

# Wing Optimization with Active Load Control

Integrating Maneuver Load  
Control and Gust Load Alleviation  
in Wing Structural Optimization of  
Large Transport Aircraft

J.H. Bussemaker

Faculty of Aerospace Engineering  
Flight Performance and Propulsion

Delft University of Technology



 **TU Delft** **AIRBUS**



# Wing Optimization with Active Load Control

## Integrating Maneuver Load Control and Gust Load Alleviation in Wing Structural Optimization of Large Transport Aircraft

by

Jasper Herm Bussemaker

to obtain the degree of Master of Science

at the Delft University of Technology,

to be defended publicly on Wednesday 4 April, 2018 at 14:00.

Student number: 4144023  
Project duration: May 1, 2017 – February 28, 2018  
Thesis committee: Prof.dr.ir. L.L.M. Veldhuis, TU Delft, FPP, Chair Flight Performance  
Dr.ir. R. De Breuker, TU Delft, ASM, Supervisor  
Dr.ir. G. La Rocca, TU Delft, FPP  
Dipl.-Ing. S. Binder, Airbus, Aeromechanics Systems, Supervisor  
P.M.G.J. Lancelot, TU Delft, ASM

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.



# Abstract

In order to improve the performance of large transport aircraft, wing control surfaces can be used to actively control aeroelastic loads: loads arising from the interaction between aerodynamics and structural dynamics. Two of such active aeroelastic load control techniques are Maneuver Load Control (MLC) and Gust Load Alleviation (GLA), which reduce loads in maneuvering flight and in flight through turbulence, respectively. In aircraft design, many disciplines may interact with each other. Taking more disciplines into account, from earlier in the design process produces more optimal designs. Therefore, to fully capture the benefit of MLC and GLA, these active load control systems should be designed at the same time as the wing structure. In this thesis, MLC and GLA are integrated in the structural optimization process of large transport aircraft. Focus is placed on keeping the impact on computational time low, since in the early design process it is important to be able to analyze many aircraft configurations quickly.

MLC and GLA have been considered in several structural optimization studies recently, but never together. Also, GLA controllers have always been tuned using optimal control methods. Optimal control, however, involves objectives which do not directly relate to load control for weight reduction. In this thesis, it is investigated whether it is possible to concurrently optimize the GLA controller and the wing structure, without the need for optimal control techniques.

MLC and GLA are integrated in the structural analysis tool Lagrange. To facilitate the integration, a Python framework that wraps Lagrange is developed. This framework, called XLagrange, provides improvements in input file processing, handling of aircraft trimming, constraint aggregation, and the adding of extra design disciplines. Optimization is performed using the OpenMDAO framework.

MLC is integrated by automatically defining wing control surface deflections as design variables in static aeroelastic analysis. Support for constraining the maximum sectional lift is added, to support the fact that wing maximum lift might be a limiting factor for the application of MLC. GLA is integrated in frequency-domain gust response analysis, for discrete gusts. Gust loads are calculated by adding static wing displacements to dynamic gust response displacements. Feedback stability and sensitivity constraints are implemented based on data-driven frequency response methods. Passive Turbulence Alleviation (PTA), a method for load relief during gusts by statically deploying wing control surfaces, is achieved by applying MLC to the static gust deflection load case.

To investigate the impact of MLC and GLA on the wing structural design process, optimizations are performed on the wing of a representative large transport aircraft. Using MLC on a design where gust loads are not critical results in significant weight reduction. With initially critical gust loads, feedforward GLA is able to effectively reduce gust loads until these are not sizing anymore. Using feedforward GLA together with MLC results in a weight reduction that is similar to when only MLC is used with non-critical gust loads. Including PTA in optimization with MLC and feedforward GLA results in a slightly larger weight reduction, as PTA gives the optimizer more choice in which design variables to use for gust load control. The implemented feedback GLA stability constraints do not offer enough guidance to the optimizer to result in a feasible, stable design.

Pre-tuning the GLA controller using optimal control techniques results in a slightly larger weight reduction compared to not pre-tuning the controller. However, it is concluded that in

general no optimal control techniques are needed to tune the GLA controller: the optimizer automatically recognizes the application of both MLC and GLA for load control to facilitate weight reduction. Concurrent optimization of controller and structure outperforms sequential optimization: sequential optimization results in less effective gust load control and a design where gust loads are partially sizing, as opposed to the concurrently optimized design where gust loads are not sizing.

Adding MLC and GLA to structural optimization approximately doubles to triples the time needed per iteration, mainly due to the addition of GLA. However, time needed per iteration is reduced compared to analysis in Lagrange, due to the availability of analytical gust response gradients.

# Acknowledgments

This document is the final report of my master's thesis performed over the course of nine months at Airbus in Ottobrunn, Germany. It marks the end of my years as a student of the Aerospace Engineering faculty at Delft University of Technology.

First, I would like to thank Roeland de Breuker, my university supervisor, and Andreas Wildschek, my team leader at Airbus, for the amazing opportunity in the Aeromechanics Systems team they provided me with for my thesis research. It has been an absolute pleasure to work on the forefront of developments in this field, and being part of the long-term industry vision on optimization and active load control. Furthermore, from Airbus I would also like to thank Simon Binder, my industry supervisor, for guiding me in the final months and providing valuable insights and feedback on aeroelastic modeling and optimization, and Daniel Nussbächer, with whom I worked closely on the improvement of LagPy. My colleagues in the student room also deserve an acknowledgment for the company and the fun times, in particular my best friend and colleague Daniël de Vries, with whom I had many interesting discussions on optimization and aircraft design in general.

Then, I would like to thank my parents and brother for their interest in my work and support while writing my thesis. Finally, I would like to express my gratitude to my girlfriend, Melissa, for her unconditional love and support and for keeping me motivated throughout.

*Jasper Bussemaker  
Ottobrunn, Germany, February 2018*



# Nomenclature

## Roman Symbols

$A_k$	Aerodynamic Influence Coefficient (AIC) matrix
$C_L$	Lift coefficient
$C_l$	Sectional lift coefficient
$C_p$	Pressure coefficient
$f$	Objective function
$G_{gk}$	Aerodynamic-structural load splining matrix
$G_{kg}$	Structures-aerodynamic displacement splining matrix
$G_{kg,u}$	Unsteady structures-aerodynamic displacement splining matrix
$\vec{g}$	Design constraint vector
$\vec{g}^c$	Consistency constraint vector
$H_a$	Actuator transfer function matrix
$H_c$	Controller transfer function matrix
$H_s$	Sensor transfer function matrix
$H_{sys}$	Control system transfer function matrix
$K$	Stiffness matrix
$K_{hh}$	Generalized stiffness matrix
$k$	Reduced frequency
$k_e$	Smearred wingbox properties correction factor
$L$	Lift
$M$	Mass matrix
$M_{hh}$	Generalized mass matrix
$M$	Mach number
$M_s$	Maximum feedback sensitivity
$n$	Load factor
$n_r$	Number of rigid body modes
$n_s$	Number of structural modes

N

$n_t$	Number of gust response output time steps	
$n_z$	Normal vector z-component of panel or strip	
$\mathbf{P}_{aero}$	Aerodynamic loading matrix	
$\vec{p}_e$	External force vector	
$\vec{p}_{tot}$	Total external load vector	
$p^+$	Number of right-hand-side poles of a transfer function	
$P_{db}$	Plant transfer function magnitude	dB
$Q$	Youla parameter	
$Q_c$	Control aerodynamic matrix	
$Q_s$	Structural aerodynamic matrix	
$q$	Dynamic pressure	Pa
$r_p$	Plant transfer function order	
$SF$	Safety factor	
$\vec{u}$	Structural displacements vector	
$V$	True airspeed	
$W$	Weight	N
$w_g$	Gust input	
$\vec{x}$	Design variable vector	
$\dot{z}$	Vertical velocity	
$z^+$	Number of right-hand-side zeros of a transfer function	

### Greek Symbols

$\alpha$	Angle of attack	deg
$\Delta_0^\infty$	Total phase shift from zero to infinite frequency	rad
$\delta_a$	Aileron deflection	deg
$\delta_c$	Control surface deflection	deg
$\delta_e$	Stabilizer deflection	deg
$\delta_s$	Spoiler deflection	deg
$\delta_{max}$	Maximum control surface deflection range	deg
$\dot{\delta}_{max}$	Maximum control surface deflection rate	deg/s
$\epsilon$	Finite difference step	

$\zeta_a$	Actuator damping ratio	
$\eta$	Spanwise location	
$\theta$	Pitch angle	
$\vec{\lambda}$	Eigenvalue vector	
$\vec{\xi}$	Modal coordinates vector	
$\rho$	Kreisselmeier-Steinhauser tolerance parameter	
$\sigma$	Stress	MPa
$\phi_c$	Control modes matrix	
$\phi_r$	Rigid body mode matrix	
$\phi_s$	Structural modes matrix	
$\vec{\phi}_g$	Gust mode vector	
$\vec{\phi}_{s_i}$	Structural eigenvector	
$\omega$	Angular frequency	rad/s
$\omega_a$	Actuator natural frequency	rad/s

### Abbreviations

AFS	Active Flutter Suppression
AIC	Aerodynamic Influence Coefficients
CG	Center of Gravity
DFT	Discrete Fourier Transform
DLM	Doublet Lattice Method
DOF	Degrees of Freedom
FCS	Flight Control System
FE	Finite Element
FEM	Finite Element Method
FIR	Finite Impulse Response
FSP	Front Spar
GLA	Gust Load Alleviation
IDF	Individual Discipline Feasible
IDFT	Inverse Discrete Fourier Transform
IIR	Infinite Impulse Response

---

ITLT	In-the-loop Tuning
KKT	Karush-Kuhn-Tucker
KS	Kreisselmeier-Steinhauser function
LQR	Linear Quadratic Regulator
LSK	Lower Skin
MDA	Multidisciplinary Design Analysis
MDF	Multidisciplinary Feasible
MDO	Multidisciplinary Design Optimization
MIMO	Multi-Input Multi-Output
MLC	Maneuver Load Control
OOP	Object Oriented Programming
PD	Proportional-Derivative Control
PSD	Power Spectral Density
RFA	Rational Function Approximation
RHS	Right-hand Side
RSP	Rear Spar
SISO	Single-Input Single-Output
UML	Unified Modeling Language
USK	Upper Skin
XDSM	Extended Design Structure Matrix

**Subscripts**

$f$	Set of free structural DOFs (f-set)
$g$	Set of all structural DOFs (g-set)
$h$	Set of generalized structural DOFs (h-set)
$i$	$i^{\text{th}}$ element of a vector
$k$	Set of all aerodynamic DOFs (k-set)
$l$	Set of structural DOFs (l-set) in the f-set but not in r-set
$r$	Set of rigid body structural DOFs (r-set) in the f-set

# Contents

<b>Abstract</b>	<b>v</b>
<b>Acknowledgments</b>	<b>vii</b>
<b>Nomenclature</b>	<b>ix</b>
<b>Contents</b>	<b>xiii</b>
<b>List of Figures</b>	<b>xv</b>
<b>List of Tables</b>	<b>xvii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 XLAGrange Framework</b>	<b>5</b>
2.1 Motivation . . . . .	5
2.2 Framework Architecture Description . . . . .	6
2.2.1 XLAGrange Discipline Programming Interface . . . . .	7
2.2.2 Integration of Optimization Features in XLAGrange . . . . .	7
2.3 Constraint Aggregation Using the Kreisselmeier-Steinhauser Function . . . . .	8
2.4 Aeroelastic Trimming Constraints . . . . .	9
2.4.1 Coupled Design and Trimming Constraint Gradients . . . . .	11
<b>3 Maneuver Load Control</b>	<b>13</b>
3.1 Load Alleviation Concept . . . . .	13
3.2 MLC Control System . . . . .	14
3.3 Maximum Sectional Lift Constraint . . . . .	15
<b>4 Gust Load Alleviation</b>	<b>17</b>
4.1 Load Alleviation Concept and Integration Strategy . . . . .	17
4.2 Implications of the Frequency Domain . . . . .	19
4.3 Addition of Static Loads to Dynamic Response Loads . . . . .	19
4.4 Frequency-Domain Gust Response Analysis . . . . .	21
4.4.1 Identification of Structural Vibration Modes . . . . .	21
4.4.2 Aeroservoelastic System Formulation . . . . .	23
4.4.3 Derivation of Unsteady Splining Matrix . . . . .	24
4.4.4 XLAGrange Gust Response Analysis Verification . . . . .	24
4.5 GLA Control System . . . . .	25
4.5.1 Controller Tuning in Structural Optimization . . . . .	26
4.5.2 Control Laws and Performance Indices . . . . .	28
4.5.3 Sensors and Actuators . . . . .	28
4.5.4 Feedback Control System Constraints . . . . .	29
4.5.4.1 Stability Constraints . . . . .	29
4.5.4.2 Sensitivity Constraints . . . . .	30
4.5.4.3 Feedback Controller Design Space Exploration . . . . .	32
4.6 Passive Turbulence Alleviation Concept . . . . .	34

<b>5</b>	<b>Numerical Experiments Setup</b>	<b>37</b>
5.1	Aircraft Model Description . . . . .	37
5.2	Optimization Problem Formulation . . . . .	38
5.2.1	Load Cases . . . . .	38
5.2.2	Objective Function . . . . .	40
5.2.3	Design Variables and Bounds. . . . .	40
5.2.4	Design Constraints . . . . .	41
5.3	Computational Time Impact Assessment Method . . . . .	41
<b>6</b>	<b>Results</b>	<b>43</b>
6.1	Structural Optimization Without Active Load Control. . . . .	43
6.1.1	Implicit Aeroelastic Trimming . . . . .	43
6.1.2	Explicit Aeroelastic Trimming . . . . .	46
6.2	Structural Optimization with Maneuver Load Control. . . . .	47
6.2.1	Relaxed Aileron Deflection Bounds . . . . .	49
6.2.2	Influence of Gust Loads . . . . .	49
6.3	Structural Optimization with Gust Load Alleviation . . . . .	50
6.3.1	MLC and Feedback + Feedforward GLA . . . . .	50
6.3.2	MLC and Feedforward GLA . . . . .	50
6.3.3	MLC, Feedforward GLA and Passive Turbulence Alleviation . . . . .	53
6.4	Structural Optimization with a Pre-Tuned GLA Controller . . . . .	55
6.4.1	Controller Tuning Using Optimal Control . . . . .	55
6.4.2	Sequential Optimization of Controller and Structure. . . . .	56
6.4.3	Concurrent Optimization of Pre-Tuned Controller and Structure. . . . .	59
6.5	Results Overview . . . . .	61
6.6	Impact on Computational Time . . . . .	61
6.6.1	Maneuver Load Control . . . . .	61
6.6.2	Gust Load Alleviation. . . . .	63
6.7	Discussion . . . . .	63
<b>7</b>	<b>Conclusions and Recommendations</b>	<b>65</b>
7.1	Conclusions . . . . .	65
7.2	Recommendations. . . . .	66
	<b>Bibliography</b>	<b>69</b>

# List of Figures

1.1	Collar’s triangle showing the different types of forces and loads. . . . .	2
2.1	UML diagram of the XLAGrange framework main classes. . . . .	6
2.2	Behavior of the Kreisselmeier-Steinhauser (KS) function. . . . .	8
2.3	Comparison between trimming XDSM schemes. . . . .	9
2.4	Comparison between inequality and equality trimming constraints. . . . .	10
2.5	Comparison of a constraint function progression with and without implicit re-trimming. . . . .	12
3.1	Lift redistribution principle of Maneuver Load Control. . . . .	14
3.2	Bending moment relief principle of Maneuver Load Control. . . . .	14
3.3	XDSM scheme of MLC integration in a static aeroelasticity MDO problem. . . . .	15
3.4	Notional behavior of the maximum lift constraint. . . . .	16
4.1	UML diagram of gust response and GLA discipline classes. . . . .	18
4.2	Representation of a discrete gust. . . . .	18
4.3	Selection of critical time steps procedure. . . . .	21
4.4	Verification of XLAGrange gust response analysis. . . . .	25
4.5	UML schema of GLA control system classes. . . . .	26
4.6	Comparison between GLA controller design XDSM schemes. . . . .	27
4.7	Comparison of actuator responses in a GLA feedback system. . . . .	31
4.8	Design space exploration of feedback controller tuning. . . . .	33
4.9	Effect of Passive Turbulence Alleviation (PTA) on strength constraints due to gust. . . . .	35
5.1	A typical CS-25 maneuver diagram. . . . .	39
6.1	Optimization without active control, implicit trimming. . . . .	44
6.2	Load case hierarchy of optimization without active control, implicit trimming. . . . .	45
6.3	Design space exploration at the optimized structural design point. . . . .	45
6.4	Optimization without active control, explicit trimming. . . . .	46
6.5	Optimization including MLC, implicit trimming. . . . .	47
6.6	Load case hierarchy of optimization with MLC. . . . .	48
6.7	Lift distribution before and after optimization with MLC. . . . .	49
6.8	Optimization with MLC and feedback and feedforward GLA, implicit trimming, with critical gust loads. . . . .	50
6.9	Optimization with MLC and feedforward GLA, implicit trimming, with critical gust loads. . . . .	51
6.10	Load case hierarchy of optimization with MLC and feedforward GLA, with critical gust loads. . . . .	52
6.11	Gust response of structure optimized with MLC and feedforward GLA, with critical gust loads. . . . .	52
6.12	Gust response of structure optimized with MLC, feedforward GLA and PTA, with critical gust loads. . . . .	53

---

6.13 Optimization with MLC, feedforward GLA and PTA, implicit trimming, with critical gust loads. . . . .	54
6.14 Comparison of GLA controller tuning results. . . . .	55
6.15 Feedforward controller tuning progression. . . . .	56
6.16 Sequential GLA controller/structure optimization with MLC, implicit trimming, with critical gust loads. . . . .	57
6.17 Load case hierarchy of sequential GLA controller/structure optimization, with critical gust loads. . . . .	58
6.18 Gust response of sequential GLA controller/structure optimization, with critical gust loads. . . . .	58
6.19 Gust response of structure optimized with MLC and pre-tuned feedforward GLA, with critical gust loads. . . . .	59
6.20 Optimization with MLC and pre-tuned feedforward GLA, implicit trimming, with critical gust loads. . . . .	60
6.21 Impact of MLC discipline on computational time. . . . .	62
6.22 Impact of GLA disciplines on computational time. . . . .	62

# List of Tables

5.1	Load cases used in the optimization problems. . . . .	39
6.1	Structural optimization results overview. . . . .	61



# Introduction

Modern transport aircraft are designed with stringent performance requirements to keep operational costs and environmental impact as low as possible [1]. In order to maximize aerodynamic performance, wings are designed to have high aspect ratios [2]. This means that, because wings are also designed to be as lightweight as possible, flexibility has a large effect on the aerodynamic characteristics of wings [3, 4]. When flying, aerodynamic loads on the wing interact with the wing structure itself: aerodynamic loads cause deformations, but these deformations in turn cause changes in the aerodynamic loads. This intricate interconnection between the aerodynamics and structural dynamics is studied in the field of aeroelasticity [5, 6]. Aeroelastic loading and adverse aeroelastic effects [7, 8] pose major challenges that must be taken into account during structural design [9, 10]. The different types of forces that are at play for different types of aeroelastic loads are shown in Figure 1.1. Being able to modify the aeroelastic behavior of a given wing such that structural stresses are reduced opens up opportunities for performance improvements, which can come in the form of weight reductions, wing span increases, or a combination of the two [11–16].

In the pursuit of creating ever longer and more slender wings, many techniques have been developed, ranging from aeroelastic tailoring using composite materials [17–19] to designing completely new aircraft configurations like the blended-wing-body [20]. Another way of achieving performance improvement is by using active aeroelastic load control: the use of wing control surfaces to modify the aeroelastic behavior of the wing [5, 16]. Active aeroelastic load control encompasses many techniques, with diverse objectives including stability augmentation, ride quality control, fatigue life enhancement, and stress reduction [5, 10].

Aircraft design is a prime example of an engineering branch where many disciplines play a role and interact with each other [21, 22]. Design optimization problems with multiple disciplines are studied in the field of Multidisciplinary Design Optimization (MDO) [23, 24]. Optimization is the process of automated design synthesis, such that some measurable figure-of-merit, such as wing weight or fuel burn, is minimized. The field of MDO deals with the theoretical background of the interactions between disciplines and guaranteeing consistency between different types of analysis models used by different disciplines. In the context of wing structural design, the multidisciplinary nature is present in the interaction between aerodynamics and structures, as in aeroelastic problems [25, 26], and in combination with control, as in aeroservoelastic problems [27, 28]. Taking more disciplines into account, from earlier in the design process produces more optimal designs than a traditional sequential design process, since synergies between disciplines can then be exploited more effectively [29–32]. Therefore, to fully capture the potential of active load control on the wing structural

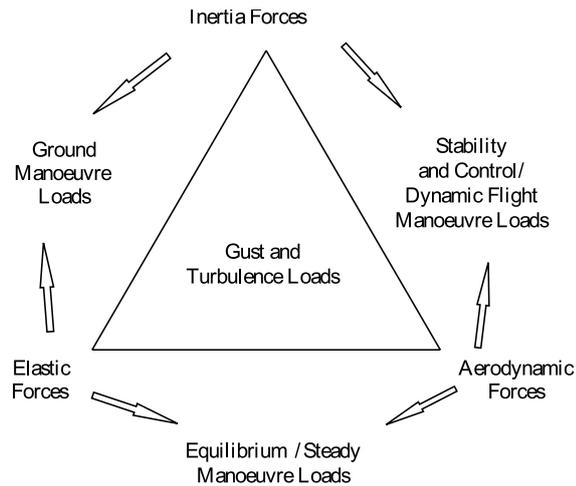


Figure 1.1: Collar's triangle showing the different types of forces playing a role in different kinds of aeroelastic loads. Adopted from [6].

design, the active load control techniques should be taken into account from early on in the structural design process, and the load control systems and structure should be designed at the same time.

Two major load cases that wing structures are sized for, as specified in the CS-25 regulations [33], are maneuver and turbulence gust loads. Maneuver loads are determined from a combination of elastic and aerodynamic forces, while for turbulence loads inertia forces also play a role, see Figure 1.1. To reduce aeroelastic loads during maneuvering flight and during flight through turbulence, the active load control techniques of Maneuver Load Control (MLC) and Gust Load Alleviation (GLA) can be used, respectively [6, 34, 35]. MLC is a load control concept where wing control surfaces are statically deflected during maneuvers. GLA is a load control technique where a controller dynamically commands the deflection of wing control surfaces when encountering turbulence. Control laws can be based on either a feedback loop (based on wing tip acceleration) or feedforward channel (based on changes in aircraft angle-of-attack) design. A combination of feedback and feedforward control is also possible. Employing MLC and GLA can result in significant weight savings, or increased wing spans for the same weight.

MLC and GLA are standard features on modern large transport aircraft [14, 36, 37], but the design tools to include these techniques early on in the design process have not yet matured [38]. MLC has been considered in several optimization studies recently [11, 12, 39], however without the simultaneous consideration of gust loads and GLA. Feedback GLA has been considered in design optimization studies by Nam et. al. [40] and Pusch et. al. [41]. Other studies have considered the synthesis of both feedback and feedforward GLA controllers for already existing aircraft designs [13, 15, 42–46], without the re-optimization of the designs after the addition of GLA. In the studies of Nam and Pusch, the feedback GLA controller has been tuned using optimal control synthesis methods. However, loads are reduced indirectly by these optimal control methods: it can be difficult to formulate an optimal control objective that coincides with the load alleviation goal of GLA [45]. Another way of optimizing the wing structure while taking GLA into account is by letting the optimizer freely vary the controller gains, as was done by Suzuki et. al. [47]. This way, GLA is used directly for the purpose of load control, instead of through the intermediary goal of designing an optimal controller. The study by Suzuki indicates that the optimizer is well able to apply GLA for load control, but does not consider MLC, does not consider feedforward GLA, and uses outdated optimization

techniques, as the study was performed in 1993. This thesis considers the integration of both MLC and GLA into the wing structural optimization process, without the need for optimal control methods. Also, both feedback and feedforward GLA controllers are considered.

For the earlier stages of the wing structural design, where many configurations are analyzed and optimized relatively quickly, it also is important to have time-effective design and optimization tools available. Therefore the main research question of this thesis is:

**Can Maneuver Load Control (MLC) and Gust Load Alleviation (GLA) be integrated in the wing structural optimization process of large transport aircraft without causing a significant increase of computational time, and can the optimizer automatically apply them for load control, without the need for optimal control techniques?**

Keeping impact on computational time low is especially challenging for gust response analysis, which is needed to assess the impact of GLA. Gust response analysis methods can be classified as time-domain and frequency-domain methods [48]. In past studies, time-domain methods have been used for the integration of GLA [13, 15, 40–45], since this allows nonlinearities to be modeled. Frequency-domain methods, on the other hand, can take less time to compute [49, 50]. In this thesis, it is hypothesized that integrating GLA into frequency-domain gust response analysis still results in realistic load predictions, and thereby providing a time-effective method for assessing the effects of GLA.

The integration and testing will be performed in the context of wing structural design for conventional large transport aircraft that are certified according to the CS-25 regulations [33], and the active control concepts will be integrated in the Airbus structural analysis and optimization tool: Lagrange [51]. Optimizations will be performed using a gradient-based optimization algorithm, as is common for aerostructural problems with many design variables and constraints [23, 52–54]. Given this, the main research objective is:

**To integrate MLC and GLA into the large transport aircraft wing structural optimization process, show this can be done without significantly increasing computational time, and investigate whether the optimizer automatically applies the MLC and GLA for load control or not, by integrating them in a computationally efficient way into Lagrange and performing gradient-based optimizations of a large transport aircraft wing where structural and control parameters are optimized simultaneously.**

This thesis report is structured as follows. The framework developed to facilitate the integration of MLC and GLA into Lagrange is presented and discussed in Chapter 2. The integration of Maneuver Load Control and Gust Load Alleviation are discussed in Chapter 3 and Chapter 4, respectively. The setup of the numerical experiments used to investigate the impact of MLC and GLA is presented in Chapter 5. Chapter 6 presents and discusses the results. Finally, the report is concluded with Chapter 7, where the conclusions and recommendations are presented.



# 2

## XLagrange Framework

This chapter presents the XLagrange framework developed for this thesis to facilitate the integration of active control in structural optimization. First, the motivation for developing the framework is presented in Section 2.1. Then, Section 2.2 presents the design of the framework. Finally, the constraint aggregation and trimming features are discussed in Sections 2.3 and 2.4.

### 2.1. Motivation

Lagrange is a structural analysis and optimization tool developed by Airbus since the 1980's [10, 51], with focus on aeroelasticity, structural constraints, composite materials, and analytical gradients. Lagrange uses input files that have a format similar to Nastran input files [55], which contain the analysis structural model, load cases, design variables, objective functions, and constraint functions. Using this, Lagrange can be used in numerical optimization, where by varying the design variables using an optimization algorithm, an objective is minimized (or maximized, depending on the convention used) while satisfying design constraints [56, 57].

An interface to Lagrange has been developed for the Python programming language: LagPy [58]. The Python programming language [59] is a popular interpreted, high-level, object-oriented, general-purpose programming language. Its strength lies in the clear syntax, extensive built-in library, and large user base. Even though it is not as fast as low-level languages like C or Fortran, it is very suitable for scripting and prototyping. LagPy enables the features of Lagrange to be used in Python, thereby enabling the integration of external disciplines in Lagrange analysis, or the integration of Lagrange into larger scale analysis or optimization software packages, for example as the structural optimizer in an aerostructural Multidisciplinary Design Optimization (MDO) problem [60]. However, LagPy lacks features in some areas needed for successful structural design and optimization, for example: easy manipulation of input files, distribution of engineering problems over multiple input files, modifying and/or adding design variables and/or constraints, and automated aircraft trimming. To solve this, the XLagrange framework (Extended Lagrange) has been developed. The main goals of the XLagrange framework are:

1. Extend LagPy with features that automate and simplify many common operations;
2. Facilitate easy merging of multiple Lagrange problems into one;
3. Offer a common interface for adding extra disciplines to Lagrange analysis.

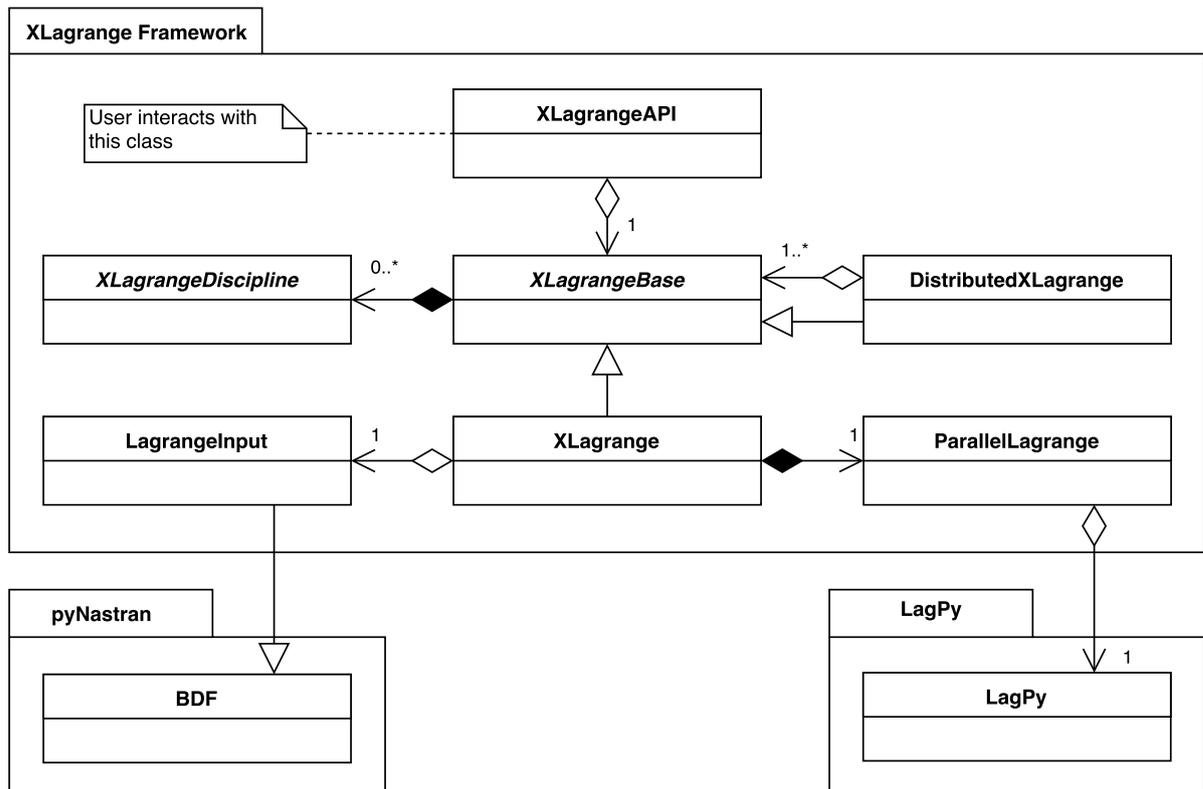


Figure 2.1: UML diagram of the XLAGrange framework main classes.

## 2.2. Framework Architecture Description

Setting up and operating an XLAGrange session is similar to operating a LagPy session, to exploit any possible previous user experience with LagPy. LagPy is operated by instantiating a `LagPy` object with a Lagrange input file, which then supports operations for interpreting (pre-processing) the file, analyzing (designing) the model, changing the design vector and evaluating the objective function and design constraints. An XLAGrange session is initiated in a similar way: an `XLAGrange` object is instantiated for a Lagrange input file. This object then automatically takes care of instantiating a `LagPy` object in the background and initializing extra disciplines as requested by the user. The framework is set up using the Object Oriented Programming (OOP) paradigm [61], allowing the separation of functionality into classes. Figure 2.1 presents the Unified Modeling Language (UML) [62] class diagram of the XLAGrange framework, visualizing the framework layout as described in this section.

Python parallelization libraries have been used in the XLAGrange framework, to use the processor as much as possible, in order to speed up the analysis and optimization process. The `LagPy` instance is wrapped by the `XLAGrange` instance through the use of the remote proxy pattern [61], represented by the `ParallelLagrange` object, which contains the `LagPy` object in a parallel Python process [63].

Knowledge is required about the used input file for parsing the types of design variables and constraints and for modification of the input file. This knowledge is provided through the use of the `XLAGrangeInput` class, which extends the `pyNastran` Python package [64] to enable a convenient OOP interface to the Nastran input format, and extend it with Lagrange input format specifics.

Due to the nature of the Lagrange input format, it can be necessary to divide one engineering problem over multiple input files. For example, one input file can only represent

one mass case. Since each `LagPy` instance only handles one input file, the merging of multiple `XLagrange` instances into one object for user interaction is made possible using the composite design pattern [61]: individual objects and composition of objects are treated uniformly. This pattern can be seen in the UML schema in Figure 2.1, where both the `XLagrange` and `DistributedXLagrange` classes are implementations [62] of the `XLagrangeBase` class, but that a `DistributedXLagrange` instance also contains one or more `XLagrangeBase` instances. Only the `XLagrangeBase` interface is interacted with by disciplines and the user, so that a session with one Lagrange input file is treated exactly the same as a session with multiple input files.

Finally, because the `XLagrangeBase` class contains many methods that are irrelevant to the end user, the protection proxy pattern [61] is used to create the `XLagrangeAPI` class. An Application Programming Interface (API) is the programming interface of a program, enabling interaction with said program. In this case, the `XLagrangeAPI` class represents the interface used by the Python scripts developed by the user to interact with the `XLagrange` framework, hence the name.

### 2.2.1. XLagrange Discipline Programming Interface

Extra engineering disciplines are integrated into Lagrange design problems using the discipline interface of the `XLagrange` framework. Disciplines act on the optimization problem in general by defining extra design variables, providing a different objective function and defining extra constraints, and act on the Lagrange optimization problem in specific by overriding Lagrange design variables and constraints and requesting Lagrange data like the Finite Element (FE) stiffness and mass matrices, displacements, pressure distributions, Aerodynamic Influence Coefficient (AIC) matrices, and relevant gradients.

Disciplines are added to the optimization problem by instantiating the relevant discipline classes, which are implementations of the abstract `XLagrangeDiscipline` class, and passing them on to the `XLagrange` or `DistributedXLagrange` object when initializing a session. These discipline classes contain all data and logic relevant to that discipline, effectively separating basic structural design data from discipline specific data. This leads to a very flexible system, where the original Lagrange input does not have to be adjusted in order to add support for these extra disciplines.

### 2.2.2. Integration of Optimization Features in XLagrange

Lagrange itself contains an optimizer, but does not support the addition or modification of design variables and constraints. So for the `XLagrange` framework, the choice has been made to re-implement the optimization features in Python. An existing Python optimization framework is used, exposing the `XLagrange` framework through a common optimization interface. Providing a common interface for optimization enables the possible future use of `XLagrange` as the structural discipline in a multidisciplinary aircraft design problem.

One optimization framework available in Python and also offering support for multidisciplinary optimization problems is `OpenMDAO` [65]. `OpenMDAO` is an open-source state-of-the-art [66–68] multidisciplinary design analysis and optimization (MDAO) framework written in Python and developed by NASA. `OpenMDAO` offers the benefit of being well supported, open source, written in Python, offering a wide array of optimization algorithms, and offering easy visualization through the use of `Recorder` classes. Therefore, the choice is made to use `OpenMDAO` for the optimization interface of the `XLagrange` framework.

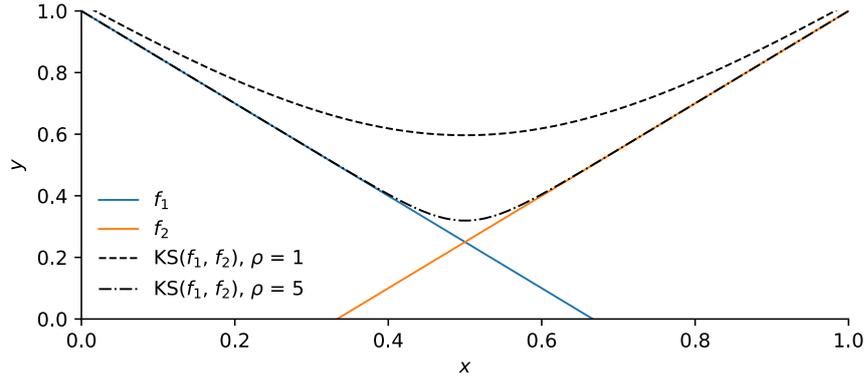


Figure 2.2: Behavior of the Kreisselmeier-Steinhauser (KS) function for different values of tolerance parameter  $\rho$ .

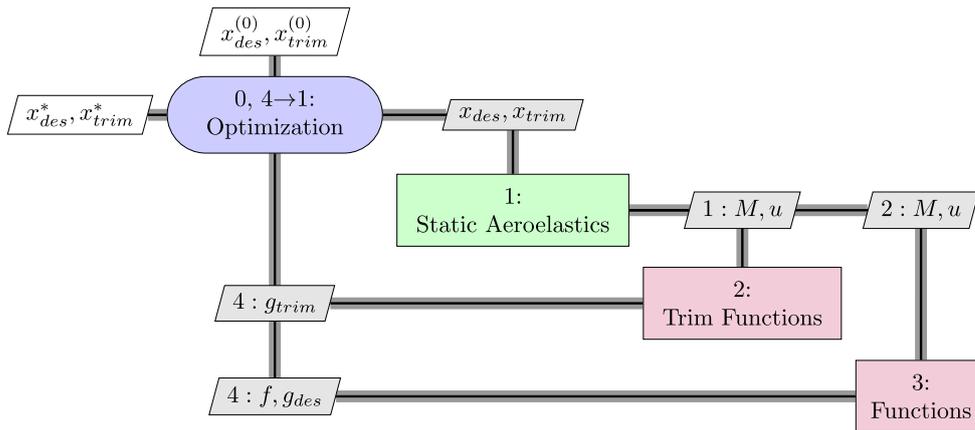
### 2.3. Constraint Aggregation Using the Kreisselmeier-Steinhauser Function

In a large-scale structural optimization problem, the number of constraints can quickly grow. For example, strength constraints are defined per structural element, and for gust response analysis also per output time step. It is not uncommon to work with tens of thousands or more constraints. In order to reduce the number of constraints, to provide more insight into the behavior of constraints, and to reduce the execution time of the optimization algorithm, XLAGrange offers the option of aggregating inequality constraints. Lagrange defines inequality constraints to be satisfied when  $g \geq 0$ , so the most critical constraints are the ones with the lowest values. A trivial solution for aggregation is then to take the minimum of the constraint values to be aggregated. However, at the point where one of the to-be-aggregated constraint functions takes over as being the most critical as design variables are being changed, the  $\min$  function would yield a step-change in the gradient of the aggregated function. Constrained optimization problems where the optimum has active constraints, which is how structural optimization problems behave, always have an optimum at the intersection of multiple inequality constraints [24]. If these multiple inequality constraints are now replaced by one aggregated inequality constraint, this means that at the point of intersection the gradient must be zero in order to be able to satisfy the Karush-Kuhn-Tucker (KKT) [24] optimality conditions. Due to the gradient step-change, using  $\min$  for aggregation does not allow for the satisfaction of the KKT conditions, and thus makes it unusable for gradient-based optimization.

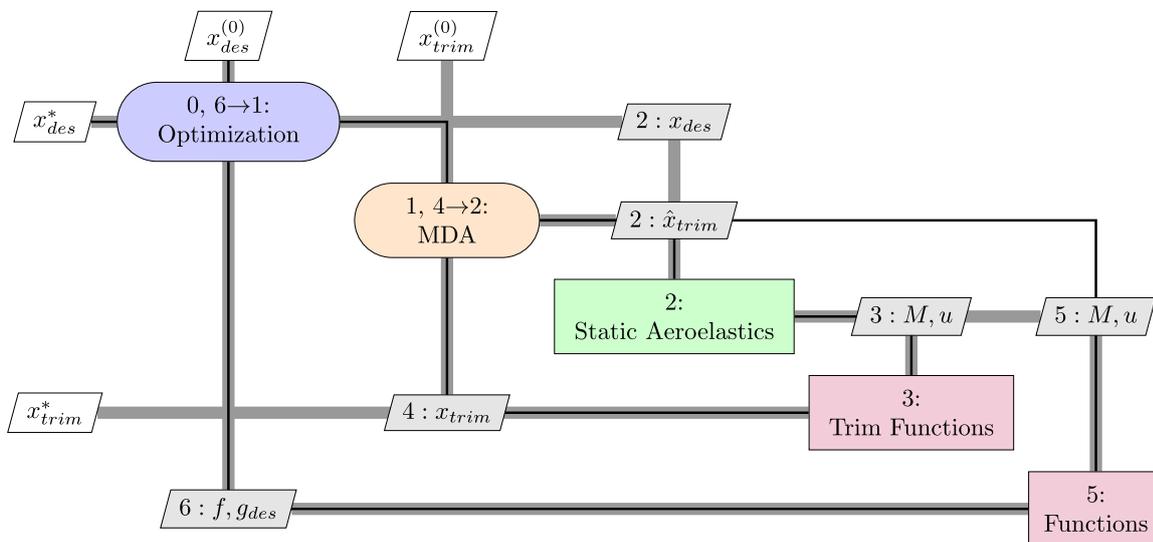
To solve the gradient issue, the Kreisselmeier-Steinhauser (KS) function is used [69–71]:

$$f_{ks}(x) = f_{max} + \frac{1}{\rho} \ln \sum e^{\rho(f_i(x) - f_{max})}, \quad (2.1)$$

where  $\rho$  is the KS tolerance parameter,  $f_i$  represents the values to be aggregated, and  $f_{max}$  is the maximum value of  $f_i$ . The use of  $f_{max}$  in the equation is done to prevent numerical overflow when calculating the exponential, but otherwise has no influence on the solution [70]. In order to aggregate the inequality constraints, the KS function is simply evaluated with the negatives of the constraint values, so that the lowest-value constraint indeed is the most critical. Figure 2.2 shows the behavior of the KS function. It shows that the KS function approaches the behavior of  $\max$  when values are far apart, but provides a well-defined smooth gradient that is zero at the point of cross-over. Figure 2.2 also shows that increasing the value of  $\rho$  pulls the KS function closer to the  $\max$  function, but at the cost of yielding a less smooth gradient. The disadvantage of the KS function is that it may yield conservative designs, since the value of the



(a) IDF or “explicit trimming” scheme: the system optimizer trims the aircraft.



(b) MDF or “implicit trimming” scheme: the aircraft is automatically trimmed for each evaluation.

Figure 2.3: Comparison between trimming XDSM schemes.

KS function at the function cross-over is higher (i.e. more critical) than the actual underlying functions.

By default, constraints of similar type and load case are aggregated into one. XLAGrange disciplines can define their own aggregation rules.

## 2.4. Aeroelastic Trimming Constraints

In order to perform static aeroelastic analysis, the aircraft has to be trimmed, which is handled by trimming variables and constraints. Trimming constraints for example specify that lift should be equal to weight, and trimming variables are then adjusted to make sure this is indeed the case. Trimming constraints are equality constraints: constraints which are satisfied when  $g(\vec{x}) = 0$ , where  $\vec{x}$  is the design vector consisting of the structural design variables  $\vec{x}_{des}$  and the trimming variables  $\vec{x}_{trim}$ . Trimming constraints can be regarded as consistency constraints and trimming variables as state variables, in the nomenclature of Martins [23].

In practical MDO applications, all MDO problem architectures are derived from two base

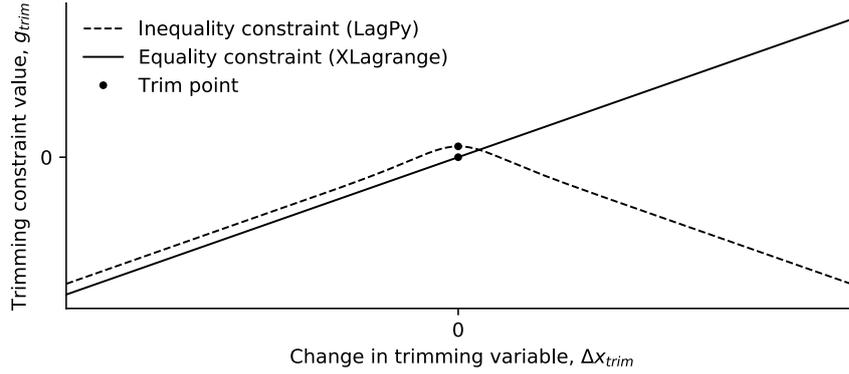


Figure 2.4: Comparison between trimming constraints implemented as inequality (as in LagPy) or equality (as in XLagrange) constraints.

architectures: Individual Discipline Feasible (IDF) and Multidisciplinary Feasible (MDF). In an IDF architecture, the optimizer controls the state variables and in addition to satisfying the design constraints also aims to satisfy the consistency constraints. In an MDF architecture, the optimizer only controls the design variables and a Multidisciplinary Design Analysis (MDA) block makes sure the consistency constraints are satisfied, by internally varying the state variables. For the trimming problem, this difference is visualized using the eXtended Design Structure Matrix (XDSM) [72] in Figure 2.3. In the IDF scheme, the trimming discipline outputs the trimming constraints, whereas in the MDF scheme the trimming discipline outputs an updated trimming vector, based on the internally calculated trimming constraint values. The XLagrange framework supports both formulations intrinsically. The IDF scheme is referred to as explicit trimming, whereas the MDF scheme is referred to as implicit trimming.

Lagrange trimming variables represent perturbations of Degrees of Freedom (DOFs) in the aerodynamic model. Aerodynamic surfaces are discretized in panels, where aerodynamic DOFs are the panel angle-of-attack  $\alpha$  values. From these panel  $\alpha$ 's, aerodynamic forces are calculated using Aerodynamic Influence Coefficient (AIC) matrices [73], which are generated by an external source for analysis in Lagrange. Each trimming variable corresponds to one control mode vector in the control mode matrix  $\phi_c$ , which relates the trimming variable values to the aerodynamic DOF perturbations.

In Lagrange, trimming constraints are inequality constraints, as shown in Figure 2.4. This makes the trimming constraints unusable for the calculation of coupled constraint gradients (see Section 2.4.1), because at the trimmed state, the trimming constraint function behaves highly nonlinear and makes it impossible to extrapolate the behavior of the constraint function to a non-trimmed state: if a change in  $g_{trim}$  is known, it is not known in which “direction” the trimming variable  $x_{trim}$  should be corrected to stay trimmed (see Figure 2.4). To solve this problem, trimming constraints are re-implemented as equality constraints in XLagrange using the discipline system. An equality constraint has the same gradient sign at the trimmed point as at any other point, therefore if a change in  $g_{trim}$  is known, it can directly be calculated by how much and in which “direction” the trimming variable  $x_{trim}$  should be adjusted to stay trimmed (see Figure 2.4).

The general Finite Element Method (FEM) linear static aeroelastic equilibrium equation, can be formulated as follows [73]:

$$(\mathbf{K} - q\mathbf{Q}_s)\vec{u} = q\mathbf{Q}_c\vec{x}_{trim} + \vec{p}_e, \quad (2.2)$$

where  $\mathbf{K}$  is the stiffness matrix,  $q$  the dynamic pressure,  $\mathbf{Q}_s$  the structural aerodynamic matrix,

$\vec{u}$  the structural displacements vector,  $\mathbf{Q}_c$  the control aerodynamic matrix and  $\vec{p}_e$  the external force vector. Total external load vector  $\vec{p}_{tot}$  can be expressed as the sum of external loads and aerodynamic loads:

$$\vec{p}_{tot} = q\mathbf{Q}_s\vec{u} + q\mathbf{Q}_c\vec{x}_{trim} + \vec{p}_e. \quad (2.3)$$

Trimming constraints are constraints posed on  $\vec{p}_{tot}$ , where the value generally is constrained to be zero. In general, a trim constraint can be formulated as

$$\vec{g}_{trim}(\vec{x}_{des}, \vec{x}_{trim}) = \mathbf{A}_{trim}(\vec{x}_{des}) \cdot \vec{p}_{tot}(\vec{x}_{des}, \vec{x}_{trim}), \quad (2.4)$$

where  $\mathbf{A}_{trim}$  is a linear operator which is different for each trimming constraint function.  $\mathbf{A}_{trim}$  and  $\vec{p}_e$  are functions of  $\vec{x}_{des}$ , since they depend on the Center of Gravity (CG) location and the weight, respectively. The CG location and structural weight both depend on the mass distribution, which is modified by the design variables  $\vec{x}_{des}$ .

The  $\partial\vec{g}_{trim}/\partial\vec{x}$  sensitivities are calculated by differentiating Equation 2.4:

$$\frac{\partial\vec{g}_{trim}}{\partial\vec{x}} = \frac{\partial\mathbf{A}_{trim}}{\partial\vec{x}}\vec{p}_{tot} + \mathbf{A}_{trim}\frac{\partial\vec{p}_{tot}}{\partial\vec{x}} = \frac{\partial\mathbf{A}_{trim}}{\partial\vec{x}}\vec{p}_{tot} + \mathbf{A}_{trim}\left(q\mathbf{Q}_s\frac{\partial\vec{u}}{\partial\vec{x}} + q\mathbf{Q}_c\frac{\partial\vec{x}_{trim}}{\partial\vec{x}} + \frac{\partial\vec{p}_e}{\partial\vec{x}}\right). \quad (2.5)$$

The derivatives of  $\mathbf{A}_{trim}$  and  $\vec{p}_e$  with respect to  $\vec{x}$  depend only on the mass distribution and can be calculated from the mass matrix gradient  $\partial\mathbf{M}/\partial\vec{x}$ , which is provided by LagPy. The  $\partial\vec{u}/\partial\vec{x}$  derivative is directly provided by LagPy as well. The value of  $\partial\vec{x}_{trim}/\partial\vec{x}$  is trivial to obtain.

### 2.4.1. Coupled Design and Trimming Constraint Gradients

In explicit trimming mode, the previously obtained solution for the trim constraint gradients is enough. In implicit trimming mode, however, sensitivities are calculated by relating the change in output functions  $d\vec{g}_{des}$  to the change in design variables  $d\vec{x}_{des}$  and the change in state variables  $d\vec{x}_{trim}$ , which themselves depend on the design variables: the gradients of the output functions to the design variables are coupled to the trimming gradients. This subsection outlines the method used for calculating the coupled gradients.

The linearized relations between changes in  $\vec{x}_{des}$ ,  $\vec{x}_{trim}$ ,  $\vec{g}_{des}$  and  $\vec{g}_{trim}$  are given by:

$$\begin{bmatrix} \frac{\partial\vec{g}_{des}}{\partial\vec{x}_{des}} & \frac{\partial\vec{g}_{des}}{\partial\vec{x}_{trim}} \\ \frac{\partial\vec{g}_{trim}}{\partial\vec{x}_{des}} & \frac{\partial\vec{g}_{trim}}{\partial\vec{x}_{trim}} \end{bmatrix} \begin{bmatrix} d\vec{x}_{des} \\ d\vec{x}_{trim} \end{bmatrix} = \begin{bmatrix} d\vec{g}_{des} \\ d\vec{g}_{trim} \end{bmatrix}, \quad (2.6)$$

where  $\vec{g}_{des}$  represents the constraints other than trimming constraints. For implicit trimming, the point at which the gradients are evaluated is at a trimmed point where  $\vec{g}_{trim} = 0$ . The coupled gradients are calculated for situations where the aircraft stays trimmed, so  $d\vec{g}_{trim} = 0$ . From this and Equation 2.6,  $d\vec{x}_{trim}$  can be expressed as function of  $d\vec{x}_{des}$ :

$$d\vec{x}_{trim} = -\left(\frac{\partial\vec{g}_{trim}}{\partial\vec{x}_{trim}}\right)^{-1} \frac{\partial\vec{g}_{trim}}{\partial\vec{x}_{des}} d\vec{x}_{des}, \quad (2.7)$$

which requires the  $\partial\vec{g}_{trim}/\partial\vec{x}_{trim}$  matrix to be invertible. This translates to the requirement that there should be as many trimming constraints as trimming variables, and that the trimming constraints can be controlled effectively by the trimming variables. If the number of trimming constraints and trimming variables differ, the problem becomes an overdetermined or underdetermined trimming problem [74], which XLAGrange currently does not support. Substituting Equation 2.7 in Equation 2.6 gives

$$\frac{d\vec{g}_{des}}{d\vec{x}_{des}} = \frac{\partial\vec{g}_{des}}{\partial\vec{x}_{des}} - \frac{\partial\vec{g}_{des}}{\partial\vec{x}_{trim}} \left(\frac{\partial\vec{g}_{trim}}{\partial\vec{x}_{trim}}\right)^{-1} \frac{\partial\vec{g}_{trim}}{\partial\vec{x}_{des}}. \quad (2.8)$$

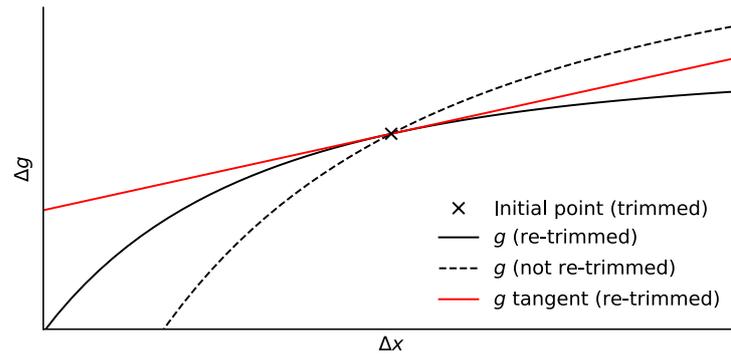


Figure 2.5: Comparison of a constraint function progression with and without implicit re-trimming.

This result is consistent with gradients of multidisciplinary systems with consistency constraints [75] or residual-based governing equations [76]. Equation 2.8 is implemented in the adjoint formulation [76], so the  $\partial \vec{g}_{trim} / \partial \vec{x}_{trim}$  matrix only has to be inverted once.

Figure 2.5 shows the comparison between the same strength constraint while being evaluated with and without re-trimming, as seen from the initial trimmed point. It is clear that without re-trimming the aircraft when changing  $x$ , the progression of the constraint function can be very different than when trimming the aircraft again when changing  $x$ . The tangent as shown in Figure 2.5 was calculated by the previously outlined coupled gradient calculation procedure. It shows that it is, indeed, representing the gradient of the re-trimmed constraint line. Therefore, the coupled gradient procedure is considered to be verified.

# 3

## Maneuver Load Control

This chapter presents the concept of Maneuver Load Control (MLC) and its integration into the XLAGrange framework. The concept itself is presented in Section 3.1. The integration of the MLC control system into XLAGrange is discussed in Section 3.2. Finally, Section 3.3 presents the maximum lift constraint.

### 3.1. Load Alleviation Concept

Maneuver Load Control (MLC), in literature also referred to as Maneuver Load Alleviation (MLA), is an active control concept with the objective to reduce wing loads during maneuvering flight and thereby providing a potential for weight savings through reduction of the required strength of the wing structure [34, 77]. MLC in general can be applied to both asymmetric maneuvers, like rolling, and symmetric maneuvers [78, 79]. Symmetric maneuvering flight refers to flight conditions where the wings are level, but where lift  $L$  is not equal to weight  $W$ . In this study, MLC is only studied for symmetric maneuvers: 2.5g pull-up and -1g push-over, which are the critical symmetric maneuver load cases to be considered for large transport aircraft according to the CS-25 regulations [33]. When an increased load factor  $n = L/W$  is required, the aircraft angle-of-attack  $\alpha$  is increased, which then increases the lift generated by the wings. In principle, the spanwise lift distribution shape then remains the same if no active measures are taken. The spanwise lift distribution is designed such that it is as close as possible to an elliptical lift distribution during cruise, as that reduces induced drag [22]. However, during maneuvering flight the objective is not necessarily to have minimum drag, as a transport aircraft typically only spends a small portion of its mission in maneuvering flight. Therefore, when applying MLC, the objective is rather to reduce structural loading.

The strategy behind MLC is to use the wing control surfaces to move the center of lift inboard during maneuvers, such that the wing root bending moment, a measure of wing structural loading, is reduced [80]. Figure 3.1 shows the principle: outboard the lift is decreased while inboard the lift is increased to compensate for the lost lift near the outboard sections, such that the total lift generated remains the same. Also shown is the influence of the maximum lift coefficient (the upper curves), showing that the maximum lift is an important constraint to consider when designing an MLC system. The effect of MLC on the lift distribution, shear force and bending moment is shown in Figure 3.2, where it is clearly seen that the bending moment over especially the inboard section of the wing is significantly reduced when using MLC. The amount of load relief that can be achieved using MLC then depends on the maximum deflection of the trailing edge control surfaces and the maximum sectional lift [12]. Nowadays, MLC is a standard feature of large transport aircraft, and it has been applied to the latest Boeing and Airbus aircraft [14, 36, 37].

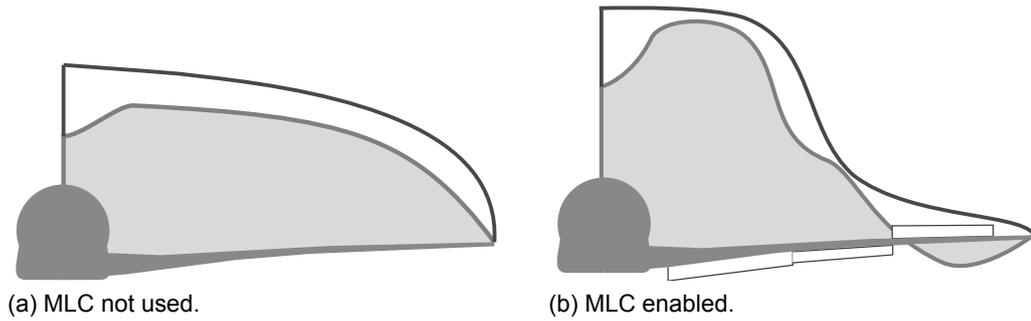


Figure 3.1: Lift redistribution principle of Maneuver Load Control, from [12].

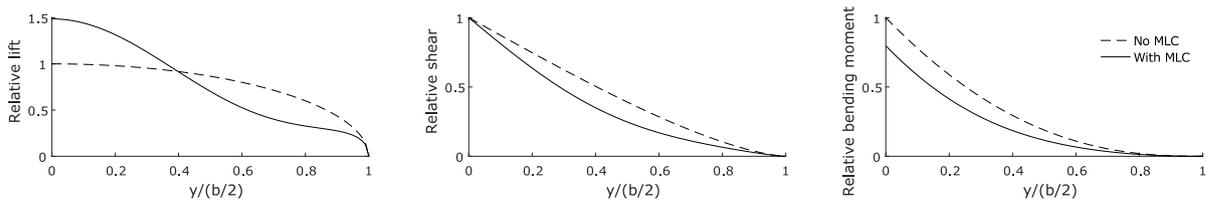


Figure 3.2: Bending moment relief principle of Maneuver Load Control.

### 3.2. MLC Control System

Starting in 1971, Boeing conducted the Control Configured Vehicles (CCV) program together with the Air Force Flight Dynamics Laboratory, where several active control techniques were tested on a B-52 aircraft, including MLC [77]. In the 1975 conference paper by Hodges [77] it is concluded that MLC could be used to achieve a reduction of the wing root bending moment of 10%. It was also concluded that care should be taken when integrating an MLC system in the flight control system. The control system consisted of a control surface deflection scheduling system based on the load factor as the scheduling variable. Vertical accelerometers placed near the aircraft's center of gravity (CG) were used to determine the load factor. To reduce interference with gust accelerations, a band-pass filter was installed.

The outlined integration issues mainly relate to the practical system-oriented integration of an MLC system into an actual aircraft and are less relevant for structural design. In an actual Flight Control System (FCS), control surfaces are actuated based on the load factor [77, 80, 81]. In structural design, however, maneuver flight conditions are considered to be steady-state flight conditions and so no interference with any dynamic control system exists, and control surface deflections for the different maneuver cases can be treated as separate design variables [11, 12, 39, 52]. It can be assumed that the quasi-steady deflections that are the output of the optimization process, can be integrated in the FCS in a later stage of the design process, without interfering with the nominal functions of the FCS [81, 82]. The control surfaces used in MLC consist of wing control surfaces, like ailerons, flaps, and spoilers [12, 81].

The control system is integrated in Lagrange by defining extra control surfaces of which the deflection is linked to design variables. The MLC discipline, implemented in the `ManeuverLoadControl` class, takes care of this, and also lets the `XLagrange` instance know that these extra defined control surfaces are in fact design variables and not trimming variables, in order not to interfere with the trimming algorithm. The integration of the MLC discipline into the optimization problem is visualized in Figure 3.3: the discipline provides control surface deflections to the static aeroelastic analysis, and then uses the analysis results to compute the maximum lift constraint.

CS-25 regulations dictate that for load cases where an active control system is used, the

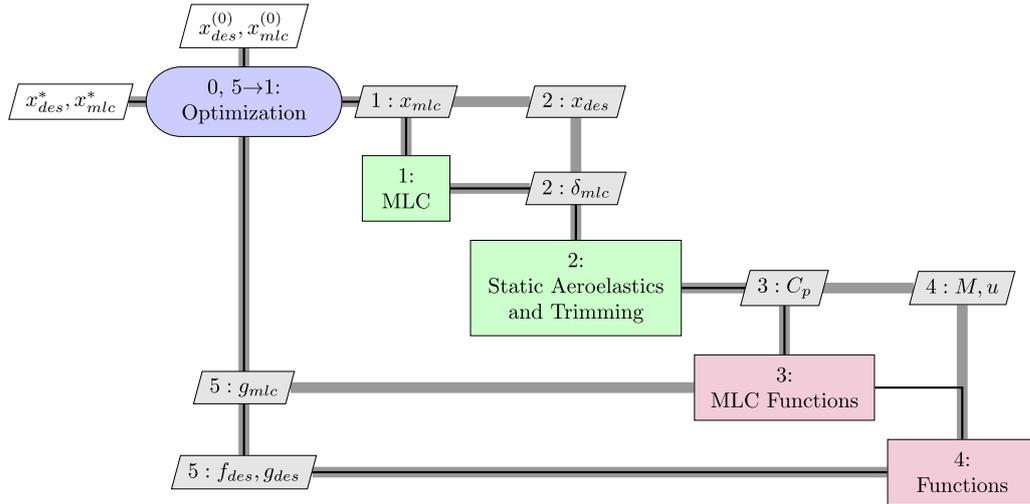


Figure 3.3: XDSM scheme of MLC integration in a static aeroelasticity MDO problem.

same load cases must also be considered with the active control system in failure mode, but then with a lower safety factor compared to the original load case [33]. The MLC discipline offers the automatic addition of such failure load cases. These load cases then are duplicates of the original load case, but without the addition of MLC, and with a modified safety factor. The modified safety factor is based on the failure probability of the control system, as specified in the CS-25 regulations [33].

### 3.3. Maximum Sectional Lift Constraint

The loss of lift of the outboard wing section when applying MLC is compensated by increasing the aircraft angle-of-attack  $\alpha$  to increase lift. This makes the maximum sectional lift in the inboard section a constraining factor in the effectiveness of MLC [12], which is especially true for lower speeds, since there the angle-of-attack and lift coefficient are higher to achieve the same lift. A maximum sectional lift constraint is offered by the MLC discipline to account for this effect.

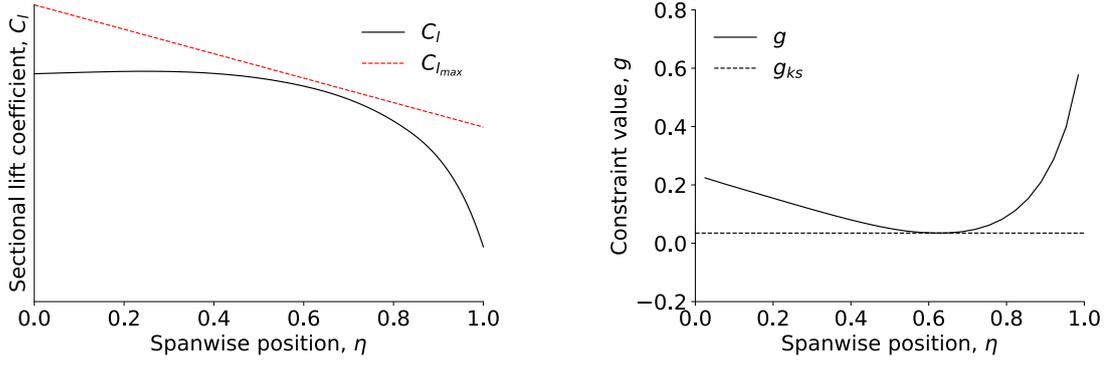
A maximum lift curve is specified by the user, which relates the spanwise position  $\eta$  of the wing to the maximum sectional lift coefficient  $C_{l_{max}}$ , which is the maximum lift coefficient  $C_{L_{max}}$  per unit span. The maximum lift constraint is then defined as

$$g(\vec{x}) = 1 - \frac{C_l(\vec{x})}{C_{l_{max}}} \geq 0, \quad (3.1)$$

where  $C_l$  is the sectional lift coefficient. This equation also works for maximum negative lift, which is relevant for the -1g push-over maneuver. One constraint is defined per strip, a set of streamwise panels, and its value is evaluated for the quantities per strip, where  $C_l$  is calculated from panel pressure coefficient  $C_p$  data from LagPy, using

$$C_l(\vec{x}) = n_z \frac{\sum C_{p_i}(\vec{x}) \cdot A_i}{\sum A_i}, \quad (3.2)$$

where  $n_z$  is the z-component of the normal vector of the strip,  $C_{p_i}$  is the pressure coefficient of panel  $i$ , and  $A_i$  is the area of panel  $i$ .  $C_{l_{max}}$  of a specific strip is found by evaluating a spline fitted through the user-provided maximum lift curve and evaluating this spline at the strip spanwise position.



(a) Sectional lift coefficient and maximum sectional lift coefficient.

(b) Associated constraint values, including the KS-aggregated value.

Figure 3.4: Notional behavior of the maximum lift constraint, as calculated from the lift distribution.

The behavior of the  $C_{l_{max}}$  constraint is visualized in Figure 3.4, where it is seen that  $C_l$  values closed to  $C_{l_{max}}$  result in lower constraint values. This is correct, since constraints are satisfied if  $g \geq 0$ . The KS-aggregated constraint value is also shown, which is the constraint value that is sent to the optimizer if constraint aggregation is used (see Section 2.3).

The constraint gradient is found by differentiating Equation 3.1 and Equation 3.2 to  $x$ :

$$\frac{\partial g(\vec{x})}{\partial \vec{x}} = -n_z \frac{\sum \frac{\partial C_{p_i}(\vec{x})}{\partial \vec{x}} \cdot A_i}{C_{l_{max}} \cdot \sum A_i}, \quad (3.3)$$

where  $\partial \vec{C}_p / \partial \vec{x}$  is the gradient of the panel pressure coefficients to changes in design variables. Panel pressure coefficients are found from

$$\vec{C}_p(\vec{x}) = \mathbf{A}_k \cdot (\mathbf{G}_{kg} \vec{u}_g(\vec{x}) + \boldsymbol{\phi}_c \vec{\delta}_c(\vec{x})), \quad (3.4)$$

where  $\vec{C}_p$  is the panel pressure coefficient vector,  $\mathbf{A}_k$  the Aerodynamic Influence Coefficient (AIC) matrix,  $\mathbf{G}_{kg}$  the structural-aerodynamic displacement splining matrix,  $\vec{u}_g$  the g-set structural mesh displacements,  $\boldsymbol{\phi}_c$  the aerodynamic control mode matrix, and  $\vec{\delta}_c$  the vector of control surface deflections. The  $k$  and  $g$  subscripts refer to the Nastran nomenclature k- and g-sets [73, 83], which represent all degrees of freedom in the aerodynamic and structural meshes, respectively. The  $\mathbf{G}_{kg}$  matrix relates structural displacements to panel  $\alpha$  values, is constructed using Infinite Plate Splines [84], and is provided to the XLAGrange framework by LagPy. Differentiating Equation 3.4 gives

$$\frac{\partial \vec{C}_p(\vec{x})}{\partial \vec{x}} = \mathbf{A}_k \cdot \left( \mathbf{G}_{kg} \frac{\partial \vec{u}_g}{\partial \vec{x}} + \boldsymbol{\phi}_c \frac{\partial \vec{\delta}_c}{\partial \vec{x}} \right), \quad (3.5)$$

where the  $\vec{u}_g$  gradient is supplied by LagPy and the  $\vec{\delta}_c$  gradient can be directly derived from the design variable type, as for an aerodynamic (MLC or trimming) design variable this gradient is 1 and for the other design variables 0.

# 4

## Gust Load Alleviation

This chapter presents the integration of Gust Load Alleviation (GLA). Section 4.1 presents the concept and strategy of integration into XLAGrange. The frequency domain and its impact on analysis is discussed in Section 4.2. The integration of GLA into XLAGrange is then covered in Sections 4.3 through 4.5. Finally, Passive Turbulence Alleviation (PTA), a static gust load reduction technique, is presented in Section 4.6.

### 4.1. Load Alleviation Concept and Integration Strategy

Turbulence was recognized as a serious problem for aircraft since the early beginnings of powered flight. Already in 1914 the first patent for a system for reducing the impact of disturbances caused by turbulence gusts was filed [85], and in 1915 the first report of the newly established National Advisory Committee for Aeronautics (NACA) contained a section about the theory behind aircraft encountering gusts [86]. It was not until the 1950s, however, that the feasibility of a Gust Load Alleviation (GLA) system was demonstrated in flight tests by Douglas Co., NACA and Avro [87, 88], although with limited success.

Gust Load Alleviation is an active load control concept which provides load relieve when encountering atmospheric turbulence [35, 89, 90]. Disturbances are sensed and reacted upon by actuating the wing control surfaces. The control system can use a feedback loop [29, 82], a feedforward channel [12, 15, 42, 90, 91], or a combination of both [43, 92].

After the catastrophic structural failure of a Boeing B-52 tail due to turbulence loads in 1964, the US Air Force initiated the Load Alleviation and Mode Suppression (LAMS) program [87, 88, 93]. This program yielded the first successful application of a GLA system, which was subsequently integrated into the ride-control systems of the B-52 and Lockheed C-5A [14, 88]. Important conclusions of this program include that care must be taken to ensure that the GLA controller does not interfere with the rest of the Flight Control System (FCS) of the aircraft, and that actuator design presented significant challenges due to high gain requirements. The Lockheed L-1011 used a GLA system employing its horizontal stabilizer for reducing gust loading leading to an extended wing span [94]. The L-1011 also featured a Maneuver Load Control (MLC) system, which confirms that GLA benefits are fully exploited only when combined with an MLC system [12, 13, 94]. GLA is currently applied to all recent Airbus and Boeing aircraft [14, 36, 37]. The Boeing 787 features a feedforward controller and the oncoming gust is sensed by static air sensors.

For this study, GLA is implemented in the `GustLoadAlleviation` XLAGrange discipline. Lagrange already contains a gust response analysis module, which can compute the displacement response based on a gust input. However, in order to add the GLA control

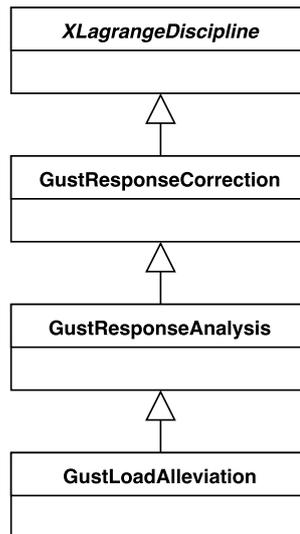


Figure 4.1: UML diagram of gust response and GLA discipline classes.

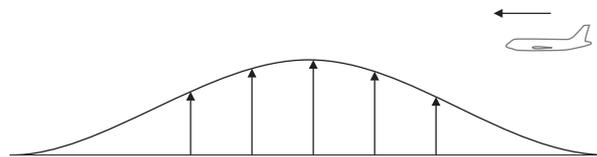


Figure 4.2: Representation of a discrete gust, from [6]: the aircraft flies through a  $1 - \cos$  shaped disturbance in airflow direction.

system to the gust response analysis, the gust response analysis is re-implemented in the XLagrange framework. Also, to compute actual loads encountered during flight through turbulence, the stresses must be computed based on the displacements of both the gust response and the 1g trimmed flight static aeroelastic displacements [6, 95]. Therefore, the GLA system is developed in three steps:

1. The `GustResponseCorrection` (GRC) discipline, which adds the displacements of the static aeroelastic and dynamic gust responses and uses these results to compute loads and strength constraint values;
2. The `GustResponseAnalysis` (GRA) discipline, which re-implements the gust response analysis of Lagrange in the XLagrange framework;
3. The `GustLoadAlleviation` (GLA) discipline, which adds the control system to the gust response analysis.

The implementation of this system is aided by the Object Oriented Programming (OOP) paradigm, allowing the sharing of functionality between the disciplines and separating discipline-specific logic and data into separate classes. The relation between the classes is visualized in Figure 4.1.

Gust and turbulence analysis can be performed for discrete gusts, where the change in angle-of-attack is described in the time domain, and for continuous turbulence, where the disturbance is specified in the frequency domain. The gust response chain integrated in the XLagrange framework is developed for time-domain discrete gust responses, where the input disturbance signal and the output displacements are all specified in the time domain. A discrete gust is specified as a disturbance in the airflow direction, usually specified as a so-called  $1 - \cos$  shaped gust [6, 33], see Figure 4.2.

## 4.2. Implications of the Frequency Domain

Gust response analysis methods can be divided into time-domain and frequency-domain methods [6, 48]. Aerodynamic forces are calculated using Aerodynamic Influence Coefficient (AIC) matrices, which relates panel angle-of-attacks to pressures. These matrices are calculated using some aerodynamic theory like the DLM method. For unsteady aerodynamics, these matrices are calculated at discrete harmonic oscillation frequencies, and thus are readily applicable for frequency-domain methods [48]. In order to use the aerodynamics for a time-domain simulation, the AICs first must be expressed as a continuous function of the Laplace parameter  $s$  [96], which is usually done using a Rational Function Approximation (RFA) method [40, 97, 98]. The implementation of an RFA method is not trivial, can introduce significant modeling errors, and is costly in terms of computational time [99, 100]. This makes the conversion of the AICs from frequency domain to time domain less suitable for optimization.

Analyzing the response in the frequency domain has the limitation that the complete system must be linear, which forces some simplifications to be made in the control system. Nonlinear actuator effects like rate and deflection saturation must be taken into account [101, 102]. However, if frequency-domain simulation is used, the only option is to constrain the system such that these limits are never reached in the first place, rather than directly simulating the nonlinearities [5, 12, 13, 52, 103], which leads to a conservative design where the benefit of the GLA system might not be fully captured [101].

In this study, gust response analysis is performed in the frequency domain to save computational time. The general solution sequence then consists of first converting the input disturbance signal from time domain to frequency domain using the Discrete Fourier Transform (DFT), then calculating the system response in the frequency domain, and at last converting the system response back to the time domain using the Inverse DFT (IDFT) [5, 73].

Another result of using the frequency domain, is that the conversion to the time domain using the IDFT yields responses that are of a periodic nature. When flying through a discrete gust, after vibrations have damped out, the aircraft will have experienced rigid body displacement [95]. This means that in reality, the displacement cannot be periodic. To solve this issue, the triple-loading technique is used [73]. Using this technique, the input time signal contains the gust three times, of which the second time the gust is applied in the opposite direction and with twice the amplitude. Then not only the accelerations will be periodic, but also the velocities and displacements. Displacements and stresses are then only extracted for the part of the output time sequence before the second application of the gust.

## 4.3. Addition of Static Loads to Dynamic Response Loads

The first step in the integration of Gust Load Alleviation (GLA) in XLAGrange is providing realistic gust response loads, which is handled by the `GustResponseCorrection` (GRC) discipline. The goal is to provide the optimization problem with strength constraints based on stresses in the wing deformed by both the static component of 1g trimmed flight and the dynamic component of the gust response. Strength constraints are based on some failure criterion value, like the Von Mises yield criterion. Since such failure criteria are in general a nonlinear function of structural grid displacements, failure criterion values cannot be summed directly. Instead, displacements are summed and these summed displacements are used for calculating loads and the related strength constraints.

The output of the frequency-domain gust response analysis is converted to the time domain using the Inverse DFT, where output is defined at  $n_t$  time steps. This means that for each time step, the displacement vector  $\vec{u}_g$ , element stresses, and strength constraint values are calculated. Strength constraints values are calculated from displacements using

$$\vec{g}(\vec{x}) = f_g(\vec{u}), \quad (4.1)$$

where  $f_g$  is a function which calculates the constraint values based on the displacement vector  $\vec{u}$ . Since the relation between displacements and strength constraints is non-trivial, and to prevent the need for programming all constraints in the XLaGrange framework, the constraint values are calculated from displacements using LagPy. Upon session initialization, the GRC discipline starts a new `LagPy` instance in the background, which is used to evaluate Equation 4.1.

The gust response is usually calculated for the application of a positive gust, but negative gusts can be sizing as well. The linearity of the gust responses allows the calculation of negative gust disturbance constraints by subtracting the positive gust response displacements from the static displacements. The displacement vector  $u$  is then calculated from

$$\vec{u} = \vec{u}_{stat} + \vec{u}_{gust} \quad (4.2)$$

$$\vec{u} = \vec{u}_{stat} - \vec{u}_{gust}, \quad (4.3)$$

where  $\vec{u}_{stat}$  is the displacement vector of the static aeroelastic load case and  $\vec{u}_{gust}$  is the displacement vector for a given time step in the gust response.

Constraint gradients are calculated using

$$\frac{\partial \vec{g}}{\partial \vec{x}} = \frac{\partial \vec{g}}{\partial \vec{u}} \frac{\partial \vec{u}}{\partial \vec{x}}, \quad (4.4)$$

where the  $\partial \vec{u} / \partial \vec{x}$  gradient is provided by LagPy or the `GustResponseAnalysis` discipline. The  $\partial \vec{g} / \partial \vec{x}$  gradient, however, is not directly available. Instead, the `XLaGrange` instance previously used for the calculation of combined constraints is also used for the calculation of constraint gradient, using a forward finite difference scheme:

$$\frac{\partial \vec{g}}{\partial \vec{u}} = \frac{f_g(\vec{u} + \partial \vec{u} \cdot \epsilon) - f_g(\vec{u})}{\epsilon}, \quad (4.5)$$

where  $\epsilon$  is the finite difference step. This results in semi-analytical gradients for corrected gust response constraints.

One evaluation of Equation 4.1 is computationally expensive in terms of time, and constraint gradients have to be calculated for each output time step, leading to a significant portion of the time needed for strength gradient calculations being spent on the evaluation of Equation 4.5. By selecting critical output time steps and only calculating gradients for these time steps, computation time can be reduced. A critical time step is a time step where any of the strength constraints are critical for the optimization problem. Since the critical time steps may change with changing system dynamics, the critical time steps cannot be determined before the optimization starts, but rather need to be determined after each iteration. To hide the potentially changing numbers of critical time steps from the optimization problem, the constraint aggregation technique of Section 2.3 is used. Normally, the aggregation of constraints is handled by the `XLaGrange` instance. However, when using the selection of critical time steps technique, the constraint aggregation is moved to the `GustResponseCorrection` discipline and only the aggregated constraints are published to the main optimization problem.

The selection of critical time steps works as follows:

1. Displacements and associated constraints are calculated normally;
2. Critical time steps are selected;
3. Constraints are aggregated time-wise: one constraint, instead of  $n_t$  constraints, remains for each structural element;

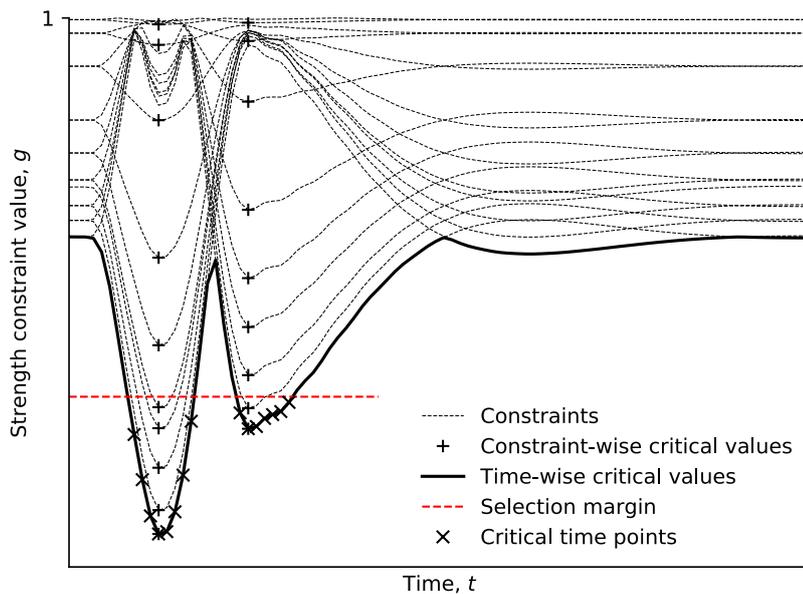


Figure 4.3: Selection of critical time steps procedure: constraint-wise critical time steps are combined with time-wise critical time steps that lie within a margin from the most critical value.

4. Displacement gradients are calculated for all time steps;
5. Constraint gradients are only calculated for the critical time steps;
6. Constraint gradients are aggregated time-wise.

The selection of critical time steps is visualized in Figure 4.3. First, critical time steps are selected for each constraint. This resulting set is then extended by taking all time steps where the most critical constraint at that time step  $g_{crit_i}$  lies within a certain margin of the most critical constraint value of all  $g_{crit_{all}}$ . This last step makes sure that time step gradients of elements near the most critical are sufficiently included in the Kreisselmeier-Steinhauser gradient calculation.

## 4.4. Frequency-Domain Gust Response Analysis

The second step in adding GLA to XLaGrange is the re-implementation of the frequency-domain gust response analysis, in the `GustResponseAnalysis` (GRA) discipline. This is needed to be able to add-in the GLA control system dynamics later on. The class extends the `GustResponseCorrection` class.

### 4.4.1. Identification of Structural Vibration Modes

For dynamic aeroelastic response analysis, structural displacements are expressed as a linear combination of the structural vibration modes, to reduce computational time [6, 95, 104]. The structural vibration modes and associated natural frequencies are the eigenvalue and eigenvector pairs, the eigenpairs, of the structural system. These are found from the standard structural eigenvalue problem [105]

$$(\mathbf{K} - \lambda_i \mathbf{M}) \vec{\phi}_{s_i} = 0, \quad (4.6)$$

where  $K$  is the stiffness matrix,  $M$  is the mass matrix,  $\lambda_i$  is the  $i^{\text{th}}$  eigenvalue, and  $\vec{\phi}_{s_i}$  is the  $i^{\text{th}}$  eigenvector. Equation 4.6 is solved for the  $n_s$  lowest frequency modes using the Lanczos algorithm [106–108]. The number of modes  $n_s$  is specified by the user. Modal vectors are normalized to the mass matrix, which results in the following identities [109]:

$$\mathbf{M}_{hh} = \boldsymbol{\phi}_s^T \mathbf{M}_{gg} \boldsymbol{\phi}_s = \mathbf{I} \quad (4.7)$$

$$\mathbf{K}_{hh} = \boldsymbol{\phi}_s^T \mathbf{K}_{gg} \boldsymbol{\phi}_s = \text{diag}(\vec{\lambda}) \quad (4.8)$$

where  $\boldsymbol{\phi}_s$  is the structural mode matrix and  $\vec{\lambda}$  is the vector of eigenvalues. The  $g$  and  $h$  subscripts refer to the Nastran nomenclature  $g$  and  $h$ -sets, which represent the global set of all structural DOFs and the generalized DOFs, respectively [110]. The structural mode matrix  $\boldsymbol{\phi}_s$  is built-up of the structural mode vectors  $\vec{\phi}_{s_i}$  in its columns.

The lowest  $n_r$  modes are the rigid body modes: modes where all masses displace in the same direction, and where the structure does not deform internally [6]. Rigid body modes have an eigenfrequency of zero and are enabled by freeing specific degrees of freedom at a structural node near the CG. Due to numerical errors in the Lanczos algorithm, the frequencies are not always exactly zero, and the mode shapes offer no resemblance to the actual rigid body shapes [106]. Therefore, clean rigid body modes are constructed using the stiffness-based approach of Nastran [110], which was developed in [106]. Clean rigid body modes are defined by specifying a nodal DOF as being the independent rigid body DOF which is followed by all other nodes of the  $f$ -set: the set of all structural DOFs not constrained otherwise. From this distinction, the  $f$ -set is separated in the  $r$ -set and the  $l$ -set, which represent the rigid body DOFs and all other free DOFs, respectively. The idea behind the construction of the clean rigid body modes is then to subject one rigid body DOF to a unit displacement at a time and calculate the displacement of the  $l$ -set DOFs due to this unit displacement of the  $r$ -set DOF. Partitioning the  $f$ -set stiffness matrix into  $r$ -set and  $l$ -set parts and realizing that no external loads act on the  $l$ -set, gives

$$\begin{bmatrix} \mathbf{K}_{rr} & \mathbf{K}_{rl} \\ \mathbf{K}_{lr} & \mathbf{K}_{ll} \end{bmatrix} \begin{bmatrix} \vec{u}_r \\ \vec{u}_l \end{bmatrix} = \begin{bmatrix} \vec{p}_r \\ 0 \end{bmatrix}, \quad (4.9)$$

where the bottom row is then rewritten to

$$\vec{u}_l = \mathbf{D} \vec{u}_r \quad (4.10)$$

$$\mathbf{D} = -\mathbf{K}_{ll}^{-1} \mathbf{K}_{lr} \quad (4.11)$$

which can then be used to construct the rigid body modes:

$$\boldsymbol{\phi}_{r1} = \begin{bmatrix} \mathbf{I} \\ \mathbf{D} \end{bmatrix}. \quad (4.12)$$

These rigid body modes are then normalized to the  $f$ -set mass matrix to comply with Equation 4.7:

$$\mu_i = \vec{\phi}_{r1_i}^T \mathbf{M}_{ff} \vec{\phi}_{r1_i} \quad (4.13)$$

$$\vec{\phi}_{r2_i} = \frac{\vec{\phi}_{r1_i}}{\sqrt{\mu_i}} \quad (4.14)$$

where the  $i$  index refers to the  $i^{\text{th}}$  rigid body mode.

The next step is to orthogonalize the modes to improve quality and improve compliance with Equation 4.7. This is done using the Gram-Schmidt procedure:

$$\vec{\phi}_{r3_i} = \vec{\phi}_{r2_i} - \sum_{j=1}^{i-1} \left( \vec{\phi}_{r2_i}^T \mathbf{M}_{ff} \vec{\phi}_{r2_j} \right) \vec{\phi}_{r2_j}, \quad (4.15)$$

which is done for all but the first rigid body mode vector. Finally, the modes are normalized again using Equation 4.13 and Equation 4.14 to yield the final normalized and orthogonalized rigid body modal matrix  $\phi_r$ . These modes are then placed into the  $\phi_s$  matrix as found by the Lanczos algorithm, replacing the first  $n_r$  modal vectors. The first  $n_r$  eigenvalues are set to zero.

One property of rigid body modes is that the mode shapes impose no internal strain on the structure. This means that the first rigid body mode matrix  $\phi_{r1}$  never changes during optimization if the nodal coordinates do not change, since they are not dependent on the current stiffness or mass matrix. Therefore, the  $\partial\phi_r/\partial\vec{x}$  gradient is calculated simply by differentiating Equation 4.13 through Equation 4.15, which then only depends on the mass matrix gradient. The gradients of the other structural modes as found by the Lanczos algorithms are calculated using Nelson's method, as presented in [111].

#### 4.4.2. Aeroservoelastic System Formulation

The aeroservoelastic gust response dynamic system is expressed in the frequency domain as [48]

$$(\mathbf{M}_{hh}s^2 + \mathbf{K}_{hh} - q\mathbf{Q}_{hh}(s))\vec{\xi}(s) = q\vec{Q}_{hg}(s)w_g(s) + (\mathbf{M}_c s^2 + q\mathbf{Q}_{hc}(s))\vec{\delta}_c(s), \quad (4.16)$$

where  $\mathbf{M}_{hh}$  is the generalized mass matrix,  $\mathbf{K}_{hh}$  is the generalized stiffness matrix,  $q$  is the dynamic pressure,  $\mathbf{Q}$  are the generalized aerodynamic force matrices,  $\vec{\xi}$  is the vector of modal coordinate displacements,  $w_g$  is the gust input signal,  $\mathbf{M}_c$  is the control surface mass matrix, and  $\vec{\delta}_c$  is the vector of control surface deflections. In this study, the control surface mass matrix is not taken into account. The generalized aerodynamic force matrices relate aerodynamic inputs, structural deformations for  $\mathbf{Q}_{hh}$ , gust input for  $\vec{Q}_{hg}$ , and control surface deflections for  $\mathbf{Q}_{hc}$ , to generalized forces:

$$\mathbf{Q}_{hk}(s) = \phi_s^T \mathbf{G}_{gk} \mathbf{P}_{aero} \mathbf{A}_k(s) \quad (4.17)$$

$$\mathbf{Q}_{hh}(s) = \mathbf{Q}_{hk}(s) \mathbf{G}_{kg,u}(s) \phi_s \quad (4.18)$$

$$\vec{Q}_{hg}(s) = \mathbf{Q}_{hk}(s) \vec{\phi}_g \quad (4.19)$$

$$\mathbf{Q}_{hc}(s) = \mathbf{Q}_{hk}(s) \phi_c \quad (4.20)$$

where  $\mathbf{Q}_{hk}$  is the generalized aerodynamic force matrix,  $\phi_s$  is the structural mode matrix,  $\mathbf{G}_{gk}$  the aerodynamic-structural load splining matrix,  $\mathbf{P}_{aero}$  the aerodynamic loading matrix,  $\mathbf{A}_k$  the AIC matrix,  $\mathbf{G}_{kg,u}$  the unsteady structural-aerodynamic displacement splining matrix,  $\vec{\phi}_g$  the gust mode vector, and  $\phi_c$  the control modes matrix. The aerodynamic loading matrix  $\mathbf{P}_{aero}$  converts the pressure coefficients  $\vec{C}_p$  as outputted by the AIC matrix to pressures, by multiplying each panel  $C_p$  with its area.

The control modes matrix  $\phi_c$  relates control surface deflections to panel  $\alpha$ 's and are constructed based on definitions in the input file. The gust mode vector  $\vec{\phi}_g$  relates the gust input signal to panel  $\alpha$ 's. The gust input is applied to the panels as a simple time delay based on the airspeed and the streamwise position of the panel relative to the starting point of the gust, and is constructed using [48]

$$\phi_{g_i}(s) = n_{z_i} e^{s(x_i - x_0)/V}, \quad (4.21)$$

where  $n_{z_i}$  is the z-component of the normal vector of the panel,  $x_i$  is the streamwise position of the panel aerodynamic center,  $x_0$  is the gust starting location and  $V$  is the true airspeed.

The unsteady AIC matrices  $\mathbf{A}_k$  are provided at certain discrete reduced frequencies  $k$  and Mach numbers  $M$  in the aerodynamic database, where in Lagrange  $k$  is defined as

$$k = \frac{\omega \bar{c}}{V}, \quad (4.22)$$

where  $\omega$  is the angular frequency and  $\bar{c}$  the reference chord length. Note that in the Lagrange definition,  $k$  has twice the value of the definition commonly used in literature and by Nastran, where  $k$  is normalized against the reference half-chord [73, 104]. For all values of  $k$  available in the aerodynamic database, the generalized aerodynamic force matrix  $Q_{hk}$  is constructed. For the frequency response analysis, the aerodynamic force matrices are interpolated between the available  $M$  and  $k$  values to yield the  $Q_{hk}$  matrices at the desired reduced frequencies. Interpolation is performed after the construction of the  $Q_{hk}$  matrices, to save time and reduce computer memory usage compared to first interpolating and then constructing  $Q_{hk}$  matrices, since the  $Q_{hk}$  matrices are much smaller than the full AIC matrices.

To calculate the gust response, Equation 4.16 is solved. At each frequency point, the left-hand-side matrix is inverted and multiplied by the right-hand-side vector, yielding the  $\xi$  vector as the gust response. Analytical displacement gradients are provided by differentiating Equation 4.16 due to  $\vec{x}$ . The gradient is strictly a function of eigenvector, eigenfrequency and control system dynamics gradients. Starting from these gradients, no further matrix inversions have to be performed, since the gradient of an inverted matrix is given by

$$\frac{\partial \mathbf{A}^{-1}}{\partial \vec{x}} = -\mathbf{A}^{-1} \frac{\partial \mathbf{A}}{\partial \vec{x}} \mathbf{A}^{-1}. \quad (4.23)$$

The displacement gradients are then used to calculate strength constraint gradients in the `GustResponseCorrection` class.

#### 4.4.3. Derivation of Unsteady Splining Matrix

The splining matrices are constructed using Infinite Plate Splines [73, 84], which relate quantities normal to one planar mesh grid to quantities normal to another. The load splining matrix  $G_{gk}$  (aero-to-struct) converts loads in the aerodynamic mesh to loads in the structural mesh. The  $G_{kg,u}$  (unsteady struct-to-aero) matrix converts unsteady displacements in the structural mesh to unsteady displacements in the aerodynamic mesh. Panel angle-of-attack as induced by structural deformation can be expressed as [109]

$$\alpha(t) = \theta(t) + \frac{\dot{z}(t)}{V}, \quad (4.24)$$

where  $\theta$  is the pitch angle,  $\dot{z}$  is the vertical velocity and  $V$  is the true airspeed of the aircraft. In the static aeroelastic case, all time derivatives are zero, so only  $\theta$  is relevant for calculating  $\alpha$ , which is transferred to the aerodynamic mesh by the  $G_{kg}$  (steady struct-to-aero) matrix. The  $G_{kg}$  matrix converts quantities in one mesh, vertical displacements, to the streamwise derivative of the quantity in another mesh,  $\alpha$ , and is therefore used to represent the static part of Equation 4.24. The vertical velocity  $\dot{z}$  can be expressed using the  $G_{gk}$  (aero-to-struct) matrix. Originally this matrix is used for converting forces from the aerodynamic to the structural mesh, but this can also be interpreted as converting quantities in one mesh directly to the other mesh, rather than taking the streamwise derivative. Since  $\dot{z}$  directly relates to  $\alpha$ , this matrix can also be used to convert vertical displacement in one mesh to  $\alpha$  due to these vertical displacements in another. Combined with taking the time derivative in the frequency domain, the following expression for  $G_{kg,u}$  is derived:

$$G_{kg,u}(s) = G_{kg} + s \frac{G_{gk}^T}{V}. \quad (4.25)$$

#### 4.4.4. XLagrange Gust Response Analysis Verification

The gust response analysis discipline of the XLagrange framework is verified by comparing the output as calculated by XLagrange and Lagrange for the same input, the results of

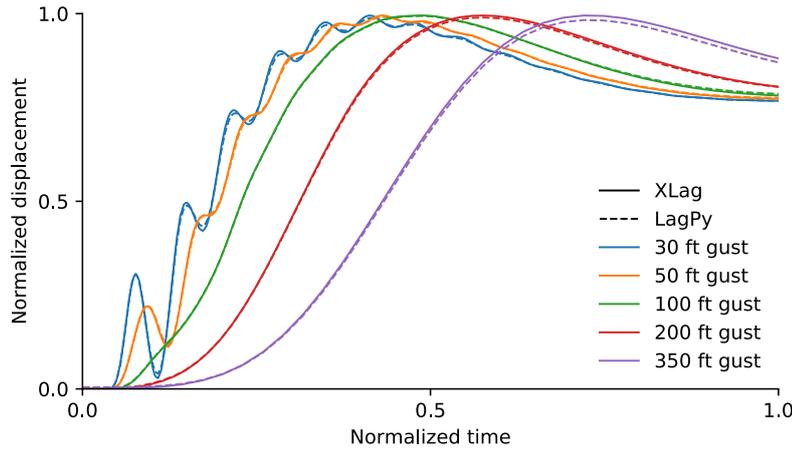


Figure 4.4: Verification of XLAGrange gust response analysis: comparison between vertical CG displacements calculated by XLAGrange and Lagrange.

which are presented in Figure 4.4. In general the responses are the same, but the two outputs deviate most near the vibration peaks for the short length gusts and in the damping-out part of the response. The deviation in the vibration peaks is not a problem since the XLAGrange solution only differs from the Lagrange solution by a small relative amount and in a conservative manner: the displacements are larger in the XLAGrange solution. The deviation in the damping-out part is not a problem either, since the damping-out part of the response is not critical for design stress calculation. The gust response analysis implementation is considered to be verified.

## 4.5. GLA Control System

The Gust Load Alleviation features are finally represented by the control system, which expresses the control surface deflections  $\vec{\delta}_c$  as a function of the gust disturbance  $w_g$  for feedforward control and  $\vec{\delta}_c$  as a function of modal coordinates  $\vec{\xi}$  for feedback control:

$$\vec{\delta}_c(s) = \mathbf{H}_a(s)\mathbf{H}_{c,ff}(s)\vec{H}_{s,ff}(s)w_g(s) = \vec{H}_{sys,ff}(s)w_g(s) \quad (4.26)$$

$$\vec{\delta}_c(s) = \mathbf{H}_a(s)\mathbf{H}_{c,fb}(s)\mathbf{H}_{s,fb}(s)\vec{\xi}(s) = \mathbf{H}_{sys,fb}(s)\vec{\xi}(s) \quad (4.27)$$

where  $\mathbf{H}_a$  is the actuator transfer function matrix,  $\mathbf{H}_c$  the controller transfer function matrix,  $\mathbf{H}_s$  the sensor transfer function matrix, and  $\mathbf{H}_{sys}$  the control system transfer function matrix. The subscripts *ff* refers to feedforward control, and *fb* refers to feedback control. For feedforward control, the sensor and control system transfer function matrices are actually column vectors, but are treated equally in computations as a vector can also be regarded as a matrix with one column. Combining Equation 4.26 and Equation 4.27 with Equation 4.16 gives the closed-loop aeroservoelastic gust response system:

$$(\mathbf{M}_{hh}s^2 + \mathbf{K}_{hh} - q(\mathbf{Q}_{hh} + \mathbf{Q}_{hc}\mathbf{H}_{sys,fb}))\vec{\xi} = q(\vec{Q}_{hg} + \mathbf{Q}_{hc}\vec{H}_{sys,ff})w_g, \quad (4.28)$$

where the dependency on  $s$  notation has been dropped for clarity.

The `GustLoadAlleviation` (GLA) class provides the interface with the XLAGrange session. The GLA class contains one `ControlSystem` instance, which contains all the control system logic and supplies the  $\mathbf{Q}_{hc}\mathbf{H}_{sys}$  matrices to the GLA class. The control system contains three types of components: sensors, controllers, and actuators. Sensors

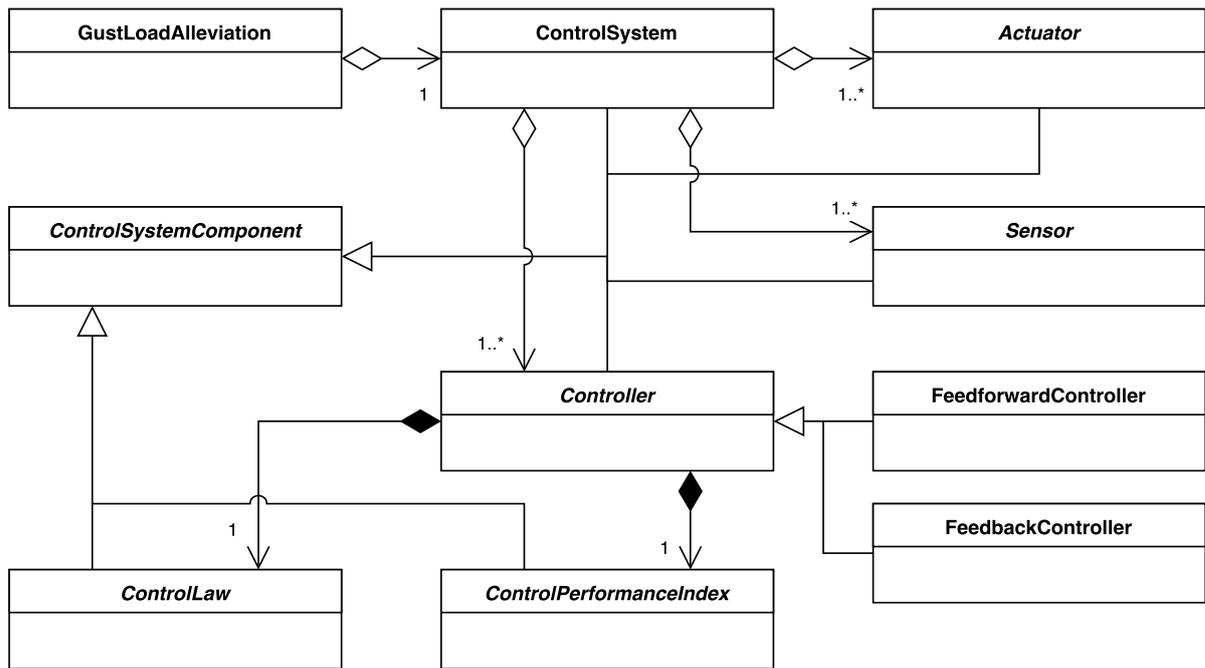


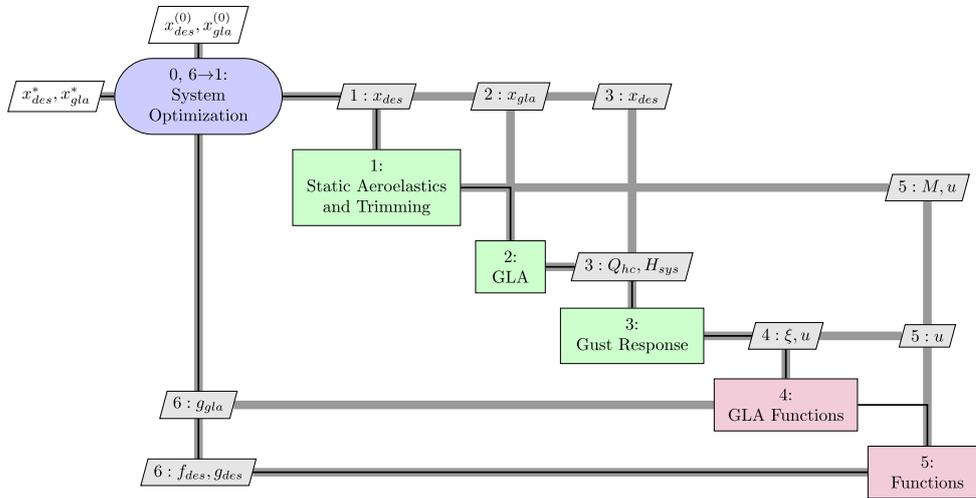
Figure 4.5: UML schema of GLA control system classes.

generate input signals for the controllers, based on structural displacements or the gust disturbance signal. Controllers consist of a control law, which constructs the controller transfer function matrix, and a performance index, which is used for tuning the controller using optimal control methods. Performance indices are implemented to enable investigation into the differences between using optimal control methods for controller tuning, and not using them, as is partly the topic of this thesis. Actuators represent the actuation system, converting actuator commands to control surface deflections. Figure 4.5 presents the UML schema of all classes involved in modeling the GLA control system. All control system components extend the `ControlSystemComponent` class, which defines a standard interface for interacting with components: it for example defines functions that return the relevant transfer function matrices, and functions for the definition of design variables and constraints. The OOP approach enables great flexibility in setting up the control system, where different sensors, controllers, control laws, performance indices, and actuators can easily be swapped out.

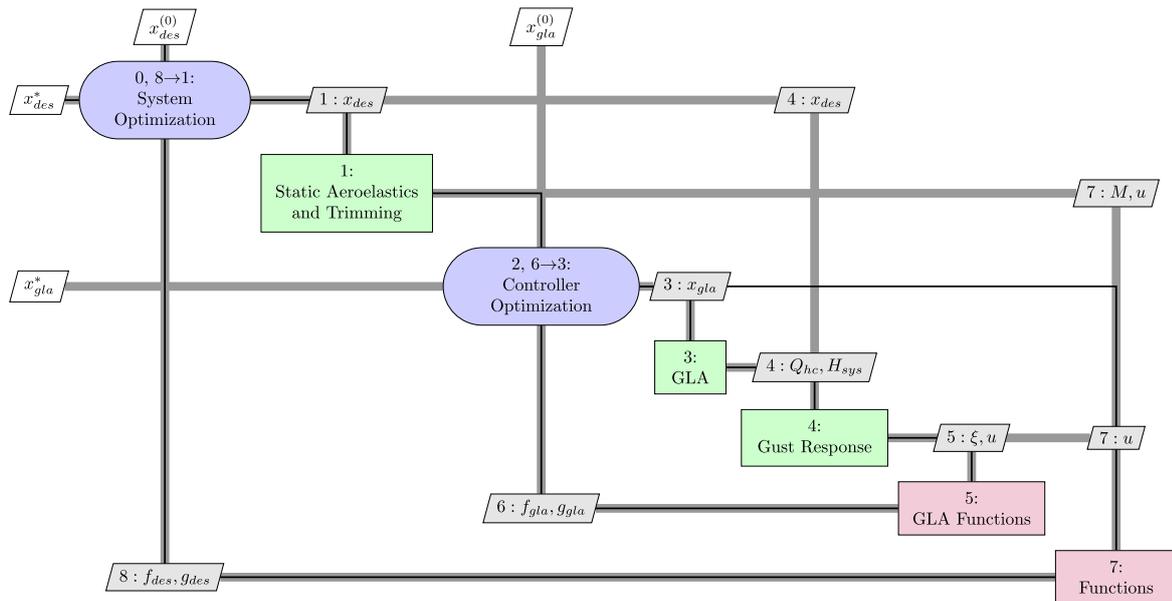
Analogous to Maneuver Load Control (MLC), see Section 3.2, the Gust Load Alleviation discipline also allows the automatic addition of failure load cases. These load cases are duplicates of the gust load cases used for GLA, but then without GLA present, and with a different safety factor.

#### 4.5.1. Controller Tuning in Structural Optimization

The GLA controller is tuned by modifying its transfer function parameters, which are directly linked to the controller design variables. In structural optimization, the controller design variables can be varied in two ways: using direct mode and using in-the-loop tuning (ITLT) mode. In direct mode, controller parameters are included in the main structural optimization problem. In ITLT mode, the controller parameters are tuned using the controller performance index in a sub-optimization loop, which relates to the theory of optimal control [112, 113]. The different design modes are visualized by XDSM schemes in Figure 4.6. The direct and ITLT modes roughly correspond to the Individual Discipline Feasible (IDF) and Asynchronous Subspace Optimization (ASO) MDO architectures [23], respectively. In ITLT mode, analytical



(a) Direct mode: controller parameters are tuned by the system optimizer.



(b) In-the-loop tuning mode: controller parameters are tuned in a sub-optimization loop, based on a selected performance index.

Figure 4.6: Comparison between GLA controller design XDSM schemes.

gradients are not available, since for ASO analytical gradients require the availability of second order constraint and objective gradients [53], which are not available at the moment.

Controller design variables can also be tuned outside of a structural optimization loop, in which case it is called controller pre-tuning. The controller is pre-tuned using the sub-optimization loop of Figure 4.6b.

#### 4.5.2. Control Laws and Performance Indices

Three different control laws are available: Proportional-Derivative (PD) control [114], the Infinite Impulse Response (IIR) filter, and the Finite Impulse Response (FIR) filter [43]. PD is the simplest form of control, but IIR and FIR offer better transfer function shaping performance. IIR is able to approximate desired transfer functions with less coefficients, i.e. design variables, than FIR filters, but FIR filters can never go unstable since they do not have poles.

Three different performance indices are available: Linear Quadratic Regulator (LQR) [52, 82, 115], Power Spectral Density (PSD) [43] and  $H_\infty$  [116, 117]. Normally, these performance indices are minimized by solving an algebraic equation that includes the definition of the plant in the solution. For that, however, the plant needs to be expressed as a continuous function of the Laplace parameter  $s$ , which is not available in the frequency-domain gust formulation used in this thesis. The performance indices can also be minimized using conventional numerical optimization techniques, which is what is done when in-the-loop tuning is used or when the controller is pre-tuned.

The LQR performance index is evaluated based on the sensor and actuator time-domain responses, and therefore cannot guarantee the performance of the GLA control system for input different than the one used for the gust response analysis. The PSD and  $H_\infty$  methods are frequency-domain methods, acting on the system transfer functions. The PSD method is relatively cheap to evaluate, but does not take MIMO interference effects into account. Also, because it acts on the whole transfer function, using the PSD method can result in the amplification of certain responses as long as the total PSD is reduced. The  $H_\infty$  method, however, calculates its performance index based on the  $\infty$ -norm of the system transfer function, which does take MIMO effects into account correctly. The  $\infty$ -norm can be interpreted as the maximum gain peak, and the minimization of the  $\infty$ -norm therefore reduces the response of the system, which might be closer to the load alleviation objective of GLA.

#### 4.5.3. Sensors and Actuators

Several different types of sensors are offered, of which the most important are the acceleration, displacement, and  $\alpha$  sensors. Accelerometers are used as feedback and  $\alpha$ -probes as feedforward sensors. Displacement sensors cannot be realized in reality and should therefore not be used as sensors for actual controller input, but they can be used as system tuning sensors, where the performance index of controllers can be based on the transfer function from gust input to the displacement sensors. This can be advantageous, as displacements give a more direct relation to stresses than accelerations. Multiple sensors of the same types can be linearly combined into one controller input signal using the `SensorFusion` class. Using this class, the root bending moment can for example be approximated by subtracting CG vertical displacements from wing tip vertical displacements.

Currently only one type of actuator is implemented, the `SimpleActuator`, which models the actuator dynamics as a simple second order low-pass filter [118]:

$$H_{act}(s) = \frac{\omega_a^2}{s^2 + 2\zeta_a\omega_a s + \omega_a^2}, \quad (4.29)$$

where the actuator damping ratio  $\zeta_a$  is set to 0.65 [119] and the actuator natural frequency  $\omega_a$  is based on a user-supplied maximum deflection rate  $\delta_{max}$ . Assuming the maximum deflection

rate is achieved at a harmonic oscillation of the control surface at the natural frequency, the natural frequency  $\omega_a$  is found from

$$\omega_a = 2\pi \frac{2\dot{\delta}_{max}}{\delta_{max}}, \quad (4.30)$$

where  $\delta_{max}$  is the maximum control surface deflection range. Constraints are posed on the actuator output so that the posed actuator deflection limits are not violated. Actuator deflection rate limits are enforced by the low-pass actuator transfer function.

#### 4.5.4. Feedback Control System Constraints

If feedback control is used, two types of constraints are posed on the the feedback loop: stability constraints and sensitivity constraints. Stability constraints make sure that the Multi-Input Multi-Output (MIMO) aeroservoelastic system is stable. Sensitivity constraints make sure that the individual transfer functions compromising the open-loop system are robustly stable.

##### Stability Constraints

The stability constraints are based on theory developed by Keel et al. [120], where stability is determined purely using the data-driven frequency response of the plant, which makes it suitable for frequency-domain GLA controller synthesis. The basis of the method is that the order of the plant  $r_p$  corresponds to the total phase shift  $\Delta_0^\infty$  for  $\omega = 0 \rightarrow \infty$  [120]:

$$\Delta_0^\infty = -\frac{\pi}{2} (r_p + 2(z^+ - p^+)), \quad (4.31)$$

where  $z^+$  and  $p^+$  represent the number of right-hand-side (RHS) zeros and poles, respectively. The total phase shift can be determined directly from the transfer function bode phase plot, and  $r_p$  is found from the slope of the transfer function at  $\omega \rightarrow \infty$  [120]:

$$r_p = -\frac{1}{20} \left. \frac{dP_{db}(\omega)}{d(\log_{10} \omega)} \right|_{\omega \rightarrow \infty}, \quad (4.32)$$

where  $P_{db}$  is the plant transfer function magnitude in dB.

The stability check is performed on each of the Single-Input Single-Output (SISO) transfer functions of the MIMO Youla parameter  $Q$  [113, 121]. The Youla parameter is a convenient parameterization of the feedback system, because if the Youla parameter is found to be stable, this means that the feedback loop is internally stable as well [121]. The Youla parameter is found using

$$Q = (I + CP)^{-1}C, \quad (4.33)$$

where  $C$  is the controller transfer matrix and  $P$  is the plant transfer matrix. Actuators and sensors are included in the plant transfer matrix, so that  $P$  relates actuator input to sensor output, and  $C$  relates sensor output to actuator input.

From Equation 4.31 and the knowledge of  $\Delta_0^\infty$  and  $r_p$ , the difference between the amount of RHS zeros and poles can be found, not the absolute number of RHS zeros and poles at the same time. However, only RHS poles make a system unstable. To be able to use the method of Keel for determining whether the GLA feedback system is stable or not, assumptions have to be made about the properties of the transfer functions. It is assumed that:

- $C$  is stable, because it does not make sense to implement an unstable controller;
- $P$  is stable, because it represents a dynamic aeroelastic response system below the flutter instability margin.

This means that both  $C$  and  $P$  have no RHS poles, and therefore neither has the open-loop system  $I + CP$ . Closing the feedback loop turns RHS zeros into RHS poles and vice-versa [116]. And since the closed-loop system  $(I + CP)^{-1}$  has no RHS zeros ( $z^+ = 0$ ), as the open-loop system has no RHS poles, Equation 4.31 can be written as:

$$p^+ = \frac{\Delta_0^\infty}{\pi} + \frac{1}{2}r_p. \quad (4.34)$$

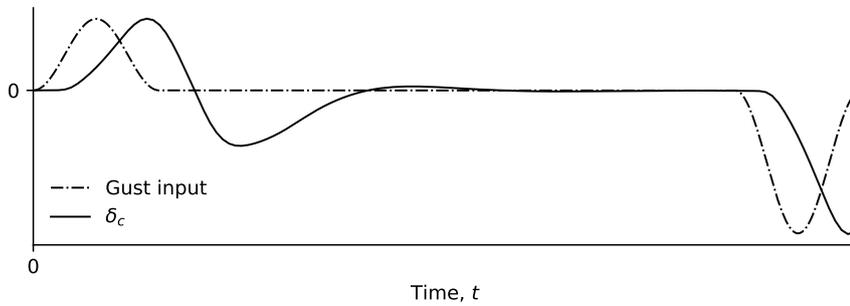
Therefore, if from Equation 4.34 it is found that  $p^+ \neq 0$ , the feedback control system is unstable.

The method of Keel et. al. determines whether the system is stable or not, but not how close it is to stability or instability. The margin to instability, however, is needed to construct a constraint function with a continuous gradient, as is needed for gradient-based optimization. To construct the constraint function, for each controller parameter  $x_{ctrl_i}$ , a line search is performed to find the value of that parameter where the system changes between stable and unstable:  $x_{ctrl_i}^*$ . The constraint function is then constructed based on the difference between  $x_{ctrl_i}$  and  $x_{ctrl_i}^*$ .

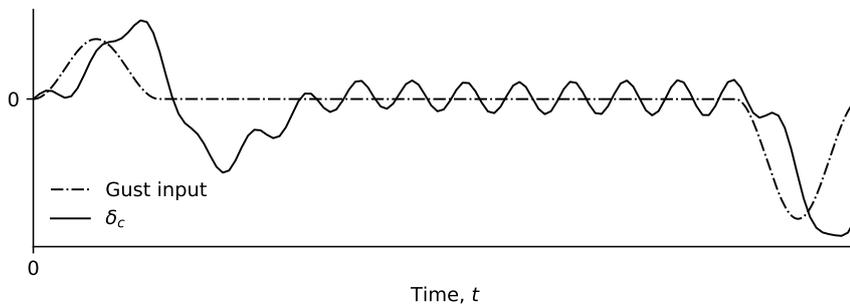
Figure 4.7 shows the relation between the stability constraint values and the system response of a Proportional-Derivative (PD) controller, where the derivative term gain is varied to change the stability of the system. It can be seen that the stability constraint  $g_{stab}$  is clearly able to identify unstable systems. One interesting observation is that an unstable system of which the response is calculated using frequency domain methods will result in bounded output, which is contrary to the usual definition of unstable systems that states that unstable systems result in unbounded output [114, 122]. The bounded output comes from the inverse Fourier transform used to calculate time-domain output from the frequency-domain response, as discussed in Section 4.2. Rather than the response becoming unbounded, the control system starts acting as a predictor and starts reacting to the system input before the input occurs, which violates the principle of causality. To show the predicting behavior, the application of the second gust in the triple-loading input is also shown in Figure 4.7. The figure shows that for the unstable system, the actuator response is already responding before the gust is applied, whereas for the stable system the response is well damped out before the application of the second gust, and actuator response only happens after the application of the gust. For the marginally stable case, the actuator response is a an undamped oscillation. The large response near the gust input is due to the stable proportional gain of the PD-controller used in the example. Compared to the unstable response, the marginally stable response has a larger sustained oscillation amplitude. This is because the gain peak in the transfer function approaches infinity near the point where the system becomes unstable: for a SISO system this would be near the -1 point in the Nyquist plot. However, when the open-loop transfer function then moves away from the instability point, the gain peak reduces again and the response magnitude is reduced.

### Sensitivity Constraints

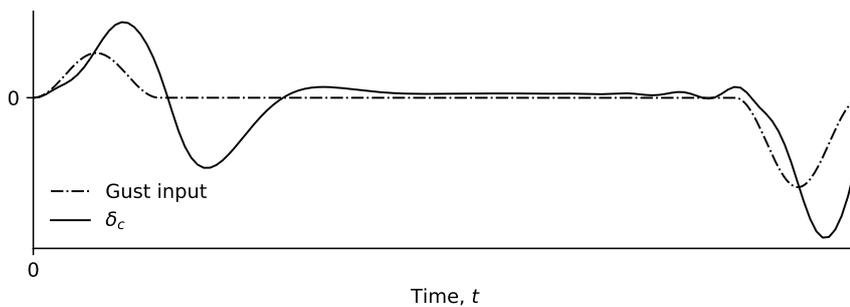
The sensitivity, or robustness, constraints are based on Single-Input Single-Output (SISO) Nyquist robustness theory [114], and therefore do not fully guarantee that the closed-loop MIMO system is robustly stable [116], however it serves as a good first approximation. Also, no analytical model of the plant is needed: availability of the frequency response suffices. Theory exists to guarantee closed-loop stability for MIMO systems, however for that a rational transfer function representation of the plant is needed, which is not available without performing Rational Function Approximation (RFA) of the Aerodynamic Influence Coefficients (AIC) [48]. One option, however, might be to use the  $b_{P,C}$  parameter [123], which is a metric based on the reciprocal of the  $\infty$ -norm of the combination of several transfer functions representing all possible interactions between the plant and the controller. This metric approaches zero



(a) Stable system,  $g_{stab} > 0$ : response dampens out.



(b) Marginally stable system,  $g_{stab} = 0$ : response does not dampen out.



(c) Unstable system,  $g_{stab} < 0$ : system acts as a predictor, but response does not become unbounded due to use of inverse Fourier transform.

Figure 4.7: Comparison of actuator responses ( $\delta_c$ ) in a GLA feedback system, with scaled gust input until the second gust of the triple-loading technique plotted as reference.

near the point where the system becomes unstable and correctly accounts for possible MIMO interaction effects. However, the function cannot become negative, which is needed to let the optimizer know the constraint is violated, so it is not suitable for optimization. Also, if this metric would be used as a sensitivity constraint, some limit has to be set on the value of the  $b_{P,C}$  metric, which is rather arbitrary and does not relate to more intuitive quantities like the gain and phase margins.

The sensitivity constraints are based on the distance to the -1 point in the Nyquist plot for the open-loop transfer function [114]. Normally, for that the sensitivity function is used, however for the purpose of creating a feasible constraint function to be used in optimization, the distance to the -1 point is used directly. This creates a function which reduces in value, the closer to instability the system becomes, but then “bounces” back up when the system becomes unstable. To guide the optimizer towards robust stability, the function is negated in case of instability, to yield a smooth function with a continuous gradient. The constraint function is then translated down, to create a feasible region for distances to the -1 point equal to or greater than  $1/M_s$ , where  $M_s$  is the maximum sensitivity as described in literature [114].

### Feedback Controller Design Space Exploration

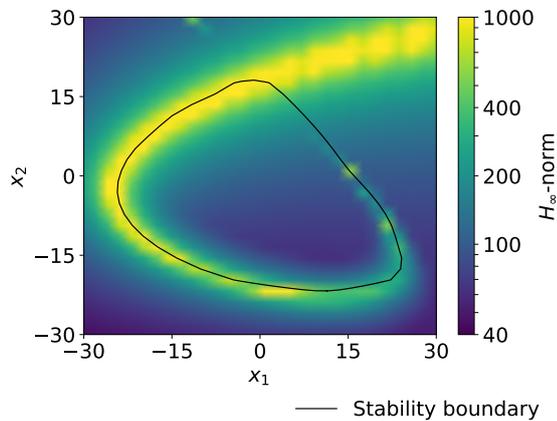
To verify whether the feedback stability and sensitivity constraints are behaving as expected, and in a manner that is appropriate for gradient-based optimization, a design space exploration is performed. A generic aeroservoelastic system with a feedback GLA controller that controls the aileron deflection based on wing tip accelerations using an eight-coefficient FIR filter is considered. The design space of the first two coefficients is explored, keeping the remaining coefficients at zero. The  $H_\infty$ -norm, sensitivity constraint, and the two stability constraints corresponding to the two varied coefficients are plotted in Figure 4.8. The  $H_\infty$ -norm is relevant as it can be used as the performance index to be minimized during controller tuning.

As can be seen in Figure 4.8, the design space is marked off by the stability boundary, which is determined from the stability constraints, and overlaid on the sensitivity and  $H_\infty$  plots to show their relation with the stability boundary. It can be seen in Figure 4.8a that on the stability boundary, the  $H_\infty$ -norm tends towards higher values. These higher-value boundaries can cross each other, which is caused by different transfer function peaks taking over from each other from being the highest peak. From control theory, it is known that at the stability boundary, the  $H_\infty$ -norm should be infinite [116]. However, due to the fact that no analytical transfer function of the plant is available, the  $H_\infty$ -norm can only be calculated at discrete frequencies, and therefore the calculated value of the norm can never actually be infinite [113]. Also, by definition, the  $H_\infty$ -norm has a value of  $\infty$  for an unstable system [116], however in the XLaGrange framework the performance indices are not aware of the feedback control constraints. Therefore, the  $H_\infty$ -norm acts more as a measure of how close the system is to the stability boundary, and thus has a gradient away from the stability region if the system is unstable. To prevent the optimization algorithm from becoming stuck in the unstable region, the stability and sensitivity constraints should exhibit behavior that guides the optimizer back towards the stability region.

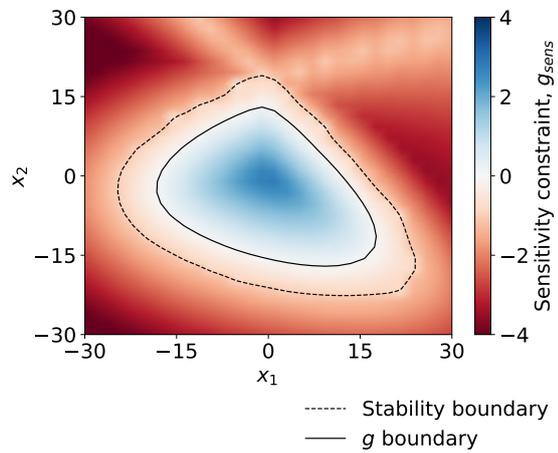
Due to the nature of the stability constraints, every controller parameter has one corresponding stability constraint function. For the two parameters varied in the design space exploration, the corresponding functions are plotted in Figures 4.8c and 4.8d. The stability constraints are well able to find the stability boundary and to provide a gradient in the direction of the stability region. The switching of the gradient sign in the middle of the stability region is not of concern, as constraints are only relevant to the optimizer if they are violated or close to being violated. What is more reason for concern is that if on the line search no stability boundary is found, the constraint function gets a static value of  $-1$ , thereby not providing the optimization algorithm with a gradient towards the stability region. This effect can be seen

near the  $x_2$  bounds in Figure 4.8c and near the  $x_1$  bounds in Figure 4.8d.

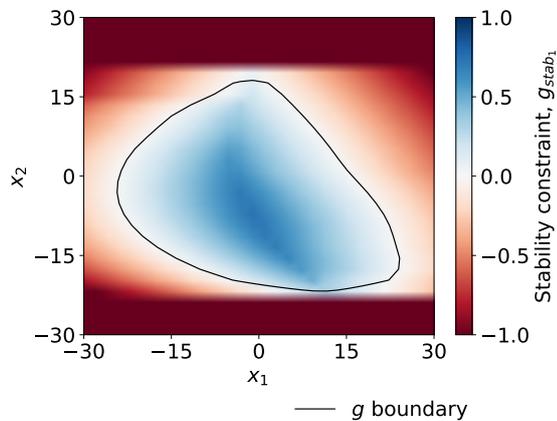
The sensitivity constraint, plotted in Figure 4.8b, correctly constrains the design space to be within a margin, the robustness margin, of the stability boundary. Outside the stability region, the sensitivity constraint provides a gradient towards the stability region at all points in the design space. Therefore, the sensitivity constraint should act as a way for the optimizer to be guided back to the feasible stability region if the stability constraints are stuck at  $-1$ .



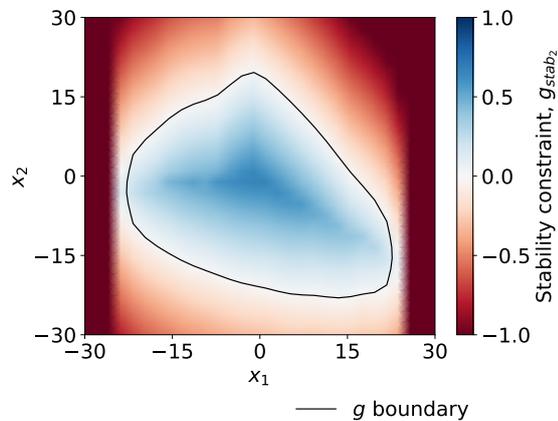
(a)  $H_\infty$ -norm: the objective function of the optimization problem.



(b) Sensitivity constraint function: ensures a robustness margin between optimal point and becoming unstable.



(c) Stability constraint function for  $x_1$ : line search over  $x_1$  to determine location of stability boundary.



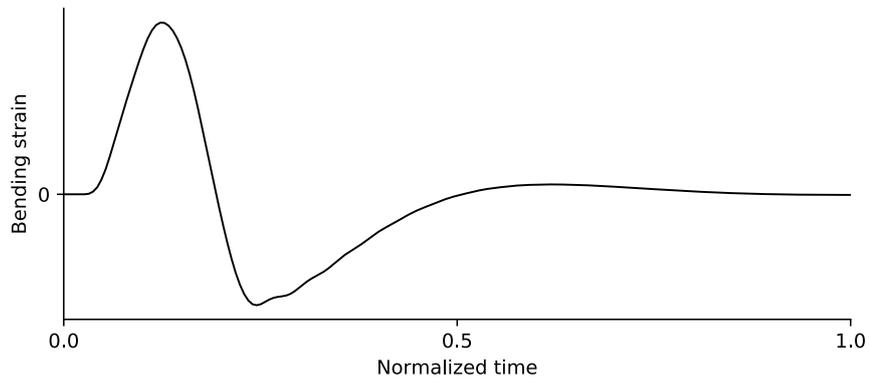
(d) Stability constraint function for  $x_2$ : line search over  $x_2$  to determine location of stability boundary.

Figure 4.8: Design space exploration of first two variables ( $x_1$  and  $x_2$ ) in an 8-coefficient FIR filter feedback controller tuning optimization problem.

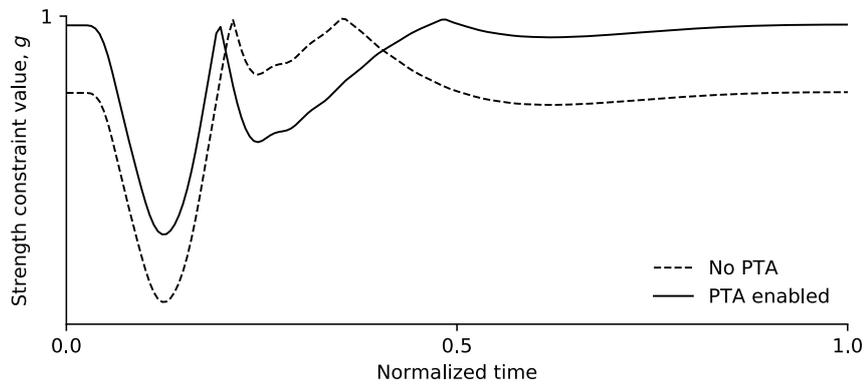
#### 4.6. Passive Turbulence Alleviation Concept

Passive Turbulence Alleviation (PTA) is a passive load alleviation system used for reducing loads due to turbulence [36, 124], and can therefore also be considered a form of gust load alleviation. It can be combined with the dynamic GLA system to achieved greater load reduction in flight through turbulence and works using a principle similar to Maneuver Load Control (MLC) (see Section 3.1): wing control surfaces deflect statically when turbulence is detected, in order to move the lift inboard, thereby reducing the wing bending moment [103]. Since lift is reduced on the outboard wing section, the wing can become more susceptible to negative gusts. PTA is implemented by using the MLC discipline on the static aeroelastic load case used for computation of the static loads in the gust response analysis.

To demonstrate the effect of PTA on the strength constraints, a representative gust response is calculated with and without the application of PTA. The result is presented in Figure 4.9. It can clearly be seen that the application of PTA can reduce the steady-state stress, apparent from the increased constraint value, which denotes lower stress. This is in agreement with the effect of MLC. PTA has no effect on the gust response bending strain, since PTA only affects the steady-state aeroelastic solution, and not the gust response. Strength constraints are defined as a function of some failure criterion like Von Mises stress, where a value of 1 denotes no stress and a value of 0 denotes that the element is about to fail. Since a failure criterion acts on the stress magnitude, it cannot become negative, meaning the strength constraint cannot rise above 1. This results in the mirroring effect as seen in the figure: once the response reaches 1, it simply mirrors and bounces down again. Application of PTA then reduces stress acting in the same direction as the static stresses, but might increase stresses acting in the opposite direction. Therefore the application of PTA clearly involves a trade-off to be solved by the optimizer.



(a) Gust response bending strain: not affected by PTA, since PTA only acts on the static aeroelastic displacements.



(b) Strength constraint values: PTA reduces steady-state stresses and gust response stresses acting in same direction, but increases stresses acting in opposite direction. A constraint value of 1 indicates no stresses in element.

Figure 4.9: Effect of Passive Turbulence Alleviation (PTA) on strength constraints due to gust.



# 5

## Numerical Experiments Setup

This chapter discusses the setup of the structural optimization studies performed to investigate the effects of Maneuver Load Control (MLC) and Gust Load Alleviation (GLA). The aircraft model is described in Section 5.1. Section 5.2 presents the optimization problem formulation. The methodology for assessing the impact on computational time is discussed in Section 5.3.

### 5.1. Aircraft Model Description

Optimizations are performed on the wing of a representative conventional large transport aircraft design. A linear structural Finite Element (FE) model is developed, where the wingbox is represented using shell elements, constructed out of aluminum 7050 [69, 125]. Stringers are not modeled, as buckling behavior mainly drives their design [69, 126], and buckling is not considered in this optimization study. The optimized thicknesses of the spars and skin thus represent smeared properties. and to correct for this, the material yield stresses are multiplied by an empirical correction factor  $k_e = 0.8$  [45, 127]. Ribs are included at every spanwise element boundary to prevent a change in cross-sectional shape, but are assigned no structural mass and are not sized, as their thickness is mainly based on buckling behavior [52]. Optimizations are performed only for the Maximum Take-Off Weight (MTOW) mass case.

Aircraft aerodynamics is modeled using a panel method mesh with Aerodynamic Influence Coefficients (AICs) generating by an in-house Doublet Lattice Method (DLM) [128] solver. The DLM is a linear panel method [129], and because of this, several aerodynamic phenomena are not modeled. Effects that are not modeled include viscous drag, nonlinear compressibility effects, boundary layers, and flow separation [130]. The aerodynamic mesh is coupled to the structural mesh using infinite-surface splines [84], coupled to structural nodes distributed along the wing chord, which are rigidly connected to the wingbox center line nodes. Only horizontal surfaces, the wing and the horizontal tail plane, and the engine nacelles are included in the aerodynamic model and the fuselage is not modeled explicitly. Fuselage aerodynamic influence is modeled by the inclusion of aerodynamic panels spanning the wing-fuselage intersection. Wing incidence is modeled by static angle-of-attack values imposed on the relevant wing panels, since DLM panels themselves are strictly oriented in the streamwise direction.

Ailerons and spoilers are used for MLC, while only ailerons are used for GLA. Accelerometers are placed at the wing tip and the CG, and the signal fed into the feedback controller is the difference between the two. An angle-of-attack sensor with a 4 Hz bandwidth [43] is placed near the nose of the aircraft for feedforward control. Both feedback and feedforward control systems essentially are SISO systems, since only one actuator and one sensor is used

for control in both cases. For tuning of the controllers, displacement sensors are placed at the same locations as the accelerometers. For the actual control system, however, displacement sensors cannot be used as these are not available in real-life applications. For GLA, aileron actuators are modeled as second-order low-pass filters, with a bandwidth corresponding to a maximum deflection rate of 80 deg/s [41, 131], which is slightly on the higher side of the spectrum for large transport aircraft, but compensates for the fact that deflection and deflection rate saturation is not available in the performed gust response analysis, see Section 4.2. The feedback and feedforward controllers are modeled as Finite Impulse Response (FIR) filters [42, 43] with 8 coefficients each. The control system has a sensor sampling frequency of 25 Hz, and a time delay of 1.5 samples [43, 132] is implemented to account for computation and measuring delays.

## 5.2. Optimization Problem Formulation

Numerical optimization in general, is defined as the minimization or maximization of an objective function by modifying design variables, while satisfying equality and inequality constraints, and can be expressed as [23, 24]

$$\begin{aligned} & \text{minimize} && f(\vec{x}) \\ & \text{with respect to} && \vec{x} \\ & \text{subject to} && g_i^c(\vec{x}) = 0 \\ & && g_i(\vec{x}) \geq 0 \end{aligned}$$

where  $f$  is the objective function,  $\vec{x}$  the design variable vector,  $\vec{g}^c$  the vector of consistency constraints, and  $\vec{g}$  the vector of design constraints. More traditionally, design constraints can be separated into equality and inequality constraints. However, it generally holds that for every equality design constraint, one design variable can be eliminated, as an equality constraint represents a direct relation between design variables [24]. For the optimizations performed in this thesis, design constraints are only of the inequality type, and are satisfied when  $g_i(x) \geq 0$ . Consistency constraints are always equality constraints [23].

In the context of Multidisciplinary Design Optimization (MDO), the structural optimization as performed using Lagrange can be seen as one discipline in an aerostructural optimization problem. However, since MDO problems are recursive in nature [67, 75], the structural discipline itself can also be expanded into multiple sub-disciplines. In this study, the analysis model is separated in the following disciplines: static aeroelastic analysis, aeroelastic trimming, MLC, gust response analysis, GLA and Passive Turbulence Alleviation (PTA).

The optimization problems will be solved by the gradient-based SQP algorithm, interfaced through the ScipyOptimizer implementation in OpenMDAO. The rest of this section presents the load cases, objective, design variables, and constraints used in the optimizations.

### 5.2.1. Load Cases

In this study, the wing is sized for maneuver loads and gust loads, as specified in the CS-25 regulations [33]. Maneuver loads include the quasi-steady 2.5g pull-up and the -1g pull-over as symmetric maneuver loads (CS-25.337), and the full aileron deflection as rolling maneuver load (CS-25.349). Gust loads (CS-25.341) are evaluated for a series of discrete 1 – cos shaped gusts of lengths 30 ft, 50 ft, 100 ft, 200 ft, and 350 ft. More emphasis is placed on the shorter gust lengths, since there potential interaction with the wing vibration modes is higher [69]. Continuous turbulence is not taken into account in this study. Gusts are assumed to be spanwise uniform [133, 134]. Both positive and negative vertical gusts are considered, and lateral gusts are not considered. Since rigid body motion must be taken into account for accurate results [6, 95, 135] and only vertical spanwise uniform discrete gusts are considered,

Table 5.1: Load cases used in the optimization problems.

Nr.	Analysis type	Speed [33]	Load case type
1	Static Aeroelastic	$V_A$	2.5g pull-up maneuver
2	Static Aeroelastic	$V_D$	2.5g pull-up maneuver
3	Static Aeroelastic	$V_H$	-1g push-over maneuver
4	Static Aeroelastic	$V_C$	-1g push-over maneuver
5	Static Aeroelastic	$V_A$	Full aileron deflection maneuver
6	Static Aeroelastic	$V_C$	Trimmed flight at gust speed
7	Gust Response	$V_C$	30 ft discrete gust
8	Gust Response	$V_C$	50 ft discrete gust
9	Gust Response	$V_C$	100 ft discrete gust
10	Gust Response	$V_C$	200 ft discrete gust
11	Gust Response	$V_C$	350 ft discrete gust

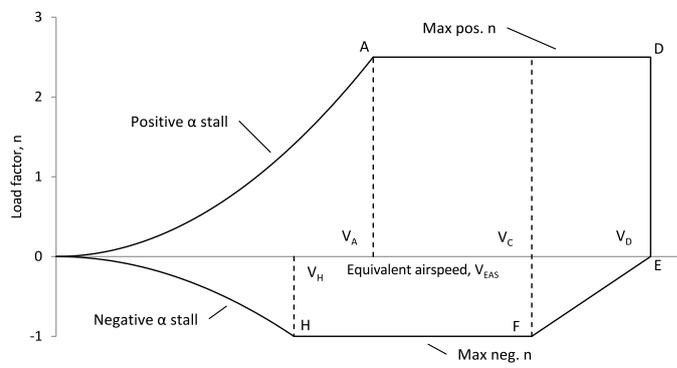


Figure 5.1: A typical CS-25 maneuver diagram, based on [33].

only the pitch and plunge motion is left free in the gust response analysis [95]. The first 30 structural modes are used for calculations. In all static aeroelastic load cases, aircraft rigid body motion is prevented by clamping the model in the Center of Gravity (CG).

Other aeroelastic effects like buffet, control effectiveness and divergence are not taken into account. Buffet is not sizing for structural design [69, 136], control effectiveness issues are solved by not using outboard ailerons for roll control at high speeds [21, 22], and divergence is not critical for swept-back wings [6, 137]. Other potentially sizing loads, like taxi and propulsion loads [6], are also not taken into account.

Table 5.1 lists the load cases. Symmetric maneuver loads are evaluated at the corners of the maneuver flight envelope, see Figure 5.1. The total lift is the same for a given load factor, but the lift distribution changes with speed, depending on wing twist distribution, wing bending, and required lift coefficient. This means that for the same load factor, different parts of the wing might be sized for different speeds. Steady-state pitch rates for the pull-up and push-over maneuvers are included in the analysis. The steady-state rolling maneuver load case is analyzed with an aileron deflection trimmed for a steady-state roll rate of 15 deg/s at  $V_A$  in order to meet CS-25 controllability requirements [41]. The rolling maneuver will be evaluated at a load factor of 1.67, which is two thirds of the maximum symmetric load factor of 2.5, in accordance with CS-25.349(a). The gust responses are evaluated at cruise speed  $V_C$ , since there the gust response is expected to be the most critical [127, 138].

### 5.2.2. Objective Function

The final objective in aircraft design is to design for minimum costs [17]. Since accurately estimating costs is difficult, the cost objective is commonly reduced to a fuel burn reduction objective, where the fuel burn is calculated using the Breguet range equation [75].

Due to the nature of the DLM method used for the aerodynamic model, the total drag of the aircraft cannot be computed reliably. This is not a problem, however, since in large-scale aerostructural optimizations, a separate aerodynamics discipline would be added which, using high-fidelity Computational Fluid Dynamics (CFD) methods [5], is able to predict drag more accurately. This means that the structural discipline would only provide the weight to the Breguet range equation, and thus that if high-fidelity aerodynamics are not taken into account, only the weight is relevant for optimization. Therefore, the wing structural weight is used as the objective function in this study.

An interesting note is that the structural model that is optimized is the jig shape, the undeformed shape. In flight, this jig shape is deformed by aerodynamic loads, which introduces coupling between the structural and aerodynamic disciplines. If structural optimization is considered on its own, constraints should be included to ensure that the deformed 1g-flight shape is the same as the shape prescribed from a previous aerodynamic design process. Another approach would be to deduce the jig-shape from the 1g-flight shape, and from there on perform the aeroelastic maneuver and gust load calculations [6]. An inverse static aeroelastic calculation can be performed to deduce the jig shape from the prescribed 1g-flight shape. The jig shape then is an output of the structural analysis procedure. In this study, the influence of the desired 1g-flight shape is not taken into account.

GLA controller tuning is performed using optimal control methods, where the controller design variables are used to minimize a performance index in a sub-optimization problem. The performance index is the  $H_\infty$  norm [114, 116, 121] of the transfer function from gust input to displacement sensor output.

### 5.2.3. Design Variables and Bounds

Structural design variables can be classified as three types: sizing, shape, and topology design variables [27]. Topology design variables represent changes in layout, often of discrete nature. Examples include the number of ribs and stiffeners. Since for a gradient-based optimization algorithm the design variables have to be continuous in nature, topology design variables are not used. Shape design variables influence the analysis model by changing FE nodal positions and thereby influencing the stiffness and mass matrices. Examples include height of spars, chordwise location of spars and spanwise locations of ribs. Changing the shape of wingbox elements effectively changes the Outer Mold Line (OML) [25], which means planform optimization is performed. Such an optimization problem, however, requires the need for modeling of high-fidelity nonlinear aerodynamic effects [139], which are not captured by the DLM method as used by Lagrange. Therefore, in this study, no shape design variables are used.

In the performed optimization studies, the only structural design variables that are used are sizing variables, which influence the model by changing sizes of elements. The sizing variables are defined as the thicknesses  $t$  of the front and rear spar and lower and upper skin defined at ten spanwise wing segments. The thicknesses are allowed to vary between a lower bounds of 4 mm absolute thickness, and an upper bound of twice their initial thickness. This way, a realistic lower bound for manufacturability and a large enough upper bound to give the optimizer enough freedom are provided. Two trimming design variables are defined per static aeroelastic load case: the angle-of-attack  $\alpha$  and the stabilizer deflection  $\delta_e$ . In addition to that, the aileron deflection load case also contains the aileron deflection  $\delta_a$  as a trimming variable.

Depending on whether trimming is handled explicitly or implicitly, these variables are either included in the problem as design variables or as consistency constraints.

Maneuver Load Control (MLC) adds eight design variables representing the quasi-steady aileron and spoiler deflections  $\delta_a$  and  $\delta_s$  for each of the symmetric maneuver load cases. Aileron deflections are budgeted to  $30 \text{ deg}$  deflection up and down, and the spoilers are bounded so that they only deflect up, which corresponds with a negative value for  $\delta_s$ . The spoilers are limited to a deflection of  $20 \text{ deg}$  up, to prevent a too large error due to unmodeled spoiler drag and flow separation effects. Gust Load Alleviation (GLA) adds the FIR coefficients for the feedforward and feedback controllers. Passive Turbulence Alleviation (PTA) adds two design variable for the aileron and spoiler deflections for the static trimmed gust load case.

#### 5.2.4. Design Constraints

Constraints are imposed on the optimization problem to ensure a feasible design. Strength constraints are defined along the wingbox for each load case. Strength constraints are defined as

$$g(\vec{x}) = 1 - \frac{\sigma(\vec{x}) \cdot SF}{\sigma_{yield}} \geq 0, \quad (5.1)$$

where  $\sigma$  is the von-Mises stress,  $SF$  is the safety factor, and  $\sigma_{yield}$  is the yield stress of the wingbox material. A safety factor  $SF$  of 1.5 is applied for nominal load cases (CS-25.203 [33]). For each load case where an active control system is used, a second analysis is performed where the active control system is in failure mode. This analysis is performed with a lower safety factor, depending on the probability of being in failure condition (CS-25, K25.2(c)(2)). It will be assumed that the probability is sufficiently low such that this safety factor can be taken to be 1. A failed system is represented as an analysis without the system present.

Trimming constraints are defined to ensure that lift equals weight and the pitching moment around the CG is zero. These trimming constraints are either added in the optimization problem if explicit trimming is used, or they are used to calculate the new trimming variables if implicit trimming is used.

MLC adds maximum lift constraints for the pull-up and push-over load cases at the low speed corners of the maneuver diagram. The maximum lift curves are calculated from the maximum lift coefficients, a method of estimating maximum section lift distributions from [140], and adjusting the maximum section lift distribution to match the maximum wing lift coefficient.

GLA adds closed-loop stability and robustness constraints if a feedback controller is used. The maximum sensitivity  $M_s$  for the robustness constraints is set to 2, which roughly corresponds to a gain margin of  $6 \text{ dB}$  and a phase margin of  $30 \text{ deg}$  [114]. PTA adds an aileron deflection constraint, which ensures that the static PTA aileron deflection combined with the dynamic GLA deflection does not violate the defined bounds.

Constraint aggregation using the Kreisselmeier-Steinhauser (KS) function is used to aggregate all same-type strength constraints per element. Other constraints, like the MLC maximum lift constraint are aggregated per load case. To speed up calculation of gust response constraint gradients, the selection of critical time steps technique as explained in Section 4.3 is used.

### 5.3. Computational Time Impact Assessment Method

To quantify the impact of using MLC and GLA on computational time, the time to perform one design point evaluation is measured for including MLC in static aeroelastic load cases, and including GLA in gust response load cases. The time measurement is repeated to increase the accuracy of the results. The number of structural design variables used for gradient evaluation is also varied, to assess the relative impact of gradient calculation. If this number is zero, it

means no gradient evaluation is performed. A linear regression line is drawn for each test case, to show the trend of the computational time due to an increasing number of design variables.

To investigate the impact on adding Maneuver Load Control to static aeroelastic analysis, the following three configurations are compared:

1. LagPy static aeroelasticity;
2. LagPy static aeroelasticity with MLC;
3. LagPy static aeroelasticity with MLC and with a  $C_{l_{max}}$  constraint.

For Gust Load Alleviation, a comparison is made between the following three configurations:

1. LagPy gust response analysis, including the `GustResponseCorrection` discipline;
2. XLAGrange gust response analysis, using the `GustResponseAnalysis` discipline;
3. XLAGrange GLA, using the `GustLoadAlleviation` discipline.

In the LagPy case, the `GustResponseCorrection` discipline is added to the optimization problem to prevent unrealistic gust loads.

The time measurements are executed using a smaller aircraft model. The analyses include five static aeroelasticity and five gust load cases. MLC is applied to two of the five static aeroelasticity load cases. Gust response analysis and GLA is applied to all gust load cases. The GLA control system includes a feedback controller, an accelerometer, and a simple actuator. For both MLC and GLA, extra load cases representing the active control system in failure mode are added.

# 6

## Results

This chapter presents and discusses the results. Results for optimizations performed without active control are presented in Section 6.1. Results with MLC and GLA are discussed in Sections 6.2 and 6.3, respectively. GLA controller pre-tuning is discussed in Section 6.4. Results are summarized in Section 6.5. Impact on computational time is presented in Section 6.6. Section 6.7 discusses the results in relation to the research project.

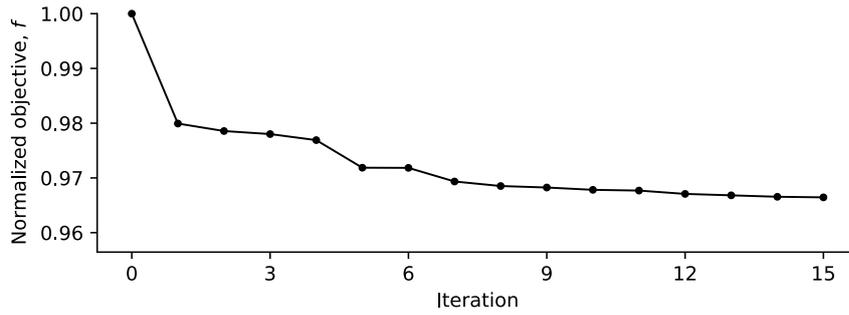
### 6.1. Structural Optimization Without Active Load Control

This section presents and discusses the optimization problem without active control and the impact of using implicit or explicit trimming.

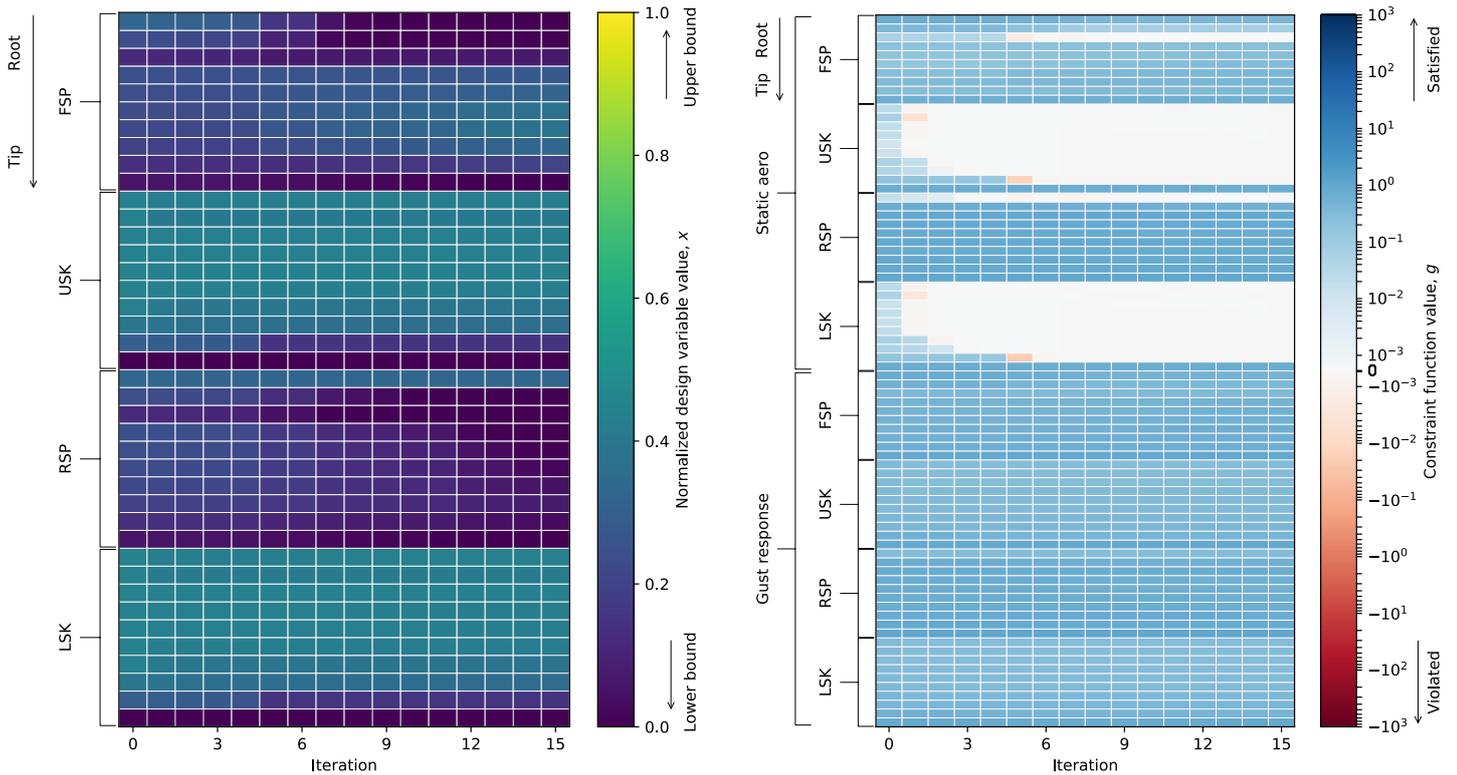
#### 6.1.1. Implicit Aeroelastic Trimming

Optimization progression of the structural optimization problem without active control, and with trimming handled implicitly, is presented in Figure 6.1. Reduction in total wing weight from the initial design is 3.4%. The structural design variables represent the skin and spar thicknesses and are ordered in four sets of ten design variables spanning from inboard to outboard, representing the front spar (FSP), the upper skin (USK), the rear spar (RSP), and the lower skin (LSK). As can be seen in Figure 6.1, the optimizer quickly (within eight iterations) moves towards the final solution by mainly decreasing rear spar thickness and decreasing front spar thickness inboard of the engine, which is located at 32% span.

The constraint value progression is presented in Figure 6.1c. The constraints are ordered in the same manner as the design variables. Similar constraints are aggregated, which results in the distinction between the static aero strength constraints, covering all static aeroelasticity load cases, and the gust response strength constraints, covering the gust response load cases. As can be seen, the static aero constraints are critical for the design, as these values approach zero (white color) near the end of the optimization, indicating that they are close to becoming violated. The optimization is mainly driven by the skin constraints. For the spars, only the front spar just inside of the engine and the rear spar in the root are critically constrained. Concerning the upper and lower skin, the thickness reduction is performed from inboard to outboard. The reason for this could be that a thickness reduction for elements located inboard achieve a larger reduction of the objective function and are therefore considered first by the optimizer. The outboard thicknesses are then only reduced when no more weight gains can be achieved by reducing inboard thicknesses.



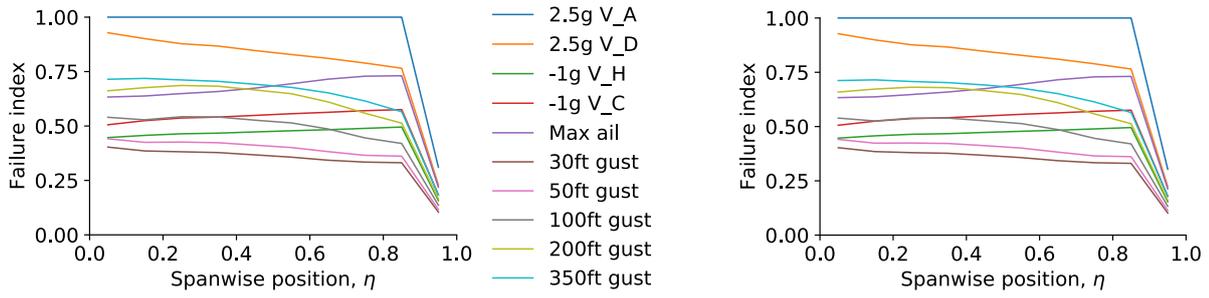
(a) Normalized objective function progression.



(b) Normalized design variable values progression. (c) Constraint value progression.

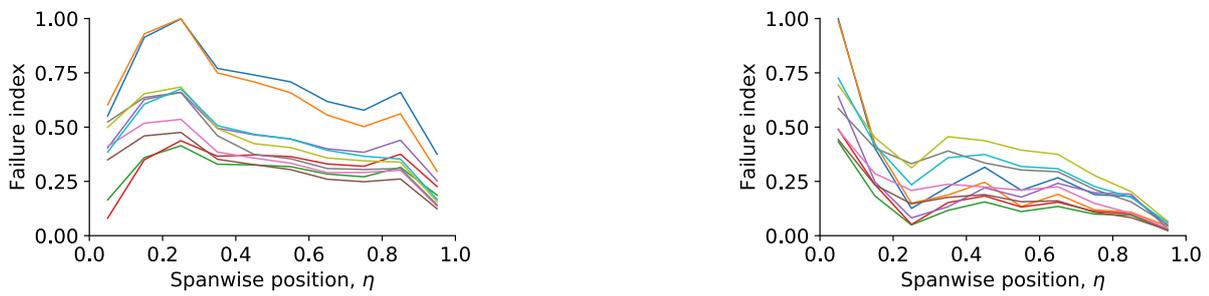
Figure 6.1: Optimization without active control, implicit trimming.

Figure 6.2 presents the load case hierarchy for the optimization without active control. The complete wing is sized by the 2.5g pull-up maneuver at  $V_A$ . Of the gust loads, the most critical gust lengths are the longer gust lengths of 200 ft and 300 ft. This is contrary to the initial assumption made in Section 5.2.1. For the skins and the rear spar, at any spanwise locations where the structure is not critically loaded, the thicknesses are at their lower bound of 4 mm. For the front spar another effect is at play: at the optimal point, the optimizer has found a point at which the wing weight cannot be reduced anymore by varying the skin and/or front spar thickness, without violating the upper skin strength constraint. This effect is shown in Figure 6.3: the line of  $\Delta g_{usk} = 0$  is located in the area of increased weight. In other words, to reduce the front spar thickness until the front spar is critically loaded, an increase in skin thickness is needed that results in a net weight increase. This effect is at play over the whole span at points where the front spar loads are not critical.



(a) Lower skin.

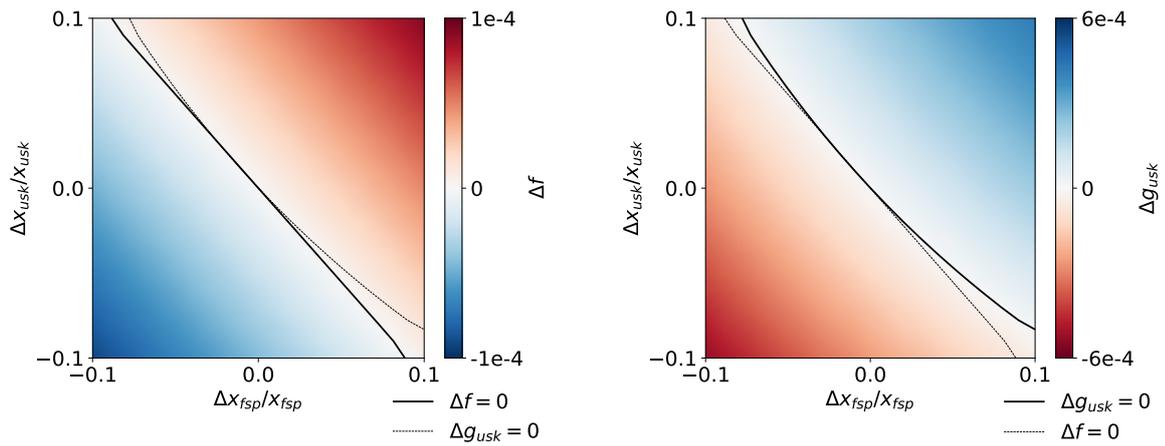
(b) Upper skin.



(c) Front spar.

(d) Rear spar.

Figure 6.2: Load case hierarchy of optimization without active control, implicit trimming.



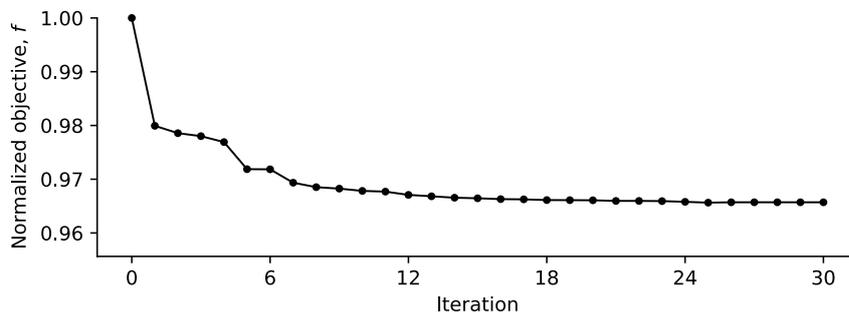
(a) Change in objective function: lower is better.

(b) Change in upper skin strength constraint:  $\Delta g$  must be zero or higher for a feasible design, since  $g_0 = 0$ .

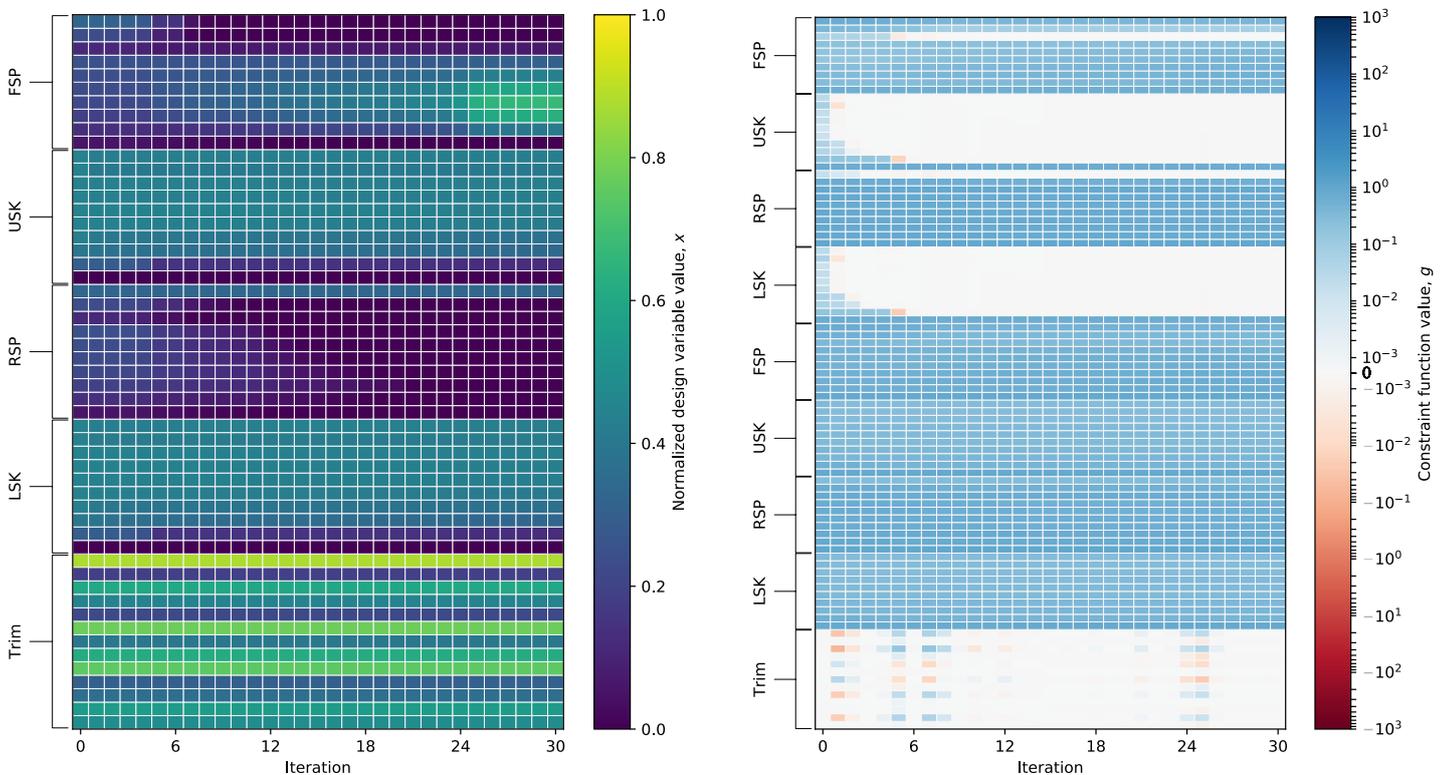
Figure 6.3: Design space exploration at the optimized point of the structural optimization without active load control, for the front spar and upper skin design variables at 15% span.

### 6.1.2. Explicit Aeroelastic Trimming

Figure 6.4 presents the optimization progression for the optimization problem without active control, where trimming is handled explicitly: the trimming design variables and constraints are included in the optimization problem directly. The reduction in wing weight is the same as was achieved using implicit trimming. The objective function, design variable and constraint progression is similar for the first eight iterations, as well as the final load case hierarchy, which has been omitted in this report for brevity. After that, the optimizer is mainly occupied with making sure the trimming constraints are satisfied. The end result is that the optimizer needs more iterations to converge. Needing more iterations is a problem, considering that the computation time mainly comes from the gust response analysis (see Section 6.6), which itself does not depend on whether the aircraft is trimmed or not. Therefore, the rest of the optimizations will all be performed in implicit trimming mode.



(a) Normalized objective function progression.

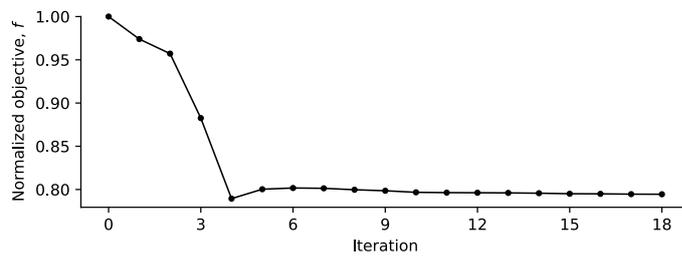


(b) Normalized design variable values progression. (c) Constraint function value progression.

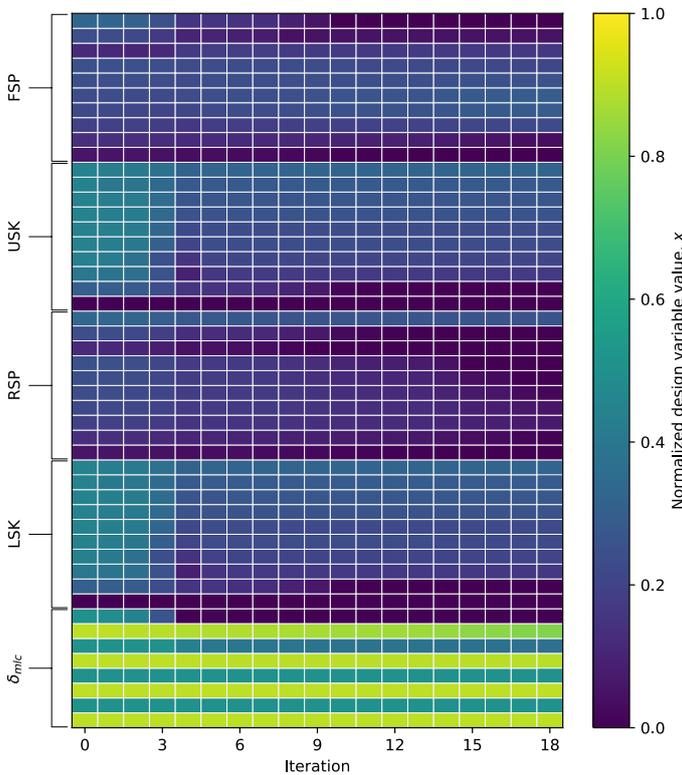
Figure 6.4: Optimization without active control, explicit trimming.

### 6.2. Structural Optimization with Maneuver Load Control

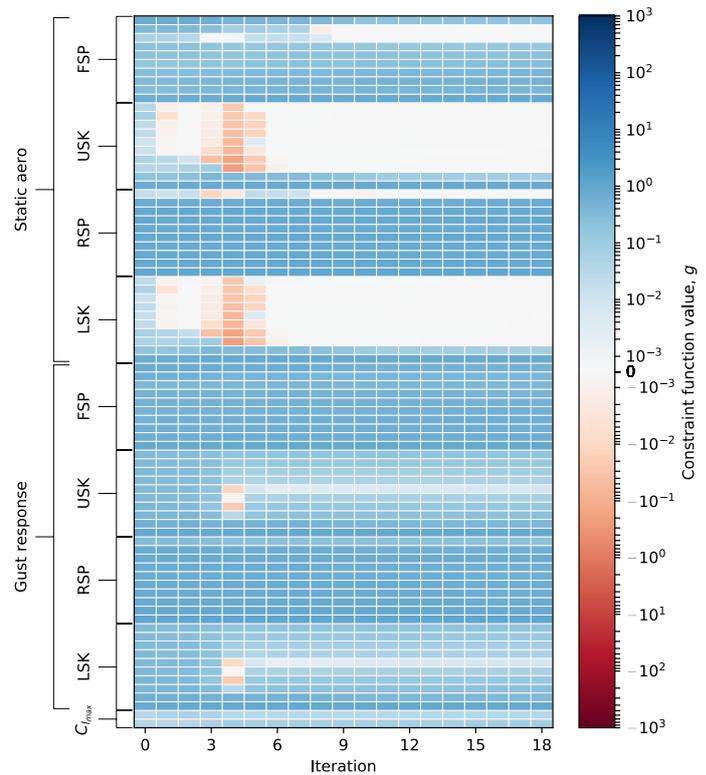
Figure 6.5 presents the optimization progression of the structural optimization problem with MLC enabled. A reduction in wing weight compared to optimization without active load control of 17.6% is achieved. This shows that MLC reduces stresses to enable weight savings. Structural design variables are the same as for the problem without active control, with the addition of the MLC variables, which represent the four aileron and four spoiler deflections at load cases 1 through 4 (see Table 5.1). The spoiler deflections are the design variables which start at and stay near their upper bound, since that corresponds with a zero-degree deflection. Compared to the problem without active control, two extra constraints are added that represent the  $C_{l_{max}}$  constraints at the low-speed  $V_A$  and  $V_H$  maneuver load cases. At iteration four, the optimizer overpredicts the amount of load reduction MLC helps to achieve, resulting in violated strength constraints. It recovers by slightly increasing skin thicknesses.



(a) Normalized objective function progression.



(b) Normalized design variable values progression.



(c) Constraint value progression.

Figure 6.5: Optimization including MLC, implicit trimming.

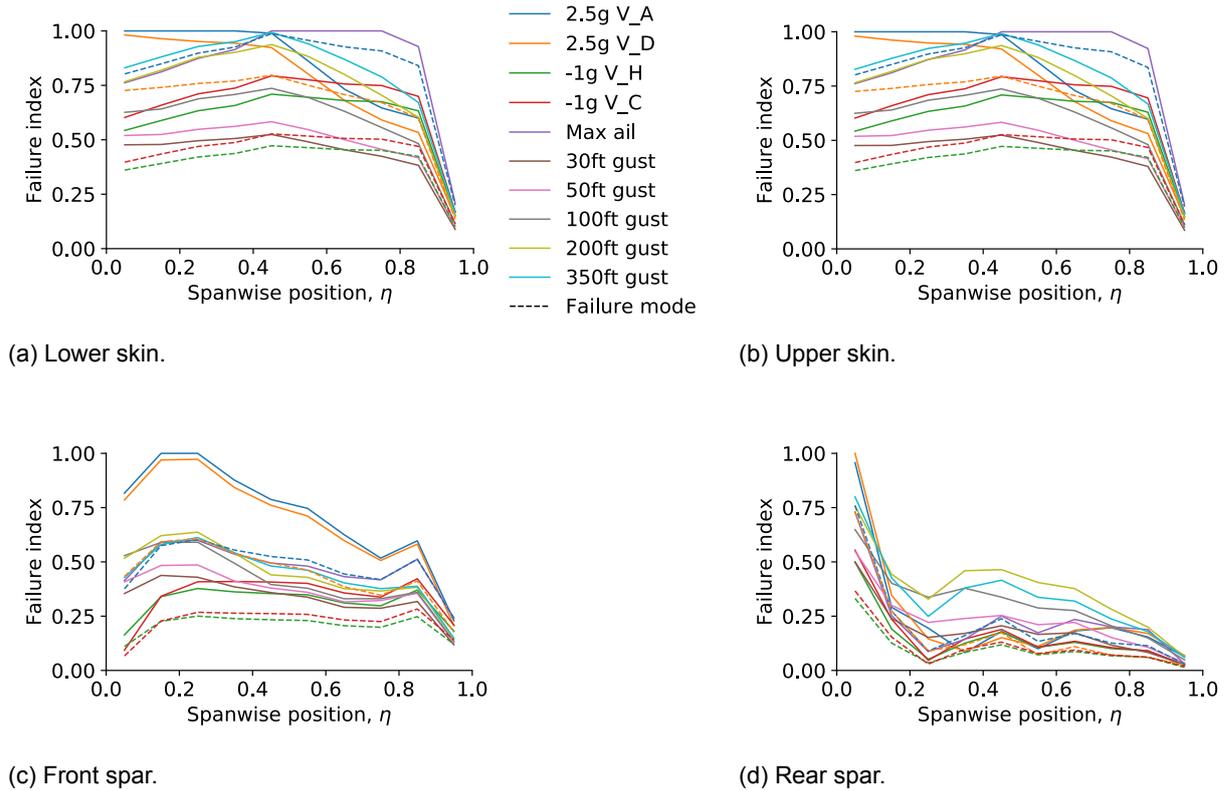


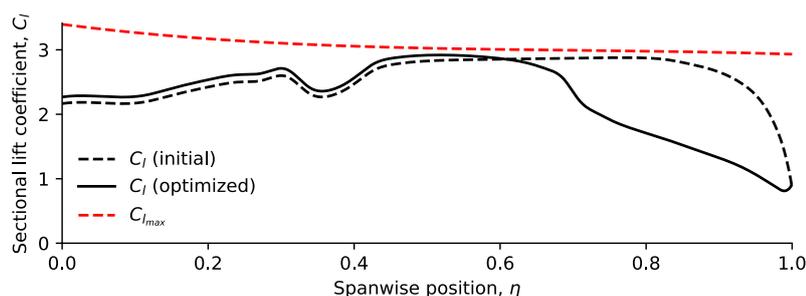
Figure 6.6: Load case hierarchy of optimization with MLC.

MLC variables are only used in the two 2.5g pull-up maneuver load cases (1 and 2). For load case 1, the 2.5g pull-up load case at  $V_A$ , the aileron is fully deflected upwards until its bound, the spoiler is deflection  $2 \text{ deg}$  upwards. For load case 2, the 2.5g pull-up load case at  $V_D$ , the aileron is deflected  $8 \text{ deg}$  upwards, and the spoiler is not deflected. This indicates that the optimizer only uses MLC as much as it is needed to make a load case not design-critical. For the -1g push-over, MLC is not used at all, indicating that the strength constraints in all four areas of the wing box are only critical for 2.5g pull-up. This is reflected in Figure 6.6, where it can be seen that the push-over load cases are not critical. Also, compared to the optimization without active load control, the outer wing is now sized by the aileron deflection load case, since the loads generated by the pull-up load case have been greatly reduced in the outboard wing. The 350  $ft$  load case has become more critical, and almost sizes the wing at 45% span.

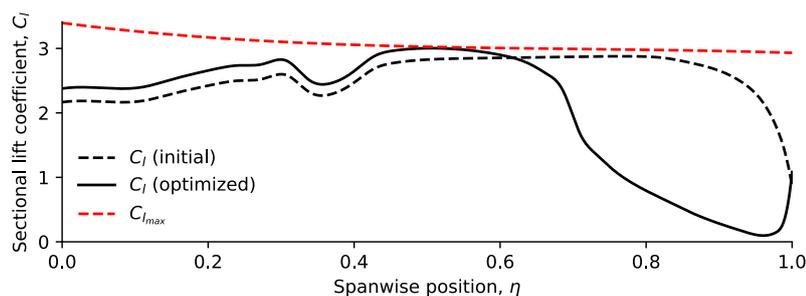
The  $C_{l_{max}}$  constraint for pull-up is not critical, as can be seen from the sectional lift distribution in Figure 6.7a. The deflection of the aileron for MLC greatly reduces the outboard wing lift. The inboard lift is increased to compensate for this. However, not the full loss of lift has to be compensated for, because:

- The weight has been reduced, and thus less lift is required in total;
- Less stabilizer downforce is applied,  $\delta_e^* = -14 \text{ deg}$  while  $\delta_e^{(0)} = -18 \text{ deg}$ , due to a reduced wing pitch down moment, leading to less lift required to be generated by the wing.

Note that the dip in lift coefficient at 35% span comes from the nacelle downwash.



(a) Optimization with initial aileron deflection bounds.



(b) Optimization with relaxed (i.e. increased) aileron deflection bounds.

Figure 6.7: Lift distribution for 2.5g pull-up at  $V_A$  before and after optimization with MLC, with  $C_{l_{max}}$  constraint, for two aileron deflection bounds.

### 6.2.1. Relaxed Aileron Deflection Bounds

When the aileron deflection bounds are increased to  $60 \text{ deg}$  up and down, i.e. made less constraining, the  $C_{l_{max}}$  constraint of the 2.5g pull-up at  $V_A$  load case becomes constraining, as is shown in Figure 6.7b. The inboard lift increase due to MLC is most critical closest to the MLC control surfaces. This result can be expected, since the wing root usually has a very high stall  $\alpha$  for swept-back wings [137]. It is thus shown that the limit to the application of MLC is either determined by the  $C_{l_{max}}$  constraint or the deflection bounds, but that it cannot be determined in advance which one of the two it will be. Note that deflection bounds of  $60 \text{ deg}$  up and down are not realistic in a real aircraft design, since flow separation would occur at these deflections most likely.

### 6.2.2. Influence of Gust Loads

Nominal gust loads, as calculated based on the CS-25 regulations, are not sizing for both the optimization problem without active control and with MLC, as seen in Figure 6.2 and Figure 6.6. This suggests that in order to speed up optimization, the optimization can be executed once without gust loads present, so that only static aeroelasticity load cases remain, and afterwards a check can be done for the gust loads. If gust loads are then critical, the structural optimization has to be performed again with gust loads included. This would speed up development, since gust response analysis is much more computationally costly than static aeroelastic analysis.

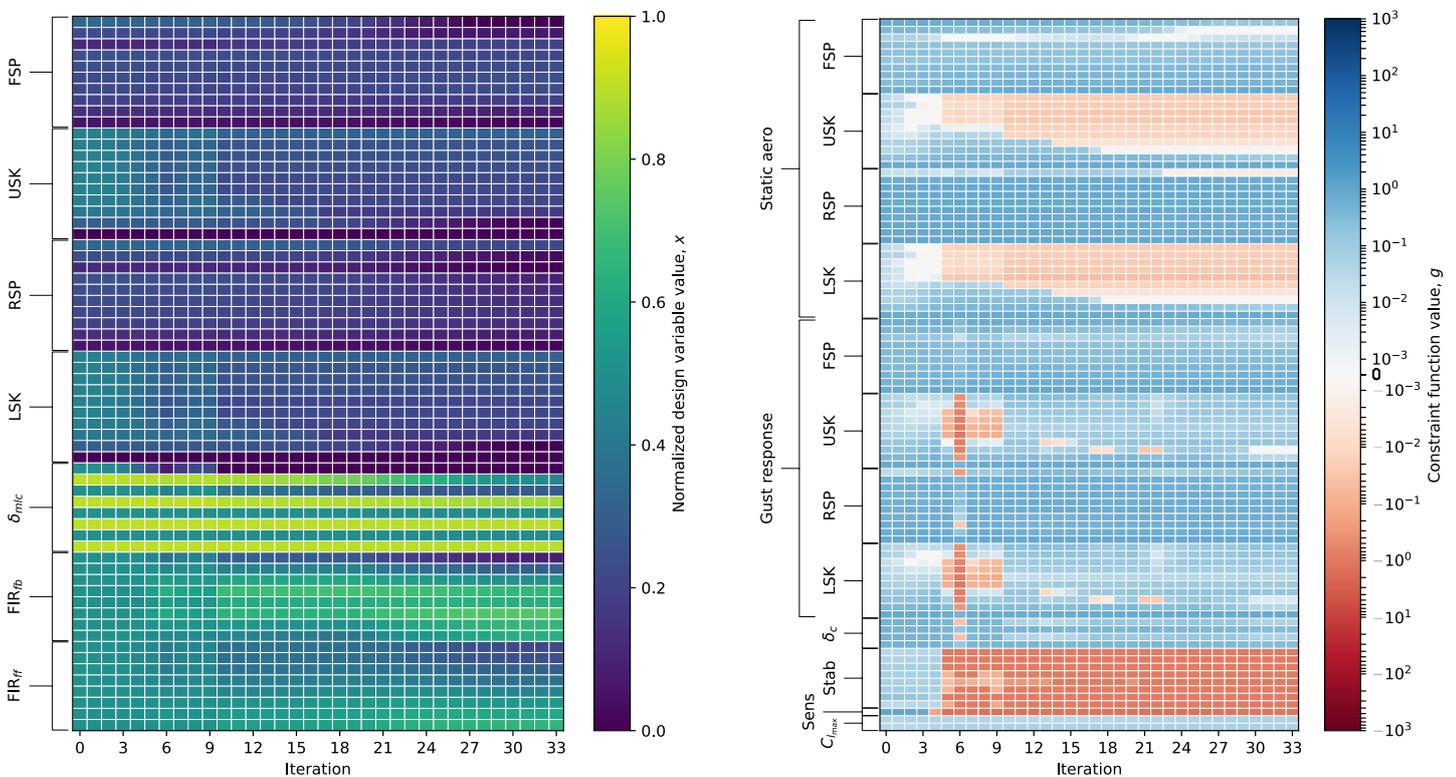
To investigate the impact of Gust Load Alleviation (GLA), however, gust loads should be design-critical. This is why, for the remainder of this study, the discrete gust intensity has been increased to make gust loads design-critical. Optimizing the wing with MLC again, but this time with the critical gust intensity, results in a weight reduction compared to optimization without active load control of only 3.8%, and a wing completely sized by the  $350 \text{ ft}$  discrete gust.

### 6.3. Structural Optimization with Gust Load Alleviation

This section discusses the effects of adding Gust Load Alleviation (GLA), next to MLC, to the structural optimization problem. All optimization problems in this sections include critical gust loads (see Section 6.2.2).

#### 6.3.1. MLC and Feedback + Feedforward GLA

Figure 6.8 shows the progression of optimization performed with MLC and feedback and feedforward GLA. The optimization converges to an infeasible design point, as constraints are violated. The reason for this lies in the inability of the optimizer to move back into the region of feedback stability after changing the feedback FIR coefficients at the fifth iteration. After that, the optimizer actually does come close to stabilizing the system for a few iterations. However, many of the stability constraints remain stuck at  $-1$ . Also, the sensitivity constraint does not change in magnitude when the system moves closer to or further away from stability. This indicates that the stability and sensitivity constraints do not provide the optimizer with enough guidance back to the stable region.



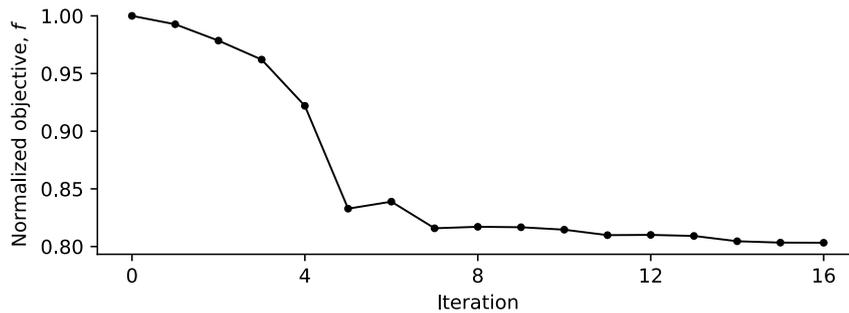
(a) Normalized design variable values progression. (b) Constraint value progression.

Figure 6.8: Optimization with MLC and feedback and feedforward GLA, implicit trimming, with critical gust loads.

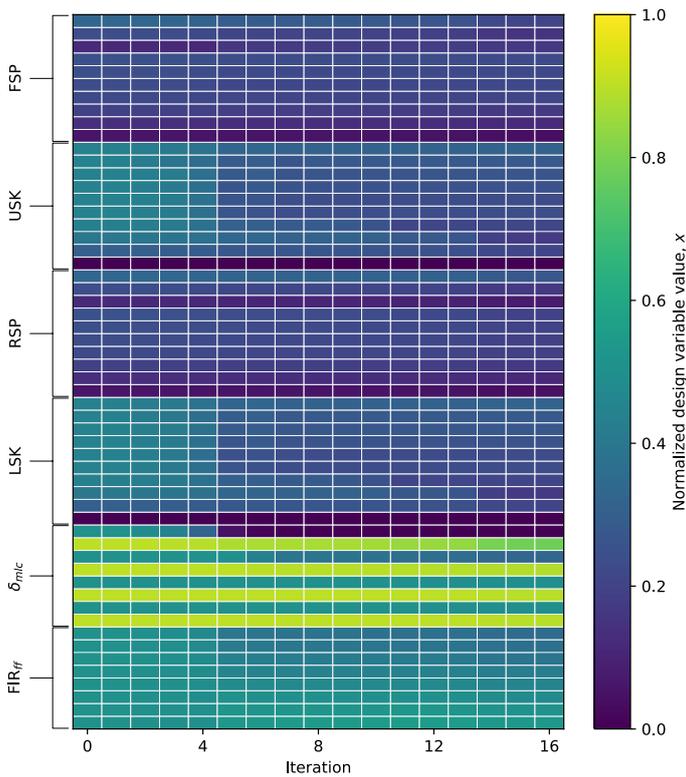
#### 6.3.2. MLC and Feedforward GLA

Considering that using feedback GLA can cause the system to not converge to a feasible design point, only feedforward GLA control is used for the next structural optimization run. The optimization progression is shown in Figure 6.9. A wing weight reduction compared to optimization without active load control of 16.9% is achieved. This reduction is similar to the weight reduction achieved by structural optimization that only uses MLC and has non-critical gust loads (see Section 6.2). In both optimization problems, the optimizer starts by reducing

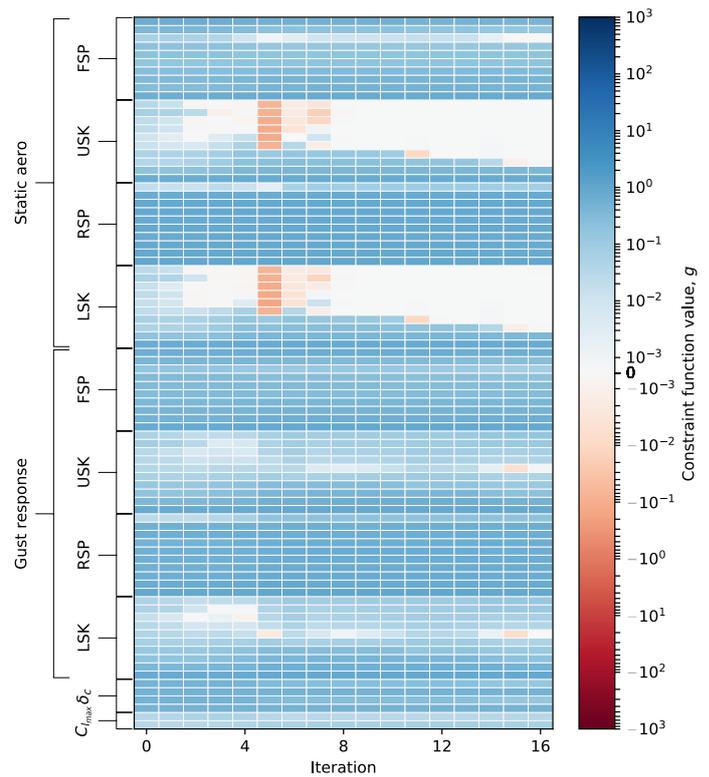
thicknesses to achieve weight reduction, until active load control is needed to further reduce loads. Then, in the optimization problem with both MLC and GLA, the optimizer applies both MLC and GLA at the same time. The impact on load reduction achieved by MLC is slightly overestimated by the optimizer, resulting in the violated strength constraints in the static load cases at the fifth iteration. The end result is that the skins are sized inboard for the  $V_A$  2.5g pull-up load case and outboard for the steady-state roll maneuver, as can be seen in Figure 6.10. This result is the same as when only MLC is used, which means that GLA effectively reduces gust loads until they are not design-critical anymore. Figure 6.11 presents the system response of the optimized structure with and without GLA. It can be seen that without GLA, the strength constraints would indeed have been violated.



(a) Normalized objective function progression.



(b) Normalized design variable values progression.



(c) Constraint value progression.

Figure 6.9: Optimization with MLC and feedforward GLA, implicit trimming, with critical gust loads.

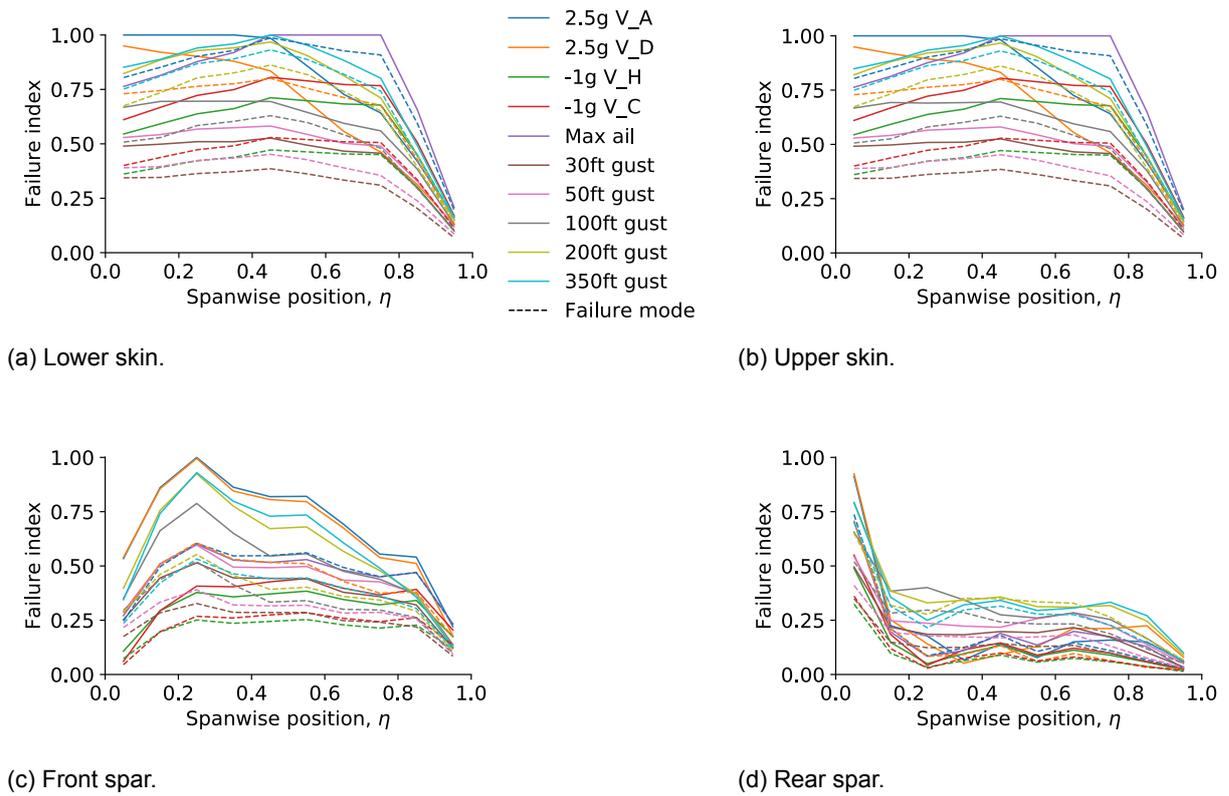


Figure 6.10: Load case hierarchy of optimization with MLC and feedforward GLA, with critical gust loads.

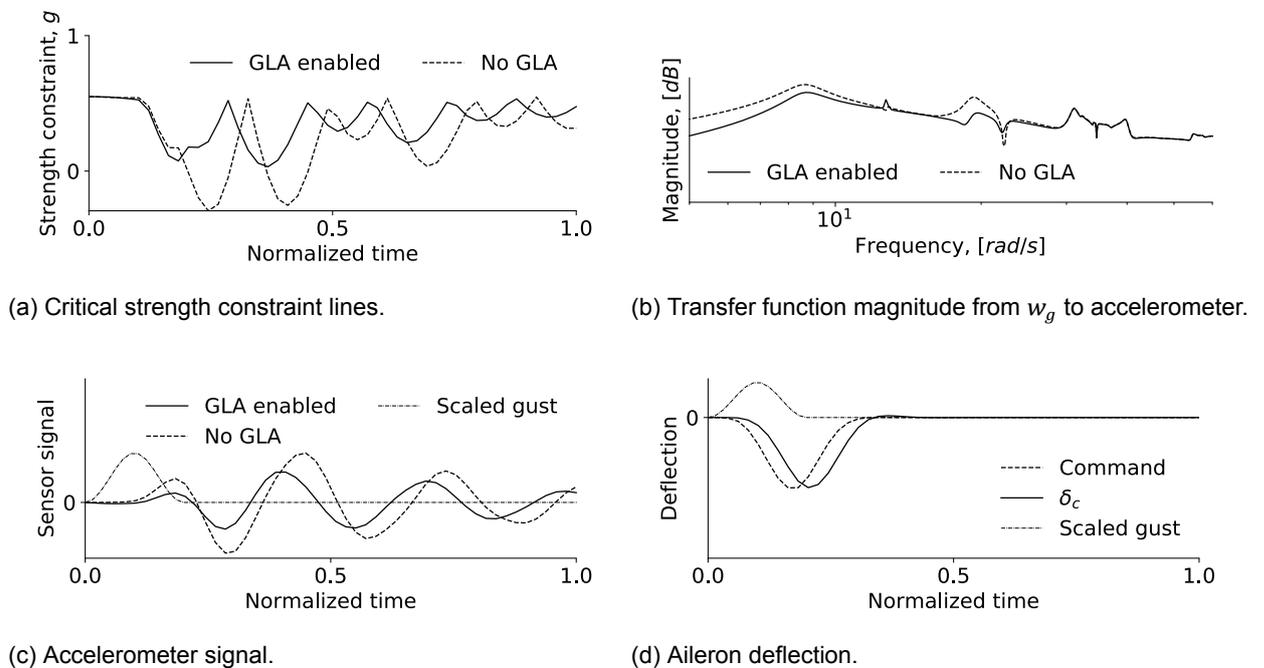
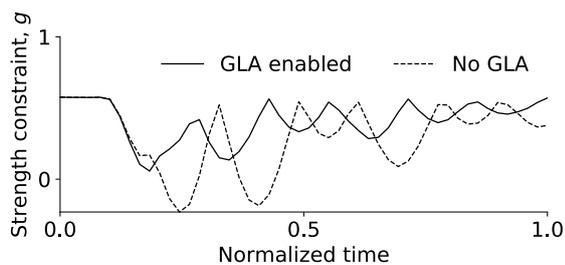


Figure 6.11: Gust response of structure optimized with MLC and feedforward GLA, with critical gust loads. Results are plotted for the 200 ft discrete gust response.

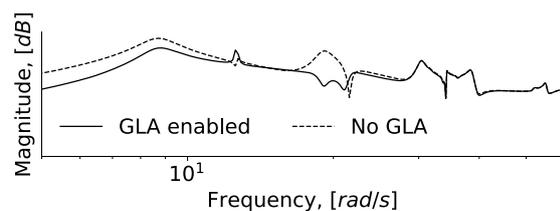
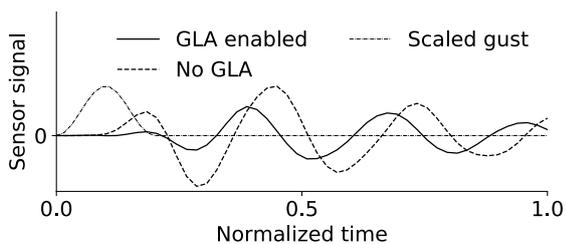
### 6.3.3. MLC, Feedforward GLA and Passive Turbulence Alleviation

Passive Turbulence Alleviation (PTA) is added to the structural optimization with MLC and feedforward GLA. Achieved weight reduction compared to optimization without active load control is 18.1%, and the optimization progression is plotted in Figure 6.13. The PTA design variables are the last two design variables grouped under the  $\delta_{mlc}$  label, as PTA is implemented using the MLC discipline. In the end, only  $2 \text{ deg}$  up aileron deflection is applied as PTA. The maximum aileron deflection constraints are critical, because both PTA and GLA are using the same aileron deflection budget, so including PTA leaves less space for GLA to use the aileron. Comparing Figure 6.12 to Figure 6.11, it can be seen that the gust response of the optimization problem including PTA is similar to the gust response of the structural optimization problem with only MLC and feedforward GLA: the sensor response, strength constraint, and actuator responses all have similar shapes, and the transfer function magnitude is both reduced most for lower frequencies and the peak near  $20 \text{ rad/s}$ .

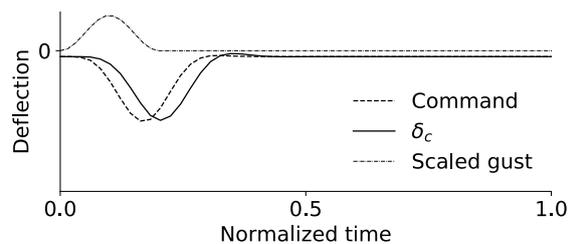
The optimization progression is practically the same as the progression of the optimization problem with MLC and feedforward GLA before iteration 18 (compare Figure 6.13 to Figure 6.9). After iteration 18, the problem without PTA has converged, but the problem with PTA continues its weight reduction. This indicates that PTA is only used when MLC and GLA are at the limits of their load control ability. GLA is thus more effective at gust load control than PTA, but PTA does offer the optimizer with more choice for gust load control in general, thereby increasing weight reduction.



(a) Critical strength constraint lines.

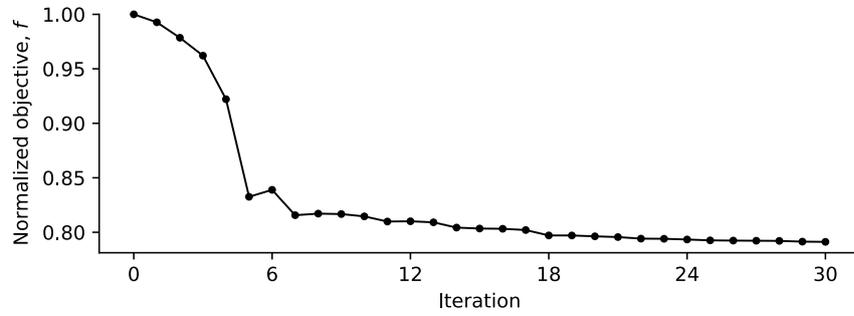
(b) Transfer function magnitude from  $w_g$  to accelerometer.

(c) Accelerometer signal.

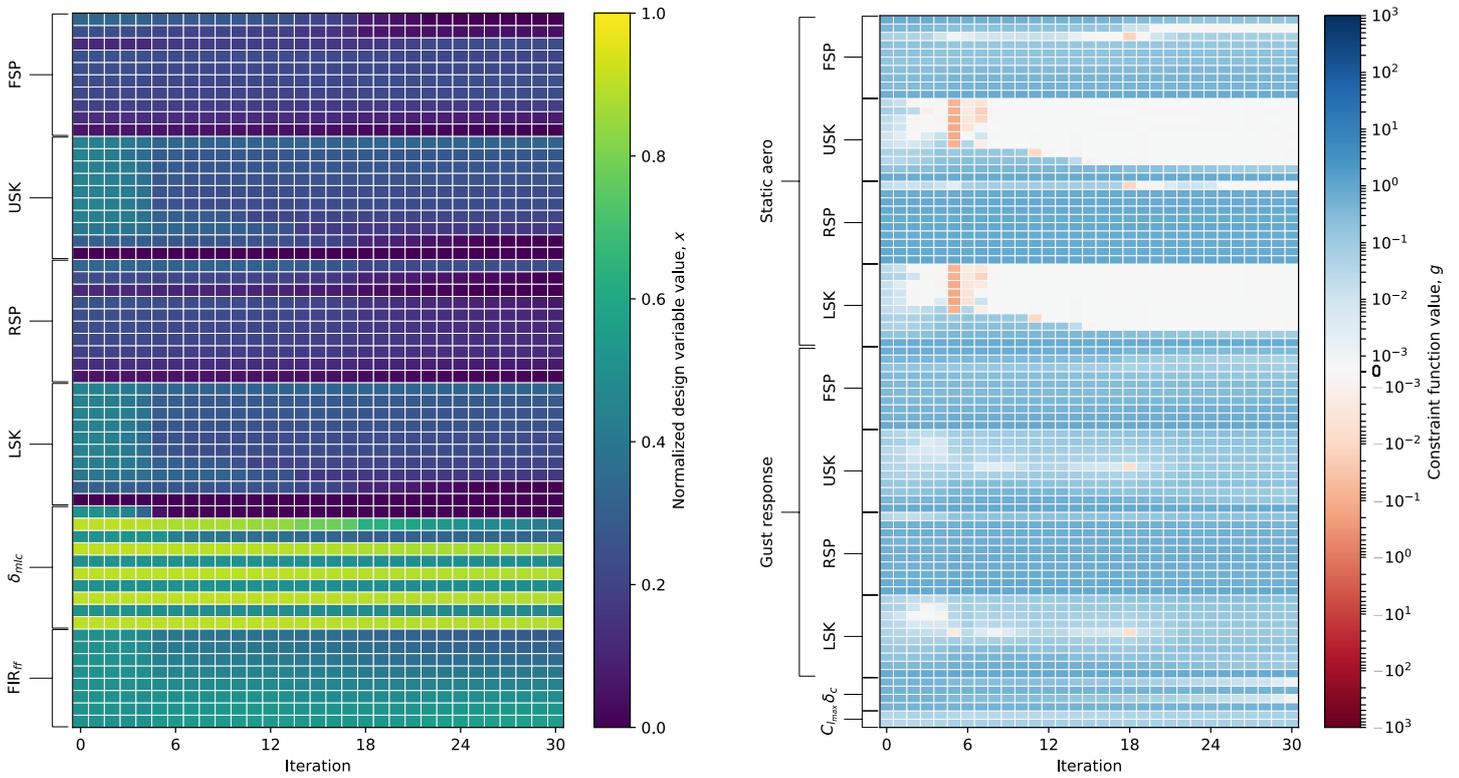


(d) Aileron deflection.

Figure 6.12: Gust response of structure optimized with MLC, feedforward GLA and PTA, with critical gust loads. Results are plotted for the  $200 \text{ ft}$  discrete gust response.



(a) Normalized objective function progression.



(b) Normalized design variable values progression. (c) Constraint value progression.

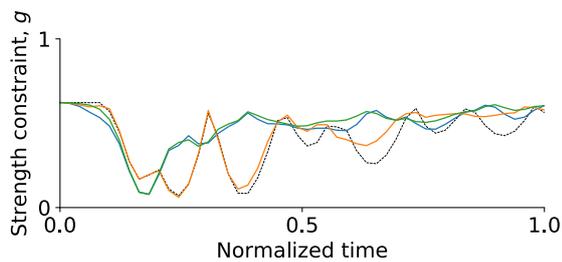
Figure 6.13: Optimization with MLC, feedforward GLA and PTA, implicit trimming, with critical gust loads.

## 6.4. Structural Optimization with a Pre-Tuned GLA Controller

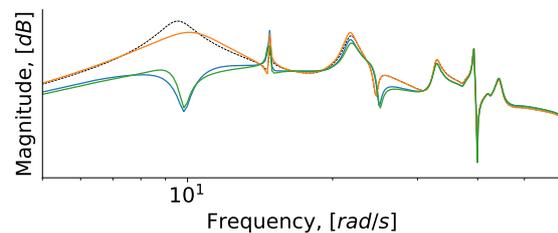
This section discusses the effects of pre-tuning the Gust Load Alleviation (GLA) controller on the structural optimization problem with MLC and GLA. Also in this section all optimization problems include critical gust loads, as explained in Section 6.2.2.

### 6.4.1. Controller Tuning Using Optimal Control

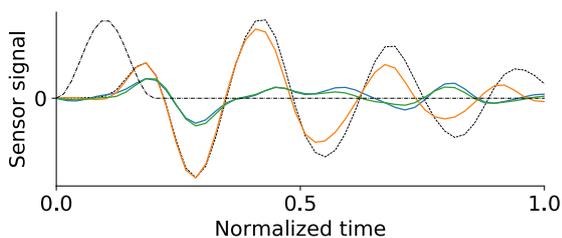
To investigate the impact of pre-tuning the GLA controller using optimal control techniques, the GLA controller is tuned to the  $H_\infty$  performance index at the initial conditions of the structural model. The results are plotted in Figure 6.14, comparing between no GLA, feedback control, feedforward control, and both feedback and feedforward control. Feedforward control moves the main stress peak to an earlier time, and slightly reduces the magnitude of it, as seen in Figure 6.14a. After the main peak, stresses are mostly much lower than when no GLA is used. Feedback control is ineffective at reducing the stress peaks in the first half of the time series, however provides some load alleviation at a later time. The reason for this, is that the feedback controller is not effective at reducing the maximum peak at  $9 \text{ rad/s}$  in the system transfer function, as can be seen in Figure 6.14b. Instead, the main peak is only reduced slightly, until the peak at  $14 \text{ rad/s}$  takes over as the highest peak. The ineffectiveness of feedback control is also seen in the accelerometer and aileron deflection signals in Figures 6.14c and 6.14d, respectively: the accelerometer signal barely changes from the case where no GLA is used, and the aileron indeed is barely actuated near the main load peak. If feedback and feedforward control are used at the same time, the response is dominated by the feedforward controller.



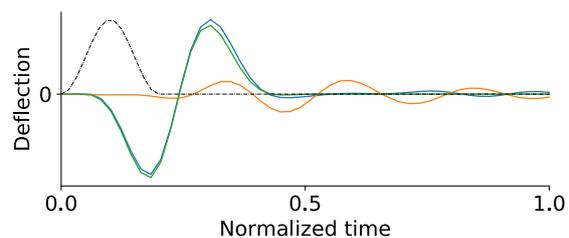
(a) Critical strength constraint lines.



(b) Transfer function magnitude from  $w_g$  to accelerometer.



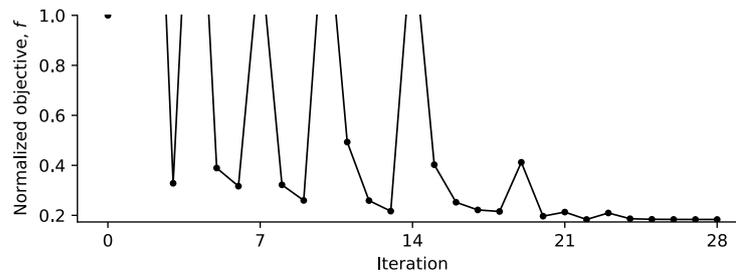
(c) Accelerometer signal.



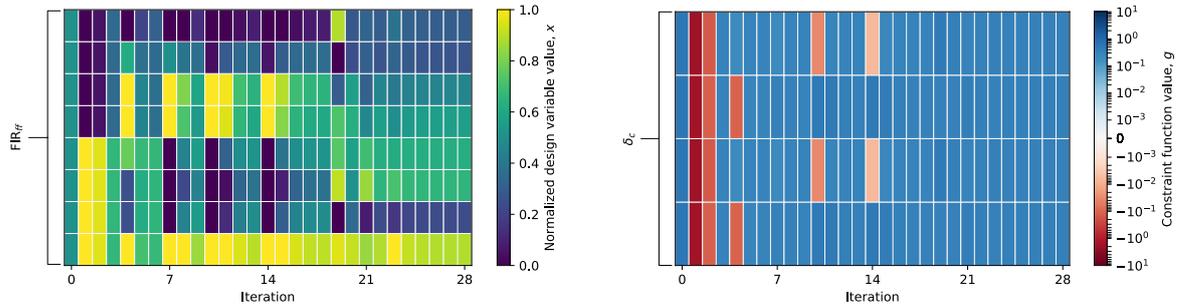
(d) Aileron deflection.



Figure 6.14: Comparison of GLA controller tuning results for various combinations of feedback and feedforward control. Tuning is performed at the initial conditions of the aircraft model. Results are plotted for the  $200 \text{ ft}$  discrete gust response.



(a) Normalized objective function progression.



(b) Normalized design variable values progression.

(c) Constraint value progression.

Figure 6.15: Feedforward controller tuning progression: optimization of  $H_\infty$ -norm of transfer function from  $w_g$  to accelerometer. Tuning is performed at the initial conditions of the aircraft model.

Figure 6.15 presents the progression for the feedforward controller tuning optimization problem. The design variables are the eight FIR coefficients. The only constraints that are included are actuator deflection constraints. As can be seen, a couple of times the optimizer changes the design variables too aggressively, resulting in violated actuator deflection constraints, and an increased objective function value. This indicates that the behavior of the objective function is highly nonlinear and can suddenly change a lot over a small distance. This is in accordance with the behavior of the  $H_\infty$ -norm in Figure 4.8a.

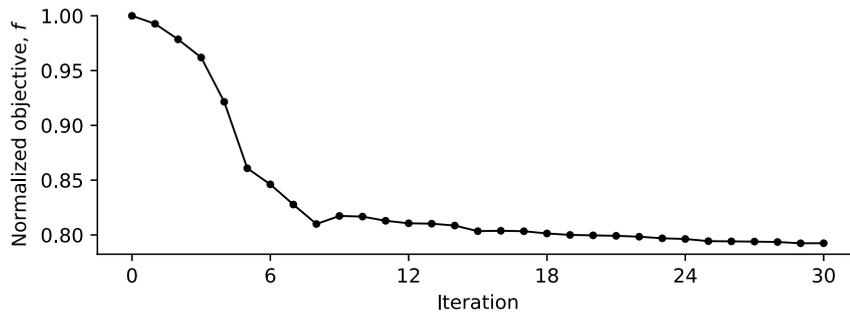
Comparing Figure 6.9b with Figure 6.15b, it can be seen that in controller tuning, the FIR coefficient design variables are moved much more in the direction of their bounds than when the coefficients are tuned concurrently with the structure. This shows that in the structural optimization, the optimizer will only apply as much GLA as is needed for making the gust loads not design-critical. This conclusion is analogous to the earlier conclusion that the optimizer will only apply as much MLC as is needed to make the corresponding maneuver load case not critical.

The concurrently optimized GLA controller achieves more load relief than the tuned controller. This can be explained by comparing the transfer functions (Figure 6.14b and Figure 6.11b): for the tuned controller mainly the peak at  $9 \text{ rad/s}$  is reduced, but for the concurrently optimized controller, the magnitude of the whole transfer function for  $\omega < 11 \text{ rad/s}$  and the peak near  $20 \text{ rad/s}$  are reduced. This indicates that when concurrently optimizing the structure and controller, the optimizer is better able to use GLA for actual load control.

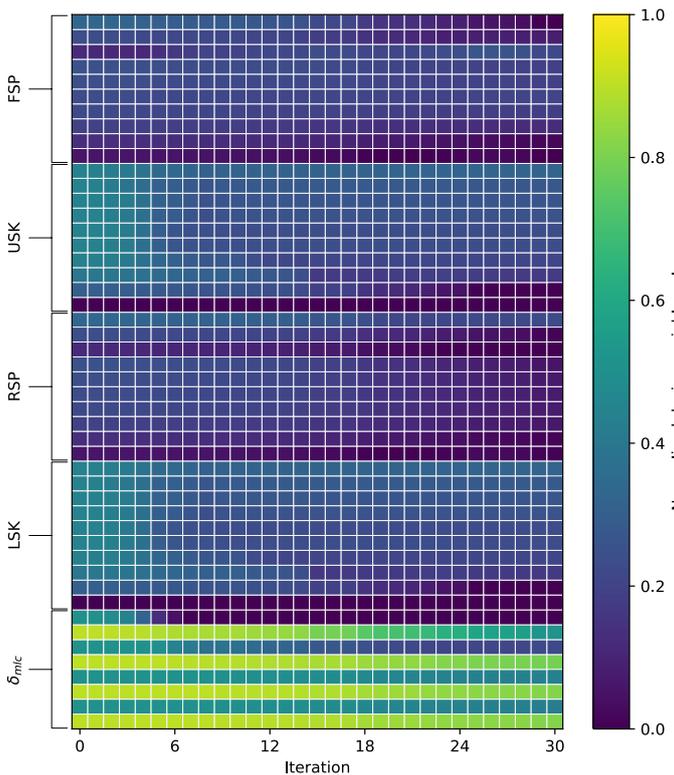
#### 6.4.2. Sequential Optimization of Controller and Structure

The feedforward tuning results presented in Section 6.4.1 are used as a starting point for structural optimization. The controller and structure are optimized sequentially: the structure is optimized after the controller is tuned, and the controller tuning parameters are kept fixed during structural optimization. Feedback control is not included, as it is shown to be ineffective

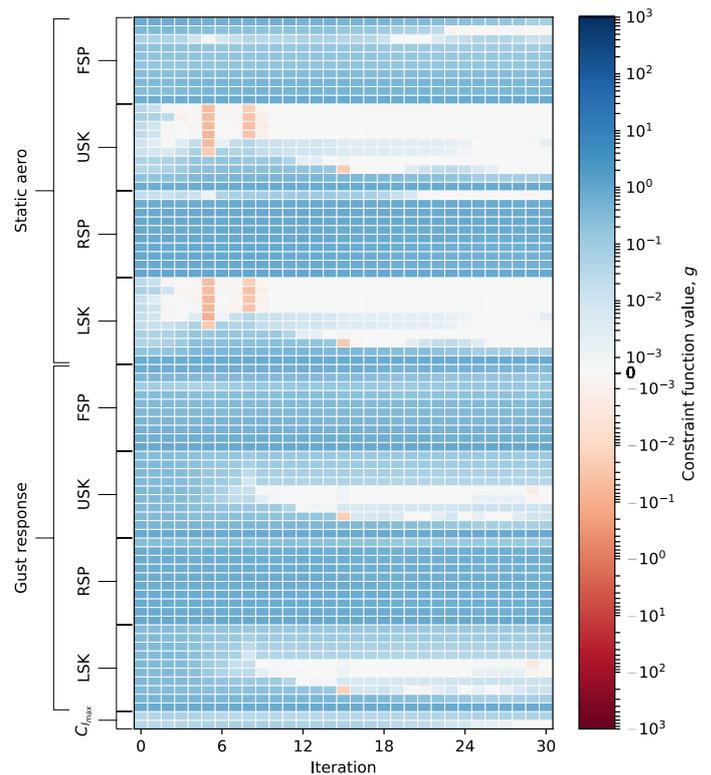
at loads control. A weight reduction of 17.1% is achieved compared to the optimization without active load control, and the progression of the structural optimization is presented in Figure 6.16. As seen in Figure 6.16c and Figure 6.17, the wing is sized by the 350 ft discrete gust between 40% and 60% span: the optimizer is not able to reduce gust loads until they are not sizing anymore. This is because the GLA controller is designed according to the  $H_\infty$  performance index, rather than for load control directly. It is expected that if the critical gust loads intensity is increased further, the concurrently optimized controller will still be able to fully reduce gust loads until they are not critical anymore, while the sequentially optimized structure would then be sized by gust loads for an even larger portion of the span.



(a) Normalized objective function progression.



(b) Normalized design variable values progression.



(c) Constraint value progression.

Figure 6.16: Sequential GLA controller/structure optimization with MLC, implicit trimming, with critical gust loads.

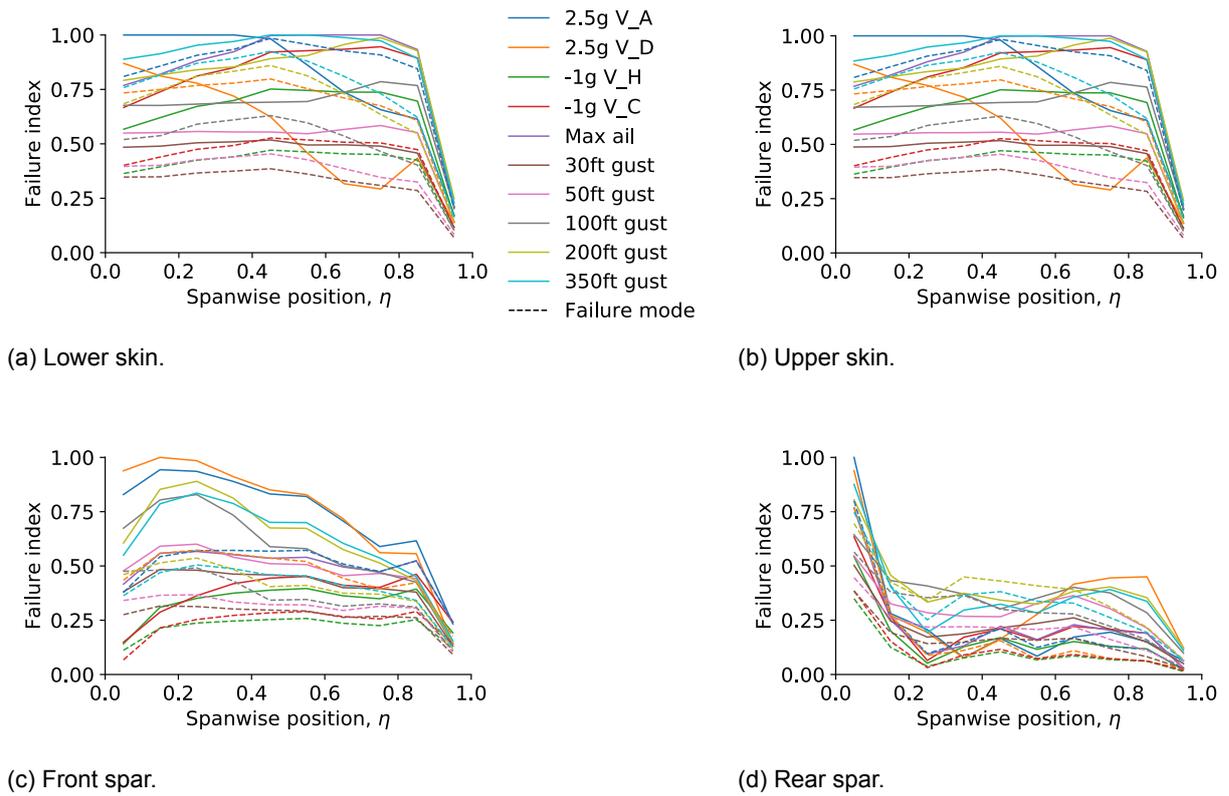


Figure 6.17: Load case hierarchy of sequential GLA controller/structure optimization, with critical gust loads.

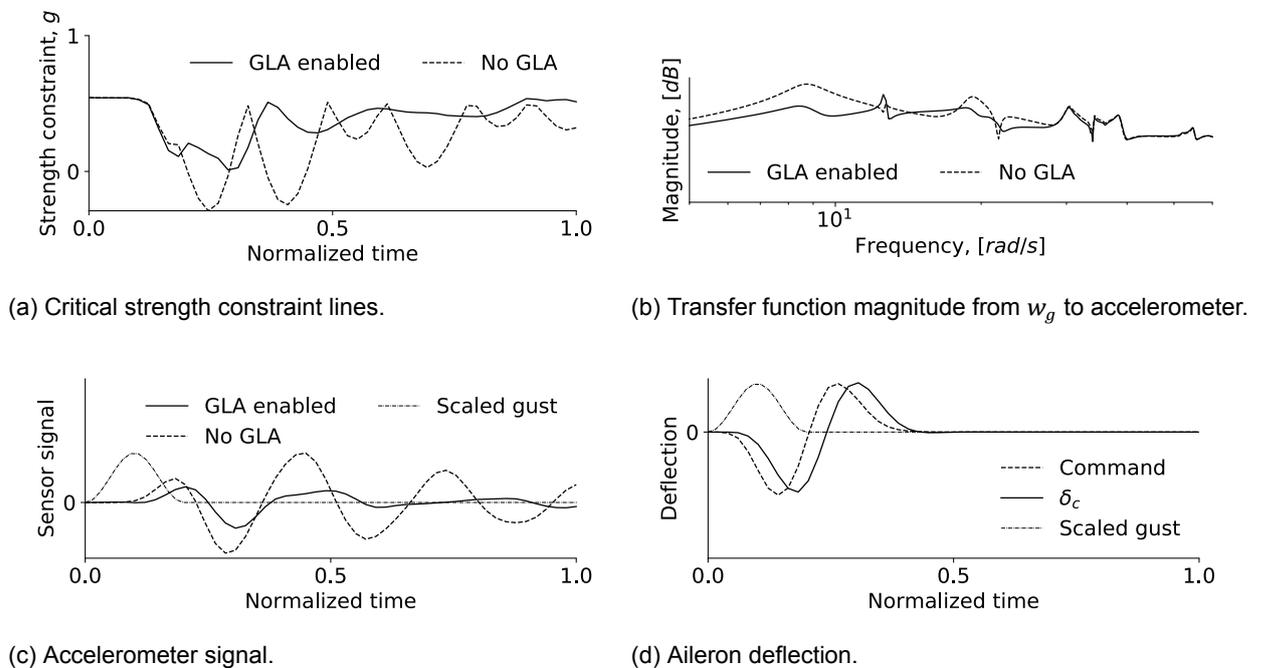


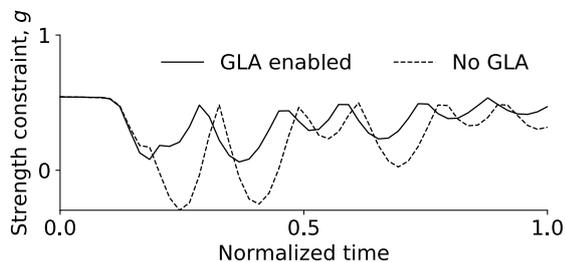
Figure 6.18: Gust response of sequential GLA controller/structure optimization with MLC, with critical gust loads. Results are plotted for the 200 ft discrete gust response.

### 6.4.3. Concurrent Optimization of Pre-Tuned Controller and Structure

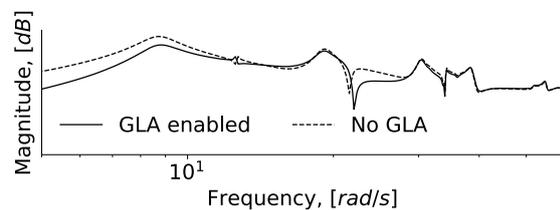
A weight reduction compared to optimization without active load control of 18.0% is achieved when after pre-tuning the feedforward GLA controller, the controller parameters are also included in the structural optimization problem. The optimization progression is presented in Figure 6.20. The weight reduction is similar compared to when the GLA controller is not tuned before the start of the optimization (see Section 6.3.2), but more iterations are needed for convergence. More iterations are needed, because when the controller is not tuned before starting the optimization, both MLC and GLA are applied at the same time at the fifth iteration. However, when the controller is pre-tuned, an adjustment to the GLA parameters is not yet needed at that time: only at the fifteenth iteration the GLA controller parameters are adjusted by the optimizer to facilitate the thickness reduction of the outboard skins.

Contrary to sequentially optimizing the controller and structure, also including the controller parameters in the structural optimization after tuning the controller enables the optimizer to fully reduce gust loads until they are not sizing anymore. The slightly larger weight reduction compared to when the controller is not tuned before concurrent optimization (see Section 6.3.2) shows that pre-tuning the controller provides the optimizer with a better starting point.

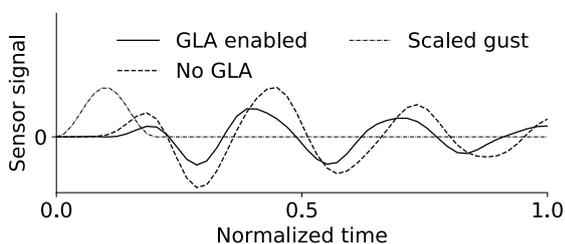
Comparing the concurrently optimized gust response in Figure 6.19 to the sequentially optimized response in Figure 6.18, it can be seen that the concurrently optimized response is more similar to the response of the concurrently optimized structure without a pre-tuned controller (see Figure 6.11). This shows that regardless of the starting point of the controller parameters, concurrently optimizing the controller and structure will result in similar gust response behavior. The concurrently optimized results also show that the strength constraint response is nearly critical at two time-points, instead of only one time-point for the sequential optimization, showing that concurrent optimizer enables a much more detailed control of the gust response and better trade-off capability.



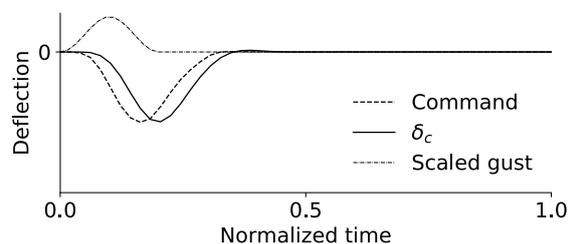
(a) Critical strength constraint lines.



(b) Transfer function magnitude from  $w_g$  to accelerometer.

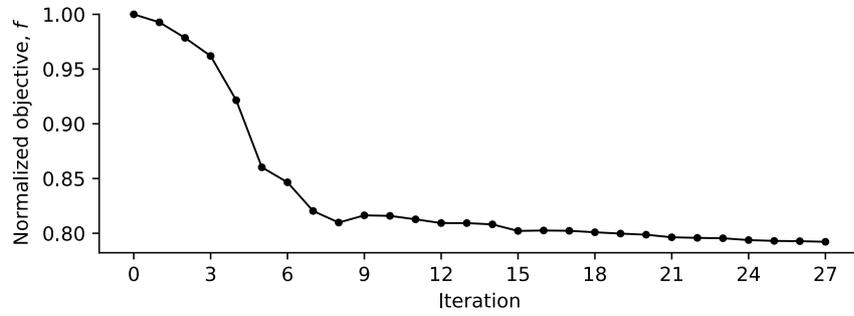


(c) Accelerometer signal.

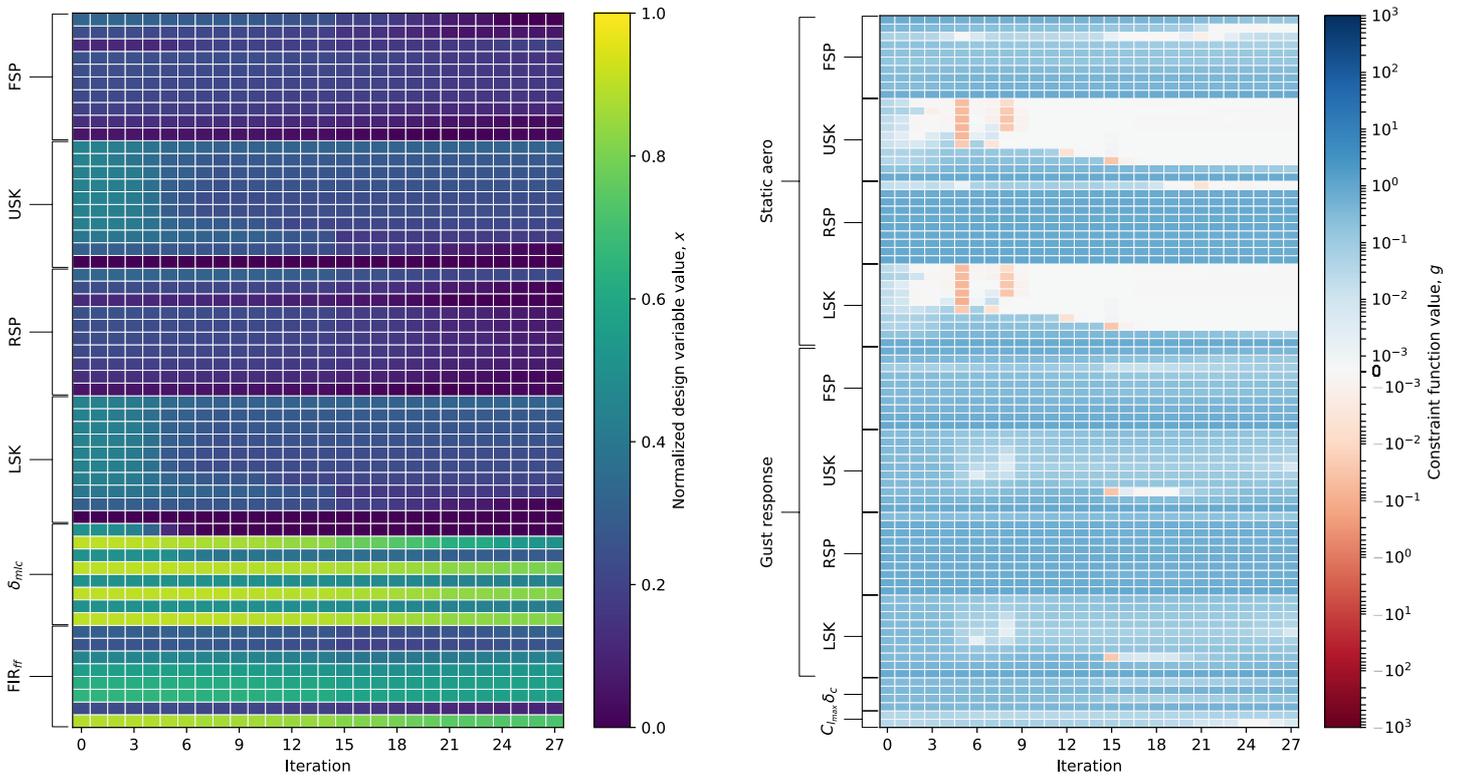


(d) Aileron deflection.

Figure 6.19: Gust response of structure optimized with MLC and pre-tuned feedforward GLA, with critical gust loads. Results are plotted for the 200  $ft$  discrete gust response.



(a) Normalized objective function progression.



(b) Normalized design variable values progression. (c) Constraint value progression.

Figure 6.20: Optimization with MLC and pre-tuned feedforward GLA, implicit trimming, with critical gust loads.

Table 6.1: Structural optimization results overview. Weight reduction is compared to optimization without active load control. FB and FF refer to feedback and feedforward control, respectively.

Active load control			Weight reduction	Gust critical	Notes
MLC	GLA	PTA			
			—		
			—		Explicit trimming
Yes			17.6%		
Yes			3.8%	Yes	
Yes	FB & FF		N/A	Yes	Infeasible design
Yes	FF		16.9%	Yes	
Yes	FF	Yes	18.1%	Yes	
Yes	FF		17.1%	Yes	Sequential controller/structure
Yes	FF		18.0%	Yes	Pre-tuned controller

## 6.5. Results Overview

To provide a clear overview to the reader, the results are summarized in this section. A concise overview of the numerical results obtained from the various structural optimizations with and without active load control is presented in Table 6.1. Using implicit trimming results in less iterations needed for the optimizer to converge than when explicit trimming is used.

Using MLC makes the maneuver load cases less design critical. The optimizer correctly recognizes the use of MLC for load reduction as needed for further weight reduction. The maximum lift constraint only becomes critical at unrealistically large control surface deflections. This shows that maximum lift, as opposed to what was assumed when the decision was made to implement the maximum lift constraint, does not have to be constraining for the application of MLC for large transport aircraft.

Feedback GLA is ineffective at reducing loads when the feedback controller is tuned using optimal control techniques. When feedback GLA control is used in structural optimization, the optimization problem does not converge to a feasible solution due to the violation of feedback stability constraints. When feedforward GLA is used alongside MLC in structural optimization, and gust loads are design-critical initially, the optimizer is able to automatically apply GLA for gust load control, effectively reducing gust loads until they are not design-critical anymore. The result of using GLA, indeed, is that the wing weight is reduced by approximately the same amount and that the same load cases are sizing as when only MLC is used on a wing design with non-sizing gust loads.

Adding PTA to the optimization problem, or pre-tuning the GLA controller results in slightly larger weight reduction, but at the cost of an increased number of iterations needed for convergence. Sequentially optimizing the controller and structure reduces the amount of load control that can be achieved by the structural optimization, and is not able to reduce gust loads until they are not sizing anymore.

## 6.6. Impact on Computational Time

This section presents the results of the study on impact on computational time of adding MLC and GLA to structural optimization.

### 6.6.1. Maneuver Load Control

The results for adding MLC to structural optimization are shown in Figure 6.21. The following observations can be made:

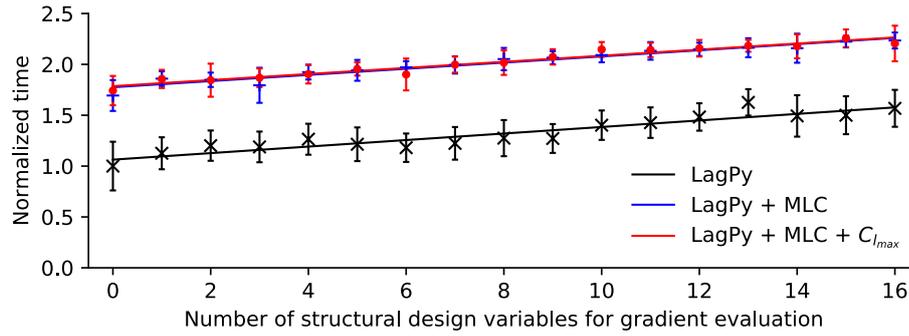


Figure 6.21: Impact of MLC discipline and the  $C_{l_{max}}$  constraint on computational time, with only static aeroelasticity load cases.

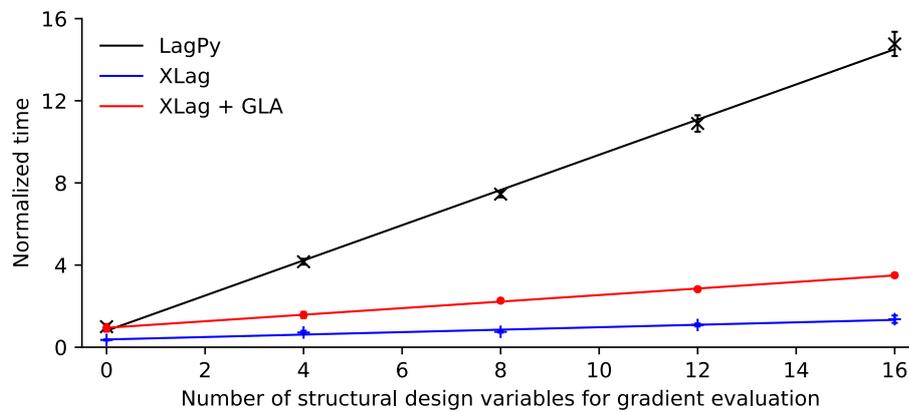


Figure 6.22: Impact of XLAGrange gust response analysis and GLA disciplines on computational time.

1. Gradient evaluation takes a relatively short time per design variable;
2. MLC adds a fixed offset to the evaluation time;
3. The  $C_{l_{max}}$  constraint adds a negligible extra time.

LagPy calculates objective, constraint and static aeroelasticity displacement gradients for all design variables, regardless of which design variables are used for gradient calculation by XLAGrange. The increase in computational time per design variable thus comes from the requesting of mass and stiffness matrix gradients, and the calculation of trimming constraints, which both are performed only when needed.

The fixed increase in computational time when using MLC, comes from the addition of failure load cases, which increases the total number of load cases to analyze, and thus mainly influences the analysis part and not the gradient evaluation part.

Adding MLC to the Lagrange problem increases the computational time by 40% to 75%, but this amount decreases with an increased number of design variables. It should however be noted that an optimization problem with MLC will also include the MLC control surface deflections as design variables, compared to one without MLC, which slightly increases gradient evaluation time as well.

### 6.6.2. Gust Load Alleviation

The results for adding GLA to structural optimization are shown in Figure 6.22. The following observations can be made:

1. XLAGrange gust response analysis takes approximately half the time of LagPy gust response analysis;
2. LagPy gradient evaluation needs much more time than XLAGrange solutions;
3. Using GLA approximately doubles to triples time needed for calculations compared to not using GLA.

The reason that XLAGrange gust response analysis takes less time than LagPy analysis, is because the code is optimized to cache much of the calculated parameters that do not change over time in memory. Examples of such parameters are the gust modes and AIC matrices. For each evaluation, LagPy calculates this data again, leading to a longer calculation time. The reason XLAGrange is able to cache much of this data, is that assumptions are made about which kind of data is not changing, which are not made in the LagPy implementation. For example, if nodal coordinates or aerodynamic mesh would also change for each iteration, the data dependent on this has to be calculated again, reducing or voiding the time gain of XLAGrange over LagPy.

The gradient evaluation takes much less time in XLAGrange than in LagPy, and also increases with less extra time per extra design variable, because XLAGrange has implemented analytical gradients. For LagPy, gradient evaluation requires finite differencing, which means that per extra design variable, one extra gust response analysis has to be performed. This can also be seen in Figure 6.22, where the LagPy line increases approximately with a gradient of one.

When adding GLA, calculation time increases because more values and gradients have to be calculated internally due to the control system, for example transfer functions, aileron deflections, stability constraints, and robustness constraints. Next to that, an optimization problem with GLA will have more design variables than one without GLA, due to the addition of controller parameter design variables. Finally, GLA will also add control system failure load cases, increasing computational time further.

Adding GLA to a LagPy optimization problem greatly reduces computation time, which is mainly due to the availability of analytical gradients. However, compared to a non-GLA gust response analysis that does have analytical gradients, computational time is increased. The amount by which the computational time is increased depends on the GLA control system configuration, like the number of stability constraints, frequency points and time output steps, but can be expected to be between two and three times the time needed for non-GLA gust response analysis.

Note that one gust response analysis performed by XLAGrange takes roughly an order of magnitude longer, depending on many details like the resolution of the frequency grid and the number of gust output time steps, than performing static analysis. This means that any impact on computational time when using both MLC and GLA, will mainly come from the addition of GLA.

## 6.7. Discussion

It is observed that for both MLC and GLA, the optimizer is well able to apply both techniques for load control purposes, in order to reduce wing weight. GLA is able to reduce gust loads until they are not design-critical anymore. Using feedback GLA does not result in a feasible design due to violated feedback stability constraints. However, the main reason for including

feedback control if feedforward is also used, is to guarantee a robust response if the actual system dynamics are different than the system dynamics for which the controllers are designed [46, 114, 141]. In the context of load control, however, robustness might not be relevant. Rather, the prediction of the amount of load control that is possible is important. And because the feedforward controller is much more effective at load control, feedback gust load control might not have to be taken into account in structural optimization.

GLA controller tuning is not needed to let the optimizer apply GLA for load control, which shows that the concurrent optimization of GLA controller parameters and structure can be used to successfully predict whether or not gust loads are indeed sizing or not, and by what amount gust loads can be reduced by a GLA controller. This is relevant, since it shows that optimal control techniques traditionally used for GLA controller tuning, which involve objectives not directly applicable to load relief, are not needed in order to predict the amount of load relief that is achievable during structural optimization. The tuning performance index can be adjusted to make controller tuning more applicable for load control, for example by using frequency weighting or choosing a different performance index, but this requires detailed knowledge of the behavior of the gust response system and makes the application of GLA to structural design much more complicated. Sequential optimization of controller and structure results in an optimized design where gust loads are partly critical, whereas in all concurrently optimized designs gust loads are always reduced until they are not sizing.

Pre-tuning the GLA controller before starting the optimization, but then still concurrently optimizing the controller and structure afterwards, result in a slightly larger weight reduction, showing that pre-tuning might provide a better starting point for the concurrent optimization. Including PTA in the concurrent optimization with MLC and GLA results in slightly more weight reduction than if only MLC and GLA are used, resulting from the increased freedom in gust loads control the optimizer has that is offered by PTA: the PTA design variables are used after the iteration where the problem without PTA would have converged already.

The impact on computational time by including MLC and GLA in the structural optimization process mainly comes from gust response analysis and the inclusion of GLA, since gust response analysis takes roughly an order of magnitude longer than the static aeroelastic analysis where MLC applies to. Due to the implementation of analytical constraint gradients, the evaluation of one design point in a gradient based optimization problem takes much less time compared to the Lagrange implementation of the gust response analysis, which only supports finite differencing. Addition of MLC and GLA to a structural design problem which uses the XLAGrange gust response analysis, however, increases computational time by a factor of two to three.

Referring back to the research question

Can Maneuver Load Control (MLC) and Gust Load Alleviation (GLA) be integrated in the wing structural optimization process of large transport aircraft without causing a significant increase of computational time, and can the optimizer automatically apply them for load control, without the need for optimal control techniques?

it can be concluded that:

1. Computational time is significantly increased when adding MLC and GLA to gust response analysis code which calculates analytical gradients, but is reduced compared to Lagrange gust response analysis which only supports finite differencing for gradient calculation;
2. The optimizer can automatically apply MLC and GLA for load control and thereby providing opportunity for weight reduction, without the need for optimal control techniques.

# Conclusions and Recommendations

This chapter presents the conclusions and recommendations.

## 7.1. Conclusions

Maneuver Load Control (MLC) and Gust Load Alleviation (GLA) have been integrated into the structural optimization process by developing a Python framework called XLAGrange that interfaces with the structural optimizer Lagrange. XLAGrange supports automatically trimming the aircraft using an implicit trimming scheme, which makes sure that for every aeroelastic analysis that is performed, the aircraft is trimmed. Explicit trimming is also supported, where trimming variables and constraints are handled by the system optimizer. Using explicit trimming results in the same optimized design as when implicit trimming is used, but needs more iterations to converge. Implicit trimming therefore offers a more computationally efficient solution.

MLC is implemented by defining new control surfaces as MLC design variables, that are not used for trimming, but rather are freely varied by the optimizer. Maximum lift constraints are implemented by comparing strip lift coefficients to a user-supplied maximum lift curve, to support the fact that maximum lift might be constraining at low speeds. Using MLC in structural optimization results in a larger weight reduction compared to not using active loads control. The optimizer automatically applies MLC when needed, and only applies as much MLC as needed. The application of MLC is limited by either the surface deflection bounds or by the  $C_{l_{max}}$  constraint. With the application of MLC, other load cases, like the aileron deflection and MLC failure mode load cases, become sizing. Tail loads might be reduced, as less elevator downforce is required for trimming the aircraft, due to a reduced pitch down moment at the wing.

GLA is implemented in three steps: the addition of static loads to gust response loads, the re-implementation of gust response analysis in the XLAGrange framework, and the addition of the GLA control system to the gust response analysis. Gust response analysis is implemented in the frequency domain only, and only for discrete gusts. It is verified against gust responses as calculated by Lagrange. Stability and sensitivity constraints are implemented for feedback GLA control, based on data-driven frequency response methods.

Using feedback GLA in structural optimization results in the convergence to an infeasible design, due to violated stability constraints, showing that the implemented stability constraints do not offer useful guidance for the optimizer to keep the feedback system stable. However, feedback control might not have to be taken into account if both feedback and feedforward control are used, since in that case feedback control is only used for robustness, which is not

relevant for structural optimization. Using feedforward GLA in structural optimization together with MLC, for an aircraft design with critical gust loads, results in an optimized design with a similar weight reduction and load case hierarchy as when only MLC is used on an aircraft design with non-critical gust loads. This shows that GLA is effective in reducing the gust loads far enough so that they are not design-critical anymore. The optimizer correctly recognizes the application of GLA for loads reduction, and only applies as much as is needed.

Tuning the GLA controller based on the  $H_\infty$ -norm of the transfer function from gust input to wing tip accelerometer shows that feedforward control reduces loads effectively, while feedback control is much less effective. The  $H_\infty$  performance index behaves highly nonlinear, resulting in aggressive changes in design variables. Tuning the controller based on the  $H_\infty$  performance index before starting structural optimization with free controller parameters provides the structural optimization with a better starting point than if the controller is not tuned before, resulting in a slightly larger weight reduction. Sequential optimization of controller and structure results in less effective load control and the inability of the optimizer to make gust loads non-sizing.

Passive Turbulence Alleviation (PTA), static control surface deflections during gust encounters, is together with GLA also effective in reducing gust loads until these loads are not design-critical anymore. Weight reduction is slightly increased if PTA is used, because the optimizer has more choice in which design variables to use for gust loads reduction: at the iteration where the optimization without PTA converges, the optimization with PTA starts using the PTA design variables for gust load control.

The XLAGrange gust response analysis discipline greatly reduces gradient analysis time compared to LagPy gust analysis due to the availability of analytical gradients, and the selection of critical time steps for gradient calculation. MLC and GLA increase computational time due to the addition of failure mode load cases, design variables, and constraints. The calculation of gust responses takes about an order of magnitude longer than the calculation of static aeroelastic responses, so the impact on computational time comes from the addition of GLA if both GLA and MLC are included in aeroelastic optimization. Compared to XLAGrange gust response analysis, the addition of GLA increases the computational time by approximately a factor of two to three.

In conclusion:

- MLC and GLA do increase computational time, but not compared to Lagrange gust response analysis;
- And, the optimizer can automatically apply MLC and GLA for load relief, without the need for optimal control techniques.

## 7.2. Recommendations

Considering the results of this report, the author recommends the following:

1. Use implicit trimming in aeroelastic optimization problems, where the aircraft is automatically made sure to be trimmed for each analysis, to reduce the number of iterations needed for convergence.
2. Rewrite the analysis code in a more computationally efficient programming language like C or Fortran. The developed framework, XLAGrange, offers a first step in the direction of the integration of active controls in the structural optimization process, but using an interpreted language like Python is not efficient when many calculations are involved. Focus should be placed on enabling the use of large-scale models with tens or hundreds of thousands of degrees of freedom. XLAGrange can however still function as a research

platform, enabling quick prototyping and the convenient swapping out of components due to its OOP-focused implementation.

3. Investigate whether a maximum lift constraint is as constraining as previously assumed, since the control surface deflections might not be large enough to make the wing stall in reality. Especially since large transport aircraft generally have swept-back wings, where the wing root is very stall resistant.
4. Combine active load control with passive tailoring to achieve even more weight reduction and study the interaction between active load control and passive tailoring.
5. Apply the GLA control system for more goals than the reduction of stresses. For example, through the inclusion of a CG z-accelerometer in the performance index objective, it might be possible to not only achieve weight reduction, but also ride comfort control at the same time. This would render the optimization problem a multi-objective optimization problem, but it could very well be that one objective does not affect the other, since the GLA control system is already able to fully reduce gust loads until these have no effect on the structural optimization problem anymore, so any further modification of the gust load controller can then be fully dedicated to the ride comfort control objective.
6. Investigate whether also taking the elevator deflection and tail structure into account in the optimization yields additional benefits. An even more complete Flight Control System (FCS) simulation could be included for this purpose. Assumptions about the interaction between MLC/GLA and the FCS could then also be validated.
7. Increase the efficiency of strength constraint calculations from displacements, through the implementation of a LagPy function that directly calculates these gradients, instead of using the current finite-difference-based scheme, to speed up time needed for computation.
8. Study the application of eigenpair gradient methods other than the currently used exact, but computationally inefficient, Nelson's method. A good starting point is given in [111], where Nelson's method is compared with other methods that are potentially faster. Nelson's method requires repeated factorization of large sparse matrices, which can be a problem especially when scaling-up the number of degrees of freedom of the structural model.
9. Develop data-driven frequency-domain feedback stability constraints that behave more smoothly and are better able to keep the optimizer in the stability region, or guide the optimizer back to it if the system is unstable, to enable the successful convergence of structural optimization where feedback GLA is used.
10. Implement MIMO data-driven feedback sensitivity constraints, possibly based on the  $b_{p,c}$  metric of [123]. The currently implemented method is based on the open-loop transfer function and the Nyquist plot, which does not take MIMO interference effects into account.
11. Investigate whether function weighting for the  $H_\infty$  performance index can be beneficial to the automated tuning process. This might increase user effort required to correctly tune the controller for load control, but it would offer a more precise control over what exactly is tuned.

12. Develop a smoother behaving objective function for controller tuning, that does not cause optimizer algorithm restarts and highly fluctuating objective values over the course of the optimization. For the  $H_\infty$  performance index specifically, value fluctuation might be reduced by taking the base-10 logarithm of the norm value.
13. Use a hybrid global-local optimization approach for controller tuning, for example by using multiple random starting points for gradient-based optimization [113]. This could help improve the convergence behavior of the controller tuning optimization problem and increase the chance of finding the global controller optimum.
14. Investigate whether the omission of the active control system is a correct interpretation of performing analysis with the active control system in failure mode, as specified by CS-25 [33]. For example, an approach based on a failure tree analysis might be used, which enables a more precise identification of worst-case scenarios.
15. Quantify the impact of the requirement of linearity when using frequency-domain analysis on the gust response, so that more data-backed decisions can be made on the way actuator limits are modeled. Other nonlinearities, like spoilers, should also be investigated.
16. Add flutter analysis and Active Flutter Suppression (AFS) to the structural optimization. Interaction between the GLA and AFS control systems should be investigated. Feedback stability theory will have to be partly redeveloped, as currently it is assumed that the aeroelastic plant  $P$  is stable, while for a system experiencing aeroelastic flutter this is not the case.
17. Combine the structural optimization with high-fidelity aerodynamic optimization, to assess the impact of active controls on the complete wing design process, instead of only the structural design.
18. Couple the maximum lift constraint of the MLC discipline to high-fidelity aerodynamic analysis. This would result in a more realistic maximum lift constraint, but depends on the ability of the high-fidelity code to correctly predict the maximum lift point, and the ability of the panel code used for aeroelastic analysis to correctly match the high-fidelity lift.
19. Integrate the XLAGrange OpenMDAO optimization features with collaborative MDO frameworks like AGILE [38], through the use of tools like KADMOS [142] and OpenLEGO [143]. This would enable the XLAGrange framework to be integrated more easily in large-scale aircraft design MDO problems.

# Bibliography

- [1] T. Lefebvre, P. Schmollgruber, C. Blondeau, and G. Carrier. Aircraft Conceptual Design in a Multi-Level , Multi-Disciplinary Optimization Process. *ICAS 2012*, 2012.
- [2] J. Chuprun Jr. Wings. In *The evolution of aircraft wing design; Proceedings of the Symposium*, Reston, Virginia, mar 1980. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1980-3031.
- [3] M. Leitner, R. Liepelt, T.M. Kier, T. Klimmek, R. Müller, and M. Schulze. A Fully Automated Structural Optimization Framework to Determine Critical Design Loads. *Deutscher Luft- und Raumfahrtkongress 2016*, 2016.
- [4] J. Schweiger and J. Krammer. Active Aeroelastic Aircraft and Its Impact on Structure and Flight Control System Design. In *RTA AVT Specialists' Meeting on Structural Aspects of Flexible Aircraft Control*, Ottawa, Canada, 1999. URL <http://www.dtic.mil/dtic/tr/fulltext/u2/p010485.pdf>.
- [5] E. Livne. Active Flutter Suppression: State of the Art and Technology Maturation Needs. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, number January, pages 1–83, Reston, Virginia, jan 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-1119.
- [6] J.R. Wright and J.E. Cooper. *Introduction to Aircraft Aeroelasticity and Loads*. John Wiley & Sons Ltd, Chichester, UK, dec 2014. doi: 10.1002/9781118700440.
- [7] K.G. Bhatia. Airplane Aeroelasticity: Practice and Potential. *Journal of Aircraft*, 40(6): 1010–1018, nov 2003. doi: 10.2514/2.7210.
- [8] L. Bairstow and A. Page. Oscillations of the Tail Plane and Body of an Aeroplane. Technical report, *R&M 276, part 2*, Aeronautical Research Council, Cranfield, UK, 1916.
- [9] D.P. Raymer. *Aircraft Design: A conceptual Approach*. American Institute of Aeronautics and Astronautics, 1992. ISBN: 0930403517.
- [10] H. Hoenlinger, J. Krammer, and M. Stettner. MDO technology needs in aeroelastic structural design. In *7th AIAA/USAF/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization*, pages 206–216, Reston, Virginia, sep 1998. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1998-4731.
- [11] I. Narayanaswamy, S. Narayanan, and V. Chandamouli. Maneuver Load Control for Flexible Delta Wing Using Multidisciplinary Design Optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number 1718, Reston, Virginia, sep 2008. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2008-5971.
- [12] J. Xu and I. Kroo. Aircraft Design with Maneuver and Gust Load Alleviation. In *29th AIAA Applied Aerodynamics Conference*, volume 51, pages 1532–1545, Reston, Virginia, jun 2011. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2011-3180.

- [13] F. Fonte, S. Ricci, and P. Mantegazza. Gust Load Alleviation for a Regional Aircraft Through a Static Output Feedback. *Journal of Aircraft*, 52(5):1559–1574, sep 2015. doi: 10.2514/1.C032995.
- [14] C.D. Regan and C.V. Jutte. Survey of applications of active control technology for gust alleviation and new challenges for lighter weight aircraft. Technical report, *NASA-TM-2012-216008*, 2012.
- [15] Y. Zhao, C. Yue, and H. Hu. Gust Load Alleviation on a Large Transport Airplane. *Journal of Aircraft*, 53(6):1932–1946, nov 2016. doi: 10.2514/1.C033713.
- [16] P.M. Burris, R.B. Holloway, and R.P. Johannes. Aircraft performance benefits from modern control systems technology. *Journal of Aircraft*, 7(6):550–553, nov 1970. doi: 10.2514/3.44211.
- [17] M.H. Shirk, T.J. Hertz, and T.A. Weisshaar. Aeroelastic tailoring - Theory, practice, and promise. *Journal of Aircraft*, 23(1):6–18, 1986. doi: 10.2514/3.45260.
- [18] R.W. Lynch and W.A. Rogers. Aeroelastic Tailoring of Composite Materials to Improve Performance. In *17th Structures, Structural Dynamics, and Materials Conference*, Reston, Virginia, may 1976. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1976-1505.
- [19] T.A. Weisshaar. Aeroelastic tailoring of forward swept composite wings. In *21st Structures, Structural Dynamics, and Materials Conference*, volume 18, pages 669–676, Reston, Virginia, may 1980. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1980-795.
- [20] R. Liebeck. Design of the Blended-Wing-Body subsonic transport. In *40th AIAA Aerospace Sciences Meeting & Exhibit*, number January, pages 1–23, Reston, Virginia, jan 2002. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2002-2.
- [21] E. Torenbeek. *Synthesis of Subsonic Airplane Design*. Delft University Press, 1982. ISBN: 90-247-2724-3.
- [22] E. Obert, R. Slingerland, D.J.W. Leusink, T. van den Berg, and J.H. Koning. *Aerodynamic Design of Transport Aircraft*. IOS Press, 2009. ISBN: 9781586039707.
- [23] J.R.R.A. Martins and A.B. Lambe. Multidisciplinary Design Optimization: A Survey of Architectures. *AIAA Journal*, 51(9):2049–2075, sep 2013. doi: 10.2514/1.J051895.
- [24] J. Sobieszczanski-Sobieski, A. Morris, and M. van Tooren. *Multidisciplinary Design Optimization Supported by Knowledge Based Engineering*. Wiley, 2015. ISBN: 978-1-118-49212-3.
- [25] J.R.R.A. Martins, J.J. Alonso, and J.J. Reuther. High-Fidelity Aerostructural Design Optimization of a Supersonic Business Jet. *Journal of Aircraft*, 41(3):523–530, may 2004. doi: 10.2514/1.11478.
- [26] A.P. Antunes and J.L.F. Azevedo. Studies in Aerodynamic Optimization Based on Genetic Algorithms. *Journal of Aircraft*, 51(3):1002–1012, may 2014. doi: 10.2514/1.C032095.
- [27] E. Livne. Integrated Aeroservoelastic Optimization: Status and Direction. *Journal of Aircraft*, 36(1):122–145, jan 1999. doi: 10.2514/2.2419.

- [28] G. Bindolino, S. Ricci, and P. Mantegazza. Integrated Servostructural Optimization in the Design of Aerospace Systems. *Journal of Aircraft*, 36(1):167–175, jan 1999. doi: 10.2514/2.2423.
- [29] E. Livne, L.A. Schmit, and P. Friedmann. Integrated structure/control/aerodynamic synthesis of actively controlled composite wings. *Journal of Aircraft*, 30(3):387–394, may 1993. doi: 10.2514/3.46347.
- [30] E. Livne. Future of Airplane Aeroelasticity. *Journal of Aircraft*, 40(6):1066–1092, nov 2003. doi: 10.2514/2.7218.
- [31] Sohrab Haghghat, Joaquim R. R. A. Martins, and Hugh H. T. Liu. Aeroservoelastic Design Optimization of a Flexible Wing. *Journal of Aircraft*, 49(2):432–443, mar 2012. doi: 10.2514/1.C031344.
- [32] P.S. Zink, D.E. Raveh, and D.N. Mavris. Integrated Trim and Structural Design Process for Active Aeroelastic Wing Technology. *Journal of Aircraft*, 40(3):523–531, may 2003. doi: 10.2514/2.3126.
- [33] EASA. *CS-25 Certification Specifications for Large Aeroplanes*. 2007.
- [34] R. White. Improving the airplane efficiency by use of wing maneuver load alleviation. In *Meeting on the Prospects for Improvement in Efficiency of Flight*, volume 8, pages 769–775, Reston, Virginia, jul 1970. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1970-877.
- [35] J. Wykes. Structural dynamic stability augmentation and gust alleviation of flexible aircraft. In *5th Annual Meeting and Technical Display*, number 68, Reston, Virginia, oct 1968. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1968-1067.
- [36] M. Guinot. Innovative Systems and Aircraft Design. *FAST A380 Special Edition*, (November), 2016. URL <http://www.aircraft.airbus.com/support-services/publications/>.
- [37] N. Fezans. An Unusual Structure for a Feedforward Gust Load Alleviation Controller. In *Advances in Aerospace Guidance, Navigation and Control*, pages 47–68. Springer International Publishing, Cham, 2018. doi: 10.1007/978-3-319-65283-2\_3.
- [38] P.D. Ciampa and B. Nagel. The AGILE Paradigm: the next generation of collaborative MDO. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia, jun 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-4137.
- [39] S.A. Ning and I. Kroo. Multidisciplinary Considerations in the Design of Wings and Wing Tip Devices. *Journal of Aircraft*, 47(2):534–543, 2010. doi: 10.2514/1.41833.
- [40] C. Nam, A. Chattopadhyay, and Y. Kim. Optimal wing planform design for aeroelastic control. *AIAA Journal*, 38(8):1465–1470, jan 2000. doi: 10.2514/3.14569.
- [41] M. Pusch, A. Knoblach, and T. Kier. Integrated Optimization of Ailerons for Active Gust Load Alleviation. In *IFASD 2015: 16th International Forum on Aeroelasticity and Structural Dynamics*, 2015.

- [42] J. Zeng, B. Moulin, R. De Callafon, and M. Brenner. Adaptive Feedforward Control for Gust Load Alleviation. *Journal of Guidance, Control, and Dynamics*, 33(3):862–872, may 2010. doi: 10.2514/1.46091.
- [43] A. Wildschek. *An Adaptive Feed-Forward Controller for Active Wing Bending Vibration Alleviation on Large Transport Aircraft*. PhD thesis, Technische Universität München, 2009.
- [44] W. Fan, H.H. Liu, and R. Kwong. Gust Load Alleviation Control for a Flexible Aircraft with Loss of Control Effectiveness. *AIAA Guidance, Navigation, and Control Conference*, (January):1–16, 2017. doi: 10.2514/6.2017-1721.
- [45] A. Wildschek, B. Prananta, T. Kanakis, H. van Tongeren, and R. Huls. Concurrent Optimization of a Feed-Forward Gust Loads Controller and Minimization of Wing Box Structural Mass on an Aircraft with Active Winglets. In *16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number June, pages 1–24, Reston, Virginia, jun 2015. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2015-2490.
- [46] N. Fezans and H. Joos. Combined Feedback and LIDAR-Based Feedforward Active Load Alleviation. In *AIAA Atmospheric Flight Mechanics Conference*, Reston, Virginia, jun 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-3548.
- [47] S. Suzuki and S. Yonezawa. Simultaneous structure/control design optimization of a wing structure with a gust load alleviation system. *Journal of Aircraft*, 30(2):268–274, mar 1993. doi: 10.2514/3.48276.
- [48] M. Karpel, B. Moulin, and P.C. Chen. Dynamic Response of Aeroservoelastic Systems to Gust Excitation. *Journal of Aircraft*, 42(5):1264–1272, 2005. doi: 10.2514/1.6678.
- [49] R. Thormann and M. Widhalm. Linear-Frequency-Domain Predictions of Dynamic-Response Data for Viscous Transonic Flows. *AIAA Journal*, 51(11):2540–2557, nov 2013. doi: 10.2514/1.J051896.
- [50] P. Bekemeyer, R. Thormann, and S. Timme. Frequency-Domain Gust Response Simulation Using Computational Fluid Dynamics. *AIAA Journal*, 55(7):2174–2185, jul 2017. doi: 10.2514/1.J055373.
- [51] J. Schweiger, J. Krammer, and H. Hoernlein. Development and application of the integrated structural design tool LAGRANGE. In *6th Symposium on Multidisciplinary Analysis and Optimization*, pages 1616–1624, Reston, Virginia, sep 1996. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1996-4169.
- [52] B.K. Stanford. Optimization of an Aeroservoelastic Wing with Distributed Multiple Control Surfaces. *Journal of Aircraft*, 53(4):1131–1144, jul 2016. doi: 10.2514/1.C033613.
- [53] I.R. Chittick and J.R.R.A. Martins. An asymmetric suboptimization approach to aerostructural optimization. *Optimization and Engineering*, 10(1):133–152, mar 2009. doi: 10.1007/s11081-008-9046-2.
- [54] D.W. Zingg, M. Nemec, and T.H. Pulliam. A comparative evaluation of genetic and gradient-based algorithms applied to aerodynamic optimization. *Revue européenne*

- de mécanique numérique*, 17(1-2):103–126, mar 2008. doi: 10.3166/remn.17.103-126.
- [55] MSC Nastran 2017 - Quick Reference Guide, 2017.
- [56] J. Nocedal and S. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2006. doi: 10.1007/978-0-387-40065-5.
- [57] A. Rothwell. *Optimization Methods in Structural Design*, volume 242 of *Solid Mechanics and Its Applications*. Springer International Publishing, Cham, 2017. doi: 10.1007/978-3-319-55197-5.
- [58] D. Nussbächer, M. Hanel, F. Daoud, and M. Hornung. Multidisciplinary Design Optimization of Flight Control System Parameters in Consideration of Aeroelasticity. *CEAS 2015*, 2015. URL [http://aerospace-europe.eu/media/books/CEAS2015\\_095.pdf](http://aerospace-europe.eu/media/books/CEAS2015_095.pdf).
- [59] Python 2.7 Documentation. URL <https://docs.python.org/2.7/>.
- [60] F. Daoud, R. Maierl, P.D. Ciampa, and X. Gu. Aeroelastic Shape and Sizing Optimization of Aircraft Products supported by AGILE Design Paradigm. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number June, pages 1–13, Reston, Virginia, jun 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-4141.
- [61] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design Patterns - Elements of Reusable Object-Oriented Software*. Pearson Education, oct 1995. ISBN: 0-201-63361-2.
- [62] M. Fowler. *UML Distilled*. Addison-Wesley, third edit edition, 2004. ISBN: 0-321-19368-7.
- [63] N. Singh, L. Browne, and R. Butler. Parallel astronomical data processing with Python: Recipes for multicore machines. *Astronomy and Computing*, 2:1–10, aug 2013. doi: 10.1016/j.ascom.2013.04.002.
- [64] S. Doyle. pyNastran - A Python-based Interface Tool for Nastran’s File Formats, 2017. URL <https://github.com/SteveDoyle2/pyNastran>.
- [65] K. Moore, B. Naylor, and J. Gray. The Development of an Open Source Framework for Multidisciplinary Analysis and Optimization. In *12th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, number September, pages 1–13, Reston, Virginia, sep 2008. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2008-6069.
- [66] R. Coroneos and S. Pai. Deterministic Design Optimization of Structures in OpenMDAO Framework. In *12th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference and 14th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia, sep 2012. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2012-5482.
- [67] J.T. Hwang. Reconfigurable model execution in the OpenMDAO framework. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia,

- jun 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-3821.
- [68] J.S. Gray, T.A. Hearn, K.T. Moore, J. Hwang, J. Martins, and A. Ning. Automatic Evaluation of Multidisciplinary Derivatives Using a Graph-Based Problem Formulation in OpenMDAO. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia, jun 2014. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2014-2042.
- [69] G. Kenway, G. Kennedy, and J. Martins. Aerostructural optimization of the Common Research Model configuration. In *15th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 1–19, Reston, Virginia, jun 2014. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2014-3274.
- [70] J. Sobieszczanski-Sobieski. A technique for locating function roots and for satisfying equality constraints in optimization. *Structural Optimization*, 4(3-4):241–243, sep 1992. doi: 10.1007/BF01742751.
- [71] S.S. Sethi and A.G. Striz. On using the Kreisselmeier-Steinhauser function in simultaneous analysis and design. In *38th Structures, Structural Dynamics, and Materials Conference*, volume 2, pages 1357–1365, Reston, Virginia, apr 1997. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1997-1289.
- [72] A.B. Lambe and J.R.R.A. Martins. Extensions to the design structure matrix for the description of multidisciplinary design, analysis, and optimization processes. *Structural and Multidisciplinary Optimization*, 46(2):273–284, aug 2012. doi: 10.1007/s00158-012-0763-y.
- [73] MSC Nastran 2017 - Aeroelastic Analysis User's Guide, 2017.
- [74] A. De Marco, E. Duke, and J. Berndt. A General Solution to the Aircraft Trim Problem. In *AIAA Modeling and Simulation Technologies Conference and Exhibit*, number AUGUST 2007, pages 1–40, Reston, Virginia, aug 2007. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2007-6703.
- [75] J.T. Hwang. *A modular approach to large-scale design optimization of aerospace systems*. PhD thesis, 2015.
- [76] H.M. Adelman and R.T. Haftka. Sensitivity Analysis of Discrete Structural Systems. *AIAA Journal*, 24(5):823–832, may 1986. doi: 10.2514/3.48671.
- [77] G. Hodges and J. McKenzie. B-52 control configured vehicles maneuver load control system analysis and flight test results. In *13th Aerospace Sciences Meeting*, Reston, Virginia, jan 1975. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1975-72.
- [78] S.L. Gaulocher, Cl. Roos, and C. Cumer. Aircraft Load Alleviation During Maneuvers Using Optimal Control Surface Combinations. *Journal of Guidance, Control, and Dynamics*, 30(2):591–600, mar 2007. doi: 10.2514/1.25577.
- [79] J.A. Woods-Vedeler and A.S. Pototzky. Rolling Maneuver Load Alleviation using active controls. In *Dynamics Specialists Conference*, Reston, Virginia, apr 1992. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1992-2099.

- [80] S.V. Thornton. Reduction of structural loads using maneuver load control on the Advanced Fighter Technology Integration (AFTI)/F-111 mission adaptive wing. Technical report, *NASA-TM-4526*, 1993.
- [81] T.E. Disney. C-5A Active Load Alleviation System. *Journal of Spacecraft and Rockets*, 14(2):81–86, feb 1977. doi: 10.2514/3.57164.
- [82] V. Mukhopadhyay and E. Livne. Aeroservoelasticity. *Encyclopedia of Aerospace Engineering*, pages 1–18, 2010. doi: 10.1002/9780470686652.eae154.
- [83] MSC Nastran 2017 - Linear Static Analysis User's Guide, 2017.
- [84] R. L. Harder and R. N. Desmarais. Interpolation using surface splines. *Journal of Aircraft*, 9(2):189–191, feb 1972. doi: 10.2514/3.44330.
- [85] A. Sprater. Stabilizing Device for Flying-Machines, 1914.
- [86] E.B. Wilson. Theory of an Aeroplane Encountering Gusts. Technical report, *NACA Report 1, part 2*, 1915.
- [87] H.M. Davis and R.L. Swaim. Controlling dynamic response in rough air. In *3rd Annual Meeting*, number 66, Reston, Virginia, nov 1966. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1966-997.
- [88] D. McLean. Gust-alleviation control systems for aircraft. *Proceedings of the Institution of Electrical Engineers*, 125(7):675, 1978. doi: 10.1049/piee.1978.0159.
- [89] B. Etkin. Turbulent Wind and Its Effect on Flight. *Journal of Aircraft*, 18(5):327–345, may 1981. doi: 10.2514/3.57498.
- [90] A. Wildschek, R. Maier, K. Hahn, D. Leissling, M. Press, and A. Zach. Flight Test with an Adaptive Feed-Forward Controller for Alleviation of Turbulence Excited Wing Bending Vibrations. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–22, Reston, Virginia, aug 2009. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2009-6118.
- [91] O. Sensburg, J. Becker, H. Lusebrink, and F. Weiss. Gust Load Alleviation on Airbus A300. Technical report, *ICAS-82-2.1.1*, 1982.
- [92] M. Alam, M. Hromcik, and T. Hanis. Active gust load alleviation system for flexible aircraft : Mixed feedforward / feedback approach. *Aerospace Science and Technology*, 41:122–133, 2015. doi: 10.1016/j.ast.2014.12.020.
- [93] M.A. Burris, P.M. Bender. Aircraft Load Alleviation and Mode Stabilization (LAMS). Technical report, *AFFDL-TR-68-161*, 1969.
- [94] J.F. Johnston. Accelerated development and flight evaluation of active controls concepts for subsonic transport aircraft - Vol I: Load alleviation/extended span development and flight tests. Technical report, *NASA-CR-159097*, 1979. URL <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19820007203.pdf>.
- [95] Frederic M. Hoblit. *Gust Loads on Aircraft: Concepts and Applications*. American Institute of Aeronautics and Astronautics, Washington DC, jan 1988. doi: 10.2514/4.861888.

- [96] V.J. Sallee. Improved curve fitting techniques of frequency domain generalized aerodynamic forces. In *15th Atmospheric Flight Mechanics Conference*, Reston, Virginia, aug 1988. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1988-4355.
- [97] M. Idan, M. Karpel, and B. Moulin. Aeroservoelastic Interaction Between Aircraft Structural and Control Design Schemes. *Journal of Guidance, Control, and Dynamics*, 22(4):513–519, jul 1999. doi: 10.2514/2.4427.
- [98] B. Moulin, M. Idan, and M. Karpel. Aeroservoelastic Structural and Control Optimization Using Robust Design Schemes. *Journal of Guidance, Control, and Dynamics*, 25(1): 152–159, jan 2002. doi: 10.2514/2.4860.
- [99] Z. Wang and P.C. Chen. Accurate Rational Function Approximation for Time-Domain Gust Analysis. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, Reston, Virginia, jan 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-0414.
- [100] W. Eversman. A reduced cost rational-function approximation for unsteady aerodynamics. In *31st Structures, Structural Dynamics and Materials Conference*, Reston, Virginia, apr 1990. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1990-1155.
- [101] P.J. Goggin. A general gust and maneuver load analysis method to account for the effects of active control saturation and nonlinear aerodynamics. In *Dynamics Specialists Conference*, Reston, Virginia, apr 1992. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1992-2126.
- [102] N.K. Banavara and J.R. Newsom. Framework for Aeroservoelastic Analyses Involving Nonlinear Actuators. *Journal of Aircraft*, 49(3):774–780, may 2012. doi: 10.2514/1.C031246.
- [103] M Jeanneau, N. Aversa, S. Delannoy, and M. Hockenhull. AWIATOR’s Study of a Wing Load Control: Design and Flight-Test Results. *IFAC Proceedings Volumes*, 37(6):469–474, jun 2004. doi: 10.1016/S1474-6670(17)32219-X.
- [104] M. Karpel. Procedures and Models for Aeroservoelastic Analysis and Design. *ZAMM*, 81(9):579–592, sep 2001. doi: 10.1002/1521-4001(200109)81:9<579::AID-ZAMM579>3.0.CO;2-Z.
- [105] M.S. Eldred, V.B. Venkayya, and W.J. Anderson. Mode tracking issues in structural optimization. *AIAA Journal*, 33(10):1926–1933, oct 1995. doi: 10.2514/3.12747.
- [106] A.A. Abdallah, A.R. Barnett, T.W. Wildrick, R.T. Manella, and R.P. Miller. Stiffness-Generated Rigid-Body Mode Shapes for Lanczos Eigensolution with SUPORT DOF by Way of a MSC/Nastran DMAP Alter. In *MSC World Users’ Conference Proceedings*, Orlando, FL, 1994.
- [107] R.B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK Users’ Guide*. Society for Industrial and Applied Mathematics, jan 1998. doi: 10.1137/1.9780898719628.
- [108] MSC Nastran 2017 - Numerical Methods User’s Guide, 2017.
- [109] S.J. Hulshoff. *Aeroelasticity - Course Notes AE4930*. Delft University of Technology, 2011.

- [110] MSC Nastran 2017 - Dynamic Analysis User's Guide, 2017.
- [111] K.F. Alvin. Efficient Computation of Eigenvector Sensitivities for Structural Dynamics. *AIAA Journal*, 35(11):1760–1766, nov 1997. doi: 10.2514/2.24.
- [112] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, 1995. ISBN: 978-0134565675.
- [113] A.J. Den Hamer. *Data-driven optimal controller synthesis: a frequency domain approach*. PhD thesis, Eindhoven University of Technology, 2010.
- [114] K.J. Aström and R.M. Murray. *Feedback Systems: An Introduction for Scientists and Engineers*. Princeton University Press, 2008.
- [115] S. Haghghat, H.H.T. Liu, and J.R.R.A. Martins. Model-Predictive Gust Load Alleviation Controller for a Highly Flexible Aircraft. *Journal of Guidance, Control, and Dynamics*, 35(6):1751–1766, 2012. doi: 10.2514/1.57013.
- [116] D. Bates and I. Postlethwaite. *Robust Multivariable Control of Aerospace Systems*. Delft University Press, 2002.
- [117] G.I. Bara and M. Boutayeb. Static output feedback stabilization with  $H_\infty$  performance for linear discrete-time systems. *IEEE Transactions on Automatic Control*, 50(2):250–254, feb 2005. doi: 10.1109/TAC.2004.841922.
- [118] B.S. Liebst. Accelerometer placement in active flutter suppression systems. *Journal of Guidance, Control, and Dynamics*, 10(5):441–446, sep 1987. doi: 10.2514/3.20238.
- [119] G.J. Balas, C. Moreno, and P.J. Seiler. Robust Aeroservoelastic Control Utilizing Physics-Based Aerodynamic Sensing. In *AIAA Guidance, Navigation, and Control Conference*, pages 1–11, Reston, Virginia, aug 2012. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2012-4897.
- [120] L.H. Keel and S.P. Bhattacharyya. Controller Synthesis Free of Analytical Models: Three Term Controllers. *IEEE Transactions on Automatic Control*, 53(6):1353–1369, jul 2008. doi: 10.1109/TAC.2008.925810.
- [121] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control*. Wiley, second edition, 2001.
- [122] B.C. Kuo and F. Golnaraghi. *Automatic Control Systems*. Wiley, 9th edition, 2009. ISBN: 978-0470048962.
- [123] G. Vinnicombe. Frequency domain uncertainty and the graph topology. *IEEE Transactions on Automatic Control*, 38(9):1371–1383, 1993. doi: 10.1109/9.237648.
- [124] N. Fabre-Raimbault, M. Adurno, M. Berthereau, and S. Cote. Method and Device for Dynamically Alleviating Loads Generated on an Airplane, 2008.
- [125] Properties of Wrought Aluminum and Aluminum Alloys. Technical report, *ASM Handbook, Volume 2: Properties and Selection: Nonferrous Alloys and Special-Purpose Materials*, 1990.

- [126] J. Arbocz and J. Hol. On the Accuracy of Buckling Calculations using Smeared Stiffeners. In *47th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 14th AIAA/ASME/AHS Adaptive Structures Conference 7th*, Reston, Virginia, may 2006. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2006-2030.
- [127] K. Seywald. Wingbox Mass Prediction considering Quasi-Static Nonlinear Aeroelasticity. Technical Report October, *Master Thesis*, Technical University of Munich, 2011.
- [128] E. Albano and W.P. Rodden. A doublet lattice method for calculating lift distributions on oscillating surfaces in subsonic flows. In *6th Aerospace Sciences Meeting*, number 68, Reston, Virginia, jan 1968. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1968-73.
- [129] W.S. Rowe and H.J. Cunningham. On the convergence of unsteady generalized aerodynamic forces. In *Dynamics Specialists Conference*, Reston, Virginia, apr 1981. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.1981-647.
- [130] J. Ballmann, R. Eppler, and W. Hackbusch. *Panel Methods in Fluid Mechanics with Emphasis on Aerodynamics*. 1988. doi: 10.1007/978-3-663-13997-3.
- [131] J. Roskam. *Airplane Design - Part IV: Layout Design of Landing Gear and Systems*. DARcorporation, 1985.
- [132] S.J. Elliot. *Signal Processing for Active Control*. Elsevier, 2001.
- [133] J.P. Crimaldi, R.T. Britt, and W.P. Rodden. Response of B-2 aircraft to nonuniform spanwise turbulence. *Journal of Aircraft*, 30(5):652–659, sep 1993. doi: 10.2514/3.46394.
- [134] F.W. Diedrich and J.A. Drischler. Effect of Spanwise Variations in Gust Intensity on the Lift Due to Atmospheric Turbulence. Technical report, *NASA-TN-3920*, 1957. URL <https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/19930084862.pdf>.
- [135] D. Tang, A. Grasch, and E.H. Dowell. Gust Response for Flexibly Suspended High-Aspect Ratio Wings. *AIAA Journal*, 48(10):2430–2444, oct 2010. doi: 10.2514/1.J050309.
- [136] Z. Yang and H. Dang. Buffet Onset Prediction and Flow Field Around a Buffeting Airfoil at Transonic Speeds. In *51st AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference 18th AIAA/ASME/AHS Adaptive Structures Conference 12th*, Reston, Virginia, apr 2010. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2010-3051.
- [137] R. Vos and S. Farokhi. *Introduction to Transonic Aerodynamics*, volume 110 of *Fluid Mechanics and Its Applications*. Springer Netherlands, Dordrecht, 2015. doi: 10.1007/978-94-017-9747-4.
- [138] G.J.J. Ruijgrok. *Elements of Airplane Performance*. Delft University Press, 1990. ISBN: 9789062756087.
- [139] G.J. Kennedy, G.W. Kenway, and J.R.R.A. Martins. High Aspect Ratio Wing Design: Optimal Aerostructural Tradeoffs for the Next Generation of Materials. In

- 52nd Aerospace Sciences Meeting*, Reston, Virginia, jan 2014. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2014-0596.
- [140] J. Roskam. *Airplane Design - Part VI: Preliminary Calculation of Aerodynamic, Thrust and Power Characteristics*. DARcorporation, 1985.
- [141] J. Nandong. A unified design for feedback-feedforward control system to improve regulatory control performance. *International Journal of Control, Automation and Systems*, 13(1):91–98, feb 2015. doi: 10.1007/s12555-014-0090-8.
- [142] I. van Gent, P.D. Ciampa, B. Aigner, J. Jepsen, G. La Rocca, and J. Schut. Knowledge architecture supporting collaborative MDO in the AGILE paradigm. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, Reston, Virginia, jun 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-4139.
- [143] D. de Vries and I. van Gent. OpenLEGO Demonstration - A Link Between AGILE and OpenMDAO. In *First European OpenMDAO Workshop*, Toulouse, France, 2017. URL [http://pure.tudelft.nl/ws/files/34525330/20171013\\_OpenLEGO\\_demo.pdf](http://pure.tudelft.nl/ws/files/34525330/20171013_OpenLEGO_demo.pdf).