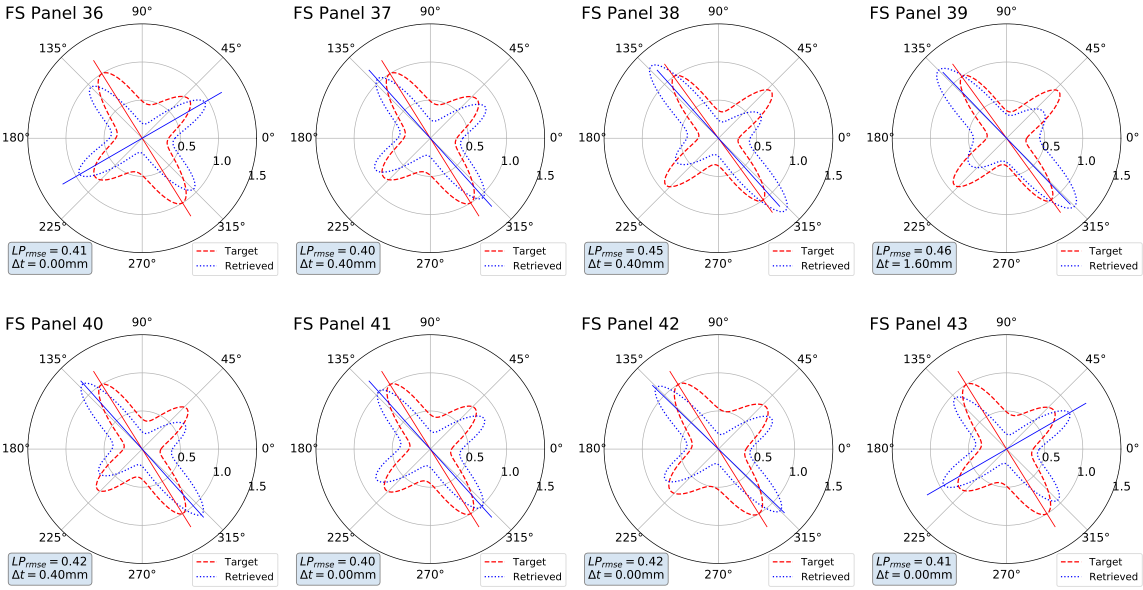


Figure A.7: Retrieved FS Polar stiffness E_{1m} of Solution 2 (blended target, Displacement Matching). Normalized according to target polar stiffness.

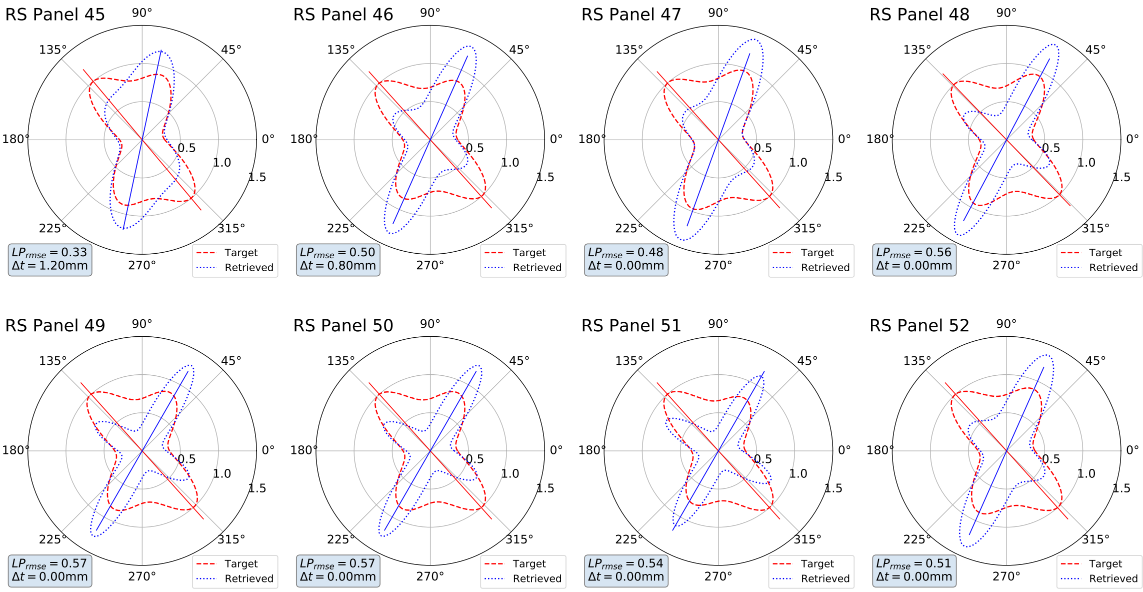


Figure A.8: Retrieved FS Polar stiffness E_{1m} of Solution 2 (blended target, Displacement Matching). Normalized according to target polar stiffness.

A.3 Polar plots of Solution 3

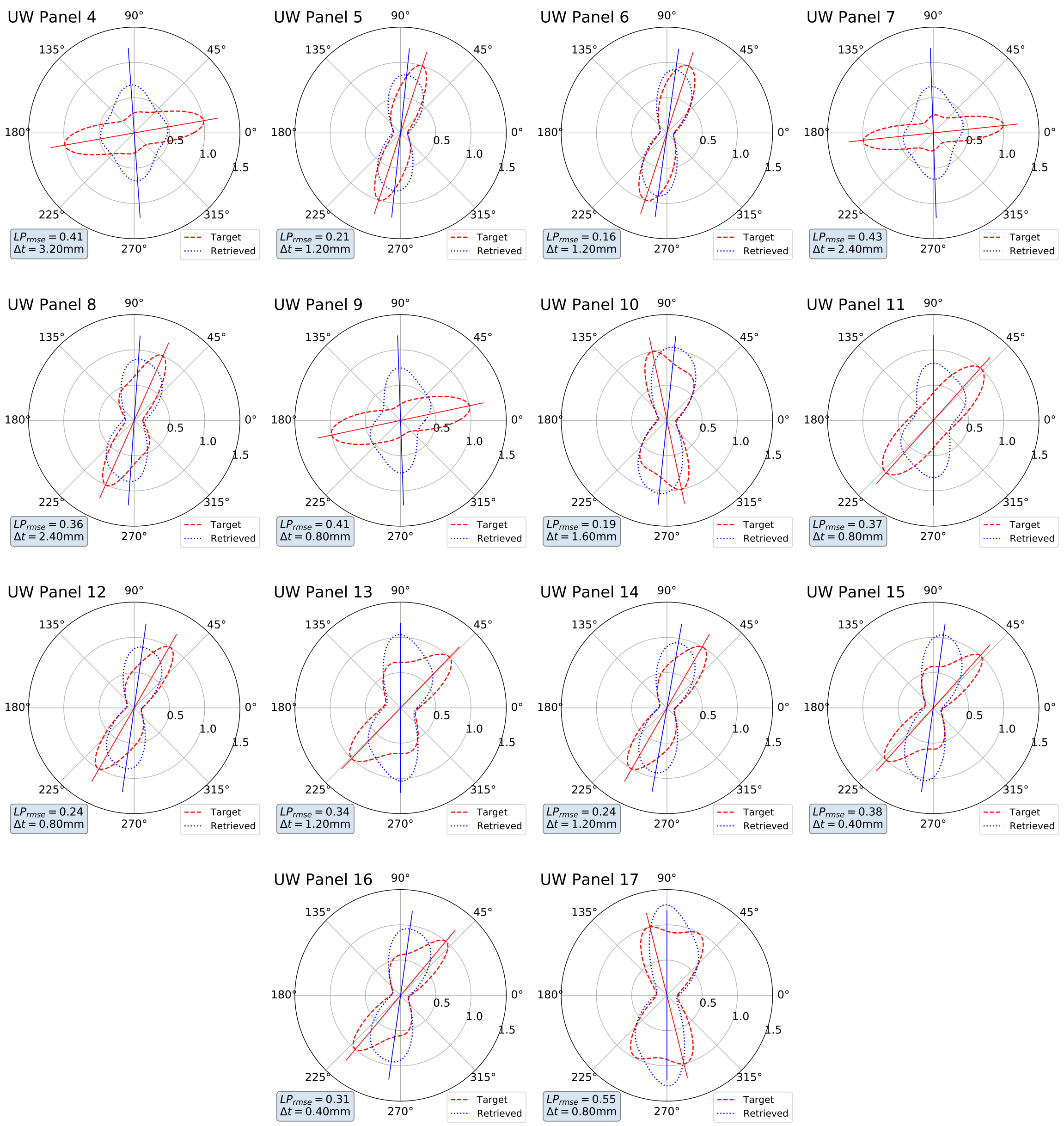


Figure A.9: Retrieved UW Polar stiffness E_{1m} of Solution 3 (unblended target). Normalized according to target polar stiffness.

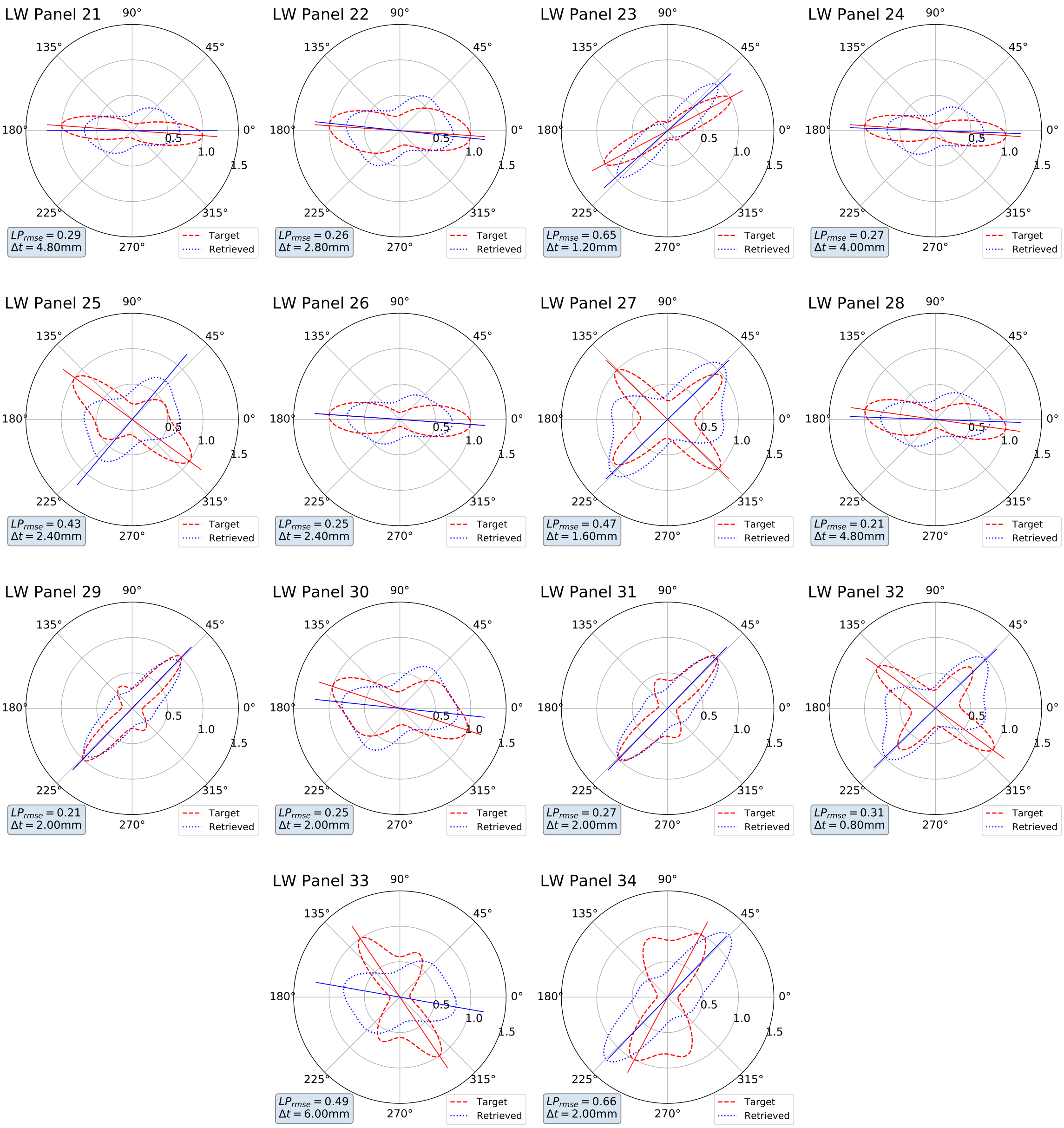


Figure A.10: Retrieved LW Polar stiffness E_{1m} of Solution 3 (unblended target). Normalized according to target polar stiffness.

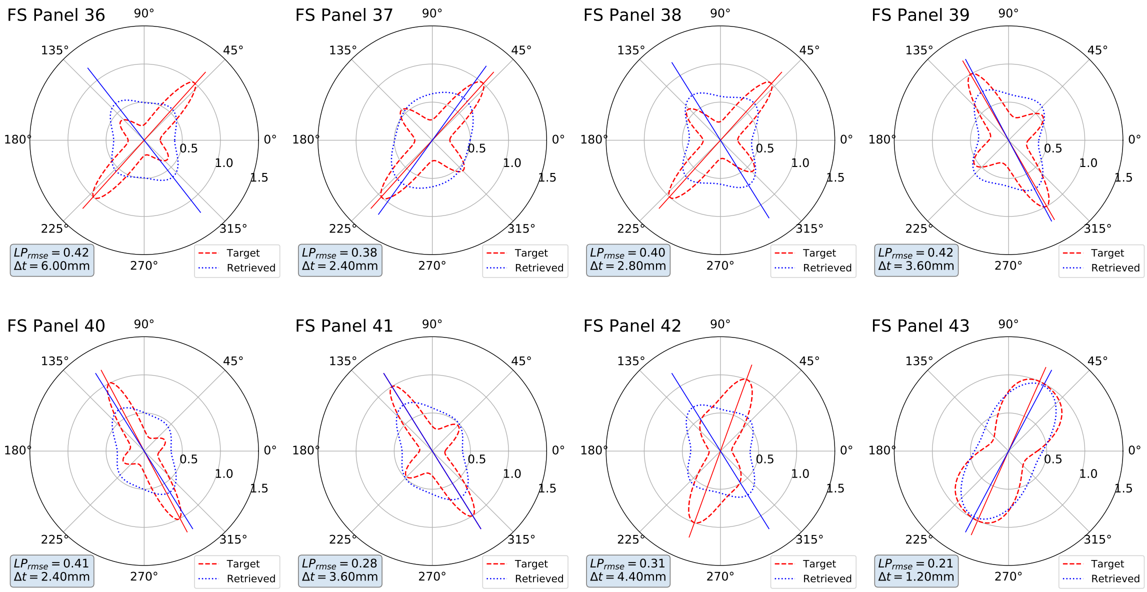


Figure A.11: Retrieved FS Polar stiffness E_{1m} of Solution 3 (unblended target). Normalized according to target polar stiffness.

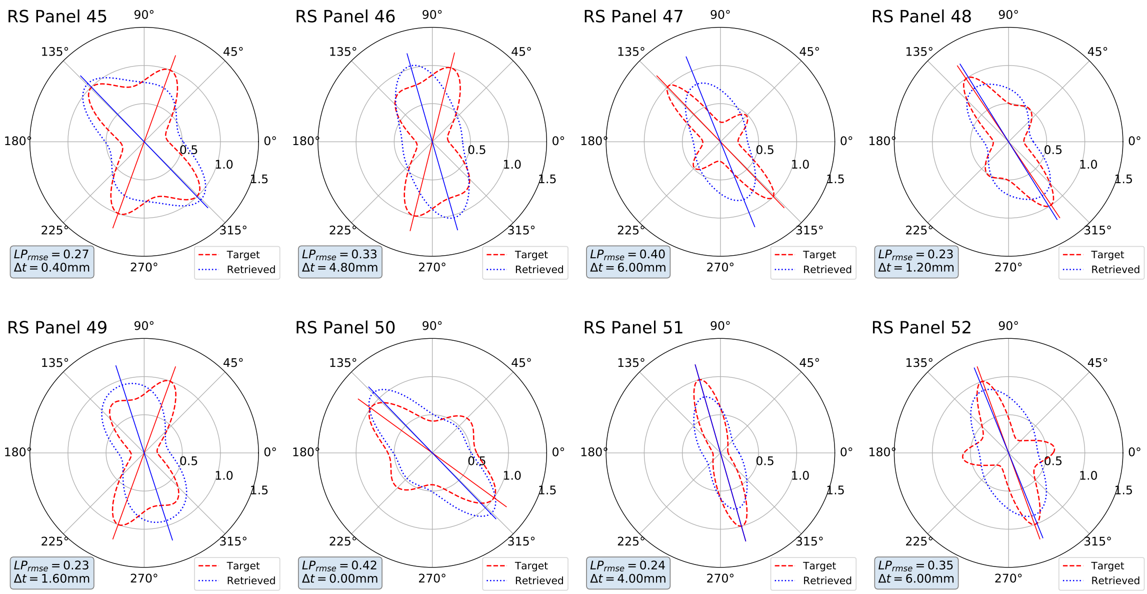


Figure A.12: Retrieved FS Polar stiffness E_{1m} of Solution 3 (unblended target). Normalized according to target polar stiffness.