Master of Science Thesis

Development of surrogate-based multidisciplinary optimization methodology with flutter constraints for aircraft conceptual design

A.C. Lambers BSc

July 22, 2016

Faculty of Aerospace Engineering · Delft University of Technology

Development of surrogate-based multidisciplinary optimization methodology with flutter constraints for aircraft conceptual design

Master of Science Thesis

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

A.C. Lambers BSc

July 22, 2016

Faculty of Aerospace Engineering · Delft University of Technology

Cover picture: Boeing SUGAR concept drawing

Delft University Of Technology
Department Of
Flight Performance and Propulsion

The undersigned hereby certify that they have read and recommend to the Faculty of Aerospace Engineering for acceptance a thesis entitled **"Development of surrogate-based multidisciplinary optimization methodology with flutter constraints for aircraft conceptual design"** by **A.C. Lambers BSc** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dated: July 22, 2016

Chair holder: _____
prof.dr.-Ing. G. Eitelberg

Reader: _____
dr. ir. G. La Rocca

Reader: _____
dr. R.P. Dwight

Supervisor TU Delft: _____
ir. I. van Gent

Supervisor NLR: _____
dr.ir. W.J. Vankan

Summary

Aircraft concepts with high aspect ratio wings have been investigated more extensively in the recent past, as such configurations would reduce the induced drag significantly. At the same time, the high aspect ratio challenges the wings aero-structural stability (flutter). Therefore there is a need to perform aero-structural analyses already within the conceptual design phase, preferably in an automated fashion to enable the execution of multidisciplinary design optimization (MDO). This study has investigated methods to directly incorporate flutter speed predictions as constraints in an automated optimization process for conceptual design of aircraft.

A case study using a strut-braced wing (SBW) was set up to apply the developed methods. The strut reduces the loading on the wing and wing root which enables an increase in aspect ratio. SBW concepts have shown in recent research to enable aspect ratios significantly higher than modern commercial aircraft. This case study in this research has shown an implementation of the flutter speed constraint in the automated optimization. The proposed method has shown to be effective but more case studies should be applied to further develop the method.

The computationally expensive flutter calculations have been modelled using different surrogate models (SM) to reduce calculation time in the automated design process. Combinations of different surrogate and design of experiments (DOE) methods have been investigated and compared. Each combination of SM and DOE provides an indication of the trade-off between computation time and precision.

The *MultiFit* software tool, developed by the Netherlands Aerospace Centre, was used to set up the different SMs. Artificial neural networks have shown the best capabilities in representing the behaviour of the flutter speed. Furthermore, surrogate-based optimization (SBO) methods have been able to find the same optimum design point at lower computational cost compared to the classical (non-surrogate) multidisciplinary feasible (MDF) MDO architecture.

In an attempt to reduce optimization time even further, an adaptive surrogate modelling approach has been incorporated in the surrogate-based optimization method. The proposed method starts with a small DOE to create an initial SM. Then, an SBO is

performed on this model. The value of the resulting optimum is recalculated using the complete multidisciplinary analysis and this new design point is then added to the DOE. A new surrogate model is created and an SBO is performed using the earlier found optimum as a starting point. This process is repeated until the process converges. The case study in this research has shown that adaptive surrogate-based optimization can be an efficient method to find the optimum of the objective function at a low computational cost, although more advanced enrichment and convergence criteria are recommended to improve the algorithm.

Acknowledgements

This thesis is the final work of my master track in Flight Performance and Propulsion and marks the end of my study period in Delft. I would like to thank a number of people for their support during my thesis work.

First of all I would like to thank Jos and Huub from NLR for the opportunity to perform the thesis work at the Netherlands Aerospace Centre in Amsterdam. I would like to thank Huub for helping me with all the hours of debugging AMLoad that we have performed. I am pleased to see that my contributions have led to a more mature version of the tool and of the implementation of batch mode calculations. I would like to thank Jos for all the support during my time in Amsterdam. The Monday morning discussions greatly contributed to the final thesis result. Jos was always available for assess my work and to help me review the results. Second I would like to thank the members of the graduation committee (Gianfranco La Rocca, Georg Eitelberg and Richard Dwight) for the time they took to asses my work and to share their opinions.

The next people I owe thanks to are the fellow students at NLR. Especially Andy and Sebastiaan who have been at NLR for the same period as I have. The coffee brakes and lunch runs have contributed greatly to a successful graduation project.

I would also like to thank my parents Astrid and Frans for their support during the graduation project but also during my entire period as a student. Finally I would like to thank Fanny for her motivation and support throughout the entire project.

Delft, The Netherlands — A.C. Lambers BSc
July 22, 2016

Contents

Summary v

Acknowledgements vii

List of Figures xiv

List of Tables xv

Nomenclature xvii

I Introduction & Background 1

1 Introduction 3
- 1.1 Research question and objective . . . 6
- 1.2 Thesis structure . . . 6
- 1.3 Case study . . . 8

2 Background 11
- 2.1 Strut-braced wing . . . 11
- 2.2 Surrogate-based optimization . . . 13
  - 2.2.1 Surrogate modelling procedure . . . 14
  - 2.2.2 Design of experiments . . . 15
  - 2.2.3 Design of experiments size . . . 18
- 2.3 Surrogate modelling methods . . . 18
  - 2.3.1 Polynomial functions . . . 18
  - 2.3.2 Kriging . . . 19
  - 2.3.3 Artificial Neural Network . . . 19
- 2.4 Adaptive surrogate-based optimization . . . 19

II Disciplinary analyses **23**

3 Static aeroelastic analysis **25**
3.1 Method description . . . 25
3.2 Implementation . . . 26

4 Aerodynamic drag analysis **27**
4.1 Method description . . . 28
4.1.1 Q3D . . . 28
4.1.2 AMLoad . . . 29
4.1.3 Friction . . . 30
4.2 Comparison between drag estimation tools . . . 32
4.3 Implementation . . . 33

5 Range analysis **35**
5.1 Governing equations . . . 35
5.2 Implementation . . . 36

6 Structural analysis **37**
6.1 Governing equations . . . 37
6.1.1 Wing structure . . . 38
6.1.2 Strut structure . . . 40
6.2 Implementation . . . 41

7 Flutter analysis **43**
7.1 Governing equations . . . 44
7.1.1 k-method . . . 45
7.1.2 p-k-method . . . 46
7.1.3 p-method . . . 47
7.2 Implementation . . . 48
7.2.1 Flutter speed calculation in AMLoad . . . 48
7.2.2 Implementation of flutter speed in optimization . . . 51
7.2.3 Flutter mode visualisation . . . 51
7.3 Wing bending mode . . . 52
7.4 Tail flutter mode . . . 53
7.5 Results . . . 54
7.6 Conclusion . . . 55

III Multidisciplinary analysis and surrogate-based optimisation **57**

8 Multidisciplinary design optimization **59**
8.1 Implementation . . . 59
8.2 Results . . . 60
8.3 Comparison with SUGAR . . . 61
8.4 Flutter constraint in the optimization . . . 61

9 Surrogate-based optimisation **67**
9.1 Error determination . . . 67
9.2 Surrogate-based optimization architecture . . . 69
9.3 Surrogate model flutter speed prediction . . . 70
9.4 Surrogate model Von Mises stress . . . 71
9.5 Surrogate model range . . . 71
9.6 Design of experiments results . . . 71
9.7 Prediction error comparison . . . 72
9.8 Adaptive surrogate modelling . . . 73
9.9 Results . . . 76
9.9.1 MDO using 10 starting points . . . 77
9.9.2 Best feasible point in large DOE . . . 77
9.9.3 SBO using different DOEs . . . 77
9.9.4 Adaptive SBO . . . 78
9.9.5 Optimum variables . . . 78

10 Conclusions and recommendations **81**
10.1 Conclusions . . . 81
10.2 Recommendations . . . 83
10.2.1 Improving methods . . . 83
10.2.2 Improve strut-braced wing conceptual design . . . 84

References **87**

A Aerodynamic tools comparison **93**
A.1 Friction validation . . . 93
A.2 NASTRAN vs AVL . . . 94
A.3 Friction versus Q3D . . . 95
A.3.1 Input file Q3D . . . 95
A.3.2 Input file Friction . . . 97

B Mode switch **99**

List of Figures

1.1 Wing weight and induced drag trends versus wing span [1] 4
1.2 XDSM of multidisciplinary analysis 7
1.3 Front view of SUGAR reference aircraft [2] 8
1.4 Schematic wing top view showing the wing design variables 9
1.5 Schematic wing side view showing the wing design variables 10

2.1 Example of a modern strut-braced wing aircraft (Cessna Caravan) 12
2.2 Picture of the strut-braced wing aircraft concept (ALBATROS) developed by ONERA [3] 12
2.3 Example of strut-braced wing aircraft concept developed by Boeing (SUGAR research) 12
2.4 Example of a strut-braced wing and a truss-braced wing [4] 13
2.5 Surrogate modelling steps [5] 14
2.6 Three-dimensional two-level full factorial example 15
2.7 Three-dimensional Box-Behnken example 16
2.8 Two-dimensional Latin Hypercube Sampling example 17
2.9 Three-dimensional Latin Hypercube Sampling example 17
2.10 Combination of Latin Hypercube Sampling, Box-Behnken and two-level Full Factorial designs of experiments 18
2.11 Schematic of an artificial neural network [6] 19
2.12 Adaptive surrogate modelling steps [5] 20
2.13 Surrogate model update example [7] 21

3.1 XDSM of static aeroelastic analysis 25

4.1 XDSM of aerodynamic drag analysis 27

4.2 Overview of drag components [8] 28
4.3 Wave drag [8] 31
4.4 Total drag coefficient versus Mach number for $C_L = 0.2$ 33

5.1 XDSM of range analysis 35

6.1 XDSM of structural analysis 37
6.2 Wing box geometry overview 38
6.3 Schematic wing side view indicating the wing root section as region of interest 38

7.1 XDSM of flutter analysis 44
7.2 p-method versus k-method [9] 47
7.3 p-method versus p-k-method [9] 48
7.4 Flutter plot created by AMLoad 50
7.5 Wing bending mode visualisation 52
7.6 Tail and wing flutter mode visualisation 53
7.7 Flutter speed versus wing span for different strut locations 55

8.1 XDSM of the multidisciplinary design optimization 60
8.2 Flutter constraint not active 63
8.3 Active flutter constraint 64
8.4 Flutter constraint not active 65

9.1 XDSM of the surrogate-based optimization 69
9.2 Range prediction error (RMSE) for different DOE sizes 72
9.3 Prediction error (RMSE) for different DOE sizes 73
9.4 XDSM of the adaptive surrogate-based optimization 74
9.5 Adaptive SBO LHS 25 and 50 76

A.1 Fortran results compared with Matlab results 94
A.2 C_L versus M for AVL and AMLoad 95

B.1 Flutter modes showing tail as critical mode 99
B.2 Flutter modes showing wing as critical mode 100

List of Tables

1.1 Aspect ratios of modern aircraft 5
1.2 XDSM variable definition 7
1.3 SUGAR reference values [2] 9

2.1 Number of Box-Behnken design points [10] 16

4.1 Overview of the tools used in Q3D and AMLoad 28
4.2 Input variables for transonic Fokker 100 wing 33

8.1 Optimization results compared with reference 61
8.2 Optimization with active flutter constraint 62

9.1 XDSM variable definition 70
9.2 Flutter speed surrogate model prediction errors 70
9.3 Von Mises stress surrogate model prediction errors 71
9.4 Range surrogate model prediction errors 71
9.5 Efficiency and quality comparison of different optimization methods . . . 79

Nomenclature

Latin Symbols

AR	Aspect ratio	$[-]$
b	Wing span	$[m]$
C_L	Total lift coefficient	$[-]$
C_T	Specific fuel consumption	$[N/Ns]$
$C_{D,F}$	Friction drag coefficient	$[-]$
C_{D_i}	Induced drag coefficient	$[-]$
C_{Dw}	Total wave drag coefficient	$[-]$
C_{dw}	Local wave drag coefficient	$[-]$
C_l	Local lift coefficient	$[-]$
D	Total drag force	$[N]$
D_i	Induced drag force	$[N]$
e	Wing span efficiency factor	$[-]$
FF	Form factor	$[-]$
ff	Fuel fraction	$[-]$
k	Reduced frequency	$[-]$
L	Total lift force	$[N]$
M	Mach number	$[-]$
M_{cr}	Critical Mach number	$[-]$
M_{DD}	Drag divergence Mach number	$[-]$
R	Range	$[m]$
S	Wing planform area	$[m^2]$
S_c	Local wing strip area	$[m^2]$
S_{ref}	Reference area	$[m^2]$
S_{wet}	Wetted surface	$[m^2]$

t/c	Thickness to chord ratio	$[-]$
V	Velocity	$[m/s]$
V_d	Dive speed	$[m/s]$
V_f	Flutter speed	$[m/s]$
W_{fuel}	Fuel mass	$[kg]$
$W_{TO_{max}}$	Maximum take-off weight	$[kg]$
y_{pred}	Prediction value	$[-]$
y_{val}	Validation value	$[-]$

Greek symbols

α	Angle of attack	$[^\circ]$
Γ	Circulation strength	$[m^2/s]$
κ_A	Korn factor	$[-]$
$\Lambda_{0.5}$	Midchord sweep angle	$[^\circ]$
ρ	Air density	$[kg/m^3]$
σ	Standard deviation	$[-]$
σ_{xx}	Normal stress in x-direction	$[MPa]$
τ_{xy}	Shear stress in xy-plane	$[MPa]$

Abbreviations

ANN	Artificial Neural Network
BB	Box-Behnken
CFD	Computational Fluid Dynamics
DLM	Dual Lattice Method
DOE	Design Of Experiments
DSM	Design Structure Matrix
FF	Full Factorial
LHS	Latin Hypercube Sampling
MDA	Multidisciplinary Analysis
MDF	Multidisciplinary Design Feasible
MDO	Multidisciplinary Design Optimization
MOI	Moment of inertia
NLR	Netherlands Aerospace Center
RBF	Radial Basis Function
RMSE	Root Mean Square Error
SBO	Surrogate-Based Optimization
SBW	Strut-Braced Wing
SM	Surrogate model

TBW	Truss-Braced Wing
VLM	Vortex Lattice Method
XDSM	Extended Design Structure Matrix

Part I

Introduction & Background

Chapter 1

Introduction

Current environmental targets that are set out worldwide put clear restrictions on the usage of fossil fuels. Further development of conventional aircraft configurations will not allow for a sustainable growth in air traffic [11]. In 2011, professor Agarwal of the department of mechanical engineering and material science, Washington University in St. Louis presented some predictions on the future impact of aviation on the environment [12]. He estimated that there are approximately 500.000 air vehicles and 750 million ground vehicles active worldwide. Together they are responsible for 50% of oil consumption and 60% of greenhouse gas production worldwide. Due to the prospected growth in world population, the increase in urbanisation and the economic growth of emerging countries, air traffic numbers are estimated to be doubled in 2050 compared to the current situation [12].

The European Union published the report Flightpath 2050 in the beginning of 2011, in which the future of aviation in Europe is set out [13]. Furthermore, clear goals are defined to minimize the environmental impact of aviation. The report estimates the number of flights within Europe will have grown to 25 million a year. This growth should be attained in a sustainable manner. In 2050 a 75% reduction in CO_2 emissions per passenger kilometre should be achieved, together with a 90% reduction in the production of greenhouse gasses and a reduction in perceived noise emission of 65% relative to the noise produced by aircraft available in 2000.

These estimations and ambitious targets clearly show the need for highly energy efficient innovative technologies in aviation. These innovations may come from new propulsive, systems, material or operational technologies, or from completely new aircraft concepts.

One aspect that reduces fuel consumption is a reduction in drag. Recent research has focussed on decreasing the induced drag by increasing the wing aspect ratio [14] [15]. An increase in aspect ratio can be reached by increasing the span and/or by decreasing the chord length. To determine the induced drag, the following well-know equation is used [16]:

$$C_{D_i} = \frac{C_L^2}{\pi ARe} \tag{1.1}$$

where AR is the aspect ratio of the wing, C_L is the lift coefficient and e represents the wing span efficiency factor.

A low induced drag can be achieved by increasing the wing AR, however, such an increase will have structural implications. With the same wing properties, an increase in span increases the moment arm of the forces acting on the wing, thereby increasing the total bending moment that has to be carried by the wing root section. The larger moments and forces can lead to stresses in the wing that are larger than allowable stresses set by material limits. This problem can either be solved by increasing the allowable stress or by decreasing the loads that have to be carried.

An indication of the trends of induced drag and wing weight with increasing span is given in Figure 1.1. One can see that the induced drag decreases rapidly with increasing span. The overall aircraft optimum however has more components than only aerodynamic efficiency. An increase in wing weight with increasing span is also shown in Figure 1.1.

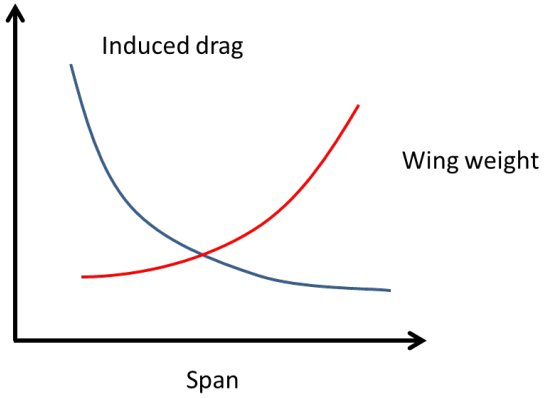


Figure 1.1: Wing weight and induced drag trends versus wing span [1]

A practical optimum aspect ratio of between 7 and 10 is reached for large commercial aircraft, taking the coupling between stiffness, thickness and mass into account, as is shown in Table 1.1[1].

[1]From: L. Jenkinson; P. Simpkin; D.Rhodes. Civil jet aircraft design. http://booksite.elsevier.com/9780340741528/appendices/data-a/default.htm. Last accessed:May 2016.

Table 1.1: Aspect ratios of modern aircraft

Aircraft model	Aspect ratio
B747-100,200	6.95
B747-400,800	7.39
A380-100	7.53
B767-200,200ER,300,300ER	7.99
A340-500,600	8.56
B777-200,300	8.68
B737-200	8.83
B737-300,400,500	9.17
A350-XWB, B787-8	9.25
A330-200,300; A340-200,300	9.26
A319-100;A320-200;A321-200	9.39
B737-600,700,800	9.44

Solutions to implement high aspect ratio wings

Different methods for increasing the wing aspect ratio can be applied:

- **Use more material in the wing structure** An increase in allowable stress can be reached by using more material in the wing, this however leads to a heavier structure.
- **Increase wing thickness** A different solution is to increase the wing thickness, but this leads to an increase in drag.
- **Make use of a strut** Strut-braced wing aircraft (SBW) concepts enable a decrease in the loads that have to be carried by the wing. The strut takes over part of the loading that previously had to be carried by the wing. This study uses the SBW concept as a case study to use implement aspect ratio wings in an automated optimization.

All solutions have to be traded off against the potential benefits of an increase in aspect ratio. This is can be done using a multidisciplinary design optimization (MDO). An MDO allows for an automated design process that incorporates the multiple disciplines, such as structures and aerodynamics, that play a role in aircraft design.

Challenges of increasing the wing aspect ratio

By increasing the aspect ratio of a wing, the risk of flutter also increases [17]. This needs to be taken into account during conceptual design. Since flutter calculations are relatively time consuming for a conceptual design study, flutter checks are often only performed on the final optimization result [18][19][20]. By adding a flutter check module to the optimization routine, every intermediate result is checked for the presence of flutter.

Decreasing the optimization time

Surrogate modelling methods can be applied to decrease optimization times. An optimization using surrogate models has optimization times in the order of seconds, whilst optimizations using the full flutter speed determination module have computational times in the order of hours.

1.1 Research question and objective

This research focussed on the implementation of flutter speed predictions integrated in the optimization loop. The principles and developed methods are applied to the optimization of a strut-braced wing. In previous research, flutter calculations were mostly performed ad hoc and outside the optimization loop [19] [21]. An MDO was set up to take all disciplines into account. The methods used for the different modules have a low-fidelity and several simplifications were applied to attain low computational times. It was aimed for one complete iteration of the optimization to take 1 to 2 minutes, which allows for a large number of iterations. Putting the focus on low computation times is however at the expense of accuracy of the results. The goal of the study was to come up with methods to implement the flutter speed calculation in an aircraft conceptual design optimization. Once it is determined how this can be done in a correct manner, the other modules of the MDO can be replaced by higher fidelity tools if a more accurate aircraft concept result is required.

This research furthermore introduced a method of flutter speed determination in which the actual flutter speed is fed back to the optimizer. In this manner, the optimizer has the ability to determine the influence of a geometry change on the flutter speed. Once flutter is encountered, the gradients of the flutter speed versus configuration parameter can direct the optimization in the correct direction. Surrogate-based optimization (SBO) methods have been applied to further decrease the optimization times. Surrogate models can be used to replace computationally expensive computations to enable fast automated optimizations.

The research questions that have been answered in this research are:

- How can flutter speed calculations be integrated effectively in a multi-disciplinary design optimization?
- Is it possible to use surrogate models to represent the flutter behaviour of the wing?
- Which surrogate model is able to represent flutter behaviour most effectively?
- How can surrogate modelling methods be implemented efficiently in a multi-disciplinary design optimization with flutter speed calculations?

1.2 Thesis structure

The thesis is split into three separate parts. The first part introduces the topics that are discussed in the report and an overview of research performed on the topics is given. The

second part shows the different analysis modules that are used in the optimization. The third and final part presents results, conclusions and recommendations.

All disciplines are represented by their own chapter in Part II of the thesis. The disciplinary analysis chapters follow the structure of the case study. An overview of the different disciplines and their couplings is given in a detailed Extended Design Structure Matrix (XDSM) (Figure 1.2) of the multidisciplinary analysis (MDA) [22]. Definitions of the variables in the figure are presented in Table 1.2. The design variables (x) are shown on top and are input for all modules. The process flow is indicated by the black line and follows the numbering presented in the XDSM.

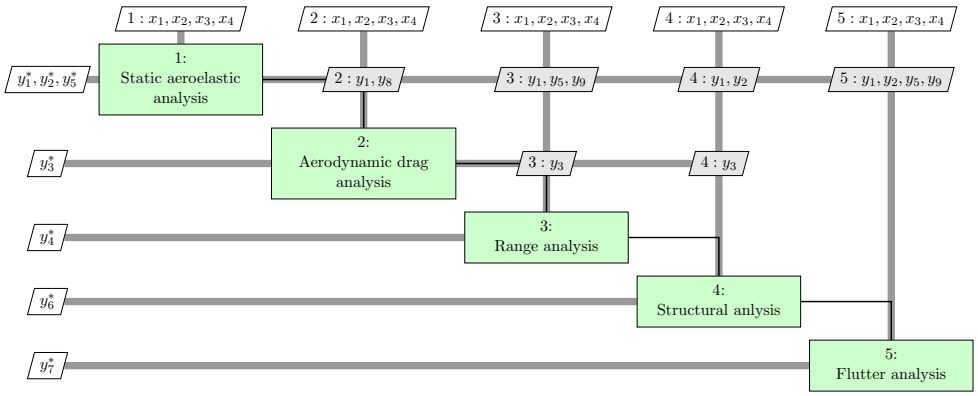


Figure 1.2: XDSM of multidisciplinary analysis

Table 1.2: XDSM variable definition

name	result
x^*	Optimal solution design variables
y^*	Final result
x_1	Root chord length
x_2	Wing semi-span
x_3	Root thickness
x_4	Span wise strut location
y_1	Wing lift
y_2	Wing bending moments
y_3	Wing and aircraft drag
y_4	Range
y_5	Maximum take-off weight
y_6	Maximum wing root stress
y_7	Flutter speed
y_8	Induced drag
y_9	Wing mass

1.3 Case study

A case study has been set up to evaluate the developed methods. The SBW aicraft concept developed in the SUGAR research performed by Boeing is used as a reference case [2]. A front view of this reference aircraft is shown in Figure 1.3. A Multidisciplinary Design Feasible (MDF) problem has been set op to perform the automated optimization [23]. In an MDF problem, all disciplines are coupled and the design variables are optimized at the top level by the optimizer.

The main geometric characteristics of the SUGAR aircraft are shown in Table 1.3. These values are used in case study. Four wing design variables are selected for the case study; root chord length, semi-span, root thickness and strut location. These variables are shown in Figure 1.4 and Figure 1.5. The variables refer to the names in Table 1.2.

Since the focus of the research was on high aspect ratio wings supported by a strut, the selected design variables are the ones that have a high impact on these parameters; wing span, wing chord length, wing thickness and strut attachment point on the wing. All other geometry parameters were kept constant and their values were selected from the SUGAR reference aircraft. The number of design variables was kept low to reduce the problem complexity and computational time. The goal of this research was not to find the best SBW design, but to develop methods that can be applied to the optimization of high aspect ratio wing. Many iterations and calculations have been performed since the calculation times were limited, which enabled the comparison of different methods. Significantly longer computation times would have resulted in a lower number of studies that could have been performed. The assumptions made in this research however have resulted in a low-fidelity final result since many geometrical components were kept constant. The simplifications have to be re-evaluated when a more accurate aircraft conceptual design is aimed for.

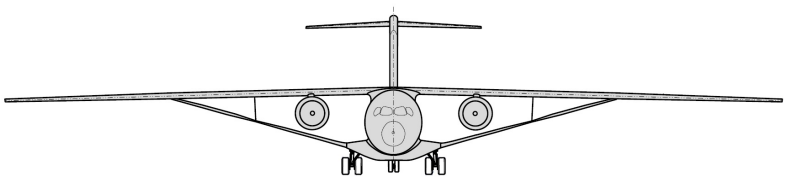


Figure 1.3: Front view of SUGAR reference aircraft [2]

Table 1.3: SUGAR reference values [2]

Fuselage	
Length $[m]$	42.4
Diameter $[m]$	3.8
Wing	
Area $[m^2]$	137
Span $[m]$	52
Root chord $[m]$	3.3
Aspect ratio [-]	19.74
Taper ratio [-]	0.35
1/4 chord sweep [°]	12.5
Strut location [-]	0.57
Vertical tail	
Area $[m^2]$	27.6
Height $[m]$	5.3
Taper ratio [-]	1
1/4 chord sweep [°]	41
Horizontal tail	
Area $[m^2]$	27.5
Span $[m]$	11.7
Root chord $[m]$	3.5
Taper ratio [-]	0.35
1/4 chord sweep [°]	25.3

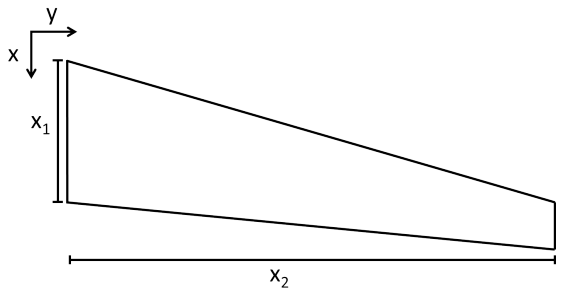


Figure 1.4: Schematic wing top view showing the wing design variables

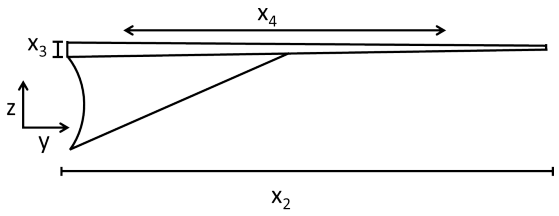


Figure 1.5: Schematic wing side view showing the wing design variables

Chapter 2

Background

2.1 Strut-braced wing

As explained in the Introduction (Chapter 1) the strut-braced wing concept was used as a case study. In this manner the methods developed in this research can be tested on a relevant aircraft concept. This chapter introduces the SBW concept.

A strut-braced wing aircraft concept enables a decrease in the loads that have to be carried by the wing and at the wing root. The strut alleviates the cantilever loading of the wing, which allows the wing aspect ratio to increase and the airfoil thickness to decrease. A thinner wing produces less drag. The internal structure of the wing can be optimized for the reduced wing loading, resulting in a lighter wing structure. Furthermore, the favourable drag effects allow the wing to apply a lower sweep angle, which increases the areas of natural laminar flow and allow for even further weight reductions [3]. Previous studies have shown significant (up to $10 - 20\%$) potential increase in fuel efficiency by addition of the strut [2] [3] [24] [25] [26].

The main disadvantages of the SBW concept are the weight and drag penalties of the strut. The drag penalty is largely determined by the interference drag at the intersections of the strut and fuselage and of the strut and wing [27]. The concept of using a strut to support the wing is applied to modern smaller aircraft (see example Figure 2.1), but not on large scale commercial aircraft. The drag and weight attached to the concept of SBW made the concept unattractive to use for commercial aircraft manufacturers in the early days of the development of large commercial aircraft [28]. Advancements in technology of structures and aerodynamics made it relevant to re-evaluate the trade-offs and to investigate if the SBW aircraft can be beneficial for future aircraft concepts. For this reason an increased amount of research has been conducted in recent years to investigate if it is beneficial to apply a strut on large commercial aircraft.

Figure 2.1: Example of a modern strut-braced wing aircraft (Cessna Caravan)

An example of recently performed research on SBW concepts is shown in Figure 2.2. This research project called ALBATROS was conducted at the French Aerospace Lab ONERA and was focussed on the advantages and disadvantages of the strut-braced wing configuration. The study showed that the concept enables an aerodynamic efficiency increase of the aircraft over conventional cantilever wing concepts. The research also confirmed the strong coupling between aerodynamics and structures.

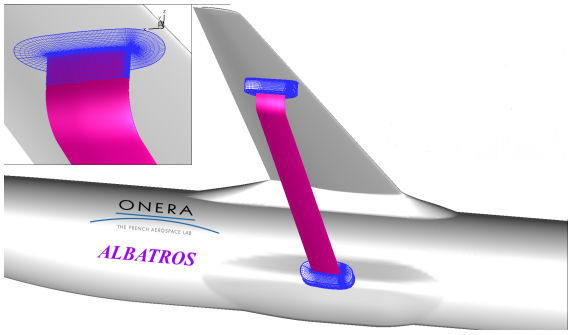


Figure 2.2: Picture of the strut-braced wing aircraft concept (ALBATROS) developed by ONERA [3]

Another recent research on SBW aircraft is Subsonic Ultra Green Aircraft Research (SUGAR) which was conducted by Boeing. The SUGAR Volt aircraft that was the result of this conceptual research makes use of strut-braced wings (a conceptual drawing is shown in Figure 2.3). Several papers have been published on the topics that are investigated as part of this research [2] [29] [30] [31]. Wind tunnel experiments showed the susceptibility of the high aspect ratio model to flutter. A clear recommendation made by the authors is to include detailed flutter analyses in any optimization routine of SBW aircraft concepts. This was an important goal of the thesis research, more information on this will be presented in the following chapters.

Figure 2.3: Example of strut-braced wing aircraft concept developed by Boeing (SUGAR research)

Different topologies can be used to implement a strut. A truss configuration with an extra structural member was implemented in the SUGAR research. Truss configurations make use of one or more structural members (sometimes also called juries) connecting the strut to the wing. Since adding extra members to the configuration increases the complexity of the model, truss-braced wing (TBW) configurations are not considered. Examples of a SBW and TBW configuration are shown in Figure 2.4.

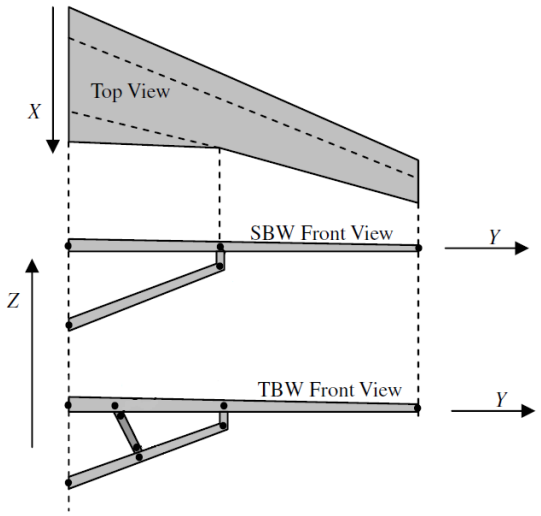


Figure 2.4: Example of a strut-braced wing and a truss-braced wing [4]

The well-known coupling between aerodynamics and structures discipline was indicated in previous research on SBW. This aero-structural coupling is taken into account using MDO methods for conceptual design. MDO can be a computationally demanding procedure mainly due to the disciplinary analyses applied. SBO focusses on methods that can reduce computation time significantly. The field of SBO focusses on the smart replacement of expensive analyses by surrogate models. These models are significantly less complex to evaluate and thus can be solved relatively fast.

2.2 Surrogate-based optimization

Surrogate modelling methods are applied to MDO applications for several reasons. First of all once the surrogate model is created, optimization time is reduced to several seconds instead of hours or days. The optimization is then performed using the surrogate model instead of the full multidisciplinary analyses.

Since it is unknown beforehand what the design space looks like and where the optimum lies, a mapping of the results is needed to obtain more understanding of the behaviour of the optimization. Once this mapping is created using a number of sample points, surrogate modelling methods are fast and efficient tools to use this data.

A surrogate model also gives flexibility. Since optimization time using surrogate models is reduced to seconds, performing optimization studies is no longer a time consuming procedure. All kinds of different studies can be performed without the need to run any of

the costly analysis tools again. When for example the objective function is changed, the resulting optimum can be found using the same DOE in a very short amount of time.

2.2.1 Surrogate modelling procedure

Several steps need to be taken before a surrogate-based optimization can be performed, these steps are shown in Figure 2.5.

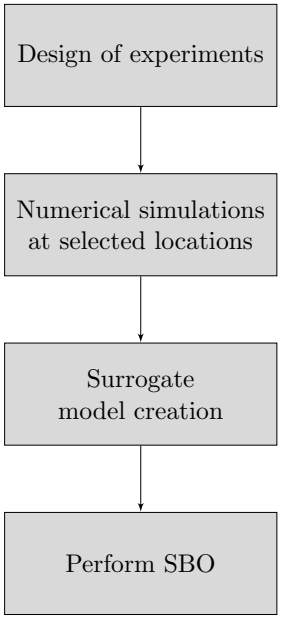


Figure 2.5: Surrogate modelling steps [5]

The first step is to create a set of data points on which the surrogate model is based. This set of data points is called the design of experiments (DOE). Several methods are available for the creation of a DOE. The exact method that is used has an influence on the quality of the surrogate models that can be created. Furthermore the size of the DOE also has a significant influence on the quality of the surrogate model. The creation of the DOE takes the most calculation time of an SBO. The determination of the number of data points is mainly a trade-off between accuracy of the results and computational time it takes to create the DOE. Once the DOE is selected, the second step is to perform the multidisciplinary analysis on all DOE points to generate the results.

The third step is to create a surrogate model on the available data points. Many methods are available to create a surrogate model. All methods have their own characteristics and can only be applied to a certain set of data points.

The final step is the actual optimization using the created surrogate model. The optimizer takes the initial values of the design variables as input and performs the optimization using the surrogate models.

2.2.2 Design of experiments

Design of experiments sampling method

For the creation of a design of experiments, different methods are available. Every method creates a different distribution of the points over the design space, thereby influencing the characteristics of the surrogate model that is created. Three basic and commonly used methods are described in this section.

Two-level full factorial

In a two-level Full Factorial (FF) design, every design variable is represented by two points, one point at its lower and one at its upper bound [32] [33]. This method does not place any design points at the center of the design space which can give problems when that region is of interest for the design study. Combinations of all points lead to a design space consisting of 2^x points, where x is the number of design variables. This method leads to a DOE of 16 points at the corners of the design space when four design variables are used. When a larger number of design variables is selected, the number of data points increases rapidly. An example of a FF design using three variables is shown in Figure 2.6. The design points are indicated by the dots in the figure.

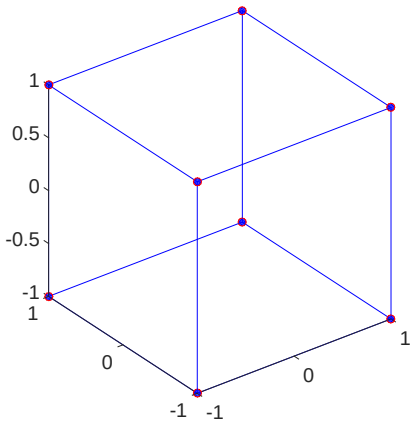


Figure 2.6: Three-dimensional two-level full factorial example

Box-Behnken

The Box-Behnken (BB) method places design points on the midpoints of the edges [10]. In this manner this method covers the edges of the design space. The BB method also places a design point at the center. An example of a BB DOE for three design variables is shown in Figure 2.7. The number of DOE points is shown for different numbers of design variables in Table 2.1. For higher number of design variables this results in a much lower total number of design points than the two-level full factorial method. This method has a better representation of the center part of the design space than the two-level FF method, but might still have too little points at that location to create accurate surrogate models.

Table 2.1: Number of Box-Behnken design points [10]

Number of design variables	Number of DOE points
3	13
4	25
5	41
6	49
7	57
9	97
10	161
11	177
12	193
16	385

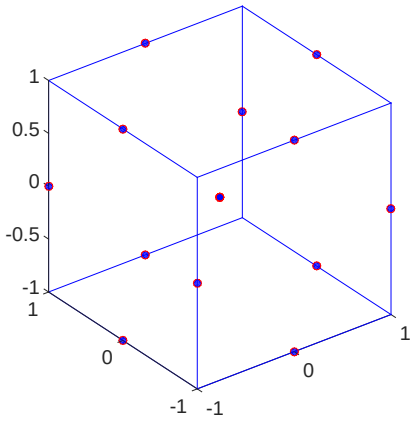


Figure 2.7: Three-dimensional Box-Behnken example

Latin hypercube sampling

A method that better covers the center region of the design space in a uniform manner is the Latin Hypercube Sampling (LHS) method [7] [34]. The complete design space is divided into N number of sections in every dimension. Then a total of N points are randomly divided over the entire design space in such a manner that only one point is placed in every row and column. In this manner a full coverage of the total design space is guaranteed. A two-dimensional example of a LHS is shown in Figure 2.8, a three-dimensional example is shown in Figure 2.9. The number of DOE points can be selected independent of the number of design variables. A downside of using this method is the moderate coverage of the edges of the design space.

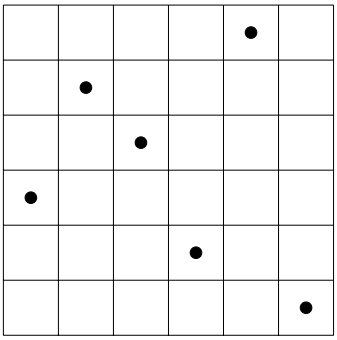


Figure 2.8: Two-dimensional Latin Hypercube Sampling example

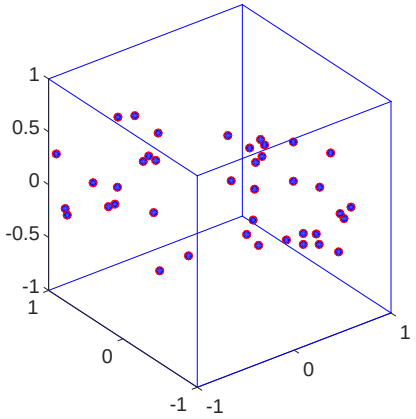


Figure 2.9: Three-dimensional Latin Hypercube Sampling example

Combination of different DOE methods

As the LHS method places design points on the inside of the design space and the two-level full factorial and Box-Behnken placed points at the corners and edges, a combination of all models covers the entire design space. A three-dimensional example is shown in Figure 2.10. The number of design points used in this method can increase rapidly for higher numbers of of design variables, since this DOE involves the full factorial method.

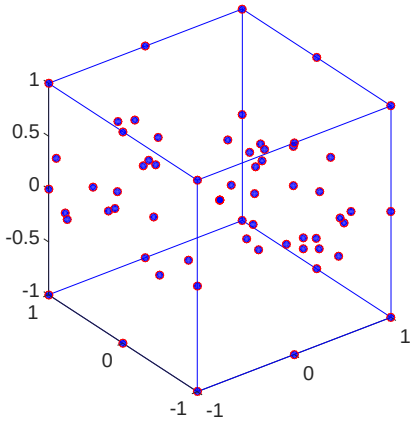


Figure 2.10: Combination of Latin Hypercube Sampling, Box-Behnken and two-level Full Factorial designs of experiments

2.2.3 Design of experiments size

The number of sampling points used in the DOE has a large influence on the final quality of the surrogate model that can be created. A larger design of experiments results in a better representation of the actual behaviour. The creation of design points is however expensive and should be minimized. A trade-off needs to be made between the computational cost and required accuracy of the results.

2.3 Surrogate modelling methods

A surrogate model can be created once a DOE is defined and the MDA evaluations are performed. Methods for the creation of a surrogate model have been developed in previous research [35] [36]. Some commonly used methods are selected to be investigated in more detail. For the creation of the surrogate models *MultiFit* is used[1]. *Multifit* is a tool developed by NLR[2] and is able to use different fitting methods to create surrogate models of the disciplinary analyses.

2.3.1 Polynomial functions

A basic surrogate modelling method is the polynomial. An n-degree polynomial is fitted on the DOE data. The fitting is based on the minimization of the least squares error between the data points and the fit. This approximation method is fast and simple, but is not always able to create a reliable surrogate model. Polynomial functions work best when the surface of the represented data set is smooth, sudden jumps in the data set might be difficult to represent by a polynomial function. Polynomial functions might be

[1] Multi-dimensional data fitting tool. Developed by National Aerospace Centre, Aerospace Vehicles division, Collaborative Engineering department.

[2] Netherlands Aerospace Centre in Amsterdam

suitable to create surrogate models for the objective function, but might have problems with the flutter speed.

2.3.2 Kriging

The concept of Kriging is based on the prediction of the values of a function at a given point by computing a weighted average of known values in the neighbourhood of that point [37]. Kriging is an interpolation method which means that the fit is exact at the DOE points. As a result of this, the surrogate model gives exact solutions at the design points, but is less accurate in locations between those points. Since the method is forced to include all points exactly, it is difficult to come up with a smoothed function. It depends on the underlying data if this is present. Different methods are available to produce Kriging interpolations, two methods are evaluated here, the *KriginglC* and *KrigingcG* method. The models differ in the method used for approximation, the first is a linear method using cubic splines, the second is a constant method using Gaussian approximation.

2.3.3 Artificial Neural Network

The final method that is considered is the Artificial Neural Network (ANN). An ANN consists of a large number of interconnected processing elements (the neurons). Connections between the neurons create communication networks. A weighting factor is associated to every separate connection to identify individual importance. These weights can be changed during optimization by training of the ANN. After every iteration the weights of the interconnections are updated to improve the network and thus to the coefficients of a surrogate model [38]. Since this method has the ability to train itself and to create a model of any shape, it is expected that an ANN method is well suited for the approximation of the flutter speed behaviour. A visual representation of the process is presented in Figure 2.11.

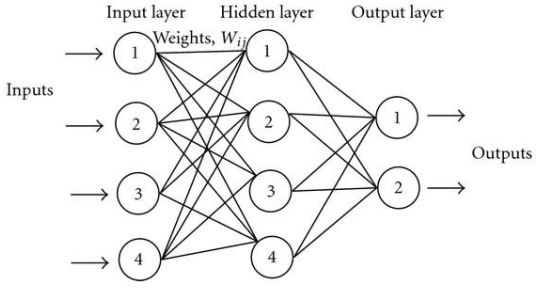


Figure 2.11: Schematic of an artificial neural network [6]

2.4 Adaptive surrogate-based optimization

The presented DOE methods a distribution of design point over the entire design space. Only a small region of this design space is interesting during the optimization, namely the

region where that optimum is located. Adaptive surrogate-based optimization methods could be applied to limit the total number of DOE points and to place these points largely in the region of interest, thereby saving computational time.

Adaptive SBO methods follow the same principle as the SBO method introduced in Section 2.2.1, the important difference lies in the update of the model. Once the first surrogate-based optimization is performed, the found optimum variables are used to determine the actual values for the flutter speed, Von Mises stress and range. This information is added as a separate point in the DOE and the surrogate model is updated using the new DOE. The process is visualised in Figure 2.12.

Several model updates are shown in Figure 2.13. In the first figure the true function, three DOE sampling points and the created surrogate model (polynomial) are shown. Two update points are shown in the next figure. These points are added at the minimum found using the surrogate model. The third figure shows the addition of more updating points, thereby improving the representation of the true function by the surrogate model. An advantage of this model lies in the fact that DOE points are only added at regions close to the predicted optimum. The accuracy of the surrogate model is increased at those interesting locations whilst a lower accuracy is accepted at other locations.

An alternative approach would be to add design points to the DOE other than the optimum. This might further increase the optimization speed and the accuracy of the result [39]. The methods presented in this study follow the simple principle of adding optimum points to the DOE.

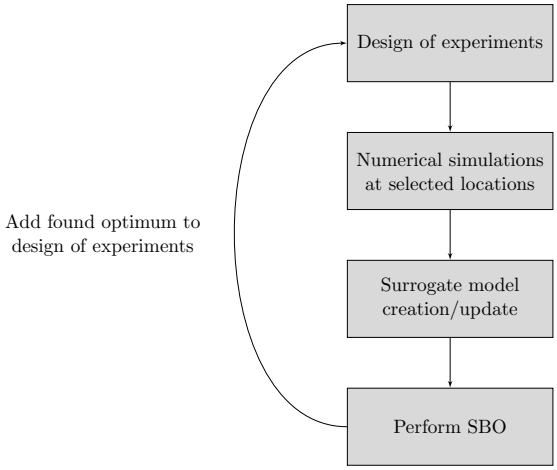
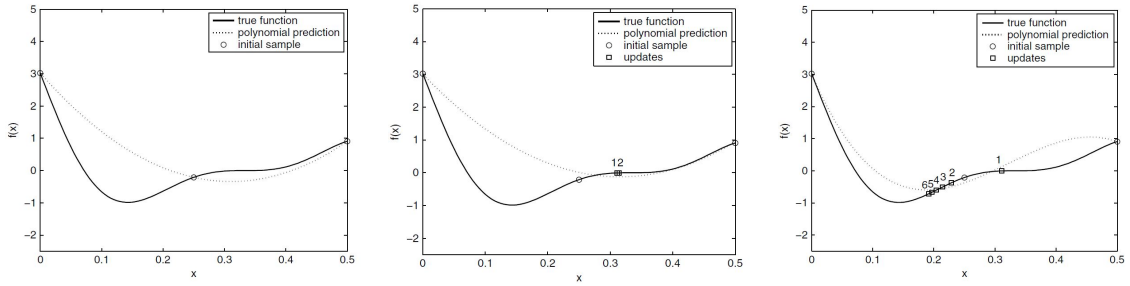


Figure 2.12: Adaptive surrogate modelling steps [5]



a: Initial surrogate model

b: Surrogate model after two updates

c: Surrogate model after multiple updates

Figure 2.13: Surrogate model update example [7]

Part II

Disciplinary analyses

Chapter 3

Static aeroelastic analysis

The first module in the MDA is the static aeroelastic analysis. The wing geometry variables are input for this module. The module determines a trimmed aircraft condition where the total lift created is equal to the total aircraft mass. A section of the XDSM indicating the module is shown in Figure 3.1.

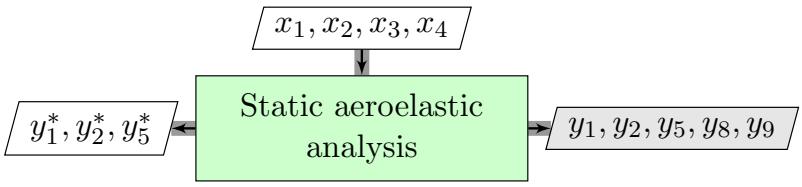


Figure 3.1: XDSM of static aeroelastic analysis

3.1 Method description

The module performs a static aeroelastic calculation using the *NASTRAN* solver. This solver is implemented in *AMLoad*[1]. *AMLoad* is developed by the Netherlands Aerospace Center and is an interactive, *MATLAB*[2] based tool that can be used for Aeroelastic Modelling and Loads/Flutter analysis. *AMLoad* creates input files for *NASTRAN* using geometrical data and the values of the variables. The result files are then read by *AMLoad* and the results are sent to the other modules in the MDA.

For the calculation of lift forces, *NASTRAN* uses a Vortex Lattice Method (VLM). VLM models a wing as an infinitely thin sheet composed of vortices. The method is based

[1] Aeroelastic Modelling and Loads analysis tool. Developed by National Aerospace Centre, Aerospace Vehicles division, Flight Physics and Loads department.

[2] Matrix laboratory. Numerical computing environment developed by MathWorks, Natick, Massachusetts, USA

on the lifting line theory developed by Ludwig Prandtl [16]. In the lifting line theory, a lifting surface is replaced by a line of vortices. These so called horseshoe vortices all have their own strength. This circulation strength of the vortex determines the local lift force. The method is based on the assumption that the flow is inviscid, which means the viscous drag is not computed. The only drag component that is calculated by the VLM method is induced drag.

The total mass of the aircraft is determined in *AMLoad*. Most component masses are kept constant, their values are taken from the SUGAR reference case as indicated in 1.3. Only the fuel mass, wing mass and strut mass are varied for different configurations. It is assumed that 50% of the total wing volume is used to store fuel [1]. A change in wing volume results in this case also in a change in fuel mass. The wing mass estimation is based on SUGAR reference results. Using this reference data a relation is used between wing surface area and mass. This simplification enables a wing mass estimation in a fraction of a second, which is beneficial for the overall computational time. If more accurate results are needed, a wing mass estimation tool can be implemented at the cost of more computation time. The strut mass depends on the length of the strut. In this simplified method it is assumed that the strut is homogeneous, meaning the same internal structure is used for the entire strut. The mass is then scaled linearly with the length of the strut. This simplification enables fast calculations.

Two load cases are evaluated in the static calculations module; a $1g$ horizontal flight condition is used for the range optimization, whilst a $2.5g$ load case is used for the determination of the maximum stress. The *NASTRAN* solver uses the mass and aircraft geometry to determine the angle of attack associated with these load cases. This is an iterative process in which the deformations of the wing under influence of the lifting forces are determined.

3.2 Implementation

The static calculations module feeds the resulting wing lift and induced drag to the aerodynamics module as can be seen in the XDSM (Figure 1.2). The wing lift and maximum take-off weight are used for the range determination. Furthermore, the wing lift and wing forces and moments are given to the structural analysis and flutter modules as the objective functions and constraints.

Chapter 4

Aerodynamic drag analysis

The next module of the MDA is the drag analysis module. In Figure 4.1 the input and output values are presented. The goal of the module is to calculate the drag of the aircraft.

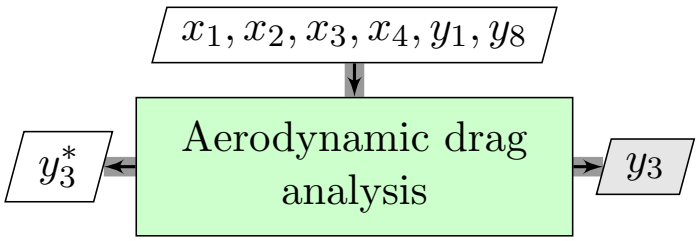


Figure 4.1: XDSM of aerodynamic drag analysis

Aerodynamic drag consists of several different components, an overview is given in Figure 4.2. For an aircraft MDO analysis, all drag components need to be calculated. As explained in the previous section, the static calculations preformed by *NASTRAN* only provide the induced drag, all other drag components represented in Figure 4.2 are not determined. To obtain a complete picture of all drag components, extra tools were used.

Two drag estimation tools were available to use in this study; *Friction*[1] and *Q3D* [40] [41]. Friction is available for free and could be implemented in the in-house NLR tool *AMLoad*, whilst *Q3D* is a tool developed at the Delft University of Technology and could not be implemented in *AMLoad*. A trade-off was made between different tools based on the accuracy and computational time of the tools. Furthermore, the validity of the low-fidelity tools in the applied design case was checked.

[1] W.H. Mason. Friction: Skin Friction/Form Factor Drag estimation. Virginia Tech.

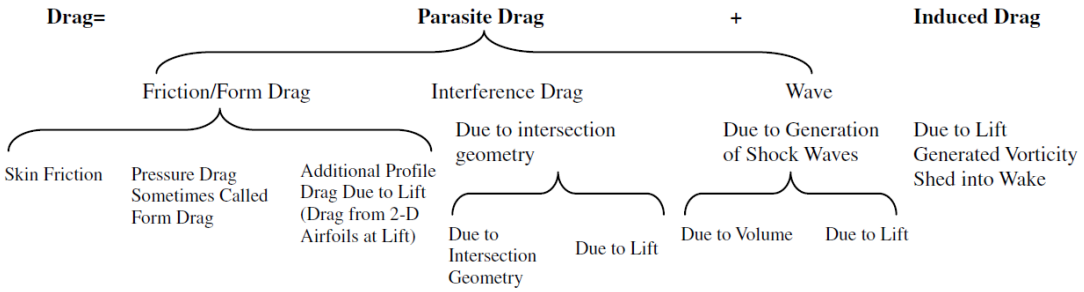


Figure 4.2: Overview of drag components [8]

In this conceptual design study the focus was put on the determination of the flutter speed and the implementation of a flutter speed calculation module in the MDO. To enable many configuration calculations in a short time span, the computational time of the aerodynamics module is an important parameter.

4.1 Method description

As explained, *Q3D* and *Friction* were selected as possible tools for the drag estimation module. Friction was implemented in *AMLoad* to combine its results with the results from *NASTRAN*. Both tools are compared for computational time and accuracy of the result. Final results of the comparisons are presented in Section 4.2. An overview of the tools used for different drag components in *Q3D* and *AMLoad* is shown in Table 4.1.

Table 4.1: Overview of the tools used in Q3D and AMLoad

Tool	Friction drag	Pressure drag	Wave drag	Induced drag
Q3D	*Xfoil* (M<0.55) *VGK* (M>0.55)	*Xfoil* (M<0.55) *VGK* (M>0.55)	*VGK*	*AVL*
AMLoad	*Friction*	*Friction*	*Friction*	*NASTRAN*

4.1.1 Q3D

Q3D is a quasi-three-dimensional aerodynamic solver, developed by Ali Elham (Delft University of Technology), that can be used to calculate lift and drag forces of a finite and 3-dimensional wing [40] [41]. *Q3D* uses of a combination of three different tools that together cover the full range of drag components. The following sections briefly introduce the tools and explain how they are utilized in *Q3D*.

Athena Vortex Lattice

The first tool used in *Q3D* is Athena Vortex Lattice (*AVL*)[2]. This is a calculation tool based on the Vortex Lattice Method. Using the lift force calculated by *AVL*, the induced

[2]M. Drela and H. Youngren. AVL (Athena Vortex Lattice) 3.26, 2006.

drag is determined as explained in Section 3.

Xfoil

Xfoil is a viscous/inviscid airfoil design and calculation tool developed at the Massachusetts Institute of Technology. The method uses an inviscid linear-vorticity panel method with a Karman-Tsien compressibility correction [42]. The tool can handle both laminar as well as turbulent boundary layers and a two-equation lagged dissipation integral method is used to represent the viscous layers. Especially for low Mach numbers, Xfoil is a reliable tool with a short calculation time (less than a second). When the Mach number becomes too high (close to the transonic area), the results of Xfoil are not accurate enough and another tool is needed. *Q3D* makes use of *VGK* for this range of Mach numbers.

VGK

The Viscous Garabedian and Korn method (*VGK*) makes use of a finite difference solution of the full potential equations integrated with the integral method for the boundary layer to calculate the pressure- and friction drag of an airfoil [43]. This method can handle both laminar as well as turbulent boundary layers [44].

Implementation in Q3D

Either *Xfoil* or *VGK* is used in *Q3D* depending on the Mach number. Up to Mach numbers of 0.55, *Xfoil* is accurate enough to predict the results at a much lower computational time. At higher Mach numbers, the results of *Xfoil* are no longer valid and *VGK* is used instead. Both *Xfoil* and *VGK* are two-dimensional airfoil calculation tools, it is not possible to calculate the drag of complete wings directly with these tools. Using simple sweep theory the two-dimensional airfoil results are used to find the three-dimensional wing lift and drag [41].

Elham has validated the *Q3D* tool intensively in his research for different ranges of flight conditions [40]. This makes the tool a reliable source of reference for other aerodynamic tools.

4.1.2 AMLoad

For lift and induced drag calculations, *AMLoad* makes use of *NASTRAN*[3] as explained in Chapter 3. Friction is implemented in *AMLoad* to calculate the remaining drag components.

[3]MSC NASTRAN: Multidisciplinary structural analysis, MSC software corporation.

4.1.3 Friction

Friction is a tool developed by Mason and consists of several analytical expressions that are used for drag estimation [8]. Since the method of *Friction* is based on analytical equations, the computation time of an evaluation is limited. A complete determination of the aircraft drag components by Friction takes only a fraction of a second.

It is however important to find out the accuracy of *Friction* and the validity of the assumptions made. It is to be expected that using simplified analytical equations for the calculation of drag instead of more advanced tools results in a less reliable solution. Since the focus of the study is on the implementation of flutter speed calculations in the optimization routine, a lower degree of accuracy of the other tools is acceptable if this results in lower computation times. The following sections show several comparisons between different tools to test the validity of Friction as a drag estimation tool.

As can be seen in Figure 4.2 the friction/form drag is divided into three different components; skin friction drag, pressure drag and lift-related profile drag. The third component is neglected by Friction since its value is relatively low compared to the other components [8]. The following sections introduce the equations used and determine their validity.

Friction and from drag

The friction/form drag can be calculated using the following equation:

$$C_{D,F} = C_F FF \frac{S_{wet}}{S_{ref}} \tag{4.1}$$

where C_F is the flat-plate skin-friction coefficient, FF represents the form factor (Equation 4.2), S_{wet} is the wetted surface of the wing and S_{ref} is the reference area. Friction can calculate the drag of an entire aircraft, dividing it into different components. Every component (wing, fuselage, nacelle, fuel tank, etc.) has its own form factor.

Various calculation methods for the form factor have been investigated in the past. This version of Friction makes use of the model developed by Torenbeek for the calculation of the form factor of a wing [45] [46]:

$$FF_{wing} = 1 + 2.7\frac{t}{c} + 100(\frac{t}{c})^4 \tag{4.2}$$

where t/c is an average of the thickness to chord ratio of the entire wing.

The flat-plate skin-friction coefficient, C_F can be calculated for a laminar as well as for a turbulent boundary layer. Friction is unable to determine if the flow inside the boundary layer is laminar or turbulent, neither can this tool calculate the transition point itself, this is an input to the solver.

Laminar flow
The calculation makes use of the Blasius skin friction formula, adjusted for compressibility using the Eckert Reference Temperature Method [47].

Turbulent flow
In case of a turbulent boundary layer, the skin friction is calculated using a method called the van Driest II method and is described by Hopkins [48].

Wave drag

Wave drag is a type of drag originating from shock waves that appear in transonic and supersonic flight conditions. At low Mach numbers the wave drag is zero, the drag creep in Figure 4.3 shows the slow increase in wave drag starting at Mach numbers still far below the speed of sound. The drag creep suddenly increases starting from a certain Mach number. This Mach number is called the drag divergence Mach number. Two definitions of its value are given in the figure. The first definition looks at the slope of the drag creep line. The point at which the slope is higher than 0.1 marks the critical Mach number:

$$\frac{dC_{Dw}}{dM}|_{M=M_{DD}} = 0.1 \tag{4.3}$$

The other definition is the 20 drag-counts definition; the point where the drag creep is equal to 20 drag counts ($C_{Dw} = 0.002$). These two definitions give almost the same drag divergence Mach number as can be seen in the figure.

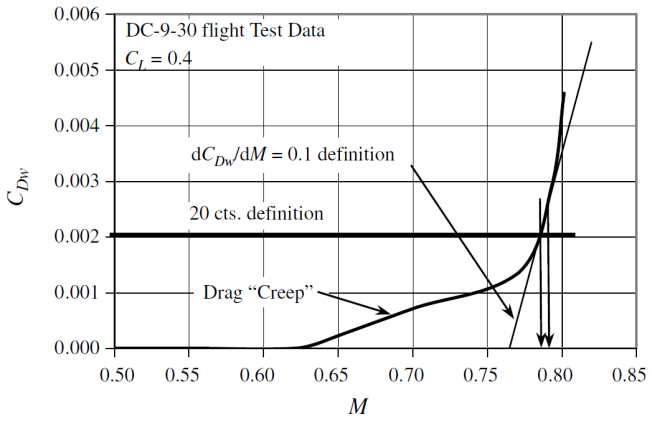


Figure 4.3: Wave drag [8]

$$C_{dw} = \begin{cases} 0 & M \leq M_{cr} \\ 20(M - M_{cr})^4 & M > M_{cr} \end{cases} \tag{4.4}$$

Inserting Equation 4.4 into Equation 4.3 results in the following definition of critical Mach number:

$$M_{cr} = M_{DD} - \sqrt[3]{\frac{0.1}{80}} \tag{4.5}$$

The drag divergence Mach number can be found by using the Korn equation extended with simple sweep theory [49].

$$\kappa_A = M_{DD} \cos \Lambda_{0.5} + \frac{C_l}{10 \cos^2 \Lambda_{0.5}} + \frac{t/c}{\cos \Lambda_{0.5}} \tag{4.6}$$

The value of the Korn factor, κ_A, for conventional airfoils is 0.87 [49].

The wing is divided into a limited number of sections for which the wave drag is calculated. The final wave drag of the three-dimensional wing is calculated using:

$$C_{Dw} = \sum C_{dw} \frac{S_c}{S_{ref}} \tag{4.7}$$

where S_c is the area of the local wing strip and S_{ref} is the total wing reference area.

Validation

Friction has been validated by Mason using several test cases; a subsonic wing, a transonic wing and a wing body-interaction case [8]. The conclusion of this validation was that *Friction* is a rapid and reliable drag estimation method, especially suitable for MDO applications. The model however has several limitations. For example *Friction* does not take flow separation into account, which makes the tool less suitable at high subsonic angles of attack. Furthermore, *Friction* does not handle angle of attack as an input variable, this means the drag can not be calculated at a specific angle of attack. When high angle of attack calculations are aimed for, this might not be a suitable tool. For an MDO application of a transonic aircraft in cruise conditions these limitations should not cause any problems.

4.2 Comparison between drag estimation tools

The previous section substantiates the benefits of using *Friction* as a drag estimation tool added to *AMLoad*. The results are compared to results generated by *Q3D*. *Q3D* is validated intensively by Elham for the transonic flight regime [40].

Since the computational time of *Friction* is less than *Q3D* (approximately 0.1 second versus 20 seconds), using *Friction* is preferred over *Q3D*. Another benefit of using *Friction* is the possibility to determine the drag of all aircraft components which enable a full aircraft drag determination. *Q3D* is only designed for the determination of wing drag. Furthermore, *Friction* is a free tool that could be implemented in the in-house NLR tool. It was checked if the accuracy of *Friction* was acceptable for the application.

Elham makes use of a transonic aircraft to validate his method. The test case is the Fokker 100 aircraft at $C_L = 0.2$ and an altitude of $10.000m$; a flight condition that is comparable to the application used for the strut-braced wing. Figure 4.4 shows the results of Friction versus Q3D. The comparison of the drag components between *Friction* and *Q3D* shows that *Friction* is good enough for the transonic cruise conditions (Mach number between 0.7 and 0.8, and low α). As explained in Section 4.1.3 the lift-related profile drag component is neglected. At small angles of attack, which is generally the case during cruise, this assumption is valid. For higher angles of attack this component will increase in value compared to the other two friction/form-drag components, making the assumption invalid. In Appendix A the input files for both tools are presented. For the evaluation of aircraft drag at transonic flight conditions *Friction* seems to produce results that are close enough to higher fidelity tools but at a lower level of complexity and against significantly lower computational times [8].

Table 4.2: Input variables for transonic Fokker 100 wing

Property	Value
Wing span [m]	14.04
Root chord [m]	5.6
Tip chord [m]	1.26
Altitude [m]	11000
Mach number [-]	0.77
C_L	0.2

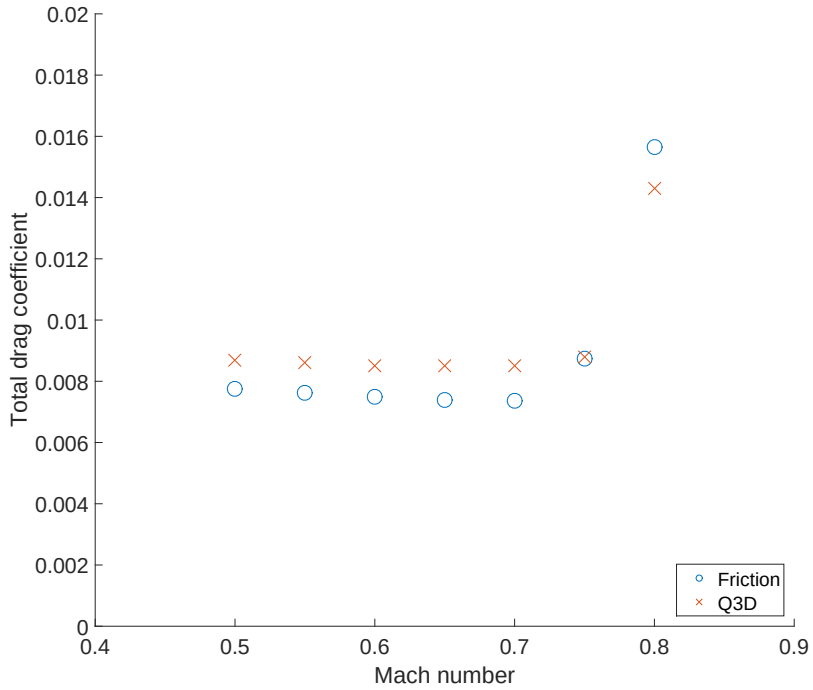


Figure 4.4: Total drag coefficient versus Mach number for $C_L = 0.2$

4.3 Implementation

The XDSM (Figure 1.2) shown in Chapter 1 indicates the location of the aerodynamics analysis module in the MDO routine. As explained the lift and induced drag results determined in the static analysis module serve as input for the aerodynamics module. The resulting drag is used in following modules to determine the stresses and range. *Friction* is implemented integrally in *AMLoad*, which enables simple and fast data transfer.

The strut is modelled as a straight wing with a symmetrical wing profile. The lift of the strut is assumed to be zero in the case study since the angle of attack is small during cruise flight. Only the friction and pressure drag components were calculated. This calculation was performed using *Friction*.

Chapter 5

Range analysis

The next module that is implemented in the multi-disciplinary design analyses is the range determination. The range analysis is a means to calculate the maximum total distance an aircraft configuration can cover. Several components of the aircraft are represented in this module. Section 5.1 presents the governing equations that are used, whilst Section 5.2 explains how these equations are implemented in the optimization routine. The input and output variables are shown in the section of the XDSM presented in Figure 5.1.

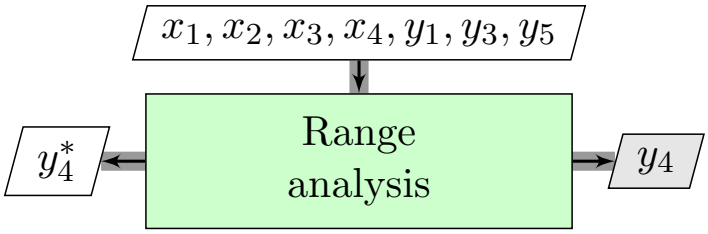


Figure 5.1: XDSM of range analysis

5.1 Governing equations

Use is made of the well known Breguet range equation (Equation 5.1), named after the French aviation pioneer Louis Charles Breguet [50]. The range can be calculated using a limited number of variables as shown in the following equations.

$$R = \frac{V}{C_T} \cdot \frac{L}{D} \cdot ln\left(\frac{W_{start-cruise}}{W_{end-cruise}}\right) \tag{5.1}$$

where

$$\frac{W_{start-cruise}}{W_{end-cruise}} = \frac{1}{\left(-\frac{W_{fuel}}{W_{TO_{max}} \cdot \text{ff}} + \frac{1}{\text{ff}}\right)} \tag{5.2}$$

Combining Equations 5.1 and 5.2 results in the final range equation used in the range analysis module of the optimization

$$R = \frac{V}{C_T} \cdot \frac{L}{D} \cdot ln \left(\frac{1}{\left(-\frac{W_{fuel}}{W_{TO_{max}} \cdot \text{ff}} + \frac{1}{\text{ff}} \right)} \right) \tag{5.3}$$

where ff represents a fuel factor, the ratio of fuel used during cruise over the total amount of fuel needed for the entire flight (including taxi, take-off and landing). The specific fuel consumption of the engine is given by C_T and assumed to be constant during cruise phase. The L/D ratio is calculated in the aerodynamic analyses as described in Chapter 4. The input V of the range equation represents the cruise airspeed. The fuel weight, W_{fuel}, is also taken into account for the calculation of the maximum take-off weight, $W_{TO_{max}}$. W_{fuel} is calculated as explained in Chapter 3. The other components adding up to $W_{TO_{max}}$ are the wing weight and empty aircraft mass minus wings. The latter component is assumed to be constant during the optimization since changes are only made to the wing. The wing mass variation with wing geometry is explained in Chapter 3.

5.2 Implementation

The Breguet range equation as presented by Equation 5.3 is implemented as a separate module in the MDA (see Figure 1.2). The four design variables are input to the range module together with the lift, drag and mass results from other modules. The case study that has been performed used the range as objective function.

Chapter 6

Structural analysis

A structural analysis was implemented as a separate module in the MDA. The XDSM of the module is presented in Figure 6.1. This module determined the maximum stress present in the wing root. This stress is used in the case study as a constraint.

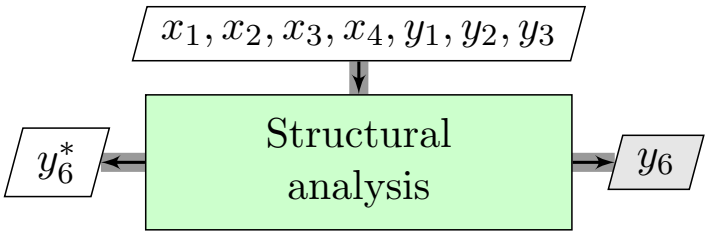


Figure 6.1: XDSM of structural analysis

As explained in Chapter 1 an increase in aspect ratio is beneficial for the aerodynamic efficiency of a wing. Limitations to the increase in aspect ratio were also indicated. The structural limitations are presented in this chapter in more detail. A simplified method or the determination of the maximum stress was developed.

6.1 Governing equations

The structural analysis module makes use of results coming from the aeroelastic calculations performed by *AMLoad* in the static calculations module (see Section 3). All moments and forces are calculated by *NASTRAN* and translated to the wing root section by *AMLoad*. The value of the wing root bending moment itself does indicate if the structure is able to handle these bending moments. For this reason the wing root bending moment needs to be translated to a maximum stress occurring in the wing structure. Every material has its own maximum value of stress that can be handled.

The method implemented in this study is a low-fidelity method. The low-fidelity resulted in calculation times of only a fraction of a second. In this study low computation times were preferred over high quality results (see Section 1.3). Higher-fidelity tools can be implemented when a higher accuracy of the result is required.

6.1.1 Wing structure

For the structural analysis the wing is assumed to consist of an aluminium wing box as is common in aircraft industry. The representation of the wing box as shown in Figure 6.2 is a simplification of an actual wing box structure. The wing root section that is of interest is indicated in Figure 6.3. For a conceptual design study this is accurate enough. The simplified geometry allows for fast calculations of all areas and moments of inertia that are needed for the stress calculations. The limited calculation time and complexity are suitable for conceptual design studies. For this type of studies it is important that all trends are represented correctly. For example an increase in thickness of the wing box increases the moment of inertia and decreases the stress due to bending as shown in Equations 6.1 to 6.3.

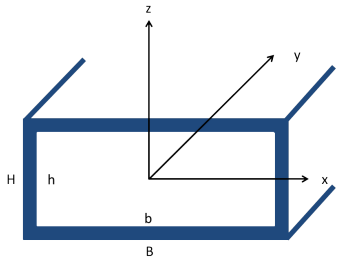


Figure 6.2: Wing box geometry overview

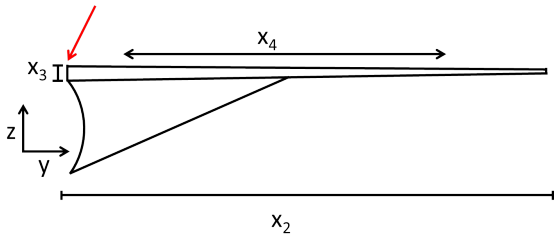


Figure 6.3: Schematic wing side view indicating the wing root section as region of interest

The maximum allowable stress of the wing box is set to $295 MPa$, which is a material property of aluminium[1]. On top of that, a safety factor of 1.5 is used to accommodate for any uncertainties in the material or calculations. Using a safety factor of 1.5 is common in aircraft design[2]. It is investigated what the maximum stress in the wing box at the

[1] Aircraft materials database. http://www.aircraftmaterials.com/data/aluminium/6082.html. Last accessed: November 2015.

[2] Federal Aviation Regulations Part. 25: Airworthiness standards: Transport category airplanes. Federal Aviation Administration, Washington, DC, 7, 2002.

plane of wing root is. For the calculation of this maximum stress, the Von Mises criterion is used [51]:

$$\sigma_v = \sqrt{\frac{1}{2}\left[(\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2 + 6\left(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2\right)\right]} \tag{6.1}$$

The six different stress components can be divided into two different types of stress; normal stress and shear stress. The first three stresses listed in the Von Mises equation are the normal stresses. The stresses are composed of a component due to a normal force and a component due to bending. The normal stress in the direction of the y-axis is calculated using:

$$\sigma_{yy} = \frac{F_y}{A_{yy}} + \frac{M_x z}{I_{xx}} \tag{6.2}$$

where A_{yy} indicates the surface perpendicular to the y-axis. The other two normal stresses are determined using similar equations.

The MOI of the wing box around the neutral axis is represented by I. The MOI of a rectangle can be calculated using the following equation [51]. This example shows the moment of inertia around the x-axis, I_{yy} and I_{zz} can be calculated using similar equations.

$$I_{xx} = BH^3 - bh^3 \tag{6.3}$$

where B and H are the outer width and height of the wing box, whilst b and h are the inner width and height of the wing box as indicated in Figure 6.2. The moment of inertia of the hollow wing box is calculated by subtracting the MOI of the inner rectangle from the MOI of the outer rectangle.

The other three stress components in the Von Mises stress equations are shear stresses. These stresses are composed of two components due to shear forces in two directions and due to a torsional moment. The mathematical expression is given below.

$$\tau_{zx} = \frac{F_z}{A_{yy}} + \frac{F_x}{A_{yy}} + \frac{M_y \rho_y}{J_y} \tag{6.4}$$

where ρ is the distance to the center of rotation (in this case the outer corners of the wing box will have the highest stresses due to torsion) and J is the polar moment of inertia.

The polar moment of inertia for a hollow rectangular cross-section is defined as:

$$J_y = \frac{BH}{12}\left(H^2 + B^2\right) - \frac{bh}{12}\left(h^2 + b^2\right) \tag{6.5}$$

This example shows the calculation of J_y, the other polar moments of inertia can be calculated in a similar manner.

Using the Von Mises equation described in this section, all forces and moments are taken into account for the determination of the maximum stress that is present in the wing box structure of the root. This leads to a maximum stress present at the corners of the wing box. In this specific application only some of the stress components are relevant and have

a significant influence on the final result. Either the magnitude of the force or moment is relatively low, or the value of the moment of inertia is relatively large.

For a wing loaded during a $1g$ or $2.5g$ flight condition and using the coordinate system as indicated in Figure 6.2, the following components have a significant influence on the maximum stress present in the structure:

- F_z - The dominant force is the force in the z-direction, caused by the lift of the wing. The force in z-direction is more than one order of magnitude larger than the force in y-direction. The C_L/C_D ratio of the reference configuration of approximately 25 indicates that the force in z-direction is 25 times higher than the force in x-direction [2] (see Table 8.1).
- M_x - The dominant moment is the moment around the x-axis. This is the bending moment of the wing as a results of the lifting force. M_x is larger than the moments in the two other directions partly due to the higher force in z-direction as explained before, and partly because the moment arm of this force is significantly larger than moment arms in the other directions. For the reference configuration M_x is one to two orders of magnitude larger than M_z. M_y is less than one order of magnitude smaller than M_x so its value is still significant. The total contribution of the different moments depends on the moments of inertia as is indicated in Equation 6.2.

With this force and moment as dominant over the other components, the following stresses are significant for the Von Mises stress:

- σ_{yy} - This is the normal stress in the y direction and is dominated by the bending moment of the wing. The lifting forces on the wing bend the wing upwards, resulting in a deformation of the wing. This deformation causes stresses in the upper and lower part of the wing box.
- τ_{zx} - This shear stress is dominated by the lifting forces on the wing. Since the forces on the wing in (positive) z-direction are dominant over the forces in other directions, the shear stress resulting from this forces is significant over the other shear stresses.

6.1.2 Strut structure

The strut is an important structural component in a strut-braced wing aircraft. The goal of the strut is to take part of the forces and moments working on the wing. As a result of this, the forces and moments acting on the wing root will decrease. A lower resulting Von Mises stress makes is possible to increase the wing span or to decrease the wing thickness. As explained earlier an increase in wing span increases the aerodynamic efficiency and a thinner wing decreases the drag force.

A simplified structure of the strut was selected for this study. The strut was modelled as a straight wing with constant chord and thickness. It is chosen to let the strut geometry vary with the wing geometry. The strut chord and thickness are chosen to be 25% of the wing parameters. This assumption has created a strut that did not buckle under

a -1g loading. This is again a simplification based on the SUGAR reference aircraft [2] [3]. By choosing this simplification, no extra design variables were selected to keep the computational time limited (as explained in Section 1.1). For this optimization, a detailed optimization of the strut was not necessary and would complicate the overall optimization. More detailed optimizations of the strut structure can be found in earlier research on SBW concepts [20].

It is assumed that the strut consists of a symmetrical wing profile and does not produce any lift during cruise. The strut is connected to the fuselage and wing with a fixed connection; all forces and moments are transferred from the wing to the strut [4].

It is assumed that the strut consists of a single element. As explained in Chapter 2, truss configurations are not considered. Other research performed on the geometry of the strut also considered truss configurations [4] [24] [30] [52].

6.2 Implementation

The structural module of the optimization typically sets a limit to geometrical values of the wing. The maximum Von Mises stress needs to be smaller than the maximum allowable material stress. The sizing load factor for this is the $2.5g$ loading [28]. During this load case, the loads on the wing are higher than during any other load case, thus resulting in the highest stresses. The aircraft structure needs to be able to withstand these maximum values.

The XDSM in Figure 1.2 shows the location of the structures module in the MDA framework. Results from static and aerodynamic modules are given as input for the structures module. The resulting Von Mises stress is fed to the constraint function in the MDO.

Chapter 7

Flutter analysis

Under the influence of aerodynamic forces working on the aircraft structure, deformations can occur. Especially the wings of an aircraft are susceptible to deformation under loading. During the entire flight envelope, wings show this aeroelastic behaviour, which is acceptable (for example the Boeing 787 wing bends more than $5m$ upwards during cruise flight).

A dangerous form of aeroelastic behaviour is flutter. This dynamic instability is characterised by a positive feedback between the force acting on the body and the resulting deformation. An example is wing bending under influence of the lifting force acting on the structure. Flutter occurs if this beam bending is not damped but increasing in amplitude. This divergence causes the deformations to increase rapidly, ultimately resulting in failure of aircraft components.

The designer of an aircraft needs to be certain that flutter will not occur during the entire flight envelope of the aircraft, but this behaviour is difficult to predict. The speed at which flutter will appear (the flutter speed (V_f)) should always be higher than 1.15 times the dive speed (V_d) of the aircraft as set by regulations[1]. The dive speed is the speed the aircraft should never exceed during its entire surfacing lifetime. This speed is normally only reached during certification flight testing. So if the flutter constraint is met, one ensures that flutter will never occur during operation of the aircraft.

Flutter calculations are complex and can be time consuming compared to low-fidelity tools used during the conceptual design phase. One complete calculation using the MDA described previously took around 90 seconds[2]. Of those 90 seconds, 60 seconds are used for the determination of the flutter speed. During an optimization process of a conventional aircraft, it is often not necessary to check the flutter constraint for every intermediate configuration, only the final result is checked for the presence of flutter [4] [21] [52].

Since flutter is an important criterion in high aspect ratio aircraft, it would be beneficial to implement flutter as a constraint in the optimization routine. In this manner, the

[1] EASA CS25. Certification specifications and acceptable means of compliance for large aeroplanes.

[2] The evaluations are performed on a computer with an Intel Core i5-4200M 2.50 GHz processor with 4 GB of RAM memory

optimizer always finds a solution that is completely flutter free and the constraint ensures a valid design point.

The XDSM of the module is presented in Figure 7.1. Results of the other modules are given as input and the resulting flutter speed is the output. This value was used in the case study as a constraint for the optimization.

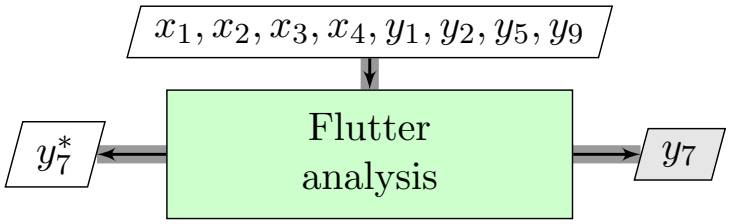


Figure 7.1: XDSM of flutter analysis

7.1 Governing equations

AMLoad is used to calculate the flutter speed of an aircraft configuration. This tool developed by *NLR* is responsible for the pre- and post-processing of *NASTRAN* calculations.

The following equation shows the general equations of motion of an aircraft:

$$[M]\{\ddot{q}\} + [D]\{\dot{q}\} + [K]\{q\} = \{Q_A\} \tag{7.1}$$

where $[M]$ is the mass matrix, $[D]$ represents the structural damping, $[K]$ is the stiffness matrix, $\{q\}$ is the vector of generalised coordinates and $\{Q_A\}$ represents the generalised aerodynamic forces.

A general form of solutions is assumed as:

$$\{q\} = \{\hat{q}\}\, e^{p\frac{Ut}{b}} \tag{7.2}$$

For p the following non-dimensional expression is used:

$$p = \frac{b}{U}(\sigma + i\omega) = \delta + ik \tag{7.3}$$

Implementing Equations 7.3 and 7.2 in Equation 7.1 results in the following equation of motion:

$$\left[\frac{U^2}{b^2}[M]\,p^2 + \frac{U}{b}[D]\,p + [K]\right]\{\hat{q}\} = \frac{1}{2}\rho U^2\,[A\,(p)]\,\{\hat{q}\} \tag{7.4}$$

Several methods are available to calculate the flutter speed. The following sections explains those methods and compares the different possibilities.

7.1.1 k-method

The k-method was first developed by Theodorsen in 1935 [53]. This method is also called the American method. His theory is based on the assumption of a fictitious structural damping g, and a purely harmonic response, which means $p = \frac{b}{U}(i\omega)$. Inserting this in Equation 7.4, results in the following equation of motion:

$$\left(-\omega^2 [M] + (1+ig)[K]\right)\{\hat{q}\} = \frac{1}{2}\rho U^2 [A(p)]\{\hat{q}\} \tag{7.5}$$

Now the eigenvalue problem can be written as:

$$[K]^{-1}\left([M] + \frac{1}{2}\rho b^2 \left[\frac{A(k)}{k^2}\right]\right)\{\hat{q}\} = \frac{1+ig}{\omega^2}\{\hat{q}\} \tag{7.6}$$

or in a more general form:

$$[B(k)]\{\hat{q}\} = \lambda\{\hat{q}\} \tag{7.7}$$

with

$$\lambda = \frac{1+ig}{\omega^2} \tag{7.8}$$

To solve this eigenvalue problem, the following procedure needs to be followed. Hassig named this procedure "Determinant Iteration":

1. a value for the density ρ needs to be chosen. This value often follows from a selection of the altitude.
2. a value for the reduced frequency k needs to be selected. This value can be used to calculate the aerodynamic data at this reduced frequency $[B(k)]$.
3. with these results, the eigenvalues can be calculated for every wanted mode.
4. the frequency at every mode can be calculated using $\omega_i = \frac{1}{\sqrt{\lambda_i'}}$.
5. the flight speed can finally be determined for every mode using $U_i = \omega_i \frac{b}{k}$ and the fictitious damping $g_i = \omega_i^2 \lambda_i''$.

To find the flutter speed, this process needs to be repeated for every mode for different values of k until it is found that the fictitious damping value changes from a negative value to a positive value.

This method provides a useful tool to find solutions to the flutter problem, however, the method is mathematically incorrect [54]. The assumption of a simple harmonic motion combined with an artificial damping is incorrect. Solutions to these equations are only correct in case $g = 0$, in other cases the system behaviour does not correctly represent the system.

7.1.2 p-k-method

Around the same time Theodorsen developed the k-method, Frazer and Duncan worked on a different solution method [55]. This method was later called the British method. In 1971 Hassig further developed this method and called it the p-k-method [9]. This method is based on the assumption that the aerodynamics of sinusoidal motions with slowly increasing or decreasing amplitudes can be approximated using purely harmonic aerodynamic results. The assumed aerodynamic forces $A\left(k, M_\infty, ...\right)$ are implemented in the equations of motion resulting in:

$$\left[\frac{U^2}{b^2}\left[M\right]p^2 + \frac{U}{b}\left[D\right]p + \left[K\right]\right]\left\{\hat{q}\right\} = \frac{1}{2}\rho U^2\left[A\left(k, M_\infty, ...\right)\right]\left\{\hat{q}\right\} \tag{7.9}$$

or

$$\left[F\left(p, k\right)\right]\left\{\hat{q}\right\} = 0 \tag{7.10}$$

Hassig describes an iterative process to find the solution for p [56]. The process starts by forming initial guesses for p_1 and p_2. These can be made based on the natural frequency of the mode in absence of aerodynamic forces.

1. Using the initial guesses for p_1 and p_2, $\left[A\left(k_1, M_\infty, ...\right)\right]$ and $\left[A\left(k_2, M_\infty, ...\right)\right]$ can be computed.
2. The determinants F_1 and F_2 can be calculated using $F_1 = |\left[F\left(p_1, k_1\right)\right]|$ and $F_2 = |\left[F\left(p_2, k_2\right)\right]|$.
3. The following recurrence formula is used to determine the new value for p. $p_{i+2} = \frac{p_{i+1}F_i - p_i F_{i+1}}{F_i - F_{i+1}}$.
4. Using the result for p_3, all values can be updated where $p_1 = p_2$ and $p_2 = p_3$.
5. Steps 1 to 4 need to be repeated until the values converge.
6. The next flight condition (altitude, speed, Mach number, Reynolds number, etc.) can be chosen.
7. Two new starting values for p_1 and p_2 can be based on extrapolation from previous flight conditions.
8. Steps 1 to 7 need to be repeated until the values of p are found for every wanted flight condition.
9. This procedure can be repeated for every required mode.

A downside of this method is the longer computation time compared to the k-method. The iterative process of calculating the flutter speed for every mode can become quite expensive relative to low-fidelity tools used in conceptual design studies. This is one of the main reasons flutter calculations are generally done "off-line", meaning not inside an optimization routine. Section 7.2 introduces a different method to deal with flutter constraints.

7.1.3 p-method

A method that is accurate over the entire range of flight conditions is the p-method. For this method, an approximate expression for $[A(p)]$ needs to be used. The time lags that are present in unsteady aerodynamics make the determination of this expression complex. Once this value is found, the *Determinant Iteration* as described in Section 7.1.2 can be used. To find the approximate expression for $[A(p)]$ in the time domain, a new state variable needs to be introduced, resulting in a higher order problem than when using the p-k-method. As a result of this, the calculation procedure will generally take more time than calculations using the k- or p-k-method.

Hassig performed research on the differences between the three methods described above [9], the results are shown in Figures 7.2 and 7.3. It can be seen that there are significant difference between the p- and k-method. This can result in a erroneous flutter prediction. Figure 7.3 shows a close representation of the p-method by the p-k-method. As explained the p-k-method is more efficient than the p-method.

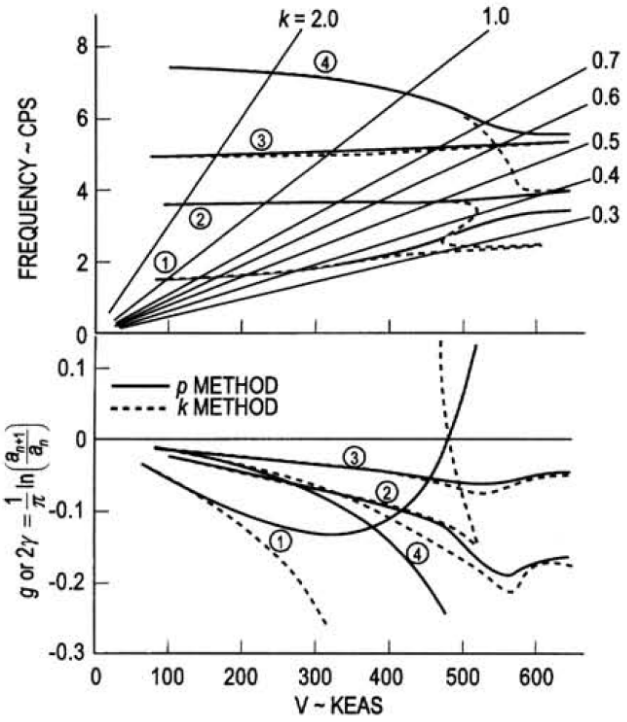


Figure 7.2: p-method versus k-method [9]

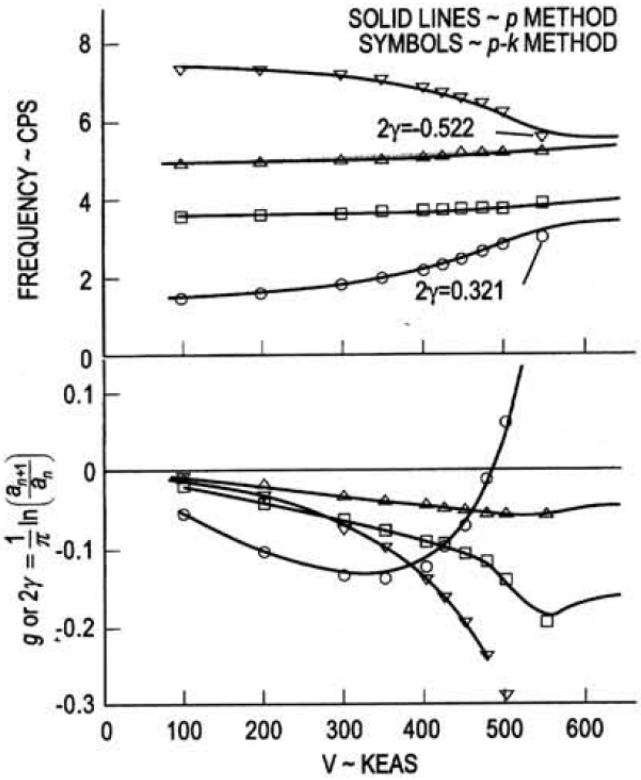


Figure 7.3: p-method versus p-k-method [9]

7.2 Implementation

7.2.1 Flutter speed calculation in AMLoad

The previous section showed the differences between the available flutter calculation methods. The p-k-method is capable of predicting the flutter behaviour accurately at a lower computational cost than the p-method. For this reason *AMLoad* uses *NASTRAN* with the p-k-method for its calculations.

The XDSM (Figure 1.2) shows the location of the flutter module in the MDA. The design variables define the aircraft model that is investigated, the static calculations model calculates all forces and moments acting on the aircraft. The flutter analysis module uses this input to determine the flutter speed of the aircraft configuration. The constraint function is defined by comparing the flutter speed with the 1.15 times the dive speed regulatory constraint.

The *Determinant Iteration* method defined by Hassig and explained in Section 7.1.1 is performed by *NASTRAN*. The trial values for k, the wanted speeds and number of modes to be calculated are given as input to the solver. With this input the *NASTRAN* solver gives the damping and frequency of the requested modes. The results are printed and can be read from a text file. *AMLoad* reads these data and creates plots of the damping values and frequencies. *AMLoad* also calculates the flutter speed from these results. It does so by finding the two speeds for a mode between which the sign of the damping

changes from negative to positive. By interpolating between these two points, the flutter speed can be found.

An example of a flutter diagram is given in Figure 7.4. The left part of the figure shows the frequency of all the modes at different speeds. The right part of the figure shows the damping of the modes at different airspeeds. The modes are ordered based on their frequency. In every configuration the frequencies of the modes are different, thereby also changing the mode numbers. The same mode in two configuration can for this reason have a different number. This complicates the optimization process since flutter speeds cannot be linked directly to separate flutter modes.

Most modes in this figure have a negative damping coefficient for every airspeed. For this configuration four modes cross the vertical axis, meaning the damping coefficient changes from negative to positive. These modes are identified by *AMLoad* and indicated with their corresponding speeds in the legend. A next step is to determine which of these modes represent critical flutter modes and should be considered in the determination of the overall flutter speed.

Specific regulations concerning flutter speeds are stated in the CS-25 certification specifications presented by the European Aviation Safety Agency (EASA)[3]. A mode should be considered critical if the damping value is larger than zero according to these certification specifications. Exceptions are made for the modes with a flutter speed smaller than $1.15V_d$, for these modes it should also be checked if these modes have a damping value larger than 0.03 at $1.15V_d$. If this is not the case, these modes can be neglected as critical flutter modes. This excludes mode 9 shown in Figure 7.4 as a critical mode. The other three modes are treated as critical modes and are used in the constraint function.

The p-k-method of *NASTRAN* calculates in some cases a non-zero damping value for the first six eigenvalue points, in which the frequency equals zero. This phenomenon is also recognized in the *NASTRAN* manual[4]. As stated in the manual, this damping value should be regarded as a computed zero and should not be interpreted as instability. Since the damping is non-zero and sometimes positive, it results in unrealistic flutter points. Those flutter points are found in the analyses at the starting velocity ($90m/s$).

Once all critical flutter modes and their corresponding speeds are determined, the lowest speed of the remaining modes is identified as the overall flutter speed and is fed to the optimizer.

[3]EASA CS25. Certification specifications and acceptable means of compliance for large aeroplanes.
[4]MSC.software, Aeorelastic Analysis User's Guide, MSC.Nastran Manual, 2012.2.

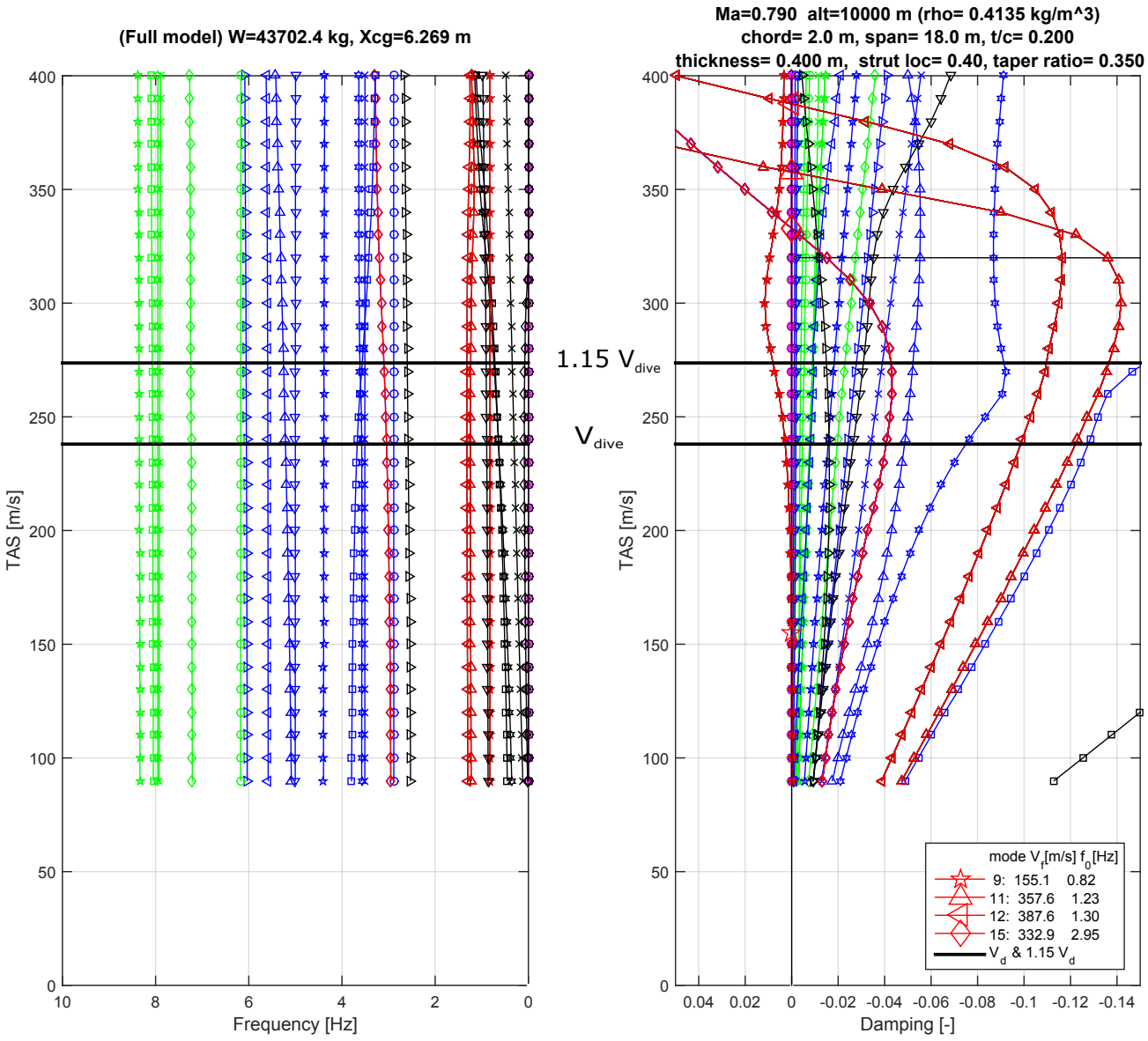


Figure 7.4: Flutter plot created by AMLoad

7.2.2 Implementation of flutter speed in optimization

As mentioned earlier, flutter calculations are usually performed outside the design loop and only for several configurations. In this manner only a check is performed on the final configuration, once the flutter regulations are met, the conceptual design can continue [19] [20].

Strut-braced wing aircraft concepts with high aspect ratio wings are more prone to flutter, for that reason it is proposed to take the flutter constraint into account during every optimization step. This means that every intermediate configuration is checked for the presence of flutter. Earlier performed research state the need for an implementation of a flutter constraint inside the optimization [17].

Several options are available to implement flutter predictions. In the optimization described in Chapter 8 the flutter speed is implemented as a constraint. The speed as calculated by *AMLoad* is compared with $1.15V_d$. The flutter speed should always be lower than this value. By using the actual flutter speed, the optimizer can also handle the derivatives of the constraint function to steer the optimization in the correct direction.

The top horizontal line in Figure 7.4 indicates the $1.15V_d$ value. For the case shown in the figure the lowest flutter speed is higher than the constraint value, so this configurations meets the regulations and can be treated as a valid configuration.

7.2.3 Flutter mode visualisation

Two modes were dominant in this study and are therefore visualised in this section. The first mode that is shown in the wing bending mode. The second mode is a tail flutter mode. For configurations with a lower aspect ratio, this is the critical mode with the lowest flutter speed. The flutter speed of this mode is however often higher than the constraint value. The added value of the strut can be seen in both mode visualizations. The section of the wing inboard of the strut shows only little deformation, whilst the outboard section shows significantly larger deformations.

7.3 Wing bending mode

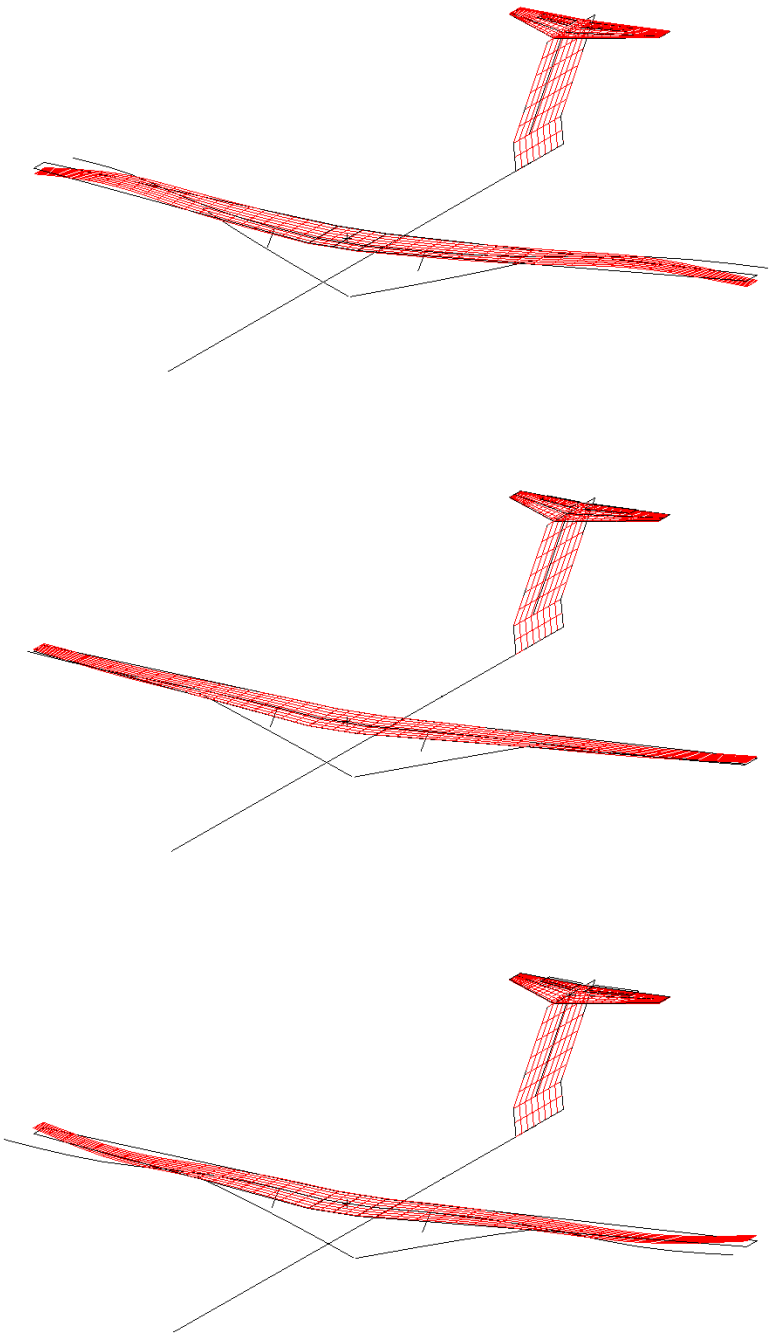


Figure 7.5: Wing bending mode visualisation

7.4 Tail flutter mode

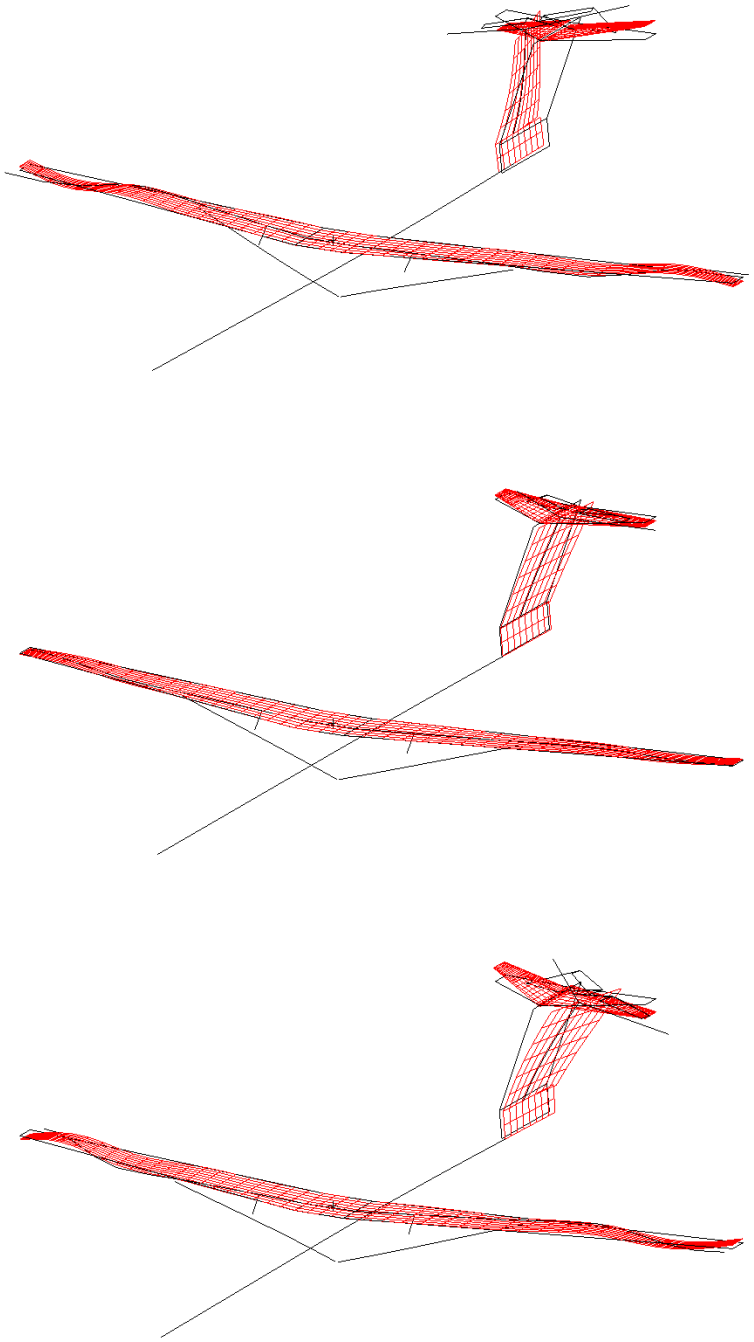


Figure 7.6: Tail and wing flutter mode visualisation

7.5 Results

A small case study is defined to check the functioning of the flutter analysis module and to show the behaviour of the flutter speed versus the wing aspect ratio. The flutter speed is calculated for every configurations. Only three variables are considered in this test case to keep the computation time limited and to isolate the relation between flutter speed and wing aspect ratio. The variables considered in this small test case are the wing span, wing chord and strut location. The first two variables determine the aspect ratio of the wing, whilst the strut location has a great influence on the flutter behaviour.

Aspect ratio and strut location variation

The flutter speed of the SUGAR reference aircraft configuration for different combinations of wing span and strut location is shown in Figure 7.7. The strut location is defined as the span wise position of the strut-wing connection point (where 0.5 means the strut is placed at 50% of the wing span). Several conclusions can be drawn from this figure.

In general it can be concluded that a relatively small aspect ratio the flutter speed is high. The flutter mode corresponding to these flutter speeds is a tail flutter mode. This mode is indicated by the red markers in the figure.

The behaviour of flutter speed versus wing span is nearly linear over a large part of the configurations. An increase in aspect ratio results in a decrease in flutter speed. There are however exceptions, of which one is clearly shown in this figure. For a configuration with a strut location at 0.5 times the wing semi-span, the flutter speed makes a sudden jump when the aspect ratio is increased from 18.5 to 21. These two points are emphasized in the figure by two circles. This jump occurs when two modes interact with each other and is present at certain configurations. Small geometric changes can have a large impact on the resulting flutter speed. This might be difficult to predict using surrogate models. High quality models would be needed to represent this behaviour.

The switching of mode is also indicated in Figure 7.7. At short wings spans, the critical mode is a T-tail flutter mode but the flutter speeds are above the constraint value. The red markers in the figure indicate these tail flutter modes. A visualisation of this type of mode is shown in Figure 7.6. Once the aspect ratio is increased the wing bending mode becomes the critical mode and the flutter speed decreases as indicated by the black markers in the figure. The wing bending mode is visualised in Figure 7.5. The exact point where this switching of modes takes place is difficult to predict, since it depends on interactions between the different geometrical parameters. The optimizer does not know which mode is critical, only the overall flutter speed is known. This makes it impossible for the optimizer to track the influence of a geometry change on the flutter speed of individual modes.

The horizontal line in Figure 7.7 indicates the value of $1.15V_d$. The lowest AR configurations clearly meet the requirement, independent of strut location. With increasing wing span, the flutter speed decreases. For outboard strut locations (from 0.5) this results in flutter speeds lower than the regulation constraint for large aspect ratios.

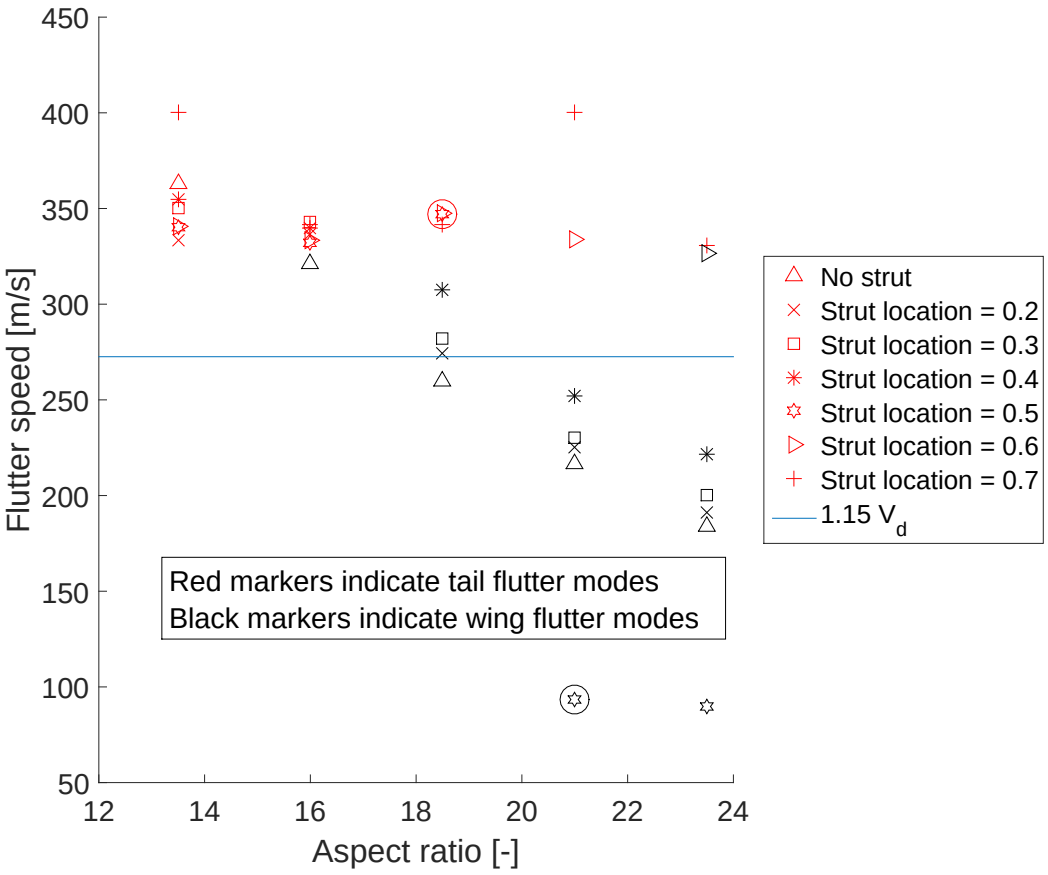


Figure 7.7: Flutter speed versus wing span for different strut locations

7.6 Conclusion

The flutter analysis module was introduced in this chapter. Some theoretical background on the calculation of the flutter speed was given and it was shown how this was implemented in the MDO framework.

Different modes are responsible for flutter as was shown in the mode plots. This makes it difficult to predict the flutter speed of a certain configuration. Flutter is highly dependent on interaction of different modes. If the frequency of one of those modes is changed, the flutter behaviour might change completely. The conclusion could be drawn from this example that the behaviour of the flutter speed is difficult to predict. This can complicate the creation of a surrogate model, which is shown in Chapter 9.

Part III

Multidisciplinary analysis and surrogate-based optimisation

Chapter 8

Multidisciplinary design optimization

8.1 Implementation

Using the modules described in Chapter 3 to Chapter 7 a Multidisciplinary Design Optimization can be set up. The following optimization problem has been defined:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & \text{-}R(x) \\ \text{subject to} \quad & g_1(x) = \frac{\sigma_{VonMises} \cdot 1.5}{\sigma_{ref}} - 1 \leq 0 \\ & g_2(x) = \frac{1.15 \cdot v_{dive}}{v_f} - 1 \leq 0, \end{aligned} \tag{8.1}$$

The range equation defined in Chapter 5 was selected as the objective function for the case study. x represents the design variables. The negative of the range is minimized in order to achieve range maximization. The Von Mises stress and flutter speed are selected as constraint functions. This objective function was selected to find results with large fuel volumes as optimum. Large wings with wing tanks completely filled with fuel increase the mass of the wing. These load cases have shown to be most susceptible to flutter [2]. This optimization problem was assumed to be a suitable case study for the implementation of a flutter constraint in the optimization routine.

The structure is presented in the XDSM (Figure 8.1). Using the limited number of four of design variables enables fast calculations and allows for many iterations. Also for the implementation of Surrogate-Based Optimization (Chapter 9) a limited number of design variables is preferred. The variables selected for the case study have been introduced in Section 1.3. When the method is implemented correctly, the number of variables might be increased in order to increase the precision of the result.

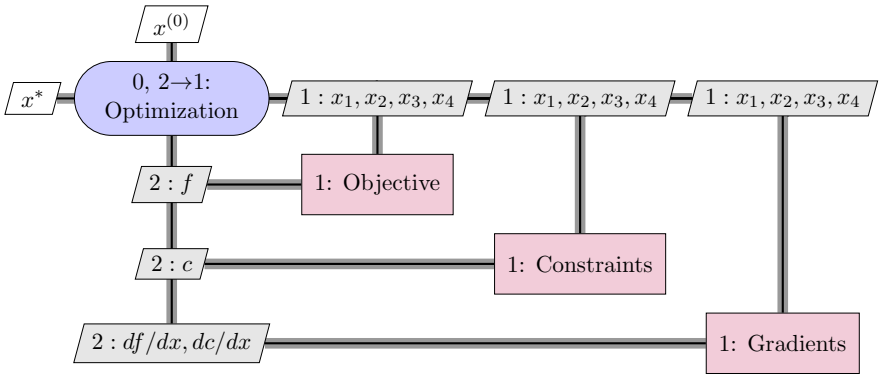


Figure 8.1: XDSM of the multidisciplinary design optimization

MATLAB was used to perform the optimization. All modules are *MATLAB* based and the *NASTRAN* solver is implemented in *AMLoad*. A gradient-based optimization (using the default interior-point algorithm) in the *fmincon* function has been used. This method is the default algorithm for *fmincon* optimization problems. In a gradient-based optimization, the optimizer determines the derivatives of the objective function with respect to every design variable. The optimizer changes the value of the design variable by a small step and determines the new results. The derivatives of the objective function and constraint functions with respect to the design variable are determined in this manner. In this case the optimizer can determine in which direction the optimum can be found.

The minimization of the negative of the range results in a maximization of the range value (see Chapter 5). The optimization problem includes two constraint functions; one function for the Von Mises stress (see Chapter 6) and one function for the flutter speed (see Chapter 7).

8.2 Results

The goal of the optimization was to find the aircraft configuration that leads to the best possible value of the objective function. This study has found that the final result of the optimization strongly depended on the starting point of the optimization. The starting point is the first configuration that is evaluated by the optimizer. The gradients are determined and the optimizer tries to find a configuration that satisfies the constraints and which results in a larger objective function value. The flutter constraint puts boundaries on the aircraft geometry variables. The flutter speed can have sudden jumps as was shown in Section 7.5. The gradients of the optimization give information on the relation between flutter speed and geometry variation locally, it is not known that there might be a better solution at a completely different design point.

For this reason it was important to find out if an optimum found by the optimizer was in fact a global optimum. A solution to this problem is the usage of multiple starting points. The same optimization is performed but every optimization starts from a different point in the design space. Using different starting positions results in different search directions, which increase the possibility to find the global optimum. If the number of different

starting points is large enough, the global optimum should be found. The *MATLAB* Multistart function was be used to perform this procedure.

The final range result was used as a reference value. Performing the same optimization 10 different times is a very time consuming process. If the computational time of one iteration would be much larger than the 1.5 minutes of the MDA presented in this study, it would be impossible to perform the Multistart procedure. It has been investigated if SBO methods are able to predict the same objective value at a lower computational cost. The results are shown in Chapter 9.

8.3 Comparison with SUGAR

The results of the performed optimization study were compared with reported results of the SUGAR project [2]. The SUGAR project was used as a reference case, as was explained in Section 1.3. In Table 8.1 the found optimum aircraft configurations have been compared. The last column represents the results found in the study described in this chapter. The optimum span and chord values are larger than the SUGAR reference aircraft, the aspect ratios are however almost equal. Differences can be the result of the simplifications applied in this study. For example the optimization result found an aircraft that is heavier and larger than the SUGAR aircraft, but other components are not scaled. A heavier and larger aircraft would need an increase in tail volume which increases the drag and mass of the aircraft. The increase in drag would then decrease the C_L/C_D ratio.

Table 8.1: Optimization results compared with reference

Wing	SUGAR	Optimization result
Area $[m^2]$	137.0	168.2
Span $[m]$	52.0	57.78
Root chord $[m]$	3.30	4.62
Root thickness $[m]$	0.40	0.45
Aspect ratio $[-]$	19.74	19.85
Strut location $[-]$	0.57	0.43
C_L/C_D $[-]$	25.2	29.4
Range $[-]$	6.88E+06	9.96E+06

8.4 Flutter constraint in the optimization

The flutter constraint was an important aspect of the MDO. The functioning of this constraint has been investigated and results are presented here. In Table 8.2 three intermediate results of the optimization are shown. A feasible solution has been found after the first iteration, both constraints are satisfied. The Von Mises stress is below the value of $1.967E+08$ (see Chapter 6) and the flutter speed is higher than 272.5 (see Chapter 7). The second iteration found a result with a higher objective function value by increasing the wing span but the flutter constraint is not satisfied. In the third iteration the wing span is decreased again to satisfy the flutter constraint. The flutter plots belonging to

these design points are shown in Figure 8.2 to Figure 8.4. The damping value of the wing bending mode (mode 9) is positive for a speed lower than the constraint value in the second figure. Mode 5 represents a rigid body mode and can be neglected (see Section 7.2).

Table 8.2: Optimization with active flutter constraint

Iteration number	1	2	3
Chord $[m]$	4.3	4.2	4.2
Semi-span $[m]$	23.6	24.8	23.6
Root thickness $[m]$	0.428	0.428	0.448
Strut location $[-]$	0.39	0.39	0.39
Von Mises stress $[MPa]$	1.189E+08	1.269E+08	1.1120e+08
Flutter speed $[m/s]$	294.1	270.3	311.4
Range $[m]$	7.901E+06	8.103E+06	8.096E+06

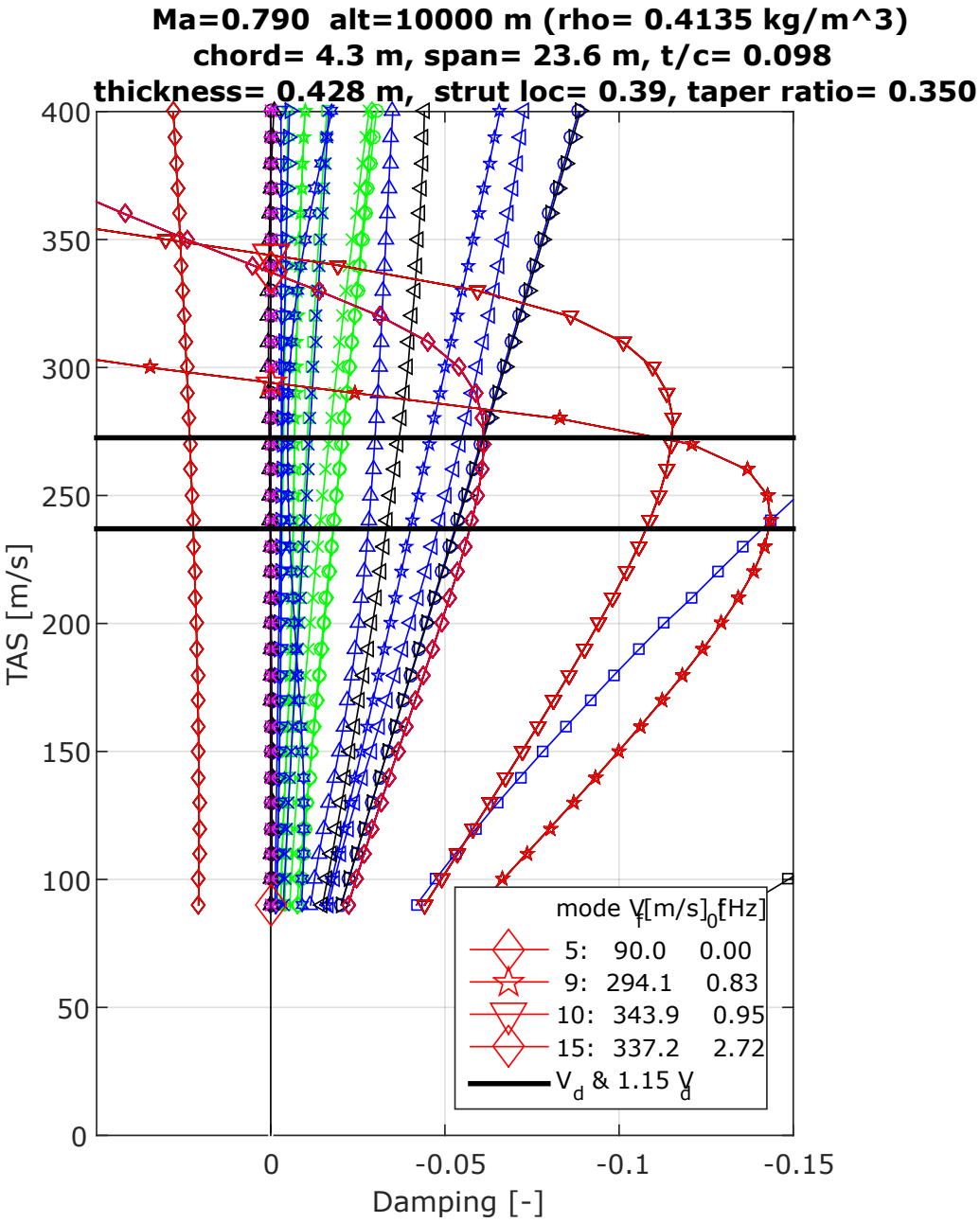


Figure 8.2: Flutter constraint not active

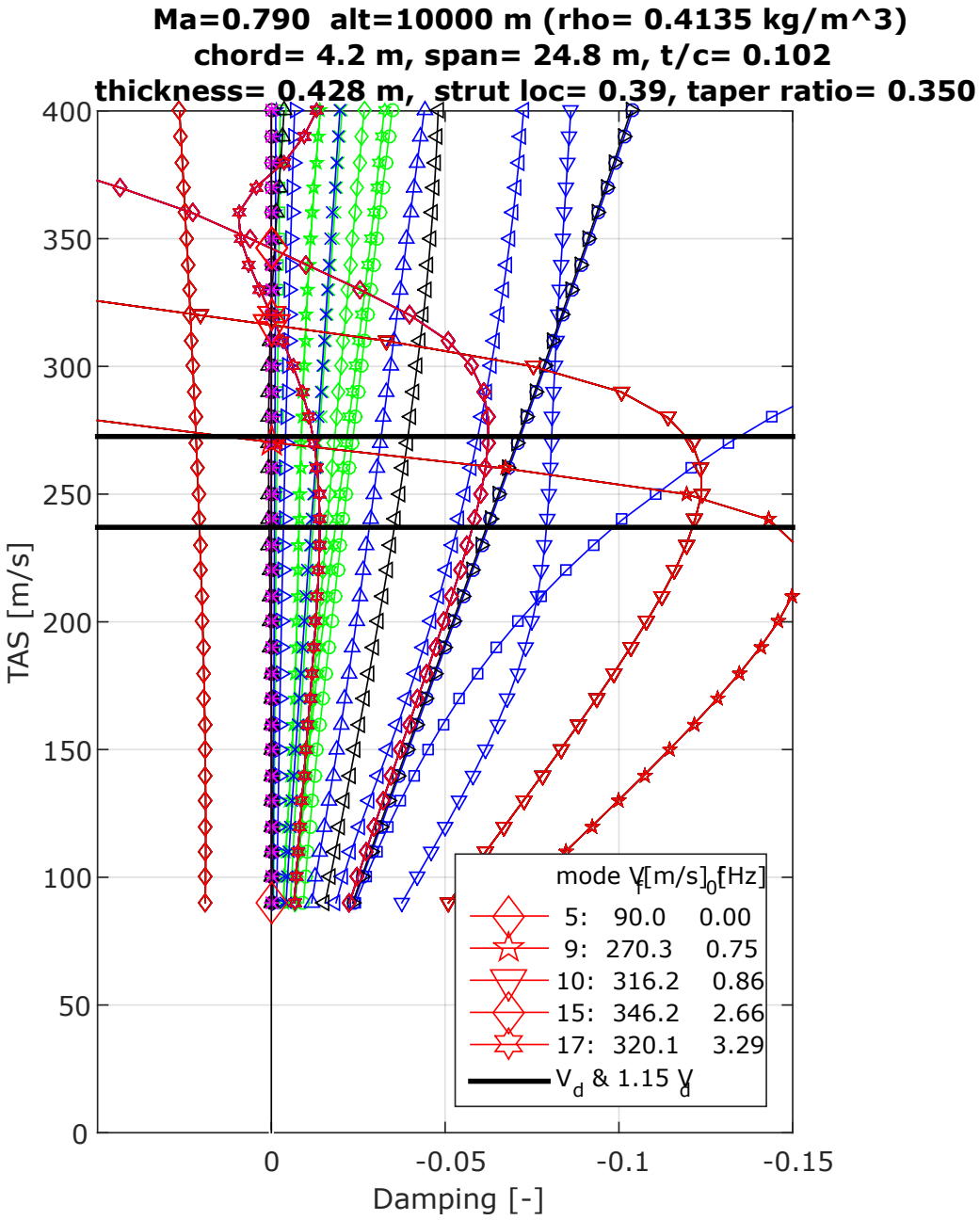


Figure 8.3: Active flutter constraint

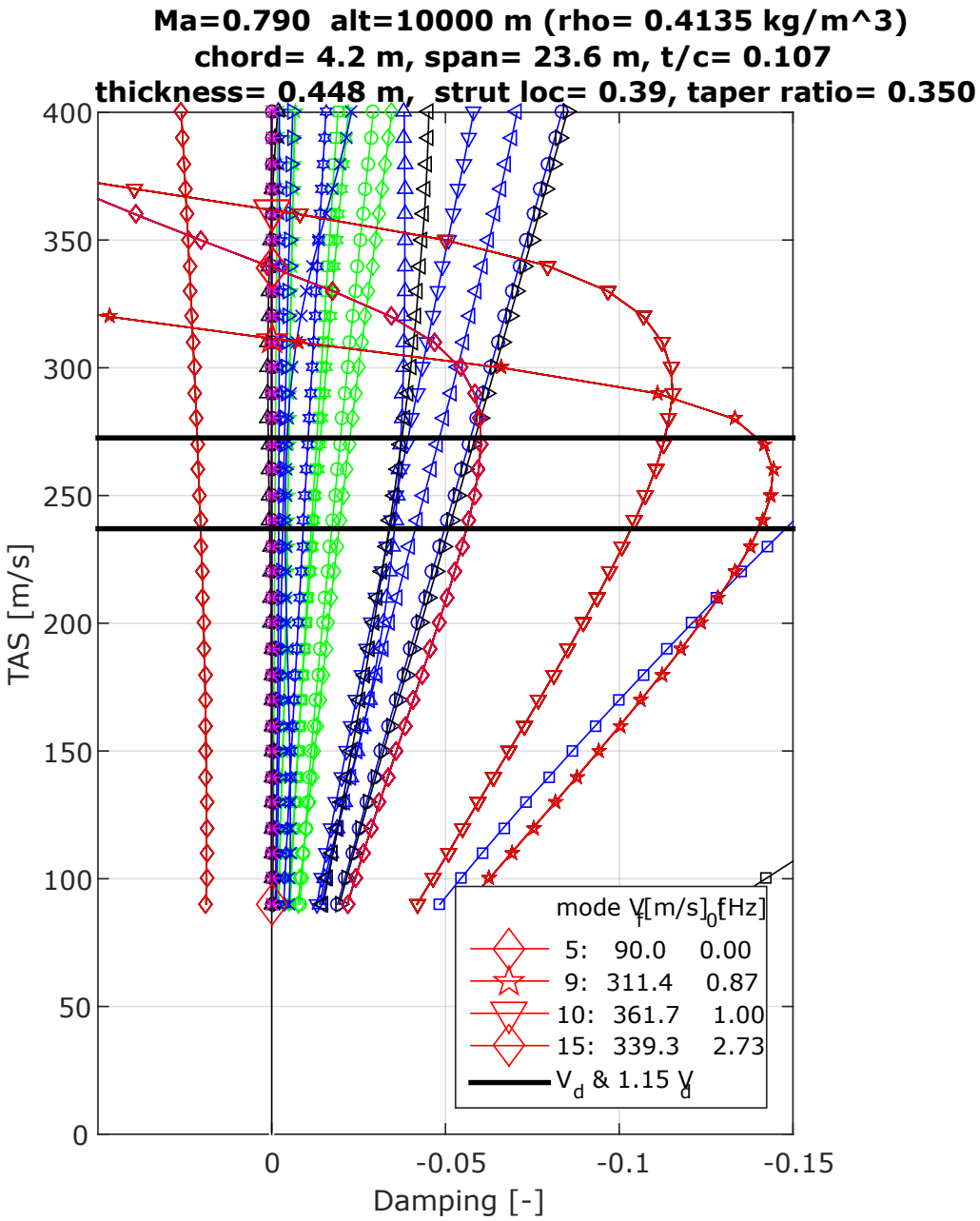


Figure 8.4: Flutter constraint not active

Chapter 9

Surrogate-based optimisation

The advantages of of applying surrogate models to an optimization problem have been explained in Section 2.2. Several methods have been applied and the results have been investigated. The steps that need to be taken to perform a surrogate-based optimization have been shown.

9.1 Error determination

In this section a method is presented for the determination of the prediction error of the surrogate models. Only when the prediction error is defined, comparisons between different methods can be made. For a conceptual design study it is impossible to know beforehand what the optimum configuration is.

The developed methods could only be compared with a known reference global optimum value. The global optimum of the case study had to be found before any comparisons between different methods could be made. One solution is to use the *MATLAB* function Multistart as explained in Section 8.2.

A different solution is to perform an SBO using a very large Design Of Experiments, in this case consisting of 1000 LHS points. All points were compared and the highest objective function value that satisfies the constraints was selected. This point can be used as a reference value. All future optimization should find this objective function value or higher. When an optimization finds a lower optimum value, it can be concluded that the global optimum is not found.

A different method to determine the quality of a surrogate model is not to look at the optimum objective function value, but to look at the prediction errors of the surrogate model. The only information that is available are the values calculated at the design points of the DOE and the estimations using the surrogate models. The difference between the predicted value and the actual value can be determined for every reference point and the individual errors can be added to obtain a total estimation error value. To compare

different surrogate models based on different DOEs it is better to use the same set of reference value to determine the error. The error is then a measure for how good the reference data points are predicted by the surrogate models. A set of 100 points is randomly selected from the reference data set of 1000 points and is used as a validation set. The error between this validation set and any surrogate model can be determined and compared to any other method. In this manner, different surrogate modelling techniques, DOE methods and DOE sizes can be compared.

For the error between the surrogate model and the validation set, five different error metrics have been used. All metrics have different characteristics, by evaluating multiple error metrics, more information concerning the quality of a surrogate model is gathered. The metrics considered are the *root mean square error* (RMSE) (Equation 9.1), *mean absolute percentage error* (MAPE) (Equation 9.2), *10% quantile*, $r^2 - method$ (Equation 9.3) and the *relative average absolute error* (RAAE) (Equation 9.4) [57] [58] [59]. The values have been used to compare the quality of different surrogate models.

Root mean square error

The root mean square error method calculates the distance between every predicted point and the true value. The distances are squared and the mean is calculated. A low RMSE indicates a close representation of the true function by the surrogate model.

$$\mathrm{RMSE} = \sqrt{\frac{\sum \left(y_{val} - y_{pred}\right)^2}{n}} \tag{9.1}$$

Where y_{pred} is the value predicted by the surrogate model and y_{val} is the value of the validation point. The number of validation points, in this case 100, is represented by n.

Mean absolute percentage error

The MAPE determines the percentage difference between the predicted value and the true value. The mean of this value is calculated. A low MAPE indicates a high quality surrogate model.

$$\mathrm{MAPE} = \frac{1}{n} \sum \left(\frac{\mid y_{val} - y_{pred} \mid}{y_{val}} \right) \cdot 100 \tag{9.2}$$

10% quantile

The relative prediction error is determined by dividing the difference between the predicted value and the true value by the true value. The 10% quantile method determines the percentage of data points that have a relative prediction error below 10%. A high value indicates a good representation of the surrogate model.

r-squared

The r-squared method is another measure to quantify the quality of the fit. A value close to 1 indicates a high quality surrogate model. The following equation was used to determine the r-squared value of the different surrogate models.

$$r^2 = 1 - \frac{\sum \left(y_{val} - y_{pred}\right)^2}{\sum \left(y_{val} - \frac{\sum y_{val}}{n}\right)} \tag{9.3}$$

Relative average absolute error

The final error metric that has been used is the RAAE. The sum of the errors is divided by the standard deviation (σ) of the true value. A low RAAE value indicates a high quality surrogate model.

$$\text{RAAE} = \frac{\frac{1}{n}\sum \mid \left(y_{val} - y_{pred}\right) \mid}{\sigma\left(y_{val}\right)} \tag{9.4}$$

9.2 Surrogate-based optimization architecture

The principle of surrogate modelling was explained in Section 2.2, an XDSM of the method is presented below. The start point of the SBO is the creation of the DOE. Several methods have been presented. Exact multidisciplinary analyses are performed at the design points ($\hat{x}$). The design points and calculated results ($\hat{y}$) are used to create a surrogate model. Different methods are presented in 2.2. In the case study all multidisciplinary evaluations have been replaced by surrogate models. A separate surrogate model is created for the objective function (y'_{obj}) and for the two constraint functions (Von Mises stress and flutter speed, y'_{con}). The SBO has the same architecture as the MDO as can be seen in Figure 9.1 and the optimization works in the same manner.

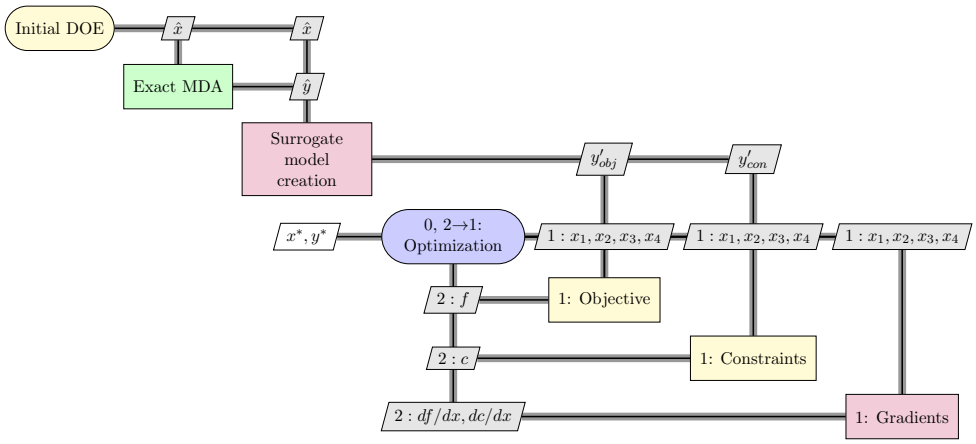


Figure 9.1: XDSM of the surrogate-based optimization

Table 9.1: XDSM variable definition

name	result
$\hat{x}$	Design points
$\hat{y}$	MDA results at design points
y'_{obj}	Surrogate model for objective function
y'_{con}	Surrogate models for constraint functions
x^*	Optimal solution design variables
y^*	Final result
x_1	Root chord length
x_2	Wing semi-span
x_3	Root thickness
x_4	Span wise strut location
f	Objective function
c	Constraint functions

The creation of the three different surrogate models and the selection of the suitable methods are presented in the following sections.

9.3 Surrogate model flutter speed prediction

The first surrogate model that has been evaluated is the flutter speed prediction. To find out what the best surrogate modelling method is to use for flutter speed calculations, a comparison between the models presented earlier is made. A combination of all DOE methods described earlier consisting of 200 points is used. The results are shown in Table 9.2. The errors are determined using the reference data set of 10 points. The table shows the different errors between the validation set and the surrogate models for the flutter speed. As expected the interpolation (Kriging) methods perform relatively poor. These methods are forced to include exact solutions which means the method will have problems with finding a smooth function. All jumps present (as the example shown in Figure 7.7) in the design points are exactly represented. The surrogate model based on the artificial neural network method shows the lowest errors for the flutter speed in all error metrics as was identified in Section 2.3.

Table 9.2: Flutter speed surrogate model prediction errors

Model	**MAPE**	**10% quantile**	**RMSE**	r^2	**RAAE**
Poly1	5.635	80	0.091	0.743	0.388
Poly2	4.797	90	0.081	0.798	0.337
Poly3	5.933	90	0.083	0.786	0.391
KrigingcG	7.734	80	0.136	0.424	0.434
KriginglC	8.029	90	0.122	0.536	0.491
ANN	**4.699**	**100**	**0.064**	**0.872**	**0.313**

9.4 Surrogate model Von Mises stress

The same reference data set as explained in Section 9.1 is used. It can be seen that the range and Von Mises prediction errors are significantly smaller than the flutter speed prediction errors. Using the same DOE method and size, it is more difficult to represent the flutter speed with a surrogate model than to represent the range and Von Mises stress as already indicated in Section 7.5.

For the Von Mises stress surrogate model it is found that the Kriging method using a linear method and cubic splines gives the most reliable result. A comparison between methods is shown in Table 9.3. The Von Mises stress function is a relatively smooth function, which can be well represented by a Kriging method.

Table 9.3: Von Mises stress surrogate model prediction errors

Model	MAPE	10% quantile	RMSE	r^2	RAAE
Poly1	4.880	**100**	0.033	0.929	0.228
Poly2	2.959	**100**	0.026	0.958	0.132
Poly3	1.630	**100**	0.011	0.993	0.073
KrigingcG	1.626	**100**	0.011	0.993	0.068
KriginglC	**0.907**	**100**	**0.007**	**0.997**	**0.043**
ANN	1.199	**100**	0.008	0.996	0.053

9.5 Surrogate model range

The final surrogate model was created for replacing the range determination. Again the same DOE size and method are used as before. For the range prediction the errors are even smaller than the errors for the Von Mises stress prediction. The same Kriging method is best in representing the range determination as can be seen in Table 9.4.

Table 9.4: Range surrogate model prediction errors

Model	MAPE	10% quantile	RMSE	r^2	RAAE
Poly1	7.907	80	0.057	0.955	0.175
Poly2	1.826	**100**	0.015	0.997	0.046
Poly3	0.399	**100**	0.004	1.000	0.010
KrigingcG	0.220	**100**	0.003	1.000	0.006
KriginglC	**0.115**	**100**	**0.001**	**1.000**	**0.003**
ANN	0.827	**100**	0.007	0.999	0.022

9.6 Design of experiments results

The range prediction error (RMSE) using the Kriging model is shwon in Figure 9.2. The determination of the error is explained in Section 9.1. Increasing the number of data points for the design of experiments decreases the error of the surrogate model. However,

an increase in DOE size increases the overall computational time. In the case presented here, the creation of the DOE with 25 points took 0.6 hours, whilst the creation of a DOE with 240 points took 6 hours[1]. After approximately 100-150 points, an increase in DOE points does not lead to significant error reductions as is shown in Figure 9.2. Depending on the required accuracy and available computational time, a suitable DOE size can be selected. An overview of the SBO results versus computational time is provided in Table 9.5.

The difference in prediction error between surrogate models based on a DOE of only LHS points versus surrogate models based on a DOE using a combination of methods is also shown in Figure 9.2. The RMSE is smaller for surrogate models using the combination of DOE methods than for a DOE created using only the LHS method.

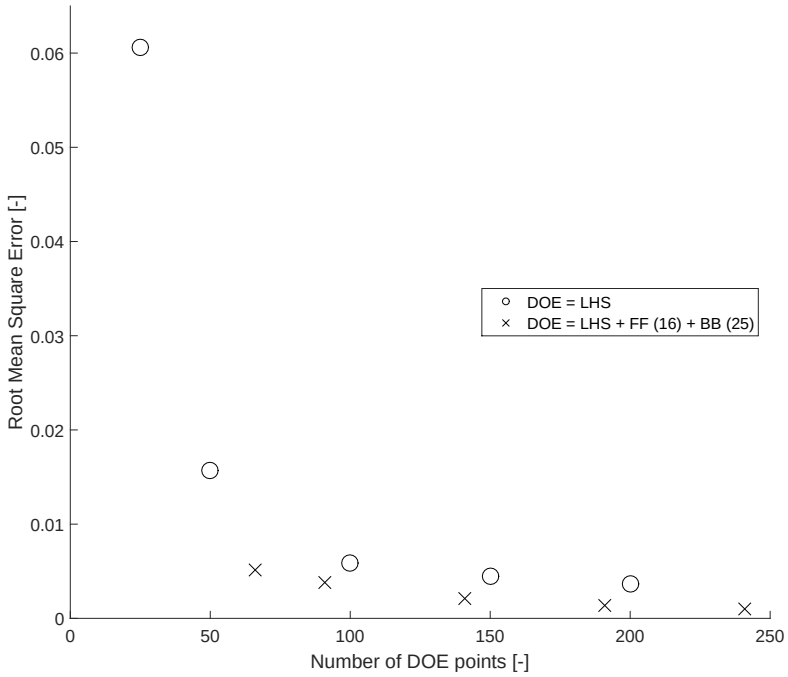


Figure 9.2: Range prediction error (RMSE) for different DOE sizes

9.7 Prediction error comparison

The differences in prediction error between the three different surrogate models are shown in Figure 9.3. As was already presented in Table 9.2 to Table 9.4 the range and Von Mises functions are relatively smooth compared to the flutter speed function. This resulted in much larger prediction errors for the flutter speed. The low quality of the flutter speed surrogate model caused problems in surrogate-based optimizations as is shown in Table 9.5.

[1] the evaluations are performed on a computer with an Intel Core i5-4200M 2.50 GHz processor with 4 GB of RAM memory

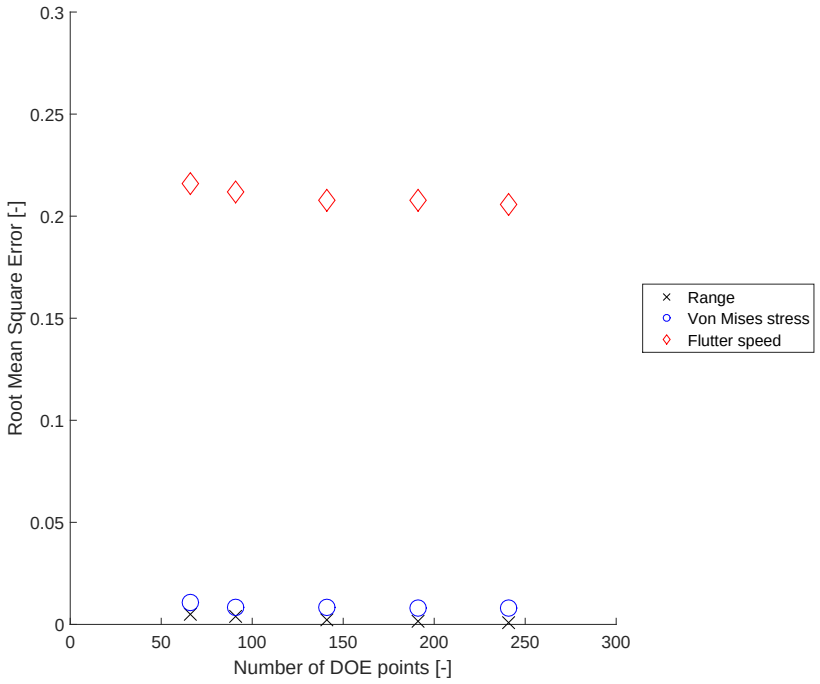


Figure 9.3: Prediction error (RMSE) for different DOE sizes

9.8 Adaptive surrogate modelling

Low prediction errors are needed for the successful implementation of surrogate models in the optimization. Especially for the flutter speed this requires a large DOE, which is computationally expensive. One of the main arguments for using surrogate models was the reduction in computational time. A different method was investigated to use surrogate models using a lower number of expensive evaluations.

The principle behind adaptive surrogate modelling was explained in Figure 2.12. An XDSM representing the case study is shown in Figure 9.4. This XDSM is almost equal to the XDSM presented for the SBO but includes the updating of the DOE. The optimum design point (x^*, y^*) is added to the DOE until the solution is converged. Convergence is checked by the first optimization module and the final converged solution is represented in the XDSM by x^{**} and y^{**}.

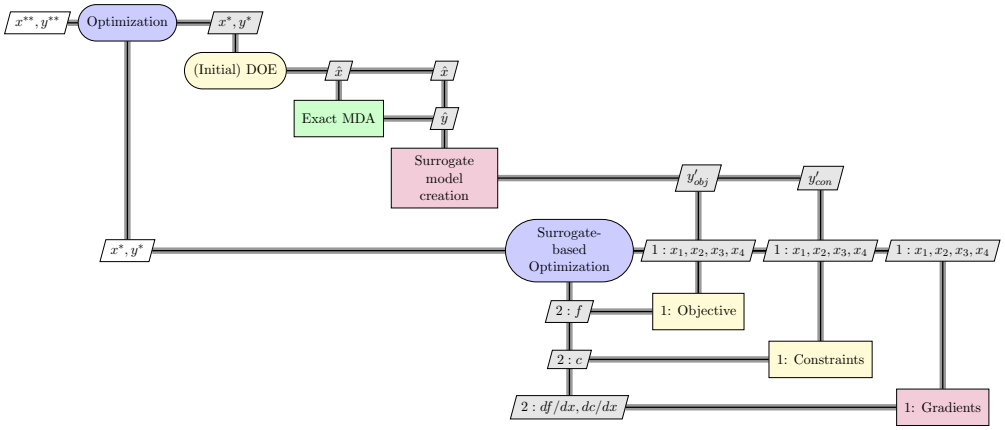


Figure 9.4: XDSM of the adaptive surrogate-based optimization

Three DOEs of different size were selected as a starting point to investigate the influence of the number of experiment points on the final result. It is preferred to start with a DOE that is as small as possible to limit the computational time. Every new SBO uses the optimum results of the previous optimization as the starting point for the following optimization.

Figure 9.5 shows the results found using three different DOEs as starting data. The first DOE was a set of 10 LHS points, the others made use of 25 and 50 LHS points. The first optimization using the smallest DOE started off with large prediction errors. This was as expected since the surrogate models were not accurate enough for small DOEs as shown in Figure 9.2. The DOEs also resulted in significant prediction errors during the first optimizations. All three adaptive optimizations converged to the same predicted range value. A difference between two consecutive optimizations smaller than 0.01% was defined as convergence criterion. This value could be used as a stopping criterion in the adaptive surrogate-based optimization. When this value is reached it means that the optimizer finds practically the same point. Minor differences can come from computational or rounding errors.

Starting off with a DOE of 10 points, convergence was reached after 12 adaptations. The starting DOE of 25 points led to convergence after 11 adaptations, whilst the largest DOE converged after only 7 adaptations. The convergence is indicated in the figures by the red markers. This trend shows that when the original DOE is larger, fewer adaptations are needed to find a converged solution. However, for the total computational time the overall number of evaluations needs to be considered. This shows that in this case study it was possible to start off with a very small DOE and adapt the surrogate model by adding the found optimum points, even though the initial surrogate model created using the small DOE itself has a very large prediction error. A comparison of these methods compared to "standard" SBO methods can be performed based on the results presented in Table 9.5.

Convergence was obtained quickly since the objective function does not need many DOE points to create a surrogate model of acceptable quality. The flutter speed has however much larger prediction errors. When the flutter constraint is not active, the predicted value does not have to be very precise since this has no influence on the resulting range.

Problems will occur when the flutter constraint is erroneously predicted to be active. This can be seen in the figure at update number 8 for the surrogate model based on the smallest DOE. At this point the flutter constraint was predicted to be active, resulting in a very low range value ($3.85E+06m$, outside of the scale of Figure 9.5). The predicted optimum point is added and the true flutter speed is calculated. The true flutter value was then calculated and the constraint was found to be inactive. A much higher range was predicted in the next iteration since the flutter speed was found to be inactive. The methods presented in this section did not have problems with converging to a local optimum which might be the result of erroneous predictions. For this reason no further actions to reduce the risk to this have been investigated or implemented. This should however be done when premature convergence is present.

The blue markers in the figure indicate active flutter constraints. From these results it can be concluded that the converged solutions are found with inactive flutter constraints. This would explain the quick convergence of the results, since the range prediction was relatively accurate as was shown in Figure 9.3. Other optimization methods found an inactive flutter constraint at the optimum as well (see Table 9.5). This indicates that different case studies should be applied to the SBO methods to further investigate the quality of the developed methods.

A comparison between adaptive surrogate-based optimization and surrogate-based optimizations using different DOE sizes is presented in Table 9.5.

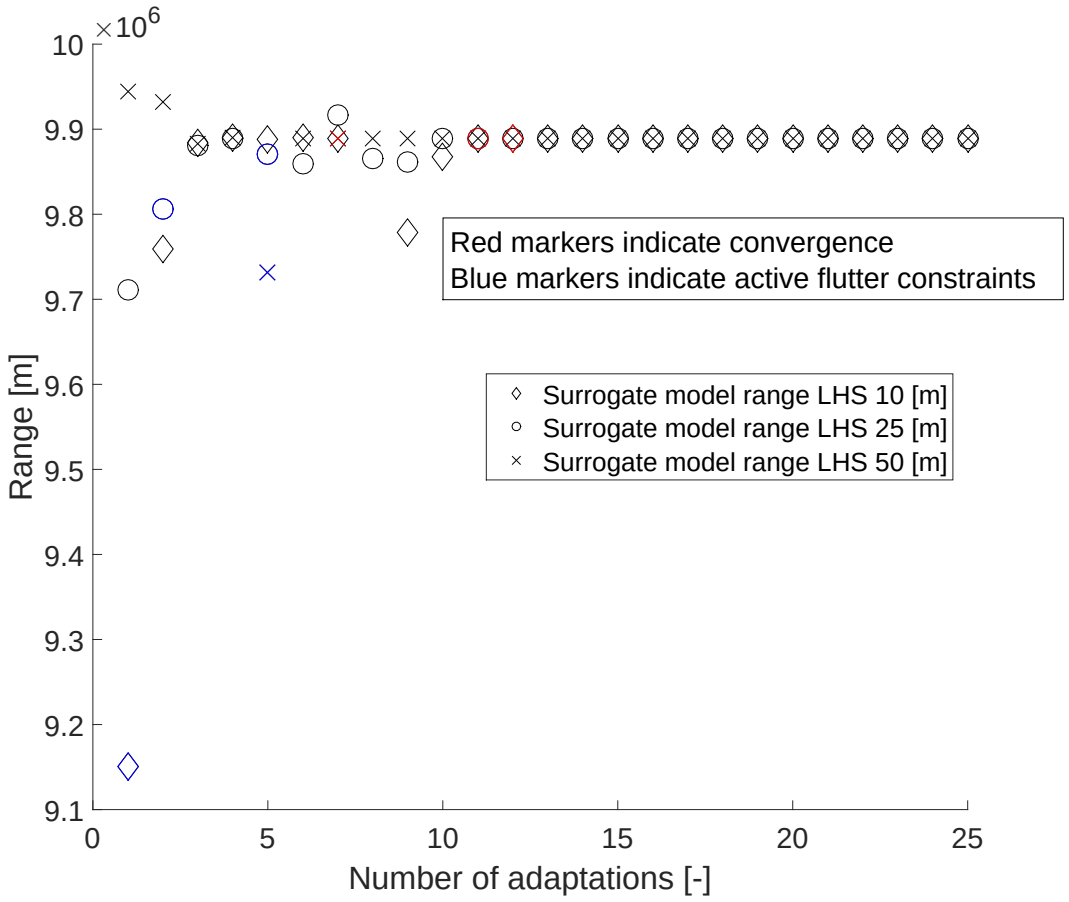


Figure 9.5: Adaptive SBO LHS 25 and 50

9.9 Results

Several different optimizations methods have been presented in the previous sections. All methods have their own characteristics. Using the input variables, three different surrogate models were created. One model represented the objective function, in this case range, and was represented using a Kriging model. The Von Mises stress was used as a constraint in the optimization and was also represented by a Kriging model. The ANN model was used for the flutter speed since this model produced the best results.

Values indicated in the table by MDO are found by applying the multidisciplinary design optimization on the full disciplines. SBO refers to an MDO using a surrogate model. All results in Table 9.5 are sorted on total number of non-surrogate MDA analyses. The following components are shown in the table:

- **Total number of MDA analyses** These numbers indicate the total number of multidisciplinary evaluations that have been performed to produce the presented result. For surrogate-based optimizations, the value includes the number of evaluations performed to create the DOE.

- **Range** These values represent the final range value that has been found.
- **Difference with reference** The range results have been compared with the reference result to determine the percentage difference.
- **Computational time** The total time used for the calculation of the result is presented in this column[2].
- **Chord length, semi-span, root thickness and strut location** The final design variable results are presented in these columns.
- **Von Mises stress** The values of the Von Mises stress constraint function are shown in this column.
- **Flutter speed** The flutter speed value is presented here.
- **RMSE flutter** RMSE results of the flutter speed prediction are shown in the final column.

9.9.1 MDO using 10 starting points

As explained in Chapter 8, the MDO result depended on the starting point of the optimization. Using multiple randomly generated starting points it was found that when 10 points were selected, the optimizer was able to find the global optimum. The procedure of using 10 starting points was time consuming (15.7 hours for this case study), which was not a very efficient strategy. The resulting range was used as a reference value, surrogate modelling methods were compared to this value in both accuracy of the final result as in computational time.

9.9.2 Best feasible point in large DOE

A reference LHS DOE consisting of 1000 points was created. The constraint values of all DOE points have been evaluated. The largest feasible point has been used as a reference for the reference point found using the MDO as described in the previous section. The large number of design points used for this reference data set has resulted in a dense representation of the design space. No optimizations have been performed with this data set, for this reason the found value was lower than the result of the optimization.

9.9.3 SBO using different DOEs

As explained earlier, Surrogate-based optimization methods might be able to obtain accurate results at a much lower computational cost than optimizations performed without using surrogate models. Different DOE methods and sizes have been discussed in the previous sections. An increase in number of DOE points increases the reliability of the result at the cost of an increased computational time as is shown in Table 9.5.

[2]The evaluations are performed on a computer with an Intel Core i5-4200M 2.50 GHz processor with 4 GB of RAM memory

The SBO using an LHS of 10 design points has found an optimum with a large error with respect to the reference value. This can be the result of large errors in the surrogate models, as was shown in Figure 9.2.

The SBO using the two level full factorial DOE method also found an optimum value with a relatively large difference with respect to the reference value. The flutter speed that has been predicted is also very high. The surrogate model of the flutter speed has a very large RMSE. Furthermore, the DOE size is too small to create a reliable surrogate model.

9.9.4 Adaptive SBO

The procedure was explained in Section 9.8. The three different DOEs described in that section have been evaluated and are shown in the table. Applying a limited number of adaptations can still lead to an accurate result at a low computational cost, despite the large prediction errors of the initial surrogate model.

9.9.5 Optimum variables

The optimum values of the variables are almost equal for all optimization results. The strut location is the variable that varies the most between different studies. The influence of the strut was the most difficult to predict by the surrogate models. The most important contribution of the strut location was its effect on the flutter speed. But since this flutter speed was difficult to predict by the surrogate models, this has resulted in different optimum strut locations. A high fidelity wing mass estimation tool would take the effect of the strut on a possible wing weight reduction into account.

The chord length and wing span values are found close to their upper bounds ($4.62m$ and $27.1m$) of the optimization. The root thickness was always found to be precisely on the upper bound for all performed optimizations.

Table 9.5: Efficiency and quality comparison of different optimization methods

Model	Total number of MDA analyses (−)	Range (m)	Difference with reference (%)	Computational time (h)	Chord length (m)	Semi span (m)	Root thickness (m)	Strut location (−)	Predicted range (m)	Von Mises stress (MPa)	Flutter speed (m/s)	RMSE flutter (−)
MDO using 10 starting points	599	9.96E+06	0	15.7	4.62	26.99	0.45	0.43	-	1.38E+08	293.08	-
Best point large DOE	1000	9.73E+06	-2.31	26.2	4.60	26.95	0.45	0.67	-	1.60E+08	357.12	-
SBO using LHS	10	7.33E+06	-26.40	0.3	4.62	27.00	0.45	0.21	7.33E+06	1.02E+08	276.98	0.40
SBO using two level FF	16	8.40E+06	-15.68	0.4	4.62	26.99	0.45	0.21	8.40E+06	1.59E+08	576.02	1.66
SBO using BB	25	9.45E+06	-5.10	0.6	4.61	26.96	0.45	0.39	9.45E+06	1.20E+08	299.00	0.22
SBO using LHS	25	9.28E+06	-6.83	0.6	4.61	26.98	0.45	0.55	9.28E+06	1.44E+08	292.25	0.28
Adaptive SBO start with 10 LHS	26	9.88E+06	-0.67	0.7	4.62	26.98	0.45	0.54	9.89E+06	1.47E+08	346.24	-
Adaptive SBO start with 25 LHS	36	9.87E+06	-0.67	0.9	4.62	26.98	0.45	0.54	9.89E+06	1.47E+08	346.11	-
SBO using LHS	50	9.84E+06	-1.19	1.3	4.62	27.00	0.45	0.46	9.84E+06	1.37E+08	307.86	0.24
Adaptive SBO start with 50 LHS	57	9.87E+06	-0.67	1.5	4.62	26.99	0.45	0.54	9.89E+06	1.47E+08	345.58	-
SBO using HS + BB + FF	91	9.96E+06	-0.31	2.4	4.62	27.00	0.45	0.36	9.93E+06	1.36E+08	272.91	0.26

Chapter 10

Conclusions and recommendations

10.1 Conclusions

One goal of this thesis was to implement a flutter speed constraint in a Multidisciplinary Design Optimization. Another goal was to investigate the possibilities to apply surrogate-based optimization methods.

It was shown that it was possible to implement flutter speed calculations in an MDO using *AMLoad* and *NASTRAN*. The flutter speed is the quantity that was fed to the optimizer. This manner of implementation enabled gradient-based optimization methods. Geometrical changes applied to the aircraft model resulted in a different flutter speed for that particular aircraft geometry. The correlations between the geometry change and change in flutter speed were identified by the optimizer and used to steer the optimizer in the right direction of the design space.

Surrogate-based optimization methods were implemented to enable relatively fast calculations. SBO studies can be performed within minutes instead of hours. Several different studies have been performed and presented in the previous chapters. The following can be concluded from this research.

Flutter constraint

High aspect ratio concepts urge the need for flutter determination during the early stages of aircraft conceptual design. Optimization studies were performed taking the flutter constraint into account during optimization using the p-k-method of *NASTRAN* implemented in *AMLoad*. The method can be used to determine the optimum strut-braced wing aircraft configuration.

Design of experiments method

Several methods to create a Design Of Experiments have been presented. Methods placing design points at the corners (two-level full factorial) and edges of the design space

(Box-Behnken) were combined with a method that places design points distributed over the inner parts of the design space (Latin Hypercube Sampling). It was shown that a combination of these methods produces a DOE that can be used to create surrogate models with the lowest prediction errors for this case study.

Design of experiments size

An increase in the number of DOE points has shown to enable the creation of surrogate models with low prediction errors. With an increase in the number of design points, the total computational time increased as well. After approximately 150 design points, further increasing this number led to only a very small decrease in prediction error in the presented case study. Depending on the required accuracy of the result and the available computational power, a suitable DOE size should be selected.

Surrogate modelling method for flutter speed

The flutter speed showed non-smooth behaviour to geometry changes. Small changes in aspect ratio showed to have a large effect on the flutter speed. Non-smooth behaviour was difficult to represent by surrogate models and large predictions errors occurred. A surrogate model created using artificial neural networks was shown to be most suitable to represent the flutter speed. The ability of the model to train itself allowed the surrogate model to handle the non-linearities. Kriging and polynomial methods were less suitable and resulted in larger prediction errors.

Adaptive surrogate modelling methods

Adaptive surrogate modelling methods have been presented as an efficient method to obtain an optimum result. In this method a relatively small DOE was used to create a surrogate model. A surrogate-based optimization was performed after which the optimum point was selected and the actual objective function value was calculated. This new point was added to the DOE to increase the accuracy of the surrogate model in the region of the predicted result. The optimum point was then used as a starting point for the next surrogate-based optimization. This procedure was repeated until the solution was converged. A low number of disciplinary evaluations was needed relative to methods that do not use adaptations of the surrogate model.

The final result of the adaptive SBO methods is less than 1% off the reference result found using the highly expensive Multistart reference evaluations. A method with a small initial DOE and several model updates showed to be more efficient for this case study then a method with a larger initial DOE. This has to be further investigated using different case studies.

10.2 Recommendations

Recommendations for further study and improvements of the methods are presented here. The recommendations are split into two different categories. The first set of recommendations apply to improvements of the developed methods. The second set of recommendations could be followed when a more accurate SBW aircraft concept is to be designed using the presented methods.

10.2.1 Improving methods

Apply different case studies

The applied case study found optimum configurations with flutter speeds much higher than the constraint values. This indicates that the used case study was not the most suitable study to assess the proper functioning of the flutter constraint. A different case study could be selected to further investigate the quality of the developed methods.

Another recommendation is to use a case study that has the global optimum in the center region of the design space to further evaluate the influence of the different DOE methods.

Use mode tracking to implement damping values in the constraint

The switching of different critical flutter modes proved to complicate the prediction of the flutter speed. The optimizer only has information on the critical flutter mode with the lowest flutter speed, but has no information on the mode itself. The optimizer would be able to track the influence of a geometry variation on every modes separately if these modes could be tracked. The optimizer then 'knows' what the flutter speed is of every individual mode. Instead of using the flutter speed as the constraint in the optimization, the damping values of individual modes could then be used. This would allow for more precise predictions of the influence of the geometry variations on the flutter speeds of different modes. Surrogate models should then be created for the damping values of every flutter mode separately.

Prevent premature convergence

When the proposed adaptive surrogate-based optimization method is applied, there is a risk of premature convergence. A small DOE was used to find the optimum design point after which the surrogate model was updated at that location. As a result of this, the optimizer might be increasing the accuracy of the surrogate model in the vicinity of a local optimum and will not be able to find the global optimum. To solve this problem, instead of only adding the optimum point to the new DOE, it is possible to also add points that are in a completely different location of the design space. In this manner it can be ensured that the local optimum is in fact also the global optimum. More research could be performed on developing methods to prevent premature convergence.

Reduce of number of optimizations

In the proposed adaptive method, with every iteration step only one point is added to the DOE and a new SBO is performed. In the presented case, performing a surrogate-based optimization is cheap measured in terms of calculation time. It might however be possible that the number of optimizations needs to be decreased (when for example the number of optimization licenses is limited or optimizations are expensive in terms of money). To reduce the number of optimizations, a different method can be applied. Instead of adding only the optimum point to the DOE, it is also possible to add multiple points in the vicinity of the optimum simultaneously. Using this method, it is possible to converge to the final solution with a lower number of optimizations.

Improve convergence criteria of adaptive surrogate-based optimization

The adaptive surrogate-based optimization presented in 9.8 made use of a stopping criterion. Based on the results found this value was set at 0.01%. This value might be changed for other applications since it has an influence on the computational time of the optimization. More research could be performed on a suitable value.

10.2.2 Improve strut-braced wing conceptual design

The focus in this study was on efficient implementation of a flutter module and on the application surrogate-based optimization. Simplifications were put on several modules of the conceptual design to keep the computational time limited and to minimize the number of designs variables. When a conceptual design is to be performed with a higher quality of the final result, the following modifications are recommended.

Increase the number of design variables

The number of design variables in the case study was limited to only four wing design variables. This had some serious implications on the final result. Different methods have been developed in this research and results were compared. This was enabled by the low number of variables which kept the problem clear and the computational times limited. The presented methods can be applied to case studies with a higher number of design variables to remove the simplifications.

Add aerodynamic forces on the strut

A recommendation influencing the drag prediction is to include the presence of aerodynamic forces on the strut. Especially at higher angles of attack, the strut will have an influence on the total lift created. Furthermore, the interaction between the lifting forces of the strut and wing is important to take into account when a more detailed evaluation of a strut-braced wing is required. These interacting forces can be a source of interference drag and will have an influence on the flutter behaviour of the wing.

Make use of higher fidelity aerodynamics tools

Previous studies have shown that the interference drag between the strut and wing can have a significant contribution to the total drag of the aircraft. When this needs to be investigated, more advanced tools are recommended. CFD calculations could be performed, this is however much more time consuming and might be impractical to implement in an optimization routine.

Consider truss configurations

Truss configurations for the strut have shown possible structural benefits in previous research. Adding extra design variables to include different truss configurations could result in a better optimization result. This however increases the computational time.

Implement a wing weight estimation tool

A wing weight estimation tool should be implemented. Geometry variations should be reflected in a more accurate prediction of the wing mass. Furthermore, load alleviation on the wing by the strut allows for a lighter structure, this should be implemented in the optimization to benefit from the presence of the strut.

Include strut optimization

Together with a wing structure optimization tool, a method should also be implemented to optimize the strut structure for the applied loads. A weight and outer shape estimation result in a more accurate result.

Include tail optimization

The tail of the aircraft was assumed to stay constant during the optimization. A more accurate aircraft concept can be found when the sizing of the tail is taken into account. A larger and heavier aircraft should be reflected in a larger tail size.

References

[1] D. P. Raymer. *Aircraft Design: A Conceptual Approach.* American Institute of Aeronautics and Astronautics, 1992.

[2] M.K. Bradley; T.J. Allen; C. Droney. Subsonic Ultra Green Aircraft Research: Phase II Volume I: Truss braced wing design exploration. Technical Report CR-2015-218704, NASA, 2015.

[3] G. Carrier; O. Atinault; S. Dequand; J.-L. Hantrais-Gervois; C. Liauzun; B. Paluch; A.-M. Rodde; C. Toussaint. Investigation of a strut-braced wing configuration for future commercial transport. In *28th international congress of the aeronautical sciences*, pages 1–10. ICAS Bonn, 2012.

[4] O. Gur; M. Bhatia; J.A. Schetz; W.H. Mason; R.K. Kapania; D.N. Mavris. Design optimization of a truss-braced-wing transonic transport aircraft. *Journal of aircraft*, 47(6):1907–1917, November-December 2010.

[5] N.V. Queipo; R.T. Haftka; W. Shyy; T. Goel; R. Vaidyanathan; P.K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41(1):1–28, January 2005.

[6] M. Mohan Raju; R.K. Srivastava; D. Bisht; H.C. Sharma; A. Kumar. Development of artificial neural-network-based models for the simulation of spring discharge. *Advances in Artificial Intelligence*, (5):1–11, January 2011.

[7] A. Forrester; A. Sobester; A. Keane. *Engineering design via surrogate modelling : a practical guide.* John Wiley & Sons, 2008.

[8] O. Gur; W.H. Mason; J.A. Schetz. Full-configuration drag estimation. *Journal of Aircraft*, 47(4):1356–1367, July-August 2010.

[9] H.J. Hassig. An approximate true damping solution of the flutter equation by determinant iteration. *Journal of Aircraft*, 8(11):885–889, 1971.

[10] G.E.P. Box; D.W. Behnken. Some new three level designs for the study of quantitative variables. *Technometrics*, 2(4):455–475, 1960.

[11] K.G. Bowcutt. A perspective on the future of aerospace vehicle design. In *12th AIAA international space planes and hypersonic systems and technologies*, 2003.

[12] R.K. Agarwal. Environmentally responsible air and ground transportation. In *49th AIAA aerospace sciences meeting including the new horizons forum and aerospace exposition*, 2011.

[13] M. Dareck; C. Edelstenn; T. Ender; E. Fernande; P. Hartman; J. Herteman; M. Kerkloh; I. King; P. Ky; M. Mathieu; et al. Flightpath 2050: Europe's vision for aviation. Directorate-General for Research and Innovation, Directorate-General for Mobility and Transport, 2011.

[14] G.J. Kennedy; G.K.W. Kenway; J.R.R.A. Martins. High aspect ratio wing design: Optimal aerostructural tradeoffs for the next generation of materials. In *Proceedings of the AIAA Science and Technology Forum and Exposition (SciTech), National Harbor, MD*, 2014.

[15] T. Long; L. Liu; L. Peng; Y. Li. Aero-structure coupled optimization of high aspect ratio wing using enhanced adaptive response surface method. In *14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, Indianapolis, Indiana*, 2012.

[16] J.D. Anderson. *Fundamentals of aerodynamics*. McGraw Hill International, 2005.

[17] I. Chakraborty; T. Nam; J.R. Gross; D.N. Mavris; J.A. Schetz; R.K. Kapania. Comparative assessment of strut-braced and truss-braced wing configurations using multidisciplinary design optimization. *Journal of Aircraft*, 52(6):2009–2020, 2015.

[18] E. Sulaeman; R.K. Kapania; R.T. Haftka. Parametric studies of flutter speed in a strut-braced wing. In *Proceeding of the 43rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 2002–1487, 2002.

[19] F.H. Gern; A. Ko; E. Sulaeman; J.F. Gundlach; R.K. Kapania; R.T. Haftka. Multidisciplinary design optimization of a transonic commercial transport with strut-braced wing. *Journal of Aircraft*, 38(6):1006–1014, 2001.

[20] M. Bhatia; R.K. Kapania; M. van Hoek; R.T. Haftka. Structural design of a truss braced wing: potential and challenges. In *50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2009.

[21] A.H. Naghshineh-Pour. Structural optimization and design of a strut-braced wing aircraft. Master's thesis, Faculty of the Virginia Polytechnic Institute and State University, 1998.

[22] A.B. Lambe; J.R.R.A Martins. A unified description of MDO architectures. In *9th World Congress on Structural and Multidisciplinary Optimization*, June 2011.

[23] A.B. Lambe; J.R.R.A Martins. Multidisciplinary design optimization: a survey of architectures. *AIAA Journal*, 51(9):2049–2075, 2013.

[24] K. Zhang; P. Ji; A. Bakar; Z. Han. Multidisciplinary evaluation of truss-braced wing for future green aircraft. In *28th international congress of the aeronautical sciences*, 2012.

[25] G.P. Chiozzotto. Conceptual design method for the wing weight estimation of strut-braced wing aircraft. In *5th CEAS Air & Space Conference*, 2015.

[26] O. Gur; M. Bhatia; W.H. Mason; J.A. Schetz; R.K. Kapania; T. Nam. Development of framework for truss-braced wing conceptual MDO. *Structural and Multidisciplinary Optimization*, 44(2):277–298, 2011.

[27] P.-A. Tétrault; J.A. Schetz; B. Grossman. Numerical prediction of interference drag of strut-surface intersection in transonic flow. *AIAA Journal*, 39(5):857–864, 2001.

[28] F.H. Gern; A.H. Naghshineh-Pour; E. Sulaeman; R.K. Kapania. Structural wing sizing for multidisciplinary design optimization of a strut-braced wing. *Journal of Aircraft*, 38(1):154–163, January-February 2001.

[29] M.K. Bradley; C. Droney. Subsonic Ultra Green Aircraft Research Phase II: N+4 advanced concept development. Technical Report CR-2012-217556, NASA, 2012.

[30] T.J. Allen; B.W. Sexton; M.J. Scott. SUGAR truss braced wing full scale aeroelastic analysis and dynamically scaled wind tunnel model development. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 5–9, 2015.

[31] W. Zhao; R.K. Kapania; J.A. Schetz; J.M. Coggin; T.J. Allen; B.W. Sexton. Nonlinear aeroelastic analysis of SUGAR truss-braced wing wind tunnel model under in-plane loads. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2015.

[32] G.E.P. Box; J.S. Hunter; W.G. Hunter. *Statistics for experimenters: design, innovation, and discovery*, volume 2. Wiley-Interscience New York, 2005.

[33] R.A. Fisher. The arrangement of field experiments. *Journal of the Ministery of Agriculture of Great Britain*, 33:503–513, 1926.

[34] M.D. McKay; R.J. Beckman. A comparison of three methods for selecting values of input variables in the analysis of output from a computer code. *Technometrics*, 21(2), 1979.

[35] A. Forrester; A. Sobester; A. Keane. Recent advances in surrogate-based optimization. *Progress in Aerospace Sciences*, 45(1-3):50–79, January-April 2009.

[36] K.C. Giannakoglou. Design of optimal aerodynamic shapes using stochastic optimization methods and computational intelligence. *Progress in Aerospace Sciences*, 38(1):43–76, January 2002.

[37] J. Sacks; W.J. Welch; T.J. Mitchell; H.P. Wynn. Design and analysis of computer experiments. *Statistical Science*, 4(4):409–423, 1989.

[38] P.R. Caixeta; F.D. Marques. Neural network metamodel-based mdo for wing design considering aeroelastic cocnstraints. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference*, April 2010.

[39] N. Bartoli; I. Kurek; R. Lafage; T. Lefebvre; R. Priem; M.A. Bouhlel; J. Morlier; V. Stilz; R. Regis. Improvement of efficient global optimization with mixture of experts: methodology developments and preliminary results in aircraft wing design. In *17th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference, At Washington DC*, 2016.

[40] A. Elham. *Weight Indexing for Multidisciplinary Design Optimization of Lifting Surfaces*. PhD thesis, Delft University of Technology, 2013.

[41] A. Elham; J. Mariens; M.J.L. van Tooren. Quasi-three-dimensional aerodynamic solver for multidisciplinary design optimization of lifting surfaces. *Journal of Aircraft*, 51(2):547–557, March-April 2014.

[42] M. Drela. An analysis and design system for low reynolds number airfoil aerodynamics. In *Conference on Low Reynolds Number Airfoil Aerodynamics, University of Notre Dame*, 1989.

[43] M.M. Freestone. VGK method for two-dimensional aerofoil sections. ESDU 69028, September 1996.

[44] R.C. Lock; B.R. Williams. Viscous-inviscid interactions in external aerodynamics. *Progress in Aerospace Sciences*, 24(2):51–171, 1987.

[45] C. E. Jobe. Prediction and verification of aerodynamic drag, part I: prediction. *Chapter IV in Thrust and Drag: Its Prediction and Verification, Ed. Eugene, CE*, 98, 1985.

[46] E. Torenbeek. *Synthesis of subsonic airplane design*. Springer Science & Business Media, 1982.

[47] F.H.M. White; I. Corfield. *Viscous fluid flow*, volume 3. McGraw-Hill New York, 2006.

[48] E.J. Hopkins; M. Inouye. An evaluation of theories for predicting turbulent skin friction and heat transfer on flat plates at supersonic and hypersonic mach numbers. *AIAA Journal*, 9(6):993–1003, 1971.

[49] W.H. Mason. Analytic models for technology integration in aircraft design. In *AIAA/AHS/ASEE Aircraft Design, Systems and Operations Conference*, 1990.

[50] G.J.J. Ruijgrok. *Elements of airplane performance*. Delft University press, 1990.

[51] R.C. Hibbeler. *Mechanics of materials*. Pearson, 2010.

[52] E. Ting; K. Reynolds; N Nguyen; J. Totah. Aerodynamic analysis of the truss-braced wing aircraft using vortex-lattice superposition approach. In *32nd AIAA Applied Aerodynamics Conference, Atlanta, GA*, 2014.

[53] T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter, NACA report 496, 1935.

[54] D.H. Hodges; G.A. Pierce. *Introduction to structural dynamics and aeroelasticity*, volume 15. Cambridge University press, 2011.

[55] R.A. Frazer; W.J. Duncan. The flutter of airplane wings. *Reports & Memoranda*, 1155, 1928.

[56] S. J. Hulshoff. *Aeroelasticity course reader*. Delft University press, 2011.

[57] J.P.C. Kleijnen. *Design and analysis of simulation experiments*, volume 20. Springer.

[58] A.S. Kapadia; W. Chan; L.A. Moyé. *Mathematical statistics with applications*. CRC Press, 2005.

[59] S. Makridakis; A. Andersen; R. Carbone; R. Fildes; M. Hibon; R. Lewandowski; J. Newton; E. Parzen; R. Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of forecasting*, 1(2):111–153, 1982.

Appendix A

Aerodynamic tools comparison

A.1 Friction validation

The original program *Friction* is developed in *FORTRAN* code. Later versions of Friction are created in *MATLAB* coding. The source code of these scripts is available to use, which made the implementation of Friction in the overall optimization program simple, since that was also performed in *MATLAB* (as well as AMLoad (section 4.1.2)). To ensure this version produced the same results as the original *FORTRAN* program, a comparison was made. The comparison is based on a Fokker 100 wing at cruise conditions. The Mach number is plotted against the form and friction drag coefficient in Figure A.1. As can be seen, the results are almost equal, differences are the result of rounding errors. Based on this comparison it can be concluded that the *MATLAB* version of *Friction* gives the same results as the validated *FORTRAN* code version.

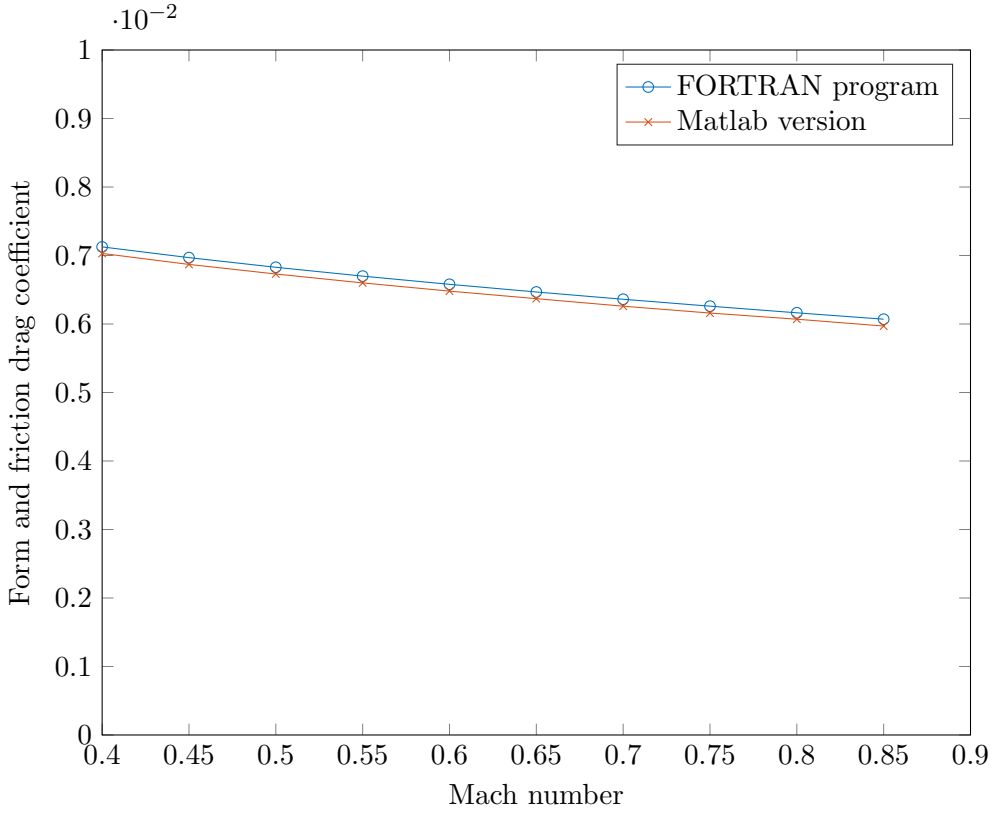


Figure A.1: Fortran results compared with Matlab results

A.2 NASTRAN vs AVL

As explained earlier, *AMLoad* makes use of *NASTRAN* for VLM calculations. TThe same method was used in *AVL* so no large differences in the results were expected. To investigate the differences in results between the two tools, the lift coefficient of a simple wing is plotted against Mach number for a range of different angles of attack. For this comparison the drag of a simplified wing with a root chord length of $4m$, a tip chord length of $2.5m$ and a span of $11m$ was calculated at $M = 0.8$ and at an altitude of $10.000m$. The results are shown in Figure A.2.

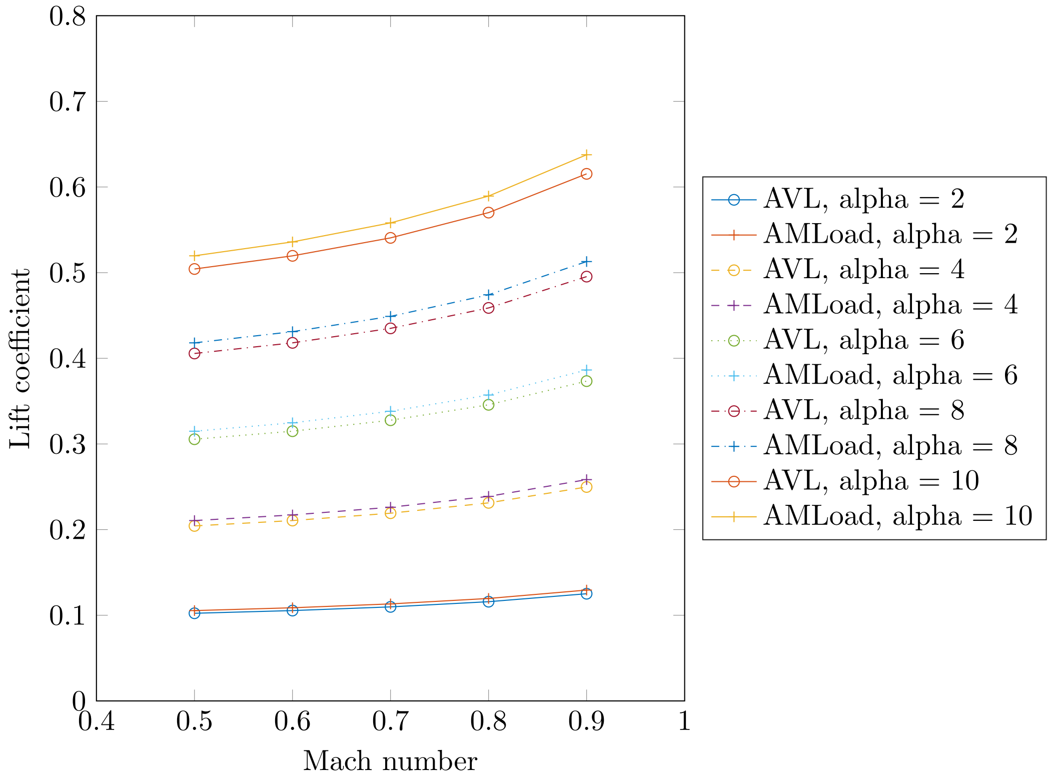


Figure A.2: C_L versus M for AVL and AMLoad

Small differences are found between the two tools. This absolute difference increases as the angle of attack is increasing, the relative difference ranges from $3 - 4\%$ for all data points. The small differences between the tools could come from differences in number of aerodynamic panels used, rounding errors, etc. The error is however small enough to assume that both tools give the same results and *AMLoad* could be used in future calculations instead of *AVL*.

A.3 Friction versus Q3D

The input files used to create Figure 4.4 are presented in this section. An input file for *Q3D* and an input file for *Friction* are shown.

A.3.1 Input file Q3D

```
%  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%      Quasi-3D aerodynamic solver
%
%         A. Elham, J. Mariens
%
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

```
% edits made by A. Lambers
% automatic loading of CST airfoil data

% OUTPUT DESCRIPTION:

% Res.Alpha    = Wing angle of attack
% Res.CLwing   = Total wing lift coefficient
% Res.CDwing   = Total wing drag coefficient
% Res.Wing.aero.Flight_cond = flight conditions including angle of attack
%                             (alpha), sideslip angle (beta),
%                              Mach number (M), airspeed (V) and air
%                              density (rho)
% Res.Wing    = Spanwise distribution of aerodynamic and geometrical
%               properties of wing
%               For example plot(Res.Wing,Yst,Res.Wing.cl) plots spanwise
%               distribution of cl
% Res.Section   = aerodynamic coefficients of 2D sections
%%

clear all
close all
clc
tic;
fprintf('Calculation running..\n')

%% Aerodynamic solver setting

% Wing planform geometry
%                  x     y      z    chord(m)     twist angle (deg)
AC.Wing.Geom      = [0     0      0     5.6   0;
                     2.34  4.6    0.2   3.6   0;
                     5.5   14.04  0.61  1.26  0];
% Wing incidence angle (degree)
AC.Wing.inc   = 0;

% Airfoil coefficients input matrix
AC.Wing.Airfoils = [ 0.1700  0.1000  0.2468  0.0982  0.1242
                    -0.1498 -0.1028 -0.2655 -0.1142 -0.0574 ;
                     0.1850  0.1132  0.2967  0.1278  0.2244
                    -0.1273 -0.0021 -0.3089 -0.0648  0.0068;
                     0.2127  0.0750  0.2441  0.1179  0.1461
                    -0.1304  0.0686 -0.2754  0.0533 -0.1333];

AC.Wing.eta = [ 0      /14.04  ;
                4.6    /14.04  ;
                14.04  /14.04  ];
```

```
% Viscous vs inviscid
AC.Visc  = 1;              % 0 for inviscid and 1 for viscous analysis

% Flight Condition
%AC.Aero.V     = 210;      % flight speed (m/s)
AC.Aero.rho   = 0.41351;  % air density  (kg/m3)
AC.Aero.alt   = 11000;    % flight altitude (m)
%AC.Aero.Re    = 1.5465e7; % reynolds number (based on MAC)
AC.Aero.M     = 0.77;     % flight Mach number
AC.Aero.CL    = 0.2;      % lift coefficient - comment this line to run
                          %     the code for given alpha%
%AC.Aero.Alpha = 5;        % angle of attack -  comment this line to run
                          %     the code for given cl

%% Results
%printing the desired results
Res = Q3D_solver(AC);
fprintf('done calculating, plotting the requested results\n')
M = AC.Aero.M
if AC.Visc == 1
    CDwing = Res.CDwing
else
    CDiwing = Res.CDiwing
end
fclose all
toc; %end of timing
beep %beep sound when calculation is finished
```

A.3.2 Input file Friction

```
%% input file for friction program
% Fokker 100 wing data
 fprintf('Creating input data for Fokker 100 wing, and calculate results\n');
 SREF = 42.76;    %reference area, wing area chosen for F100
 Scale = 1;
 N = 1;           %only wing, no body
 inmd=0;          %input 0: Mach and alt, input 1: Mach& Re per unit length
 k = 0;
 swet(1) = 87.28; %wetted area 2.0*(1+0.2*TC(1))*SREF
 refl(1) = 3.8;   %reference length, MAC chosen
 TC(1) = 0.11;    %t/c wing
 Icode(1) = 0;    %0 = planar surface, 1 = body of revolution
 Ftrans(1) = 0.05;%0 means all turbulent, 1 all laminar,
                  %in between gives transition location
```

Appendix B

Mode switch

An example of the switching of critical modes is shown in this section. This detailed investigation was conducted on the results presented in Figure 7.7. The first figure shows the damping value of the wing mode (mode 9 in this case) is just under the value of 0.03. This means that this modes can be neglected (see 7.2) and the tail mode (mode 15 in this case) is active but with a high flutter speed. In the next figure the damping is just over 0.03, which identifies mode 9 as critical.

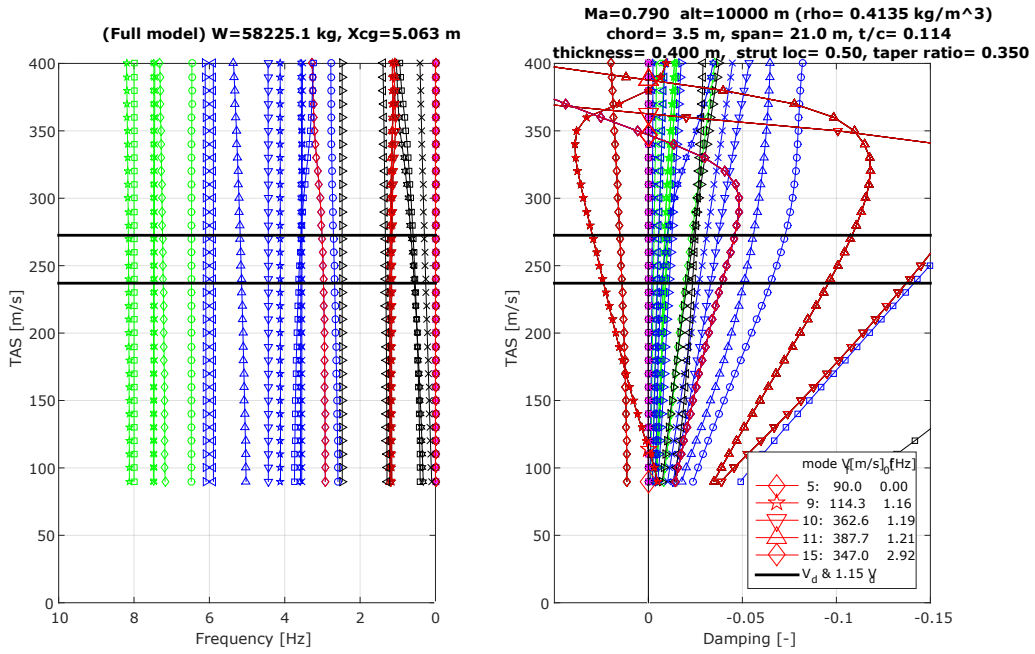


Figure B.1: Flutter modes showing tail as critical mode

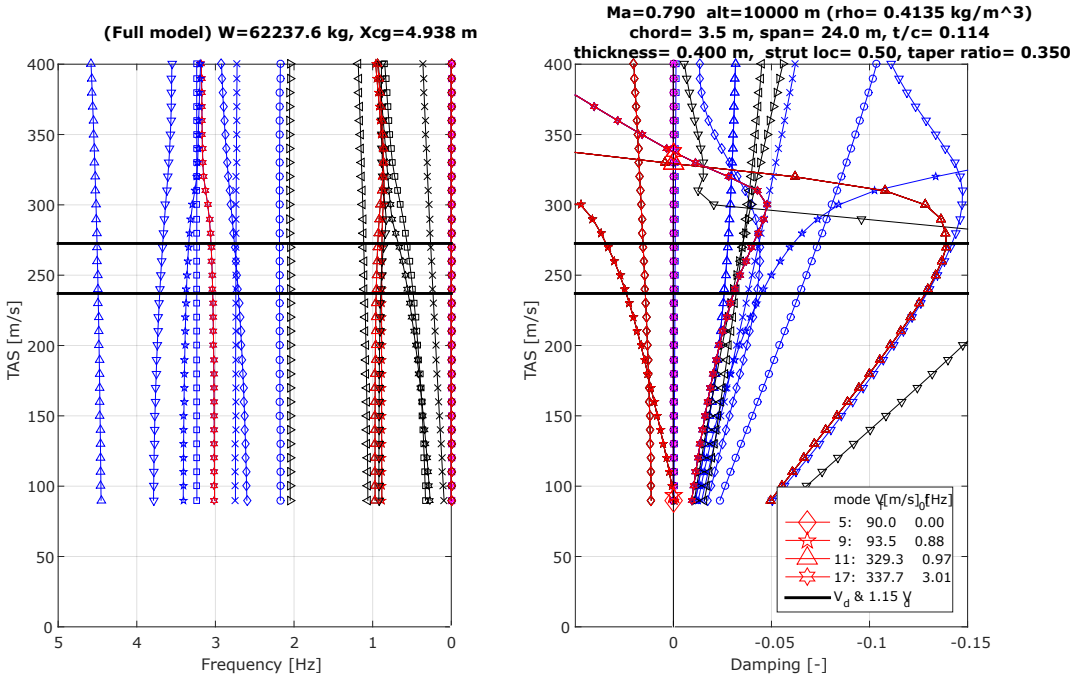


Figure B.2: Flutter modes showing wing as critical mode