



**This electronic thesis or dissertation has been
downloaded from Explore Bristol Research,
<http://research-information.bristol.ac.uk>**

Author:
Simeone, Simone

Title:
Gust Loads Reconstruction for In-Service Support

General rights

Access to the thesis is subject to the Creative Commons Attribution - NonCommercial-No Derivatives 4.0 International Public License. A copy of this may be found at <https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>. This license sets out your rights and the restrictions that apply to your access to the thesis so it is important you read this before proceeding.

Take down policy

Some pages of this thesis may have been removed for copyright restrictions prior to having it been deposited in Explore Bristol Research. However, if you have discovered material within the thesis that you consider to be unlawful e.g. breaches of copyright (either yours or that of a third party) or any other law, including but not limited to those relating to patent, trademark, confidentiality, data protection, obscenity, defamation, libel, then please contact collections-metadata@bristol.ac.uk and include the following information in your message:

- Your contact details
- Bibliographic details for the item, including a URL
- An outline nature of the complaint

Your claim will be investigated and, where appropriate, the item in question will be removed from public view as soon as possible.

Gust Loads Reconstruction for In-Service Support

Simone Simeone

Industrial Doctorate Centre in Systems

University of Bristol



A dissertation submitted to the University of Bristol in
accordance with the requirements of the degree of
Engineering Doctorate in Systems in the Faculty of Engineering.

April 1, 2020

Declaration of Authorship

I, Simone SIMEONE, declare that this thesis titled, 'Gust Loads Reconstruction for In-Service Support' and the work presented in it are my own. I confirm that:

- This work was done wholly or mainly while in candidature for a Doctorate degree at this University.
- Where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated.
- Where I have consulted the published work of others, this is always clearly attributed.
- Where I have quoted from the work of others, the source is always given. With the exception of such quotations, this thesis is entirely my own work.
- I have acknowledged all main sources of help.
- Where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself.

Signed:

Date:

“When everything seems to be going against you, remember that an airplane takes off against the wind, not with it . . .”

Henry Ford

UNIVERSITY OF BRISTOL

Abstract

Faculty of Engineering

Industrial Doctorate Centre in Systems

Engineering Doctorate

Gust Loads Reconstruction for In-Service Support

by Simone SIMEONE

Gust and turbulence events are of primary importance in the estimation of limit loads and in the analysis of flight incidents. The estimation of limit loads is required during the design stage of an aircraft, whilst the analysis of flight incidents is requested for in-service support. The former requires accuracy whereas the latter requires conservatism and speed. A method for gust reconstruction based on numerical optimisation techniques is proposed in this work and validated on the response of aircraft models of increasing complexity to known input gust profiles. Satisfactory results are then obtained when applying the reconstruction framework to realistic events where the input gust profiles are assumed to be unknown. With the results obtained, a database of reconstructed gust profiles is then built and used to define the worst case gust conditions, following a new method proposed here that consists of a combination of the singular value decomposition technique, surrogate models and optimisation routines. The results of this application show that the underlying characteristics of gust and turbulence profiles can be used to define successfully the worst case gust conditions for aircraft design.

The second part of this work focuses on the harmonisation of in-service support within Airbus. To appraise incidents, the manufacturer is requested to perform an investigation into the nature of the exceedance and to indicate the remedial action, if any, to be performed before the aircraft can be released permanently back into service. Airbus' department of loads and aeroelasticity is responsible for assessing the loads exceedances on the reported flight and ground incidents on all Airbus aircraft. But, the current process for incident resolution has evolved organically over the years with site based responsibilities, which have followed the historical evolution of the Airbus organisation. This evolution resulted into a complex and intricate network, which

has called for a general reorganisation and remodernisation. A new centralised solution is therefore proposed here as a result of a thorough analysis guided by the fundamentals of systems engineering practice. This centralised solution is designed in such a way as to incorporate the gust reconstruction method proposed in the first part of this work and new methods that may be proposed in the future.

Acknowledgements

I would like to acknowledge the support received by the University of Bristol Systems Centre and the EPSRC funded Industrial Doctorate Centre in Systems (Grant EPG0373531) as well as the support and funding received from Airbus Operations UK Ltd.

I would also like to thank my academic supervisors, Thomas Rendall and Jonathan Cooper, and my industrial supervisor, Abdul Rampurawala, for their technical and moral support received throughout the years and for the development of this thesis.

Special thanks go to all the people with whom I worked in Airbus and who have made this work possible. In particular to: Olaf Lindenau and Frank Weiss for making this project achievable and for having taught me a lot about Airbus and the industry in general.

I would also like to thank Andrea Da Ronch for all his support and co-operation and all the people with whom I embarked on this rather long journey and, particularly, Christian, Feng, Jean, Gabriele, Andrea C., Carmine, Javon, Hamza, Andrea M., Spyros and Alex for the endless discussions and coffe breaks that made this experience unforgettable.

Finally, I would like to thank my parents for the long-distance support and, particularly, my fiancée for being always by my side and for providing extraordinary support in the many ups and downs that have characterised my journey - I love you!

Thank you!

Contents

Declaration of Authorship	i
Abstract	iii
Acknowledgements	v
List of Figures	x
List of Tables	xiv
Abbreviations	xv
Symbols	xvii
I Background	1
1 Introduction	2
1.1 Objectives	7
1.2 Contribution to knowledge and value-added	8
1.3 Thesis overview	11
1.4 Contents disclaimer	12
II The Reconstruction of Gust Loads	14
2 State of the art	15
2.1 Introduction	15
2.2 The flight loads model	17
2.3 Gust loads requirements	22
2.3.1 History	22
2.3.2 Discrete Gust Design Criteria (CS 25.341a)	28
2.3.3 Continuous Turbulence Design Criteria (CS 25.341b)	29
2.4 Gust loads methodology	31
2.4.1 Discrete gust methods	33

2.4.2	Continuous turbulence methods	37
2.5	The reconstruction of gust velocity profiles	39
2.6	Conclusions	43
3	Aerodynamic and aeroelastic models	45
3.1	Introduction	45
3.2	2DoF nonlinear model	47
3.3	Unsteady Lumped Vortex Method	50
3.3.1	Validation for the flow unsteadiness: Theodorsen's theory	52
3.3.2	Validation for the indicial response: Wagner's problem	56
3.3.3	Validation for a Sharp-Edged Gust: Küssner's problem	58
3.4	The Split Velocity Method in CFD	59
3.4.1	Reduced-order model for CFD	62
3.5	3D beam-stick model	63
3.6	Conclusions	68
4	The inverse method	70
4.1	Introduction	70
4.2	Methodology	71
4.2.1	The convolution integral	71
4.2.2	The use of Fourier transforms	73
4.3	Applications	74
4.3.1	2DoF model case study	74
4.3.2	3D-beam-stick case study	77
4.4	Industrial practice	79
4.5	Conclusions	80
5	The optimisation method	81
5.1	Introduction	81
5.2	Gradient-free optimisation framework	83
5.2.1	The objective function	86
5.2.2	The optimisation algorithm	87
5.3	Gradient-based optimisation framework	89
5.3.1	Gust shape parameterisation	92
5.3.1.1	Hicks-Henne bump functions	92
5.3.1.2	Radial basis functions	93
5.3.2	The objective function	94
5.3.3	The optimisation algorithm	95
5.3.3.1	Gradient evaluation	96
5.4	Method validation	97
5.4.1	2DoF linear model	98
5.4.2	2DoF nonlinear model	99
5.4.3	ULVM model	102
5.4.4	SVM-CFD model	104
5.4.5	Reduced order model	105
5.5	Data filtering	108
5.6	Modelling errors	108

5.7	Applications	109
5.7.1	2DoF nonlinear model	110
5.7.2	3D beam-stick model	112
5.8	Conclusions	115
6	Worst case gust prediction	117
6.1	Introduction	117
6.2	Methodology	121
6.2.1	SVD for gust matrix decomposition	122
6.2.2	Surrogate model of the aircraft gust response	123
6.2.3	The optimisation framework	125
6.3	Worst case gust prediction using the 3D beam-stick aircraft model	126
6.4	Conclusions	130
III	In-Service Support Development	133
7	A System of Systems	134
7.1	Introduction	134
7.2	The in-service support system	136
7.2.1	Soft Systems Methodology (SSM)	137
7.2.2	Hierarchical Process Modelling	142
7.2.3	Conclusions	145
8	Solution development	146
8.1	Introduction	146
8.2	System in context	148
8.3	Functional top-level decomposition	149
8.4	System architecture	151
8.5	System deployment	152
8.6	Documentation management	153
8.7	Conclusions	156
9	Conclusions	158
9.1	Outcome	158
9.1.1	The development of a framework for gust reconstruction	158
9.1.2	The development of an improved in-service support solution	159
9.2	Key findings	160
9.3	Outlook	161
A	Aerodynamics Derivatives	163
A.1	Longitudinal Aerodynamics Derivatives for Rigid Aeroplane [1]	163
A.2	Gust-Related Derivatives [1]	163
A.3	Longitudinal Aerodynamics Derivatives for Flexible Aeroplane [1]	164
B	Systems Engineering tools	165
B.1	Semi-structured interviews	165

C Python scripts	167
C.1 Multiprocessing for finite difference	167
D New ways of documents management	174
D.1 Documents Management	174
D.2 Alternative solutions	179
Bibliography	182

List of Figures

1.1	Discrete gust and turbulence	3
1.2	Fleet in-service evolution	5
1.3	Rich picture of current problem situation.	5
1.4	Thesis map	9
1.5	Pre-existing in-service support process.	10
1.6	Proposed solution for in-service support process.	11
2.1	Collar’s aeroelastic triangle	16
2.2	$C_L - \alpha$ linear regions.	19
2.3	Example of design speed envelope	20
2.4	Example of design weight envelope	21
2.5	Example of flight manoeuvring envelope	22
2.6	Gust envelope	23
2.7	Timeline evolution of gust requirements.	23
2.8	Küssner and Wagner functions.	25
2.9	Typical bell-shaped ‘1 – cos’ gust	26
2.10	Typical ‘1 – cos’ gust shapes	29
2.11	Basic system components for the analysis of gust response	31
2.12	Aircraft in trimmed flight in undisturbed atmosphere (a) and during a gust encounter (b).	33
2.13	Heave and pitch response of a rigid aircraft entering a ‘1 – cos’ gust.	36
2.14	Von Kármán PSD function.	38
2.15	Basic system components for the reconstruction of gust velocity profiles.	39
2.16	Discrete-gust approach to design	39
2.17	Spectral-technique: input-output relations for gust response	40
2.18	Monte-Carlo approach: statistical gust loads analysis	41
2.19	Force prediction via the inverse method (2.19a) and the optimisation method (2.19b).	42
2.20	Basic approach for the estimation of gust velocities and structural loads	42
2.21	Loads observer algorithm applied to UW-9 Sprint aircraft	43
3.1	Aircraft models at different design stages	47
3.2	Aerofoil section with trailing edge flap.	48
3.3	Comparison between linear full-order model (FOM) and nonlinear full-order model (NFOM) of pitch (3.3a) and heave (3.3b) responses to a ‘1 – cos’ gust.	50
3.4	ULVM sketch	50
3.5	Representation of the harmonically oscillated foil used by Theodorsen.	53
3.6	Validation against Theodorsen’s theory: pure translation	55

3.7	Validation against Thedorsen's theory: pure pitch	56
3.8	The Wagner's problem.	56
3.9	The Wagner's problem: indicial lift computed by ULVM code with $\alpha = 1.0$, $V_\infty = 1.0$ and $c = 2b = 1.0$	58
3.10	The Küssner's problem.	58
3.11	The Küssner's problem: indicial lift computed by ULVM code with $\alpha = 1.0$, $V_\infty = 1.0$ and $c = 2b = 1.0$	60
3.12	FFAST crank aerofoil.	62
3.13	Forward velocity contour plot of FFAST aerofoil	63
3.14	FFAST beam stick model and DLM.	64
4.1	Force prediction via the inverse method where FFT denotes the forward Fourier transform and $IFFT$ the inverse Fourier transform.	70
4.2	Illustration of the transfer function operation.	72
4.3	Illustration of the inverse transfer function operation.	73
4.4	Example of swept-frequency cosine (<i>chirp</i>) signal	75
4.5	Systems' transfer functions for $\alpha, \dot{\alpha}, \xi$ and $\dot{\xi}$ obtained with Equation 4.10 and using the chirp signals of Figure 4.4.	76
4.6	Comparison between applications of the inverse method to the linear (a) and nonlinear (b) version of the wing typical section subject to a '1 - cos' gust	77
4.7	Systems' transfer function for the aircraft mid-fuselage acceleration	78
4.8	Comparison of inverse method applications to '1 - cos' gusts of decreasing gradient and amplitude (a) and (b) and to a gust input of the chirp form (c).	78
4.9	Main components of the industrial process used for the reconstruction of gust loads. Source: Airbus' tool user guide.	79
5.1	Force prediction via the optimisation method.	83
5.2	Proposed Data Flow.	84
5.3	Example of a parameterised gust input \tilde{f}_g	86
5.4	Nelder-Mead algorithm flowchart	88
5.5	Operations performed by the Nelder-Mead algorithm on a 2-dimensional simplex	88
5.6	Gradient-based optimisation framework	91
5.7	Example of Hicks Henne Bump Functions parameterisation with 8 basis functions. . . .	93
5.8	Example of RBF parameterisation with 8 basis functions.	94
5.9	Representation of finite difference approximation.	97
5.10	Validation diagram.	97
5.11	Gust reconstruction diagram for the linear 2DoF model.	98
5.12	Comparison of the '1 - cos' gust profile ($\bar{H} = 40.0, \bar{U}_0 = 0.01$) with the gust reconstructed via the optimisation run on the linear FOM. The error plotted in (b) is the delta between the target function and the reconstructed one.	99
5.13	Convergence plots over the gust profile for the 2DoF linear model	100
5.14	Comparison between target and reconstructed pitch (5.14a) and plunge (5.14b) for the linear full order model.	100
5.15	Gust reconstruction diagram for the nonlinear 2DoF model.	100
5.16	Comparison of the '1 - cos' gust profile ($h_g = 40.0, \Omega_0 = 0.10$) with the gust reconstructed via the optimisation run on the nonlinear FOM. The error plotted in (b) is the delta between the target function and the reconstructed one.	101
5.17	Convergence plots over the gust profile for the 2DoF linear model	101

5.18	Comparison between target and reconstructed pitch (5.18a) and plunge (5.18b) responses to a ‘1 – cos’ gust for the nonlinear case.	102
5.19	Gust reconstruction diagram for ULVM.	102
5.20	Gust reconstruction using ULVM	103
5.21	Gust reconstruction diagram for CFD.	104
5.22	Gust reconstruction using SVM-CFD model	105
5.23	Convergence plots	106
5.24	Gust reconstruction diagram for ROM.	106
5.25	Gust reconstruction using ROM model	107
5.26	Proposed Data Flow with Digital Filter.	108
5.27	Example of time domain approximation of von Kármán spectrum, Hanning-windowed.	110
5.28	Gust reconstruction diagram for the nonlinear 2DoF model.	111
5.29	Comparison between a time-approximated Von Kármán gust and its reconstruction obtained using the aerofoil nonlinear model. The error plotted in (b) is the delta between the target function and the reconstructed one.	111
5.30	Comparison between target and reconstructed pitch (5.30a) and plunge (5.30b) responses to a Von Kármán gust.	112
5.31	Gust reconstruction diagram for the 3D beam-stick model.	112
5.32	Comparison between a time-approximated Von Kármán gust and its reconstruction obtained using the 3D beam-stick model. The error plotted in (b) is the delta between the target function and the reconstructed one.	113
5.33	Comparison between target and reconstructed acceleration response to a von Kármán gust.	114
5.34	Gust reconstruction diagram for in-service applications.	115
6.1	Matched filter theory and evolutionary algorithm method	118
6.2	Response surface (surrogate) model used for the search of the worst-case gust	119
6.3	Two-dimensional loads envelope at a generic load station	120
6.4	Overview of the SVD application to the worst-case gust problem.	121
6.5	Graphical representation of the gust matrix \mathbf{W}_g	122
6.6	Sample representation of a surrogate model, also known as response surface. The red dots represent the training data points.	123
6.7	The “sample size vs. dimensions” diagram of default techniques in MACROS GT Approx	124
6.8	Definition of surrogate model.	124
6.9	Flowchart of the optimisation used for the identification of the worst-case gust.	126
6.10	First 25 SVD-reduced characteristic gust shapes.	127
6.11	Singular values σ_k (left) and cumulative energy contained in the first k modes.	128
6.12	Comparison between actual and approximated values of N_z acceleration response.	129
6.13	Worst case gust shape.	130
6.14	Acceleration response at the aircraft c.g. location (a) and shear force (b), bending moment (c) and torque (d) at the wing root.	131
7.1	Problem situation classification and systems approaches	136
7.2	The cycle of action research.	139
7.3	The learning cycle of Soft Systems Methodology (SSM)	139
7.4	Conceptual model derived from root definition.	141

7.5	Problem situation expressed by HPM	143
8.1	Pre-existing in-service support process.	147
8.2	Proposed solution for in-service support process.	147
8.3	Time-Box Iterative model	148
8.4	Actors.	149
8.5	To-be contextual view.	150
8.6	Position in the system.	150
8.7	Functional top-level decomposition.	151
8.8	ISST sequence diagram.	151
8.9	Overview of ISST deployment process.	153
8.10	WiKi portal: Installation.	155
8.11	WiKi portal: Video Tutorial (screenshot).	155
8.12	Timeline of ISST development during the EngD project.	156
9.1	Entry into service (EIS) of each iteration (release) during the EngD project.	160
9.2	Live global air traffic in a normal day of operation during January, 2019.	162
D.1	Documentation management main page	178
D.2	Working area: Home	178
D.3	Working area: Project Management	178
D.4	Working area: Technical Deliveries	179
D.5	Rapid Application Development process	181

List of Tables

2.1	Differences between original and revised gust formula	27
3.1	Summary of aeroelastic models and reconstruction methods used.	68
4.1	Types of windowing functions and their typical application	74
5.1	Performance of the models used for the validation of the optimisation framework.	116
7.1	<i>CATWOE</i> analysis applied to in-service support.	140
7.2	‘What seems to be the case now’ (AS-IS) with ‘what might be’ (TO BE)	142

Abbreviations

AC	A ircraft
ARD	A Rchitecture D ossier
BFGS	B royden– F letcher– G oldfarb– S hanno
BRD	B usiness R equirements D ossier
CG	C entre of G ravity
CFD	C omputational F luid D ynamics
CPU	C entral P rocessing U nit
DLM	D oublet L attice M ethod
DOF	D egree O f F reedom
EIS	E ntry I nto S ervice
FDR	F light D ata R ecorder
FEM	F inite E lement M odelling
FFAST	F uture F ast A eroelastic S imulation T echnologies
FFT	F ast F ourier T ransform
FRF	F requency R esponse F unction
HHBF	H icks- H enne B ump F unctions
HPM	H ierarchical P rocess M odelling
HTP	H orizontal T ail P lane
IFFT	I nverse F ast F ourier T ransform
GP	G aussian P rocesses
HDA	H igh D imensional A pproximation
ISST	I n- S ervice S upport T ool
IT	I nformation T echnology
IQ	I nteresting Q uantity
MAC	M ean A erodynamic C ord

MAX	Maximum Absolute Error
MDOF	Multiple Degree Of Freedom
PSD	Power Spectral Density
RAD	Rapid Application Development
RBF	Radial Basis Functions
ROM	Reduced Order Model
RRMS	Root Mean Square Error
RSM	Response Surface Model
SMT	Shear Moment Torque
SSM	Soft Systems Methodology
SVD	Singular Value Decomposition
SVM	Split Velocity Method
VTP	Vertical Tail Plane
ULVM	Unsteady Lumped Vortex Method

Symbols

\mathbf{A}	Aerodynamic influence coefficient matrix
\mathbf{C}	Aerodynamic damping matrix
$\bar{\mathbf{C}}$	Generalised matrix of aerodynamic damping
$\bar{\mathbf{D}}$	Generalised matrix of structural damping
\mathbf{E}	Aerodynamic stiffness matrix
$\bar{\mathbf{E}}$	Generalised matrix of aerodynamic stiffness
\mathbf{F}	Matrix of external forces
$\bar{\mathbf{F}}$	Generalised matrix of input force coefficients
$\bar{\mathbf{K}}$	Generalised matrix of structural stiffness
\mathbf{M}	Mass matrix
$\bar{\mathbf{M}}$	Generalised matrix of structural inertia
\mathbf{T}	Transformation matrix
\mathbf{T}	Transformation matrix
\mathbf{U}	Matrix of coefficients for SVD
\mathbf{V}	Unitary matrix
\mathbf{X}_{AC}	Aircraft parameters measurement vector
Φ	Matrix of normal modes of vibration
Σ	Diagonal matrix of singular values
\mathbf{q}	Displacement vector of generalised coordinates
$\dot{\mathbf{q}}$	Rate vector of generalised coordinates
$\ddot{\mathbf{q}}$	Acceleration vector of generalised coordinates
\mathbf{u}	Vector of input forces

\mathbf{u}_a	Structural physical displacements
\mathbf{x}_g	Vector of gust points
C_L	Aircraft lift coefficient
C_ℓ	Local lift coefficient
C_m	Local pitching moment coefficient
C_{m_0}	Zero lift pitching moment coefficient for the wing
C_p	Pressure coefficient
C_α	Viscous damping in pitch
C_ξ	Viscous damping in plunge
E	Total energy
E_g	Component of total energy associated with gust velocities alone
F_g	Flight profile alleviation factor
H	Gust gradient distance
\bar{H}	Nondimensional gust gradient distance
I_y	Pitch moment of inertia
I_α	Second moment of area of aerofoil about elastic axis
K_g	Gust alleviation factor
K_α	Pitch stiffness about elastic axis
K_ξ	Plunge stiffness about elastic axis
L	Lift force
\bar{L}	Characteristic scale wavelength of turbulence
L_g	Lift due to gust
L_α	Lift due to aircraft angle of attack
P_{Li}	Design limit load for the considered load quantity
$P_{(1g)i}$	Steady load for the considered load quantity
P_{li}	Max. incremental load for the considered load quantity
S	Wing reference area
S_α	First moment of area of aerofoil about e. a.
U	Gust velocity
U_0	Gust peak velocity
\bar{U}	Nondimensional gust velocity
\bar{U}_0	Nondimensional gust peak velocity

U_e	Effective gust velocity
U_{de}	Derived gust velocity
U_{ds}	Maximum design gust velocity
U_{ref}	Reference gust velocity
U_σ	Power-spectral scale factor
$U_{\sigma_{ref}}$	Reference turbulence intensity
V	Aircraft true air speed
V_∞	Freestream velocity
V^*	Reduced velocity, $V/b\omega_\alpha$
V_A	Design manoeuvring speed
V_B	Design speed for maximum gust intensity
V_C	Design cruise speed
V_D	Design dive speed
W	Weight force
a	Lift curve slope
\bar{a}	Pitch axis location relative to aerofoil mid-chord
b	Aerofoil semi-chord
c	Wing/aerofoil chord
f_g	Gust velocity profile function
\tilde{f}_g	Approximation of gust velocity profile function
l	Distance between wing aerodynamic centre and tail aerodynamic centre
l_W	Distance between wing aerodynamic centre and aircraft centre of mass
l_T	Distance between tailplane aerodynamic centre and aircraft centre of mass
k	Reduced frequency, $k = \omega b/V$
m	Aerofoil sectional mass
n	Load factor
n_x, n_y, n_z	Aircraft acceleration in x, y and z direction
q_∞	Dynamic pressure
r_a	Radius of gyration of aerofoil about e. a. axis, $r_a^2 = I_\alpha/mb^2$
s	Distance penetrated into the gust
t	Time
u, v	Velocity components

u_g, v_g	Gust prescribed velocity components
x^w	Distance between wake vortices
z	Displacement in the vertical direction
w_{gEAS}	Gust vertical velocity in equivalent air speed
Γ	Net circulation about the aerofoil
Ω	Scaled frequency, rad/m
Ψ	Küssner function
Φ	Wagner function
Φ_{VK}	Von Kärman power spectral density
α	Aircraft incidence
α_g	Angle of attack due to gust
$\beta_\alpha, \beta_{\alpha_5}$	Nonlinear rotational spring constants
β_ξ, β_{ξ_5}	Nonlinear spring constants
γ	Flight path angle
γ_j	Dihedral angle of j^{th} panel
μ	Aerofoil mass ratio, $m/\pi\rho b^2$
μ_g	Mass parameter
ω	Angular frequency
$\bar{\omega}$	Ratio of ω_ξ/ω_α
ω_α	Uncoupled pitching mode natural frequency about e. a.
ω_ξ	Uncoupled plunging mode natural frequency
ρ	Density of air
ρ_0	Density of air at sea level
σ_x	Root mean square value of the quantity x
τ	Non-dimensional time, Vt/c
$\tau_{1/2}$	Non-dimensional time, Vt/b
θ	Pitch angle
ξ	Plunge motion
ζ_α	Damping ratio in pitch motion
ζ_ξ	Damping ratio in plunge motion

$\mathcal{F}\{\cdot\}$	Fourier transform
$\mathcal{F}^{-1}\{\cdot\}$	Inverse Fourier transform
$*$	Convolution operation
$\dot{()}$	Differentiation w.r.t. time, t
$()'$	Differentiation w.r.t. non-dimensional variable, τ

To my past, my present and my future...

Part I

Background

Chapter 1

Introduction

An incident is an event in flight or on the ground due to which the maximum allowable loads on the aircraft and its components are deemed to have exceeded. Aircraft incidents are classified into two major categories: flight and ground. Flight incidents include excessive loads due to turbulence and flight manoeuvres and ground incidents include hard landings and ground manoeuvres. This work mainly focuses on flight incidents and, particularly, on the calculation of the source of the dynamic loads exerted on the aircraft structure during operation. An increasing level of attention is being paid on the study and definition of turbulence when these are encountered by in-service aircraft under various operating conditions. Turbulence¹, or gust, is generally defined as the movement of the air through which an aircraft passes and where any component of the velocity of the air that is normal to the flight path (i.e. gust velocity) will change the effective incidence of the aerodynamic surfaces; in turn, this will cause sudden changes in the lift forces and hence a dynamic response of the aircraft involving flexible deformation [1]. The major problems that arise from turbulence are related to the static design strength of the aircraft and to fatigue but additional problems may also arise with turbulence i.e. difficulties in the precise control of flight paths (for example landing approach), passenger and crew discomfort, loss of control triggered by clear air turbulence, high speed dive, high elevator forces or, in military aircraft, weapon aiming/delivery and air-air refuelling. Clearly, turbulence encounters are one of the major causes for the additional costs incurred by aircraft

¹The term *turbulence* should not to be confused with turbulence modelling, which is typically discussed for CFD applications.

manufacturers and operators, due to mission cancellations and diversions, aircraft loss or repairs for damages, reduced airframe life due to fatigue, requirements for special crew trainings, increased aircraft inspections or even medical costs due to injuries and insurance claims.

Aircraft are expected to encounter atmospheric turbulence (or ‘rough air’) of varying degrees of severity during their in-service lifecycle. Because severe turbulence may affect the static design strength of the aircraft and moderate turbulence may cause fatigue damage to their structure [2, 3], all modern flight vehicles require the evaluation of dynamic loads in response to discrete and random gust excitation as early as the design stage. Figures 1.1b and 1.1a show extracts from real events that are indicative of the two categories, namely *discrete gusts* and *continuous turbulence*, used for design purposes since the early days of civil and military aircraft development. The former models gust velocities that change in a deterministic manner and the latter models, instead, random changes of gust velocity. Depending on the category, different methods in the time domain or in the frequency domain and different aircraft models exist for the analysis. But, with the increased complexity of today’s aircraft, the two idealised categories may no longer cover for the higher level of interaction of all the aircraft system components and the advancement in technology and computer power shall be used to open new ways for the design and in-service analysis of these events.

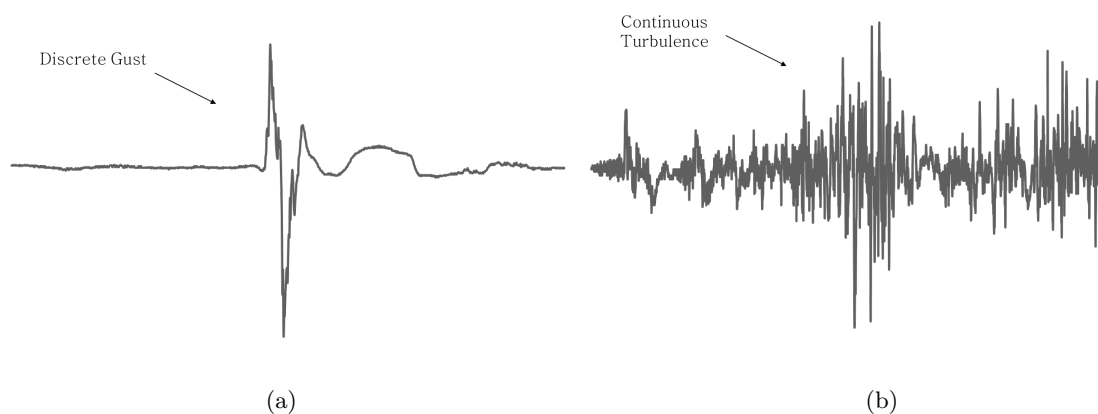


FIGURE 1.1: Vertical gust speed time history from real events of a discrete gust experienced by a long haul aircraft (a) and a continuous turbulence recorded during a hailstorm on a short haul aircraft (b).

Henceforth, this work aims to propose a new method for the reconstruction and decomposition of the properties of atmospheric turbulence, considering real (recorded) events as starting points. The reconstruction of atmospheric turbulence is not a straight forward process as, on today

aircraft instrumentation, it is not currently possible to distinguish (or separate) the portion of the aircraft response that is purely due to the deflection of control surfaces from the effects that atmospheric turbulence have on the behaviour of the aircraft. For example, when considering real recorded events, the aircraft incidence parameter available from the flight data recorder includes both the effects of the pilot manoeuvring the aircraft and atmospheric turbulence impacting the aircraft, and no accurate methods exist to perform such distinction efficiently. In mathematical terms, the sought objective is to obtain a relation where the total aircraft incidence parameter is defined as

$$\alpha_{tot} = \alpha_{a/c} + \alpha_g \quad (1.1)$$

with $\alpha_{a/c}$ being the aircraft incidence due to the manoeuvring operations of the aircraft only and α_g being the incremental incidence due to gust only. Additionally, the isolation of the gust contributions could be translated into a database to be used for future aircraft development. Such database could set the ground for a data analytics approach that could be used to remove conservatism when determining whether the aircraft state is inside the bounds for safe operation, either during design or after an in-service event, or to establish new ways for the estimation of the fatigue life of the aircraft. Furthermore, it could determine a new way of defining the worst possible gust conditions.

A deeper understanding of turbulence would also lead to a better reconstruction of the resulting loads, which could be of major benefit also for the operator of the aircraft; in fact, knowing the loads at any location of the aircraft structure during or soon after flight can avoid unnecessary grounding and/or extensive and expensive inspection of the structure. As a result, the time interval of regular aircraft inspections could be increased, or these inspections only be conducted on special parts of the aircraft if information on the loads experienced by the aircraft during flight were readily available, thus reducing operating costs of the aircraft and increasing its availability [4]. Aircraft on ground (AOG) are, in fact, a threat for operators as they imply huge losses of money in a market that is continuously growing, with an additional demand of 34,900 aircraft required within the next 20 years [5]. Figure 1.2 shows the likely evolution of the civil aviation market until the year 2036, where circa 40% of the aircraft demand are needed for the replacement of the current fleet and an impressive 60% are needed for the growth. This increase in in-service aircraft will proportionally increase the number of reported incidents. A reported incident requires the manufacturer to perform an investigation on the nature of the exceedance and the remedial action, if any, to be performed before the aircraft can be released permanently back into service. The Airbus department for loads and aeroelasticity is responsible

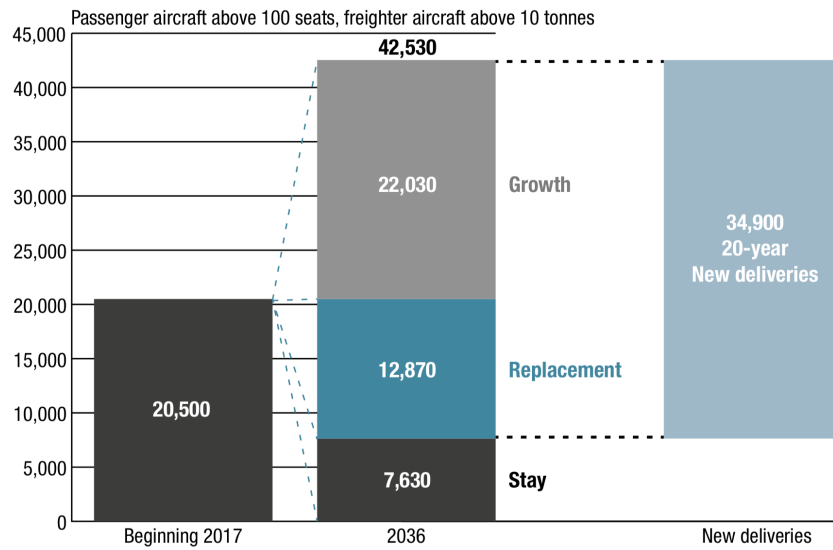


FIGURE 1.2: Fleet in-service evolution (rounded figures to the nearest 10) [5].

to assess the loads exceedances on the reported flight and ground incidents on all Airbus aircraft. The current process for incident resolution requires running specific tools for various aircraft families and for each type of incident. The typical workflow for the assessment of an aircraft

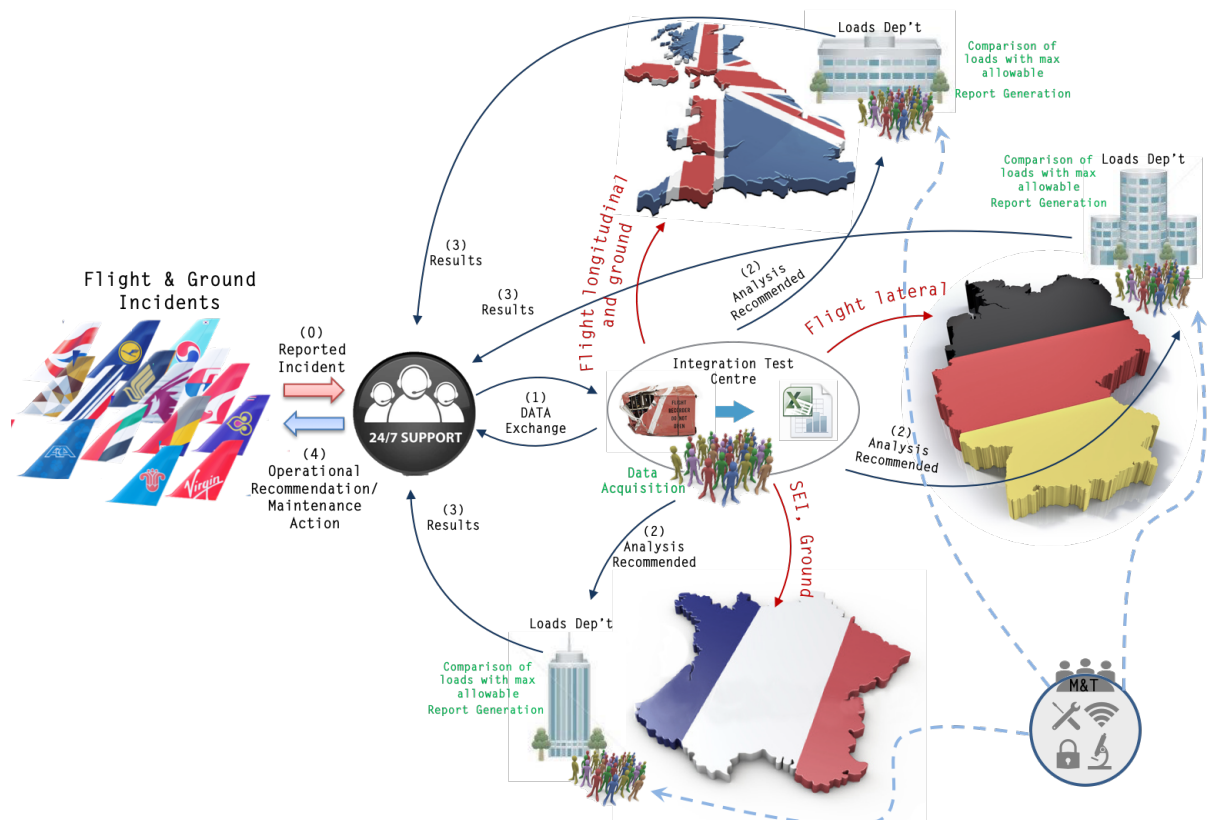


FIGURE 1.3: Rich picture of current problem situation.

incident is represented in Figure 1.3 and follows the points outlined below:

- a flight or ground incident is reported by an airline to Airbus customer services (0);
- data from the flight data recorder (FDR)² is transferred (1) to the Integration Test Centre (ITC), which is responsible to convert the data into a human readable format, in the form of databanks and/or reports;
- the on-call engineer (available 24/7) receives the request (2) to perform the incident analysis directly from the ITC;
 - based on the nature of the incident, the on-call engineer is located in France, Germany or the UK and has access to site-based expertise and ad-hoc software solutions;
- a first assessment of the reported incident is obtained with the aid of look-up charts and tables, which classify the incident based on the level of severity;
 - if the level of severity is medium or high, a detailed analysis is performed by the on-call engineer, who may require up to 2 weeks to clear the incident, depending on the tools available to perform the appropriate analysis;
- results in the form of reports, signed by the relevant lead engineer, are sent back to customer services (3);
- customer services are then responsible to communicate any operational recommendation and/or maintenance action to the airline.

This process has evolved organically over the years with site based responsibilities, which have followed the historical evolution of the Airbus organisation. It is now recognised that to support an ever increasing fleet there is a requirement to reorganise the overall process to efficiently respond to incidents. As of today, incidents' analyses could require from as low as a couple of hours, in the case of hard landing incidents where dedicated tools are already in place, to as much as several weeks in all other cases. This brings the attention to the second goal of this work that is to level down the turnaround time to few hours for all types of incidents, for all aircraft families, by reorganising the incident support chain and centralising it with the introduction of a unique support tool.

²The flight data recorder (FDR) is a device used to record specific aircraft performance parameters. Its purpose is to store data on a medium that is designed to survive under extreme conditions that can result from an accident. It can include more than 900 parameters and the format and content usually vary with the aircraft.

1.1 Objectives

The objectives of this project can be summarised as:

1. The development of a method for the evaluation of gusts and turbulence properties based on the reconstruction of gust velocity profiles obtained through an optimisation framework applied to aircraft parameters available from flight data recorders;
 - (a) The availability of an extensive collection of real gust and turbulence profiles shall open the opportunity for the derivation of the worst possible gust condition;
2. The centralisation of the incident support process that implies:
 - (a) A reduction in aircraft on ground events;
 - (b) The rationalisation of aircraft maintenance costs;
 - (c) A reduction in man-hour costs for the training and operation of incidents support tools.

The second objective can be bonded in a concise root definition, developed by means of systems engineering practice (§7.2.1), which reads:

“A system to transfer existing incidents’ analysis tools to a unique interface to achieve the localisation of the incidents’ analysis process that allows a more efficient customer service and a reduction in aircraft on ground, maintenance costs and training sessions required.”

This system can be represented in a global context as a black box that provides connections to external tools that allow handling, analysis and representation of recorded flight data. In this context, only two user profiles shall be identified in the interaction with this in-service support black box; these are the *Customer Services* and the *On-call Team Member*. The former provides the initial inputs, i.e. loads reports³ and flight data recordings⁴, and is the recipient of the results of the analyses; it doesn’t perform any computation with the tool itself. The latter

³The “Loads Report 15” is a report produced by the aircraft instrumentation following an incident that provides aircraft parameters data such as gross weight, c.g. location, pitch, roll and yaw angles and correlated max/min vertical, longitudinal and lateral acceleration at the time of the event.

⁴Flight data recordings are extracted from the Flight Data Recorder (FDR) which is located on board of the aircraft.

receives the inputs and performs the computations with the tool; s/he is part of the department of loads and aeroelasticity.

If on one hand, the first objective aims to fulfill the novelty requirement from a purely academic point of view, on the other hand, its solution is designed to be industrialised as part of the system developed to achieve the second objective of this work. Figure 1.4 shows how these two objectives fit together within this work and it is meant to be read as the map of this thesis. In fact, the top branch extends towards the theoretical development of the time domain optimisation method, which represents the novel approach for the reconstruction of gusts and moves away from inverse methods; the bottom branch extends towards the development of the in-service support tool, achieved through Systems Engineering practice. These two branches are interconnected by the implementation, within the support tool, of the most suitable reconstruction method resulting from the top branch i.e. the gradient based optimisation framework with CFD reduced order models. This reconstruction method, in fact, given its strong computational performance may be used as a solver for real time applications of the in-service support tool and, thus, reconstruct gust profiles lively from in-service data (see Chapter 5). A third branch develops towards the top of the figure and represents the opportunity to derive worst case gust conditions - an opportunity that arises from the large amount of data produced by the methods developed in this work (see Chapter 6).

1.2 Contribution to knowledge and value-added

The first main objective of this work aims to contribute to the body of knowledge by proposing a method for extracting gust and turbulence properties from incident data. As mentioned earlier, the reconstruction of the gust shape would allow a better estimation of the loads exerted on the structure. In fact, understanding the properties of any turbulence encountered by in-service transport aircraft under various operating conditions is of increasing interest to aircraft manufacturers, which are putting effort into this process in order to collect valuable information for future aircraft development. Over the last four years, a method has been developed based on a framework designed around numerical optimisation algorithms that are able to reconstruct virtually any turbulence profile starting from the analysis of flight data recorders [6]. Although it is not part of the software for in-service support yet, this method has been tested and validated on several academic case studies, including computational fluid dynamics applications and reduced order models, and so far results have been published in three conference papers [6–8]. Moreover,

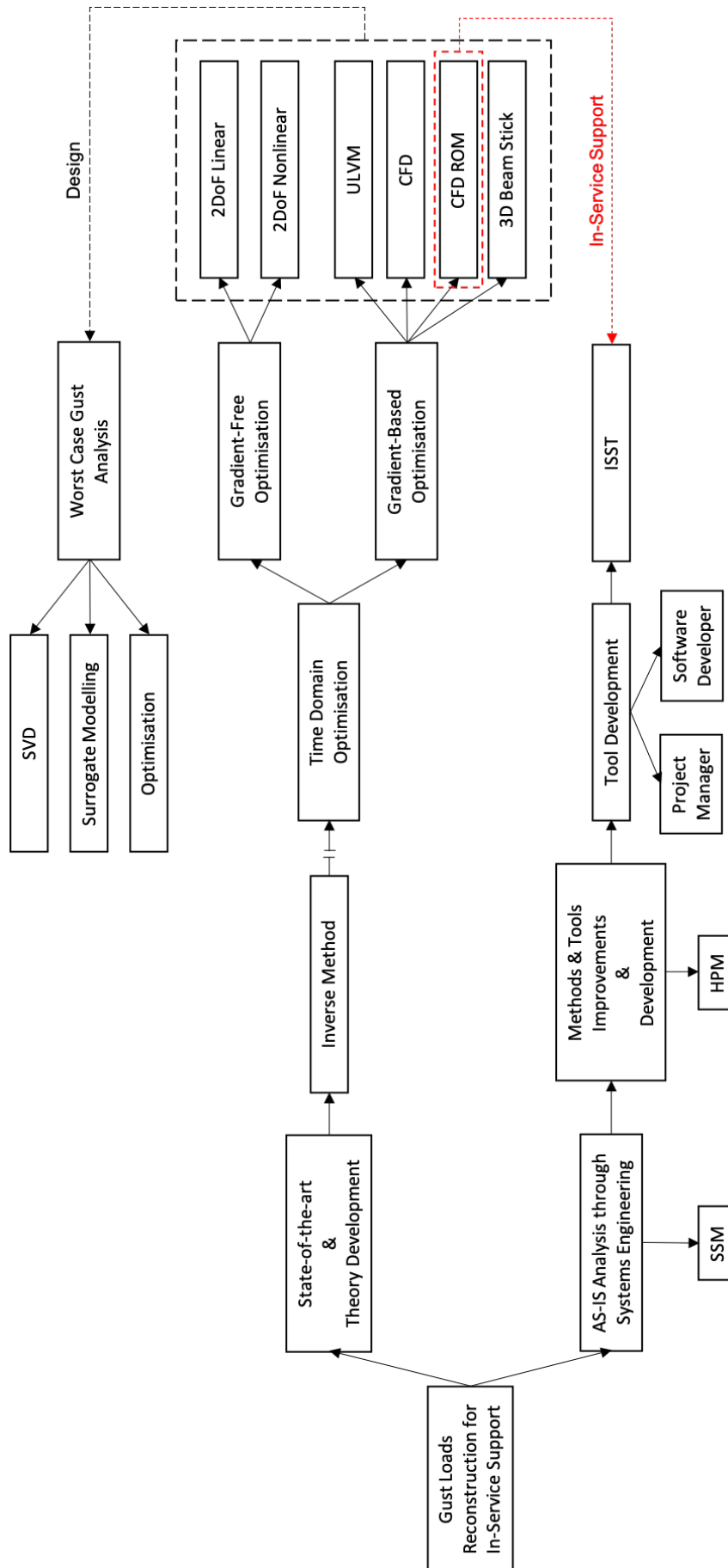


FIGURE 1.4: Thesis map

the proposed reconstruction method has been applied within a framework established through this work that aims to contribute to the search of the worst case gust condition, which is of increasing interest among aircraft designers for certification purposes.

The second main objective of this work, instead, has been approached with formal methods in systems engineering that “[...] added value to the final deliverable as well as improved the efficiency in managing the various organisational interfaces” – A. Rampurwala, Airbus Operations Ltd. This strategy covered the application of two problem structuring methods, namely Soft Systems Methodology (SSM) and Hierarchical Process Modelling (HPM)⁵; the former was used to identify and formulate the problem and to plan the actions required for the resolution, all through the use of conceptual models and interviews, and the latter was used to define the structure for the development of the agreed solution. Finally, the agreed solution consists in the migration from a situation where the on-call engineer (one of the key actors in the in-service support chain, responsible for running the analyses) has specific skills and expertise – that are function of aircraft and incident type - required to assess the incident, to a situation where any member of the on-call engineer’s team can assess any incident without the need of a specific set of requirements. The following flow charts clearly show to what extent the new solution simplifies the pre-existing in-service support process.

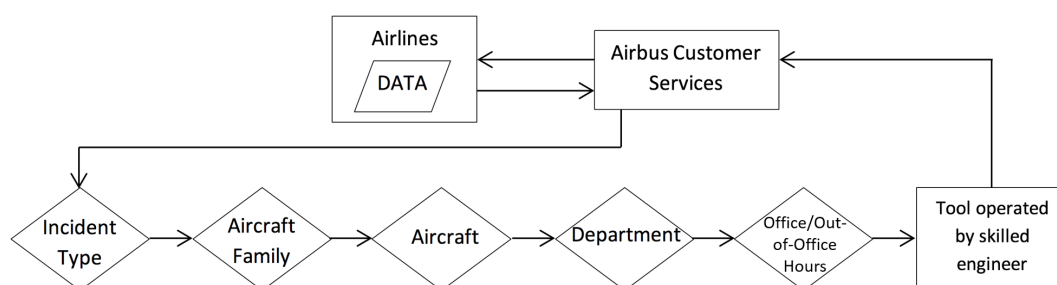


FIGURE 1.5: Pre-existing in-service support process.

In fact, the new solution promotes the change from a situation where the incident data only reached the skilled on-call engineer after a 5-step decision process, as shown in the lower part of Figure 1.5, to a situation where a single actor external to the customer service team, i.e. the on-call team member⁶, has access to a unique software solution that does not require incident

⁵Soft Systems Methodology (SSM) and Hierarchical Process Modelling (HPM) are problem structuring methods used within Systems Engineering practice to tackle problems that are not well defined, involve many interested parties with different perspectives and are subject to multiple interpretations [9, 10]. A description of these methods, and the application within this work, is discussed in Chapter 7.

⁶The on-call team member is a skilled engineer who is responsible for running the incident analysis promptly, after a request is received from Customer Services. His/her availability is controlled by an internal rota and, when on-call, s/he responds to incident analysis requests 24 hours a day.

specific skills and that has the capability to handle the analysis of a wide range of/virtually all incidents (Figure 1.6). Thus, this solution shall improve the in-service support process by

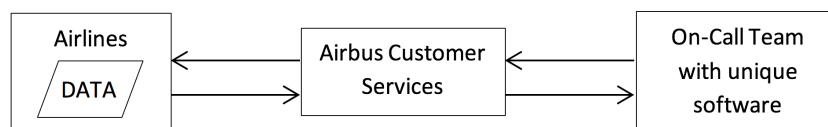


FIGURE 1.6: Proposed solution for in-service support process.

reducing the lead time for aircraft back-to-service status, saving revenues at airlines level as well as improving Airbus reputation. It is a solution that relies on the development of new and harmonised software that:

- is unique for all types of aircraft and all types of loads incidents, thus enabling flexible and robust on-call service;
- has greater accuracy from use of production tools implemented for in-service support;
- delivers faster resolution from increased automation and vastly improved graphical user interface.

1.3 Thesis overview

This thesis is divided into three parts. **Part I** forms the background and includes this introductory chapter. In **Part II** the reconstruction of gust loads is discussed from Chapter 2 to Chapter 6. **Chapter 2** introduces the needs, the historical development and the methodologies used in industry and academia for the reconstruction of gust loads. **Chapter 3** introduces the aerodynamic and aeroelastic models that are used in this work, starting from a flat plate modelled using potential flow theory and then following with models of higher levels of complexity such as CFD and three dimensional models. **Chapter 4** gives an overview of the inverse method approach in frequency domain. This consists of the reconstruction of an exciting force by observing the response measurements and evaluating the inverse of the forward system by means of Fourier transforms. The limitations of applicability of this method are emphasized in this chapter. **Chapter 5** covers the description and application of optimisation techniques in the time domain for the reconstruction of gusts and turbulence profiles. Two methods for the optimisation are discussed in this chapter i.e. gradient-free and gradient based algorithms. The strategy for the validation of the method is also presented, followed by its application to

the aerodynamic and aeroelastic models introduced earlier. **Chapter 6**, the last one of Part II, proposes a novel approach for the prediction of the worst case gust based on the application of the singular value decomposition technique and the use of optimisation with surrogate models. **Part III** presents the in-service support topic from the software development point of view and includes the last three chapters of this thesis. **Chapter 7** provides an analysis of the in-service system from the systems engineering perspective. The application of problem structuring methods, including Soft Systems Methodology and Hierarchical Process Modelling, is illustrated here and concluded by a critical review of the methods. **Chapter 8** focuses on the methods & tools part and provides a description of the steps necessary for the industrialisation of the new proposed solution. **Chapter 9** provides a summary of the achievements and the industrial impact as well as an outlook on future applications and on different areas of engineering, and concludes this work.

1.4 Contents disclaimer

Some parts of this work are taken or adapted from the following papers, which have been published externally or submitted as module assignments to the University of Bristol to fulfill the requirements of the ‘EngD in Systems Engineering’ programme:

- S. Simeone, C. Agostinelli, T. Rendall and A. Rampurawala, “Gust Reconstruction from Digital Flight Data Recorder via Numerical Optimisation” in 57h AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, San Diego, California, 2016.
- S. Simeone, T. Rendall, A. Da Ronch and A. Rampurawala, “A Gust Reconstruction Framework Applied to a Nonlinear Reduced Order Model of a Wing Typical Section” in 58h AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Grapevine, Texas, 2017.
- S. Simeone, T. Rendall, S. Williams, C. W. Wales, J. Cooper, D. Jones and A. L. Gaitonde, “Reconstruction of Gust Velocity Profiles via Potential Flow, CFD and ROM techniques” in International Forum for Aeroelasticity and Structural Dynamics, Como, Italy, 2017.

-
- S. Simeone, “Problem Structuring Methods applied to define In-Service Support Development in a Large Aircraft Manufacturer” submitted to the University of Bristol as a requirement of the Socio Technical Systems module, EngD Programme, Bristol, UK, 2014.
 - S. Simeone, “A Systems Engineering Approach for Information Systems Development in a Large Aircraft Manufacturer” submitted to the University of Bristol as a requirement of the Systems Engineering module, EngD Programme, Bristol, UK, 2014.
 - S. Simeone, “Software Development: Using an Alternative Approach to Deliver Documentation for Passing the Project Launching and Commitment Milestone Reviews (M3/M5)” submitted to the University of Bristol as a requirement of the Integrating Engineering and Management Systems module, EngD Programme, Bristol, UK, 2015.
 - S. Simeone, J. Ramsden, “A cross-examination of Life Cycle Analysis in building and aerospace industries as a means of defining and achieving sustainability objectives” submitted to the University of Bristol as a requirement of the Complex Systems Design module, EngD Programme, Bristol, UK, 2015.

Part II

The Reconstruction of Gust Loads

Chapter 2

State of the art

The first objective of this work is to develop a method for the reconstruction and evaluation of gusts and turbulence properties for the assessment of aircraft loads. Thus, a general understanding of the theory that is the basis for to the calculation of aircraft loads is fundamental. This chapter covers the historical development of the theory and the requirements that set the ground for the application of gust loads reconstruction in large aircraft manufacturer industry. Moreover, a general overview is given on the process followed by the loads & aeroelastics department for the definition and assessment of aircraft loads, during the several design stages of a commercial aircraft.

2.1 Introduction

Generally, the assessment of loads is required for the definition of the design strength or fatigue capability that the structure of an aircraft should be able to withstand during the entirety of its lifespan. Two types of loads are considered, namely, static and dynamic loads and fatigue loads. Static and dynamic loads form the vast majority of calculations and are those loads that the structure must withstand without permanent deformation or failure. Fatigue loads are much smaller than static and dynamic loads but the aircraft structure must be able to withstand the repeated loading of fatigue loads without leading to failure. Examples of loads to be assessed are shear force, bending moments and torque in a wing, fuselage or tail, and normal and tangential loads or pitching moments in structural members. Usually, knowledge of wing shear force, bending moment and torque, also known as SMT loads, is sufficient but for

more detailed designs the evaluation of inertia loads, aerodynamic loads and external loads (e.g. engine thrust) and internal stresses may also be necessary. Information on loads are needed as early as the “pre-design” stage of an aircraft for the overall sizing of the aircraft configuration but their knowledge is also required at later stages. At the “detail design” stage, in fact, loads provide strength targets for the design of each components; at the “checkstress” stage, they are required to confirm that all the elements of the detailed design are in accordance with the design targets; at the “certification” stage, where the structural engineers have to demonstrate that the aircraft is strong enough to withstand the loading conditions specified in the airworthiness requirements; and, finally, during in-service, as the aircraft shall be released permanently back into service after a reported flight or ground incident. Flight and ground incidents, in fact, represent one of the main sources of loads that an aircraft will experience during its life; these are generated by turbulence in the atmosphere, landing at various descent velocities and/or ground manoeuvres such as taxiing, turning, braking, surface bumps etc. Other sources of loads are, instead, pilot induced manoeuvres within defined load factor limits and control systems and/or engine failure cases.

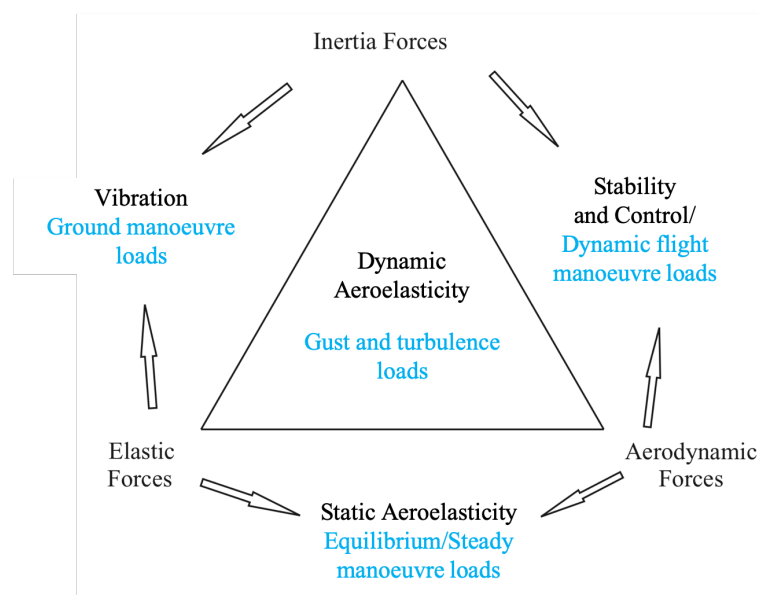


FIGURE 2.1: Collar’s aeroelastic triangle and correlated load cases. Adapted from Wright and Cooper [1].

In all cases, loads are ultimately a combination of the applied load condition (e.g. gust, control surface displacement, landing gears reaction, etc.) and the inertia loading from the distributed mass of the aircraft. Collar’s aeroelastic triangle can be adopted to provide a fuller picture of the forces involved in the generation of the different loading conditions [1]. This is shown in Figure 2.1 where it can be seen that the equilibrium (or steady) manoeuvres loads derive

from the interaction of elastic forces with aerodynamic forces and, thus, sit within the discipline of static aeroelasticity; also, the interaction of aerodynamic forces with inertia forces generates dynamic flight manoeuvres loads that are embraced by the discipline of stability and control; and part of vibration studies are, instead, ground manoeuvre loads that generate from the mutual interaction of inertia forces and elastic forces. Finally, the full interaction of inertia forces with aerodynamic forces and elastic forces form the gust and turbulence loads that involve the study of dynamic aeroelasticity. These four families of loads are classified for certification into two main categories: *bookcase* and *rational* [1]. The former refers to relatively artificial states of the aircraft in which applied and inertia loads are in equilibrium; these are mostly considered for equilibrium manoeuvres but also apply for some ground manoeuvres. The latter, instead, refers to more realistic conditions where the aircraft dynamics are modelled and simulated accordingly; these mostly apply to dynamic and ground manoeuvres and gust or turbulence events. These load cases define the maximum loads (also defined as limit loads in the certification requirements) that are expected in service and that the aircraft structure must be able to withstand without detrimental permanent deformation. Ultimate loads are also part of the certification requirements and are defined as limit loads times a safety factor (in general of 1.5); they must also be supported by the aircraft without experiencing failure.

2.2 The flight loads model

The evaluation of load cases requires a well-established flight loads process. This is generally based upon a model constructed over the MDOF system described by the following generalised equation

$$\bar{\mathbf{M}}\ddot{\mathbf{q}} + (\rho V \bar{\mathbf{C}} + \bar{\mathbf{D}})\dot{\mathbf{q}} + (\rho V^2 \bar{\mathbf{E}} + \bar{\mathbf{K}})\mathbf{q} = (\rho V)\bar{\mathbf{F}}\mathbf{u} \quad (2.1)$$

where $\bar{\mathbf{M}}$, $\bar{\mathbf{C}}$, $\bar{\mathbf{D}}$, $\bar{\mathbf{E}}$, $\bar{\mathbf{M}}$ and $\bar{\mathbf{F}}$ are, respectively, generalised matrices of structural inertia, aerodynamic damping, structural damping, structural stiffness, aerodynamic stiffness and input force coefficients; ρ is the density of air, V the true air speed, \mathbf{q} the displacement vector of generalised coordinates with dimensions $(N_m \times 1)$, with N_m being the number of modes considered sufficient to cover the frequency range of interest in the MDOF system, $\dot{\mathbf{q}}$ is the rate vector of generalised coordinates (mode shapes) with dimensions $(N_m \times 1)$, $\ddot{\mathbf{q}}$ is the acceleration vector of generalised coordinates with dimensions $(N_m \times 1)$ and \mathbf{u} is the vector of input forces and has dimensions $(N_f \times 1)$. The generalised matrices provide the contributions per unit mode and are

defined as

$$\begin{aligned}
 \bar{\mathbf{F}} &= \mathbf{\Phi}^T \mathbf{F} && \text{with dimensions} && (N_m \times N_f) \\
 \bar{\mathbf{M}} &= \mathbf{\Phi}^T \mathbf{M} \mathbf{\Phi} && \text{with dimensions} && (N_m \times N_m) \\
 \bar{\mathbf{C}} &= \mathbf{\Phi}^T \mathbf{C} \mathbf{\Phi} && \text{with dimensions} && (N_m \times N_m) \\
 \bar{\mathbf{E}} &= \mathbf{\Phi}^T \mathbf{E} \mathbf{\Phi} && \text{with dimensions} && (N_m \times N_m)
 \end{aligned}$$

where $\mathbf{\Phi}$ is the matrix of normal modes of vibration and $\mathbf{\Phi}^T$ its transpose, \mathbf{F} is the original matrix for external forces due to engine thrust, control surface deflections, gust inputs, etc., \mathbf{M} is the original mass matrix, \mathbf{C} is the original aerodynamic damping matrix and \mathbf{E} is the original aerodynamic stiffness matrix. An auxiliary equation is then used to get the forces on a certain part of the structure, i.e.

$$\text{Loads} = \mathbf{Y} = \bar{\mathbf{F}}_T \mathbf{u} - \bar{\mathbf{M}}_T \ddot{\mathbf{q}} - \bar{\mathbf{C}}_T \dot{\mathbf{q}} - \bar{\mathbf{E}}_T \mathbf{q} \quad (2.2)$$

with

$$\begin{aligned}
 \bar{\mathbf{F}}_T &= \mathbf{T} \bar{\mathbf{F}} \\
 \bar{\mathbf{M}}_T &= \mathbf{T} \bar{\mathbf{M}} \\
 \bar{\mathbf{C}}_T &= \mathbf{T} \bar{\mathbf{C}} \\
 \bar{\mathbf{E}}_T &= \mathbf{T} \bar{\mathbf{E}}.
 \end{aligned} \quad (2.3)$$

Here, \mathbf{T} is known as the T-matrix and is used as a ‘mask’ to filter only the areas of interest in the structure; this matrix, in fact, defines what are known as Interesting Quantities (IQs). For example, setting parts of \mathbf{T} to zero will isolate the wing or other components from the rest of the structure. Each unit contribution multiplied by \mathbf{q} , i.e. the amount of each mode, provides the overall mass, aerodynamic damping and stiffness contributions.

It is clear from equations 2.1 to 2.3 that the flight loads model consists of four principal building blocks, namely, **aerodynamic data**, **mass data**, **T-matrix** and **modes**. From an industrial point of view, each building block translates to data that is produced by the relative department. The aerodynamic block contains thousands of “data labels” that describe the aerodynamic characteristics of the aircraft but only those that are applicable to loads calculations are considered in the flight loads model. Key labels are: *zero lift pitching moment coefficient for the wing* (C_{m_0}), *wing pitching moment coefficient as a function of incidence* ($\partial C_m / \partial \alpha$), *fuselage lift coefficient as a function of incidence* ($\partial C_L / \partial \alpha$), *downwash gradient* ($\partial \epsilon / \partial \alpha$) and *alpha transitions*

$(\alpha_1, \alpha_2, \alpha_3)$. Further data delivered within the aerodynamic block concern the engines (pods), control surfaces and more general terms such as downwash gradients and C_{Lmax} . To simplify

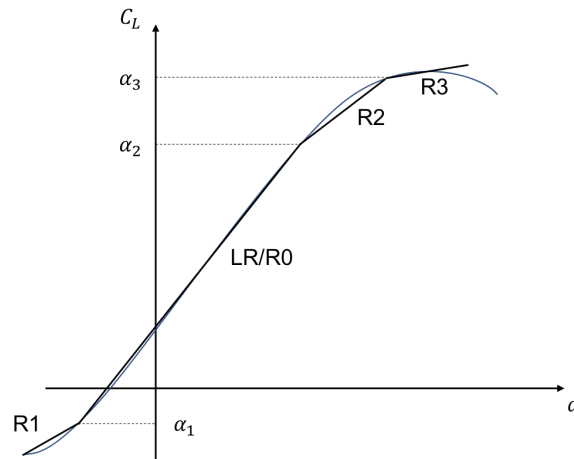


FIGURE 2.2: $C_L - \alpha$ linear regions.

the calculations, data that is dependent on incidence (α) is linearised and split into four linear aerodynamic regions (R0-3), which are defined as in Figure 2.2. Particular attention goes to region R0, also referred to as the leading region (LR), because it represents the area where the aircraft flies (mostly) at $1g$ in cruise and it is, indeed, the area where the wing is optimised to fly. Also, all gusts are calculated within this region, because the incidence achieved by the aircraft, even in extreme gusts, rarely exceeds the limits of R0. For structural sizing, however, the loads within all regions must be considered; this is only achievable by using different aerodynamic models that consider all the possible wing configurations i.e. slats, flaps and airbrakes retracted or extended and combinations of these. Furthermore, all loads cases must be covered for all the Mach numbers that the aircraft is capable to fly.

The mass properties department and the fuel group provide the mass data block. This includes details on the mass of the major components of the aircraft, i.e. wing, fuselage, horizontal tail plane (HTP), vertical tail plane (VTP), engines and landing gears. The sum of these components provides the mass of the entire aircraft. Each component mass is delivered in the form of condensed mass points; for large components such as wings, fuselage and tails, several mass points are needed whereas engines and landing gears are generally delivered as one mass point, with an associated CG with respect to a reference point. These information form two basic mass cases; both consider the mass of the aircraft without any fuel but one is for retracted landing gears and one is for extended landing gears. Although the mass remains the same in both cases, the CG and inertia properties are different. These two mass cases are labelled as OWER - Operators Weight Empty Retracted - and OWEE - Operators Weight Empty Extended

- and form the basis of a wide range of mass cases that differ in the level of fuel that sits in each tank of the aircraft (wing tanks, fuselage tank and trim tank).

As stated above, the T-matrix defines the IQs of the aircraft or, in other words, the position of the stations for which loads are sought. When additional information are included in the T-matrix (e.g. distances from the wing root) bending moments, torque, accelerations, or even elevator deflection and tail incidence, can be extracted. Two subparts form the T-matrix: one that provides the loads contribution given by the aerodynamic matrices and one that provides the contribution from the structural matrices. However, only loads quantities that are obtained by linear summation can be considered.

The last component of the flight loads model is the modes data block, which is delivered by the aeroelastics team. The computed aircraft modes are based on a Finite Element Model (FEM) that is the combination of different FEMs delivered by the wing team, the fuselage team, the engine/pylon team, the VTP team and HTP team. These modes combined with the mass and aerodynamic matrices provide the generalised matrices of equations 2.2 and ??.

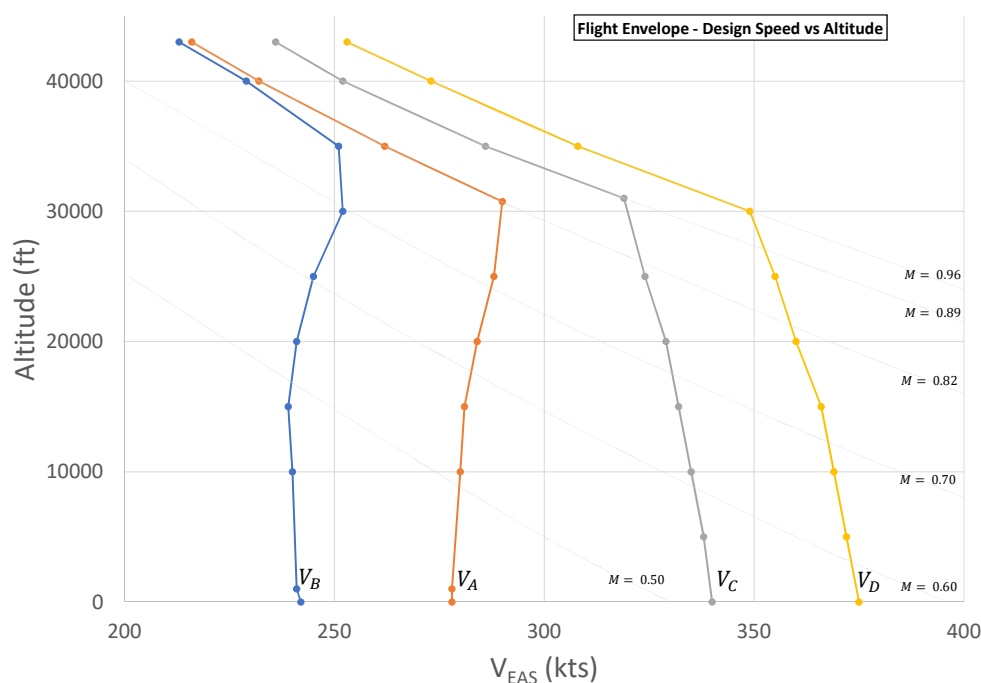


FIGURE 2.3: Example of design speed envelope. The coloured dotted lines indicate the aircraft design speeds (i.e. V_A is the maximum speed for manoeuvring, V_B is the maximum speed for maximum gust intensity, V_C is the maximum cruise speed and V_D is the maximum dive speed) at the different altitudes and the light grey lines indicate the equivalent Mach number.

The complete flight loads model can perform calculations for all Mach numbers and all mass cases. However, the combination of Mach numbers, mass cases and also altitudes generate an infinite number of flight points that would be impossible to evaluate. *Flight envelopes* are thus defined to enclose the flight points within well-established boundaries, or speedcurves, that refer to specific aircraft performance criteria. These speedcurves are defined by the design manoeuvring speed (V_A), the design speed for maximum gust intensity (V_B), the design cruise speed (V_C) and the design dive speed (V_D), which is a combination of extreme conditions that define the maximum speed that the aircraft should ever reach. Figure 2.3 shows an example of design speed envelope; this is used to identify the critical load cases within the speed/altitude range of the aircraft and at enough points on (and sometimes within) the boundary lines to ensure that the maximum loads for each part of the aeroplane structure are obtained (CS25.321 [11]). Other flight envelopes are also used to identify the critical load cases at each weight and for any practical distribution of fuel and payload (Figure 2.4) and to define maximum speeds and load factors (Figure 2.5).

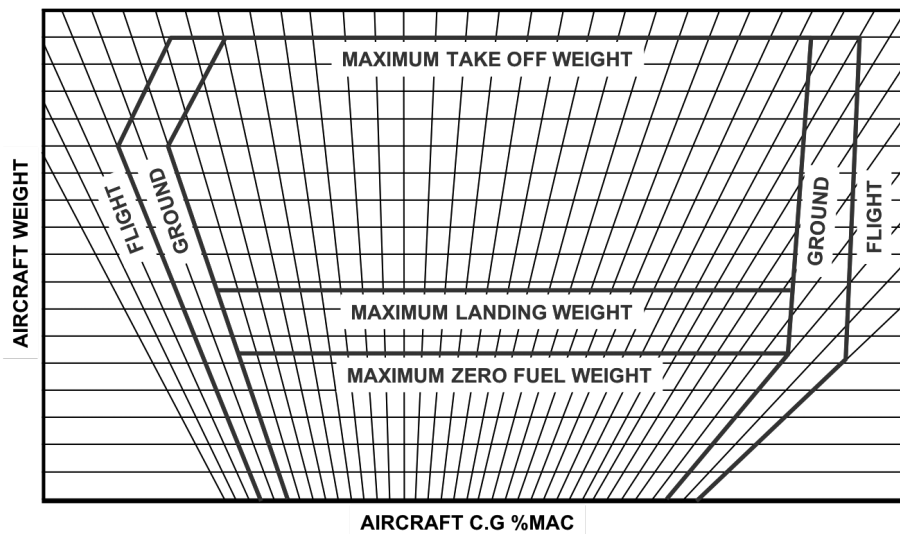


FIGURE 2.4: Example of design weight envelope where the aircraft c.g. is expressed as a percentage of the wing Mean Aerodynamic Cord (MAC). Source: Wright and Cooper [1].

A ‘gust envelope’ also exists (Figure 2.6), which is a graphical representation of

$$n = 1 + \frac{\rho_0 w_{gEAS} a K_g}{2 W/S} V_{EAS} \quad (2.4)$$

otherwise known as Pratt’s equation [12], which was used historically to define the load factors for design at different flight conditions [1]. The envelope is constructed evaluating Equation 2.4 for different gust velocities w_{gEAS} and drawing lines between the intersection of load factors n

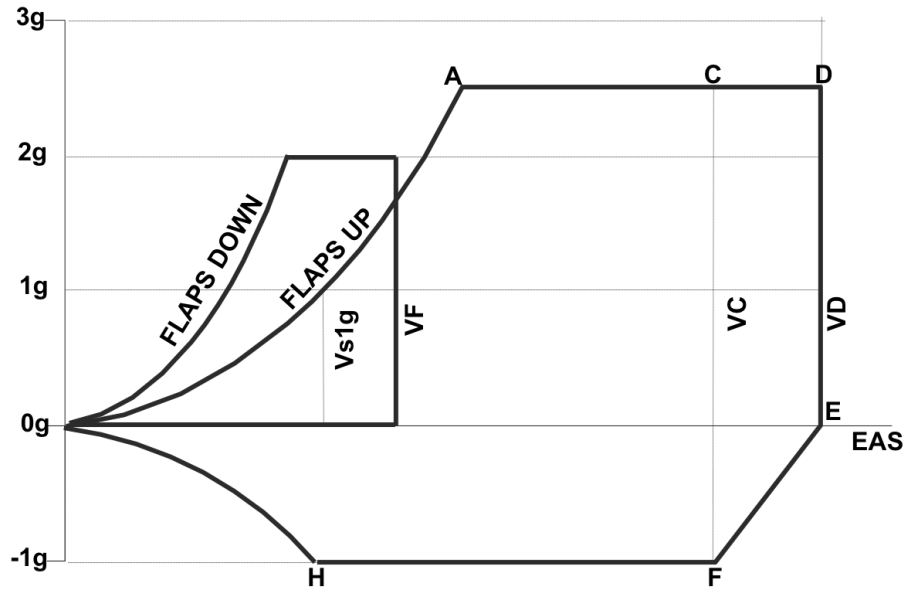


FIGURE 2.5: Example of flight manoeuvring envelope. Source: Certification Specifications [11].

and the specified flight velocities V_B , V_C and V_D , expressed in EAS. The corners represent the areas of major interests where most of the design calculations are to be carried out. However, this approach is no longer used by aircraft manufacturers and does not form part of the certification requirements [1]. A brief history on the evolution of aircraft gust loads design requirements is given in the next section.

2.3 Gust loads requirements

2.3.1 History

The two gust design criteria that can be found in the Certification Specifications [11] and are reported in Section 2.3.2 and 2.3.3, are the result of studies that started in the first half of the 20th century and whose milestones are indicated in Figure 2.7. These studies were first documented in a report from NACA (U.S. National Advisory Committee for Aeronautics, now NASA) titled “Theory of an Aeroplane Encountering Gusts” and published in the year 1917 [13]. Fuller, in his journal article on the evolution of gust loads design requirements [14], states that the first gust loads regulation was established in 1934 and was based on an aeroplane encountering a sharp edge or step function gust of intensity U ; then, with the assumption of quasi-steady aerodynamics and no vertical motion of the aeroplane, the incremental lift was

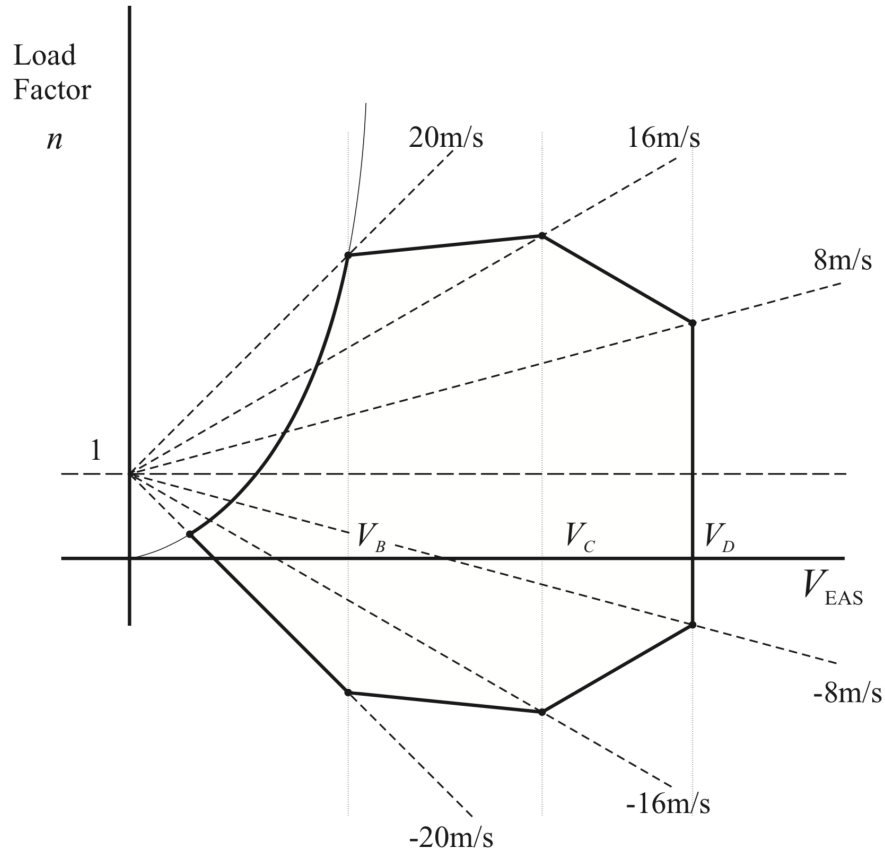


FIGURE 2.6: Gust envelope on a forward velocity vs aeroplane load factor (V - n) diagram. Note: gust velocities are expressed in Equivalent Air Speed (EAS). Source: Wright and Cooper [1].

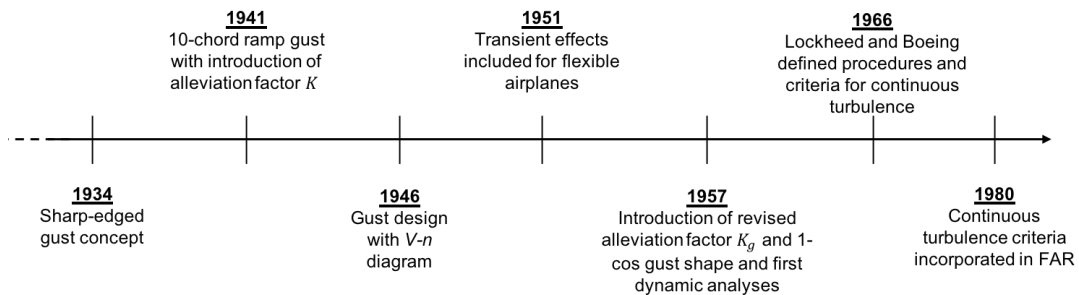


FIGURE 2.7: Timeline evolution of gust requirements.

given by

$$\Delta L = a \left(\frac{\rho V^2}{2} \right) S \left(\frac{U}{V} \right) \quad (2.5)$$

where a is the lift curve slope per radian and U/V is the incremental angle of attack (in rad) due to the gust. With the help of Newton's second law, the acceleration increment could then be obtained as

$$\Delta n = \frac{1/2 \rho a V U}{W/S}. \quad (2.6)$$

In 1941, the alleviation factor K was introduced into the sharp-edge gust equation to account for the nonsteady aerodynamic effects due to gust penetration and the aeroplane vertical response motion. These two reactions were identified during studies conducted by NACA on the Lockheed XC-35 aeroplane [14]. When first introduced, the alleviation factor varied with wing loading ($K = 1.0$ being the wing loading of the Boeing 247) and was based on the idea that the gust velocity increased linearly with distance, up to a limit of 10 chords. Gust design values later became dependent on aircraft speed, with the introduction in 1946 of gust design requirements based on the V - n diagram (as the one shown in Figure 2.6), and by 1951 transient effects on flexible aircraft also entered the equation. The dimensionless mass parameter, μ_g , which takes into account not only the wing loading but also air density (ρ), wing chord (c) and lift curve slope (a), and that is defined as

$$\mu_g \equiv \frac{2W}{S a \rho c g} = \frac{2M}{S a \rho c} \quad (2.7)$$

was introduced in the mid-1950s as the sole variable defining the revised gust alleviation factor, K_g . This factor, which can be understood as the ratio between the peak normal acceleration from a discrete gust and the reference value of peak acceleration from Equation 2.6 [15], constituted the renowned *revised gust load formula*

$$\Delta n = \frac{1/2 \rho a V U}{W/S} K_g \quad (2.8)$$

also introduced earlier as Pratt's equation (Equation 2.4) and whose derivation is shown in Section 2.4. The introduction of K_g was necessary to accommodate for the aerodynamic lag that resulted from the build-up of lift during the gust penetration and for the response of the aeroplane due to the sudden change in incidence. These effects were first approximated by Küssner [16] and are, respectively, incorporated in what are known today as the Küssner (Ψ) and Wagner (Φ) functions. Küssner function accounts for the change in velocity whilst Wagner function accounts for the sudden change in angle of attack. These functions are illustrated in Figure 2.8 as a function of number of chord lengths travelled but can be described by considering the variation with time of the gust velocity U and of the aeroplane angle of attack α as a series of (infinitesimally) small incremental step changes. As a result, Ψ and Φ can be expressed as indicial functions written, respectively, as

$$\Psi(t) = \frac{L_g(t)}{L_\alpha(\infty)} \quad \text{and} \quad \Phi(t) = \frac{L_\alpha(t)}{L_\alpha(\infty)} \quad (2.9)$$

where $L_\alpha(\infty) = \frac{1}{2}\rho V^2 a S$ and gives the lift due to the aircraft angle of attack. Defining the step incremental changes in gust velocity and incidence, respectively, as

$$\Delta U = \dot{U} \Delta t \quad \text{and} \quad \Delta \alpha = \Delta \left(\frac{\dot{z}}{V} \right) = \frac{1}{V} \dot{z} \Delta t, \quad (2.10)$$

the total incremental lifts $L_g(t)$ and $L_\alpha(t)$ at time t given by the sum of all the step changes due to, respectively, gust velocity and incidence become

$$L_g(t) = \frac{1}{2} V^2 a S \sum_0^t \Psi\{t-t^*\} \dot{U}\{t^*\} \Delta t^* \equiv \frac{1}{2} V^2 a S \int_0^t \Psi\{t-t^*\} \dot{U}\{t^*\} dt^* \quad (2.11)$$

$$L_\alpha(t) = \frac{1}{2} V a S \sum_0^t \Phi\{t-t^*\} \dot{z}\{t^*\} \Delta t^* \equiv \frac{1}{2} V a S \int_0^t \Phi\{t-t^*\} \dot{z}\{t^*\} dt^* \quad (2.12)$$

where z indicates the displacement of the overall aircraft in the vertical direction.

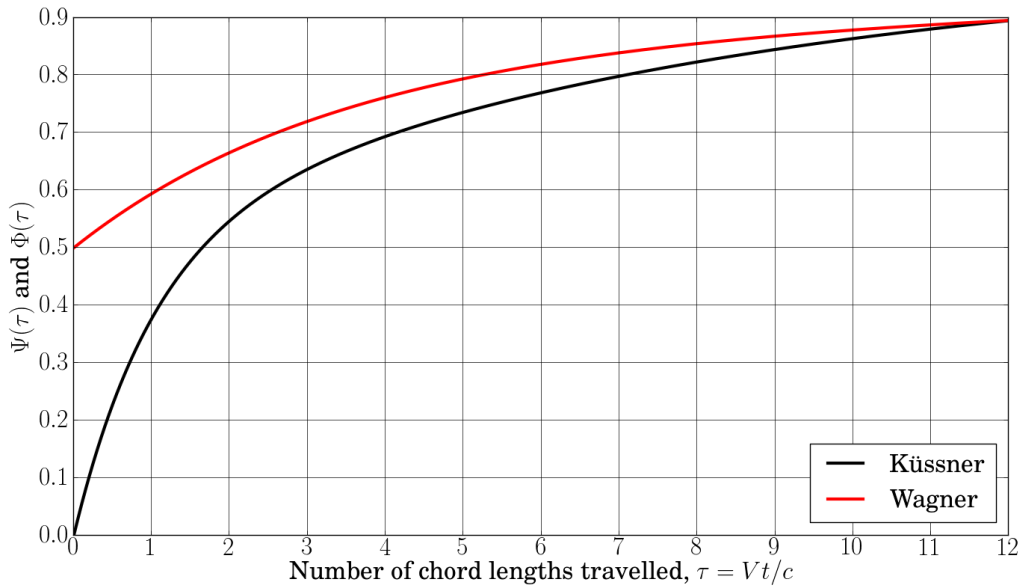


FIGURE 2.8: Küssner and Wagner functions.

The alleviation factor K_g also took into account the introduction of a revised gust shape. In fact, the latter was switched to the bell-shaped ‘ $1 - \cos$ ’ function (Figure 2.9) with $H = 12.5c$ [12]. Interestingly, the reason behind the choice of the ‘ $1 - \cos$ ’ shape appears to be connected with the view that no gust could start with a finite slope but, as argued by Houbolt [3], this is a notion irrelevant to the assumption that realistic gust encounters could be represented by discrete gusts or, in fact, by any shape.

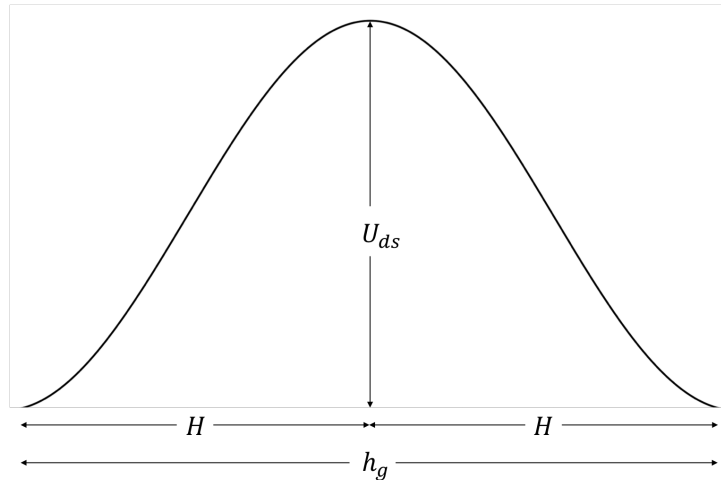


FIGURE 2.9: Typical bell-shaped ‘1 – cos’ gust required by the certification specifications for discrete gust calculations. U_{ds} is the maximum design gust velocity, H is the gust gradient and h_g is the total gust length.

Equation 2.8 was finally incorporated in Part 4B-3 of the U.S. Civil Air Regulations in the year 1957; this remarked the first step towards the development of more advanced techniques for the analysis of the aeroplane dynamic response [14]. A summary of the differences discussed so far between the original gust load formula and its revised version is given in Table 2.1, which opens the discussion to some important aspects. The first one of these is the difference between K and K_g and their representation as a function of, respectively, wing loading and mass parameter. Because K was not used in terms of mass parameter, a family of curves, each varying with mass, had to be used versus wing loading; however, the single curve shown in the table for K , normalised to unity at the wing loading of the Boeing 247, was the result of various engineering assumptions and was considered to be sufficient for aeroplane design and operations of that generation [12]. For modern aircraft design, instead, the single representation of K as a function of wing loading was no longer considered valid and was, thus, substituted with the revised alleviation factor K_g represented as a function of the mass parameter μ_g . Another relevant difference highlighted in the table was in the definition of the gust velocity. In fact, the original gust formula was function of the effective gust velocity U_e , which was only a fraction of the maximum equivalent velocity of the gust shape, whereas the derived gust velocity U_{de} , equal to the maximum equivalent velocity of the gust shape, was introduced in the revised gust formula.

In the early 1950s, just before the incorporation of the revised gust formula into the regulation requirements, the continuous nature of atmospheric turbulence was also recognised and

	Original	Revised
Gust load formula	$\Delta n = a\rho V U_e K / (2W/S)$	$\Delta n = a\rho V U_{de} K_g / (2W/S)$
Gust shape		
Gust gradient, H	10 chords	12.5 chords
Alleviation and gust factor		
Gust velocity	$U_e \propto U \sqrt{\frac{\rho}{\rho_0}}$	$U_{de} = U \sqrt{\frac{\rho}{\rho_0}}$

TABLE 2.1: Differences between original and revised gust formula. Adapted from Pratt and Walker, 1954 [12].

research studies were initiated by NACA to adapt power-spectral methods of generalised harmonic analysis (until then only used in the communications industry) to aircraft gust loads [17]. The research on power-spectral methods did not see its limits in the definition of atmospheric turbulence but also extended to the quantification of the dynamic elastic response of the aeroplane. Later on, studies performed by Lockheed and Boeing led, in 1966, to a complete definition of the procedures and criteria required to apply this method [14]. However, it was only in September 1980 that the FAA incorporated into the U.S. Federal Aviation Regulations (FAR) a criteria based on the power-spectral technique. In fact, before then, the only reference to power-spectral criteria for gust loads was the statement: “25.305 *Strength and deformation* [...] (d) *The dynamic response of the airplane to vertical and lateral continuous turbulence must be taken into account*” [2]. Long discussions and arguments between the FAA and the contracted agencies (Lockheed and Boeing) on the definition of the power-spectral scale factor U_σ were the main reason for the delayed incorporation of the continuous turbulence criteria in the FAR requirements [14].

2.3.2 Discrete Gust Design Criteria (CS 25.341a)

It is since the 1950s [14, 18] that the requirement set by the airworthiness authorities reads that an aeroplane in its design stage must be subject to symmetrical vertical and lateral gusts in level flight and that the shape of the gust shall be taken as

$$U = \frac{U_{ds}}{2} \left(1 - \cos \frac{\pi s}{H} \right) \quad \text{for} \quad 0 \leq s \leq 2H \quad (2.13)$$

which describes the typical ‘1 – cos’ gust shape defined mathematically in the space domain and assumed constant versus the time at a given point, as shown in Figure 2.9 [11, 19]; here, H is the gust gradient i.e. the distance parallel to the aeroplane’s flight path for the gust to reach its peak velocity, s is the distance penetrated into the gust and U_{ds} is the maximum design gust velocity specified as

$$U_{ds} = U_{ref} F_g \left(\frac{H}{107} \right)^{1/6} \quad (2.14)$$

where U_{ref} is a reference gust velocity¹ in EAS, F_g is the flight profile alleviation factor² (both are defined in CS-25 [11]) and 107 (if considering metres, otherwise 350 for feet) is the limit gust gradient that shall be used to satisfy the certification requirements. Equation 2.14 was derived by the Aircraft Industries Advisory Group (AECMA) in an effort to counter-propose a Notice of Proposed Amendment to JAR 25 entitled “A Unified Discrete Gust Requirement and Associated Means of Compliance” issued in May 1988 [14]. The AECMA counterproposal was accepted because their formula was shown to give the best fit to the load levels resulting from full profile analyses and for a range of different load quantities evaluated on several types of aircraft. Figure 2.10 shows the maximum design gust velocities over the gust penetration distance; up to 20 or more gust gradient distances H should be considered in the range 9.1 m to 107 m (30 to 350 ft) in sufficiently small increments to satisfy the discrete gust design condition. Shown in the same figure are also the results of Equation 2.14 as it can be seen that longer gusts have larger gust velocities. Besides, the certification specifications require for the

¹For an aircraft flying between the maximum speed for maximum gust intensity (V_B) and maximum cruise speed (V_C), the reference gust velocity can be reduced linearly from 17.07 m/s EAS at sea level to 13.41 m/s EAS at 4572 m and, consequently, further reduced linearly to 6.36 m/s EAS at 18288 m; for an aircraft flying at maximum dive speed (V_D), instead, the same rules apply but for halved speeds [11].

²The flight profile alleviation factor was introduced by the Federal Aviation Administration (FAA) to account for the variation of gust occurrences with altitude and for the proportion of time that an aircraft operates in a given altitude [14]. It increases linearly from a value at sea level, which is function of aircraft weight and maximum operating altitude, and 1 at the operating altitude and it has the aim of weighing the design gust speeds based on the flying altitude.

dynamic analysis to take into account unsteady aerodynamic characteristics and all significant structural degrees of freedom including rigid body motions.

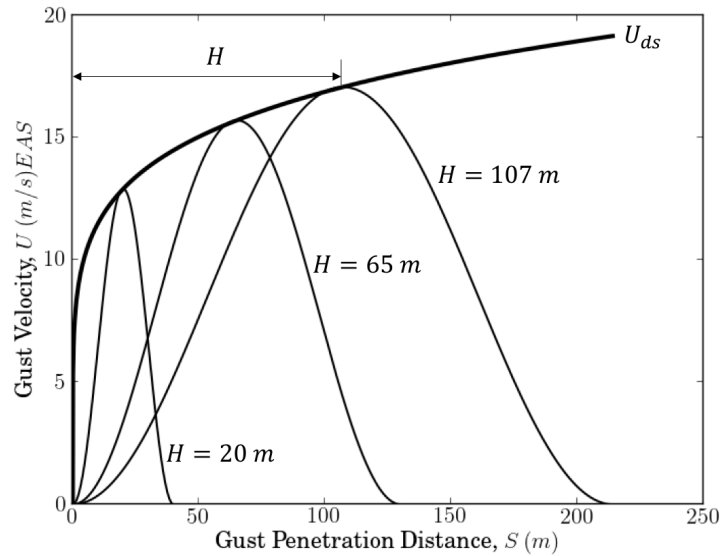


FIGURE 2.10: Typical ‘1 – cos’ gust shapes used for certification which form the design gust velocity envelope.

The design limit load, P_{Li} , for a given load quantity is then defined according to the following equation

$$P_{Li} = P_{(1g)i} \pm P_i \quad (2.15)$$

where $P_{(1g)i}$ is the 1g steady load for the considered load quantity indicated by the index i (e.g. bending moment) and P_i is the maximum incremental load defined by the peak values of the time histories that result from the series of separate ‘1 – cos’ gust profiles considered.

Within the framework of discrete gusts criteria, it is sometimes considered necessary to treat the loads arising from the combined action of vertical and lateral gusts. This analysis, not covered in this thesis, is known as *round-the-clock gust* and assumes that these loads can be obtained using a linear combination of the load time histories induced from pure vertical gusts and pure lateral gusts [2].

2.3.3 Continuous Turbulence Design Criteria (CS 25.341b)

Part (b) of the CS-25.341 regulation states that the dynamic response of the aeroplane to vertical and lateral *continuous turbulence* must be taken into account and that the limit loads must be determined for all the critical flight conditions (altitudes, weights and their distribution,

and speeds) specified in the certification requirements [11]. As for the discrete gust criteria, unsteady aerodynamic characteristics and all the significant degrees of freedom must be taken into account. The following equation must be used for the calculation of the limit loads due to continuous turbulence,

$$P_{Li} = P_{L(1g)i} \pm U_{\sigma} \bar{A}_i \quad (2.16)$$

where $P_{L(1g)i}$ is the steady 1g load for the condition, the index i indicates the type of load (e.g. bending moment), U_{σ} is the limit turbulence intensity in true airspeed ($U_{\sigma} = U_{\sigma_{ref}} F_g$ with $U_{\sigma_{ref}}$ equal to the reference turbulence intensity, which varies linearly with altitude), and \bar{A}_i is the ratio of root-mean-square incremental load for the condition to root-mean-square turbulence velocity. \bar{A}_i must be determined by the equation

$$\bar{A}_i = \sqrt{\int_0^{\infty} |H_i(\Omega)|^2 \Phi_{VK}(\Omega) d\Omega} \quad (2.17)$$

where $H_i(\Omega)$ is the frequency response function that describes the relation between the loads exerted on the structure and the external atmospheric turbulence and is obtained by dynamic analysis; $\Phi_{VK}(\Omega)$ is the normalised power spectral density³ (PSD) of atmospheric turbulence given by

$$\Phi_{VK}(\Omega) = \frac{\bar{L}}{\pi} \frac{1 + \frac{8}{3}(1.339\bar{L}\Omega)^2}{[1 + (1.339\bar{L}\Omega)^2]^{\frac{11}{6}}} \quad (2.18)$$

where $\Omega = \omega/V$ is the scaled frequency (rad/m) with V being the flight speed (m/s TAS), and \bar{L} is the characteristic scale wavelength of the turbulence (typically 762 m, but usually quoted in feet i.e. 2500 ft, to best fit experimental data) that dictates the variation of the PSD with frequency. Also known as von Kármán spectrum, Equation 2.18 is a mathematical model of atmospheric turbulence that best matches experimentally observed data [1, 2] and where the constants were derived by von Kármán following on the studies of Loitsianskii and Kolmogoroff on isotropic turbulent flow [21, 22].

³The power spectral density of a signal describes the power present in the signal as a function of frequency, per unit frequency [20]. The term “power” indicates that the quantity to which the various frequency components contribute is the mean square value of the variable; the term “spectral” indicates a measure of frequency content and the term “density” that the frequencies are continuously distributed rather than being discrete, or in other words that it is only possible to consider the contribution of a band of frequencies rather than a single frequency [2].

2.4 Gust loads methodology

The historical developments discussed in the previous section led to the two design criteria, CS 25.341 (a) and (b), described respectively in Section 2.3.2 and 2.3.3. This section and the following two sub-sections, instead, provide a description of the methods used in the present days to satisfy these requirements.

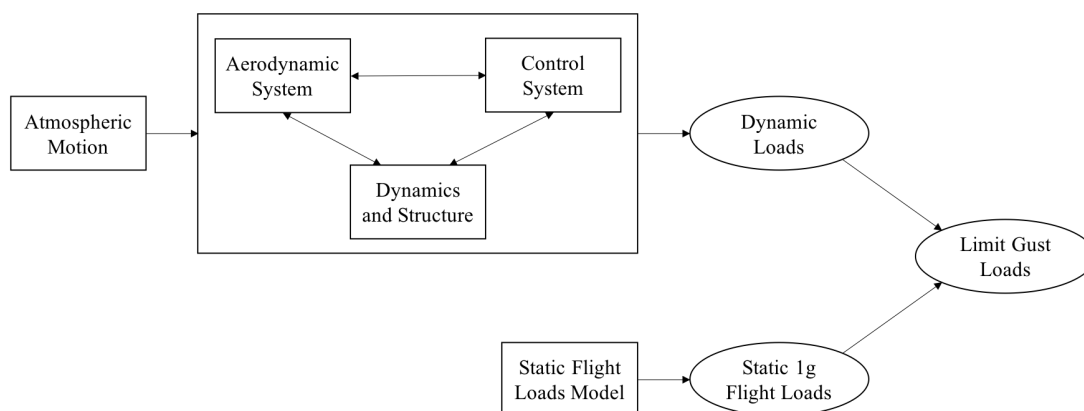


FIGURE 2.11: Basic system components for the analysis of gust response. Adapted from Certification Specification document (CS-25) [11].

The basic system components for the analysis of an aircraft responding to a gust are shown in Figure 2.11. The limit, or design, gust loads result from the combination of steady level 1g flight loads and dynamic loads, which include gust incremental loads. Static aeroelastic methods suffice for the analysis of steady 1g flight loads, which can be defined by considering basic external parameters such as speed, altitude, weight and fuel load. For discrete gust evaluations, a trimmed steady level flight is assumed as the initial condition of the aircraft; for continuous turbulence, instead, the trimmed steady level flight is assumed to be the mean flight condition. It is important to assure that this steady level flight condition is chosen in such a way that the maximum total load on each part of the aircraft is achieved and that all the possible effects originating, for example, from control surfaces or power settings are taken into account.

Of a more complex nature is the process that leads to the calculation of gust incremental loads, as this needs to account for the mutual interaction between the atmosphere, and hence atmospheric turbulence, and aeroplane rigid body and elastic motions. Both linear and nonlinear analysis methods may be part of this process. Linear analysis methods consist in the simple superimposition of the incremental loads to the steady level 1g loads and are of reasonable application when linear models can efficiently approximate the aeroplane and its flight control

system. Nonlinear analysis methods, instead, become necessary when these linear models cannot conservatively represent the aeroplane and its flight control system. This is the case, for example, when atmospheric turbulence of greater amplitude excite the nonlinear components of the system in a stronger and more evident form and, thus, imply their modelling. Examples of such components are the loads alleviation system, the stability augmentation system, the autopilot and other factors that affect the active control system of the aircraft such as non-proportional feedback gains, rate and amplitude limiters, changes in the control laws, hysteresis, hinge moment performance and saturation of aerodynamic control actuators. But, modelling nonlinear components increases the complexity of the system and often requires the analysis to be carried out in time domain, due to the inability of frequency domain methods to model nonlinearities (Chapter 4).

When the calculation of limit loads can be limited to linear analysis methods, several considerations must be taken into account for the successful modelling of the aeroplane. First of all, the structural dynamics model must represent all the rigid and flexible aeroplane degrees of freedom and should include a sufficient number of flexible modes⁴ to correctly reconstruct the response of the aeroplane to high frequency excitations. Furthermore, to complete the ‘Dynamics and Structure’ box of Figure 2.11, damping properties shall also be included in the model together with mass and stiffness properties. The ‘Aerodynamic System’ shall be able to compute the aerodynamic forces produced by both the gust velocity and by the aeroplane response and a complete analysis shall also include steady and unsteady aerodynamic effects at a range of reduced frequencies⁵. These include a delay that should be applied in the build-up of lift when the aeroplane enters the gust gradient. Moreover, all the interactions originating from the control systems components (‘Control System’) shall be included in the analysis and each component shall be considered in all its relevant modes of operation. Last but not least, discrete gust and continuous turbulence also induce gyroscopic loads on engines and auxiliary power units (APUs) which should be considered in the gust analysis. Finally, stability must be guaranteed by the dynamic model in both discrete and continuous gust conditions.

⁴The number of flexible modes must be sufficient to cover with satisfactory accuracy the frequency range of interest in the loads analysis.

⁵The reduced frequency is defined as $k = \frac{\omega c}{V}$ where ω is the angular frequency, c is the aerofoil chord and V is the aircraft speed; it is a parameter that indicates the degree of unsteadiness of the flow.

2.4.1 Discrete gust methods

A fairly simplistic approach that satisfies the certification requirements is described into a data item of the Engineering Science Data Unit (ESDU) technical series [15] and, nonetheless, detailed in one of the American Institute of Aeronautics and Astronautics (AIAA) Education Series books by Hoblit [2] and in many books of loads and aeroelasticity [e.g. 1, 23].

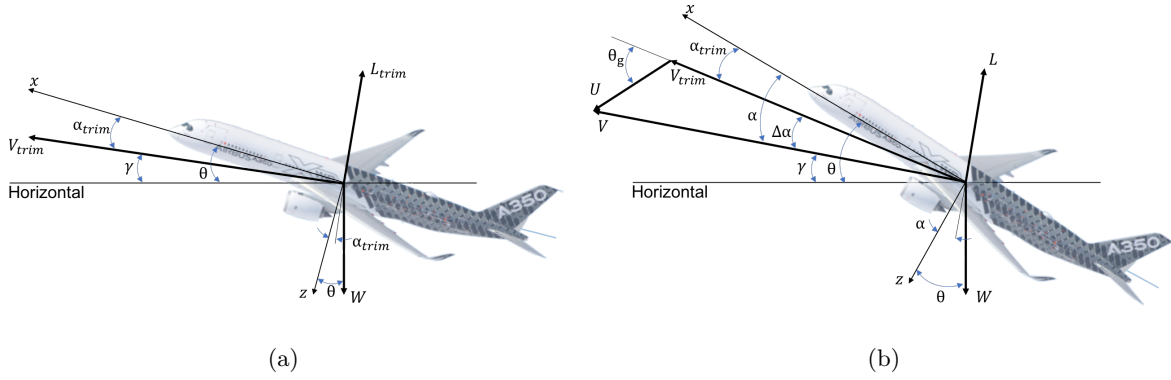


FIGURE 2.12: Aircraft in trimmed flight in undisturbed atmosphere (a) and during a gust encounter (b).

The proposed approach starts from the vertical response of a rigid aeroplane, which is initially in trimmed climbing flight conditions (Figure 2.12a) at flight path angle γ , pitch angle θ , with an angle of attack α_{trim} and flight velocity V_{trim} , and where the lift, L_{trim} , is given by

$$L_{trim} = \frac{1}{2} \rho V_{trim}^2 a S \alpha_{trim}. \quad (2.19)$$

When entering a simple discrete ‘1 – cos’ gust (Figure 2.12b), described mathematically as

$$U = \frac{U_0}{2} \left(1 - \cos \frac{\pi V}{H} t \right) \quad (2.20)$$

the increased lift, L , on the aeroplane can be written as

$$L = \frac{1}{2} \rho V^2 a S \alpha = \frac{1}{2} \rho V^2 a S (\alpha_{trim} + \Delta\alpha_g) \quad (2.21)$$

where V is the resulting velocity of the aeroplane during the gust encounter and $\Delta\alpha_g$ ($\approx \frac{U+\dot{z}}{V}$) is the angle-of-attack increment due to the gust. With the help of Newton’s second law, the

following differential equation of motion can be derived from Equation 2.21,

$$M\ddot{z} = -\frac{1}{2}\rho V^2 a S \left(\alpha_{trim} + \frac{U + \dot{z}}{V} \right). \quad (2.22)$$

The acceleration response above eventually leads to the acceleration increment defined in Equation 2.6 [15]. This approach then extends to include the delay in the build-up of lift and incidence by considering the following differential equation of motion

$$M\ddot{z} + \frac{1}{2}V^2 a S \int_0^t \Phi\{t - t^*\} \ddot{z}\{t^*\} dt^* = \frac{1}{2}V a S \int_0^t \Psi\{t - t^*\} \dot{U}\{t^*\} dt^* \quad (2.23)$$

where $\Psi(t)$ and $\Phi(t)$ are, respectively, the Küssner and Wagner functions introduced in the previous section. Because $\Psi(t)$ and $\Phi(t)$ are only a function of the number of chord lengths travelled, as represented in Figure 2.8, it is convenient to replace the time t by a non-dimensional time τ , which is exactly equal to the number of chord lengths travelled, i.e. $\tau = Vt/c$, and gives the relations

$$t = \frac{c\tau}{V}; \quad \frac{d}{dt} = \frac{V}{c} \frac{d}{d\tau}; \quad \frac{d^2}{dt^2} = \frac{V^2}{c^2} \frac{d^2}{d\tau^2} \quad (2.24)$$

Furthermore, τ is used to obtain an approximation of Küssner and Wagner functions, which are conveniently defined by the following exponential functions⁶

$$\Psi(\tau) \approx 1.0 - 0.5e^{-0.13\tau} - 0.5e^{-1.0\tau} \quad (2.25)$$

$$\Phi(\tau) \approx 1.0 - 0.165e^{-0.0455\tau} - 0.335e^{-0.3\tau} \quad (2.26)$$

Referring to Figure 2.8, it can be seen how the Küssner and Wagner functions build up, respectively, from 0 and 0.5 when the aircraft enters the gust and, then, asymptotically tend towards unity to account for the delay in the build up of lift given by the sudden change in velocity (Equation 2.25) and angle of attack (Equation 2.26). Equation 2.23 thus becomes in non-dimensional form

$$M \frac{V^2}{c^2} \frac{d^2 z}{d\tau^2} + \frac{1}{2} V^2 a S \int_0^\tau \Phi\{\tau - \tau^*\} \frac{V^2}{c^2} \frac{d^2 z}{d\tau^{*2}} \frac{c}{V} d\tau^* = \frac{1}{2} V a S \int_0^\tau \Psi\{\tau - \tau^*\} \frac{dU}{d\tau^*} \frac{c}{V} d\tau^* \quad (2.27)$$

and, after further manipulation, can be rewritten as

$$\frac{d^2 z}{d\tau^2} + \frac{1}{\mu_g} \int_0^\tau \Phi\{\tau - \tau^*\} \frac{d^2 z}{d\tau^{*2}} d\tau^* = \frac{1}{\mu_g} \frac{c U_{ds}}{V} \int_0^\tau \Psi\{\tau - \tau^*\} \frac{d}{d\tau^*} \left(\frac{U}{U_{ds}} \right) d\tau^* \quad (2.28)$$

⁶The exponential functions of equations 2.25 and 2.26 are two of the most used approximations for Küssner and Wagner functions and are, respectively, attributed to Sears and Sparks and to R. T. Jones [24].

where μ_g is the mass parameter defined in Equation 2.7 and $(\frac{U}{U_{ds}})$ can be obtained from Equation 2.13. The solution of Equation 2.28 can be found numerically for a series of values of μ_g and returns the non-dimensional time histories of $\frac{d^2z}{d\tau^2}$ and $\frac{dz}{d\tau}$. The incremental load factor can then be obtained by considering the maximum value of $\frac{d^2z}{d\tau^2}$ as follows

$$\Delta n = \frac{\ddot{z}}{g} = \frac{1}{g} \frac{V^2}{c^2} \frac{d^2z}{d\tau^2} = \frac{1}{g} \frac{V^2}{c^2} \frac{c U_{ds}}{V} \left(\frac{d^2z}{d\tau^2} \right)_{\frac{c U_{ds}}{V}=1} \quad (2.29)$$

If multiplied and divided by the mass parameter μ_g , Equation 2.29 then becomes

$$\Delta n = \underbrace{\mu_g \left(\frac{d^2z}{d\tau^2} \right)_{\frac{c U_{ds}}{V}=1}}_{K_g} \frac{\rho U_{ds} V a}{2W/S} \quad (2.30)$$

where the first two terms of the right hand side define the revised gust alleviation factor K_g , which can be approximated empirically by

$$K_g = \frac{0.88\mu_g}{5.3 + \mu_g}. \quad (2.31)$$

In order to consider the pitch response of the aircraft entering the gust, the rigid aircraft model employed so far needs to account for the extra degree of freedom required by the pitch motion θ ; in Figure 2.12 this is defined positive nose up and with respect to the centre of mass. Also, the effects of the tailplane must be introduced into the analysis. As a result, the resulting incremental lift forces are split into ΔL_W and ΔL_T that act, respectively, at the aerodynamic centres of the wing and of the tailplane

$$\Delta L_W = \frac{1}{2} \rho V^2 S_W a_W (\Delta \alpha_{gW}) \quad \text{and} \quad \Delta L_T = \frac{1}{2} \rho V^2 S_T a_T (\Delta \alpha_{gT}) \quad (2.32)$$

where $\Delta \alpha_{gW}$ and $\Delta \alpha_{gT}$ are, respectively, the increments in wing and tailplane incidence defined as

$$\Delta \alpha_{gW} = \frac{U(t) + \dot{z}_W}{V} + \theta \quad \text{and} \quad \Delta \alpha_{gT} = \frac{U(t - l/V) + \dot{z}_T}{V} + \theta \quad (2.33)$$

with $\dot{z}_W = \dot{z} - l_W \dot{\theta}$ being the heave velocity at the wing aerodynamic centre, and $\dot{z}_T = \dot{z} + l_T \dot{\theta}$ the heave velocity at the aerodynamic centre of the tailplane; l_W and l_T are, respectively, the distance between the wing and tail aerodynamic centres and the centre of mass of the aircraft; l/V is a time quantity given by the distance between the wing and tail aerodynamic centres divided by the aircraft speed. The incremental heave and pitch responses can be obtained from

the solution of the equations of motion of the aircraft, which are obtained from the combination of equations 2.32 and 2.33 with Newton's second law. In compact form, these can be written as the following system

$$\begin{aligned} & \begin{bmatrix} M & 0 \\ 0 & I_y \end{bmatrix} \begin{Bmatrix} \ddot{z} \\ \ddot{\theta} \end{Bmatrix} + \begin{bmatrix} -Z_{\dot{z}} & -Z_q \\ -M_{\dot{z}} & -M_q \end{bmatrix} \begin{Bmatrix} \dot{z} \\ \dot{\theta} \end{Bmatrix} \\ & + \begin{bmatrix} 0 & -Z_\alpha \\ 0 & -M_\alpha \end{bmatrix} \begin{Bmatrix} z_C \\ \theta \end{Bmatrix} = \begin{Bmatrix} Z_{gW} \\ M_{gW} \end{Bmatrix} U(t) + \begin{Bmatrix} Z_{gT} \\ M_{gT} \end{Bmatrix} U \left[t - \frac{l}{V} \right] \end{aligned} \quad (2.34)$$

where I_y is the pitch moment of inertia and where the longitudinal aerodynamic derivatives for the change of pitch θ , rate of change of pitch $\dot{\theta}$, rate of heave \dot{z} and gust-related derivatives (Appendix A) are introduced to simplify the equations. In this case, the response of the aircraft will be more elaborate due to the influence of the tailplane on the wings. In fact, as shown in Figure 2.13, in the scenario of a '1 - cos' gust, the aircraft initially pitches nose up and has a negative heave acceleration that pushes the aircraft upwards, due to the increased lift on the wings; when the gust then reaches the tailplane, the lift resulting on the latter causes the aircraft to pitch nose down and the heave acceleration becomes positive, pushing the aircraft downwards and leaving it under its free response, away from the forced motion induced by the gust.

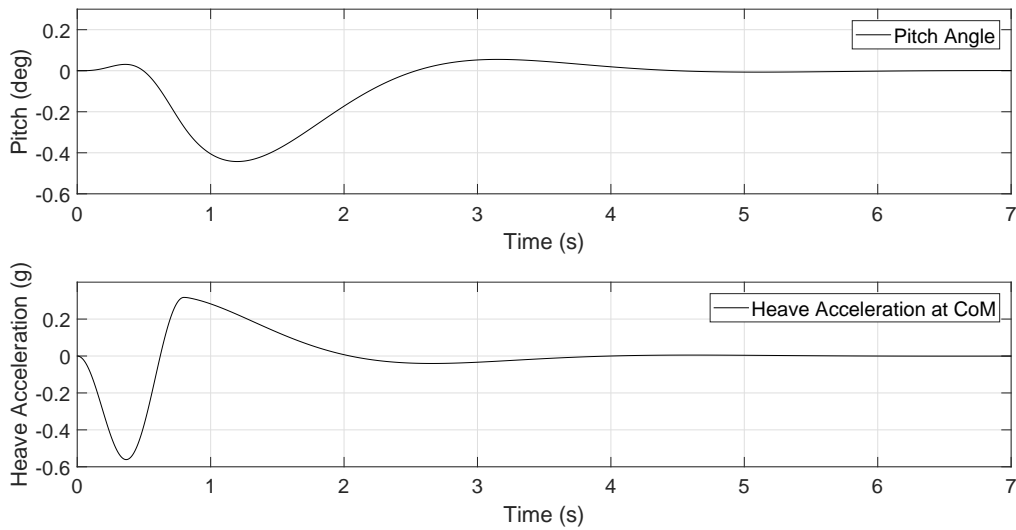


FIGURE 2.13: Heave (+ve downwards) and pitch (+ve nose up) response of a rigid aircraft with maximum take off weight of 10 tonnes flying at $V = 150$ m/s EAS at an altitude of 2,000 m and entering a '1 - cos' gust with gradient $H = 75$ m and maximum gust velocity $U_0 = 5.0$ m/s EAS. Source: companion software available from Wright and Cooper [1].

For consistency with Equation 2.1, the generalised form of the equations of motion discussed so far for the analysis of gusts in time domain is given below

$$\bar{\mathbf{M}}\ddot{\mathbf{q}} + (\rho V \bar{\mathbf{C}} + \mathbf{D})\dot{\mathbf{q}} + \rho V^2 \bar{\mathbf{E}}\mathbf{q} * \Phi + \bar{\mathbf{K}}\mathbf{q} = \rho V \mathbf{R}_{gW} U(t) * \Psi + \rho V \mathbf{R}_{gT} U\left(t - \frac{l}{V}\right) * \Psi \quad (2.35)$$

where $*$ marks the convolution integral with Küssner and Wagner (equations 2.11 and 2.12), which are required to take into account the unsteady aerodynamic effects, and \mathbf{R}_{gW} and \mathbf{R}_{gT} are gust-dependent aerodynamic vectors for the wing and the tailplane defined as

$$\mathbf{R}_{gW} = \begin{Bmatrix} Z_{gW} \\ M_{gW} \end{Bmatrix} \quad \text{and} \quad \mathbf{R}_{gT} = \begin{Bmatrix} Z_{gT} \\ M_{gT} \end{Bmatrix}. \quad (2.36)$$

2.4.2 Continuous turbulence methods

Concerning the response to a continuous turbulence, this is generally approached in terms of frequency response functions (FRFs) of the aeroplane and through the methods of power spectral density (PSD) of a random variable. In mathematical terms, the FRF (or transfer function) of a system is the Fourier transform of the impulse response function of the system, H_{Im} , which is expressed by

$$H(i\omega) = \int_0^{\infty} H_{Im}(t) e^{-i\omega t} dt. \quad (2.37)$$

where ω has units of rad/s. More generally, FRFs are implicit properties that describe how the system behaves under harmonic excitation at any frequency and can be represented by the relationship between the input and the output of the system, i.e.

$$\Phi_O(\omega) = |H(i\omega)|^2 \Phi_I(\omega) \quad (2.38)$$

where $\Phi_O(\omega)$ denotes the PSD function of the output from the linear system and $\Phi_I(\omega)$ the PSD function of a random input disturbance to a linear system. It thus becomes evident that if $\Phi_I(\omega)$ is the PSD function of a gust input and the FRFs of the rigid (or flexible) aeroplane have been determined, i.e. one per given load quantity (e.g. how the bending moment at a given wing station relates to the input gust velocity) at all frequencies of interest, Equation 2.38 provides the output PSD function for the given load quantity. As discussed in Section 2.3, the primary shape for the gust velocity PSD, required by the authorities and, thus, used for design purposes, is based on the von Kármán spectrum defined in Equation 2.18 and shown in Figure

2.14. Other gust PSD spectra also exist (e.g. Dryden) but the von Kármán PSD provides the best fit to observed data [1, 2].

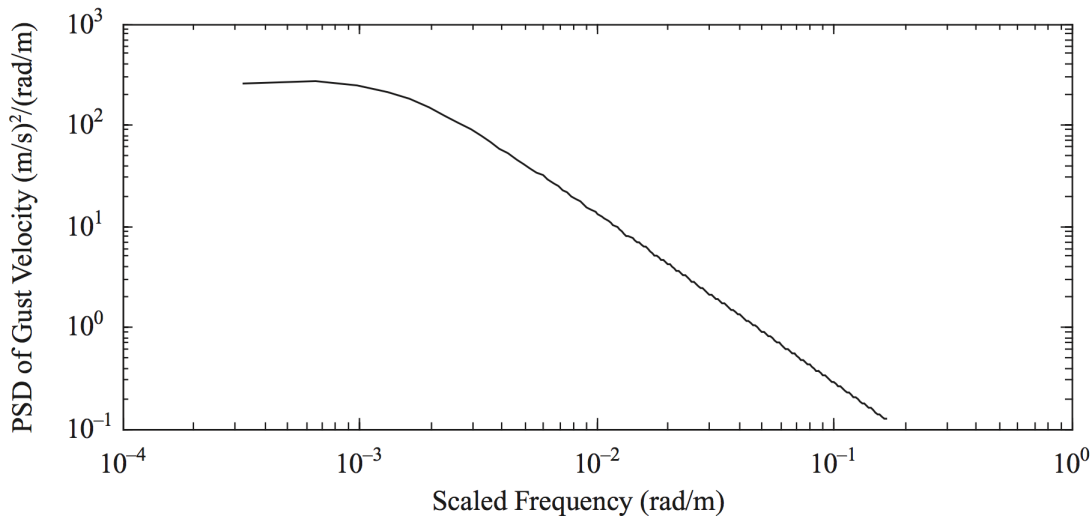


FIGURE 2.14: Von Kármán PSD function.

It can be appreciated from Figure 2.14 that the characteristic ‘knee’ of the von Kármán spectrum separates the lower frequencies from the higher ones. The position of the ‘knee’ is determined by the scale of turbulence \bar{L} . A lower scale of turbulence increases the frequency at which the ‘knee’ occurs and thus increases the range of frequencies for which the PSD is constant. But the desired solution is keeping the ‘knee’ at a frequency much lower than the frequency of the short period mode, which is the region where the gust frequencies are greatly amplified [15]. There is no such thing as a correct value for the scale of turbulence and, in fact, controversy still exists about its definition and meaning; however, 2500 feet has been agreed to be an appropriate value for design use because any values higher than this do not affect the higher frequencies at which gusts influence the response of the aeroplane. The gust (incremental) limit loads are thus determined with the equations 2.16, 2.17 and 2.18.

Another practical application of the output PSD is to understand how gust loads increase due to the introduction of structural flexibility, including elastic-mode dynamic effects. This can be determined by evaluating the root-mean-square value of the output PSD of interest of the rigid aeroplane and relate it to the corresponding output PSD of the flexible aeroplane [2].

2.5 The reconstruction of gust velocity profiles

As stated in Section 1.1 and at the beginning of this chapter, the reconstruction of gust velocity profiles marks the first objective of this work. Figure 2.11 in the previous section introduced the process required for the evaluation of gust loads. But, only a subset of those system's components are required for this objective and are shown in Figure 2.15. Starting from the aircraft gust response ("Dynamic Loads") and considering the main constituents of the "Aerodynamic System", the "Control System" and the aircraft "Dynamics and Structure", the aim is for the reconstruction of the contents of the "Atmospheric Motion" block.

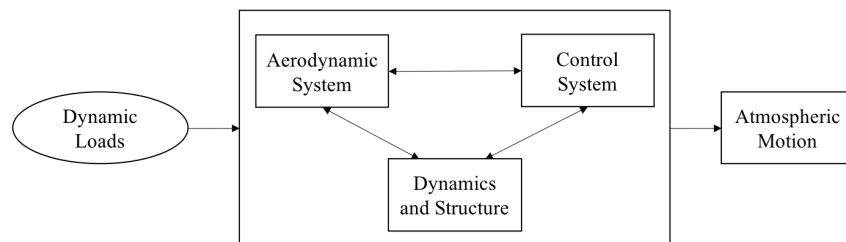


FIGURE 2.15: Basic system components for the reconstruction of gust velocity profiles.

The earliest approaches for the study of the gust velocity profiles [3, 25] considered in the 40s and 50s were based on the analysis of peak vertical accelerations Δn directly measured by the aircraft flying in gusts. These accelerations were assumed to be originated by a series of isolated discrete gusts and were used to derive gust gradient distances H and the maximum gust velocities U_{max} . The resulting gusts were then used to establish the loads that could be expected on a new design. This process was known as the discrete-gust approach and is shown in Figure 2.16 where Δn is assumed in the form of the revised gust formula (Equation 2.8). But, in 1961,

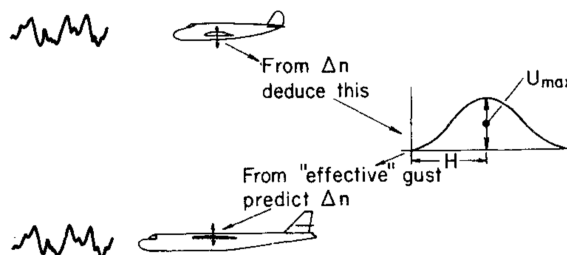


FIGURE 2.16: Discrete-gust approach to design. Source: Houbolt, 1973 [3].

J. K. Zbrozek argued that this approach was not able to return the real air turbulence and its application was not suitable for estimating the response of lightly damped modes [26]; however, it provided a model that could be still satisfactory for the evaluation of normal accelerations on

future designs.

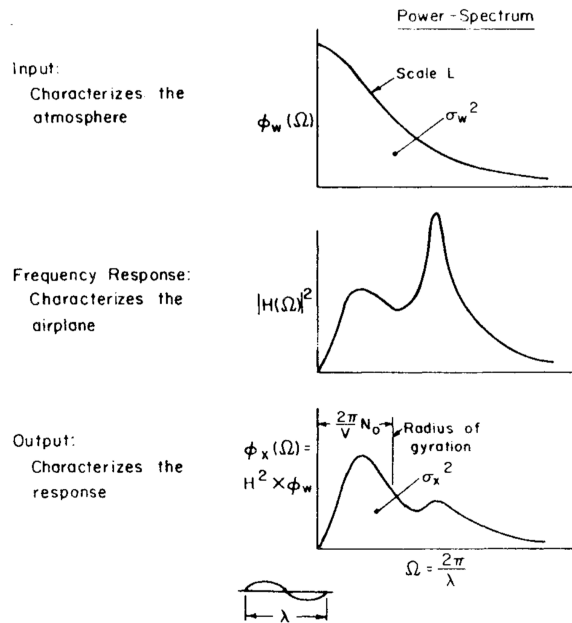


FIGURE 2.17: Spectral-technique: input-output relations for gust response. Source: Houbolt, 1970 [25].

Further efforts were made in the 60s and 70s on the development of spectral techniques for designing aircraft subject to gust encounters [3]. These techniques required two main assumptions: the time histories of the gust velocities obtained from the aircraft may be converted to space-fixed histories and the gust profiles were uniform in the spanwise direction. Figure 2.17 summarises the three-steps process. The spectrum for the gust velocity input Φ_w , a scale value L and a RMS gust severity value σ_w were the required parameters; the output response spectra could then be derived through the frequency response function \bar{H} . In turn, two structural parameters may be found: one to relate the RMS value σ_x of the chosen output parameter to the RMS value of gust severity σ_w , and the other to denote the frequency at which the output response crossed the mean $1g$ load level value of the chosen parameter x with the positive slope. However, the complexity in the definition of the frequency response functions \bar{H} marked a limitation for this technique, due to required considerations of the structural dynamics of the aircraft configuration combined to non-steady aerodynamics [3]. In 1999, in the context of space and missile systems, a Monte-Carlo flight gust loads analysis approach was proposed by Kim et al. [27]. The Monte-Carlo simulations were based on forcing functions derived from the extraction of the short-duration turbulent components of measured wind profiles. Gust loads were then extracted and the peak value of each load parameter of interest included in a

statistical analysis to establish the final load value with a 90% confidence level, as shown in the diagram of Figure 2.18. This, however, was only appropriate for gust profiles that consisted of wavelengths greater than 500 ft (152 m) [27].

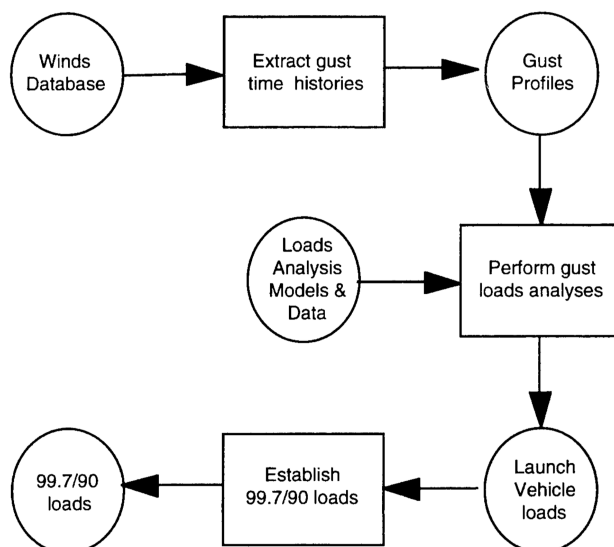


FIGURE 2.18: Monte-Carlo approach: statistical gust loads analysis. Source: Kim et al. [27].

Generally, over the years, force prediction has mainly been done by two methods: the *inverse* method and the *optimisation* method [28]. In inverse methods, the excitation $f(t)$ is calculated directly from the measured responses $a(t)$ by evaluating the inverse of the forward system model (Figure 2.19a). Optimisation methods, instead, use a forward model in an optimisation framework where the input to the forward model is tuned until the model responses match the measured responses (Figure 2.19b); the tuned input is then assumed to equal the original excitation. Although most force prediction methods are of the first type, a shift is ongoing from methods in the frequency domain towards methods in the time domain; this transition is due to the inability of inverse methods to capture very time limited events, which play a crucial role in exciting nonlinearities [29]. Parts of the acceptable means of compliance of the certification requirements also state that when the explicit simulation of nonlinearities is required, time domain methods shall be used in opposition to frequency domain methods as the influence of nonlinearities on one load quantity can greatly differ on another load quantity, thus causing their linearisation hard to achieve [11]. In 2009, Henrichfreise et al [30] proposed a method consisting of a model-based approach with an observer for a nonlinear aircraft model and a disturbance model for the estimation of gusts and structural loads, as shown in the diagram of Figure 2.20. This method uses aircraft motion measurements and parameters already available onboard modern commercial aircraft, thus making the estimation of purely manoeuvre-induced

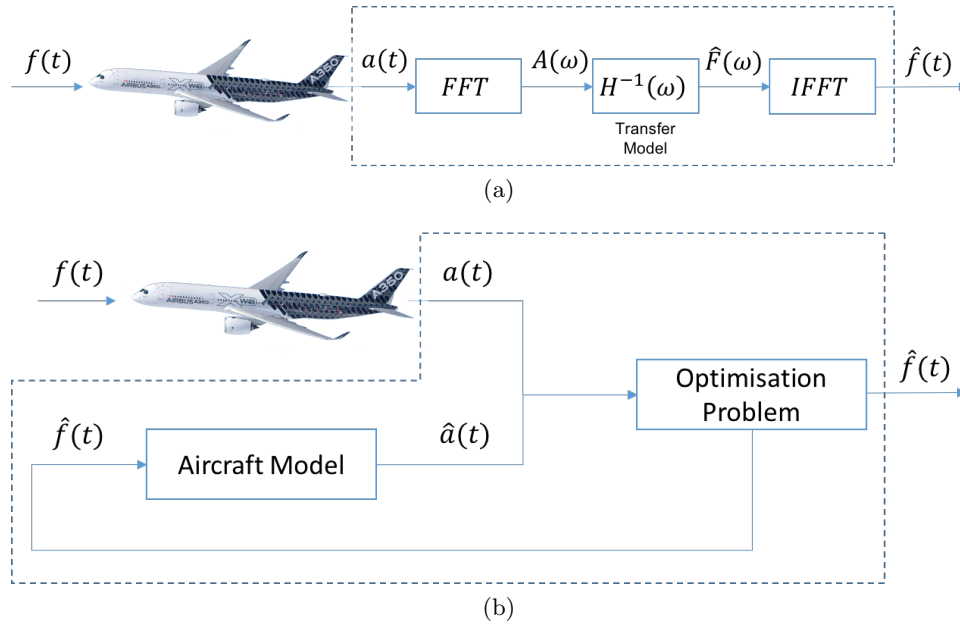


FIGURE 2.19: Force prediction via the inverse method (2.19a) and the optimisation method (2.19b).

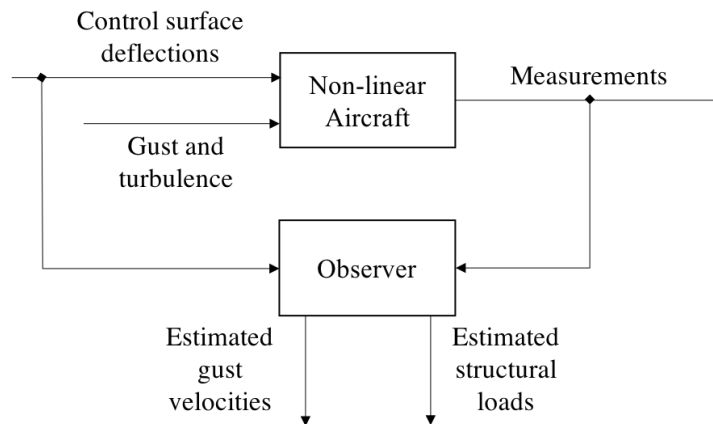


FIGURE 2.20: Basic approach for the estimation of gust velocities and structural loads. Adapted from: Henrichfreise, 2009 [30].

structural loads an easy to solve problem; the only unknowns remain the gust velocities which are determined through a nonlinear parameter optimisation that computes the gain matrix of the observer model. This method was recently validated on the flight test aircraft UW-9 Sprint⁷ at the Hamburg University of Technology [31]. The setup of the observer used for the validation is shown Figure 2.21, which is a more detailed version of Figure 2.20. Particularly, it is shown

⁷The flight test aircraft UW-9 Sprint is a high wing two seater ultra-light aircraft with a maximum takeoff weight of 472.5 kg and a maximum speed of 170 km/h. It is powered by a seven cylinder radial engine and has basic conventional controls (ailerons, elevator, rudder and thrust). Compared to other test aircraft, it has a high gust sensitivity due to low wing loading and it can withstand load factors up to $4.0g$ and $-1.5g$ at the maximum airspeed. Last but not least, it has low operating costs at a high degree of availability [31].

that a nonlinear aircraft model forms the nonlinear “Luenberger” observer, which includes a disturbance observer for the estimation of the unknown disturbances (gust velocities) and a state observer to support the simulated aircraft motion. When the real aircraft is excited by gust disturbances, these will be estimated by the disturbance observer through the computation of the difference between the measured and the simulated aircraft accelerations. In his work, Montel demonstrated and confirmed the functionality and suitability of the loads observer for the monitoring of maneuver and gust loads [31]. Today, this method currently represents the state of the art in industry, but to reduce the computational effort required for the convergence of the optimisation problem, only a limited set of elements selected from the gain matrix is considered for the observer feedback.

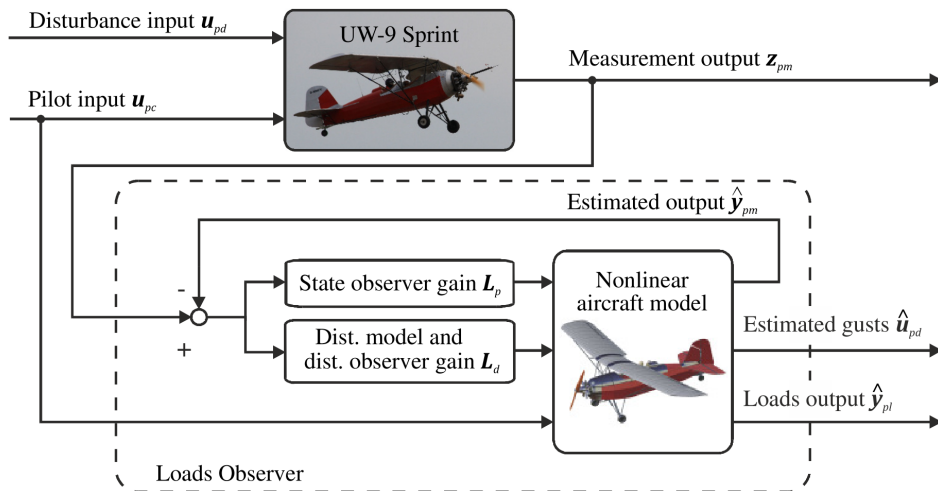


FIGURE 2.21: Schematic operation of the loads observer algorithm applied to UW-9 Sprint aircraft [31].

2.6 Conclusions

This chapter has provided an overview of the gust loads calculation process followed in industry and the historical developments that have led to today’s theory and methods. These form the background for the reconstruction of gust velocity profiles, whose historical development and needs have also been introduced here. The alternative gust reconstruction approach that is presented throughout this work aims to replace the observer block of Figure 2.20 with an optimisation routine where the objective function is designed around the measured and predicted states of the aircraft and the gust shape is either reconstructed continuously in time through gradient-free optimisation techniques or parameterised using Radial Basis Functions (RBFs) and

Hicks-Henne Bump Functions (HHBFs) and reconstructed through gradient-based optimisation techniques. It will be demonstrated that this approach may be applicable to the different design stages of the aircraft and for in-service applications, as it can be easily adapted to aeroplane models of various degrees of fidelity and complexity, i.e. from potential flow theory applied to a flat plate up to 3-dimensional models, where linear and nonlinear modelling techniques are key components of the reconstruction process and steer the successful reconstruction of the gust profile. In this view, the following chapter presents a description of the linear and nonlinear models that are representative of the different design stages of the aircraft and are, therefore, chosen for the reconstruction methods presented in this study.

Chapter 3

Aerodynamic and aeroelastic models

This chapter gives an overview of the aeroelastic models used throughout this work to assess the applicability of the reconstruction methods - presented in chapters 4 and 5 - at the different design stages of an aircraft. The first model described is of an airfoil that includes a moveable control surface at the trailing edge and nonlinear restoring forces. This is followed by an airfoil idealised as a flat plate, where potential flow theory with unsteady lumped vortices is considered. It is a model that shall be intended as a prototype for the application of the reconstruction method to models of significantly higher complexity. In fact, the use of computational fluid dynamics (CFD) for gust simulations is then introduced and employed for the airfoil of a typical modern commercial aircraft. The reduced order model (ROM) of the same airfoil then follows and, finally, the description of a typical three-dimensional beam-stick model used in industry.

3.1 Introduction

Different aerodynamic and aeroelastic models of increasing level of complexity have been used throughout the course of this work to cover the application of gust loads reconstruction during the different design stages of an aircraft. The two reconstruction methods proposed in this work (Section 5.2 and Section 5.3) are based, respectively, on gradient-free optimisation and gradient-based optimisation techniques. Based on the nature of their application and for compatibility reasons, the former has been applied to the following model:

- 2D airfoil with Theodorsen's aerodynamics and nonlinear geometric properties (§3.2),

and the latter to:

- flat plate with unsteady lumped vortex method (§3.3),
- 2D aerofoil with computational fluid dynamics (§3.4),
- 2D aerofoil modelled by the CFD-derived reduced order model (§3.4.1),
- 3D beam-stick model with the aerodynamics forces determined by the Doublet Lattice Method (§3.5).

Figure 3.1 represents the allocation of these methods across the different design stages of an aircraft. In sequence, the first model considers an aerofoil submerged into an incompressible two-dimensional flow, as per Theodorsen's aerodynamics, with the addition of a trailing edge flap and the possibility of modelling structural nonlinearities. The second model introduced is a flat plate in potential flow where only aerodynamic effects are considered. This method, similarly to the previous one, sees its application during the conceptual phase of the design of a new aircraft, when a concise and detailed model of the aircraft does not exist yet and expensive design analyses are out of scope. The flat plate is modelled using an unsteady lumped vortex method (ULVM) and the flow unsteadiness is validated against the Theodorsen theory. Wagner and Küssner problems are also validated to account for the delay in the building up of the lift when entering a "gusty" environment. Generally, this model can be regarded as a prototype for using higher fidelity CFD simulations and it is used here not only to prove the applicability of the reconstruction method to the conceptual phase of an aircraft design but also as an early indicator of the compatibility of this method with CFD. In fact, during later aircraft design stages, models of higher complexity are required to perform detailed aerodynamic and aeroelastic analyses at different flow conditions. Hence, the two-dimensional section of the typical wing of a commercial aircraft modelled in CFD is introduced and the response to gust events modelled using the split velocity method. CFD simulations, however, cannot be considered for in-service applications due to their high computational cost; nevertheless, high accuracy results must be retained for this type of analysis as they are fundamental for the safe operation of aircraft. For this reason, the reduced order model of the CFD wing typical section is also considered in this work. The scope of reduced order models (ROMs) in gust design applications is to look at rapid modelling of the aircraft response to gust and turbulence encounters, in order to extract the critical gust loads that are fundamental for sizing the aircraft. Finally, a three-dimensional beam stick model with lumped masses and the aerodynamic forces determined using the DLM is considered in

this work; this model is to be intended as representative of the industrial model used during the definition and production phases of the design of an aircraft.

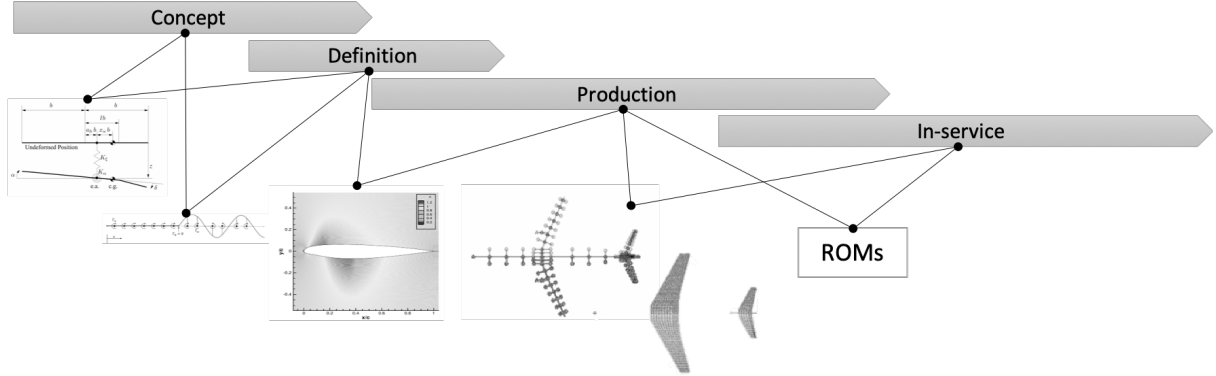


FIGURE 3.1: Representation of how the models used in this work fit with the different design stages of an aircraft. From left to right: 2DoF model, ULVM model, CFD model, 3D beam-stick with DLM and, finally, ROMs.

3.2 2DoF nonlinear model

Figure 3.2 shows an aerofoil with two degrees of freedom that define the motion about a reference elastic axis (*e.a.*). The plunge deflection is denoted by z , positive downward, and α is the angle of attack about the elastic axis, positive with nose up. The aerofoil is equipped with a massless trailing edge flap with hinge at a distance lb from the midchord. The flap deflection, δ , is defined relative to the undeflected position and not relative to the wind direction. The motion is restrained by two springs, K_ξ and K_α , and is assumed to have a horizontal equilibrium position at $z = \alpha = \delta = 0$. The system also contains structural damping in both degrees of freedom and the aerodynamics is given by modelling an incompressible two-dimensional flow as in Theodorsen [32, 33]. Nonlinear terms are included into the equations of motion to study the effects of structural nonlinearities. In dimensional form, these equations can be derived with nonlinear restoring forces in pitch and plunge using the Lagrange formulation [34]

$$m\ddot{z} + S_\alpha\ddot{\alpha} + C_\xi\dot{z} + K_\xi(z + \beta_\xi z^3 + \beta_{\xi_5} z^5) = -L \quad (3.1)$$

$$S_\alpha\ddot{z} + I_\alpha\ddot{\alpha} + C_\alpha\dot{\alpha} + K_\alpha(\alpha + \beta_\alpha\alpha^3 + \beta_{\alpha_5}\alpha^5) = M \quad (3.2)$$

with the structural nonlinearity approximated in a polynomial form obtained by means of Taylor series expansion [35]. The lift, L , is defined positive upward according to the usual sign convention in aerodynamics and the plunge displacement, z , is positive downward; M is the

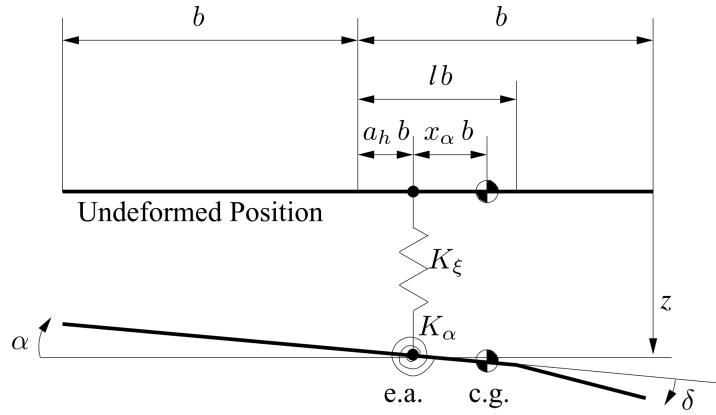


FIGURE 3.2: Aerofoil section with trailing edge flap.

pitch moment and should not be confused with the aircraft mass M defined in other sections of this thesis. In nondimensional form, the equations of motion become

$$\xi'' + x_\alpha \alpha'' + 2\zeta_\xi \frac{\bar{\omega}}{V^*} \xi' + \left(\frac{\bar{\omega}}{V^*} \right)^2 (\xi + \beta_\xi \xi^3 + \beta_{\xi_5} \xi^5) = -\frac{1}{\pi\mu} C_L(\tau_{\frac{1}{2}}) \quad (3.3)$$

$$\frac{x_\alpha}{r_a^2} \xi'' + \alpha'' + 2\zeta_\alpha \frac{1}{V^*} \alpha' + \left(\frac{1}{V^*} \right)^2 (\alpha + \beta_\alpha \alpha^3 + \beta_{\alpha_5} \alpha^5) = \frac{2}{\pi\mu r_a^2} C_m(\tau_{\frac{1}{2}}) \quad (3.4)$$

where nondimensional parameters are defined in the nomenclature. Here, differentiation with respect to t , indicated by $\dot{(\)}$, is replaced by a differentiation with respect to the non-dimensional parameter $\tau_{\frac{1}{2}}$ ¹, $\dot{(\)} = V/b(\)'$. The terms $C_L(\tau_{\frac{1}{2}})$ and $C_m(\tau_{\frac{1}{2}})$ represents the total aerodynamic loads, which are a sum of the contributions given by the aerofoil motion (*foil*), the flap deflection (δ) and the penetration into a gust (*gust*)

$$C_L(\tau_{\frac{1}{2}}) = C_L^{foil}(\tau_{\frac{1}{2}}) + C_L^\delta(\tau_{\frac{1}{2}}) + C_L^{gust}(\tau_{\frac{1}{2}}) \quad (3.5)$$

$$C_m(\tau_{\frac{1}{2}}) = C_m^{foil}(\tau_{\frac{1}{2}}) + C_m^\delta(\tau_{\frac{1}{2}}) + C_m^{gust}(\tau_{\frac{1}{2}}) \quad (3.6)$$

and are generalised into time histories by means of convolution integral, with Küssner and Wagner functions taken into account according to the exponential approximations introduced in equations 2.25 and 2.26. Equations 3.4 and 3.3 are therefore integro-differential equations (IDEs) that are difficult to solve analytically and need to be transformed into ordinary differential

¹The subscript $\frac{1}{2}$ is used here to indicate that the non-dimensional time τ is defined with respect to the semi-chord b and not the full chord c .

equations (ODEs) [33]. The equations of motion are then written in state-space form as

$$\frac{d\mathbf{w}}{d\tau_{\frac{1}{2}}} = \mathbf{R}(\mathbf{w}, \mathbf{u}_c, \mathbf{u}_d) \quad (3.7)$$

where \mathbf{R} is the nonlinear residual, \mathbf{w} is the n -dimensional state-space vector that includes α , ξ , α' , ξ' and the aerodynamic states, \mathbf{u}_c is the input vector that includes δ , δ' and δ'' , and \mathbf{u}_d is the vector that describes any external disturbance acting on the system² e.g. the gust $\bar{U}(\tau_{\frac{1}{2}})$ defined in Equation 3.8. Figures 3.3a and 3.3b show how the full-order model of the aerofoil is affected by the degree of structural nonlinearity. In this work the only nonlinear term considered is cubic stiffness in pitch; however, extensive studies on the effects of nondimensional parameters of higher order and in both degrees of freedom are available in the literature [33]. In order to demonstrate the nonlinear effect, the short-period response of the linear full-order model to a discrete gust of the ‘1 – cos’ type, as defined by certification requirements in Equation 2.13, is compared to the response of the nonlinear full-order model with cubic stiffness in pitch defined by the term $\beta_\alpha \alpha^3$ with $\beta_\alpha = 3.0$. The ‘1 – cos’ gust considered, defined as a function of the nondimensional parameter $\tau_{\frac{1}{2}}$ as

$$\bar{U}(\tau_{\frac{1}{2}}) = \frac{\bar{U}_0}{2} \left(1 - \cos \frac{2\pi}{\bar{H}} \tau_{\frac{1}{2}} \right) \quad \text{for} \quad \tau_{\frac{1}{2}} \leq 0 \leq \tau_{\frac{1}{2}} + \bar{H} \quad (3.8)$$

has a nondimensional gradient of $\bar{H} = 20.0$ and a nondimensional peak velocity of $\bar{U}_0 = 0.1$ (notice that this is not on scale in the figures). As the aerofoil enters the gust, the dynamic response of the full-order model with and without nonlinearities are practically equal; but large differences between the models can be observed some time after the gust moves away from the aerofoil. In fact, the nonzero β_α introduced in the nonlinear full-order model acts as an hardening spring that prevents the overshooting that verifies in the free response of the linear full-order model. Similar results were obtained by Khodaparast et al [36] when studying the influence of structural nonlinearities on worst-case gust load predictions and by Da Ronch et al [33] for the case of a deterministic sinusoidal gust where it was also observed that the effects of the structural nonlinearities in the dynamic response increased when increasing the intensity of the gust perturbation.

²The solution of the state-space system described in this chapter was obtained by means of differential equation solvers such as *ode45* or similar.

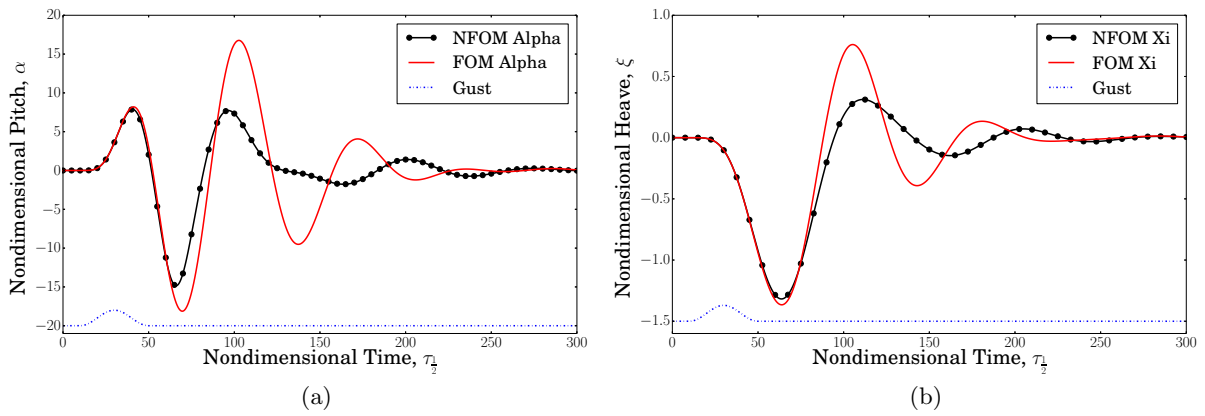


FIGURE 3.3: Comparison between linear full-order model (FOM) and nonlinear full-order model (NFOM) of pitch (3.3a) and heave (3.3b) responses to a ‘1 – cos’ gust.

3.3 Unsteady Lumped Vortex Method

Generally, the application of numerical techniques allows the treatment of realistic geometries and the fulfillment of the boundary conditions on the actual surface of the component considered. But, a quick and, yet, representative model with simple geometries is first considered for a preliminary assessment of gust and turbulence events. This consists of an idealised aerofoil, modelled with the unsteady lumped vortex method (ULVM). The ULVM is based on the surface distribution of singularity elements, namely lumped vortices, where the sought solution is the strengths of these elements. This approach is more economical from the computational point of view compared to those methods that solve for the flowfield in the whole fluid volume e.g. finite-difference methods, and yet can be considered as a valid prototype for CFD simulations. Figure 3.4 shows a representation of a two-component idealised aerofoil with two lumped vortices and two collocation points, where the boundary conditions are enforced. In addition, there is a wake model behind the trailing edge. The lumped vortices are located at the 1/4-chord point

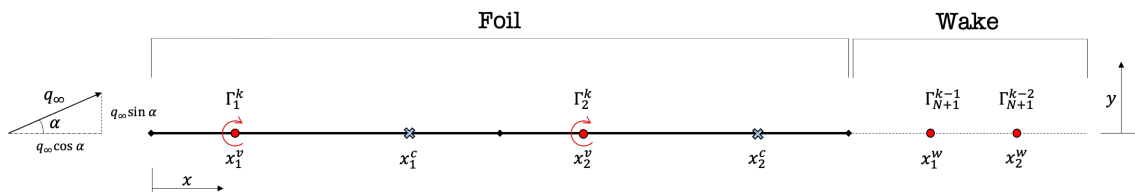


FIGURE 3.4: Representation of a two-component idealised aerofoil with vortices and control points, and with a wake. The superscript c indicates a collocation point, k the time level, v a vortex point and w a wake point.

(centre of pressure) of each component of the flat plate and the collocation points at the 3/4-chord of each component; this configuration matches thin aerofoil theory which satisfies the Kutta condition at the trailing edge of the flat plate [37]. The boundary condition requires that the normal velocity component must be zero at the collocation points. The net upwash (y -component of velocity) induced by all the aerofoil vortices at a generic control point m is given by

$$v_m = \sum_{n=1}^N \frac{\Gamma_n}{2\pi(x_n^v - x_m^c)} \quad (3.9)$$

where Γ_n is the circulation induced by the vortex n , x_n^v is the location of the vortex n and x_m^c is the location of the generic control point m . In order to satisfy the flow tangency condition, the net upwash must be balanced by the downwash (velocity in the negative y -direction) induced by the freestream ($q_\infty \sin \alpha$, with α being the direction of the freestream w.r.t. the flat plate) and the incident gust (U_k)

$$\sum_{n=1}^N \frac{\Gamma_n}{2\pi(x_n^v - x_m^c)} = -U_k - q_\infty \sin \alpha \quad (3.10)$$

Furthermore, Kelvin's condition states that the total circulation about the aerofoil and its shed vortex wake must remain constant; this translates mathematically to

$$\sum_{n=1}^{N+1} \Gamma_n(t) = \sum_{n=1}^N \Gamma_n(t - \Delta t) \quad \text{i.e.} \quad \frac{d\Gamma}{dt} = 0 \quad (3.11)$$

where Γ_{N+1} is the circulation of the shed vortex and $\sum_{n=1}^N \Gamma_n(t - \Delta t)$ is the total circulation on the aerofoil at the previous time step. This system of equations can be expressed in matrix form as $[\mathbf{A}]\{\mathbf{\Gamma}\} = \{\mathbf{b}\}$, with

$$[\mathbf{A}] = \frac{1}{2\pi} \begin{bmatrix} \frac{1}{x_1^v - x_1^c} & \frac{1}{x_2^v - x_1^c} & \cdots & \frac{1}{x_N^v - x_1^c} & \frac{1}{x_{N+1}^v - x_1^c} \\ \frac{1}{x_1^v - x_2^c} & \frac{1}{x_2^v - x_2^c} & \cdots & \frac{1}{x_N^v - x_2^c} & \frac{1}{x_{N+1}^v - x_2^c} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \frac{1}{x_1^v - x_N^c} & \frac{1}{x_2^v - x_N^c} & \cdots & \frac{1}{x_N^v - x_N^c} & \frac{1}{x_{N+1}^v - x_N^c} \\ 1 & 1 & \cdots & 1 & 1 \end{bmatrix}, \{\mathbf{b}\} = (-U_k - q_\infty \sin \alpha) \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} - \sum_{n=1}^N \Gamma_n(t - \Delta t) \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (3.12)$$

\mathbf{A} is often referred to as *influence coefficient* matrix. In order to take into account the upwash induced by the wake vortices and to include the wake vortices in the total circulation, the above system must be modified. In fact, Equation 3.12 is solved at the first time step only, when no

vortices have been shed in the wake; for the following time steps the influence coefficient matrix remains the same but the right hand side changes to

$$\{\mathbf{b}\} = \begin{Bmatrix} -q_\infty \sin \alpha - U_1^k - \sum_{i=1}^{k-1} \frac{\Gamma_{N+1}^{k-i}}{2\pi(x_i^w - x_1^c)} \\ -q_\infty \sin \alpha - U_2^k - \sum_{i=1}^{k-1} \frac{\Gamma_{N+1}^{k-i}}{2\pi(x_i^w - x_2^c)} \\ \vdots \\ -q_\infty \sin \alpha - U_N^k - \sum_{i=1}^{k-1} \frac{\Gamma_{N+1}^{k-i}}{2\pi(x_i^w - x_N^c)} \\ - \sum_{n=1}^N \Gamma_n(t - \Delta t) \end{Bmatrix} \quad (3.13)$$

Throughout this work, the spacing of the wake vortices is chosen to be the same as the spacing on the aerofoil i.e.

$$x_i^w = c(1 + i/N) \quad \text{for} \quad i = 1, 2, 3 \dots \quad (3.14)$$

where c is the foil chord. Also, $\Delta t = \Delta x/V_\infty$ so that the wake vortices are convected downstream at the freestream velocity, one position per time step. Once the strengths of the lumped vortices have been evaluated, the lift can be obtained. For the unsteady case, the Kutta-Jukowski law $L = \rho V_\infty \Gamma$ would not be sufficient to evaluate the total lift experienced by the aerofoil as it would give a zero lift for the initial time step of a transient solution. This is because at this step the lift is due to the acceleration of the flow rather than the net circulation about the aerofoil. Hence, the unsteady Bernoulli equation must be used to obtain the unsteady pressure and lift. As a result, it can be demonstrated [37] that the unsteady lift at each time step k is given by

$$L_k = \rho V_\infty \sum_{n=1}^{N+1} \Gamma_n^k + \rho \sum_{n=1}^N \left(\frac{d\Gamma_n^k}{dt} \right) \Delta x \quad (3.15)$$

From this equation it is clear that the first term accounts for the lift due to the steady flow (as expected from the Kutta-Jukowski theorem) and the second term accounts for the unsteady effects. The nondimensional lift coefficient is then

$$C_\ell = \frac{L}{\frac{1}{2}\rho V_\infty^2 c}. \quad (3.16)$$

3.3.1 Validation for the flow unsteadiness: Theodorsen's theory

The unsteady flow solver developed on the basis of equations 3.9 to 3.16 and used throughout this work has been validated against the principles of Theodorsen's theory, with separate solutions given for pure translation and pure harmonic pitching. In general, Theodorsen's approach

provides the solution to the unsteady airloads on a 2D harmonically oscillated aerofoil subject to small disturbances in inviscid, incompressible flow and with the assumption of planar wake. This is schematically represented in Figure 3.5 and described below.

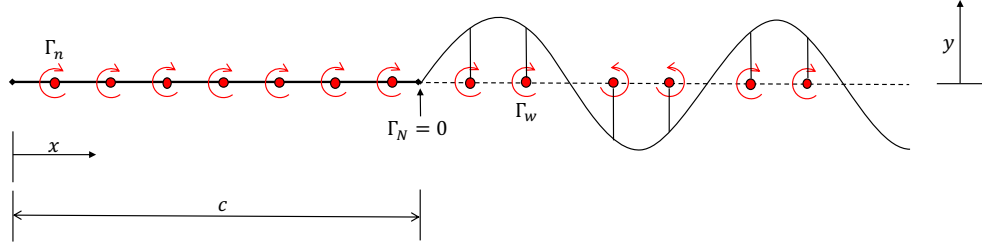


FIGURE 3.5: Representation of the harmonically oscillated foil used by Theodorsen.

For a general motion, where an aerofoil of chord $c = 2b$ is undergoing a combination of pitching (α) and plunging (z) motion in a flow of steady velocity V , using Theodorsen's theory the lift can be obtained as

$$L = \pi\rho V^2 b \left[\frac{b}{V^2} \ddot{z} + \frac{b}{V} \dot{\alpha} - \frac{b^2}{V^2} \bar{a} \ddot{\alpha} \right] + 2\pi\rho V^2 b \left[\frac{\dot{z}}{V} + \alpha + \frac{b\dot{\alpha}}{V} \left(\frac{1}{2} - \bar{a} \right) \right] C(k) \quad (3.17)$$

where \bar{a} is the pitch axis location relative to the mid-chord of the aerofoil and it is measured in terms of semi-chords [38]. The corresponding moment about the mid-chord is

$$M_{1/2} = -\rho b^2 \left[\pi \left(\frac{1}{2} - \bar{a} \right) V b \dot{\alpha} + \pi b^2 \left(\frac{1}{8} + \bar{a}^2 \right) \ddot{\alpha} - \bar{a} \pi b \ddot{z} \right] + 2\rho V b^2 \pi \left(\bar{a} + \frac{1}{2} \right) \left[V \alpha + \dot{z} + b \left(\frac{1}{2} - \bar{a} \right) \dot{\alpha} \right] C(k) \quad (3.18)$$

The first set of terms in Equation 3.17 and 3.18 are the non-circulatory terms, or the apparent mass effect³, that result from flow acceleration effects. The second term of both equations are instead the circulatory terms that arise from the creation of circulation about the aerofoil; the constituent term $C(k)$ is Theodorsen's function, which accounts for the effects of the shed wake on the unsteady airloads, and can be written as a complex valued transfer function

$$C(k) = F(k) + iG(k) = \frac{H_1^{(2)}(k)}{H_1^{(2)}(k) + iH_0^{(2)}(k)} \quad (3.19)$$

where $F(k)$ and $G(k)$ are, respectively, the real and imaginary parts and are expressed in terms of Hankel functions, i.e. $H_\nu^{(2)} = J_\nu - iY_\nu$ with J_ν and Y_ν being Bessel functions of the first and second kind⁴. Equation 3.19 is well-defined only in the case of purely sinusoidal aerofoil

³The apparent mass effect accounts for the pressure forces required to accelerate the fluid in the vicinity of the aerofoil.

⁴Bessel functions are solutions of the Bessel differential equation and are function of the reduced frequency $k = \frac{\omega b}{V}$. The subscript ν denotes the order of the function.

motion; however, several first or second order approximations of Theodorsen's function, derived empirically, exist in the literature [23, 39]. In simple terms, with respect to the lift response of an aerofoil obtained in quasi-steady conditions, Theodorsen's function introduces a reduction in amplitude and a phase lag effect on the circulatory part of the response [38].

According to a NACA technical note published by Halfman in 1951 [40], an efficient comparison of Theodorsen's theory with experimental results, which in this work reduce to the results of the ULVM, can be obtained by expressing Equations 3.17 and 3.18 in terms of complex notations and reducing them for the case of pure translational motion and pure pitching motion. Hence, the lift force resulting from a sinusoidally varying translational motion may be simply written as

$$L_T = 4q_\infty b(R_{LT} + iI_{LT})e^{i\omega t} \quad (3.20)$$

where the subscript T indicates the translational terms, q_∞ is the dynamic pressure and ω is the angular frequency of the forced motion; R_{LT} and I_{LT} are, respectively, the real and imaginary terms that apply only to the translational lift force. Equation 3.20 can also be expressed in terms of nondimensional derivative, i.e.

$$\frac{L_T}{4q_\infty b} = \sqrt{R_{LT}^2 + I_{LT}^2} e^{i(\omega t + \phi_{LT})} \quad (3.21)$$

where $\phi_{LT} = \tan^{-1} \frac{I_{LT}}{R_{LT}}$ is the phase lag. Similarly, the expression for the theoretical aerodynamic moment derivative in the translational mode may be written as

$$\frac{M_T}{4q_\infty b^2} = \sqrt{R_{MT}^2 + I_{MT}^2} e^{i(\omega t + \phi_{MT})} \quad (3.22)$$

with $\phi_{MT} = \tan^{-1} \frac{I_{MT}}{R_{MT}}$. The same equations (3.20-3.22) apply for the lift and moment due to pure pitching motion, with the only exception for the subscript T being replaced by the subscript P . Hence, the real and imaginary factors given by the theory [40], and deducible from the comparison of equations 3.17 and 3.18, for a two-dimensional wing are as follows

$$R_{LT} = \frac{\pi z_0}{b} \left(\frac{k^2}{2} + kG \right) \quad \text{and} \quad I_{LT} = -\frac{\pi z_0}{b} kF; \quad (3.23)$$

$$R_{LP} = -\pi\alpha_0 \left[\frac{\bar{a}k^2}{2} + F - \left(\frac{1}{2} - \bar{a} \right) kG \right] \quad \text{and} \quad I_{LP} = -\pi\alpha_0 \left[\frac{k}{2} + G - \left(\frac{1}{2} - \bar{a} \right) kF \right] \quad (3.24)$$

where F and G are, respectively, the real and imaginary terms of Equation 3.19 and k is the reduced frequency.

As a result, the theoretical (normalised) amplitude of pure translational lift in Equation 3.21 and its equivalent for pure pitching motion, namely $\sqrt{R_{LT}^2 + I_{LT}^2}$ and $\sqrt{R_{LP}^2 + I_{LP}^2}$, can be compared to the first harmonic⁵ extracted from the lift time histories obtained with the ULVM modelled with the equations 3.9-3.16 and run at a range of reduced frequencies.

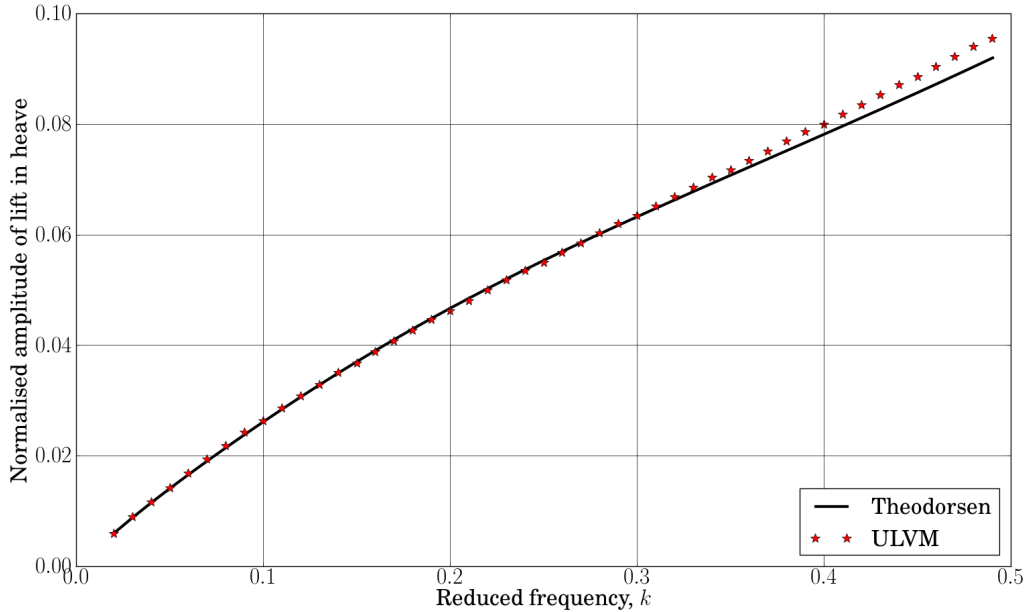


FIGURE 3.6: Validation against Theodorsen’s theory: aerofoil in pure translation. The black solid line represents the normalised lift in terms of Theodorsen’s function, with the real and imaginary terms defined in Equation 3.23 and using initial displacement $z_0 = 0.05$, semi-chord $b = 0.5$ and angle of attack $\alpha_0 = 0.0$. The red stars represent the first harmonic of the lift time history obtained with the ULVM with $\alpha = 0.0$, $V_\infty = 1.0$ and $c = 2b = 1.0$.

Figure 3.6 and Figure 3.7 show that a remarkably good agreement is obtained either in the case of pure translational motion and in the case of pure pitching motion. In both cases the theoretical results start to slightly diverge from the ULVM results at reduced frequencies higher than 0.35 i.e. as the flow becomes more unsteady. However, this is not an indication of a flaw in the ULVM. In fact, Theodorsen assumes a flat wake, whereas the ULVM does not, which may explain some of the variation seen here. Furthermore, the ULVM results may also be influenced by some discretisation error.

⁵The first harmonic of a lift time history obtained with the ULVM can be extracted by evaluating the Fourier transform of the time history and extracting the term with the maximum amplitude.

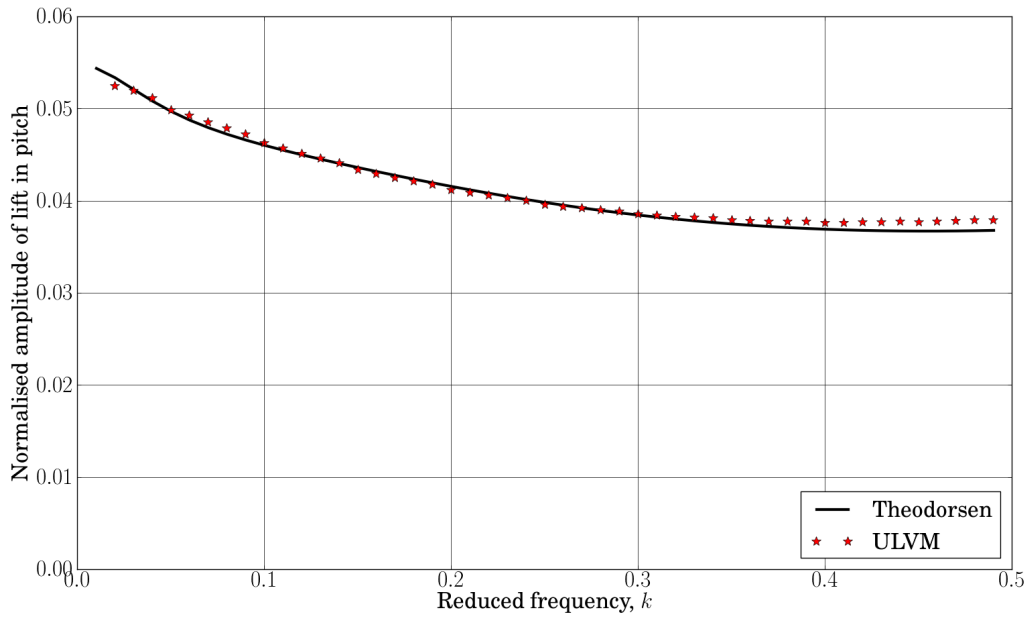


FIGURE 3.7: Validation against Theodorsen's theory: aerofoil in pure pitch. The black solid line represents the normalised lift in terms of Theodorsen's function, with the real and imaginary terms defined in Equation 3.24 and using initial displacement $z_0 = 0.0$, semi-chord $b = 0.5$, angle of attack $\alpha_0 = 1.0$ and $\bar{a} = 0.0$. The red stars represent the first harmonic of the lift time history obtained with the ULVM with $\alpha = 1.0$, $V_\infty = 1.0$ and $c = 2b = 1.0$.

3.3.2 Validation for the indicial response: Wagner's problem

As covered in sections 2.3 and 2.4, Wagner function takes into account the effects of the flow unsteadiness in the build-up of lift and can be regarded as the time-domain equivalent of Theodorsen function [38]. Specifically, it models the build-up of lift acting at the quarter-chord on the aerofoil following a step change of incidence by obtaining the effective downwash⁶ at the three-quarter chord point. Wagner function can be expressed in terms of indicial function; by definition, an indicial function is the response to a disturbance that is applied instantaneously at time zero and is held constant thereafter or, in other words, a disturbance given by a step function.

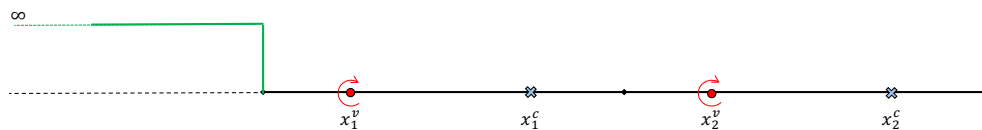


FIGURE 3.8: The Wagner's problem.

⁶The downwash is the velocity component normal to the flow.

Figure 3.8 gives a representation of the Wagner’s problem for the flat plate being modelled with the ULVM. In an incompressible flow, when a flat plate is subject to a transient step change in angle of attack, the transient chordwise pressure loading is given by

$$\frac{\Delta C_p(\bar{x}, \tau_{\frac{1}{2}})}{\alpha} = \frac{4}{V} \delta(\tau_{\frac{1}{2}}) \sqrt{(1-\bar{x})\bar{x}} + 4 \Phi(\tau_{\frac{1}{2}}) \sqrt{\frac{1-\bar{x}}{\bar{x}}} \quad (3.25)$$

where $\Phi(\tau_{\frac{1}{2}})$ is the Wagner function, which accounts for the effects of the shed wake, and $\tau_{\frac{1}{2}}$ is the distance travelled by the foil in semi-chords⁷. The first term in the RHS of Equation 3.25 is the apparent mass contribution, which for a step input is “controlled” by a Dirac-delta function $\delta(\tau_{\frac{1}{2}})$ ⁸; the second term, instead, is the circulatory term affected by the shed wake and, hence, “controlled” by the Wagner function. In Wagner’s problem, the aerodynamic centre is at mid-chord at $\tau_{\frac{1}{2}} = 0$ (i.e. when the distance travelled by the foil is 0) but it then moves at the quarter-chord for $\tau_{\frac{1}{2}} > 0$. The resulting variation in lift coefficient for a step change in angle of attack α can be written as

$$C_\ell(\tau_{\frac{1}{2}}) = \pi \delta(\tau_{\frac{1}{2}}) + 2\pi\alpha \Phi(\tau_{\frac{1}{2}}) \quad (3.26)$$

where the first term is the apparent mass and $2\pi\alpha$ is the steady-state lift coefficient, as given by steady thin-aerofoil theory, that multiplied by the Wagner function defines the circulatory terms. If the indicial response is known, then Duhamel’s integral [41] can be used to superimpose all the indicial aerodynamic responses to obtain the unsteady loads due to arbitrary changes in angle of attack; the lift coefficient would then become

$$C_\ell(\tau_{\frac{1}{2}}) = \pi\delta(\tau_{\frac{1}{2}}) + 2\pi \left(\alpha(0) \Phi(\tau_{\frac{1}{2}}) + \int_0^{\tau_{\frac{1}{2}}} \frac{d\alpha(\sigma)}{dt} \Phi(\tau_{\frac{1}{2}} - \sigma) d\sigma \right). \quad (3.27)$$

where $\alpha(0)$ is the initial angle of attack whilst σ defines the limits of the Duhamel integral, taking values between 0 and $\tau_{\frac{1}{2}}$. The difficulty in solving the Duhamel’s integral of Equation 3.27 is with the Wagner function itself as the latter is not expressed in a convenient analytical form. Therefore, for practicality, Wagner function is usually replaced by exponential approximations, as those defined in equations 2.25 and 2.26. Figure 3.9 shows that a good agreement is obtained between the lift coefficient of Equation 3.27, normalised by 2π and evaluated using the approximation of Equation 2.26, and the normalised lift coefficient obtained for the flat plate modelled with the ULVM. Despite the early evolution of the ULVM lift that may be influenced

⁷The subscript $\frac{1}{2}$ is used here to indicate that the non-dimensional time τ is defined with respect to the semi-chord b and not the full chord c .

⁸Mathematically, the Dirac-delta function is regarded as a “distribution” that is infinity at zero and zero in all other points. It is used as an operator to extract the value of a function at zero.

by discretisation effects, the results shown in Figure 3.9 validate the ULVM against Wagner's theory.

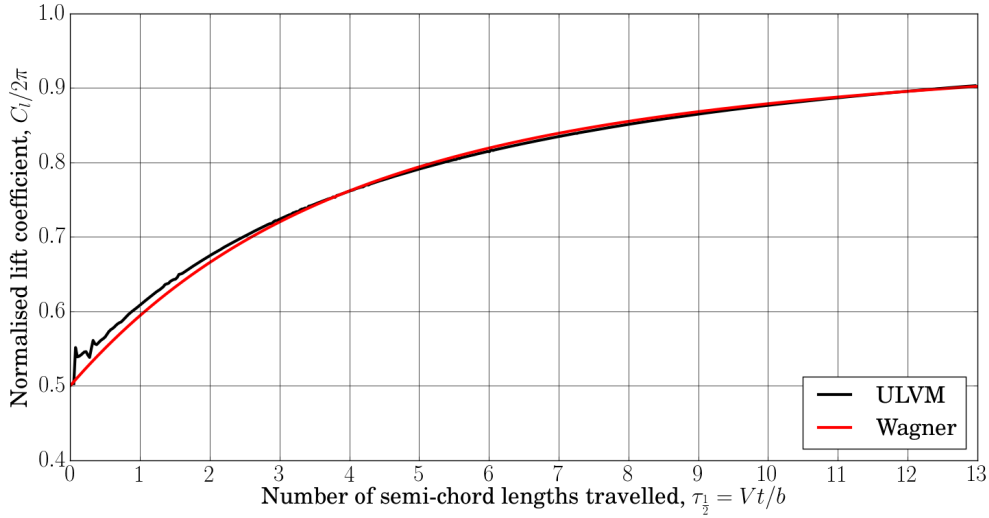


FIGURE 3.9: The Wagner's problem: indicial lift computed by ULVM code with $\alpha = 1.0$, $V_\infty = 1.0$ and $c = 2b = 1.0$.

3.3.3 Validation for a Sharp-Edged Gust: Küssner's problem

The problem of finding the transient lift response on a thin aerofoil entering a sharp-edged gust was first approached by Küssner in the early 1930s [16] and later solved by von Karman and Sears [42]. This problem is reproposed here for the flat plate considered in this work and represented in Figure 3.10.

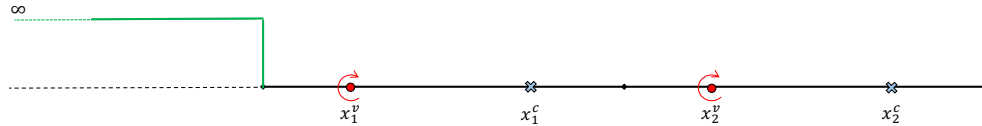


FIGURE 3.10: The Küssner's problem.

Here, the gust upwash velocity U relative to an axis at the leading edge is defined as

$$U = \begin{cases} 0 & \text{if } \bar{x} > Vt/c \\ U_0 & \text{if } \bar{x} < Vt/c \end{cases} \quad (3.28)$$

If the analysis is limited to the quasi-steady assumption, the incremental lift per unit span due to the change in angle of attack (i.e. $\tan \Delta\alpha \approx \Delta\alpha = \frac{U_0}{V}$) is added in its entirety as soon as the

aircraft enters the gust; in such case, the resulting lift is expressed by

$$L = \frac{1}{2} \rho V^2 c a \frac{U_0}{V} = \frac{1}{2} \rho V c a U_0. \quad (3.29)$$

But, in practice, the quasi-steady angle of attack changes progressively as the aerofoil penetrates into the gust front and a delay exists in the build-up of lift. Hence, and similarly to the Wagner's solution, the resulting variation in the lift coefficient can be written as

$$C_\ell(\tau_{\frac{1}{2}}) = 2\pi \left(\frac{U_0}{V} \right) \Psi(\tau_{\frac{1}{2}}) \quad (3.30)$$

where $\Psi(\tau_{\frac{1}{2}})$ is the Küssner function. The latter can be used with the Duhamel superposition integral to find the lift response to an arbitrary vertical upwash field, where the lift coefficient can be obtained using

$$C_\ell(\tau_{\frac{1}{2}}) = \frac{2\pi}{V} \left(U(0) \Psi(\tau_{\frac{1}{2}}) + \int_0^{\tau_{\frac{1}{2}}} \frac{dU(\sigma)}{dt} \Psi(\tau_{\frac{1}{2}} - \sigma) d\sigma \right). \quad (3.31)$$

As in Wagner's problem, an approximation for Küssner function (Equation 2.25) is used to obtain the theoretical results plotted in Figure 3.11, along with the results obtained with the ULVM. Despite showing a good agreement between the theoretical and the simulated results, which validates the ULVM also in this case, Figure 3.11 shows that the main property of the Küssner function is to apply a significant delay to the build-up of lift. In fact, the latter (normalised) starts from zero when the flat plate enters the gust and asymptotically tends towards unity.

3.4 The Split Velocity Method in CFD

The two approaches described in the previous sections do not allow for transonic effects, or other aerodynamic nonlinearities effects such as high angles of attack, to be captured during a gust encounter. This limitation makes the reconstruction of a gust profile impracticable for such cases and, indeed, a CFD analysis is required. Typically, the most direct way to introduce a gust into a CFD code is the modification of the boundary conditions at the far field of the computational domain. However, this application would require a very fine mesh to be defined all the way from the body to the far field in order not to dissipate the effects of the disturbances of vortices and gusts; as a result, the computational cost for such a simulation would be very high and its

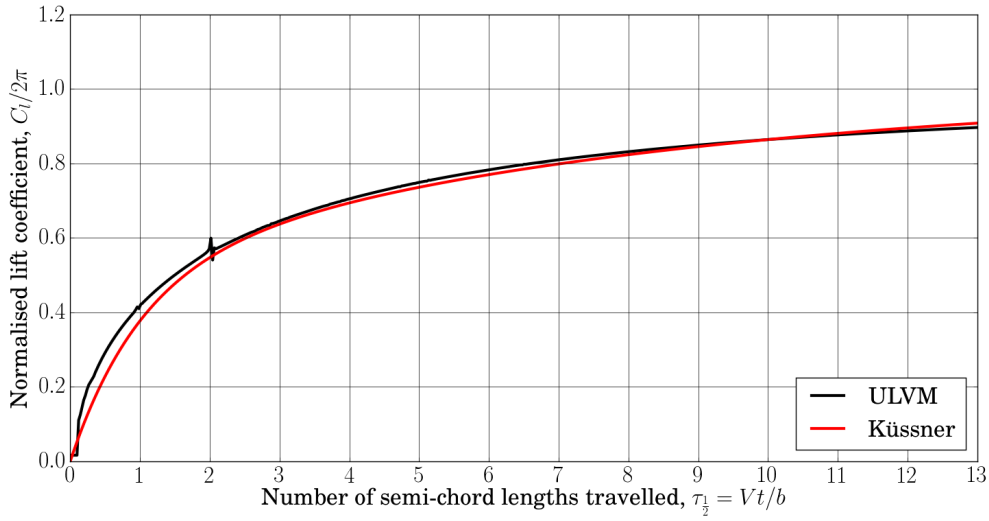


FIGURE 3.11: The Küssner's problem: indicial lift computed by ULVM code with $\alpha = 1.0$, $V_\infty = 1.0$ and $c = 2b = 1.0$.

usage very limited. In order to overcome this issue, Wales et al [43] recently proposed a solution based on the decomposition of the velocity components of the Euler equations into a prescribed gust velocity term (u_g) and the remaining velocity components (\tilde{u}). Known as the Split Velocity Method (SVM), this method is able to capture the full interaction between the body and the gust as no simplifying assumptions are made in the solution of the Euler equations. Clearly, the computational cost does not increase as the original mesh can be retained, because the gust and its prescribed motion will not be affected by dissipation effects. Hence, starting from the unsteady two-dimensional (2D) version of the Euler equations, which can be rewritten as

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho u \\ \rho v \\ \rho E \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ (\rho E + p)u \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho v \\ \rho uv \\ \rho v^2 + p \\ (\rho E + p)v \end{bmatrix} = 0 \quad (3.32)$$

with

$$p = \rho(\gamma - 1) \left[E - \frac{1}{2}(u^2 + v^2) \right], \quad (3.33)$$

the velocity and energy can be decomposed as

$$u = \tilde{u} + u_g \quad v = \tilde{v} + v_g \quad E = \tilde{E} + E_g \quad (3.34)$$

where u_g and v_g are the prescribed gust velocity components, and the split of the total energy is defined by the decomposition

$$\begin{aligned} E &= \frac{p}{\rho(\gamma - 1)} + \frac{1}{2}(u^2 + v^2) \\ &= \underbrace{\frac{p}{\rho(\gamma - 1)} + \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2)}_{\tilde{E}} + \underbrace{(\tilde{u}u_g + \tilde{v}v_g) + \frac{1}{2}(u_g^2 + v_g^2)}_{E_g}. \end{aligned} \quad (3.35)$$

Here, γ is the ratio of specific heats and should not be confused with the flight path angle γ defined in other sections of this thesis. By substituting the terms defined in Equation 3.34 into equations 3.32 and 3.33 and with further algebra manipulation - where the applied gust is separated from the rest of the solution - it can be shown [43] that the Euler equations can be rewritten as

$$\frac{\partial}{\partial t} \begin{bmatrix} \rho \\ \rho\tilde{u} \\ \rho\tilde{v} \\ \rho\tilde{E} \end{bmatrix} + \frac{\partial}{\partial x} \begin{bmatrix} \rho(\tilde{u} + u_g) \\ \rho\tilde{u}(\tilde{u} + u_g) + p \\ \rho\tilde{v}(\tilde{u} + u_g) \\ \rho\tilde{E}(\tilde{u} + u_g) + p\tilde{u} \end{bmatrix} + \frac{\partial}{\partial y} \begin{bmatrix} \rho(\tilde{v} + v_g) \\ \rho\tilde{u}(\tilde{v} + v_g) \\ \rho\tilde{v}(\tilde{v} + v_g) + p \\ \rho\tilde{E}(\tilde{v} + v_g) + p\tilde{v} \end{bmatrix} + \begin{bmatrix} 0 \\ s_m(u_g) \\ s_m(v_g) \\ s_e(u_g, v_g) \end{bmatrix} = 0 \quad (3.36)$$

where

$$p = \rho(\gamma - 1) \left[\tilde{E} - \frac{1}{2}(\tilde{u}^2 + \tilde{v}^2) \right], \quad (3.37)$$

and the source terms are

$$s_m(u_g) = \rho \left[\frac{\partial u_g}{\partial t} + (\tilde{u} + u_g) \frac{\partial u_g}{\partial x} + (\tilde{v} + v_g) \frac{\partial u_g}{\partial y} \right] \quad (3.38)$$

$$s_m(v_g) = \rho \left[\frac{\partial v_g}{\partial t} + (\tilde{u} + u_g) \frac{\partial v_g}{\partial x} + (\tilde{v} + v_g) \frac{\partial v_g}{\partial y} \right] \quad (3.39)$$

$$s_e(u_g, v_g) = \tilde{u}s_m(u_g) + \tilde{v}s_m(v_g) + p \left(\frac{\partial u_g}{\partial x} + \frac{\partial v_g}{\partial y} \right). \quad (3.40)$$

Equations 3.36 are solved in integral form on a fixed mesh such that

$$\frac{d}{dt} \iint_{\Omega} \mathbf{W} \, dx \, dy + \int_{\partial\Omega} (\mathbf{F} \, dy - \mathbf{G} \, dx) + \iint_{\Omega} \mathbf{S} \, dx \, dy = 0 \quad (3.41)$$

where

$$\begin{aligned}
 \mathbf{W} &= [\rho, \rho\tilde{u}, \rho\tilde{v}, \rho\tilde{E}]^T \\
 \mathbf{F} &= [\rho(\tilde{u} + u_g), \rho\tilde{u}(\tilde{u} + u_g) + p, \rho\tilde{v}(\tilde{u} + u_g), \rho\tilde{E}(\tilde{u} + u_g) + p\tilde{u}] \\
 \mathbf{G} &= [\rho(\tilde{v} + v_g), \rho\tilde{u}(\tilde{v} + v_g), \rho\tilde{v}(\tilde{v} + v_g) + p, \rho\tilde{E}(\tilde{v} + v_g) + p\tilde{v}] \\
 \mathbf{S} &= [0, s_m(u_g), s_m(v_g), s_e(u_g, v_g)]
 \end{aligned} \tag{3.42}$$

However, the SVM equations (3.37, 3.41 and 3.42) can be solved on a moving mesh code by adding the source terms to the moving grid equations and setting the grid velocities equal to the negative gust velocities. Indeed, Wales et al used a modified version of a moving mesh Euler code to demonstrate the efficacy of their method [43]; also, the exact same solver was used for this study. The aerofoil model used here is shown in Figure 3.12 and represent the typical section of a modern commercial aircraft [44]. Figures 3.13a and 3.13b show, respectively, a snapshot of the forward velocity contour plot at zero angle of attack in steady conditions and during a gust encounter.

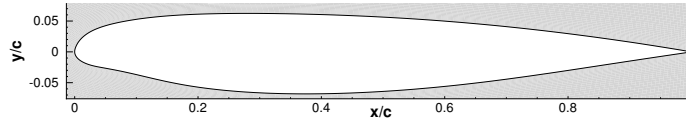


FIGURE 3.12: FFAST crank aerofoil.

3.4.1 Reduced-order model for CFD

If on one hand CFD analyses are required to capture the flow behaviour in (almost) its entirety, on the other hand savings on computational costs are essential during the early stages of the design of an aircraft. Williams et al [45] recently demonstrated that a very efficient reduced-order model can be achieved without losing the accuracy of the results obtained with full-order simulations. Undeniably, the rapidity and maintained accuracy of reduced-order models represents an invaluable characteristic for the gust reconstruction framework.

The CFD reduced-order model used within this work is built from a single sharp-edged gust, of magnitude $1m/s$, alongside a small number of steady simulations of various angles of attack. An effective step-down response can be produced by subtracting the sharp-edged gust response from the steady data (with the same angle of attack as the sharp-edged gust starts with) [46]. This step-down response is then used within an Eigensystem Realisation Algorithm (ERA) to

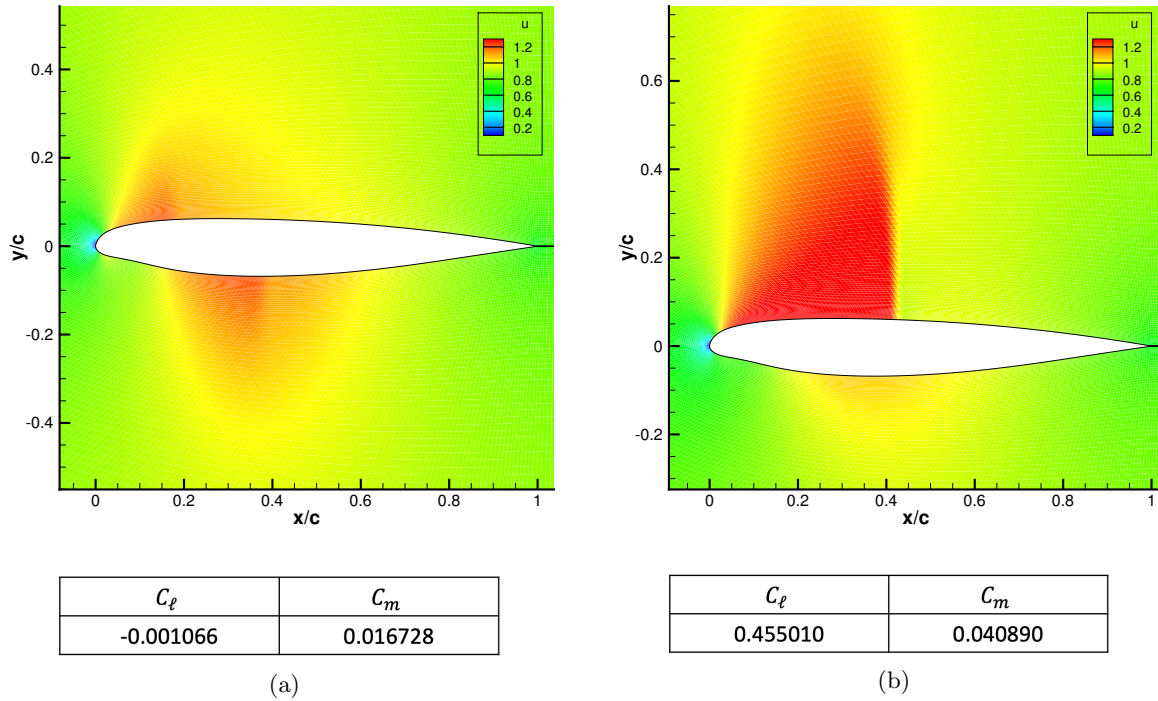


FIGURE 3.13: Forward velocity contour plot of FFAST aerofoil at zero angle of attack in steady conditions (a) and during ‘ $1 - \cos$ ’ gust encounter (b) with a width of 15.48 chords and magnitude equivalent to $\Delta\alpha_g = 3.58$ deg at $M = 0.73466$ and $Re \approx 1.43 \times 10^8$.

perform a system reduction to calculate the system matrices of a reduced order, discrete state-space model [47, 48], which can be used to express the near-linear behaviour of the given system. To ensure the ROM is stable (all eigenvalues have a magnitude of less than 1) restarting [49] and/or Schur decomposition [50] can be used. Once the system matrices have been calculated, they are valid for any gust of given Mach number and Reynolds number. However, they can also be extended to calculate the system response at any altitude. The computational cost of calculating a gust response once the matrices have been calculated is negligible. For a more detailed description of the reduced-order model, see Williams et al [45, 46].

3.5 3D beam-stick model

During the latest stages of the design of an aircraft, aerodynamic loads are typically predicted using a generalized frequency domain formulation of the aerodynamic force, coupled with structural FEM models. Figure 3.14 shows the full aeroelastic model used within this work to cover

for these applications. This model is based on the academic FFAST⁹ model, which is a close representation of a modern civil jet aircraft [51].

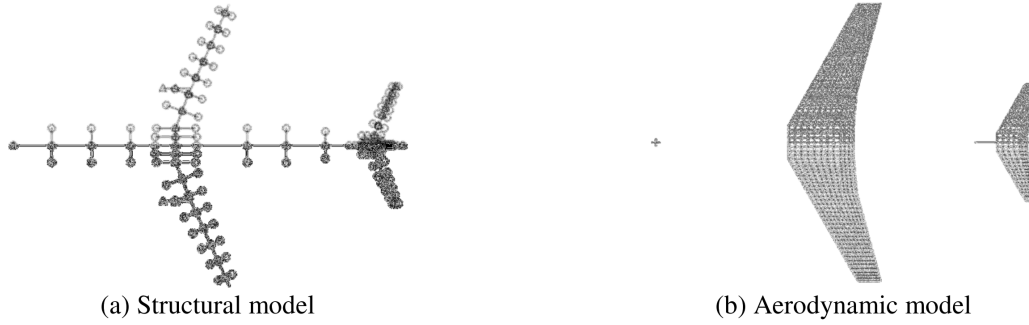


FIGURE 3.14: FFAST beam stick model and DLM.

The structure is modelled using a MDOF beam-element model with lumped masses, and the aerodynamic forces determined using the Doublet Lattice Method (DLM). Traditionally, the DLM, often corrected with CFD or wind tunnel test results, is used to evaluate the Generalized Aerodynamic Forces (GAF) matrices. These are linear aeroelastic analyses that are typically solved in the frequency domain, where the fundamental equation is

$$\left(-\omega^2 \bar{\mathbf{M}} + i\omega \bar{\mathbf{D}} + \bar{\mathbf{K}} \right) \mathbf{q} = \bar{\mathbf{Q}}_e + \bar{\mathbf{F}}_{Aero} \quad (3.43)$$

and where $\bar{\mathbf{M}}$, $\bar{\mathbf{D}}$, $\bar{\mathbf{K}}$ are, respectively, the generalized mass, damping and stiffness matrices; $\bar{\mathbf{Q}}_e$ collects the Fourier Transform of the non-aerodynamic external generalized forces while $\bar{\mathbf{F}}_{Aero}$ are the Fourier Transform of the generalized aeroelastic forces. The aeroelastic problem is formulated by considering as unknown the structural physical displacements \mathbf{u}_a expressed on a modal basis as

$$\mathbf{u}_a(x, t) = \sum_{h=1}^{N_{Modes}} \Phi_{ah}(x) \mathbf{q}_h(t) \quad (3.44)$$

where Φ_{ah} is the modal matrix, \mathbf{q}_h the modal displacements and the subscripts a and h represent, respectively, the set of the structural nodal and modal degrees of freedom. It is important to remark that Equation 3.44 introduces a significant approximation to the aeroelastic system since

⁹Future Fast Aeroelastic Simulation Technologies (FFAST) is a collaborative research project where the aim is the development, implementation and assessment of a range of numerical simulation techniques that shall accelerate future aircraft design. The efficiency and accuracy of the dynamic aeroelastic loads process shall benefit from advances in critical load identification and ROM techniques. The partners in FFAST are the University of Bristol, Institut National de Recherche en Informatique et en Automatique, the Council for Scientific and Industrial Research, Delft University of Technology, DLR - German Aerospace Center, IRIAS, University of Liverpool, Politecnico di Milano, NUMECA International, Optimad Engineering, Airbus, the Institute for Information Transmission Problems and the University of Cape Town

only a limited number of structural modes are used. In fact, to prevent any losses in accuracy during the transformation from physical base to modal base, a number of modes equal to the structural degrees of freedom shall be used instead. However, the convergence study performed here demonstrated that on top of the 6 rigid body modes, 37 flexible modes, for a maximum frequency of 25 Hz, resulted in optimal accuracy and were therefore considered sufficient for this study. The same modal formulation is used to describe the unsteady aerodynamic forces which are therefore strongly dependent on the number of modes that are used in the dynamic analysis. In the frequency domain, the unsteady aerodynamic forces are defined as

$$\tilde{\mathbf{F}}_{Aero} = q_{\infty} \left[\mathbf{Q}_{hh}(M, k) \tilde{\mathbf{q}}_h + \mathbf{Q}_{hx}(M, k) \tilde{\boldsymbol{\delta}}_x + \mathbf{Q}_{hg}(M, k) \tilde{\mathbf{w}}_g \right] \quad (3.45)$$

where $\mathbf{Q}_{hh}\{N_{Modes} \times N_{Modes}\}$, $\mathbf{Q}_{hx}\{N_{Modes} \times N_{ControlSurf}\}$ and $\mathbf{Q}_{hg}\{N_{Modes} \times 1\}$ are respectively the generalized aerodynamic forces matrices related to the Fourier Transform of the generalized coordinates $\tilde{\mathbf{q}}_h$, control surfaces vector $\tilde{\boldsymbol{\delta}}_x$ and gust shape $\tilde{\mathbf{w}}_g$. According to the DLM implementation in Nastran [52], the generalized aerodynamic forces matrices are defined through the combination of basic matrices as

$$\begin{aligned} \mathbf{Q}_{hh} &= \boldsymbol{\Phi}_{ah}^T \mathbf{G}_{ka}^T \mathbf{W}_{kk} \mathbf{S}_{kj} \mathbf{A}_{jj}^{-1} \mathbf{D}_{jk} \mathbf{G}_{ka} \boldsymbol{\Phi}_{ah} \\ \mathbf{Q}_{hx} &= \boldsymbol{\Phi}_{ah}^T \mathbf{G}_{ka}^T \mathbf{W}_{kk} \mathbf{S}_{kj} \mathbf{A}_{jj}^{-1} \mathbf{D}_{jx} \\ \mathbf{Q}_{hg} &= \boldsymbol{\Phi}_{ah}^T \mathbf{G}_{ka}^T \mathbf{W}_{kk} \mathbf{S}_{kj} \mathbf{A}_{jj}^{-1} \left(\cos \gamma_j e^{-\omega(x_0 - x_j)/V} \right) \end{aligned} \quad (3.46)$$

where

- $\mathbf{G}_{ka}\{(N_{Panels} \times 2) \times (N_{Nodes} \times 6)\}$ is the spline matrix that transfers forces and displacements between the structural and aerodynamic meshes respectively;
- $\mathbf{D}_{jk}\{N_{Panels} \times (N_{Panels} \times 2)\}$ and $\mathbf{D}_{jx}\{N_{Panels} \times N_{ControlSurf}\}$ are the differential matrices defined as $D = D^1 + ikD^2$, where D^1 defines the downwash component of on the panel due to the panel deflection, while D^2 provides the downwash component due to the panel pitching and plunging velocities. Only D^1 is used for the \mathbf{D}_{jx} definition;
- $\mathbf{A}_{jj}^{-1}\{N_{Panels} \times N_{Panels}\}$ is the Aerodynamic Influence Coefficient (AIC) matrix whose generic i, j element defines the pressure on the i^{th} panel due to a downwash on j^{th} panel;

- $\mathbf{S}_{kj}\{(N_{Panels} \times 2) \times N_{Panels}\}$ is the integration matrix that converts the pressure distribution on the panel in a lift force and pitching moment defined on the 1/4 of the panel chord;
- $\mathbf{W}_{kk}\{(N_{Panels} \times 2) \times (N_{Panels} \times 2)\}$ is a matrix that can be defined by the user to correct the DLM with experimental or CFD data [53–55];
- $\cos \gamma_j e^{-\omega(x_0-x_j)/V}$ accounts for the j^{th} panel dihedral angle as well as the delay of the gust downwash on the j^{th} panel due to the aircraft actually travelling through the gust.

Following the Nastran notation [52], the subscript j represents the aerodynamic panels set, k is the set of the aerodynamic degrees of freedom, and x is the set of the aerodynamic extra degrees of freedom. In general the control surfaces deflections, x -set, are accounted for by means of transpiration boundary conditions, i.e. by applying a local variation of the downwash velocity on the related aerodynamic panels without actually physically rotating them. The h index defines the modal degrees of freedom while the a -set the physical structural ones. The dimension of the aerodynamic degrees of freedom is twice the number of the aerodynamic panels. Each panel has only the pitching and the plunging degrees of freedom defined with respect to the local normal direction and can generate only a lift and a pitching moment, always defined with respect to the same local reference. The GAF matrices, however, are generated only for a limited set of reduced frequencies (k) and Mach numbers; the remaining intermediate values are evaluated through interpolation schemes [52].

Due to the high flexibility of the wings, nonlinearities need to be taken into account for gust loads evaluations [56]; hence, it is necessary to recast the aeroelastic equation of motion (Equation 3.43) in a time domain manner. It then becomes essential to have a good interpolation of the GAF matrices in order to achieve a reliable modelling of the unsteady aerodynamic effects that, in the frequency domain, are taken into account by the dependency of the GAF matrices upon the reduced frequency k [52]. Rational fraction approximation (RFA) methods [57–60] are used here to approximate this transcendental dependency by means of rational polynomial functions. This approach leads to the transformation from an integral-differential to a purely differential set of equations. Although several techniques exist in the literature to obtain a good RFA, the Roger's approximation scheme [57] is used within this work. According to Roger, each member

of the GAF matrices, for a fixed Mach number, can be expressed as

$$Q_{i,j}(k, M) \approx Q_0 + ikQ_1 + (ik)^2Q_2 + \sum_{l=1}^{N_{poles}} \frac{ik}{ik + b_l} Q_l \quad (3.47)$$

where the Q_l terms are real unknown coefficients to be evaluated in order to minimize the least mean square error between the actual value of $Q_{i,j}(k, M)$ and the approximated value given by Equation 3.47. The poles b_l must be positive to ensure stability of the system and are defined as

$$b_l = \frac{k_{max}}{l} \quad (3.48)$$

where k_{max} is the maximum reduced frequency of interest. The number of the poles N_{Poles} defines the order of the overall denominator polynomial. The higher N_{Poles} , the better the fitting, but the higher the computational cost to solve the aeroelastic system of equations. Notice that when a rational fraction approximation approach is used to model the unsteady aerodynamics, a reasonable compromise needs to be found between a short computational time and a good interpolation of the original GAF matrices, as the accuracy of the latter is usually improved by increasing the number of aerodynamic poles. Of particular complexity is the interpolation of the gust loads GAF matrix \mathbf{Q}_{hg} [61, 62], which in the frequency domain is modelled (in Nastran) as

$$\tilde{\mathbf{F}}_{Gust}(\omega) = q_\infty \tilde{\mathbf{w}}_{profile}(\omega) \mathbf{Q}_{hj} \mathbf{w}_j^{Nas}(\omega) \quad (3.49)$$

where $\tilde{\mathbf{w}}_{profile}$ is the Fourier Transform of the gust profile (e.g. “1-cosine”) and \mathbf{w}_j^{Nas} is the downwash vector where each component is defined as

$$w_j^{Nas}(\omega) = \cos \gamma_j e^{-i\omega(x_j - x_0)/V} \quad (3.50)$$

In Equation 3.50, γ_j is the dihedral of the j^{th} panel, $(x_j - x_0)$ is the distance of the j^{th} panel control node with respect to the gust origin and the exponential term $e^{-i\omega(x_j - x_0)/V}$ introduces a delay for each panel that accounts for the effects of the aircraft entering the gust (i.e. the wing hits the gust before the tailplane). When applying RFA to obtain the corresponding time approximation [61, 62], the highly “spiral” nature of the components of $\mathbf{Q}_{hj} \mathbf{w}_j^{Nas}$ results in a problematic interpolation of these terms. However, solutions to this problem are proposed in the literature [63–65] and the approach proposed by Kier [65] is applied here where, briefly, the idea consists in applying the RFA only to \mathbf{Q}_{hj} and defining the gust vector directly in the time

domain. This solution allowed for a reduced number of poles to be used for the interpolation. In fact, only 5 poles were found to be sufficient, given the good agreement observed in the comparison of its components to those of the \mathbf{Q}_{hj} extracted directly from the Nastran analysis. Hence, following this approach, the gust is not modelled as a single scalar value but as a vector of gusts defined for each aerodynamic panel; this also enables the modelling of non-uniform spanwise gusts by simply defining a scaling function for the intensity of the gust vector along the wingspan, as shown by A. Castrichini along with the validation of this modelling approach [66].

3.6 Conclusions

This chapter aims to provide a good level of familiarity with the aeroelastic models used throughout this work for the reconstruction of gust and turbulence properties. A summary of the models presented in this chapter and the relative optimisation method used for the reconstruction of gusts and turbulence profiles, with a direct link to their application and results, is provided in Table 3.1.

Inverse method	Gradient-free optimisation	Gradient-based optimisation
2D aerofoil with Theodorsen's aerodynamics and with linear geometric properties §4.3.1	2D aerofoil with Theodorsen's aerodynamics and with linear geometric properties §5.4.1	-
2D aerofoil with Theodorsen's aerodynamics and with nonlinear geometric properties §4.3.1	2D aerofoil with Theodorsen's aerodynamics and with nonlinear geometric properties §5.4.2, §5.7.1	-
-	-	Flat plate with unsteady lumped vortex method §5.4.3
-	-	2D aerofoil with computational fluid dynamics §5.4.4
-	-	2D aerofoil modelled by the CFD-derived reduced order model §5.4.5
3D beam-stick model with the aerodynamics forces determined by the Doublet Lattice Method §4.3.2	-	3D beam-stick model with the aerodynamics forces determined by the Doublet Lattice Method §5.7.2

TABLE 3.1: Summary of aeroelastic models and reconstruction methods used.

The first column refers to the models used to demonstrate the applicability of the inverse method, introduced in Section 2.5. This is also the topic of Chapter 4, which covers the theory behind the method and its application in industry and highlights the needs for the introduction of the optimisation method. The latter is, thus, discussed in Chapter 5 and the models used are summarised in the second and third column above.

Chapter 4

The inverse method

It was mentioned in Chapter 2 that the standard approach used for the reconstruction of an unknown input force is based on a method in the frequency domain that assumes a linear, invertible, system. This approach forms the main topic of Chapter 4 which, thus, gives an overview of what is known as the “inverse method” for the reconstruction of input forces.

The description of this method is included in this thesis to earn some familiarity with what has been, in the last few decades, common practice in the reconstruction of input forces but, also, to understand the limits of applicability of this approach and to justify the need for a time domain solution designed for the reconstruction of gust velocity profiles.

4.1 Introduction

Recalling from Section 2.5, the excitation force $f(t)$ can be calculated with inverse methods directly from the measured responses $a(t)$ by evaluating the inverse of the forward system model $H^{-1}(\omega)$ and transforming the resulting signal $\hat{F}(\omega)$ in time domain using forward and inverse Fourier transforms, as illustrated in Figure 4.1.

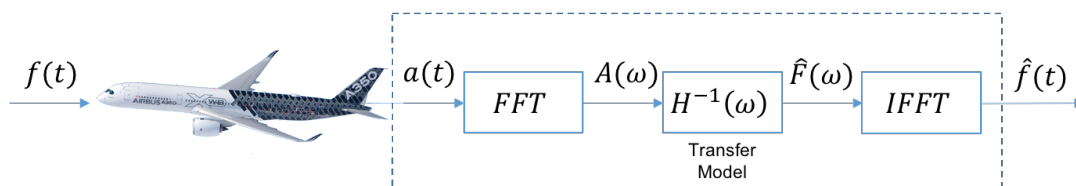


FIGURE 4.1: Force prediction via the inverse method where FFT denotes the forward Fourier transform and $IFFT$ the inverse Fourier transform.

This practice requires the system to be linear as it involves the application of Fourier transforms and their inverse, which are only valid in linear regimes. However, several studies can be found in the literature where this type of application is attempted for nonlinear systems. ChihKao and ChihChern [67] proposed an inverse method valid for nonlinear structural systems; however, this method requires the application of an extended Kalman filter and a recursive least-square estimator that both work in the time domain. Oosterhuis et al [68], instead, applied the inverse method in the frequency domain iteratively to a highly nonlinear multibody quarter car model and showed that an acceptable match between the calculated and the original excitation could be achieved; however, many iterations were needed, as well as further user interaction, to reach overall convergence. The following sections illustrate the theory behind typical inverse method approach and provide some examples of both valid and infeasible applications.

4.2 Methodology

4.2.1 The convolution integral

When performing an inverse analysis for the reconstruction of the excitation force, deconvolution is the most common technique used by researchers [69]. Assuming that the system is linear, i.e. that the response is linearly dependent on the excitation force and that the deformation of the body is small enough to neglect geometric nonlinearities, the response $a(t)$ measured at a point of the body can be related to the excitation force $f(t)$ by a linear convolution¹ integral

$$a(t) = \int_0^t h(t - \tau) f(\tau) d\tau. \quad (4.1)$$

Equation 4.1 can also be conveniently written as

$$a(t) = h(t) * f(t) \quad (4.2)$$

where $h(t)$ is the impulse response function of the system and $*$ indicates the convolution operation. Provided that the response of the system can be measured and that the impulse response function is known, the excitation force $f(t)$ can be estimated by rearranging the integral

¹Convolution can be thought of as a linear summation of a series of identical time histories where each time history is scaled in amplitude and displaced one additional step in time. So any input shape can be generated given a (usually simple) ‘unit’ input shape such as a ‘box-car’ or ‘impulse’, and if the same scalings and time displacements are used for all columns of the time history array then the resulting time history will retain the full correlation and balance properties of the seed [69].

of Equation 4.1, i.e. by means of deconvolution. Details on how to solve the convolution (or deconvolution) integral are out of the scope of this thesis but methods can be found in numerous references [e.g. 1].

When moving into the frequency domain, convolution can be transformed into a simple algebraic multiplication using Fourier transforms ($\mathcal{F}\{\cdot\}$). The Fourier transform of the impulse response function, i.e. $\mathcal{F}\{h(t)\}$, is the frequency response function (FRF), or transfer function, of the system. FRFs are implicit properties that describe how the system behaves under harmonic excitation at any frequency and are mathematically represented by the relationship between the input and the output of the system, i.e.

$$H(\omega) = \frac{A(\omega)}{F(\omega)} = \frac{\mathcal{F}\{a(t)\}}{\mathcal{F}\{f(t)\}} \quad (4.3)$$

where $A(\omega)$ is the Fourier transform of the output of the system $a(t)$, $F(\omega)$ is the Fourier transform of the input of the system $f(t)$ and ω is the frequency of interest. Although FRFs are usually defined for a single harmonic input, when solving real problems their definition can be extended by using Fourier transforms of the measured time signals (see Section 4.2.2). Equation 4.3, rearranged, may then be used to determine the response of the system to a given excitation time history, i.e.

$$\mathcal{F}\{a(t)\} = H(\omega) \cdot \mathcal{F}\{f(t)\} \quad (4.4)$$

or,

$$A(\omega) = H(\omega) \cdot F(\omega) \quad (4.5)$$

as also illustrated by the diagram of Figure 4.2.

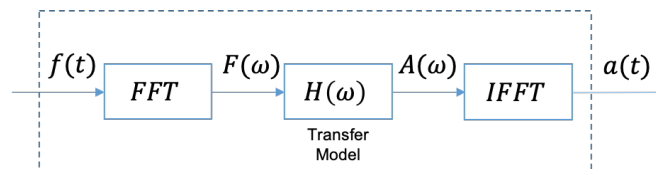


FIGURE 4.2: Illustration of the transfer function operation.

The point-wise multiplication (denoted by the operator \cdot) of the RHS of Equation 4.5 thus represents the corresponding operation in the frequency domain of the convolution defined in Equation 4.2, as stated by the *convolution theorem* [1]. Hence, considering that the inverse operation is valid and assuming that the FRF of the system is known, the time history of an unknown excitation force can be found by solving Equation 4.5 for $F(\omega)$ and, finally, evaluating

its inverse Fourier transform, i.e.

$$f(t) = \mathcal{F}^{-1}\{F(\omega)\} \quad (4.6)$$

as shown in the diagram of Figure 4.3.

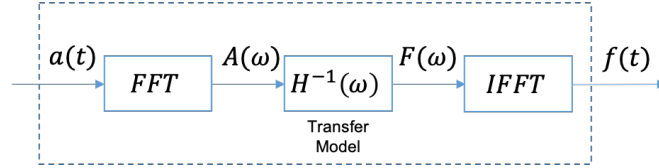


FIGURE 4.3: Illustration of the inverse transfer function operation.

4.2.2 The use of Fourier transforms

Although fast Fourier transforms (FFT) are computationally less expensive than solving the convolution integral in the time domain, care must be taken when dealing with FFT algorithms. This is because the Fourier transform is originally defined for an infinite continuous signal but, for this type of practical applications (i.e. signals of finite length T), this needs to be truncated using the definition

$$A(\omega) = \frac{1}{T} \int_0^T a(t)e^{i\omega t} dt. \quad (4.7)$$

In fact, this truncation introduces errors, also known as ‘leakage’, due to discontinuities created at the extrema of the finite data sets. The amplitude of these errors, however, can be reduced by applying windowing functions to the data [1]. In general, windowing minimises the amplitude of the discontinuities at the limits of each finite data set. This reduction is achieved by multiplying the time signal by a finite-length window that has an amplitude that gradually tends to zero at the two extrema; in this way, the endpoints of two consecutive signals meet smoothly, forming a unique continuous signal. Several windowing functions exist (e.g. Hann, Hamming, exponential) which have different frequency resolution and a different degree of effectiveness in reducing leakage. A list of the most common types of windowing functions and their applications is provided in Table 4.1. A good compromise between applicability and effectiveness is given by the exponential window [70], which is used throughout this work and it is defined as $e^{-\kappa t}$, where κ is a positive constant, and thus gives

$$\tilde{a}(t) = a(t)e^{-\kappa t}. \quad (4.8)$$

Window Type	Signal Content
<i>Hann</i>	Sine wave or combination of sine waves
<i>Flat Top</i>	Sine wave (if accuracy of amplitude is relevant)
<i>Hann</i>	Narrowband random signal (vibration data)
<i>Uniform</i>	Broadband random (white noise)
<i>Uniform / Hamming</i>	Closely spaced sine waves
<i>Force</i>	Excitation signals (hammer blow)
<i>Exponential</i>	Response signals
<i>Hann</i>	Unknown content
<i>Kaiser-Bessel</i>	Two tones with frequencies close but very different amplitudes
<i>Uniform</i>	Two tones with frequencies close and almost equal amplitudes
<i>Flat Top</i>	Accurate single tone amplitude measurements

TABLE 4.1: Types of windowing functions and their typical application. Adapted from M. Cerna et al [71].

4.3 Applications

The method described in this chapter forms the basis of the standard approach used in industry for the reconstruction of gust velocity profiles [69]. However, the linearity of the system is a fundamental requirement for the method to be valid, because it relies on the principle of superposition applied to the responses of each individual component of the Fourier transformed signal. A demonstration of this requirement follows with the application of the inverse approach to two of the models introduced in Chapter 3. These are the typical wing section represented by the aerofoil model of Section 3.2, where non-zero nonlinear terms are considered (equations 3.1 and 3.2), and the linear 3D-beam-stick model introduced in Section 3.5. Whilst the first case study demonstrates the linearity requirement, the second demonstrates the validity of the inverse approach to systems that are linear but more complex.

4.3.1 2DoF model case study

Two sets of measurements are considered for this case study with the typical wing section represented by the aerofoil model of Section 3.2. The first set represents the linear response of the aerofoil to a weak ‘1 – cos’ gust ($\bar{U}_0 = 0.01$ and $\bar{H} = 20.0$); the second set, instead, is representative of the nonlinear response resulting from a ‘1 – cos’ gust of greater amplitude ($\bar{U}_0 = 0.2$ and $\bar{H} = 40.0$). Both gust profiles are originally defined according to Equation 2.13 but, in this case study, are assumed to be the unknowns of the problem. Equation 4.3 is used to obtain the system’s transfer functions for pitch and heave, namely $H_\alpha(\omega)$, $H_{\dot{\alpha}}(\omega)$, $H_\xi(\omega)$, $H_{\dot{\xi}}(\omega)$,

where $a(t)$ is given by the time histories of the system's response (i.e. $\alpha, \dot{\alpha}, \xi, \dot{\xi}$) to a known gust input, $f(t)$. The latter, whose usage could be interpreted as a 'trainer' for the system's transfer functions, is chosen to be a swept-frequency cosine (*chirp*) signal, defined analytically [72] as

$$f_g^C = f_{g,max} \sin \left[\phi_0 + 2\pi \left(f_0 t + \frac{k^*}{2} t^2 \right) \right] \quad (4.9)$$

where $f_{g,max}$ is the maximum amplitude of the signal, ϕ_0 is the initial phase (at $t = 0$), f_0 is the starting frequency and k^* is the rate of frequency change. The reason behind the choice of the *chirp* input lies in the properties of this type of signal as it excites the system at a range of frequencies, such that the input frequency changes instantaneously. Figure 4.4 shows the chirp signal in both the time and the frequency domain and also shows that an exponential window is used to avoid the aforementioned 'leakage' problem. Thus, two sets of system's transfer

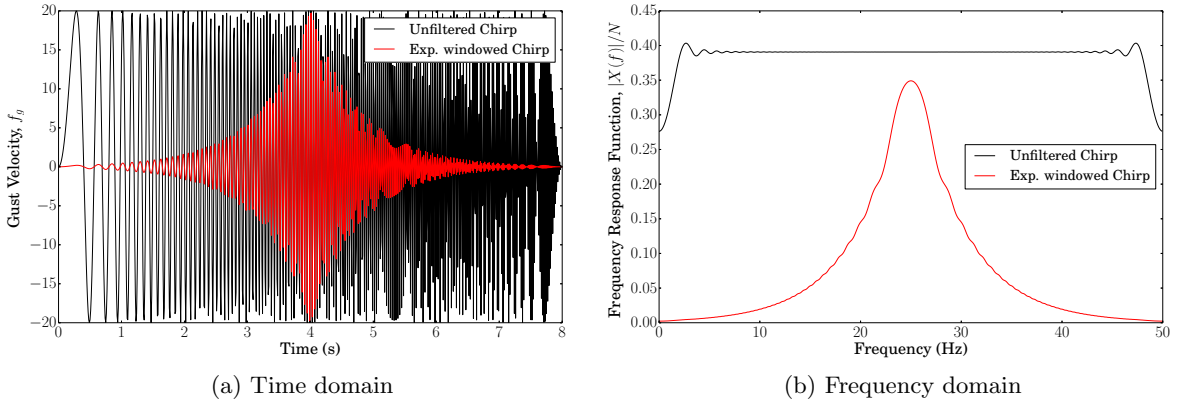


FIGURE 4.4: Example of swept-frequency cosine (*chirp*) signal with $f_{g,max} = 20.0$, $f_0 = 0.0$ Hz, $f_1 = 25.0$ Hz and $k^* = 0.4$ Hz/s.

functions are obtained using, respectively, the unfiltered and the exponential windowed chirp signals with $f_{g,max} = 1.0$, $f_0 = 0.0$ Hz, $f_1 = 2.0$ Hz and $k^* = 0.8$ Hz/s, giving

$$H_\alpha(\omega), H_{\dot{\alpha}}(\omega), H_\xi(\omega), H_{\dot{\xi}}(\omega) = \frac{\mathcal{F}\{\alpha^C(t)\}, \mathcal{F}\{\dot{\alpha}^C(t)\}, \mathcal{F}\{\xi^C(t)\}, \mathcal{F}\{\dot{\xi}^C(t)\}}{\mathcal{F}\{f_g^C(t)\}} \quad (4.10)$$

$$\tilde{H}_\alpha(\omega), \tilde{H}_{\dot{\alpha}}(\omega), \tilde{H}_\xi(\omega), \tilde{H}_{\dot{\xi}}(\omega) = \frac{\mathcal{F}\{\alpha^{\tilde{C}}(t)\}, \mathcal{F}\{\dot{\alpha}^{\tilde{C}}(t)\}, \mathcal{F}\{\xi^{\tilde{C}}(t)\}, \mathcal{F}\{\dot{\xi}^{\tilde{C}}(t)\}}{\mathcal{F}\{f_g^{\tilde{C}}(t)\}} \quad (4.11)$$

where the superscript C indicates the respective response to the unfiltered signal and the superscript \tilde{C} the respective response to the exponential windowed chirp. Figure 4.5 shows the resulting transfer functions, where black curves are used for the unfiltered chirp signal and red curves, instead, for the exponential windowed signal. It can be observed that the frequency peaks of the transfer functions built on the original signal are perfectly captured by the transfer

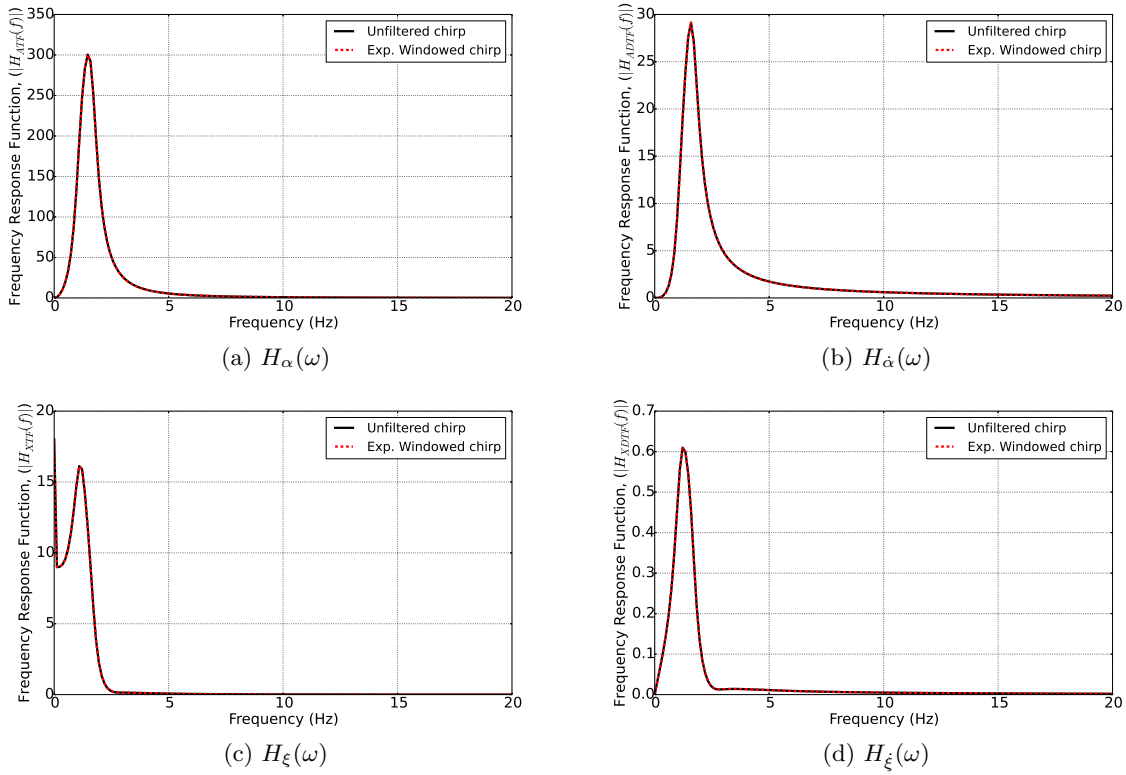


FIGURE 4.5: Systems' transfer functions for α , $\dot{\alpha}$, ξ and $\dot{\xi}$ obtained with Equation 4.10 and using the chirp signals of Figure 4.4.

functions built on the windowed signal, thus demonstrating that no information is lost with windowing. Hence, considering the system's transfer functions obtained from the exponential windowed signal, the time history of any given input excitation force can be 'reconstructed' by taking the inverse of these transfer functions, or in other words by rearranging Equation 4.5, such that

$$F(\omega) = H(\omega)^{-1}A(\omega), \quad (4.12)$$

and by evaluating the inverse Fourier transform of the resulting $F(\omega)$, i.e. $f(t) = \mathcal{F}^{-1}\{F(\omega)\}$. Figure 4.6a shows that an accurate reconstruction of the gust velocity profile can be achieved when dealing with the first set of measurements, which reproduces the linear behaviour of the system. If the gust is of greater amplitude, as in the second set of data, the inverse method no longer provides a truthful reconstruction of the input force due to the effects of geometrical nonlinearities that are not recovered by the linear systems' transfer functions (Figure 4.6b).

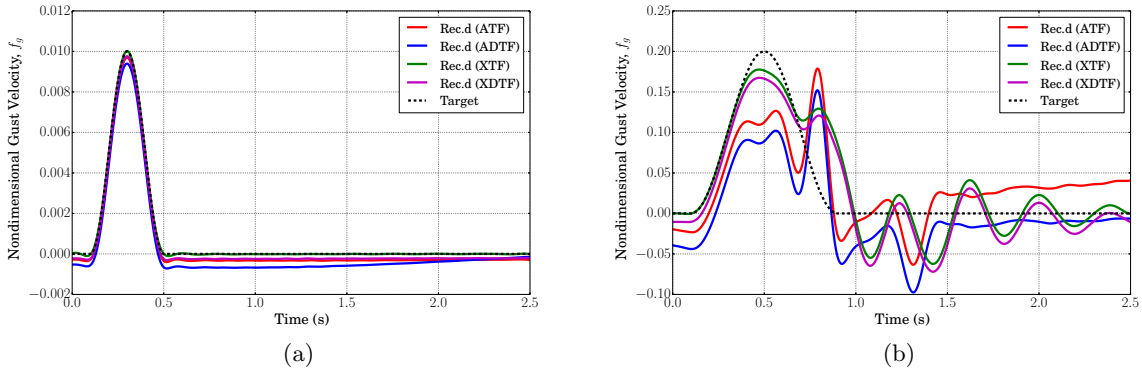


FIGURE 4.6: Comparison between applications of the inverse method to the linear (a) and nonlinear (b) version of the wing typical section subject to a ‘1 – cos’ gust. *ATF*, *ADTF*, *XTF* and *XDTF* refer to the transfer functions of Equation 4.10 constructed, respectively, for α , $\hat{\alpha}$, ξ and $\hat{\xi}$.

4.3.2 3D-beam-stick case study

The same approach used in the flex-flight case study to obtain the system’s transfer functions is used here with the linear 3D-beam-stick model described in Section 3.5. A chirp input signal is defined according to Equation 4.9 and with the properties shown in Figure 4.4 ($f_{g,max} = 20.0$, $f_0 = 0.0$ Hz, $f_1 = 25.0$ Hz and $k^* = 0.4$ Hz/s); then, the transfer function for the vertical acceleration at the centre of gravity location of the aircraft (approximately at mid-fuselage) is obtained using the following equations, respectively, for the unfiltered and exponential windowed chirp signal

$$H_{\ddot{z}_{cg}}(\omega) = \frac{\mathcal{F}\{\ddot{z}_{cg}^C(t)\}}{\mathcal{F}\{f_g^C(t)\}} \quad \text{and} \quad \tilde{H}_{\ddot{z}_{cg}}(\omega) = \frac{\mathcal{F}\{\ddot{z}_{cg}^{\tilde{C}}(t)\}}{\mathcal{F}\{f_g^{\tilde{C}}(t)\}}. \quad (4.13)$$

Figure 4.7 shows the frequency content of this transfer function and, again, a match between $H_{\ddot{z}_{cg}}$ and $\tilde{H}_{\ddot{z}_{cg}}$. The inverse of $\tilde{H}_{\ddot{z}_{cg}}$ (or, equally, the inverse of $H_{\ddot{z}_{cg}}$) is then used to obtain the input gust profile from three different acceleration response time histories, as per Equation 4.12. Two of these time histories are originally obtained from ‘1 – cos’ gust gradients that are derived from the design gust velocity envelope of Figure 2.10; the longest gust considered has a gradient of 214m and a maximum amplitude of 12m/s, whilst the shortest gust has a gradient of 128m and a maximum amplitude of 11m/s. The third gust considered is of the chirp type but has a final instantaneous frequency considerably different ($f_1 = 1.6$ Hz with $k^* = 0.4$ Hz/s) from the one used to obtain the transfer functions. The results of this application are shown in Figure 4.8, where it can be observed that a fairly good reconstruction is achieved for either the first two cases (Figure 4.8a and Figure 4.8b) and for the third case (Figure 4.8c). Virtually no

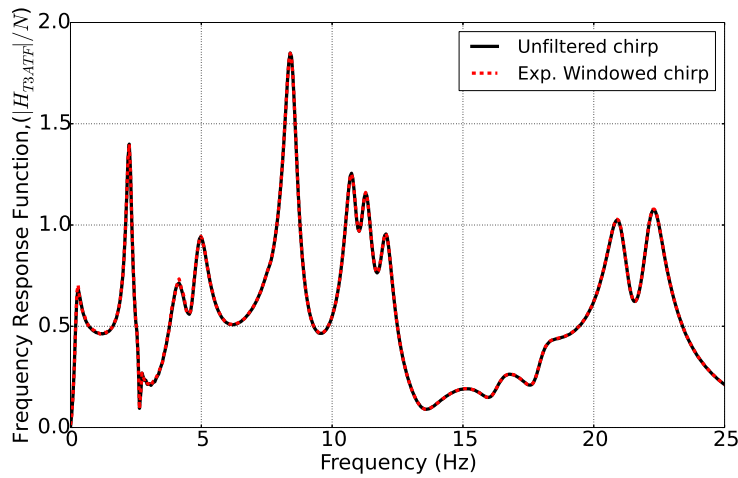


FIGURE 4.7: Systems' transfer function for the aircraft mid-fuselage acceleration obtained with Equation 4.10 and using the chirp signals of Figure 4.4.

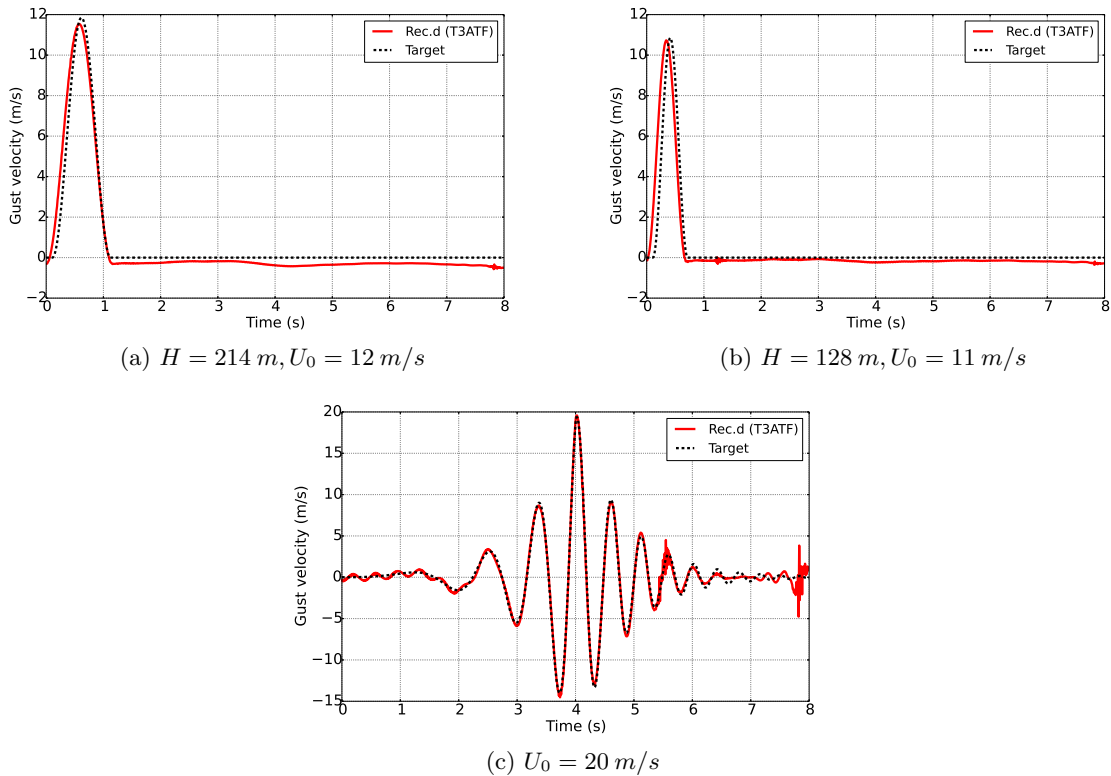


FIGURE 4.8: Comparison of inverse method applications to '1-cos' gusts of decreasing gradient and amplitude (a) and (b) and to a gust input of the chirp form (c).

differences can be observed in the reconstruction of the two '1-cos' gusts, whilst some low and high frequency noise can be seen in the reconstruction of the chirp gust, during the initial and the final two seconds of the simulation. This noise could be avoided by increasing the time of the simulation, thus allowing for the system to reach its final steady state in order to prevent any eventual leakage.

4.4 Industrial practice

The historical approach within the department of loads and aeroelasticity for the reconstruction of gust loads following a flight incident was to compare the aircraft response as indicated by the flight data recorder with the idealised load cases considered for certification. However, over the last two decades, a better approach was developed, able to simulate the event in greater detail using the certification mathematical model matched to the in-service aircraft conditions at the time of the event - in this way, the risk of loads exceedence is determined directly by inspection of the simulated load time histories. This approach is based on the method of convolution and deconvolution introduced in Section 4.2 and it is therefore intrinsically linear in nature. Figure 4.9 shows the sequence of processes performed within the industrial tool. The latter can be divided into three main constitutive blocks: inputs, identification & simulation and assessment. The aspects relevant to the reconstruction are contained within the second block (in green),

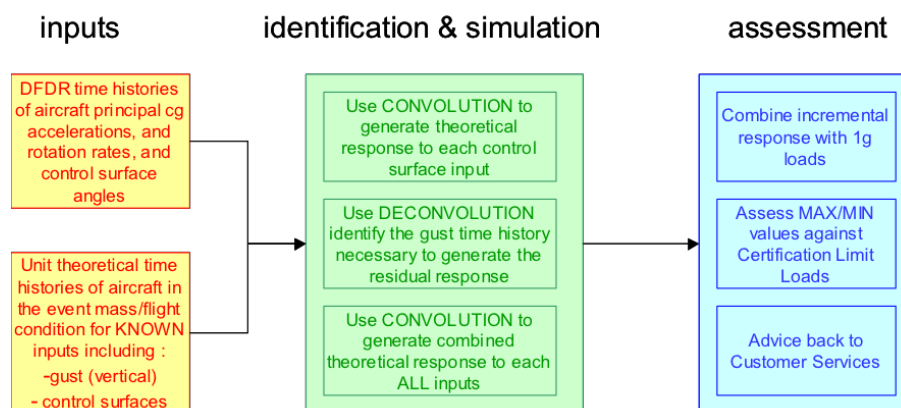


FIGURE 4.9: Main components of the industrial process used for the reconstruction of gust loads. Source: Airbus' tool user guide.

which suggests that the calculation of the loads is obtained in three stages²:

1. First subtracting out the N_z response caused by (symmetric) control surface inputs, and evaluated using Duhammel convolution, from the flight data recorder measurements;
2. Then, determining the (vertical) gust time history necessary to generate the residual aircraft response in one degree of freedom (vertical acceleration at the aircraft c.g.) by means of deconvolution;

²Notice that the process for the reconstruction of the loads is not purely limited to three stages described as several adjustments/options are applied throughout the process, e.g. correction for low frequency trends introduced by any eventual pilot stick inputs.

3. Finally, using convolution again to calculate the aircraft and loads response to the combined control surface and gust inputs.

This process is then followed by an assessment of the response against the certification loads used for that aircraft and criteria in line with the certification requirements are used to classify the resulting loads as Red or Amber, which indicate the criticality of the event.

4.5 Conclusions

A description of the standard approach, used also in industry, for the reconstruction of gust velocity profiles was given in this chapter, together with two sample applications of the method to a model that allows the inclusion of geometric nonlinearities and to a model that is used during the design and production phases of an aircraft. Despite the good results obtained with the industrial model (shown in Section 4.3.2), the inverse method showed a lack of consistency in the results obtained with the nonlinear model (Figure 4.6b). These results suggest that, considering the ever evolving complexity of modern aircraft structures and the nonlinearity of control systems, the linear model assumption may no longer be valid and methods that can handle nonlinearities need to replace the existing standard approaches. Along these lines, next chapter provides a broad discussion on the development of optimisation methods aimed at the reconstruction of gust velocity profiles. These methods handle nonlinearities and are valid in the time domain.

Chapter 5

The optimisation method

This chapter covers the description and application of optimisation techniques in the time domain for the reconstruction of gusts and turbulence profiles. Two methods for the optimisation are discussed here, i.e. gradient-free and gradient based algorithms, that are used independently based on the characteristics of the aerofoil or aircraft model used. The strategy for the validation of these two methods is also presented, followed by the application to the aircraft models described in Chapter 3.

5.1 Introduction

Chapter 4 covered the inverse method in the frequency domain and how this approach can be used for the reconstruction of gust velocity profiles. However, it was shown that some limitations occur when dealing with systems that have nonlinear properties and this approach sometimes results in the non-uniqueness of the solution given by the high sensitivity of the inverse transformation process to measurement noise. Despite a continuous effort to research methods in frequency domain that can overcome these difficulties (Section 4.1), time domain methods present a viable solution, provided that cost efficient routines can be setup for the problem being considered. As mentioned in Chapter 2, the need for time domain methods is also stated in the acceptable means of compliance (AMC) of the certification requirements that explain that these methods shall be used in opposition to frequency domain methods when the explicit simulation of nonlinearities is required. This is because the influence of nonlinearities on one load quantity can greatly differ on another load quantity, thus causing their linearisation hard to achieve [11].

Different aerodynamic and aeroelastic models have been introduced in Chapter 3, as they are used throughout the course of this work to cover the application of gust loads reconstruction during the different design stages of an aircraft. The two main branches of optimisation, i.e. gradient-free and gradient-based, are considered here to provide ad-hoc solutions that conform to the type and characteristics of the aerofoil (or aircraft) model employed and that provide a good level of compromise between accuracy and cost, which is imposed by the design stage in progress. As suggested by their name, gradient-free optimisation methods do not require gradient information and are ideal for non-smooth or discontinuous functions, but they might take longer to find the solution because direct evaluations of the objective function are required at each iteration. In the framework of this study, gradient-free methods can, therefore, be applied to problems where the objective function is computationally cheap to evaluate, which is the case of the 2D aerofoil model with Theodorsen's aerodynamics and nonlinear geometric properties described in Section 3.2. Gradient-based methods use, instead, the gradient of the objective function as the direction to follow in the search of the function minima. One of the key advantages of these methods is their ability to solving optimization problems with a significantly large design space and a high number of design variables; also, as opposed to gradient-free methods, they offer clear convergence criteria and they are computationally efficient thanks to their rapid convergence rate [73]. However, particular attention must be paid in the case of multi-modal functions, where multiple local minima exist, because gradient-based methods cannot guarantee the detection of the global minimum. This is due to the direction of the search following the steep descent of local derivatives, which might result in the optimisation algorithm being stuck in a local minima. But, this is a problem that is usually overcome by randomly searching the design space from different starting points and, in the framework of this research, is limited by the nature of the function being optimised. Thus, gradient-based optimisation is proposed within this work for models with higher complexity that are used during later design stages of the aircraft where a higher accuracy is demanded. These are, for example,

- 2D aerofoil with computational fluid dynamics (§3.4),
- 2D aerofoil modelled by the CFD-derived reduced order model (§3.4.1),
- 3D beam-stick model with the aerodynamics forces determined by the Doublet Lattice Method (§3.5).

Although either optimisation method sits within the framework of Figure 5.1 that, as introduced in Section 2.5, provides a top-level overview of how the reconstruction of an excitation input and,

specifically, gust velocity profiles can be achieved by means of optimisation¹, the two methods present fundamental differences and, also, some changes in the wiring of the objective function to the system. These differences are presented and discussed in the following sections and form the ground for the applications presented later in this chapter.

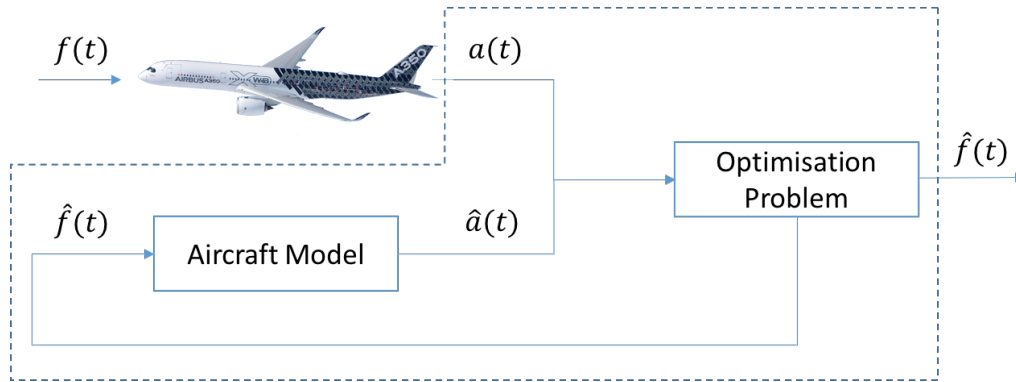


FIGURE 5.1: Force prediction via the optimisation method.

5.2 Gradient-free optimisation framework

Gradient-free methods, also known as non-gradient based algorithms, have gained a lot of attention in recent years as they are easy to implement and/or program thanks to their heuristic nature and are able to seek the optimum point for objective functions that do not have smooth first or second derivatives. These algorithms present global properties, in fact, most of them are designed as global optimisers which are able to find multiple local optima while searching for the global optimum. Gradient-free methods are numerically robust and, indeed, require no gradient information as only objective function evaluations are used to find the optimum point. However, they present some limitations due to their high computational cost, resulting in a lack of efficiency, a limited number of design variables and some necessary fine tuning during the setup of each problem. There exist various gradient-free methods and amongst the most commonly used algorithms (Genetic Algorithms, Simulated Annealing, Divided Rectangles Method, Particle Swarm Optimization, etc.) is the *Nelder–Mead Simplex (Nonlinear Simplex)*, which is detailed in Section 5.2.1 and is used for the reconstruction framework applied to the 2DoF nonlinear aerofoil model. The reason for choosing the simplex algorithm over the genetic

¹The input $\hat{f}(t)$ to a forward aircraft model is tuned via optimisation algorithms until the model response $\hat{a}(t)$ matches the measured response $a(t)$.

algorithms or the other available methods lies entirely in the performance of the former as the others proved to be more expensive in terms of computational effort.

Figure 5.2 provides the first detailed representation of how the framework of Figure 5.1 adapts to the reconstruction of gust velocity profiles and, thus, loads on all aircraft components. The process shown is based on three main steps that are wired to each other in order to form a continuous loop. The main pre-requisite is that the time histories of the response of the aircraft (acceleration, rates, Euler angle, etc.) to an atmospheric disturbance are available. Then, the first step is the estimation of the initial state of the aircraft based on (measured) input data; this is followed by the mathematical prediction of the aircraft state at $t+1$ obtained using an aircraft model, which also reads in control surfaces deflections to extract the response of the aircraft due to manoeuvre only; the third step consists in the optimisation that minimises the difference between the predicted state of the aircraft and the actual measured state at $t+1$ (available in the input dataset), using gust velocity as the design variable. A complete breakdown of these steps is provided below.

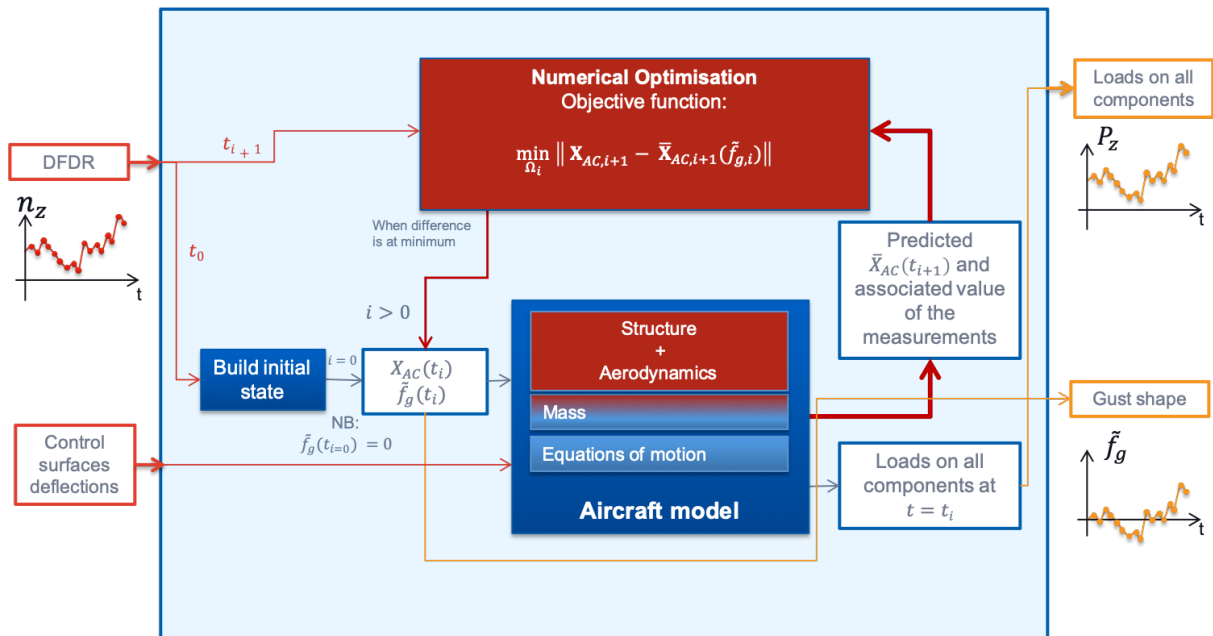


FIGURE 5.2: Proposed Data Flow.

Starting from the beginning, it is assumed that the response of the aircraft to a gust disturbance is provided in the form of a measurement vector \mathbf{X}_{AC} (obtained for example from a flight data

recorder, FDR) that includes accelerations, velocities, Euler angles, body rates and altitude

$$\mathbf{X}_{AC} = \{n_x, n_y, n_z, V_x, V_y, V_z, \phi, \theta, \psi, \dot{\phi}, \dot{\theta}, \dot{\psi}, Z\}^T \quad (5.1)$$

and that, also, a control surface deflection vector \mathbf{U} is given. With this information and considering the first time step in the simulation, the following process applies:

1. The initial state of the aircraft at $t = 0$ is built from the available dataset and the first (vertical) component of the gust velocity profile (as seen by the leading edge of the wing profile) is assumed to be zero², i.e. $f_{g,i=1} = 0.0\text{m/s}$.
2. A model is then required to obtain an estimate $\bar{\mathbf{X}}_{AC}$ at t_{i+1} of the response of the aircraft to the assumed gust input.
3. The predicted $\bar{\mathbf{X}}_{AC}(t_{i+1})$ is fed into an optimisation routine together with the actual (measured) aircraft state $\mathbf{X}_{AC}(t_{i+1})$, which is available from the input dataset.
4. The optimisation tunes the gust input component $f_{g,i}$ such that the ℓ^2 -norm of the difference between the predicted state $\bar{\mathbf{X}}_{AC}$ and the measured state \mathbf{X}_{AC} of the aircraft, at t_{i+1} , is minimum.
5. When the condition at step 4 is satisfied, the gust velocity component is updated accordingly and stored in memory.
6. The process then repeats from step 1 to 5, for all the values of t up to the total number of timesteps, N , such that the complete gust velocity profile is available in the form of a vector.

The resulting gust loads can then be obtained if the gust velocity profile constructed with the optimisation is given as an input to the flight loads model discussed in Chapter 2. However, it is important to highlight that throughout this study the reconstruction of the gust velocity profiles is merely limited to the vertical direction (also referred hereafter as gust downwash) and, therefore, lateral effects are not considered. Also, unless otherwise stated, when considering 3-dimensional aircraft models the reconstructed gust profile is assumed to be equal in the spanwise direction.

²This assumption only holds at the first time step of the simulation; from the following timesteps, the assumed initial value of the gust velocity component is equal to the result of the optimisation obtained at the previous iteration.

5.2.1 The objective function

The optimisation can be constructed as an unconstrained³ problem where the objective function is the ℓ^2 -norm⁴ of the difference between the measured states of the aircraft (accelerations, velocities and displacements) and those obtained from a model that takes as input a parameterised gust. The latter forms the design variable of the optimisation and can be defined as

$$\tilde{f}_{g,i} = \tilde{f}_g(t_i) \quad \text{for} \quad i = 1, \dots, N. \quad (5.2)$$

where $\tilde{f}_g(t_i)$ is the value of the gust velocity at the time step t_i . An example of parameterised gust is shown in Figure 5.3.

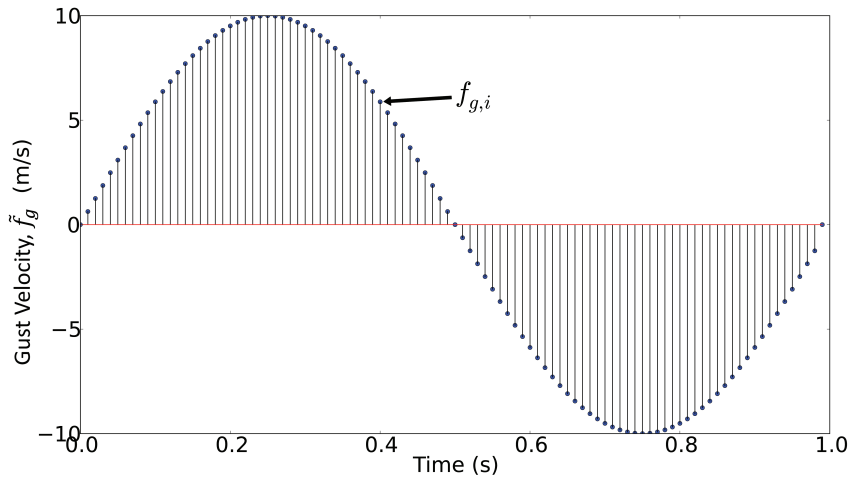


FIGURE 5.3: Example of a parameterised gust input \tilde{f}_g .

In mathematical form, the objective function can be expressed as

$$\underset{\tilde{f}_{g,i}}{\text{minimise}} \quad \left\| \mathbf{X}_{AC,i+1} - \bar{\mathbf{X}}_{AC,i+1}(\tilde{f}_{g,i}) \right\| \quad (5.3)$$

where the operator $\| \cdot \|$ denotes the ℓ^2 -norm, which can be explicitly written as

$$\underset{\tilde{f}_{g,i}}{\text{minimise}} \quad \sqrt{\sum_j \left(X_{AC,i+1}^j - \bar{X}_{AC,i+1}^j(\tilde{f}_{g,i}) \right)^2} \quad (5.4)$$

³In optimisation theory, an unconstrained problem is one that does not require constraints to be defined as part of the problem and, thus, the solution of the objective function is only controlled by the bounds of the design variables.

⁴The ℓ^2 -norm, also known as Euclidean norm, is a vector norm defined for a complex vector and, mathematically, takes the form of $|\mathbf{x}| = \sqrt{\sum_{k=1}^n |x_k|^2}$.

where j is each component of the aircraft state vector. The gust design variable $\tilde{f}_{g,i}$ is bounded by the following relation

$$\tilde{f}_g^L \leq \tilde{f}_{g,i} \leq \tilde{f}_g^U \quad (5.5)$$

where \tilde{f}_g^L and \tilde{f}_g^U are, respectively, the fixed lower and upper boundaries that can be set to significant values, based on experience, or to plus and minus infinity to leave the problem unbounded.

5.2.2 The optimisation algorithm

This optimisation problem is solved using the Nelder-Mead (NM) simplex algorithm, also known as nonlinear simplex, which can be regarded as one of the most appropriate minimisation methods for non-smooth functions, as it does not require any derivative information. In fact, this method can be also used to solve problems with discontinuous functions, whilst it remains the simplest way to minimize continuous or “well-behaved” functions, whose unknown values can be estimated by looking at the existing neighbouring values. A *simplex*, also known as hypertetrahedron, is a structure in n -dimensional space formed by $n + 1$ points that are not in the same plane. For example, a line segment is a 1-dimensional simplex, a triangle is a 2-dimensional simplex and a tetrahedron forms a simplex in 3-dimensional space. The flowchart in Figure 5.4 shows the steps followed by the optimisation algorithm to minimise the objective function. The first step performed by the NM-algorithm is to generate the initial simplex of $n + 1$ points, starting from the initial guess x_0 . The following step is the evaluation of the objective function at each point of the simplex and the consecutive categorisation of these points into the worst (x_w , or highest), the best (x_b , or lowest) and the lousy (x_l , or second highest), based on the values that the objective function takes. New points are then added to the initial (guessed) simplex via four characteristic operations that, thus, modify the structure of the simplex. These operations are *reflection*, *expansion*, *contraction* and *shrinking* and their sequence depends entirely on the value that the objective function takes at the new point, with respect to the other relevant points. Figure 5.5 provides a graphical representation of these operations that, mathematically, are defined as follows

$$\text{Reflection:} \quad x_r = x_a + \alpha(x_a - x_w)$$

$$\text{Expansion:} \quad x_e = x_r + \gamma(x_r - x_a)$$

$$\text{Contraction:} \quad x_c = x_a \pm \beta(x_a - x_w)$$

$$\text{Shrinking:} \quad x_i = x_b + \rho(x_i - x_b)$$

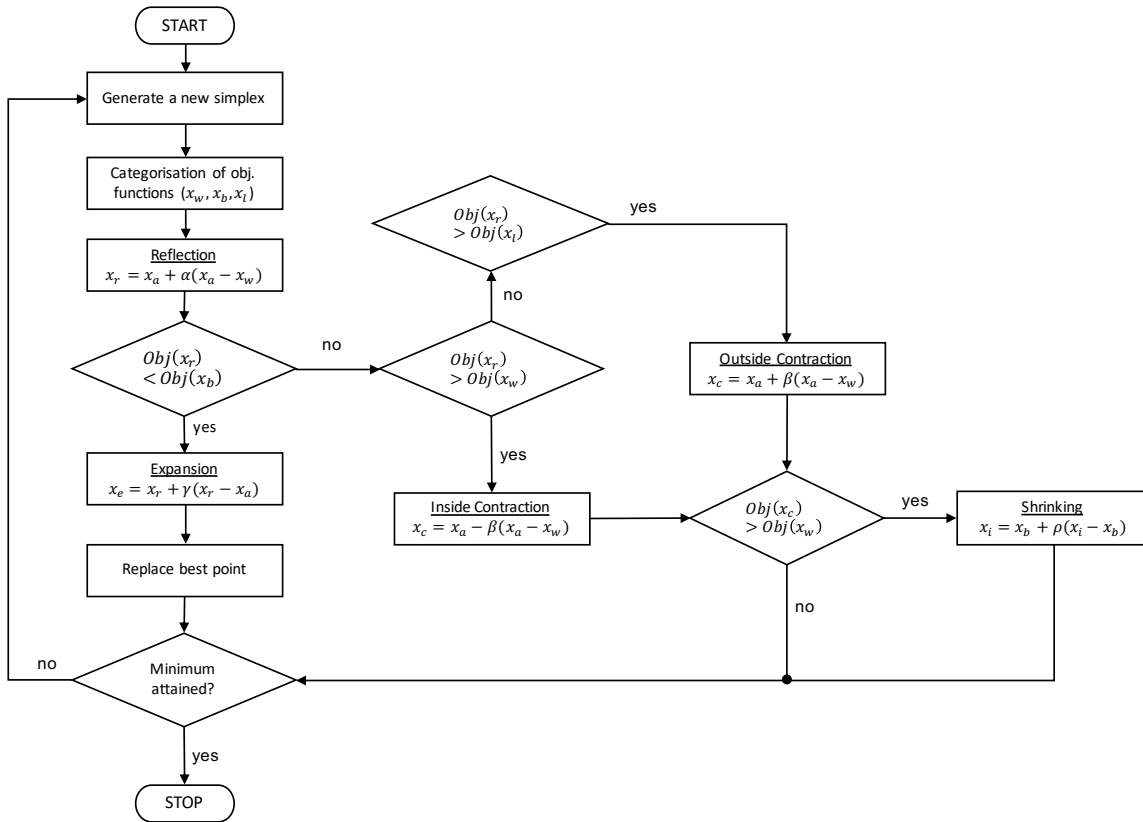


FIGURE 5.4: Nelder-Mead algorithm flowchart. x_w , x_b and x_l are respectively the worst, the best and the lousy (second highest) points. $Obj(_)$ indicates, instead, the value of the objective function obtained with the design point in brackets.

where

$$x_a = \frac{1}{n} \sum_{i=1, i \neq w}^{n+1} x_i \tag{5.6}$$

is the average and α, γ, β and ρ are, respectively, the reflection parameter, the expansion parameter, the contraction factor and the scaling parameter.

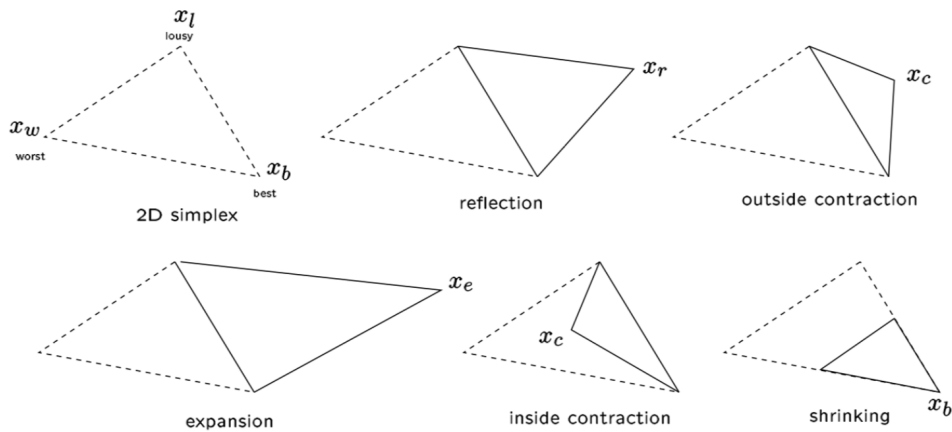


FIGURE 5.5: Operations performed by the Nelder-Mead algorithm on a 2-dimensional simplex [74].

Reflection first requires the computation of the average of the n points except the worst, x_w , such that the new point defined is on a descent line between the worst and the average. The expansion operation follows if the reflection point is better than the best point, i.e. if the objective function obtained with the reflection point, $Obj(x_r)$, is less than the objective function obtained with the best point, $Obj(x_b)$, and, thus, defines a new point in the same direction. If the reflection point is, instead, worse than the worst point or the lousy point, a contraction operation is performed that defines a new point between the worst one and the average one. The plus or minus sign of this operation depends on whether the reflected point is worse than the worst point (minus, also referred to as *inside* contraction) or worse than the lousy point (plus, also referred to as *outside* contraction). Finally, shrinking is a back-up operation that is used in case all the other operations fail. It retains the best point and shrinks all the others by finding them a new position using the scaling factor ρ . After the operations of reflection, expansion, contraction and shrinking, a check for the minimum is performed and if the latter is attained, the algorithm stops otherwise a new simplex is generated and the steps described above, and shown in Figure 5.4, repeat. The convergence of this algorithm can be measured by either setting a tolerance for the size of the simplex or by evaluating the standard deviation, i.e.

$$\sigma = \sqrt{\frac{\sum_{i=1}^n (f_i - \bar{f})^2}{n + 1}} \quad (5.7)$$

where \bar{f} is the mean of the objective function values at the $n + 1$ points. Finally, the main advantage of this algorithm is that it only requires function evaluations and, thus, represents a good choice for simple minimisation problems, where a good reduction of the objective function can be obtained after a relatively small number of function evaluations. Its main strengths are that it does not require the computation of derivatives and the objective function does not necessarily have to be smooth. However, since it does not use any gradient evaluations, it may take longer to find the minimum, which may result into a non-very-efficient method, particularly for problems with more than circa 10 design variables, where convergence becomes increasingly difficult.

5.3 Gradient-based optimisation framework

When dealing with unconstrained optimisation problems, gradient-based algorithms are the most popular choice for finding the solution. Similarly to gradient-free methods, these are

iterative processes but, at each iteration, the gradient-based algorithms exploit knowledge from the gradient of the objective function, and move towards the minimiser of this function by following the direction of its gradient. Four steps can be identified that are common to all the algorithms used for unconstrained gradient-based optimisation - these are:

- **Test for convergence**, which stops the search if the conditions for convergence are satisfied;
- **Computation of the search direction**, which computes the vector $\mathbf{p}(k)$ that defines the direction in n -space along which the solution is searched;
- **Computation of the step length**, which finds a positive scalar α_k that scales the size of the step such that the new design variable provides a solution to the objective function that is closer to the minimum, i.e.

$$f(\mathbf{x}_k + \alpha_k \mathbf{p}_k) < f(\mathbf{x}_k) \quad (5.8)$$

where \mathbf{x}_k is the starting point of the design variable;

- **Update of the design variables**, which stores the new variables, namely

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{p}_k \quad (5.9)$$

updates the iteration number and starts back from the first step.

Several types of gradient-based algorithms exist, which differentiate in the method used for the computation of the search direction. It is out of the scope of this thesis to provide a description of all the available methods; however, more details are given later on the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm on which is based the optimisation method used in this study (Section 5.3.3). Figure 5.6, instead, shows that, in order to apply the gradient-based method to the reconstruction of gust velocity profiles, a slightly modified version of the optimisation framework of Figure 5.2 is required. The main difference is in the format of the input gust, which is now provided as a full time history rather than a single point in time. This gust time history is function of a series of weighted parametric functions whose weights are “controlled” by the optimisation routine. Also, compared to the previous framework, the objective function here is defined for the time history of the total lift coefficient⁵. A more detailed

⁵Accelerations, load factor or other interesting quantities can be equally used for this type of application.

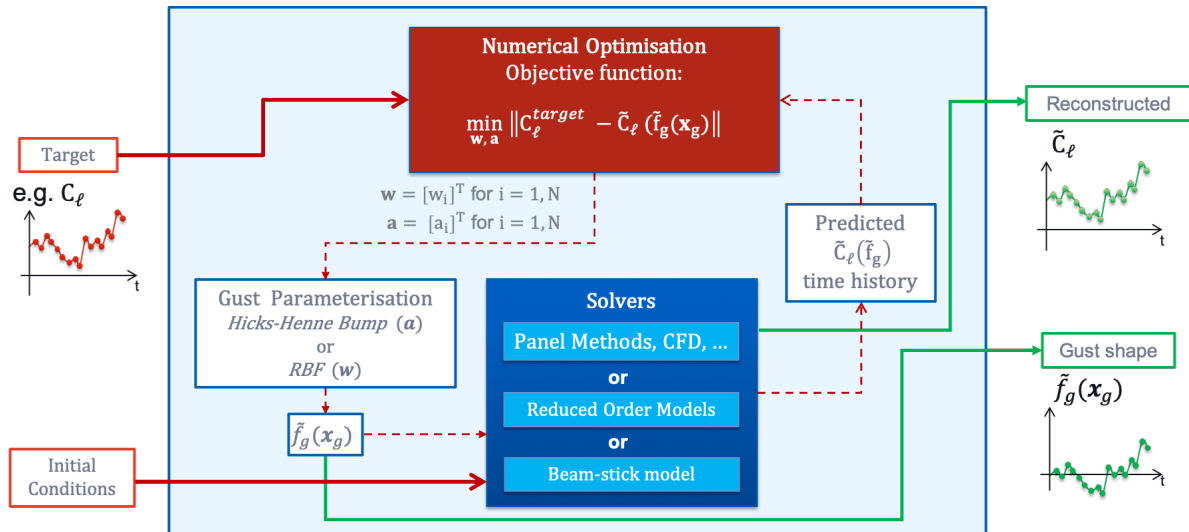


FIGURE 5.6: The numerical optimisation framework used for the reconstruction of the gust input profile with C_ℓ given as the target input. Here, C_ℓ is the lift coefficient; $\tilde{\mathbf{f}}_g(\mathbf{x}_g)$ is the parameterised gust and \mathbf{x}_g the vector of gust points; \mathbf{w} is the vector of the N RBF weights; \mathbf{a} is the vector of the N coefficients for the Hicks-Henne bump functions and N is the number of design variables.

description of the modified framework follows, where it is assumed that the response of the aircraft to a gust disturbance is available in the form of a lift coefficient time history (C_ℓ^{target}) and that, also, the initial conditions of the aircraft are known. With these information, the following process applies:

1. An initial guess for the gust shape $\tilde{\mathbf{f}}_g(\mathbf{x}_g)$ is built over a predefined set of weighted parametric functions, which can be of different nature, as described in Section 5.3.1.
2. An aircraft (or aerofoil) model⁶ is then required to obtain the time history of the response to the initial (guessed) gust excitation, e.g. $\tilde{C}_\ell(\tilde{\mathbf{f}}_g)$.
3. The resulting time history is then handled by the optimisation algorithm where the cost function is the difference between the time history of the model response and the target time history.
4. The optimiser tunes the weights of the parametric functions, i.e. the design variables of the optimisation. Each change in the design variables generates a new parametric gust and an additional call to the model/solver.

⁶This model is trimmed for initial conditions.

5. At convergence, when the cost function is minimised, the resulting gust shape is considered to be the original that caused the target response.

Equally to the gradient-free method, the resulting gust loads can then be obtained if the gust velocity profile resulting from the optimisation is given as an input to the flight loads model discussed in Chapter 2.

5.3.1 Gust shape parameterisation

In aerodynamics, generally, shape parameterisation concerns the way the aerofoil geometry is handled and deformed by an optimisation algorithm, and determines both the fidelity and range of control available. In the context of this study, instead, shape parameterisation is adapted and applied to gust profiles in order to achieve a satisfactory deformation of the initial guessed gust during the iterative optimisation process. Two parameterisation methods have been considered and their effectiveness and efficiency have been measured on the ability to cover a large design space with a limited set of design variables, as suggested by the work of Masters et al [75]. Recalling the first step of the optimisation process, where an initial guess for the gust is required, it becomes natural to assume that deformative methods are the best candidates for this type of application. In fact, deformative methods consist in the generation of a new shape following the deformation of a preliminary (assumed) shape. The two deformative methods considered here are Hicks-Henne bump functions (analytical method) and Radial Basis Functions (free-form deformation method), and are introduced in the following sections.

5.3.1.1 Hicks-Henne bump functions

Hicks-Henne bump functions use a base shape definition over which a linear combination of N basis functions defined between 0 and 1 is added to obtain the desired shape (Figure 5.7). Applied to gust shapes, these can be written as

$$\tilde{f}_g(\mathbf{x}_g) = \tilde{f}_g(\mathbf{x}_g)^{initial} + \sum_{i=1}^N a_i \phi_i(\mathbf{x}_g) \quad (5.10)$$

where a_i are the coefficients of the N basis functions. These are defined as

$$\phi_i(\mathbf{x}_g) = \left[\sin \left(\pi \mathbf{x}_g^{\ln(0.5)/\ln(h_i)} \right) \right]^{t_i}, \quad (5.11)$$

as proposed by Hicks and Henne [76]. In Equation 5.11, h_i is the location of the maxima of the basis functions and is defined as

$$h_i = \frac{1}{2} \left[1 - \cos \left(\frac{i\pi}{N+1} \right) \right], \quad i = 1, \dots, N \quad (5.12)$$

whereas t_i controls the width of the functions and can be set equal to a constant [77]. Although for this study only a_i is set as the design variable for the optimisation, in general all parameters controlling the shape functions (i.e. a_i, t_i and h_i) can be defined as design variables [75].

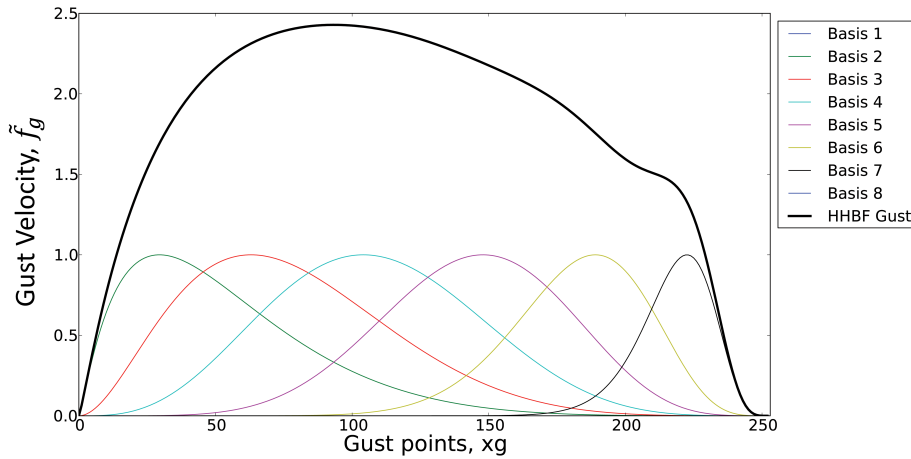


FIGURE 5.7: Example of Hicks Henne Bump Functions parameterisation with 8 basis functions.

5.3.1.2 Radial basis functions

The Radial Basis Functions (RBFs) approximation (Figure 5.8) is built upon the summation of N basis functions that are (usually) defined by the Euclidean norm of a known set of points - the control points, also defined as the RBF centres $\mathbf{c} = [c_i^T]$ - and the variables at which the approximation is sought - the gust points \mathbf{x}_g . RBFs evaluate the value of the interpolation at any point in space by a weighted influence of the value at every control point; the influence is determined by the vector distance of the evaluation point from all the control points. Here, the RBF approximation can be expressed as

$$\tilde{f}_g(\mathbf{x}_g) = \sum_{i=1}^N w_i \phi(\|\mathbf{x}_g - \mathbf{c}_i\|) + p(\mathbf{x}) \quad (5.13)$$

where $\phi(\|\cdot\|)$ are the N ‘radius’ functions, which in this work are of the Gaussian type (i.e. $\phi(\|\cdot\|) = e^{-(\epsilon\|\cdot\|)^2}$) but can likewise be of different nature (e.g. quadric, inverse quadratic, etc.), w_i are N scalar variables, also known as weights of the basis functions, and $p(\mathbf{x})$ is an (optional) added polynomial used to ensure that translation of the gust shape is captured without added deformation. The centres $\mathbf{c} = [c_i^T]$ are defined by the initial guess of the gust shape. When using this parameterisation method within the gust reconstruction framework, the vector of weights \mathbf{w} and the polynomial coefficients are the design variables of the optimisation problem.

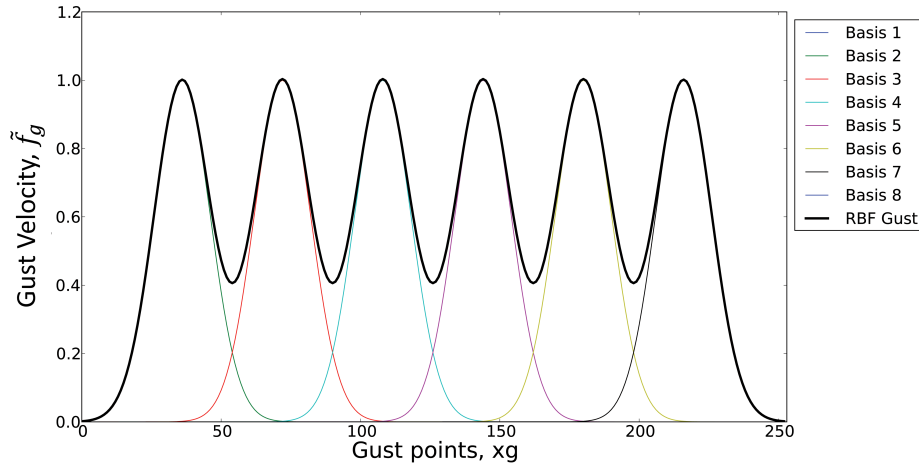


FIGURE 5.8: Example of RBF parameterisation with 8 basis functions.

5.3.2 The objective function

As mentioned earlier, for this application, the optimisation can still be constructed as an unconstrained problem where the objective function is the ℓ^2 -norm of the difference of the measured states of the aircraft and those obtained from a model where the input is the parametrised gust shape $\tilde{f}_g(\mathbf{x}_g)$ [6]. Following on from Figure 5.6, the objective function is designed around the lift coefficient time history; hence, setting $\tilde{g} = \tilde{f}_g(\mathbf{x}_g)$,

$$\underset{\mathbf{w}}{\text{minimise}} \quad \left\| C_\ell - \tilde{C}_\ell(g) \right\|_2 \quad (5.14)$$

or, explicitly,

$$\underset{\mathbf{w}}{\text{minimise}} \quad \sqrt{\sum_j \left(C_{\ell,j} - \tilde{C}_{\ell,j}(\tilde{g}) \right)^2} \quad (5.15)$$

where $\mathbf{w} = [w_i]^T$ is the vector of the weights of the N parametric functions and j counts for each time step. Also, the design variables w_i can be bounded according to the relation

$$w_i^L \leq w_i \leq w_i^U, \quad \text{for } i = 1, N. \quad (5.16)$$

where w_i^L and w_i^U are, respectively, the fixed lower and upper boundaries that can be set to significant values, based on experience, or to infinity to leave the problem unbounded.

5.3.3 The optimisation algorithm

The Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm was used to minimise the objective function of Equation 5.14. This algorithm is considered to be the most effective in the family of quasi-Newton methods. Generally, quasi-Newton methods rely only on first order information and approximate second order information directly from the values of the objective function, available from previous iterations, and the respective calculated gradients. Thus, an approximation of the Hessian matrix is generated, which in most of these methods is forced to be symmetric and positive definite in order to improve the convergence properties of the method [74]. A generic iteration of quasi-Newton methods has the search direction of Equation 5.9 defined as

$$\mathbf{p}_k = -\bar{H}_k^{-1} \nabla f(\mathbf{x}_k) \quad (5.17)$$

where \bar{H}_k is the approximated Hessian, which is usually initialised as an identity matrix and it is then updated at each iteration, and $\nabla f(\mathbf{x}_k)$ is the gradient of the function evaluated at \mathbf{x}_k . The rules for the update of the Hessian vary from method to method and for the BFGS algorithm, the quasi-Newton condition imposed on the update is

$$\bar{H}_{k+1}(\mathbf{x}_{k+1} - \mathbf{x}_k) = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k) \quad (5.18)$$

which is also known as the secant equation⁷. In order to satisfy this condition and for the Hessian to maintain symmetry and positive definitiveness, the approximation is updated according to the following equation

$$\bar{H}_{k+1} = \bar{H}_k + \frac{\mathbf{y}_k \mathbf{y}_k^T}{\mathbf{y}_k^T \mathbf{s}_k} - \frac{\bar{H}_k \mathbf{s}_k \mathbf{s}_k^T \bar{H}_k^T}{\mathbf{s}_k^T \bar{H}_k \mathbf{s}_k} \quad (5.19)$$

⁷A more detailed description of the quasi-Newton condition for the update of the Hessian matrix can be found in many works of the literature [78]

where $\mathbf{y}_k = \nabla f(\mathbf{x}_{k+1}) - \nabla f(\mathbf{x}_k)$ and $\mathbf{s}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$. The inverse of this equation is then required for each update - this inversion can be efficiently achieved through the application of the Sherman–Morrison formula [79, 80]. The new set of values is then obtained through Equation 5.9 and the convergence criteria checked. The iterations stop when the convergence criteria are met.

5.3.3.1 Gradient evaluation

As described in Section 5.3.3, the BFGS algorithm relies on the evaluation of the gradient of the objective function to compute the search direction of the optimisation. Hence, consider the inner function of Equation 5.14

$$f(g) = C_\ell - \tilde{C}_\ell(\tilde{g}) \quad (5.20)$$

and, at every single time step

$$f_j(g) = C_{\ell,j} - \tilde{C}_{\ell,j}(\tilde{g}). \quad (5.21)$$

Then, a new function h can be introduced that defines the ℓ_2 -norm of Equation 5.21, i.e.

$$h(\tilde{g}) = \|f_j(\tilde{g})\|_2 = \sqrt{\sum_{j=1}^n f_j(\tilde{g})^2}. \quad (5.22)$$

As a result, the gradient of $h(\tilde{g})$ can be written as

$$\nabla h(\tilde{g}) = \frac{1}{2} \left(\sum_j f_j(\tilde{g})^2 \right)^{-\frac{1}{2}} \left(\sum_j 2 f_j(\tilde{g}) \nabla f_j(\tilde{g}) \right) = \frac{\sum_j f_j(\tilde{g}) \nabla f_j(\tilde{g})}{\|f(\tilde{g})\|_2} \quad (5.23)$$

or, in terms of the Jacobian,

$$\nabla h(\tilde{g}) = \frac{J_f(\tilde{g})^T f(\tilde{g})}{\|f(\tilde{g})\|_2} \quad (5.24)$$

where $J_f(\tilde{g})^T$ is the Jacobian matrix of $f(\tilde{g})$. The conventional method for finding this gradient is numerically, via finite difference approximation, where a pre-determined⁸ Δw is sequentially added and subtracted to every single weight of the parametric functions and its influence is measured over the entire wing profile (Figure 5.9). A drawback of this method is that the finite difference evaluation usually becomes computationally expensive, because it requires the full simulation to be carried out at every delta increment and the results to be stored in memory. However, a multiprocessing routine (Appendix C.1) was used in this study to speed up this

⁸An appropriate Δw was taken from the “stable region” of a sweep in perturbation sizes to verify that the final value used gave a consistent gradient.

process, with the evaluation of the gradient achievable at the cost of only one single simulation.

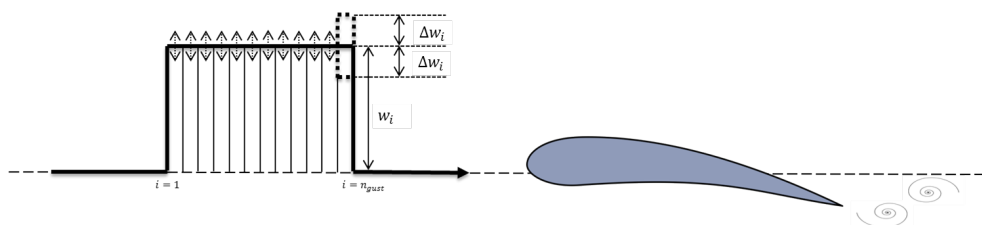


FIGURE 5.9: Representation of finite difference approximation.

5.4 Method validation

The validation strategy for the two gust reconstruction methods introduced in sections 5.2 and 5.3 is discussed here. The diagram of Figure 5.10 can be split into two main parts. The first part, which develops horizontally in the diagram, is based on the application of both the optimisation frameworks (in turn) to the time histories obtained from the models in response to a known input gust, which for convenience is chosen to be the widely accepted ‘1 – cos’ profile of Equation 2.13. The second part consists of the comparison of both the reconstructed gust and the reconstructed responses with their respective targets. The results of these comparisons are then assessed to establish the validity of the methods.

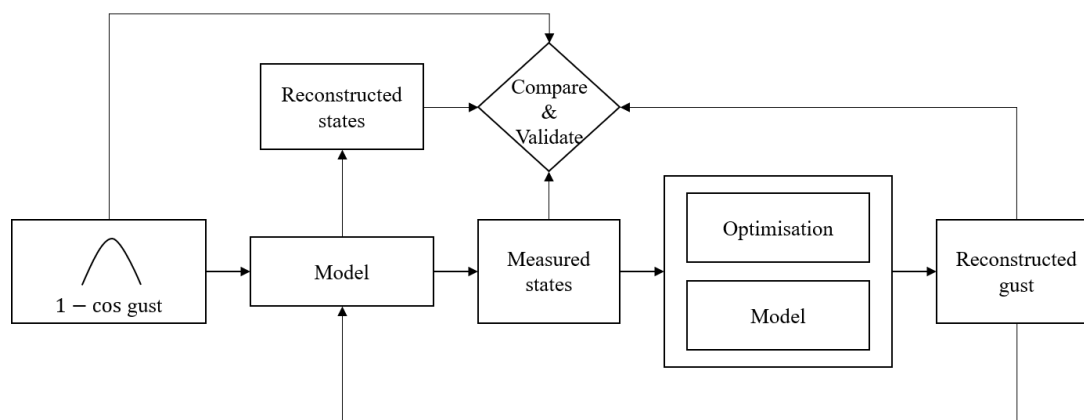


FIGURE 5.10: Validation diagram.

Three models from those introduced in Chapter 3 were used for the validation. These are the 2DoF nonlinear model (Section 3.2), in both its linear and non-linear versions, for the validation

of the gradient-free method and the ULVM model (Section 3.3) and the CFD model (Section 3.4), along with its ROM version, for the validation of the gradient-based method.

5.4.1 2DoF linear model

The diagram of Figure 5.11 represents the first part of the validation and helps to better understand the components required for this case.

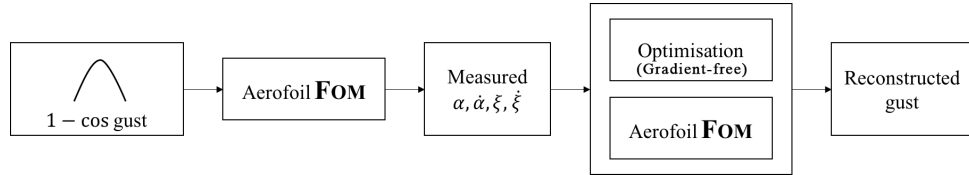


FIGURE 5.11: Gust reconstruction diagram for the linear 2DoF model.

The response to the ‘1 – cos’ gust of Equation 2.13, reformulated in nondimensional terms in Equation 3.8 and, reported for convenience below,

$$\bar{U}(\tau_{\frac{1}{2}}) = \frac{\bar{U}_0}{2} \left(1 - \cos \frac{2\pi}{\bar{H}} \tau_{\frac{1}{2}} \right) \quad \text{for} \quad \tau_{\frac{1}{2}} \leq 0 \leq \tau_{\frac{1}{2}} + \bar{H} \quad (5.25)$$

with gradient $\bar{H} = 40.0$ and maximum velocity $\bar{U}_0 = 0.01$, was first obtained with the linear version of the aerofoil, which from the nondimensional equations 3.3 and 3.4 reduces to

$$\xi'' + x_\alpha \alpha'' + 2\zeta_\xi \frac{\bar{\omega}}{V^*} \xi' + \left(\frac{\bar{\omega}}{V^*} \right)^2 \xi = -\frac{1}{\pi\mu} C_\ell(\tau_{\frac{1}{2}}) \quad (5.26)$$

$$\frac{x_\alpha}{r_a^2} \xi'' + \alpha'' + 2\zeta_\alpha \frac{1}{V^*} \alpha' + \left(\frac{1}{V^*} \right)^2 \alpha = \frac{2}{\pi\mu r_a^2} C_m(\tau_{\frac{1}{2}}). \quad (5.27)$$

The computed response was stored in dimensional terms in a measurement vector of the type described by Equation 5.1, but with the measured states limited to α and $\dot{\alpha}$ for pitch motion and ξ and $\dot{\xi}$ for plunge. The latter was then provided as the target measured data to the optimisation framework shown in Figure 5.2, whilst the input gust was assumed to be an unknown function f_g parameterised as in Figure 5.3, with $dt = 0.05 \text{ s}$ and $t_{final} = 500 \text{ s}$. The optimisation problem was set as unconstrained and with no bounds being defined on the design variables. The iterative process described in Section 5.2 could then be triggered and run until convergence.

Figure 5.12 shows the results required for the second part of the validation. The first figure on the left shows that, despite some numerical noise noticeable soon after the gust peak (shown in the adjacent figure), a very good match was obtained between the theoretical ‘1 – cos’ gust profile and the one reconstructed using the Nelder-Mead simplex algorithm. In practice, the signal shall be treated with appropriate digital filters to remove the high frequency noise, and it would be expected that the conditioned signal had a shape virtually identical to the reference gust.

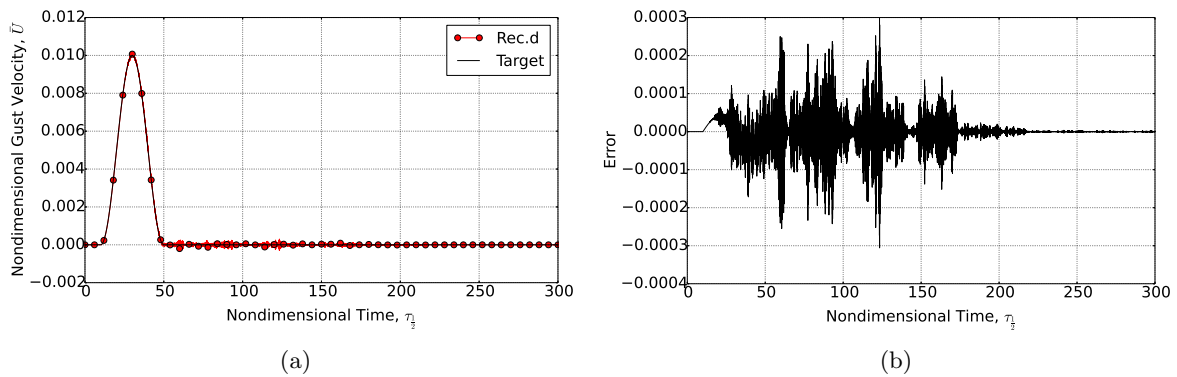


FIGURE 5.12: Comparison of the ‘1 – cos’ gust profile ($\bar{H} = 40.0$, $\bar{U}_0 = 0.01$) with the gust reconstructed via the optimisation run on the linear FOM. The error plotted in (b) is the delta between the target function and the reconstructed one.

As described in Section 5.2.2, the simplex algorithm requires multiple direct calls to the function as no gradients are given or evaluated in the calculation; however, the optimisation only took slightly more than one minute to complete on a standard PC with 4 cores, covering a time history of 10,000 time steps with $dt = 0.05$ s. Indeed, the convergence plot shown in Figure 5.13 demonstrates that less than 10 iterations were required per optimiser evolution for the objective function to converge to an amplitude of the order 10^{-9} .

Finally, the reconstructed gust of Figure 5.12a was given back as input to the linear aerofoil model and figures 5.14a and 5.14b show that a perfect match was obtained with the response resulting from the theoretical ‘1 – cos’ gust.

5.4.2 2DoF nonlinear model

The gust reconstruction diagram of Figure 5.15 shows that only the aerofoil model was updated here to reflect the changes in the nonlinear terms. Indeed, a similar ‘1 – cos’ gust profile with the peak velocity increased by one order of magnitude ($\bar{U}_0 = 0.1$) was used to evaluate the

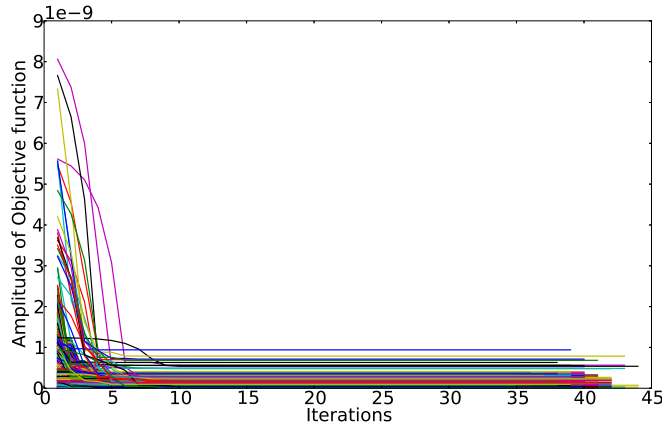


FIGURE 5.13: Convergence plots over the gust profile for the 2DoF linear model. Each line represents the convergence of a single time step.

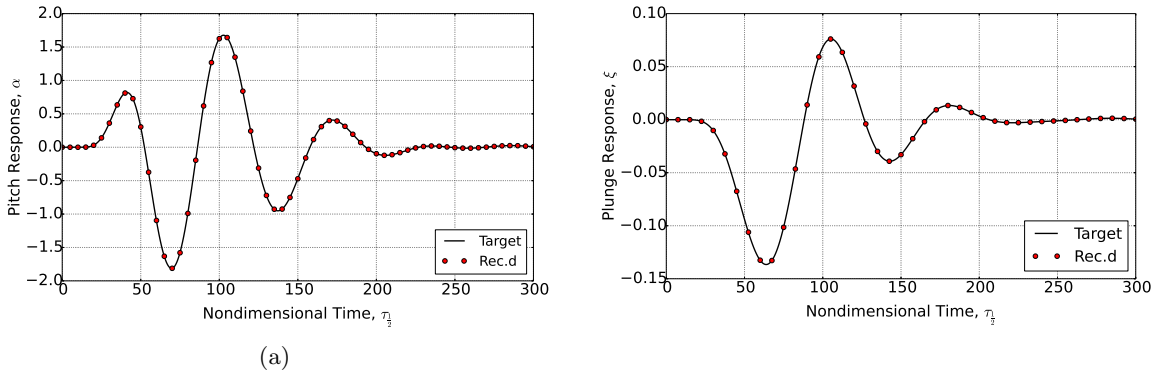


FIGURE 5.14: Comparison between target and reconstructed pitch (5.14a) and plunge (5.14b) for the linear full order model.

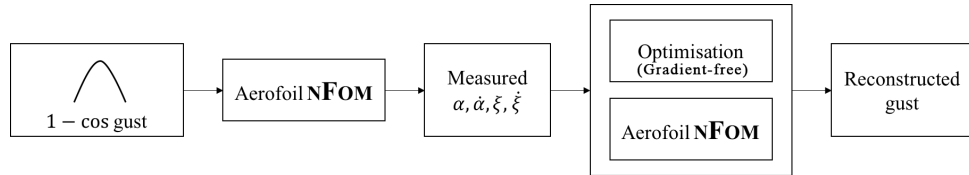


FIGURE 5.15: Gust reconstruction diagram for the nonlinear 2DoF model.

response of the aerofoil modelled by the same linear equation for plunge (Equation 5.26) but with a non-zero cubic stiffness term ($\beta_\alpha = 3.0$) defined in the equation for pitch, i.e.

$$\frac{x_\alpha}{r_a^2} \xi'' + \alpha'' + 2\zeta_\alpha \frac{1}{V^*} \alpha' + \left(\frac{1}{V^*} \right)^2 (\alpha + \beta_\alpha \alpha^3) = \frac{2}{\pi \mu r_a^2} C_m(\tau_{\frac{1}{2}}). \quad (5.28)$$

Exactly the same process of the previous validation exercise applied here, also having the gust parameterised with the same properties and setting the optimisation unconstrained and the design variable unbounded. As a result, the optimisation run in under two minutes on a standard

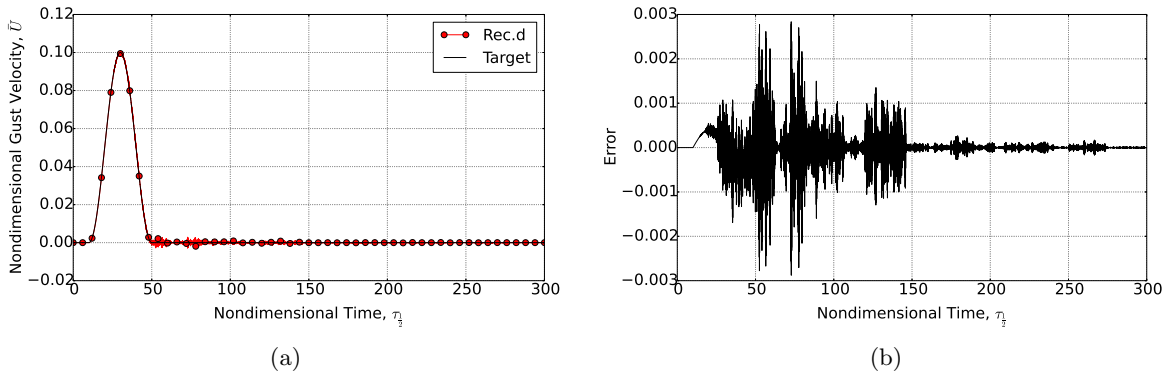


FIGURE 5.16: Comparison of the ‘1 – cos’ gust profile ($h_g = 40.0$, $\Omega_0 = 0.10$) with the gust reconstructed via the optimisation run on the nonlinear FOM. The error plotted in (b) is the delta between the target function and the reconstructed one.

PC with 4 cores and, despite an increase of one order of magnitude in the absolute error (Figure 5.16b), a good match was obtained between the theoretical ‘1 – cos’ gust and the one reconstructed, as shown in Figure 5.16a. Also, a trend similar to the linear model validation verified for the convergence plot but, in this case, to an amplitude of the order 10^{-8} (Figure 5.17).

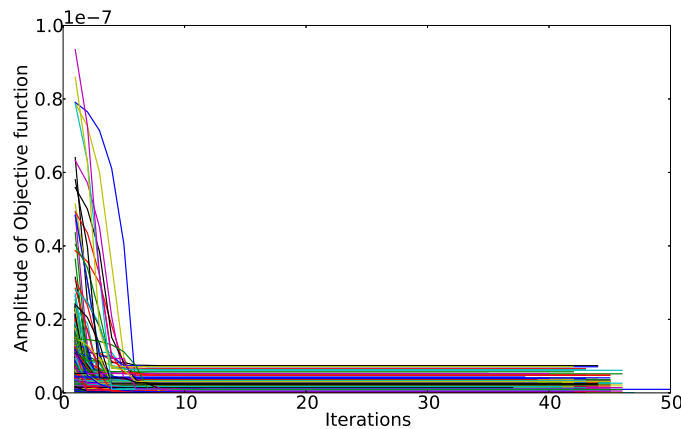


FIGURE 5.17: Convergence plots over the gust profile for the 2DoF nonlinear model. Each line represents the convergence of a single time step.

Moreover, to fulfill the requirements of the second part of the validation, figures 5.18a and 5.18b show that an excellent match is still achieved when comparing the response of the nonlinear model to the two input gusts, i.e. the numerically reconstructed one and the theoretical ‘1 – cos’ gust.

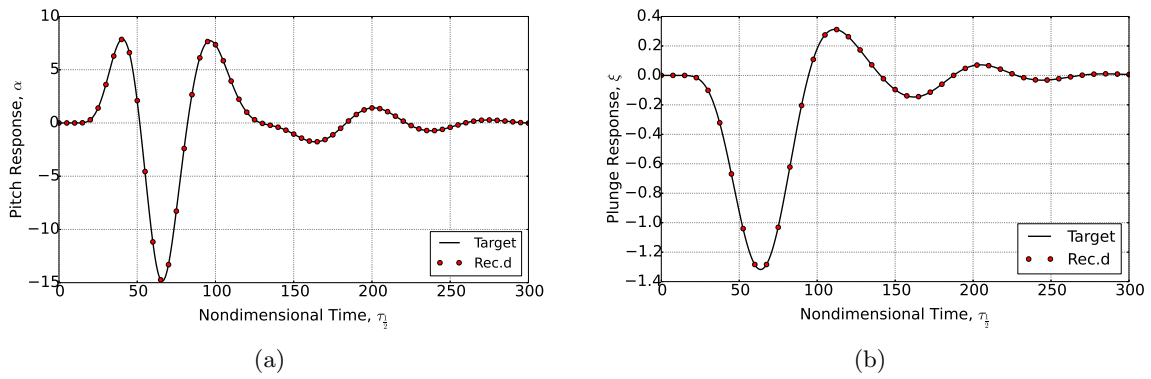


FIGURE 5.18: Comparison between target and reconstructed pitch (5.18a) and plunge (5.18b) responses to a ‘1 – cos’ gust for the nonlinear case.

5.4.3 ULVM model

As mentioned earlier, the first validation for the gradient-based optimisation was obtained with the ULVM model of a flat plate, described in Section 3.3. Figure 5.19 shows that the ULVM was used to compute the lift coefficient time history of the flat plate undergoing a ‘1 – cos’ gust that had a nondimensional maximum velocity $\bar{U}_0 = 0.15$ and a gradient 5 times greater than the chord ($\bar{H} = 5c$). The C_ℓ time history was then provided as measured data to the reconstruction framework built around the ULVM model. As the final output of the iterative process, the reconstructed gust is compared in Figure 5.20 to the reference ‘1 – cos’ gust obtained analytically.

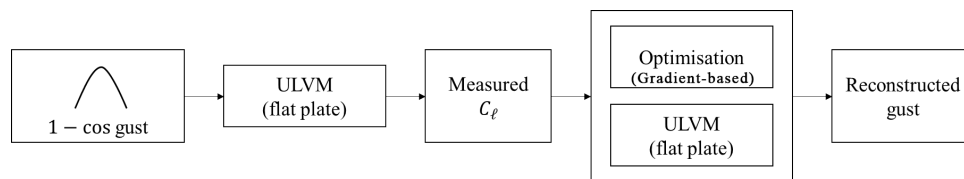


FIGURE 5.19: Gust reconstruction diagram for ULVM.

Both the two methods described in Section 5.3.1, i.e. HHBF and RBF, were used for the parameterisation of the unknown gust input, with the initial guess for the weights set to an amplitude of 0.001 to avoid division-by-0 errors. The first row of the figure shows the results obtained using the RBF parameterisation, whilst the second row shows the results obtained using Hicks-Henne Bump Functions. In both cases, the BFGS algorithm was used for the optimisation. When using RBF parameterisation, the optimisation converged in approximately 20 minutes on a standard PC with 4 cores, requiring 26 objective function evaluations and 4 gradient evaluations; however, Figure 5.20a shows that the initial and final zero-velocity gust

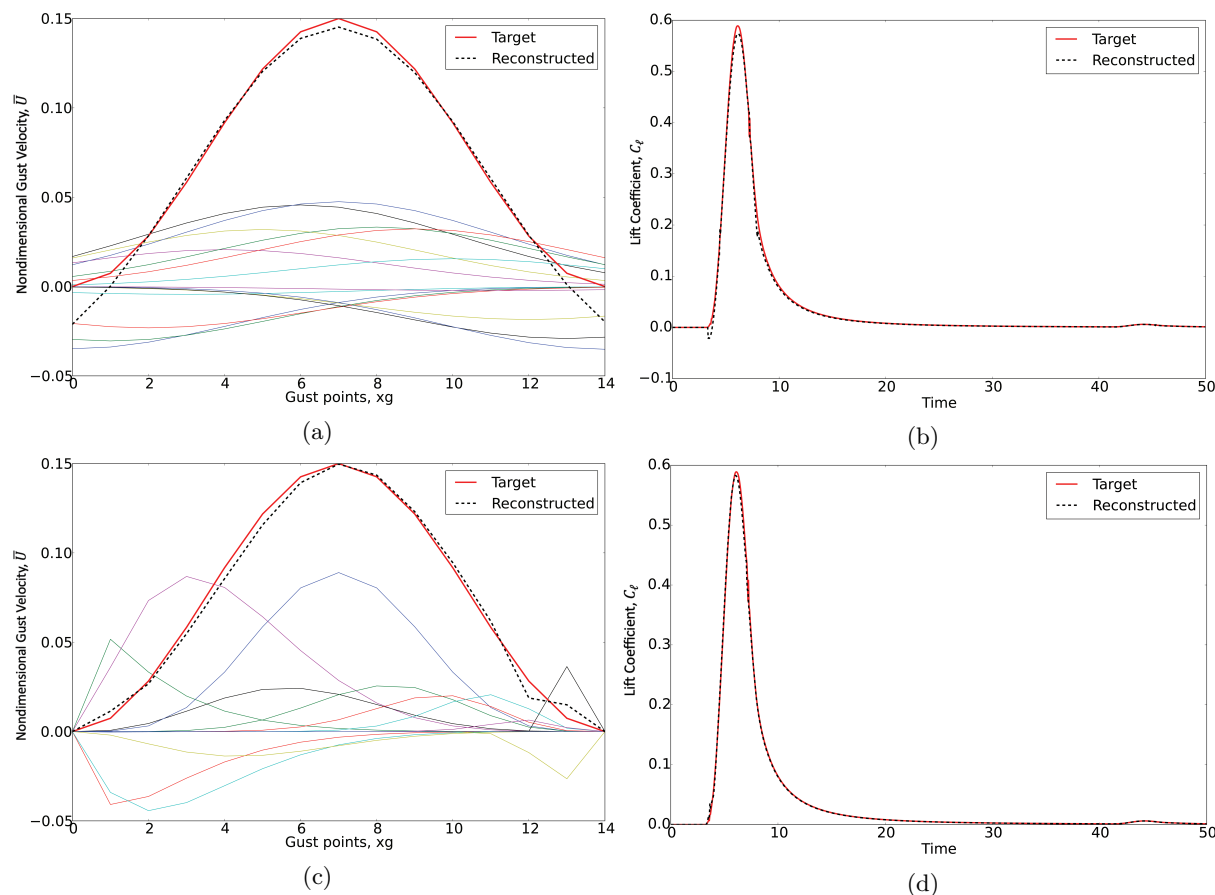


FIGURE 5.20: Reconstruction of gust profiles for a flat plate obtained with the BFGS optimisation algorithm. Figures (a) and (b) show, respectively, the reconstructed gust and the resulting lift coefficient using 14 RBFs for the parameterisation. Figures (c) and (d) show, respectively, the reconstructed gust and the resulting lift coefficient using, instead, 14 HHBFs for the parameterisation.

points, along with the peak, were not well matched by the reconstruction. These resulted in a negative C_l at the impact time and a small discrepancy at the peaks, as shown in Figure 5.20b. With the HHBF parameterisation, instead, the optimisation took longer to converge (~ 2 hours on the same standard PC) as 207 objective function evaluations were required, along with 29 gradient evaluations. However, Figure 5.20c shows that a better reconstruction was achieved, as both the zero-velocity gust points and the peak were perfectly matched; this resulted in a nearly perfect match for the reconstructed C_l time history (Figure 5.20d). Neither constraints nor bounds were set on the design variables for this optimisation problem.

5.4.4 SVM-CFD model

A further validation of the gradient-based optimisation is proposed with the use of the SVM-CFD model of the FFAST crank aerofoil [44], described in Section 3.4. A ‘1 – cos’ gust shape

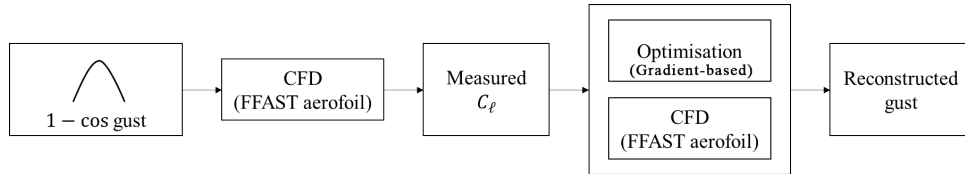


FIGURE 5.21: Gust reconstruction diagram for CFD.

with characteristics similar to the one used for the 2DoF nonlinear model validation but with a dimensional maximum velocity of $U_0 = 15.623$ m/s and gust gradient $h_g = 15c$ was given as input to the CFD model to generate the target C_ℓ time history for the optimisation (Figure 5.21). The flow initial conditions for this case were $M = 0.73466$ and $Re \approx 1.43 \times 10^8$. Also for this case, both parameterisation methods were used to parameterise the unknown gust input, again with an initial guess for the weights set to an amplitude of 0.001 to avoid division-by-0 errors. The results are compared in Figure 5.22, which shows the reconstructed gust and the C_ℓ time histories against the respective targets. It can be observed that a good match was achieved when employing the RBF parameterisation with 14 bases, but a slight discrepancy (of negligible amplitude) verified at both the initial and the final zero points and, also, at the peak of the gust profile (Figure 5.22a). As per the ULVM validation case, these discrepancies also reflect onto the reconstructed C_ℓ time history at the beginning of the build-up in lift and at the maximum C_ℓ obtained. However, they can be reduced if the number of bases is increased with a trade-off on performance. When using the HHBF method, instead, a virtually perfect match was achieved for both the gust profile and the C_ℓ time history using the same number of bases (figures 5.22c and 5.22d).

A quantitative measure of the overall accuracy of the two methods is given by the RMS value of the objective function that for the RBF case is $3.03e^{-3}$ and for the HHBF case is $7.91e^{-5}$, which are results consistent with the figures analysed. In terms of performance (Figure 5.23), the BFGS optimisation algorithm with the RBF parameterisation successfully terminated in 19 iterations, with 49 objective function evaluations and 19 gradient evaluations. The same algorithm but with the HHBF parameterisation took 10 additional iterations to converge, thus resulting in 29 gradient evaluations and 72 objective function evaluations. An additional aspect

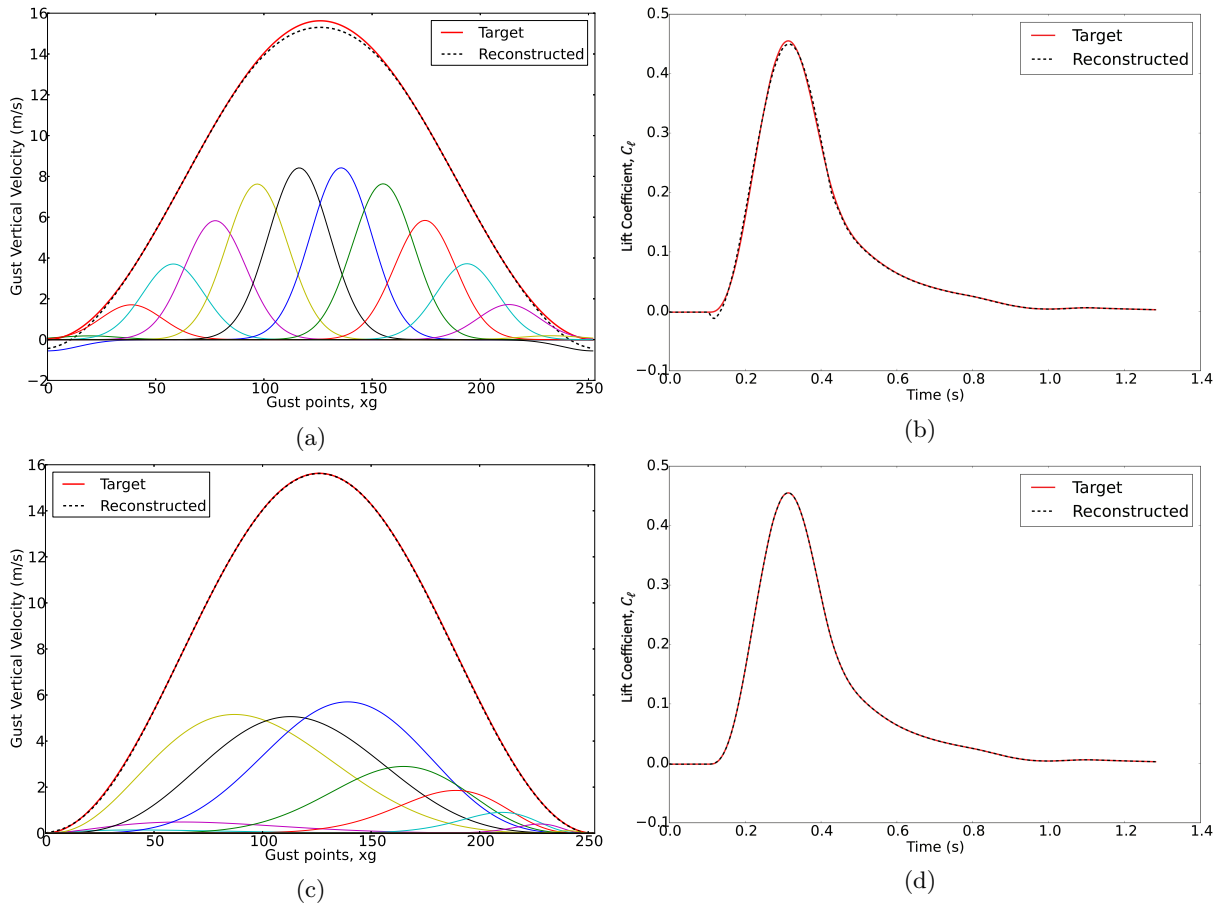


FIGURE 5.22: Reconstruction of gust profiles for the FFAST crank aerofoil obtained with the BFGS optimisation algorithm. Figures (a) and (b) show, respectively, the reconstructed gust and the resulting lift coefficient using 14 RBFs for the parameterisation. Figures (c) and (d) show, respectively, the reconstructed gust and the resulting lift coefficient using, instead, 14 HHBFs for the parameterisation.

to highlight is that, although no constraints were given for this optimisation problem, bounds on the design variables (i.e. the weights of the parametric functions) were necessary to guarantee the convergence of the CFD code. Clearly, CFD analyses are much more expensive than the potential flow approach on which are based the ULVM and the 2DoF nonlinear model; however, as mentioned earlier in Chapter 3, they become essential when analysing the effects of the aerodynamic nonlinearities.

5.4.5 Reduced order model

As represented in Figure 5.24, the measured response of the FFAST crank aerofoil to the known $1 - \cos$ gust input used for this validation exercise was given by the reduced order version of the aerofoil model, obtained according to the procedure described in Section 3.4.1 [46]. The ROM

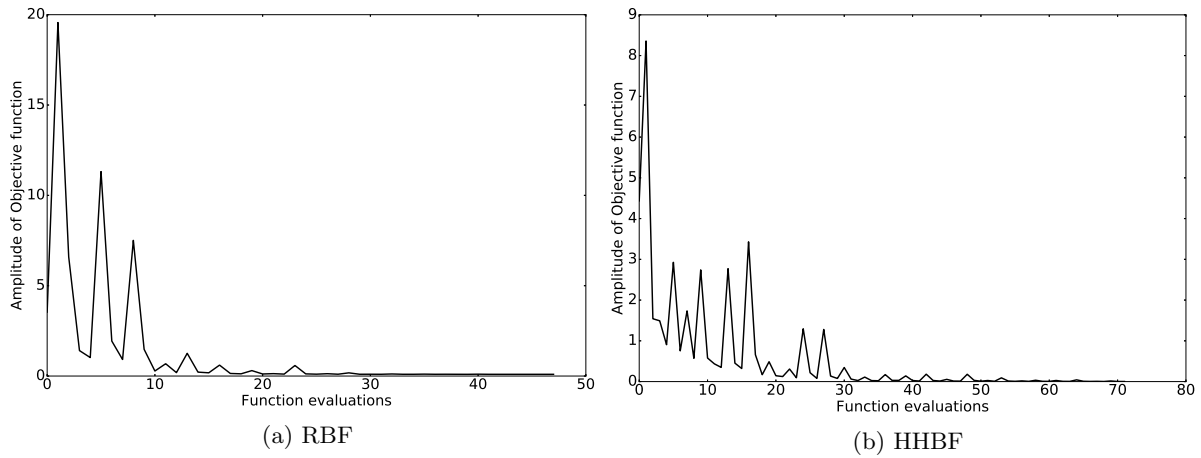


FIGURE 5.23: Convergence plots

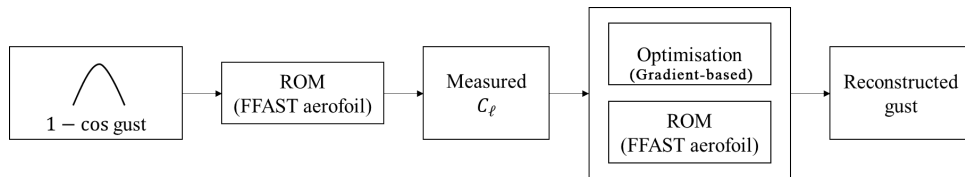


FIGURE 5.24: Gust reconstruction diagram for ROM.

was built for the same Mach and Reynolds numbers used for the CFD simulations ($M = 0.73466$ and $Re \approx 1.43 \times 10^8$ at sea level conditions), and could be used for any gust shape experienced at these flow conditions. Also, as in the previous cases and with both parameterisation methods, the BFGS algorithm was used for the optimisation whilst neither constraints nor bounds were set on the design variables for this case.

The optimisation converged in under a minute when using either RBFs (~ 55 s on a standard PC with 4 cores) or the HHBFs (~ 40 s on the same PC) and a perfect match was achieved in the reconstruction of the gust profile and the C_l time histories, as shown in figures 5.25b and 5.25d. In terms of design variables, Figure 5.25a shows that a parameterisation with 18 radial basis functions was required to achieve the same level of accuracy obtained, instead, with only 8 bump functions. From a computational point of view, the reconstruction of the gust required 600 cost function evaluations when using RBFs and 489 function calls when using HHBFs. For both the parameterisation methods, one may notice that, compared to the CFD cases studied in the previous section, a higher number of objective function evaluations was required here to reach convergence. This was due to a reduction applied on the tolerance bands of the objective function within the optimisation algorithm, which was possible thanks to the quick turnaround time of the ROM and was a desired condition to increase the level of accuracy

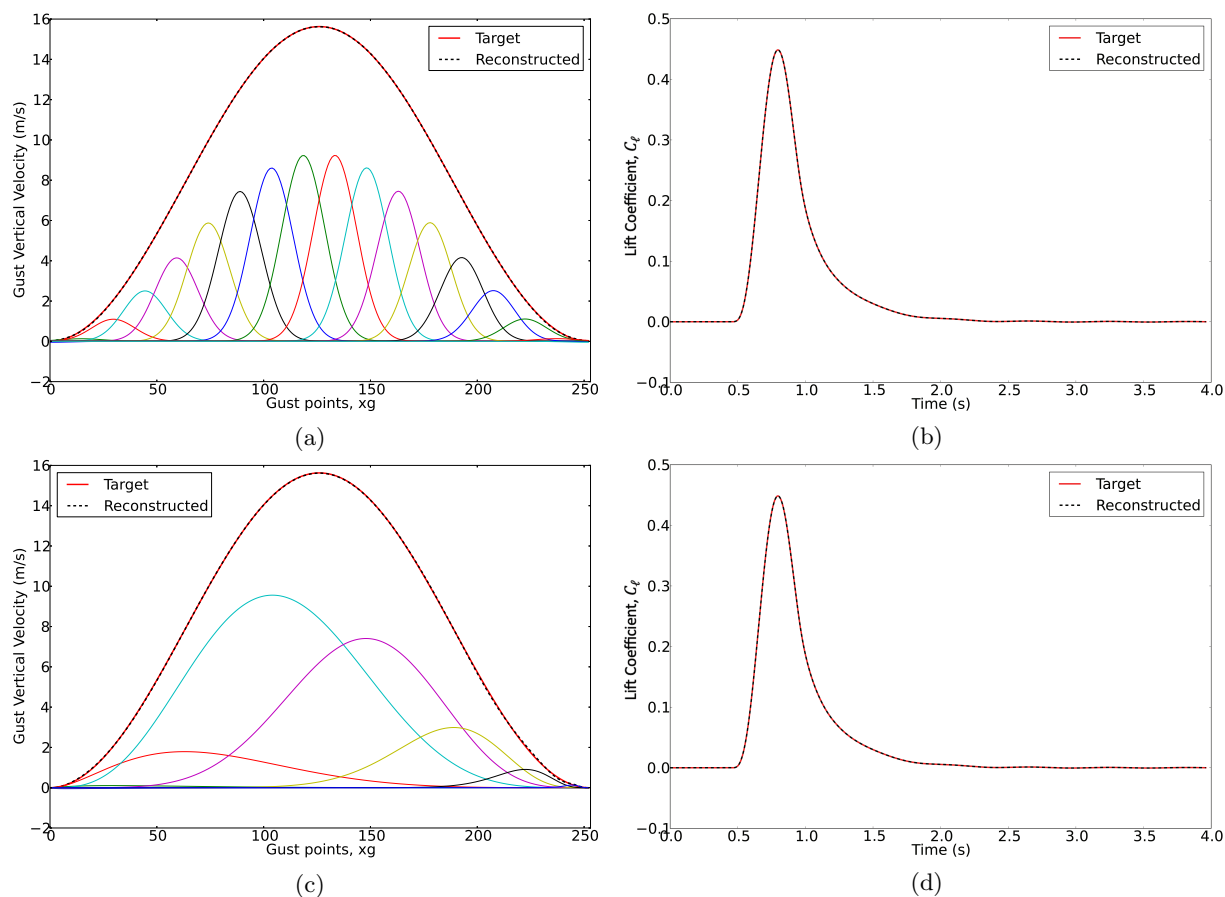


FIGURE 5.25: Reconstruction of gust profiles for the FFAST aerofoil ROM obtained with the BFGS optimisation algorithm. Figures (a) and (b) show, respectively, the reconstructed gust and the resulting lift coefficient using 18 RBFs for the parameterisation. Figures (c) and (d) instead show, respectively, the reconstructed gust and the resulting lift coefficient using only 8 HHBFs for the parameterisation.

of the reconstruction⁹.

The relatively short time required to complete the optimisation process proves that ROMs are a valid choice for this type of applications, as their major cost is in the model generation phase (not analysed in this study) and not in their use. As a result, ROMs allow a quick, efficient and more accurate reconstruction process, which has the potential to be exploited as a technology for instantaneous monitoring of the airframe in flight, which would ultimately minimise, or avoid, the need for further aircraft inspections [8].

⁹The tolerance bands on the objective function establish one of the stop criteria for the optimisation algorithm. Therefore, a reduction on the tolerance (down to $O(10^{-7})$) aims for smaller objective function minima and, thus, increases the number of iterations (and cost function calls) required to reach convergence.

5.5 Data filtering

In all cases discussed in Section 5.4, the two optimisation methods have been validated using analytical data as input. However, for in-service applications of the framework, the input data has to be extracted from flight data recorders, which include measurement noise from the aircraft instrumentation. Hence, appropriate digital filters need to be added into the process as a pre-processing step, as shown in Figure 5.26, to reduce the possibility for measurement noise of interfering with the reconstruction of the gust.

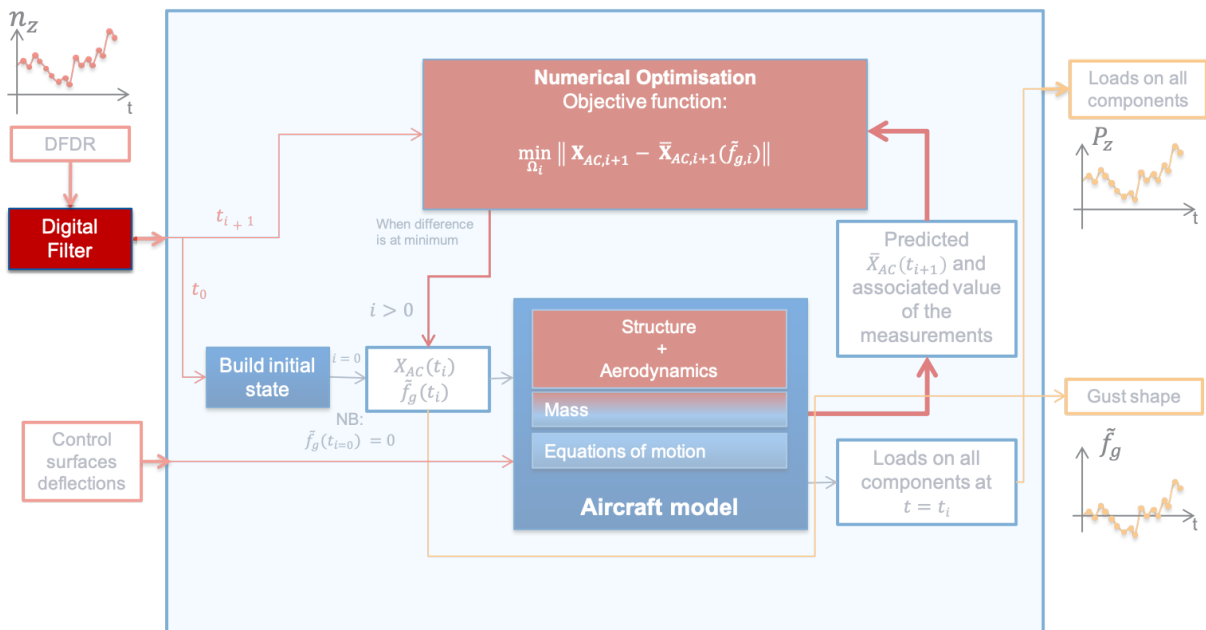


FIGURE 5.26: Proposed Data Flow with Digital Filter.

Another aspect to consider when dealing with data extracted from flight data recorders is the low sampling rate of the recorders, which can sometimes result in a very discrete set of data points and interpolation techniques and appropriate filtering become necessary for the correct functioning of the reconstruction framework [6].

5.6 Modelling errors

The reconstruction of gust profiles obtained by either optimisation framework is subject to errors or uncertainties that are a direct consequence of the modelling assumptions taken for each of the aerofoil/aircraft model adopted in the framework. In fact, a precise reconstruction of the gust profile would require the aircraft model to be complex enough to capture the full interaction

between the gust and the aircraft structure, which for example in this work is a characteristic only achieved by the split velocity method in CFD (Section 3.4). But, it will be shown later (Table 5.1) that enhancements in the model complexity often result in expensive simulations which make the reconstruction framework impracticable. Hence, the correct balance between the level of accuracy required for the results and the complexity level of the model is paramount.

Additionally, errors in input data may also result in an erroneous reconstruction of the gust profile. An example is given by the low sampling rate of the flight data recorders mentioned in the previous section. In fact, FDRs can sometimes return very discrete sets of data points where the sampling rate of some parameters is as low as 1 Hz. This may lead to significant lack of data where relevant parameter ‘peaks’ are not fully captured during the flight event and, as a result, the reconstructed gust profile may not be representative of the real event.

5.7 Applications

The best performing model used for the validation of the gradient-free method and the model that is most representative of an industrial one were considered for additional case studies that consisted in the reconstruction of random and more realistic gust profiles. These models are, respectively, the 2DoF nonlinear model (Section 3.2) and the 3D beam stick model (Section 3.5). The gust velocity profiles studied were built analytically over time approximations of the von Kármán power spectral density (PSD). These approximations are commonly taken in the literature as representative of real gust events [81]. In fact, when dealing with continuous turbulence, which is typically represented by a random variation in velocity of the air normal to (or sometimes along) the flight path of the aircraft, where the random variable is assumed to have a Gaussian distribution with zero mean and is represented by a gust PSD with units of $(\text{m/s})^2/(\text{rad/m})$ [1] (see also Section 2.4), the von Kármán spectrum (Figure 2.14) is the most commonly used as it closely matches experimentally observed data. The von Kármán PSD is defined by Equation 2.18, rewritten for convenience here

$$\Phi(\Omega) = \sigma_w^2 \frac{\bar{L}}{\pi} \frac{1 + \frac{8}{3}(1.339\bar{L}\Omega)^2}{[1 + (1.339\bar{L}\Omega)^2]^{\frac{11}{6}}} \quad (5.29)$$

where $\Omega = \omega/V$ is the scaled frequency (rad/m), σ_g is the root-mean-square turbulence velocity (m/s TAS), V is the flight speed (m/s TAS) and \bar{L} is the characteristic scale wavelength of

the turbulence (see also sections 2.3 and 2.4). For turbulence analysis, the PSD is defined as a one-sided spectrum so only a positive frequency axis is relevant.

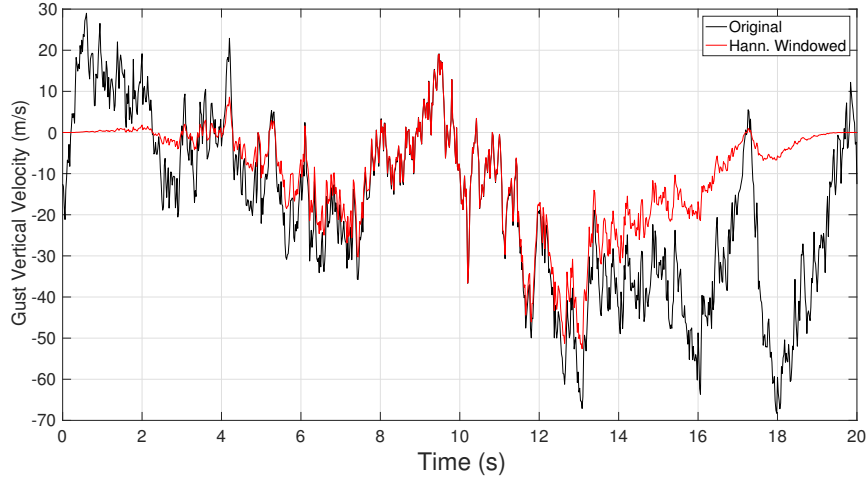


FIGURE 5.27: Example of time domain approximation of von Kármán spectrum, Hanning-windowed.

The time approximations of the von Kármán PSD (Figure 5.27) are obtained here by considering the superposition of an infinite number of sinusoidal components that differ infinitesimally in frequency and are randomly phased, from one to the next. Mathematically, each approximation can be expressed as

$$y_n(t) = \sum_{m=1}^{\infty} \sqrt{\Phi(\omega_m)\Delta\omega} \cos(\omega_m t + \phi_m) \quad (5.30)$$

where $\sqrt{\Phi(\omega_m)\Delta\omega}$ is the infinitesimal amplitude of each of the superimposed sinusoids, ω_m is the radian frequency of each component ranging from 0 to ∞ (however, a discrete subset is used here) and ϕ_m is the random phase angle. Each resulting time history is finally filtered by the application of the Hann window function (red curve of Figure 5.27) that is mathematically expressed by

$$f_{hann}(n) = \frac{1}{2} \left(1 - \cos\left(\frac{2\pi n}{N-1}\right) \right) \quad (5.31)$$

and it is used here to avoid an abrupt change in velocity at the beginning and end of each gust velocity profile modelled.

5.7.1 2DoF nonlinear model

As per the validation cases of the previous section, the block diagram of Figure 5.28 shows the components required for this case study. This time, the $1 - \cos$ gust input is replaced by a

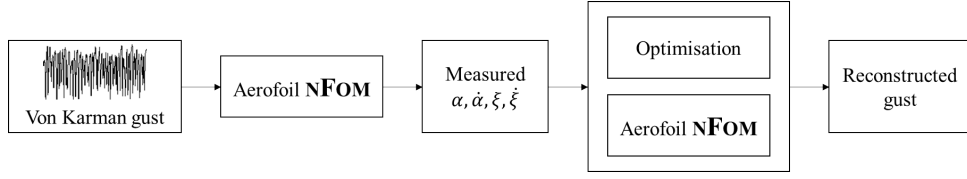


FIGURE 5.28: Gust reconstruction diagram for the nonlinear 2DoF model.

random Hann-windowed signal $y_n^H(t)$ obtained by combining equations 5.30 and 5.31, i.e.

$$y_n^H(t) = f_{hann}(n) y_n(t) \quad (5.32)$$

with amplitude bounded to ± 0.2 , radian frequency ranging from 0 to 25 rad/s and a random phase angle. The nonlinear version of the 2DoF model described by equations 5.26 and 5.28 was then used to measure the pitch and plunge responses of the profile and their respective rates. These responses were then given as input to the gradient-free optimisation, validated in Section 5.4.2, with the objective function being

$$\underset{\tilde{y}_{n,i}^H}{\text{minimise}} \left\| \mathbf{X}_{AC,i+1} - \bar{\mathbf{X}}_{AC,i+1} \left(\tilde{y}_{n,i}^H(t) \right) \right\| \quad (5.33)$$

where $\mathbf{X}_{AC,i+1} = \{\alpha_{i+1}, \dot{\alpha}_{i+1}, \xi_{i+1}, \dot{\xi}_{i+1}\}$ is the vector of the measured quantities and $\bar{\mathbf{X}}_{AC,i+1} = \{\bar{\alpha}_{i+1}, \bar{\dot{\alpha}}_{i+1}, \bar{\xi}_{i+1}, \bar{\dot{\xi}}_{i+1}\}$ is the vector of the predicted quantities.

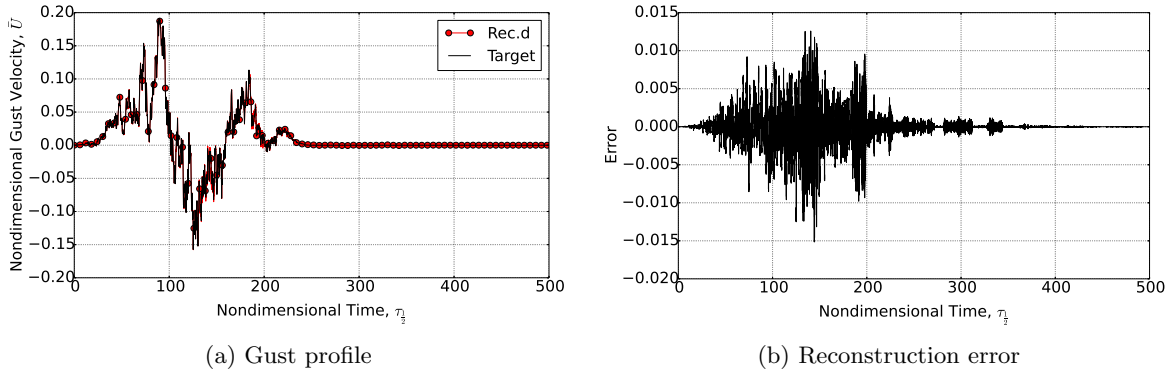


FIGURE 5.29: Comparison between a time-approximated Von Kármán gust and its reconstruction obtained using the aerofoil nonlinear model. The error plotted in (b) is the delta between the target function and the reconstructed one.

Figure 5.29 demonstrates that, despite the highly irregular shape of the original gust profile (black solid line) and the nonlinear terms included into the model ($\beta_\alpha = 3.0$ that stiffens the plunge response of the model), a very good reconstruction was obtained for this application, where the amplitude of the absolute error hardly exceeded 5% of the maximum amplitude of the

gust. Besides, the response time histories that result from the reconstructed gust are compared in Figure 5.30 to the target responses and, arguably, a virtually perfect match is obtained for both the pitch and the plunge responses.

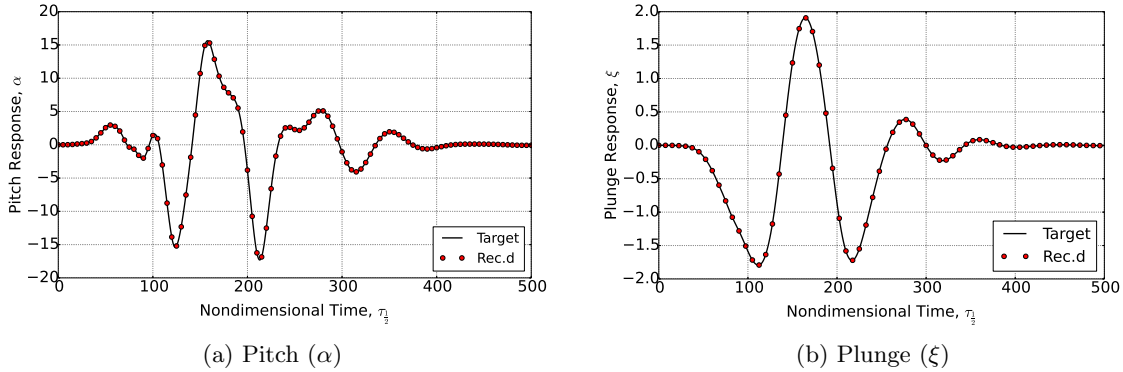


FIGURE 5.30: Comparison between target and reconstructed pitch (5.30a) and plunge (5.30b) responses to a Von Kármán gust.

If on one hand the accuracy of the results can hardly be discussed, on the other hand, when analysing the performance of the optimisation algorithm, the numbers do not open opportunities to real-time applications of the method (e.g. in-flight gust loads reconstruction). In fact, the optimisation took an average of 35 iterations per gust point to converge, with an average of 80 direct function calls and approximately 5 minutes to complete. As discussed in 5.2, this is indeed one of the main limitations of gradient-free methods that would require a dramatically efficient model to improve their overall performance. The latter is a condition that, however, may be satisfied by the use of reduced order models that would improve the performance of the optimisation framework also when using gradient-free methods.

5.7.2 3D beam-stick model

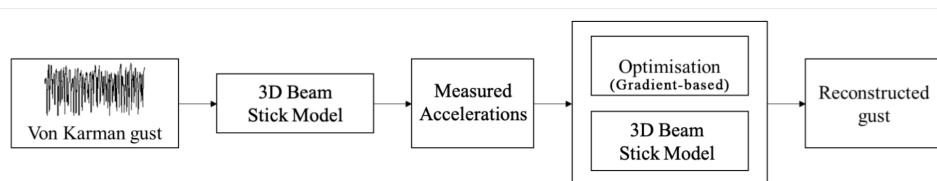


FIGURE 5.31: Gust reconstruction diagram for the 3D beam-stick model.

The transformation of the 3D beam-stick model to the time domain discussed in Section 3.5 allows for an input gust to be defined in the form of a time history. Similarly to the previous application, Equation 5.32 is used to obtain a random Hann-windowed signal that represents

a more realistic gust event. This time the amplitude of the generated signal is considered to be dimensional and its maximum and minimum values are bounded to $\pm 25\text{m/s}$. At the beginning of the simulation, the beam-stick model is trimmed in its steady-state flight conditions to represent an aircraft flying in cruise conditions at Mach 0.60 and at an altitude of 7,620 m (25,000 ft). The acceleration time history resulting from the gust event is then extracted from the node at the centre fuselage section of the beam stick model, which is the approximate c.g. location of the aircraft, and it is given as input to the gradient-based optimisation described in Section 5.3, as shown in Figure 5.31. Here, the objective function is defined as

$$\underset{\mathbf{w}}{\text{minimise}} \quad \left| n_z - \tilde{n}_z \left(\tilde{y}_n^H(\mathbf{x}_g) \right) \right| \quad (5.34)$$

where n_z is the measured acceleration time history and \tilde{n}_z the acceleration response to the reconstructed gust $\tilde{y}_n^H(\mathbf{x}_g)$, parameterised using RBFs as per Equation 5.13, i.e.

$$\tilde{y}_n^H(\mathbf{x}_g) = \sum_{i=1}^N w_i \phi(\|\mathbf{x}_g - \mathbf{c}_i\|) + p(\mathbf{x}) \quad (5.35)$$

where w_i are the RBF weights which form the design variables of the vector \mathbf{w} of the objective function.

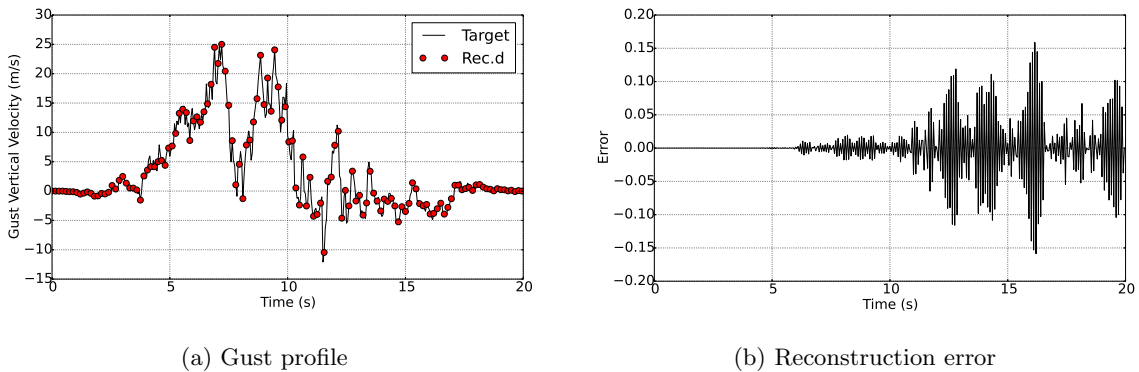


FIGURE 5.32: Comparison between a time-approximated Von Kármán gust and its reconstruction obtained using the 3D beam-stick model. The error plotted in (b) is the delta between the target function and the reconstructed one.

Figure 5.32 shows the results of the gust reconstruction process, with the reconstructed gust profile compared to its target on the left and the absolute error between the two profiles on the right. Compared to the magnitude of the gust profile, this error is remarkably small as it only increases to a maximum 0.6% just before the gust velocity stabilizes back to zero. This also reflects into a perfect match of the reconstructed acceleration response w.r.t. the measured

(target) response, as shown in Figure 5.33, which was obtained by feeding the reconstructed gust profile back as input to the beam-stick model.

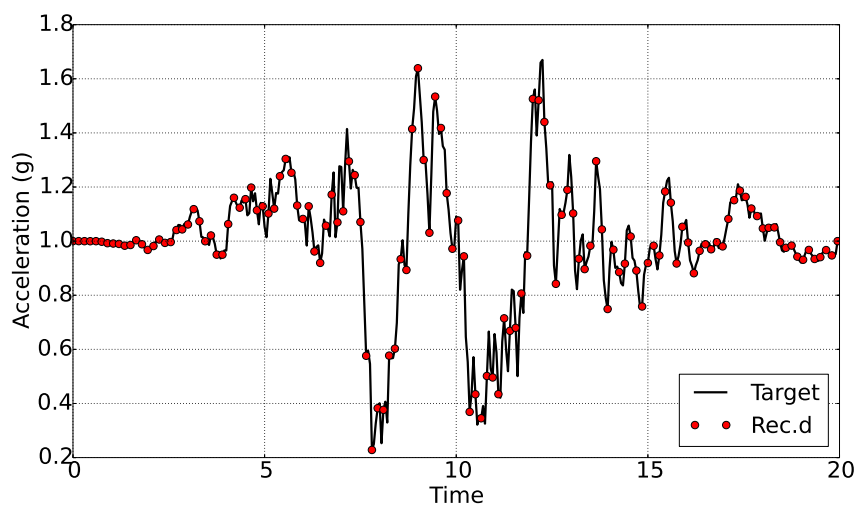


FIGURE 5.33: Comparison between target and reconstructed acceleration response to a von Kármán gust.

In terms of performance, the gradient-based optimisation took approximately 1 day to converge to a solution which is accurate up to the order of 10^{-4} and which was obtained using an increased number of basis functions (i.e. 59). This translates to 295 objective function calls and 59 gradient evaluations and an average time of approximately 5 minutes per function call. Similar results were also obtained when using HHBFs for the parameterisation of the gust and, therefore, formed no argument for a cross comparison or for a sought improvement. Instead, the time required to evaluate each solution, i.e. each objective function call, is what limited the performance of the reconstruction framework. This limitation is purely due to the way this model is scripted today; in fact, the simulation dramatically slows down when the gust is provided as input in the form of an external file due to the implementation of libraries required to import data from sources external to the program. Arguably, the quality of these results, once again, demonstrates the applicability of this framework to gust profiles of virtually any shape.

5.8 Conclusions

A gust reconstruction framework based on gradient-free optimisation and one designed around gradient-based optimisation were presented in this chapter and validated against known input gusts using the aeroelastic models described in Chapter 3. Good results were obtained with the gradient-free method applied to the 2DoF nonlinear model but it was shown that this method requires a high number of direct function calls and its performance is, thus, limited by the efficiency of the model used. The ULVM, CFD and ROM models were, instead, used for the validation of the gradient-based optimisation framework and their performance is summarised in Table 5.1 and along with the performance of the gradient-free method. Both parameterisation methods presented in this thesis, i.e. RBFs and HHBFs, were used for the validation exercise and 14 basis functions were found to be the minimum requirement for the ULVM and the CFD models to achieve an accuracy of the order of 10^{-4} . With an equal number of basis functions, the reconstruction obtained using HHBFs as the parameterisation method took longer to converge and required a higher number of calls to the objective function. The efficiency of reduced order models (already discussed in Chapter 3), however, opened to the opportunity of increasing the sought accuracy to 10^{-7} and to optimise the number of basis functions required for both the RBF method and the HHBF one. It was found that if on one hand an increased number of RBFs was required, on the other hand, 8 HHBFs were enough to converge to that level of tolerance. As a result, this reduced the number of objective function calls and the time required to converge to the optimum; thus, it may suggest that for a lower number of minimum basis functions required, HHBF may perform better than RBF. However, if the minimum number of basis functions required is high, RBF may represent a better compromise between accuracy and performance. This was the case, for example, for the application of the gust reconstruction framework to the more realistic gust shown in this chapter, where 59 basis function were needed to achieve a good reconstruction.

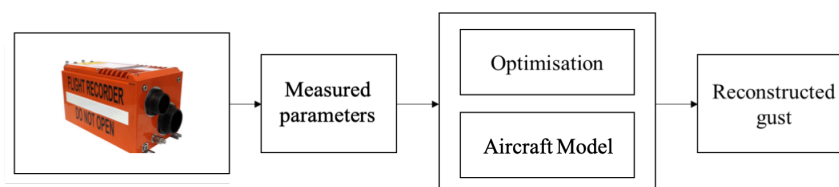


FIGURE 5.34: Gust reconstruction diagram for in-service applications.

As described in this chapter, random gust profiles were obtained analytically over time approximations of the von Kármán PSD to represent more realistic gust events. However, the two

	<i>Validation of gradient-free optimisation</i>		<i>Validation of gradient-based optimisation</i>					
	2DoF (Linear)	2DoF (Nonlinear)	ULVM		CFD		ROM	
Parameter.	-	-	RBF	HHBF	RBF	HHBF	RBF	HHBF
N. of basis functions	-	-	14	14	14	14	18	8
Calls to obj. functions	24 (on average, per time step)	35 (on average, per time step)	26	207	49	72	600	489
N. of gradient evaluations	-	-	4	29	19	29	22	47
Time to converge	~1 min	~2 min	~20 min	~3 hrs	~115 hrs	~175 hrs	~55 s	~40 s
Obj. function tolerance	10^{-9}	10^{-8}	10^{-4}		10^{-4}		10^{-7}	

TABLE 5.1: Performance of the models used for the validation of the optimisation framework.

optimisation methods presented are meant to be applied directly to data extracted from the flight data recorders of in-service aircraft, as represented in Figure 5.34. An attempt of this type was done by the author on a limited set of gust data available from a large commercial aircraft and it was shown that, despite some pre-processing of the data was necessary due to the low sampling rate of the FDR, the results obtained showed good agreements with the response of the aircraft in vertical speed, pitch and pitch rate [6]. Finally, it was shown that ROMs appear to be applicable for the reconstruction of gust profiles directly from (or even during) in-service events. Higher order models, instead, could be used to extract and collect valuable information for aircraft design.

The next chapter aims to exploit the application of the gust reconstruction method in the framework of the worst case gust prediction, where worst case conditions are sought by aircraft designers to certify that their products are designed to resist the high dynamic loads resulting from these turbulent gusts.

Chapter 6

Worst case gust prediction

A potential application of the reconstruction method presented in the previous chapter is discussed here. Aircraft designers need to identify worst case gusts because they need to certify that aircraft are designed to resist the dynamic loads associated with these turbulent gusts. Hence, a new method for the definition of the worst case gust is proposed in this chapter. This method shall follow directly from the reconstruction framework as large amount of gust data is required for the successful application of the method. First, for demonstration purposes, a collection of gust profiles is made available through the time approximation of the von Kármán power spectral density. The mathematical singular value decomposition technique is then used to decompose the gust matrix built over the n approximated gust profiles. From this decomposition, an approximation of the gust matrix can be obtained by considering only the first k non-negligible singular values. A surrogate model is then constructed that relates the coefficients U of the singular values to the *max* and *min* values of the aircraft gust response. This surrogate model is then used into an optimization framework to find out the combination of coefficients that generates the worst case gust.

6.1 Introduction

Certification specifications require that enough points, on or within the boundary of the design envelope (Section 2.2), are investigated to ensure that the most extreme loads for each part of the aircraft structure are identified. The flight conditions and maneuvers, which provide the largest aircraft loads, are not known a-priori but gusts are often the most critical load case for

structural design and are also the main fatigue loading source for the majority of the structure. Several studies can be found in the literature that focused their attention on the identification of “worst-case” gusts [81–83] i.e. the gust shape as well as the flight and atmospheric conditions that generate the worst aircraft response. In 1990, the concept of matched filter theory was introduced and it was shown that it could be applied to the analysis of time-correlated, worst-case gust loads for linear aircraft models [82]. The method consisted in evaluating, initially, the impulse response of the aircraft system and then to invert it in time and return it back as input to the system, in order to maximize its response (Figure 6.1a). This gust-load analysis method worked well only for linear systems as it could take advantage of the principle of superposition. However, the method is not applicable to nonlinear systems.

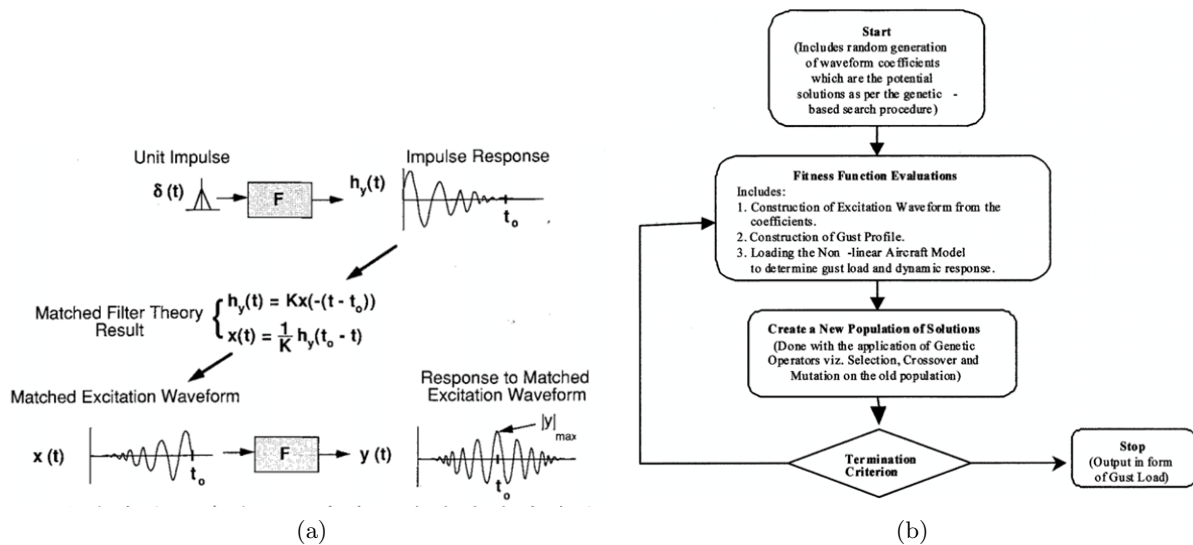


FIGURE 6.1: Left: overview of the matched filter theory proposed by Pototzky [82]. Right: overview of the evolutionary algorithm method proposed by Karr [81].

In 2004 Karr et al [81], in their approach for the discovery of the worst-case gusts, used a combination of three elements: the statistical discrete gust (SDG) model [84] to simulate a turbulent wind gust, a nonlinear aircraft model to evaluate the effect of a gust on aircraft responses, and an evolutionary algorithm (EA) to search for a worst-case gust. This formulation has at least two very desirable characteristics. In fact, on one hand, the approach is feasible for both linear and nonlinear aircraft models. On the other hand, the definition of the worst-case gust is not unique in the sense that the designer can select different criteria when determining the properties of the worst-case gust. The fundamental methodology was implemented with the genetic-based search procedure that maximizes the peak values in a given aircraft load quantity, thereby determining the associated critical gust profile. The search procedure used in their study followed the

flowchart of Figure 6.1b. The first component generates an excitation waveform from a gust generation function. The generated waveform is then passed through the second component which consists of a “modified von Kármán” gust pre-filter for conversion into atmospheric turbulence. The generated atmospheric turbulence serves as a forcing function to the third component which is a nonlinear model of a particular type of aircraft. The dynamic response, which occurs in the form of each of the four loads (wing root bending moment, wing root torque, engine lateral acceleration, or c.g. normal acceleration), is then input into the fourth component, which is an optimization loop that searches for the worst-case gust. Finally, the output is given in the form of gust loads. In 2011, Khodaparast et al also tried to efficiently and accurately determine the worst “1 – cosine” gust loads case [83]. They used surrogate modeling and optimization techniques applied across a range of different flight conditions (altitude, speed, fuel condition, center of gravity position) to enable the global worst-case to be found rapidly. They showed that by using certain data sets it is possible to deduce reasonably accurate surrogate models for the worst-case gust loads with relatively few design test cases (Figure 6.2).

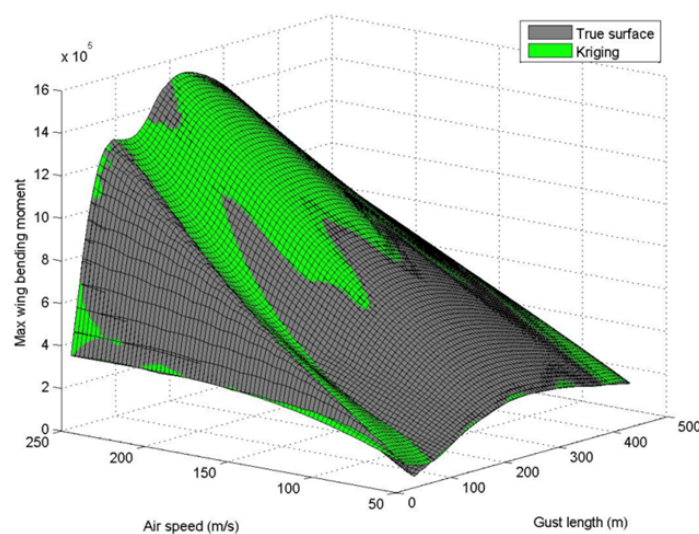


FIGURE 6.2: Response surface (surrogate) model used for the search of the worst-case gust [83].

The current practice in industry to determine worst case conditions is the use of two-dimensional load envelopes, also known as ‘phase plane’ or ‘potato plots’ [1]. Typically, load quantities such as shear force, bending moment and torque, or other interesting quantities (IQs), are extracted from the full set of available load cases¹ and are plotted against each other at a particular load

¹A set of load cases is a database that is made up by thousands of load cases that are usually investigated to cover all the possible load combinations that an aircraft is expected to experience during its life cycle. These include static and dynamic manoeuvres, gusts and ground manoeuvres for different flight/ground conditions and for different aircraft parameters.

station (e.g. wing root) to identify the load cases that generate the worst combination of shear, moment and torque. This process helps to identify the sets of correlated loads that correspond to the critical design cases for different parts of the aircraft structure. In practice, each load case is represented by a single point on the phase plot; an envelope is then traced that goes through the most external points on the plot as these are the major points of interest. It is important to note that the load cases forming the envelope are not necessarily the maximum or the minimum of one of the two load quantities but, in combination, represent the flight conditions that are critical for the design of the particular component. A typical example of phase plane is shown in Figure 6.3, where shear and torque are plotted against each other for a generic load station and where the blue solid line marks the envelope drawn from the load cases that are considered critical for design.

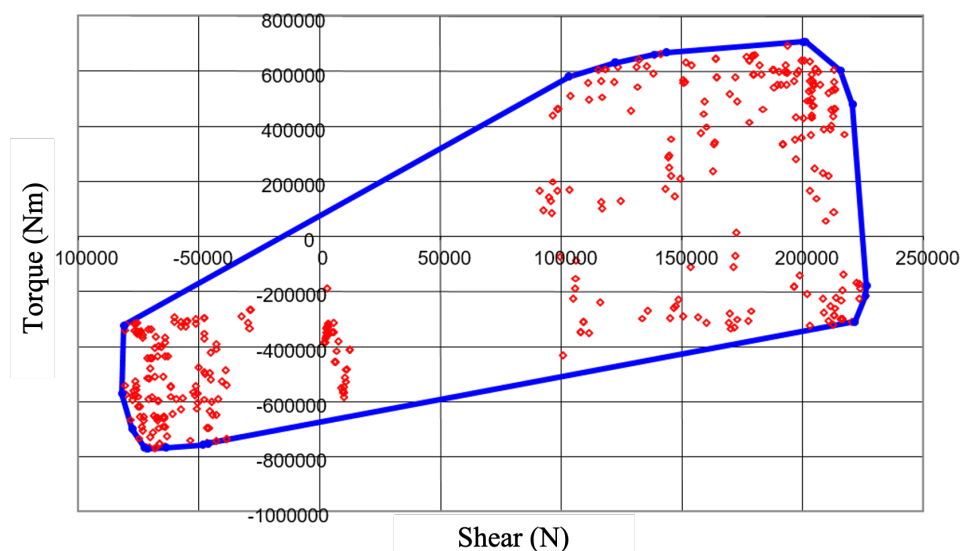


FIGURE 6.3: Two-dimensional loads envelope at a generic load station.

The study presented in this chapter follows from the reconstruction frameworks proposed earlier in this work and it is closely related to the methods proposed by Pototzky and Khodaparast. It is structured around three objectives. The first objective aims to apply the singular value decomposition (SVD) technique to decompose a gust matrix built over N_g provided² gust profiles. The second objective is to construct a surrogate model that associates the weighing coefficients

²Ideally, the gust velocity profiles required to build the matrix shall be provided by means of reconstruction algorithms (using the two methods proposed in the previous chapters of this work) applied to flight data recorders. However, not enough data was available to the author at the time of this publication and theoretical data was used instead.

of the SVD to the *max* and *min* values of the aircraft gust response. The third and last objective is to incorporate the surrogate model into an optimization framework to find out the combination of coefficients that generate the worst-case gust.

6.2 Methodology

Figure 6.4 provides a schematic of the method used in this study. The SVD technique is used to decompose the gust matrix \mathbf{W}_g into a coefficient matrix and a subset of “characteristic” gust shapes. A surrogate model is built on a training set where for each combination of coefficients correspond the *max* and *min* value of the aircraft response \mathbf{Z}_g (e.g. bending moment, displacement, lift, etc.). The surrogate model is then used within an optimisation framework to find out which combination of coefficients maximise the response (e.g. bending moment). Finally, the resulting coefficients are multiplied by the fixed set of characteristic shapes to find out what makes the “worst case” gust.

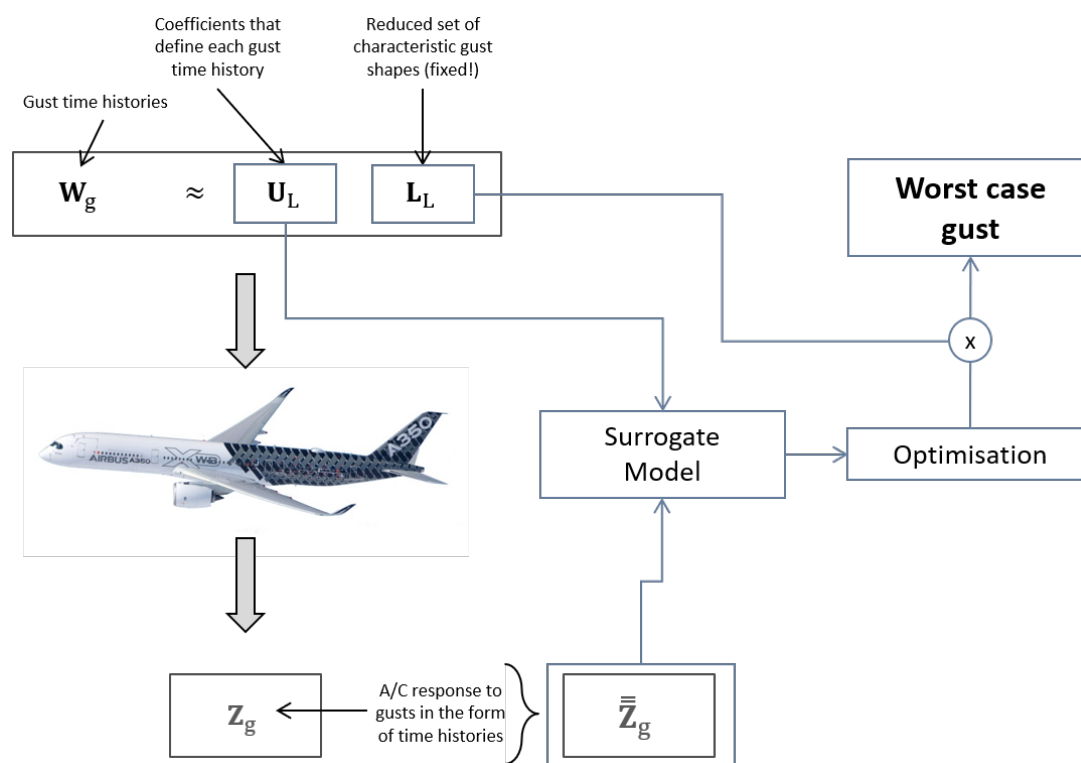


FIGURE 6.4: Overview of the SVD application to the worst-case gust problem.

6.2.1 SVD for gust matrix decomposition

Singular Value Decomposition (SVD) is a matrix decomposition technique that can be used to identify trends, patterns or consistencies effectively and is widely used in many engineering fields. It was recently used in a work between AIRBUS and Queen's University Belfast to obtain a concise representation of characteristic loads cases where less than 50 SVD-derived loads were used to reconstruct the full A350 XWB loads envelope [85]. SVD was also used in conjunction with surrogate models to enable a fast computation of correlated loads envelopes in systems where the effects of changes in design parameters need to be considered [86].

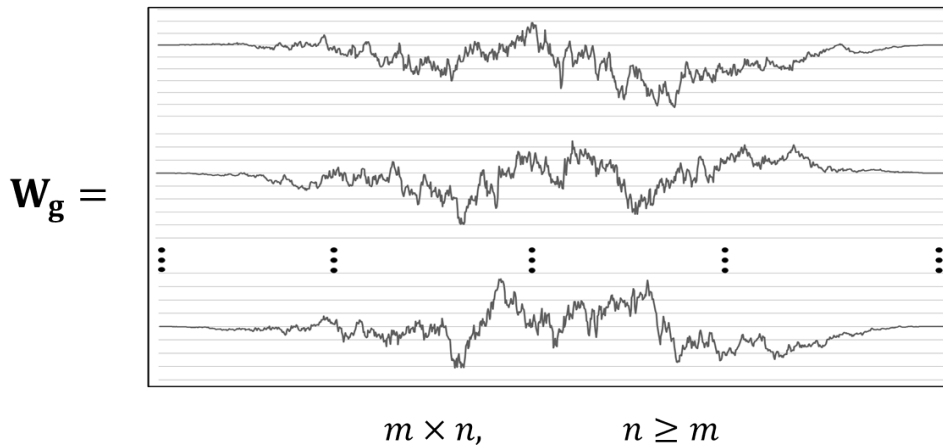


FIGURE 6.5: Graphical representation of the gust matrix \mathbf{W}_g .

Here, the Singular Value Decomposition technique is used to decompose a $n \times m$ gust matrix \mathbf{W}_g where n is the number of gust profiles and m is the length in time steps of each profile (must be equal for all). The decomposition generates 3 matrices \mathbf{U} , $\mathbf{\Sigma}$ and \mathbf{V} , such that

$$\mathbf{W}_g = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T \quad (6.1)$$

where $\mathbf{\Sigma}$ is a diagonal matrix whose diagonal includes the singular values, \mathbf{V}^T is the transpose of the unitary matrix \mathbf{V} and \mathbf{U} is a matrix of coefficients that multiplied by the so-called bases ($\mathbf{\Sigma} \mathbf{V}^T$) reconstructs the original set of data (\mathbf{W}_g). Henceforth, an approximation of the gust matrix can be obtained by considering only the first k non-negligible singular values such that

$$\mathbf{W}_g \approx \mathbf{U}_L \mathbf{L} \quad \text{with} \quad \mathbf{L} = \mathbf{\Sigma}_L \mathbf{V}_L^T. \quad (6.2)$$

where the subscript L indicates that a reduced set of singular values is considered. Here, \mathbf{L} represents the matrix of characteristic shapes where each row can be interpreted as a prescribed *gust mode*. The accuracy of the approximation can be assessed by comparing the sum of the total number of singular values in Σ with the sum of the first k non-negligible singular values in Σ_L ; the closer this ratio to 1, the better the approximation.

6.2.2 Surrogate model of the aircraft gust response

Surrogate models are defined as analytical models that approximate the input/output behavior of a complex system. They are constructed over a reduced set of training data that is obtained by running expensive simulations at particular points of interest in the design space (Figure 6.6).

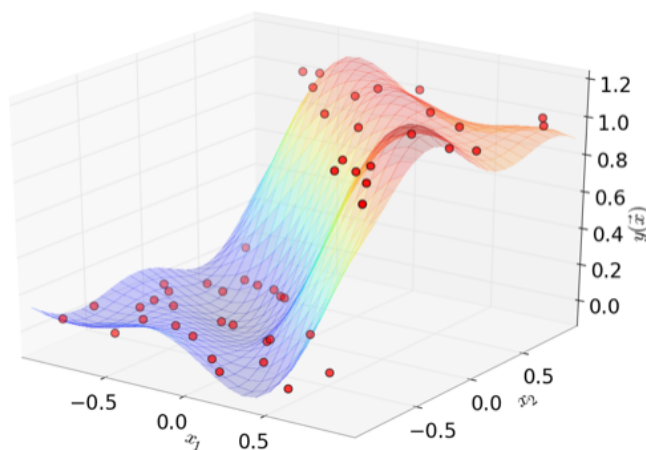


FIGURE 6.6: Sample representation of a surrogate model, also known as response surface. The red dots represent the training data points. Source: <https://www.datadvance.net>

In this study, a surrogate model is constructed to associate each singular vector of the SVD-reduced \mathbf{U}_L matrix to the maxima and minima values of the aircraft response of interest (accelerations, bending moments, torque, shear, etc.). The construction is carried out by an automatic algorithm³ that selects the most suitable modeling technique available (e.g. Kriging or GP based methods, Neural Networks, Regression Tree, Polynomial RBFs and others) for the considered training dataset. The automatic selection is based upon an internal decision tree that considers as criteria the dimensions of the training dataset and whether the state of the following switches, linear mode (Lin), exact interpolation mode (Int) and accuracy evaluation (AE), is set to *on* or *off*. The standard modeling techniques available in the toolset are: *Linear Regression*

³*Datadvance* MACROS Generic Toolkit.

(LR), Splines with Tension (SPLT), High Dimensional Approximation (HDA), Gaussian Processes (GP), Sparse Gaussian Processes (SGP), Tensor Approximation (TA) and combinations of these⁴. Figure 6.7 provides a combined representation of the criteria driving the internal decision tree and the techniques used for the construction of the models.

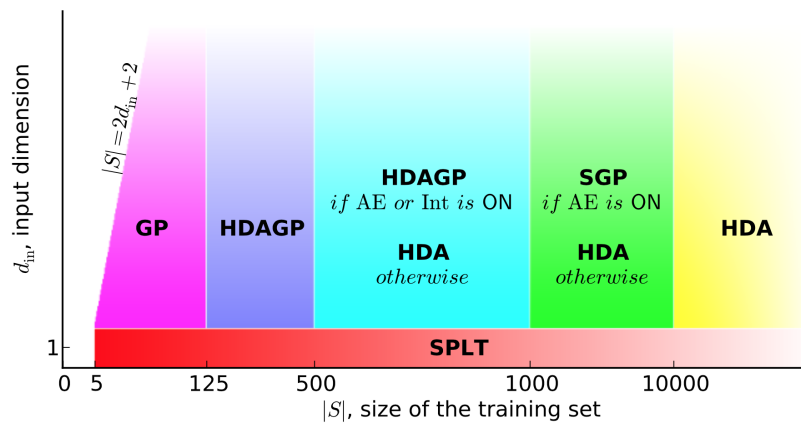


FIGURE 6.7: The “sample size vs. dimensions” diagram of default techniques in MACROS GT Approx. Source: Datadvance MACROS GT Approx User Manual.

Once obtained, the surrogate model returns an approximation of the maximum and minimum aircraft response for any new combination of k input coefficients that did not form part of the training dataset (Figure 6.8).

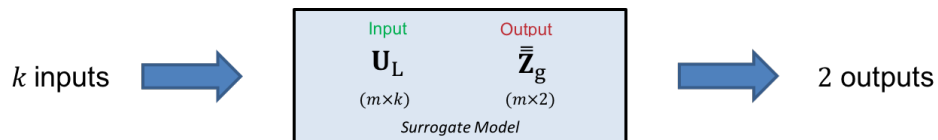


FIGURE 6.8: Definition of surrogate model.

But, it is obvious that any surrogate model needs to be validated before it can produce meaningful results and, surely, before it can be plugged within an optimisation framework. Indeed, several methods exist for the validation but, arguably, the most straightforward is a (internal) cross-validation of the outputs of the surrogate model obtained using the same input data that originally defined the training set. However, a more accurate validation can be achieved by considering additional points that were not part of the initial training data. In such cases, measures of the accuracy are given by the root mean square error (RMSE) and the maximum absolute error (MAX). These represent, respectively, an index of overall accuracy of the model

⁴A description of the mathematics behind the construction of the surrogate models is out of the scope of this work and the reader is invited to consult the existing literature and/or the *Datadvance* MACROS Generic Toolkit user manual for more information.

and an index for its local accuracy and are defined mathematically as

$$RMSE = \sqrt{\frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{m}} \quad \text{and} \quad MAX = \max |y_i - \hat{y}_i|, \quad i = 1, \dots, m$$

where m is the number of validation points and \hat{y}_i is the surrogate approximation of the actual value y_i . The lower the magnitude of $RMSE$ and/or MAX , the more accurate the surrogate model. Additionally, the R square measure can be used i.e.

$$R = 1 - \frac{\sum_{i=1}^m (y_i - \hat{y}_i)^2}{\sum_{i=1}^m (y_i - \bar{e})^2}$$

where \bar{e} is the mean of the observed values at the validation points.

6.2.3 The optimisation framework

The surrogate model introduced in the previous paragraph is used into an optimisation framework to find out the combination of \mathbf{U} coefficients that generates the worst case gust. The design variables of the optimisation are the coefficients of the SVD-reduced matrix \mathbf{U}_L and the aim of the objective function is to maximize⁵ the aircraft response \mathbf{Z}_g i.e.

$$\underset{\mathbf{U}_L}{\text{minimize}} \quad -\bar{\bar{Z}}_g \quad (6.3)$$

where

$$\bar{\bar{Z}}_g = \max \left(|\min(\mathbf{Z}_g)|, |\max(\mathbf{Z}_g)| \right) \quad (6.4)$$

Although the design variables are unbounded, the optimisation problem is constrained such that the resulting gust velocity profile does not exceed user defined maximum (U_g^U) and minimum (U_g^L) limits, i.e.

$$U_g^L \leq \mathbf{U}_L \mathbf{L} \leq U_g^U. \quad (6.5)$$

Figure 6.9 gives an overview of the optimisation framework. In sequence, an initial set of k \mathbf{U}_L coefficients is provided as input to the surrogate model. After the latter is executed, the maximum and minimum values of the aircraft response are returned and then passed to the objective function defined in Equation 6.3. The same process continues iteratively until it finally converges, in which case the resulting k \mathbf{U}_L coefficients are multiplied by the characteristic gust

⁵In optimisation problems, the maximization of the objective function can be obtained by solving the inverse problem i.e. by minimizing the same objective function preceded by the negative sign.

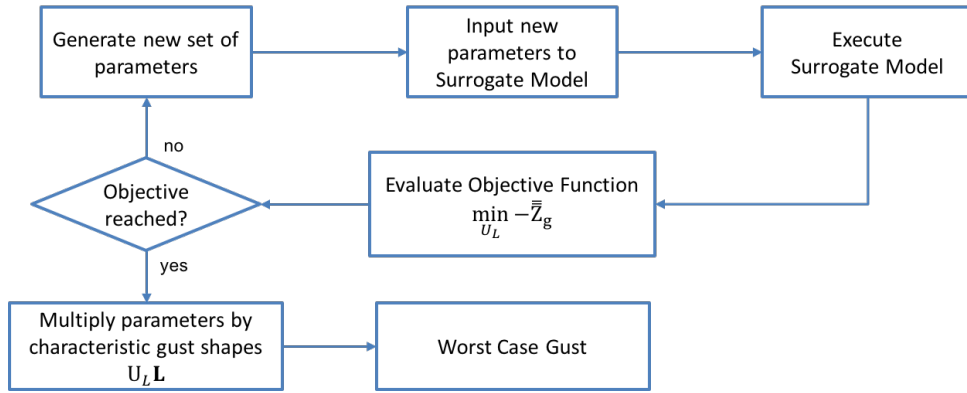


FIGURE 6.9: Flowchart of the optimisation used for the identification of the worst-case gust.

shapes \mathbf{L} . This ultimately results into the velocity profile that defines the worst case gust. As per the construction of the surrogate model, the optimisation algorithm is selected automatically by the optimisation toolbox⁶ using an internal decision tree which is based on the size of the problem.

6.3 Worst case gust prediction using the 3D beam-stick aircraft model

For this case study, the gust database \mathbf{W}_g was obtained by collating a series of 200 gust velocity profiles ($N_g = 200$) built over time approximations of the von Kármán PSD, following the method described in Section 5.7. Each of these gust velocity profiles was given as input to the 3D-beam-stick model described in Section 3.5, which assumed the following aircraft parameters⁷:

- mass of the A/C equal to 187.5 tonnes;
- wing span equal to 57.82 m;
- wing chord equal to 6.07323 m;
- free-stream velocity (V_∞) equal to 185.8 m/s;
- Mach number equal to 0.560;
- altitude equal to 7,620 m.

⁶*Datadvance* MACROS Generic Toolkit.

⁷The assumed aircraft parameters are representative of a commercial aircraft in cruise conditions.

The response time histories to each gust were then stored as rows into the matrix \mathbf{Z}_g , in the form of accelerations (nodal displacements, velocities and/or loads could also be used). The singular value decomposition was applied to the gust matrix \mathbf{W}_g , which had dimensions $\{N_g \times f_s T\}$, and resulted in

$$\mathbf{W}_g \underset{(N_g \times f_s T)}{=} \underset{(N_g \times N_g)}{\mathbf{U}} \underset{(N_g \times f_s T)}{\mathbf{\Sigma}} \underset{(f_s T \times f_s T)}{\mathbf{V}^T} \quad (6.6)$$

with $N_g = 200$. An approximation of \mathbf{W}_g could then be obtained by considering the first 50 non-negligible singular values, such that

$$\mathbf{W}_g \underset{(N_g \times f_s T)}{\approx} \underset{(N_g \times k)}{\mathbf{U}_L} \underset{(k \times f_s T)}{\mathbf{L}} \quad \text{with} \quad \underset{(k \times f_s T)}{\mathbf{L}} = \underset{(k \times k)}{\mathbf{\Sigma}_L} \underset{(k \times f_s T)}{\mathbf{V}_L^T} \quad (6.7)$$

with $k = 50$. Figure 6.10 shows the first 25 bases resulting from the SVD application, which represents those that have the highest influence in the characterisation of the gust velocity profiles. Arguably, the first basis, or *gust mode*, has similar characteristics to the typical ‘1 – cos’ gust shape used for loads’ certification and is, indeed, the one that drives the shape of most of the gust velocity profiles. The following bases, instead, have increasing frequency but lower power, as shown by the plots of singular values and cumulative energy of Figure 6.11. When

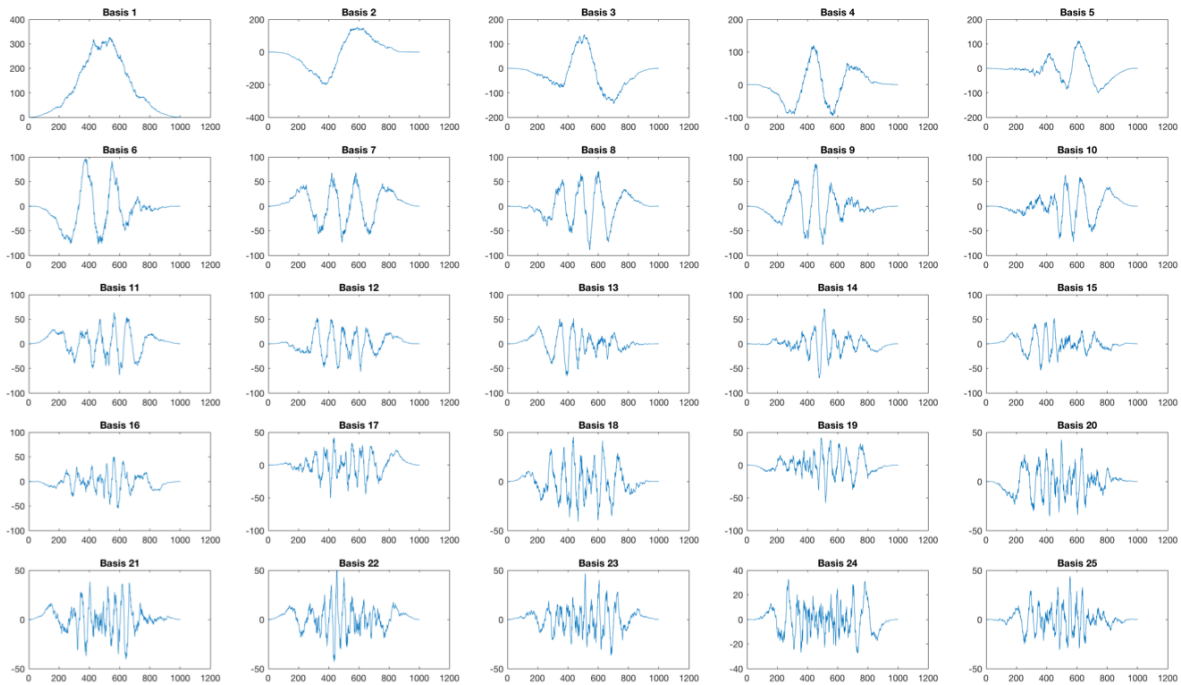


FIGURE 6.10: First 25 SVD-reduced characteristic gust shapes.

compared to the full set of singular values, the choice of 50 singular values provided an accuracy

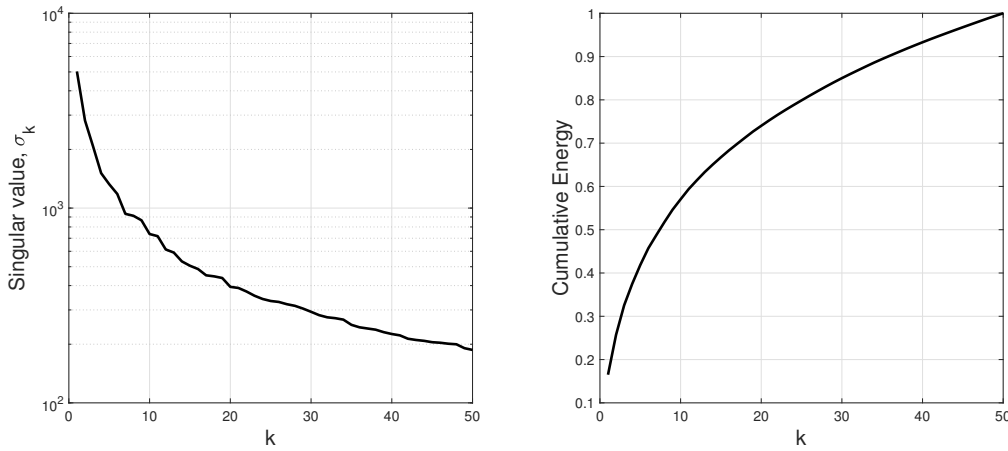


FIGURE 6.11: Singular values σ_k (left) and cumulative energy contained in the first k modes.

for the approximation equal to

$$\frac{\Sigma \text{diag}(\Sigma_L)}{\Sigma \text{diag}(\Sigma)} = 69.4\%. \quad (6.8)$$

Despite the not-too-high accuracy, this provided an acceptable approximation for the purpose of this study, where the reduced set of singular values was only required to minimize the effective input dimensions of the surrogate model. Once established, the entire set of 50 characteristic gust shapes (\mathbf{L}) remained fixed throughout the continuation of the problem and was later used as an input required to obtain the shape of the worst case gust. The matrix of coefficient obtained (\mathbf{U}_L , with N_g rows and k coefficients per row), instead, was used together with the minimum and maximum values⁸ of the considered interesting quantity, i.e. N_z acceleration, to form the training dataset used for the construction of the surrogate model. The training algorithm used for the construction of the surrogate model was the High Dimensional Approximation with Gaussian Process (HDAGP), automatically selected by the MACROS GT Approx tool. HDAGP is a flexible nonlinear approximation technique, with a wide applicability in terms of space dimensions, that extends the applicability of the GP technique to spatially inhomogeneous functions or functions with discontinuities by using the HDA-based non-stationary covariance function⁹. The workflow followed by this technique is summarised below.

- The training set is divided into proper training data and validating parts.
- Base functions (which include also sigmoid and gaussian functions) are selected from a library and a basic approximator is initialised.

⁸One set per gust case.

⁹The mathematical functions and further details of this algorithm can be found in the *Datadvance* MACROS Generic Toolkit user manual available under license.

- The basic approximator is trained using an adaptive strategy which controls the type and characteristic parameters of the used optimisation algorithm.
- Unnecessary base functions are removed during the post-processing of the structure of the basic approximator'
- Finally, a collection of basic approximators is constructed using an advanced algorithm for boosting¹⁰ and multi-start.

The accuracy of the approximation constructed with HDAGP is directly proportional to the size of the training dataset, the larger the training dataset, the more accurate the approximation.

The resulting surrogate model had dimensions $[200 \times 52]$, where 200 is the total number of samples, 50 is the effective input size¹¹ and 2 is the size of the output. Within the training dataset, the coefficients of the \mathbf{U}_L matrix formed the input variables whilst the associated minimum and maximum acceleration values formed the output variables (refer to Figure 6.8). An indication of the accuracy of the surrogate model is given in Figure 6.12, where a comparison of actual and approximated values is given for the maximum and minimum N_z acceleration response.

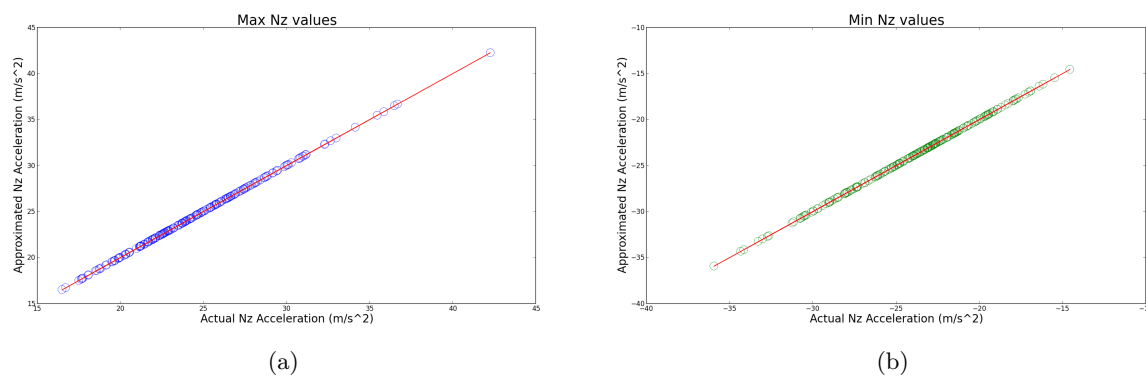


FIGURE 6.12: Comparison between actual and approximated values of N_z acceleration response.

This formed the internal validation of the surrogate model as the approximated values were obtained from the same input dataset used to build the surrogate. Nonetheless, when considering additional points, external to the training dataset, the accuracy given by the $RMSE$ and the MAX (defined in Section 6.2.2) was respectively equal to 0.021585 and 0.008211 for the

¹⁰In this context, boosting refers to the process of transforming a set of ‘weak learners’ into a ‘strong learner’. Weak learners are defined as classifiers that only have a weak connection to the true classification. On the other hand, a strong learner is a classifier that is well-associated to the true classification.

¹¹The term ‘effective’ takes into account a validation of the inputs that is performed within the surrogate model generation algorithm to eliminate duplicate or zero values.

maximum N_z values and 0.021365 and 0.005563 for the minimum N_z values, which are all an indication of the good quality of the surrogate model.

The last step of the process involved the optimisation routine described in Section 6.2.3, with the gust velocity profile constrained by Equation 6.5 with $U_g^L = -100$ m/s and $U_g^U = 100$ m/s. Based on the size and complexity of the problem, the optimisation algorithm selected automatically by the toolkit was the Sequential Quadratic Constrained Programming (SQ2P)¹². This resulted in the gust velocity profile of Figure 6.13 which had an absolute peak velocity of approximately 60 m/s and generated a peak acceleration of approximately 2.45 g (Figure 6.14a) and a maximum wing root bending moment of 1.78×10^6 Nm (Figure 6.14d). The correlated maximum shear force and torque resulted to be, respectively, 1.08×10^5 N and 8.86×10^5 Nm.

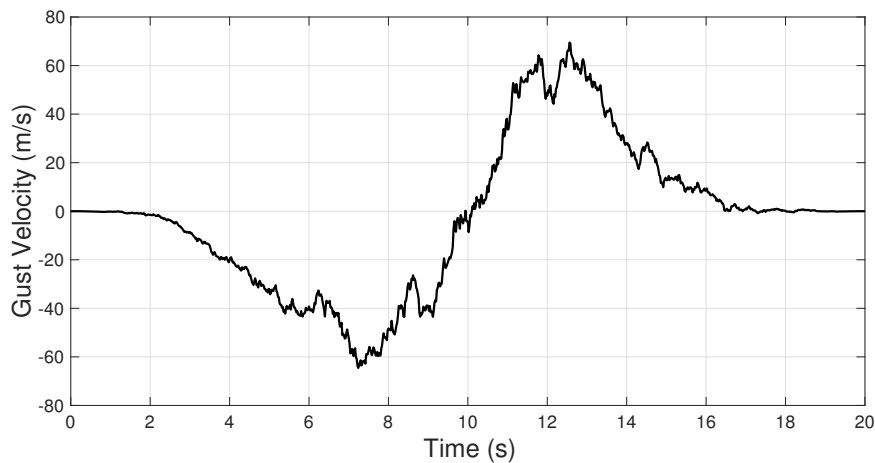


FIGURE 6.13: Worst case gust shape.

6.4 Conclusions

A new method for the definition of the worst case gust that combines SVD techniques, surrogate models and optimisation techniques, was proposed in this chapter. This method shall follow directly from the gust reconstruction framework proposed in this thesis, as this would be needed to generate the large amount of gust data required as input. However, not enough recorded data¹³ was available to the author at the time of this publication and, to demonstrate the functionality of the new method, theoretical approximations of realistic gust profiles were used instead. The latter were obtained from time approximations of the von Kármán PSD and the

¹²A description of this optimisation algorithm is out of the scope of this thesis and the reader is invited to consult the existing literature for more details.

¹³From flight data recorders.

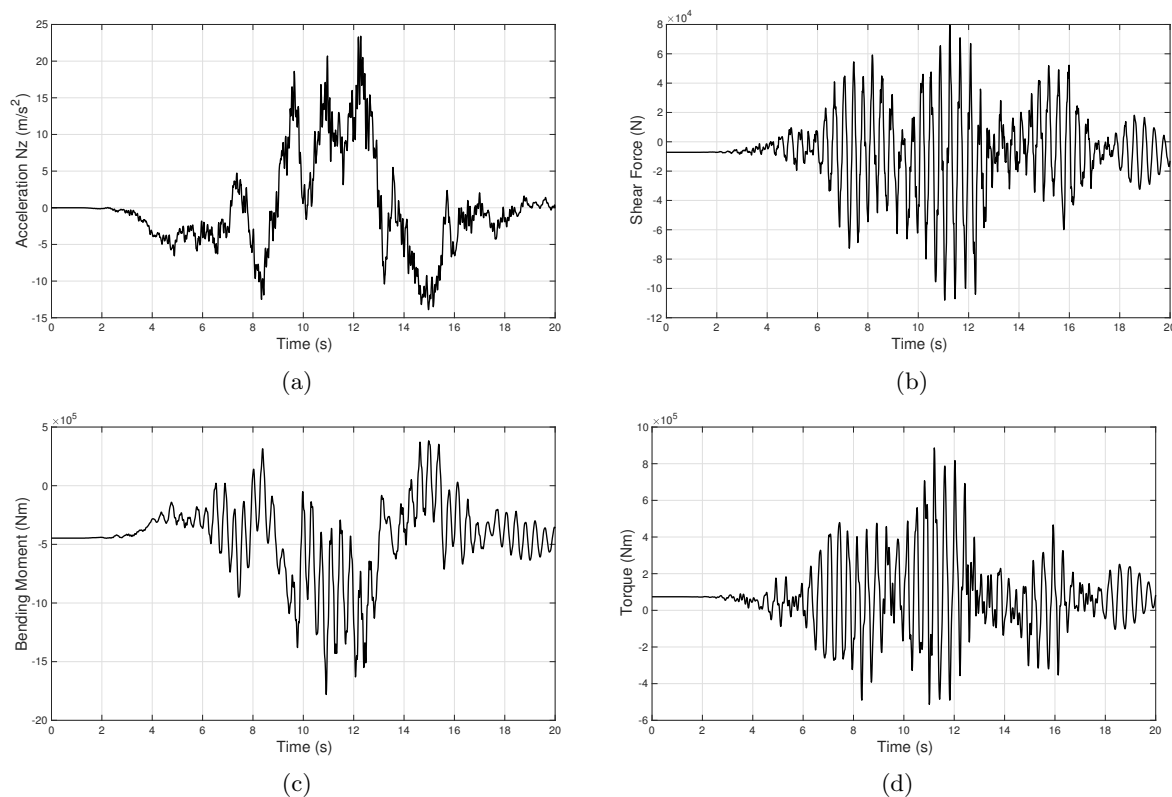


FIGURE 6.14: Acceleration response at the aircraft c.g. location (a) and shear force (b), bending moment (c) and torque (d) at the wing root.

results shown proved satisfactory. But, despite the fairly realistic values obtained as a result of this application, the gust profile of Figure 6.13 is by no means intended to be the absolute worst case profile that can be encountered by an aircraft in the atmosphere. In fact, the aircraft parameters listed in Section 6.3, which were used to model the aircraft response considered for the construction of the surrogate model, define only one single load case and, therefore, do not represent the full spectra of conditions that an aircraft encounters in flight. Additional load cases, of a number similar to those included in the phase plane approach described in Section 6.1, would need to be considered and the respective aircraft response included in the training dataset used for the construction of the surrogate model. Furthermore, it must not be underestimated that the gust profiles forming the initial database were purely approximations obtained from the von Kármán PSD. The underpinning intention, here, is to provide a method that gives a representative worst case gust that maximise the loads experienced by the aircraft structure and, particularly, by the wings. Ultimately, it was shown that:

- SVD could successfully be applied to a large matrix of gust profiles constructed over a range of von Kármán profiles approximated in time;

- A subset of 1/4 of the total number of singular values was used to construct an approximation of the gust matrix. This approximation was necessary to reduce the dimensions of the problem in order to obtain a more efficient surrogate model;
- The surrogate model was successfully built over a training dataset of 200 samples with 50 inputs (coefficients that multiply the characteristic gust shapes) and 2 outputs (*max* and *min* acceleration response at the A/C c.g.). It was then used within an optimisation framework to find out the combination of coefficients that generated the worst A/C response.

Future studies shall see the extension of this approach to cover the entire design space with the aid of an additional surrogate model. Also, and most importantly, the SVD technique shall be applied to a database of gust profiles reconstructed from real events i.e. using data from flight data recorders and with the aid of the reconstruction techniques introduced in Chapter 5.

Part III

In-Service Support Development

Chapter 7

A System of Systems

So far in this work, the reconstruction of gust loads has been approached from a purely aircraft-design point of view but, as mentioned in Chapter 1, its contribution plays an important role also in the framework of in-service incident support. In order to fulfill the remaining objectives of this work (Section 1.1) and to understand the wider picture of incident support, this chapter provides an analysis of the system from the systems engineering perspective. After a brief introduction on Systems of Systems Methodology, which presents the complexity of systems engineering, the application of Problem Structuring Methods, including Soft Systems Methodology and Hierarchical Process Modelling, is illustrated here in the framework of in-service support with the aim to achieve a well-defined problem situation.

7.1 Introduction

The growth of the civil aviation market predicted for the next 20 years (§1) is perceived as a risk for the in-service support activities executed in Airbus today. In order to understand what is already a complicated problem that could develop into a messy scenario, systems engineering practice provides the necessary tools to understand and define the complexity of the systems that sit behind such activities. These tools fall within the area of Problem Structuring Methods, which can be described as a collection of techniques used to define the problem situation and that assume all the relevant existing issues are somewhat undefined or unclear, and lack a unique, unconditional representation of the problem. The label Problem Structuring Methods, abbreviated as PSMs, began to be used in the early 1980s in the framework of *applied systems*

thinking [10]. Jackson and Keys in 1984 defined the Systems of Systems Methodology (SOSM) framework to best understand the developments that applied systems thinking achieved to counter the limitations and weaknesses of *hard systems thinking* [9]. As defined by Checkland [10], the latter is a group of systems methodologies that share several characteristics and that were born during the immediate aftermath of the Second World War, as means for tackling real-world problems by the optimisation of the systems' performance over clearly identified goals. However, criticism soon arose about the inability of this approach to handle problems of significant complexity, particularly when dealing with managerial situations and political issues, and by the 1970s the systems community found itself close to a crisis. It was to overcome this crisis that efforts were made in the development of the applied systems thinking methodology.

In the framework of SOSM, Jackson and Keys classify problem situations along two dimensions: *complexity* and *divergence of values and interests*; the first dimension is strictly related to the 'systems' that managers have to deal with; the second dimension relates to the extension of 'participants' who are in fact those that have diverging interests in the problem situation. Systems, on one axis, can be relatively simple or extremely complex; this depends on the number of elements, interrelationships and their relative nature, and the immobility or dynamism of the system. On the other axis, interrelationships among participants can be *unitary*, *pluralist* or *coercive*. Unitary relationships are those where all participants within the systems share common goals and take part to the decision-making process. Then, although being based on shared interests, pluralist relationships are open to debate, disagreement and conflict; however, compromises can be found among the participants who take part to the decision-making process. Coercive relationships exist among participants who share few interests and are solid on their diverging values and beliefs; compromises are not achievable. As shown in Figure 7.1, the combination of the two dimensions described above lead to six '*ideal type*' problem situations i.e. simple-unitary, simple-pluralist, simple-coercive and complex-unitary, complex-pluralist, complex-coercive. Figure 7.1 also shows three main streams that characterise the systems approach: functionalist, interpretive and emancipatory. In the **functionalist approach**, systems are independent of the observer and, as such, are analysed objectively; participants share common goals and objectives, as suggested by its location in the unitary area of the grid; this approach performs better for simple systems where divergence of interests is low but it can also deal with systems of relatively high technical complexity. The **interpretive approach** stream, as suggested by the name itself, sees systems subjectively, from the observer's point of view; these systems do not necessarily represent the reality as is but they depict a simplified

and customised version of it that helps the observer to better understand the interrelationships; such approach is suitable with moderate levels of human complexity but it doesn't perform well with systems of high technical complexity. Finally, the **emancipatory approach** also sees systems subjectively but, as opposed to the interpretive approach, views of the systems may vary amongst the observers, who are influenced by their role within the system and the resulting power; this approach struggles to adapt to systems with high levels of human and technical complexity.

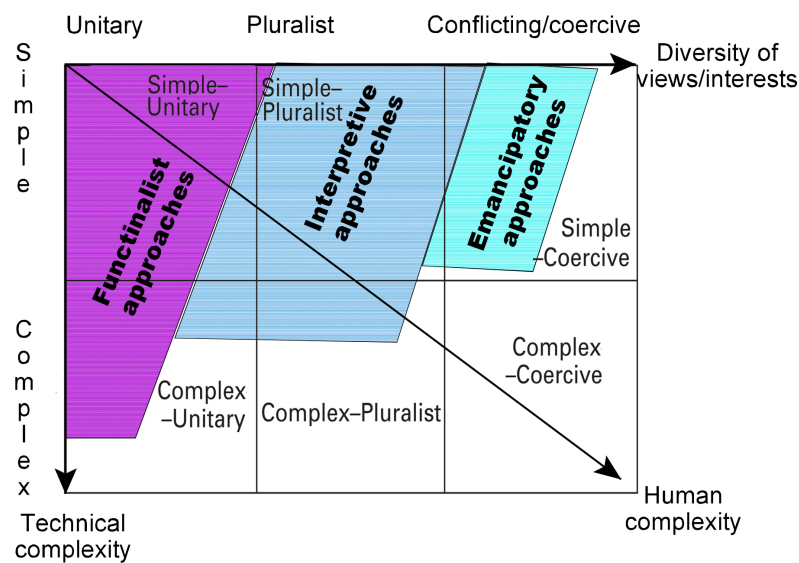


FIGURE 7.1: Problem situation classification and systems approaches. Adapted from Daellenbach, 2001 [87].

7.2 The in-service support system

In order to develop an understanding of the system of in-service support in a large aircraft manufacturing company such as Airbus, two problem structuring methods were attempted: the *Soft Systems Methodology* (SSM) and the *Hierarchical Process Modelling* (HPM), which are detailed in the following sections of this chapter. As mentioned in Chapter 1, the entity responsible for the analysis of aircraft incidents is the Loads & Aeroelastics department. Here, the analysis of the system's context is given by the tools' development perspective, where the participants or stakeholders involved were identified as

- Research & Development managers, responsible for budget and resources;
- Loads and aeroelastics tools development engineers;

- Sub-contractors, temporarily involved in the development of the tools;
- Internal customers i.e. teams within the Loads & Aeroelastics department as users of the final product;
- External customers i.e. airline companies as potential end users.

Research and Development managers are mainly interested in the efficacy of in-service support as they look for a centralised/localised system to perform all incident analyses. Despite loads and aeroelastics tools development engineers and sub-contractors usually perform the same job, their interests are largely different. Whilst the latter's main interest is most times purely limited to satisfy the customers' needs, the former are interested in the efficiency of the system as their intent is to obtain a product that can perform faster and more accurate incident analyses. Internal and external customers are focused on the effectiveness of the system as they look for high-fidelity results. From the identification of the roles it discerns that, within the SOSM perspective, the participants hold pluralist relationships since their interests are compatible but their values and beliefs differ.

7.2.1 Soft Systems Methodology (SSM)

In the late 60's, Professor Peter Checkland¹ developed a methodology within the Systems Thinking perspective that was based on models that were not supposed to represent the real world but, by using systems rules and principles, allowed the researcher to structure his/her thinking about the real world. The development of this methodology was the result of a research programme requested by the statistician Professor Gwilym Jenkins², at the pioneering Department of Systems Engineering of Lancaster University, who strongly believed, along with Professor Checkland, in 'action research' in real world situations outside the University. The action research programme had the purpose to find a new way of tackling 'wicked problems', i.e. problems that are not well defined and involve many interested parties with different perspectives or worldviews and are never static but subject to multiple interpretations. This methodology is now used and taught worldwide as Soft Systems Methodology (SSM). Soft Systems Methodology, as argued by Checkland in his first full work on this methodology [9, 10], incorporates three main concepts

¹Peter Checkland, born 18 December 1930, in Birmingham, UK, is a British management scientist and emeritus professor of Systems at Lancaster University.

²Gwilym Meirion Jenkins (12 August 1932 - 10 July 1982) was a Welsh statistician and systems engineer, born in Gowerton, Swansea, Wales.

that led to the break from systems engineering. Firstly, using systems ideas too early in the analysis of ‘*softer*’ problems, which do not have agreed goals and objectives and lack of a obvious hierarchy of systems to be engineered, can lead to a distortion of the problem situation and the development of premature unsuitable solutions; in these cases, instead, the analysis should consist of building up the richest possible picture of the problem situation rather than trying to represent it through systems models. Second, the use of SSM is for the construction of a number of models to be compared to the real world, rather than just one as in hard methodologies. The underpinning idea of this concept is to have defined a range of systems via *root definitions* or *conceptual models*, each expressing a particular worldview, which will help improve the problem situation. Third, these models should be used as epistemological devices to discover the real world and should promote debate about necessary change. Other important aspects incorporated in SSM are the lessons learnt from the articulation and pursuit of the action research programme. As shown in Figure 7.2, the researcher first seeks for a real-world problem situation that is close to his/her research interests and then, by declaring the framework of ideas (F) and the methodology (M) that he/she wants to follow to bring improvements, enters that area of concern (A). By taking action in the situation, the researcher can reflect on the consequences of his/her F and M, which can lead to new findings relevant to F, M and A and possibly lead to new research themes. Hence, reflection could take place on the philosophical underpinnings of the methodology, and refinements could be made to supportive methods and techniques [9].

The traditional SSM approach consists of seven stages which can be grouped into two main areas (Figure 7.3): stages that deal with the *real world* and stages that address a *conceptual world*. Each of these stages is described below and applied to the analysis of the in-service support system.

1. *Identification of a problem situation*

This stage sits in the real world area and seeks to acknowledge, explore and define in some way the general area that interests the researcher. Chapter 1 incorporates this first stage as a high level vision of the problem is given along with the definition of the objectives of this work.

2. *Expressing the problem situation*

As suggested by Jackson [9], the ideal way to express the problem situation is via the use of a rich picture since the need is to provide information about the structure and process

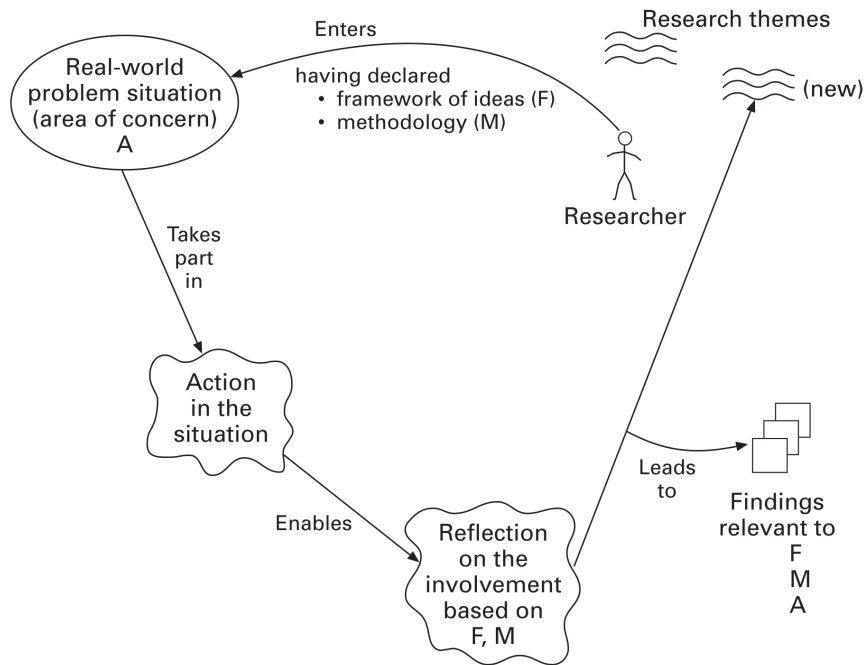


FIGURE 7.2: The cycle of action research. Source: Jackson, 2003 [9].

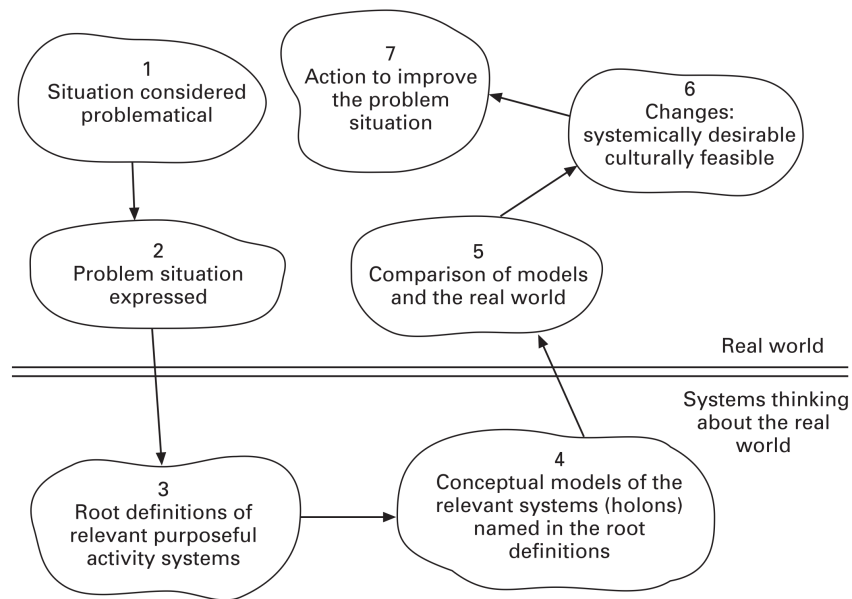


FIGURE 7.3: The learning cycle of Soft Systems Methodology (SSM). Source: Jackson, 2003 [9].

at work and the relationship between the two. Figure 1.3 in the Introduction represents the problem situation and, hence, the AS-IS process within Airbus for aircraft incidents’ analysis. Starting from the centre of the rich picture, an aircraft incident is reported to the manufacturer’s customer services department that, based on the nature of the incident, which could be flight longitudinal, vertical or lateral, ground or also resulting from other

special conditions (e.g. engine fan blade off), redirects the reported incident to the Loads & Aeroelastics team that is responsible for the characteristic incident. The redirection is channelled through the data acquisition centre that decodes and prepares the incident data into a format usable by the analysis team. The results of the analysis are then fed back to the customer services teams that, finally, provide operational recommendations and/or maintenance actions to the airlines. This process is aggravated by the fact that, due to historical reasons, each responsible team, which sits within the different sites of the organisation, will have a very limited sharing of its capabilities and know-how with the other sites' teams. This, thus, results into a very intricate and fragmented system that requires a reorganisation and redistribution of the responsibilities and capabilities.

3. *Formulating root definitions*

In order to obtain clear and concise root definitions, Checkland developed an efficient method, described and applied below, that can be easily recalled by the mnemonic word *CATWOE*, which stands as an acronym for: *Customers, Actors, Transformation, Weltanschauung (world view), Owners* and *Environment*. The customers are those who benefit from the transformation; the actors are responsible for the transformation; *Weltanschauung* is the world view that makes the transformation meaningful; the owners are those who own and control the transformation and, finally, the environmental constraints define the boundaries of the transformation process. Table 7.1 identifies each of these roles within the context of the in-service support re-development and, thus, defines the main contributors for the following root definition:

“A system to transfer existing incidents’ analysis tools to a unique interface to achieve the localisation of the incidents’ analysis process that allows a more efficient customer service and a reduction in aircraft on ground, maintenance costs and training sessions required.”

Customers	Actors	Transformation	Weltanschauung	Owners	Environmental constraints
Loads & Aeroelastics Teams	Developers	Tools’ development	Process centralisation	Product Owners	Programming languages

TABLE 7.1: *CATWOE* analysis applied to in-service support.

4. *Building of conceptual model*

By definition, a conceptual model shall illustrate all the activities that are necessary to achieve the transformation (**T**) that affects the systems involved in the root definition. Checkland suggests [10] the use of a model, pictured in Figure 7.4, that contains seven

imperatives of the same ‘scale’, which best describes this transformation process. He also argues [10] that a common feature in a conceptual model is to have a number of verbs in one subholon concerned with operations and another set of activities that are responsible for monitoring and control. Proper monitoring and control depends minimally on specifying criteria for efficiency, efficacy and effectiveness. These criteria are defined in this context as:

Efficiency - incident analysis performed faster;

Efficacy - localisation of in-service support;

Effectiveness - high-fidelity elaboration of incidents’ data.

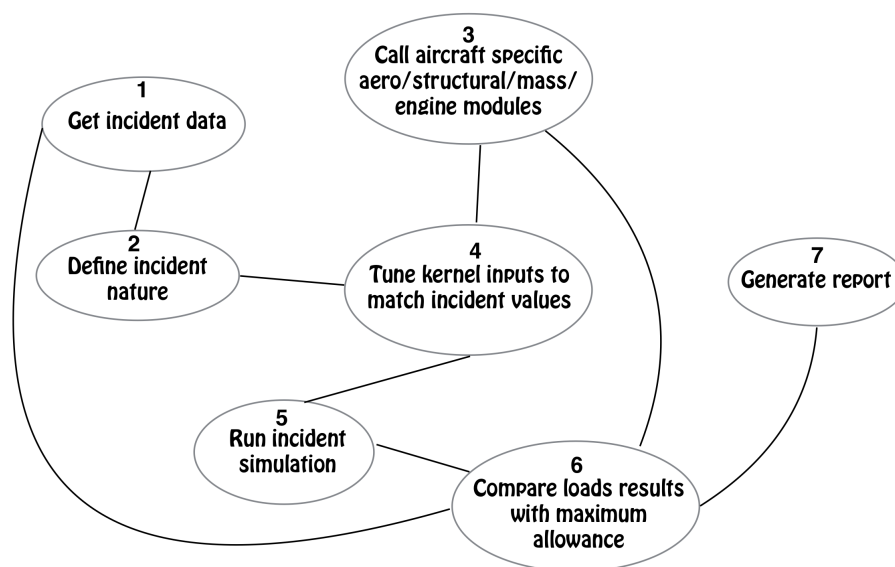


FIGURE 7.4: Conceptual model derived from root definition.

5. *Comparison of conceptual model and real world*

The rich picture and the conceptual model depicted above should be used as epistemological devices to discover the world and should promote debate about change. In order to assess these two characteristics, Checkland suggests to make a ‘comparison’ between the conceptual models and what is perceived to exist there [10]. This was done via informal discussions and semi-structured interviews (see Appendix B.1), designed to understand the main differences between the models of what might be and what seems to be the case now - here expressed in the form of a table (see Table 7.2);

6. *Defining changes that are desirable and feasible*

At this stage possible interventions are explored by re-cycling through the stages developed

AS-IS	TO BE
Incident data from the airlines is addressed to customer service located in one site but loads in-service support done in different sites	Incident data collected by the customer service team and delivered to a centralised unit
Nature of incidents is assessed prior to the analysis	Nature of incidents is assessed as part of the analysis
Different tools per each family of aircraft	One single tool that loads the required aircraft structure
Time required for incident analysis varies between one hour and several weeks, depending on the nature of the incident	Time required for incident analysis doesn't exceed the order of several hours, no matter the nature of the incident
Each tool is based on a different programming language	One single tool developed using a common programming language.

TABLE 7.2: 'What seems to be the case now' (AS-IS) with 'what might be' (TO BE)

so far; the feasibility of these changes is an important aspect of this methodology and Checkland suggests to assess it by approaching each intervention with different systems based analyses [10]. Desirable changes might be:

- Include decryption of data received from the airlines directly in the support tools;
- Release incidents' tools to the airline companies;
- Improve aerodynamic and/or structural models.

7. *Taking action to improve problem situation*

This stage fulfil the cycle and might also lead to a new one. It aims to the actualisation of the conceptual models developed so far by promoting action towards the improvement of the problem situation. In the scenario analysed here, collecting information about the structure of the existing software would improve the problem situation by providing a suitable background for the development of the single desired tool.

7.2.2 Hierarchical Process Modelling

Hierarchical Process Modelling (HPM) was developed as a modelling approach as well as a problem structuring method with the aim to support the decision-making process while assessing issues of *risk* and *uncertainty*. The idea behind the development of such a model is that

complex systems usually have a hierarchical structure i.e. a structure that can be divided into sub-components or sub-systems that individually address issues; citing Checkland, ‘*nature is hierarchically organised with emergent properties at various levels of complexity*’. In HPM all sub-systems are evaluated as processes or transformational entities that enable the simultaneous exploration of two fundamental questions: *why?* and *how?*. These two questions, in fact, can be used as keys to understand HPM, which can be figuratively seen as a *ladder of abstraction*. By going down one level in the model, the *how*’s of a process can be explored; by going up a level, the *why*’s of the process are instead explored. To contribute to the decision-making process, HPM assesses the performance of the system by measuring the likelihood of success of a process; this involves dealing with uncertainty. The way HPM assesses a process performance is by the use of the so called Italian flag i.e. a three-colour scheme that quantify uncertainty on a 0:1 scale. Green indicates that there is evidence that a process is successful; white is the total absence of evidence; red is the evidence that a process is not successful. HPM was used in this work as a fundamental guide through the development of the unique in-service incident support solution and Figure 7.5 illustrates its application with a description of the main processes below.

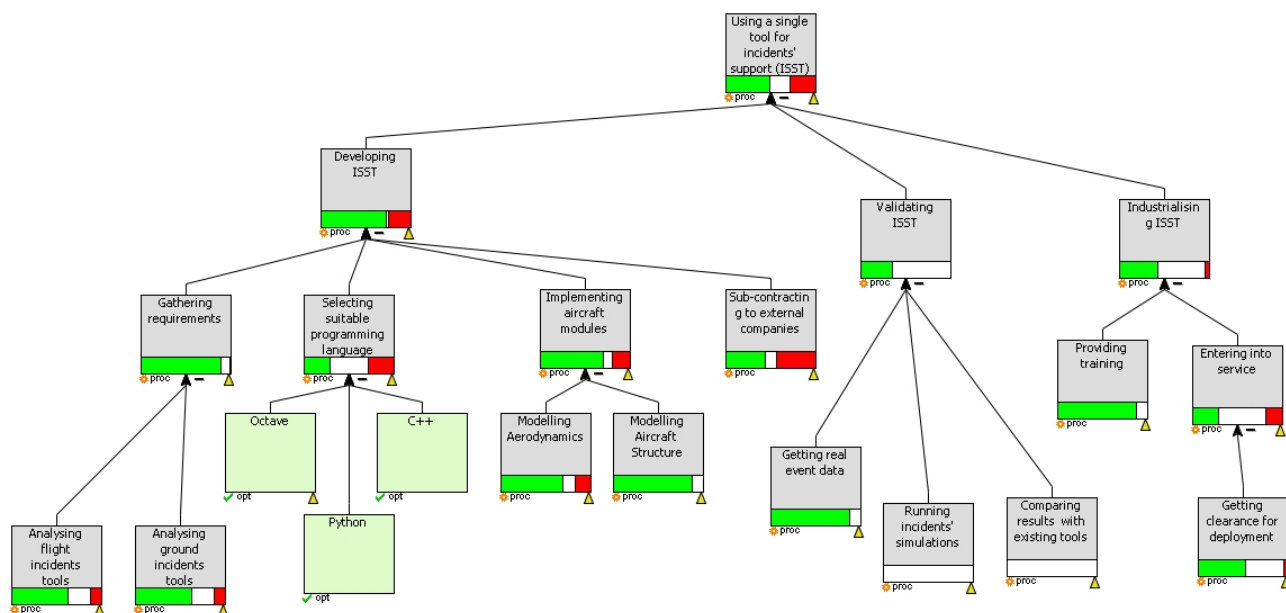


FIGURE 7.5: Problem situation expressed by HPM, where ISST is an abbreviation for in-service support tool (diagram obtained with PERIMETA).

- **Gathering requirements** This process shows a strong level of evidence of success due to its activity being based mainly on collection and analysis of existing documents. In

fact, most of the requirements were already defined during the development phases of the previous tools; however, this process presents little uncertainty due to the some times scarce traceability of these documents and/or changes/updates in the requirements specifications.

- **Selecting suitable programming language** Evidence of success is due to the applicability of almost every computer language to the development of incidents support tools. The issue here is due to the cost of licenses and tools maintenance; also a shift from one language to another is not easily accommodated within the organisation, hence the level of uncertainty of success.
- **Implementing aircraft modules** This process is divided into two sub-processes: modelling aerodynamics and modelling aircraft structure. The issue here comes from the former since the implementation of aerodynamic models needs to find a compromise between performance and quality of results.
- **Sub-contracting to external companies** This process usually shows evidence of success; however, care needs to be taken in outsourcing internal data.
- **Developing ISST** The development of the in-service support tool (ISST) is the parent process of all the others described above. It is managed by R&D and performed by the Tools Development team.
- **Getting real event data** High level of success is expected by this process since real event data is managed internally in Airbus; however, little uncertainty is due to the difficulties that might be faced when transferring huge amounts of data and by the low sampling rate available.
- **Running incidents' simulations and Comparing results with existing tools** These processes are very much dependent on the success of the development process of the tool analysed earlier.
- **Validating ISST** The validation of the new tool is a fundamental process to be completed before moving to the industrialisation process. It is the parent process for the last three processes above.
- **Industrialising ISST** It is the final process before the usage of the tool is agreed. It is sub-divided into two processes, providing training and entering into service. The latter

is dependent upon a further sub-process, getting clearance for deployment, which has to provide the GO signal for the release of the tool. Evidence of success for this latter process is mainly based on history and hence experience from the release of previous tools.

- **Using a single tool for incidents' support (ISST)** This is the main goal of the re-development.

7.2.3 Conclusions

Modelling techniques used for problem structuring are definitely powerful tools when it comes to understand and define a problem situation. In Section 7.1, in order to classify a problem situation, two dimensions were defined: complexity and divergence of values and interests; however, it is worth mentioning that it is useful to differentiate between two types of complexities: technical complexity and human complexity. In systems of high technical complexity, in fact, applying modelling techniques can sometime distort the nature of the problem by generalising low level problem situations and providing a too coarse level of details. It is indeed more appropriate the application of such methods in systems of high human or social complexity. However, the application of the Soft Systems Methodology proved satisfactory in structuring the problem situation faced by in-service support activities within Airbus due to the oncoming civil aviation market growth. SSM, in fact, together with the application of the Hierarchical Process Modelling discussed in Section 7.2.2 has made possible the development of a centralised solution named In-Service Support Tool (ISST), which will be discussed in detail in the next chapter.

Chapter 8

Solution development

Following the analysis of the system provided in Chapter 7, this chapter provides an overview from the industrial point of view of the development of the unique solution proposed in this work for the support of in-service incidents.

8.1 Introduction

To cope with an increasing and ageing aircraft fleet, a modernisation of the current loads in-service support tools and processes is proposed with this work. In Chapter 7, Soft Systems Methodology and Hierarchical Process Modelling were used to identify and formulate the problem, plan the actions required for the resolution through the use of conceptual models and interviews, and to define the structure for the development of the agreed solution. The latter consists in the migration from a situation where the on-call engineer (one of the key actors in the in-service support chain, responsible for running the analyses) has specific skills and expertise – that are function of aircraft and incident type - required to assess the incident, to a situation where any member of the on-call engineer’s team can assess any incident without the need of a specific set of requirements. Indeed, as introduced in Chapter 1, the typical path consisted in incident data being delivered from the airline operator to Airbus customer services, which is in charge of redirecting the analysis to the responsible team; the incident data, however, only reached the skilled on-call engineer after a 5-step decision process, as shown in Figure 8.1, before being delivered back to customer services and, ultimately, to the airline operator. This fragmentation was due to different tools being used per incident nature, aircraft family and type,

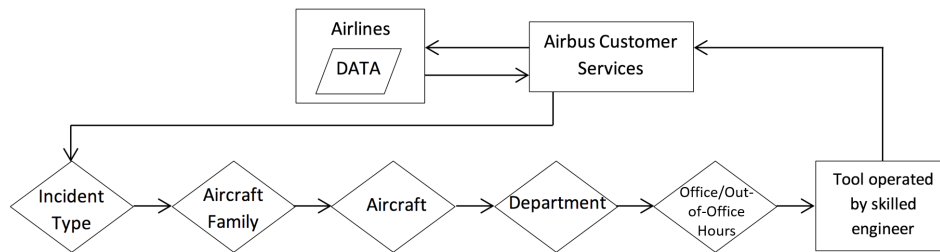


FIGURE 8.1: Pre-existing in-service support process.

site-based expertise and, finally, working and non-working hours. This implied the worksharing between teams and sites being complex, involving several toolsets, where the key disadvantages could be summarized as:

- Different tools for different aircraft and types of incidents (Sustained Engine Imbalance, ground, manoeuvre, gust and combinations);
- Overly conservative solutions;
- Manual work to setup the problems ranging from 30 minutes to 2 days;
- Heavily fragmented process, tools and know-how for covering entire fleet.

The improved process proposed is shown, instead, in Figure 8.2. This ultimately consists in a single actor external to the customer service team, i.e. the on-call team member, who has access to a unique software solution that does not require specific skills and that has the capability to handle the analysis of virtually all incidents. This solution shall improve the in-service

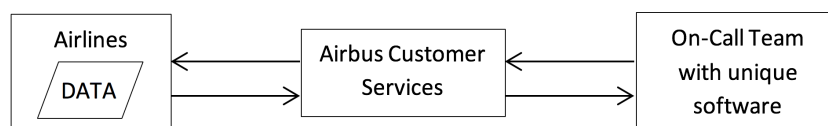


FIGURE 8.2: Proposed solution for in-service support process.

support process by reducing the lead time for aircraft back-to-service status, thus saving revenues at airlines level as well as improving Airbus reputation. It is a solution that relies on the development of new and harmonised software that:

- Is unique for all types of aircraft and all types of loads incidents, thus enabling flexible and robust on-call service;
- Has greater accuracy from use of production tools implemented for in-service support;

- Delivers faster resolution from increased automation and vastly improved graphical user interface.

The development of this solution, that for the remaining of this report will be referred to as ISST, which is the acronym for loads in-service support tool, follows a “time-box iterative”¹ process (Figure 8.3) that consists in multiple releases of the software being delivered in relatively short periods of time (circa 8-10 months). Each individual release is fully functional and adds new capabilities with respect to the previous release.

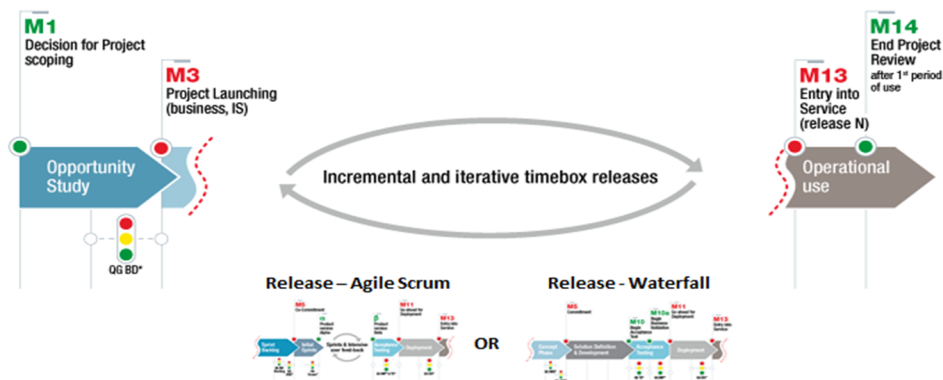


FIGURE 8.3: Time-Box Iterative model. Source: Airbus’ internal document.

The following sections provide a description of the improvements implemented from the operational and functional point of view and the development steps followed to achieve them. Furthermore, this chapter shall also provide enough information to guide the future and eventual implementation of the gust reconstruction method discussed in Chapter 5 into the ISST framework.

8.2 System in context

The purpose here is to show the new system in a global context. The first objective is to identify the functional boundaries of the system in this environment and how the system interacts with external actors (applications or users). The second objective is to identify how this environment

¹Time-Box Iterative is a project life cycle that proposes the Entry Into Service (EIS) of the IS solution in smaller batches that deliver an added value to the business through several iterations that are shorter in time and closer to each other but also cross-disciplinary. Time-Box Iterative is similar to the Agile-Scrum in terms of duration as it targets 6 months per release. It is chosen to maximise the return on value but it can also incur into extra costs due to the planned multiple entry into service milestones.

is impacted by the introduction of the new system and, particularly, how this environment is globally simplified i.e. with less applications and fewer interfaces.

Figures 8.4 and 8.5 show, respectively, the contextual view of the new system and the principal actors. Concerning the actors, only one single user profile has been identified to interact directly with the unique ISST interface: this is the user “On-call Team member”. This user receives the inputs from the Customer Services team and performs the computations with ISST. S/he is part of the Loads & Aeroelastics in-service support team. The other key actor is part of the in-service incident support process, but does not directly interact with the tool: this is the actor “Customer Services”. It provides the initial inputs (loads event, flight data recordings) to the actor “On-call Team member” and is the addressee of the output results. But, s/he does not perform any computation with ISST. The loads in-service support tool is broken down into two branches, one for ground incidents and one for flight incidents. Within each branch, based on the type of aircraft², a dedicated solver is identified and incorporated into ISST.

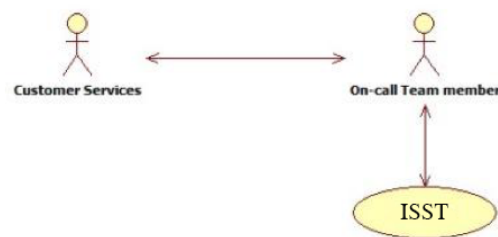


FIGURE 8.4: Actors.

Figure 8.6 represents this new system in the global context. As schematized, it may be understood as a black box providing connection to external tools in order to handle, analyze and plot recorded flight data. The figures shaded in blue require user interaction, whilst those shaded in green are performed within the loads in-service support tool, where the processing is hidden to the user, who is only accountable for the setup of the analysis and the delivery of the results to the Customer Services team.

8.3 Functional top-level decomposition

The functional view is a system-level architecture that partitions the system functionalities and data into logical components (functional packages) and their relationships (control flow or data

²The aircraft families indicated in Figure 8.5 are for demonstration purposes only and do not represent by any means the Airbus aircraft fleet.

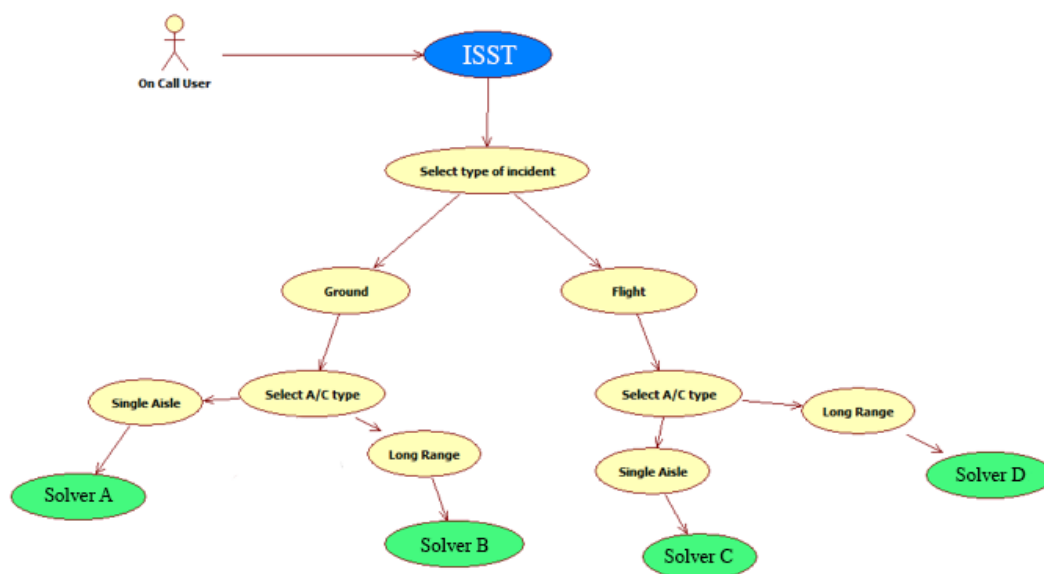


FIGURE 8.5: To-be contextual view.

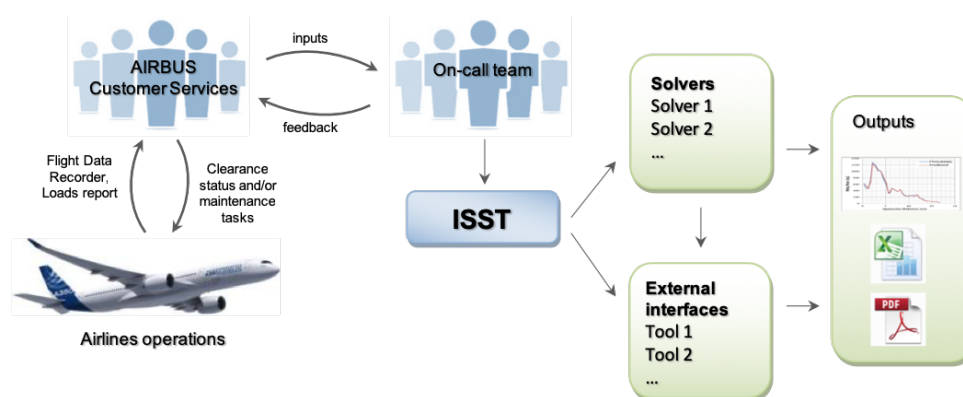


FIGURE 8.6: Position in the system.

flow). Figure 8.7 shows that ISST is decomposed into three functional layers. The first layer includes the tools/applications that are, or will be (future releases), incorporated into the main application of ISST; these applications are usually developed from scratch, however, if existing, they will be completely redesigned to fit within the scope of ISST. The second layer includes the kernels on which each application is based; these are usually wrapped into ISST without being redeveloped or with minor changes implemented to avoid compatibility issues. The third layer, instead, consists of all third-party applications that are not directly shipped with ISST and are instead available as standalone solutions. An example of third layer applications are post-processing tools used for plotting and/or further analysis of the results. Figure 8.8 shows a functional application mapping, where a sequence diagram illustrates how processes operate with one another and what is their order.

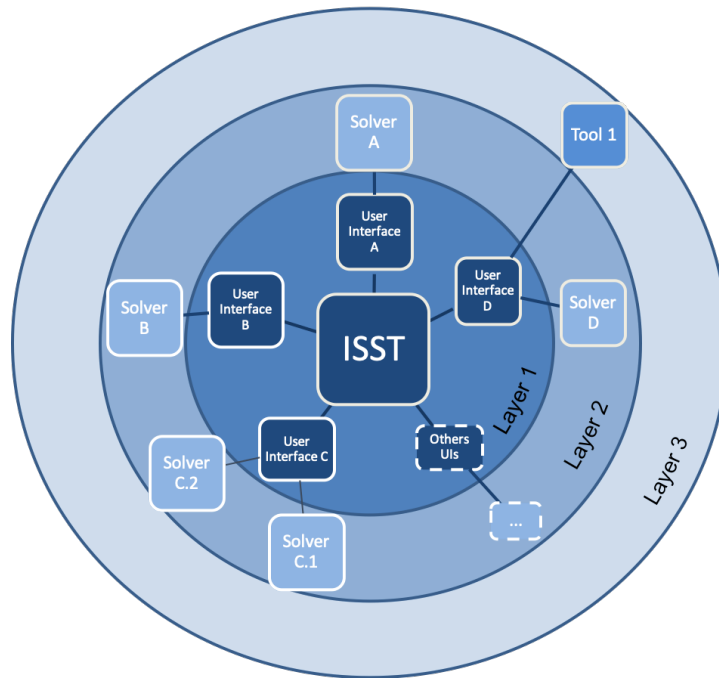


FIGURE 8.7: Functional top-level decomposition.

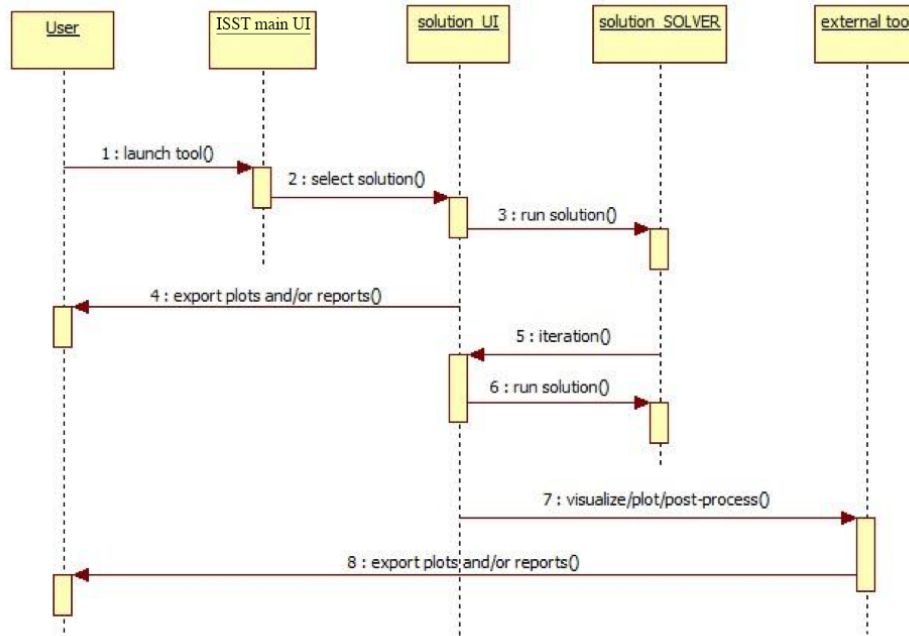


FIGURE 8.8: ISST sequence diagram.

8.4 System architecture

The main architecture drivers that constrain or otherwise influence the architecture design are briefly presented here, along with the global design rationale that describes and justifies the main decisions of the design. Hence, ISST architecture drivers are based on:

- *Reuse of existing modules developed for previous in-service support tools*

A proof of concept phase was done to assess the ability of the existing framework to be easily interfaced to other in-service analysis tools.

- *Modular and flexible functional environment*

- *License-free tool*

For many reasons, e.g. business standard technologies, reuse of existing business libraries and performances, the chosen technologies for the implementation of ISST are Python and PyQt³. However, parts of the existing solutions are coded in Fortran language and are kept unchanged (no updates or modification to be done within this area for this project).

- *User friendly GUI*

The tool is launched through the main graphical interface, inherited from an existing solution and re-formatted accordingly. Then, the user is connected to several computation kernels, depending on the aircraft and event type. To achieve this, the original tool has been reshaped and split in two components, the user interface part and the solver part, i.e. the kernel including computation functionality.

- *Interface to other in-service support tools*

On a long term view, the plan is for ISST to interface with all the in-service solutions available and that, today, are still external to ISST.

8.5 System deployment

The ISST framework will be deployed on local laptops/desktops on any of the company sites that host the in-service support on-call teams. Users will have to download the installers from a shared folder located on the department restricted area, and then perform the installation locally - all the dependencies to the solvers will be handled by the installer. Hence, the tool will run as a standalone software with no network usage required. Figure 8.9 gives an overall summary of the deployment architecture. As shown in the figure, access to the shared folder is provided via the WiKi portal described in Section 8.6. The latter is also discussed as an alternative solution for the management of the software documentation.

³PyQt is a GUI toolkit that includes abstractions of network sockets, threads, Unicode, regular expressions, SQL databases, SVG, OpenGL, XML, a fully functional web browser, a help system, a multimedia framework, as well as a rich collection of GUI widgets.

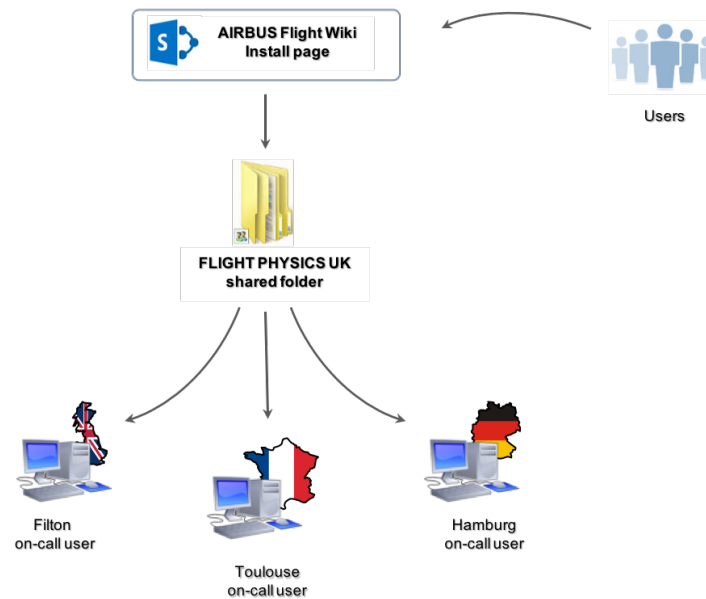


FIGURE 8.9: Overview of ISST deployment process.

8.6 Documentation management

“Any engineering artefact goes through a “lifelike” analogy to a living thing, *i.e.* conception, birth, growth, adulthood and death”⁴. Defining the lifecycle of a program/project is one of the fundamental concepts used within an organisation for the management of major systems [88]. In the definition of the lifecycle, the entire process is structured into several phases, which provide a more manageable overview of the program/project to the managers. Key Decision Points (KDPs), also known as milestones, are the events that mark significant projects checkpoints and are an indication of whether the program/project is ready to progress to the next phase of the lifecycle. Passing these checkpoints is mainly based upon the initialisation, validation and/or delivery of a number of documents specific to each milestone. A Project Manager (PM) usually creates up to 50 different types of documents for planning, tracking and reporting; these include feasibility studies, resource spreadsheets, financial and project plans, suppliers contracts, post-implementation reviews, change request forms and project status reports that are all believed to be essential ingredients in successful project management [89]. Most PMs would in fact agree that project documents play a fundamental role in the strategic development of the best possible plan by also aiming to facilitate the communication of progress and status to all stakeholders. However, documents are so complex that they need management themselves, which, if not properly done, could induce documents to take over the process that they are supposed to

⁴Citation taken from Systems Engineering Lecture Notes by T. Tryfonas, Lecturer of the “Fundamentals of Systems Engineering” module at the University of Bristol, 2013.

facilitate. Properly managing documents means going through their own lifecycle; this includes preparation (analysis and design), initial creation, feedback and modification, approval and subsequent updates and modification. Preparation and initial creation require the greatest effort in the whole process as the structure need be created and the content outlined; although the subsequent steps do not require as much effort, they usually take a very long time (order of weeks) to be completed since more people are involved for feedback, modifications and approval. Overall, a huge effort is required in terms of time and commitment to fully deliver the project documentation and, as a result, the risk of falling in poor documents' management is really high. Poorly managed documents can hide the real status of a project thus creating confusion and frustrating those who want answers and those who need to deliver [89]; they also put the success of the project itself at risk. As a matter of fact, people do think that documentation is a key aspect behind the success of the development of a new software, but why is that? According to Scott Ambler in [90], this is due to the shift of many organisations' IT departments from a "code and fix" hacking mentality to a documentation-heavy serial waterfall process that happened during the 70s and 80s - a shift that actually improved the overall projects' success rate and thus "taught" people that documentation improves the software development effort; however, this does not imply that there weren't and there aren't better ways to work.

In order to promote an agile-like development and to be in line with the time-box iterative process adopted for this development, an alternative solution for the management of the software documentation was also attempted in this work. The effort, here, was to merge the innovations brought into documentation management by both the agile and the rapid application development (RAD) methodologies to the classical approach adopted in Airbus⁵. This communion could be achieved by introducing a fully interactive web-based platform - a WiKi portal - that allows the stakeholders and the development team to access and update the required information at any time and in a seamlessly manner, thus promoting the early cooperation between the two parties. In this way, information can be provided in the form of a "blog" that can support different formats such as texts, images, videos, etc, depending on the complexity of the content that needs to be communicated. The focus was on avoiding the tedious task of creating and downloading a countless amounts of documents and having to scroll through thousands and thousands of words to finally get the information desired. FAQs, feedbacks and general comments recorded on the blog by all the actors of the project then become fundamental to keep track of the work progress, to make decisions to *go* or *not go* ahead in the project lifecycle

⁵More details on documents management in Airbus are also given in Appendix D.

and to facilitate communications. More in detail, the WiKi portal consists of a main page that gives general and specific information about the project and provides access via hyperlinks to areas whose contents would instead be written in static documents, e.g. the ARD or BRD. An “Installation” area is also included that provides step by step procedures for software installation and video tutorials that guide the user through the actions that s/he is required to do, thus substituting the users’ manuals and installation dossiers (figures 8.10 and 8.11).

ISST Installation

Download **ISST** from:

[Latest version](#)

Then follow the steps below or [watch the tutorial](#):

1. Unzip **ISST** to a local folder
2. Launch **ISST**
 - Make sure that you have [Python for Windows \(2.7.3.0\)](#) installed on your computer
3. Hit [Next >](#) and click on the folder icon to select/create the **ISST** Installation Folder
 - **Make sure to install ISST under the current user folder**
4. Hit the [Install](#) button and wait until the installation is complete
5. **ISST** has now been installed on your computer
6. Hit [Next >](#) to check location and Settings Information.
7. You can now close the Installation Wizard by hitting [Finish](#).

[How to install solvers in ISST](#)

[Back](#)

FIGURE 8.10: WiKi portal: Installation.

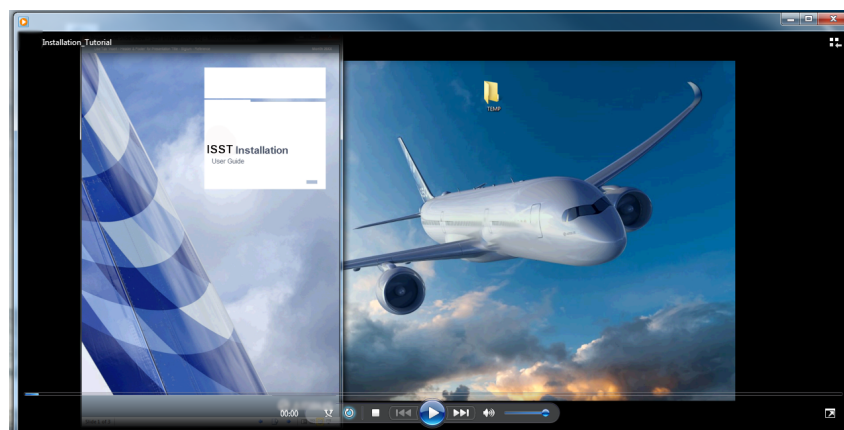


FIGURE 8.11: WiKi portal: Video Tutorial (screenshot).

By following this approach, the stakeholders have full control on the progresses of the project and they can check, at any time, how their investment is being spent. They also have the possibility to contribute to the development of the solution by providing feedback and interacting lively with the development team, thus facilitating communications.

8.7 Conclusions

This chapter has given an overview of the process behind the implementation of the solution proposed as part of this work for the harmonisation of in-service support activities within Airbus. More in general, it has provided a summary of the steps required for the industrialisation of a new asset, including the production and management of the project documentation.

The timeline of Figure 8.12 places the development of the ISST solution within the time frame of this Engineering Doctorate and highlights the major milestones that the EngD program helped to achieve during the 4-year period. The milestones shown on the timeline are:

- M1 for *project scoping*, which marks the beginning of the opportunity study phase;
- M3 for *project launching*, which marks the beginning of the concept phase;
- M5 for *commitment*, which marks the beginning of the solution definition phase;
- M13 for *entry into service*, which marks the end of the development phase and the start of the operational use.

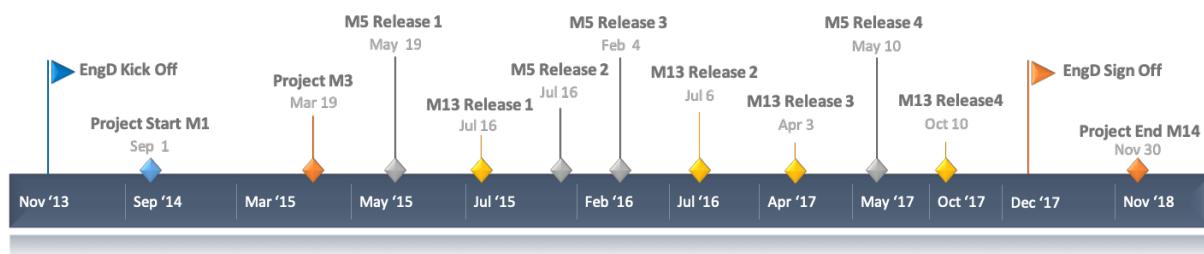


FIGURE 8.12: Timeline of ISST development during the EngD project.

Today, the development of this new solution is in phase five with the first four phases being already operational and used by the on-call engineers' team at Airbus. The benefits that this solution has brought to the business can be summarised as:

- reduction in the lead time for aircraft back-to-service status, saving revenues at airlines level as well as improving Airbus reputation;
- uniqueness of the solution for all types of aircraft and all types of loads incidents, thus enabling flexible and robust on-call service;
- improved accuracy from use of production tools implemented for in-service support;

- faster resolution from increased automation and vastly improved graphical user interface.

Chapter 9

Conclusions

This thesis is designed around two major objectives: the development of a method for the evaluation of gusts and turbulence properties and the centralisation of the incident support process in the industrial context. This chapter provides a summary of the key findings and achievements and gives suggestions on the applicability of the methods developed to other areas of engineering.

9.1 Outcome

9.1.1 The development of a framework for gust reconstruction

It has been shown in the first two chapters of this thesis that understanding the properties of any turbulence encountered by in-service transport aircraft under various operating conditions is of increasing interest to aircraft manufacturers, which are putting effort into this process in order to collect valuable information for future aircraft development. The inverse method is the state of the art in industry for this type of applications, where the reconstruction of turbulence properties is obtained by observing the system response measurements and evaluating the inverse of the forward system by means of Fourier transforms. Some limitations, however, arise when adding nonlinearities into the equations of motion and time domain methods offer improvements over frequency domain methods. Thus, the design of two time domain methods, both based on numerical optimisation algorithms, is discussed and presented in the form of gust reconstruction frameworks that may be deployed to analyse systematically any generic, nonlinear aeroelastic

model, requiring little or no knowledge of the underlying equations except from the simple input/output manipulation of the main routine. The two methods differ on the setup of the optimisation routine: the first is designed around gradient-free methods and is suitable for models that are fast in the computation but provide results with low accuracy; the second is designed around gradient-based algorithms that can be employed with models that have a slower turnaround but provide greater accuracy in the results. In order to address the validity of the two methods, the gust reconstruction frameworks have been, first, validated on the linear version of a typical wing section with the use of the well-known ‘ $1 - \cos$ ’ gust and, later, applied to the respective nonlinear version for the same gust. The validation proved satisfactory as the results showed that a perfect match was achieved between the reference and the reconstructed responses, and that the reconstructed gust well matched its reference counterpart. Finally, the reason for considering models of increasing complexity is to demonstrate that this reconstruction may be applicable at any stage of the design process and, also, for in-service support.

9.1.2 The development of an improved in-service support solution

Over the last four years, the organisational problem of providing in-service support for an ever increasing number of aircraft in operation has been approached with formal methods in systems engineering that “*added value to the final deliverable as well as improved the efficiency in managing the various organisational interfaces*” – A. Rampurawala, Airbus Operations Ltd.

The solution proposed consists in the centralisation of the in-service support process with the introduction of a new software that represents a unique entry point for the analysis of any reported incident. It is a solution that allows a new harmonised process that will deliver faster incident resolution with results of even greater accuracy obtained from the use of aircraft production tools. This solution follows an iterative development cycle where new functions are added and released at the completion of each iteration, as suggested by Figure 9.1. Today, the development is in phase five with the first four phases being already operational and used by the on-call team at Airbus.

Last but not least, positive feedback has also been collected, during several review meetings, on the alternative approach proposed for the documentation management. Members of the steering committee (the people in charge of making decisions for the progress of the project) have agreed that such an approach would indeed bring benefits; for example, the use of video tutorials would help saving a significant amount of money usually allocated for training purposes. It has

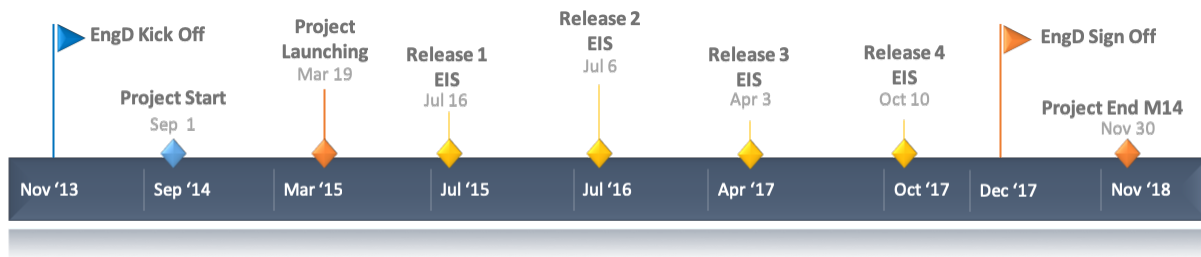


FIGURE 9.1: Entry into service (EIS) of each iteration (release) during the EngD project.

also been recognised that this approach would be ideal for the particular case of building up documentation when a prototype of the final software already exists, perhaps as an evolution of a previous version.

9.2 Key findings

- The two main branches of optimisation, i.e. gradient-free and gradient-based, had to be considered in this study in order to provide ad-hoc solutions that conform to the type and characteristics of the aerofoil (or aircraft) model employed and that provide a good level of compromise between accuracy and cost, which is imposed by the design stage in progress.
- The two optimisation methods have been applied using the aerofoil represented by the 2DoF nonlinear model and the model that is the most representative of the industrial one (i.e. the 3D beam stick model), on case studies that considered the reconstruction of random and more realistic gust profiles, obtained analytically over time approximations of the von Kármán power spectral density (PSD). The realistic gust profiles were obtained analytically in place of real events to have a mean of comparison and to evaluate the accuracy of the reconstruction; however, time histories obtained from real events are equally valid and are to be considered as the only input data required for these applications.
- The solid results obtained proved that optimisation may represent an invaluable solution for reconstructing the loads exerted on an aircraft that encounters atmospheric gusts. Indeed, the satisfactory application of the method to models that included structural nonlinearities provides a potential solution to the limitation specific to inverse methods in the frequency domain.

- When coupled with rapid methods, such as reduced order models, the optimisation method could allow in the future running the gust reconstruction framework with real-time measurements and provide an instantaneous monitoring of the airframe in flight. In fact, as the computing cost of high fidelity aeroelastic model grows rapidly, this work could represent the first steps in the exploitation of already proved efficient reduced order model techniques.
- Given its strong computational performance, the gradient based optimisation framework constructed with CFD reduced order models represents the most suitable solution to be used as a solver for real time applications of the in-service support tool developed and, thus, reconstruct gust profiles lively from in-service data.
- The successful reconstruction of gusts and turbulence profiles may open the way to a new method for the definition of the worst case gusts, which is a condition needed by aircraft designers to certify that aircraft are designed to resist the highest dynamic loads possible. Indeed, it has been shown that the reconstruction framework can be used to build a large database of real gust profiles that, following the application of singular value decomposition techniques, surrogate modelling and optimisation, could be used to define the worst gust conditions that an aircraft may experience during its operation.

9.3 Outlook

If on one hand, the gust reconstruction framework proposed in the first part of this thesis proved to be a successful method for the reconstruction of the loads acting on the aircraft structure, on the other hand, the in-service support solution presented in the second part represents the required framework needed for the industrialisation of this method. Thus, given the iterative characteristic chosen for the development of ISST, the next major release of the software may include the gust reconstruction framework in the form of a reliable tool for the analysis of flight incidents.

Also, as mentioned earlier, if coupled with rapid analysis methods such as reduced order models, the gust reconstruction framework could be used to provide an instantaneous monitoring of the airframe in flight, taking the advantage of real-time measurements. With this implementation, the tool could be deployed as part of the aircraft instrumentation and be sold as a service to the aircraft operators.

Furthermore, the reconstruction method proposed here may be applicable to areas that are not strictly related to aircraft operations. These areas include, for example, wind turbines and weather forecast. Concerning the former, studies exist in the literature on the reconstruction of the wind inflow characteristics past wind turbines [91, 92]. As per gusts and turbulence events over aircraft wings, the wind blowing over a wind turbine rotor leaves its own trace on the response; if this (e.g. the loads) can be measured accurately enough, then the rotor could act as a wind sensor used to derive the properties of the wind flow. Understanding the wind inflow is useful for a number of applications such as wind turbine control, energy harvesting, analysis of the atmosphere and, more generally, to build a more complete picture of the wind flow within a power plant. In fact, this could help in the control of wake redirection strategies, which aim at reducing the interaction with downstream machines by deliberately deflecting the wake. On the latter, instead, the reconstruction method, combined with the approach presented in Chapter 6, may be used as a mean to analyse the structure of the atmosphere and provide accurate information that could aid weather forecaster in modelling the atmosphere. In fact, given the impressive amount of aircraft that constantly fly our skies (Figure 9.2), aircraft could be considered as flying sensors that record or provide on-demand live atmospheric parameters such as pressure, temperature, humidity and wind speed to ground stations around the globe. These observations could be used as inputs for computer-based weather prediction models and help, for example, with the prediction of local severe storm, climate change research and air pollution models.

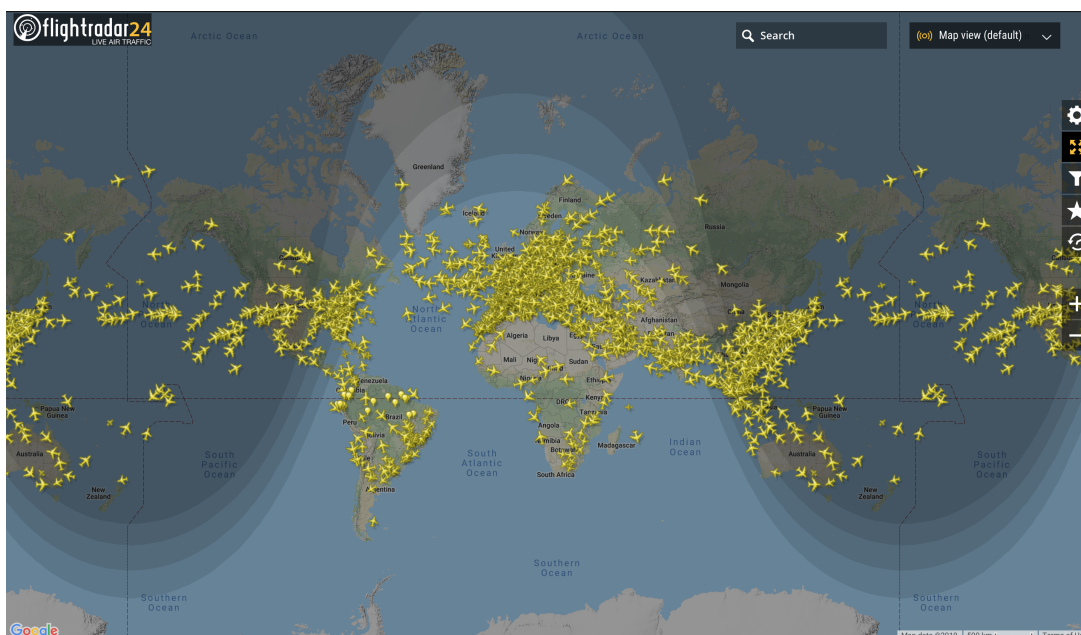


FIGURE 9.2: Live global air traffic in a normal day of operation during January, 2019. Source: <http://www.flightradar24.com>.

Appendix A

Aerodynamics Derivatives

A.1 Longitudinal Aerodynamics Derivatives for Rigid Aeroplane

[1]

$$\begin{aligned}Z_{\alpha} &= -\frac{1}{2}\rho V^2[S_W a_W + S_T a_T(1 - k_{\epsilon})] \\Z_{\dot{z}} &= -\frac{1}{2}\rho V[S_W a_W - S_T a_T(1 - k_{\epsilon})] \\Z_q &= -\frac{1}{2}\rho V S_T a_T l_T \\M_{\alpha} &= \frac{1}{2}\rho V^2[S_W a_W l_W - S_T a_T(1 - k_{\epsilon})l_T] \\M_{\dot{z}} &= \frac{1}{2}\rho V[S_W a_W l_W - S_T a_T(1 - k_{\epsilon})l_T] \\M_q &= -\frac{1}{2}\rho V S_T a_T l_T^2\end{aligned}$$

A.2 Gust-Related Derivatives [1]

$$\begin{aligned}Z_{gW} &= -\frac{1}{2}\rho V S_W a_W & \text{and} & & Z_{gT} &= -\frac{1}{2}\rho V S_T a_T(1 - k_{\epsilon}) \\M_{gW} &= \frac{1}{2}\rho V S_W a_W l_W & \text{and} & & M_{gT} &= -\frac{1}{2}\rho V S_T a_T l_T(1 - k_{\epsilon}).\end{aligned}$$

A.3 Longitudinal Aerodynamics Derivatives for Flexible Aero- plane [1]

$$\begin{aligned}
 Z_e &= \frac{1}{2}\rho V^2[-S_{WaW}J_1 - S_{TaT}\gamma_{eT}] \\
 Z_{\dot{e}} &= -\frac{1}{2}\rho V S_{TaT}\kappa_{eT} \\
 M_e &= \frac{1}{2}\rho V^2[S_{WaW}l_W J_1 - S_{TaT}l_T\gamma_{eT}] \\
 M_{\dot{e}} &= -\frac{1}{2}\rho V S_{TaT}l_T\kappa_{eT} \\
 Q_\alpha &= \frac{1}{2}\rho V^2[-S_{WaW}J_2 - S_{TaT}(1 - k_\epsilon)\kappa_{eT}] \\
 Q_{\dot{z}} &= \frac{1}{2}\rho V[-S_{WaW}J_2 - S_{TaT}(1 - k_\epsilon\kappa_{eT})] \\
 Q_q &= -\frac{1}{2}\rho V S_{TaT}l_T\kappa_{eT} \\
 Q_e &= \frac{1}{2}\rho V^2[-S_{WaW}J_3 - S_{TaT}\gamma_{eT}\kappa_{eT}] \\
 Q_{\dot{e}} &= -\frac{1}{2}\rho V S_{TaT}\kappa_{eT}^2
 \end{aligned}$$

Appendix B

Systems Engineering tools

B.1 Semi-structured interviews

The semi-structured interviews used to understand the in-service support system and to extract a conceptual model were structured in three parts:

- Collection of information on the process;
- Collection of information on the existing tools;
- Suggestions for improvements.

The list of questions asked within the Loads & Aeroelastics department follows.

- What is the process that you follow for analysing an incident?
 - How do you receive the data files?
 - How do you open the files?
 - What do you do next?
 - What do you use to plot the data?
 - How do you analyse and/or deliver results?
- What tool/tools are you currently using for incident support?
- Do you know who is currently in charge of the tool and its maintenance?

- How did you install the tool/tools?
- What were the constraints during the installation?
 - Working environment i.e. Windows, Unix, Matlab?
 - Location paths?
 - Others?
- What incident analyses are you able to perform?
 - Aircraft family?
 - Incident type?
- Do you think there is anything that could be improved in the tool and/or process?
- Any further comments and/or suggestions?

A database collecting the answers to these question is available upon request.

Appendix C

Python scripts

C.1 Multiprocessing for finite difference

```
1  '''
2  Created on 24 Jul 2017
3
4  @author: ng35694
5  @contact: simone.simeone@bristol.ac.uk
6  '''
7
8  #####
9  #####
10 '''
11     ##
12     ##
13     ##
14     ##
15     ##
16     ##
17     ##
18     ##
19     ##
20     ##
21     ##
22     ##
23     ##
24     ##
25     ##
26     ##
27     ##
28     ##
29     ##
30     ##
31     ##
32     ##
33     ##
34     ##
35     ##
36     ##
37     ##
38     ##
39     ##
40     ##
41     ##
42     ##
43     ##
44     ##
45     ##
46     ##
47     ##
48     ##
49     ##
50     ##
51     ##
52     ##
53     ##
54     ##
55     ##
56     ##
57     ##
58     ##
59     ##
60     ##
61     ##
62     ##
63     ##
64     ##
65     ##
66     ##
67     ##
68     ##
69     ##
70     ##
71     ##
72     ##
73     ##
74     ##
75     ##
76     ##
77     ##
78     ##
79     ##
80     ##
81     ##
82     ##
83     ##
84     ##
85     ##
86     ##
87     ##
88     ##
89     ##
90     ##
91     ##
92     ##
93     ##
94     ##
95     ##
96     ##
97     ##
98     ##
99     ##
100    ##
101    ##
102    ##
103    ##
104    ##
105    ##
106    ##
107    ##
108    ##
109    ##
110    ##
111    ##
112    ##
113    ##
114    ##
115    ##
116    ##
117    ##
118    ##
119    ##
120    ##
121    ##
122    ##
123    ##
124    ##
125    ##
126    ##
127    ##
128    ##
129    ##
130    ##
131    ##
132    ##
133    ##
134    ##
135    ##
136    ##
137    ##
138    ##
139    ##
140    ##
141    ##
142    ##
143    ##
144    ##
145    ##
146    ##
147    ##
148    ##
149    ##
150    ##
151    ##
152    ##
153    ##
154    ##
155    ##
156    ##
157    ##
158    ##
159    ##
160    ##
161    ##
162    ##
163    ##
164    ##
165    ##
166    ##
167    ##
168    ##
169    ##
170    ##
171    ##
172    ##
173    ##
174    ##
175    ##
176    ##
177    ##
178    ##
179    ##
180    ##
181    ##
182    ##
183    ##
184    ##
185    ##
186    ##
187    ##
188    ##
189    ##
190    ##
191    ##
192    ##
193    ##
194    ##
195    ##
196    ##
197    ##
198    ##
199    ##
200    ##
201    ##
202    ##
203    ##
204    ##
205    ##
206    ##
207    ##
208    ##
209    ##
210    ##
211    ##
212    ##
213    ##
214    ##
215    ##
216    ##
217    ##
218    ##
219    ##
220    ##
221    ##
222    ##
223    ##
224    ##
225    ##
226    ##
227    ##
228    ##
229    ##
230    ##
231    ##
232    ##
233    ##
234    ##
235    ##
236    ##
237    ##
238    ##
239    ##
240    ##
241    ##
242    ##
243    ##
244    ##
245    ##
246    ##
247    ##
248    ##
249    ##
250    ##
251    ##
252    ##
253    ##
254    ##
255    ##
256    ##
257    ##
258    ##
259    ##
260    ##
261    ##
262    ##
263    ##
264    ##
265    ##
266    ##
267    ##
268    ##
269    ##
270    ##
271    ##
272    ##
273    ##
274    ##
275    ##
276    ##
277    ##
278    ##
279    ##
280    ##
281    ##
282    ##
283    ##
284    ##
285    ##
286    ##
287    ##
288    ##
289    ##
290    ##
291    ##
292    ##
293    ##
294    ##
295    ##
296    ##
297    ##
298    ##
299    ##
300    ##
301    ##
302    ##
303    ##
304    ##
305    ##
306    ##
307    ##
308    ##
309    ##
310    ##
311    ##
312    ##
313    ##
314    ##
315    ##
316    ##
317    ##
318    ##
319    ##
320    ##
321    ##
322    ##
323    ##
324    ##
325    ##
326    ##
327    ##
328    ##
329    ##
330    ##
331    ##
332    ##
333    ##
334    ##
335    ##
336    ##
337    ##
338    ##
339    ##
340    ##
341    ##
342    ##
343    ##
344    ##
345    ##
346    ##
347    ##
348    ##
349    ##
350    ##
351    ##
352    ##
353    ##
354    ##
355    ##
356    ##
357    ##
358    ##
359    ##
360    ##
361    ##
362    ##
363    ##
364    ##
365    ##
366    ##
367    ##
368    ##
369    ##
370    ##
371    ##
372    ##
373    ##
374    ##
375    ##
376    ##
377    ##
378    ##
379    ##
380    ##
381    ##
382    ##
383    ##
384    ##
385    ##
386    ##
387    ##
388    ##
389    ##
390    ##
391    ##
392    ##
393    ##
394    ##
395    ##
396    ##
397    ##
398    ##
399    ##
400    ##
401    ##
402    ##
403    ##
404    ##
405    ##
406    ##
407    ##
408    ##
409    ##
410    ##
411    ##
412    ##
413    ##
414    ##
415    ##
416    ##
417    ##
418    ##
419    ##
420    ##
421    ##
422    ##
423    ##
424    ##
425    ##
426    ##
427    ##
428    ##
429    ##
430    ##
431    ##
432    ##
433    ##
434    ##
435    ##
436    ##
437    ##
438    ##
439    ##
440    ##
441    ##
442    ##
443    ##
444    ##
445    ##
446    ##
447    ##
448    ##
449    ##
450    ##
451    ##
452    ##
453    ##
454    ##
455    ##
456    ##
457    ##
458    ##
459    ##
460    ##
461    ##
462    ##
463    ##
464    ##
465    ##
466    ##
467    ##
468    ##
469    ##
470    ##
471    ##
472    ##
473    ##
474    ##
475    ##
476    ##
477    ##
478    ##
479    ##
480    ##
481    ##
482    ##
483    ##
484    ##
485    ##
486    ##
487    ##
488    ##
489    ##
490    ##
491    ##
492    ##
493    ##
494    ##
495    ##
496    ##
497    ##
498    ##
499    ##
500    ##
501    ##
502    ##
503    ##
504    ##
505    ##
506    ##
507    ##
508    ##
509    ##
510    ##
511    ##
512    ##
513    ##
514    ##
515    ##
516    ##
517    ##
518    ##
519    ##
520    ##
521    ##
522    ##
523    ##
524    ##
525    ##
526    ##
527    ##
528    ##
529    ##
530    ##
531    ##
532    ##
533    ##
534    ##
535    ##
536    ##
537    ##
538    ##
539    ##
540    ##
541    ##
542    ##
543    ##
544    ##
545    ##
546    ##
547    ##
548    ##
549    ##
550    ##
551    ##
552    ##
553    ##
554    ##
555    ##
556    ##
557    ##
558    ##
559    ##
560    ##
561    ##
562    ##
563    ##
564    ##
565    ##
566    ##
567    ##
568    ##
569    ##
570    ##
571    ##
572    ##
573    ##
574    ##
575    ##
576    ##
577    ##
578    ##
579    ##
580    ##
581    ##
582    ##
583    ##
584    ##
585    ##
586    ##
587    ##
588    ##
589    ##
590    ##
591    ##
592    ##
593    ##
594    ##
595    ##
596    ##
597    ##
598    ##
599    ##
600    ##
601    ##
602    ##
603    ##
604    ##
605    ##
606    ##
607    ##
608    ##
609    ##
610    ##
611    ##
612    ##
613    ##
614    ##
615    ##
616    ##
617    ##
618    ##
619    ##
620    ##
621    ##
622    ##
623    ##
624    ##
625    ##
626    ##
627    ##
628    ##
629    ##
630    ##
631    ##
632    ##
633    ##
634    ##
635    ##
636    ##
637    ##
638    ##
639    ##
640    ##
641    ##
642    ##
643    ##
644    ##
645    ##
646    ##
647    ##
648    ##
649    ##
650    ##
651    ##
652    ##
653    ##
654    ##
655    ##
656    ##
657    ##
658    ##
659    ##
660    ##
661    ##
662    ##
663    ##
664    ##
665    ##
666    ##
667    ##
668    ##
669    ##
670    ##
671    ##
672    ##
673    ##
674    ##
675    ##
676    ##
677    ##
678    ##
679    ##
680    ##
681    ##
682    ##
683    ##
684    ##
685    ##
686    ##
687    ##
688    ##
689    ##
690    ##
691    ##
692    ##
693    ##
694    ##
695    ##
696    ##
697    ##
698    ##
699    ##
700    ##
701    ##
702    ##
703    ##
704    ##
705    ##
706    ##
707    ##
708    ##
709    ##
710    ##
711    ##
712    ##
713    ##
714    ##
715    ##
716    ##
717    ##
718    ##
719    ##
720    ##
721    ##
722    ##
723    ##
724    ##
725    ##
726    ##
727    ##
728    ##
729    ##
730    ##
731    ##
732    ##
733    ##
734    ##
735    ##
736    ##
737    ##
738    ##
739    ##
740    ##
741    ##
742    ##
743    ##
744    ##
745    ##
746    ##
747    ##
748    ##
749    ##
750    ##
751    ##
752    ##
753    ##
754    ##
755    ##
756    ##
757    ##
758    ##
759    ##
760    ##
761    ##
762    ##
763    ##
764    ##
765    ##
766    ##
767    ##
768    ##
769    ##
770    ##
771    ##
772    ##
773    ##
774    ##
775    ##
776    ##
777    ##
778    ##
779    ##
780    ##
781    ##
782    ##
783    ##
784    ##
785    ##
786    ##
787    ##
788    ##
789    ##
790    ##
791    ##
792    ##
793    ##
794    ##
795    ##
796    ##
797    ##
798    ##
799    ##
800    ##
801    ##
802    ##
803    ##
804    ##
805    ##
806    ##
807    ##
808    ##
809    ##
810    ##
811    ##
812    ##
813    ##
814    ##
815    ##
816    ##
817    ##
818    ##
819    ##
820    ##
821    ##
822    ##
823    ##
824    ##
825    ##
826    ##
827    ##
828    ##
829    ##
830    ##
831    ##
832    ##
833    ##
834    ##
835    ##
836    ##
837    ##
838    ##
839    ##
840    ##
841    ##
842    ##
843    ##
844    ##
845    ##
846    ##
847    ##
848    ##
849    ##
850    ##
851    ##
852    ##
853    ##
854    ##
855    ##
856    ##
857    ##
858    ##
859    ##
860    ##
861    ##
862    ##
863    ##
864    ##
865    ##
866    ##
867    ##
868    ##
869    ##
870    ##
871    ##
872    ##
873    ##
874    ##
875    ##
876    ##
877    ##
878    ##
879    ##
880    ##
881    ##
882    ##
883    ##
884    ##
885    ##
886    ##
887    ##
888    ##
889    ##
890    ##
891    ##
892    ##
893    ##
894    ##
895    ##
896    ##
897    ##
898    ##
899    ##
900    ##
901    ##
902    ##
903    ##
904    ##
905    ##
906    ##
907    ##
908    ##
909    ##
910    ##
911    ##
912    ##
913    ##
914    ##
915    ##
916    ##
917    ##
918    ##
919    ##
920    ##
921    ##
922    ##
923    ##
924    ##
925    ##
926    ##
927    ##
928    ##
929    ##
930    ##
931    ##
932    ##
933    ##
934    ##
935    ##
936    ##
937    ##
938    ##
939    ##
940    ##
941    ##
942    ##
943    ##
944    ##
945    ##
946    ##
947    ##
948    ##
949    ##
950    ##
951    ##
952    ##
953    ##
954    ##
955    ##
956    ##
957    ##
958    ##
959    ##
960    ##
961    ##
962    ##
963    ##
964    ##
965    ##
966    ##
967    ##
968    ##
969    ##
970    ##
971    ##
972    ##
973    ##
974    ##
975    ##
976    ##
977    ##
978    ##
979    ##
980    ##
981    ##
982    ##
983    ##
984    ##
985    ##
986    ##
987    ##
988    ##
989    ##
990    ##
991    ##
992    ##
993    ##
994    ##
995    ##
996    ##
997    ##
998    ##
999    ##
1000   ##
```

```
19 import shutil ...
    ##
20 import fileinput ...
    ##
21 import tempfile ...
    ##
22 import numpy as np ...
    ##
23 import scipy.io as sio ...
    ##
24 import multiprocessing as mp ...
    ##
25 import matplotlib.pyplot as plt ...
    ##
26 import scipy.interpolate as interp ...
    ##
27 from os.path import dirname, split, join ...
    ##
28 from random import randint ...
    ##
29 from scipy.optimize import minimize ...
    ##
30 from scipy.optimize.minpack import leastsq ...
    ##
31 from scipy.misc import comb ...
    ##
32 from scipy import shape ...
    ##
33 ...
    ##
34 class RepeatingTimer(threading._Timer): ...
    ##
35 ...
    ##
36 def run(self): ...
    ##
37 ''' ...
    ##
38 ''' ...
    ##
39 while True: ...
    ##
40     self.finished.wait(self.interval) ...
    ##
41     if self.finished.is_set(): ...
    ##
42         return ...
    ##
```



```
43         else: ...
44         ##
45         self.function(*self.args, **self.kwargs) ...
46         ##
47     def status(): ...
48         print " still working... " ...
49         ##
50     nastran = './path_to_nastran_exe' ...
51     python27 = "C:\\Program Files (x86)\\python27\\python.exe" ...
52     python = "python" ...
53     pi = np.pi ...
54     ##
55     cwd = os.getcwd() ...
56     ##
57     opt_dir = os.path.dirname(cwd) ...
58     ##
59     #####
60     #####
61     #                               F U N C T I O N                               # ...
62     #
63     #####
64     #####
65     def launch_gust(RBF_weights):
66         '''
67         Launch 'Gust.py' analysis and run 'Post_process.py'
68         '''
69         cwd = os.getcwd()
70         for line in fileinput.input(os.path.join(cwd, 'Configuration_File.txt'), ...
71         inplace=1):
72             data = line.split()
73             try:
74                 CARD = data[0].strip()
75             except:
76                 continue
```

```
76     if (CARD == 'RBF_weights'):
77         updated_line = line.replace(data[1], ','.join(str(RBF_weights[i]) for ...
    i in range(nBasis)))
78         print updated_line,
79
80     else:
81         print line,
82
83     ## Run 'Gust.py' analysis to obtain MPFs
84     print '\nRunning gust analysis...'
85     p1 = subprocess.Popen([python + ' ' + os.path.join(cwd, 'Gust.py')],
86                           stdin=subprocess.PIPE, shell=True)
87     stdout, stderr = p1.communicate()
88     print stdout
89
90     ## Run 'Post_process.py' to evaluate loads, displacements, velocities and ...
    accelerations
91     p2 = subprocess.Popen([python + ' ' + os.path.join(cwd, 'Post_process.py')],
92                           stdin=subprocess.PIPE,
93                           stdout=subprocess.PIPE, shell=True)
94     stdout, stderr = p2.communicate()
95     print stdout
96
97     accls = read_res_file(os.path.join(cwd, 'Accls.res'))
98
99     return accls
100
101 def point_plus(manager, vars, d_point, eps=0.01):
102     '''
103     Run pansol with plus  $\Delta$  on 1st gust point
104     '''
105     vars[d_point] = vars[d_point] + eps
106
107     # Mandate the process to a temp subfolder
108     files2cp = ['Configuration_File.txt', 'Gust.py', 'Post_process.py' ]
109     cwd = os.getcwd()
110     tmpdir = tempfile.mkdtemp(dir=cwd)
111     shutil.copytree(os.path.join(cwd, 'PyFunctions'), os.path.join(tmpdir, ...
    'PyFunctions'))
112     all_files = os.listdir(cwd)
113     for filename in all_files:
114         full_filename = os.path.join(cwd, filename)
115         if (os.path.isfile(full_filename)):
116             if filename in files2cp:
117                 shutil.copy(full_filename, tmpdir)
118             else:
119                 pass
120
```

```
121     os.chdir(os.path.join(cwd, tmpdir))
122
123     accls = launch_gust(vars)
124     manager_entry = 'accls_plus_Δ_' + str(d_point)
125     manager[manager_entry] = accls
126
127     # Go back to parent folder and delete temporary folder
128     os.chdir('..')
129     shutil.rmtree(tmpdir)
130
131 def point_minus(manager, vars, d_point, eps=0.01):
132     '''
133     Run pansol with minus Δ on 1st gust point
134     '''
135     vars[d_point] = vars[d_point] - eps
136
137     # Mandate the process to a temp subfolder
138     files2cp = ['Configuration_File.txt', 'Gust.py', 'Post_process.py' ]
139     cwd = os.getcwd()
140     tmpdir = tempfile.mkdtemp(dir=cwd)
141     shutil.copytree(os.path.join(cwd, 'PyFunctions'), os.path.join(tmpdir, ...
142     'PyFunctions'))
143     all_files = os.listdir(cwd)
144     for filename in all_files:
145         full_filename = os.path.join(cwd, filename)
146         if (os.path.isfile(full_filename)):
147             if filename in files2cp:
148                 shutil.copy(full_filename, tmpdir)
149             else:
150                 pass
151
152     os.chdir(os.path.join(cwd, tmpdir))
153
154     accls = launch_gust(vars)
155     manager_entry = 'accls_minus_Δ_' + str(d_point)
156     manager[manager_entry] = accls
157
158     # Go back to parent folder and delete temporary folder
159     os.chdir('..')
160     shutil.rmtree(tmpdir)
161
162 def FD_parallel(vars, eps, method):
163     '''
164     Launch multiprocessing to evaluate FD gradient
165     '''
166     manager = mp.Manager().dict() ## Create a dict that will contain my results
167
168     processes = []
```

```

168     for i in range(len(vars)):
169         dpoint_plus = mp.Process(target=point_plus, args=(manager, vars, i, eps))
170         processes.append(dpoint_plus)
171         if method=='central':
172             # Use for central difference:
173             dpoint_minus = mp.Process(target=point_minus, args=(manager, vars, i, ...
eps))
174             processes.append(dpoint_minus)
175
176     print "\n\nLaunching processes in parallel..."
177     for p in processes:
178         p.start()
179
180     print "\n\nWaiting for all design points..."
181     for p in processes:
182         p.join()
183
184     print "\n\nMultiprocessing is done!"
185
186     print "Saving to matlab structure..."
187     mat_name = 'accls_ths.mat'
188     sio.savemat(mat_name, manager)
189     print "Done! Dictionary saved to %s" %(mat_name)
190
191     return manager
192
193 def FD_grad(vars, main_norm, method='forward', eps=0.01):
194     '''
195     Obtain Finite Difference gradient in parallel processing
196     -----
197     List of inputs:
198         vars(array,float): input array of design points
199         main_norm(float): used for 'forward' difference
200         method(string, optional): 'forward' (default), 'central'
201         eps(float, optional): step size
202     '''
203
204     FD_grad = np.zeros((len(vars),), float)
205
206     if method=='forward':
207         manager = FD_parallel(vars, eps, method)
208         for i in range(0, len(vars)):
209             manager_entry_plus = 'accls_plus_Δ_' + str(i)
210             FD_diff = target_accls - manager[manager_entry_plus]
211             FD_norm = np.linalg.norm(FD_diff,2)
212             FD_grad[i] = (FD_norm - main_norm)/eps
213
214     return FD_grad

```

```
215
216     elif method=='central':
217         manager = FD_parallel(vars, eps, method)
218         for i in range(0, len(vars)):
219             manager_entry_plus = 'accls_plus_Δ_' + str(i)
220             manager_entry_minus = 'accls_minus_Δ_' + str(i)
221             FD_Pdiff = target_accls - manager[manager_entry_plus]
222             FD_Pnorm = np.linalg.norm(FD_Pdiff,2)
223             FD_Mdiff = target_accls - manager[manager_entry_minus]
224             FD_Mnorm = np.linalg.norm(FD_Mdiff,2)
225             FD_grad[i] = (FD_Pnorm - FD_Mnorm)/(2*eps)
226
227         return FD_grad
228
229 def eval_FD_grad(vars):
230     '''
231     Gradient of Objective function
232     -----
233     '''
234     print "\nCalling eval_FD_grad..."
235
236     norm = int_diff(vars)
237     print " Norm evaluated in eval_FD_grad: ", norm
238     FDgrad = FD_grad(vars, norm, eps=1.e-03) # Evaluate FD grad in parallel; ...
239     method and eps are optional, default are 'forward' and 0.01, respectively.
240     print "\nJacobian of Obj. function with shape %s: \n" %(shape(FDgrad))
241     print FDgrad
242     return FDgrad
```

Appendix D

New ways of documents management

D.1 Documents Management

Many documents are required throughout the duration of the project lifecycle. These are identifiable within three different categories: **product characteristics** - documents that describe the behaviour and the properties of the product e.g. the ARD; **project management** - documents that give information on the means to manage the overall project and the relative daily activities e.g. the Project Plan; **quality records** - documents that allow to check that relevant controls, actions and decisions have been carried out during the project lifecycle e.g. quality gate spreadsheets. A detailed list of deliverables is given below. These documents need to be initialised, updated and/or validated by the end of the M3/M5 reviews.

Project Plan (PP)

The person responsible for the creation of the Project Plan is the Project Manager who is also responsible for its publishing, application and maintenance. The main objectives of this document are to define a complete set of activities that are necessary to ensure the proper planning, organisation, tracking and leadership of the project and also to assure project members and stakeholders that the project progresses according to the specified requirements and the established plans. It integrates an organisational chart; a product breakdown structure that represents the product in a tree form with uniform and consistent sub-parts; a deliverables list divided in technical deliverables and project management deliverables; a work breakdown

structure for the activities that must be done to accomplish the project; the sub-contracting strategy and, finally, plans for resource management, training, communication and development and monitoring and control. The Project Plan is signed-off by the Information Software Project Leader, the Business Project Leader and the Project Manager as well as the Quality Responsible who is in charge of the validation of the quality of the document.

Financial Business Case (FBC)

The project leader is accountable for the development of the Financial Business Case but s/he must be supported by the relevant Financial Controller. The FBC is needed for the cost/benefit analysis for a business improvement project so that the project costs can be compared against the potential benefits related to revenues, unit costs, leadtime, etc. It establishes if the return on investment is adequate and it is mandatory for projects above defined budget thresholds (usually above £500k). The PM and the cost controller sign off the document and are responsible for it.

Risk, Issues, Action follows-up (RIA)

The RIA is a document (usually in the form of a spreadsheet) used to record and track risks, issues and follow-up actions. A risk is an event or condition that may occur with some degree of uncertainty, and which will have a negative impact on objectives; it is about an uncertain future event with no impact being experienced at present. An issue is an event or condition that has happened and may already be having an impact on objectives. A follow-up action is required to respond and resolve risks and issues. The PM is responsible for risks and issues managements whilst the ISPL and the QR are responsible for the follow-ups.

Business Requirements Dossier (BRD)

The Business Requirements Dossier is a responsibility of the PM and it is needed to document the business context, objectives and capabilities that are to be satisfied by the project. It describes the targeted business context, defines the business requirements and the non-functional requirements or services such as security rules, deployment rules and service indicators. The BRD is structured to include an “AS-IS” representation of the current context that is useful to identify the improvement opportunities existing within the business; it then describes the “TO-BE” scenario to clearly define the scope of the project and identify the expected outcome of the solution being proposed by specifying the boundaries of the project as well as its contents and the stakeholders. A tabular list of the functional and non-functional requirements then follows in the structure of the document; these are also collected in a separate spreadsheet known as the Requirements Repository. Means of compliance (i.e. the methods used for validating the

coverage of the requirements by the proposed solution) must also be specified in the BRD. As mentioned earlier, a quality gate exists for the BRD; the quality responsible is therefore in charge of the quality assessment of this document, which need finally be signed off by the PM, the BPL and the ISPL.

Architecture Dossier (ARD)

The Architecture Dossier is initiated by the PM during the Concept Phase of the project (between M3 and M5) and completed during the Solution Definition Phase (soon after M5); the scope of the document is to formalise the architecture description of the target solution. It is structured into two parts, one of which describes the architecture and the other presents the system requirements and traceability. The architecture is described in terms of functional view, design constraints, architecture of system and system components, system infrastructure, architecture assessment and components catalogue. The ARD can be divided into two versions: M5 version and M7 version. In fact, when initiated before M5, it presents different architecture candidates for the proposed solution whilst only one target architecture “survives” in the M7 version of the document. A quality gate also exists for this document and it is the responsibility of the QR to assess it; the ARD is finally signed by the PM or ISPL, the Enterprise Domain Architect, the Service Line Manager, the IT Project leader and/or the Infrastructure Architect.

Concept Phase Dossier (CPD)

The Concept Phase Dossier is a summary of all the activities performed during the concept phase of the lifecycle and it is due and validated for the M5 - it is not updated after. It gives a description of the problem to be solved and presents different ways to achieve a solution that meets the requirements. It is not a mandatory document and it is up to the PM, who is responsible for it, to decide whether it should or shouldn't be created; however, it provides a good understanding of the preliminary problem analysis, the preliminary solution design and the development, deployment and make-or-buy decision. A broader description of the content of the CPD must be provided in the BRD (for the preliminary problem analysis), the ARD (for the preliminary solution design) and the PP (for the development, deployment and make-or-buy strategy part).

Configuration Management Plan (CMP)

The Configuration Management Plan is specific for the project phase and does not provide information on the support phase; it has to be initialised by the PM at M5. It is used to describe the product structure and to identify its constituents, which are treated as single

entities that evolve over time. Furthermore, it includes information on change management and release management or in other words, it provides a history of the modifications and the planned program releases and project versions. The reader will be able to understand what changes have been made to the software and to what extent they affects its operation. The way of delivering the work product must also be established in the CMP.

IS Deployment Plan

The IS Deployment Plan is a compulsory document that has to be initialised at the beginning of the Solution Definition phase; the ISPL and the PM are responsible for creating and updating the plan. The objective of this document is to provide information on the strategy, context, risks and constraints, resources and actions that are relative to the deployment of the solution.

Service Level Requirements (SLR)

The Service Level Requirements document translates business and system non-functional requirements into specific and quantifiable operational requirements. It defines and lists the characteristics of the service in such a way that the service can be operated; it includes a description of the performance of each service item as well as its availability, volume, data quality, specific key performance indicators (KPIs), the frequency of delivery, the target service level and how it is monitored. The SLR is initialised and required at milestone M5 to certify the commitment between the IS team and the business and it has to be finalised at milestone M7, together with the finalised architecture. The ISPL fills in the document and is therefore held accountable for it.

In order to neatly manage these documents, a well-established process is a must. Nonetheless, the PM is in charge of nominating people responsible for documentation management and for controlling the sharing of the working documents. These people have manuals, guidelines and several tools available to give at any time, to all internal and external actors of the project, information on how documents are organised, established and controlled. An overview of one of the tools available in Airbus is given below.

The following is a platform used to store documents in a way compliant to the project lifecycle. It is accessed via a main page that gives general information about the project i.e. the names of the main actors and the current status of the project (Figure D.1).

By accessing the working area, a folder structure divides the documents into “Project Management” and “Technical Deliveries” to be consistent with the categories described earlier (Figure

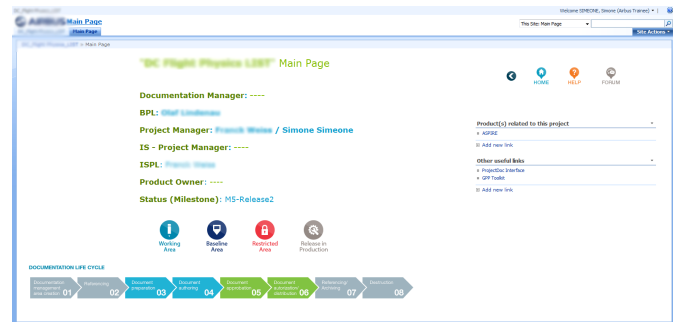


FIGURE D.1: Documentation management main page

D.2).

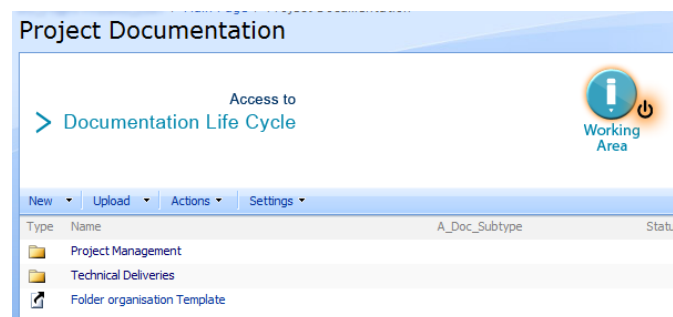


FIGURE D.2: Working area: Home

The “Project Management” area includes 7 folders that enclose all the documents relevant to Project Plans, Project Monitoring & Control, Meeting & Communications, Quality Management, Configuration Management, Supplier Management and Project Closure (Figure D.3).

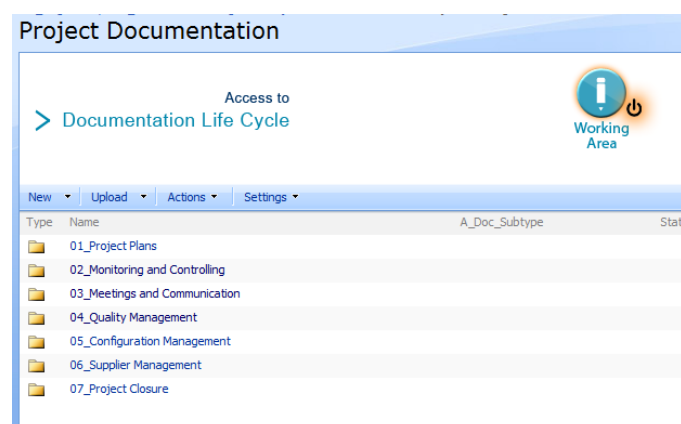


FIGURE D.3: Working area: Project Management

The “Technical Deliveries” area, instead, includes 12 folders that enclose all documents relevant

to Opportunity Study, Business Processes & Requirements, Concept & Architecture, Specification, Development, Testing, User Acceptance, Users Documentation, Installation & Deployment, Operation, Support and Product Configuration (Figure D.4)

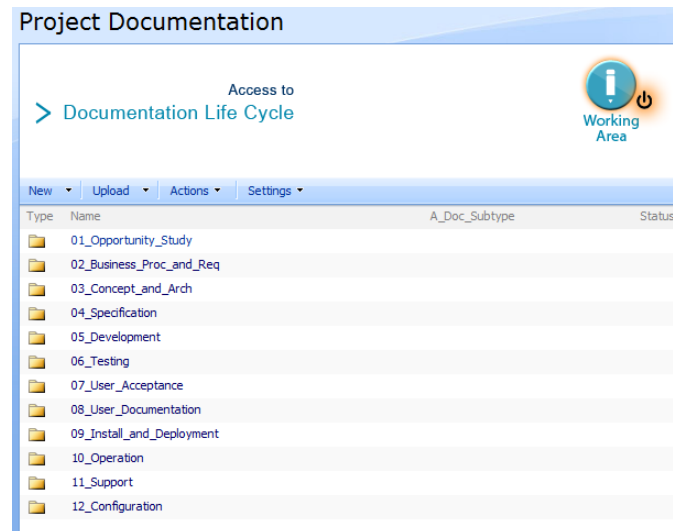


FIGURE D.4: Working area: Technical Deliveries

Several actions are available to the actors of the project to create, modify, upload or download documents; an history of all modification along with a footprint is recorded and the possibility to retrieve old versions of the same document is also available.

D.2 Alternative solutions

As shown so far, managing all these documents is highly complex and requires a huge effort and time commitment from almost all the actors of the project. However, it can be demonstrated that most of the documents required for passing the milestone reviews are either replicating the same information over and over again or remain unused or even unread during the development itself of the software or soon after its entry into service. For example, let us consider one of the documents described in the previous section: the Concept Phase Dossier. As discussed, three main parts constitutes this document: the preliminary problem analysis, the preliminary solution design and the development, deployment and make-or-buy strategy. These three parts are replicated and extended, respectively, in the BRD, the ARD and the Project Plan. Furthermore, the CPD will not be updated neither used after M5 but, of course, it still requires some time and effort to be created.

When thinking about developing an alternative solution to substitute the creation of project documentation, it is important to first understand the reasons that exist behind the need for these documents and guarantee that they will still remain covered, somehow. Ambler, in his article on “*Agile/Lean Documentation: Strategies for Agile Software Development*” [90], gives a full picture on this matter, providing a lists of “questionable” reasons behind the creation of the project documentation and the means to challenge them. What seems to be really important is the need of the requester of the documents to be seen in control of the whole process; his/her go ahead signature on, for example, the ARD or BRD puts him/her in the position of pretending that whatever was documented shall be delivered by the end of the process. However, these are the same people that would not want to end up in a situation of failure entirely due to the time lost on the creation of the documents rather than on the software development. Furthermore, as mentioned earlier, it is still mistakenly believed that documentation is directly responsible of the project success but, as discussed, it is a belief that is entirely based on a particular historical event. It then results that most documents are often requested only because it is a habit of the requester, something that s/he has been doing for years without giving too much care on what the document is actually delivering in terms of added information, or because the process simply instructs to do so. Another peculiar reason that exists behind the need for documents is that the stakeholders want reassurance that everything is proceeding well because they are investing significant resources and want to know at any time that their investment is being well spent. Last but not least, documents are also used as a communication mean. Hence, it is a matter of convincing people they are in control of the situation, questioning them about the real purpose behind each document, giving them assurance that their investment is being well spent and demonstrating that documents are not the best way of communicating information. In the framework of agile development, these questions are tackled, for example, by bringing the stakeholders of the project on board of the development and asking them active feedback on “beta” versions of the software or by setting up regular meetings to facilitate the communication of issues, risks and actions between the IS team and the business. An approach similar to the agile methodology is given by *Rapid Application Development (RAD)* (Figure D.5).

RAD is a methodology for software development which puts more emphasis on the development phase of the product rather than on its planning tasks and documentation [93]. It is a model based on the concept that higher-quality products can be developed faster through more practical processes [94], such as:

- gathering requirements using workshops or focus groups;

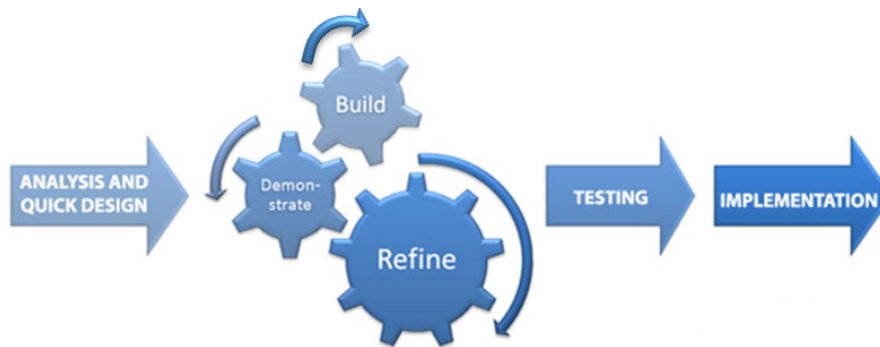


FIGURE D.5: Rapid Application Development process

- prototyping and early, reiterative user testing of designs;
- the re-use of software components;
- design improvements deferred to the next product version;
- less formality in reviews and other team communication.

It is an important methodology because “the ability of an information system to evolve to meet new requirements is one of the key quality characteristics, but systems must also be capable of rapid evolution if they are to deliver real value to the business community in this volatile business environment. Systems that cannot evolve rapidly offer little or no support to their users, who in turn become less responsive to their environment and consequently incur increased risk of failure in the marketplace” [95].

Bibliography

- [1] J. R. Wright and J. E. Cooper. *Introduction to Aircraft Aeroelasticity and Loads*. Wiley, 2007.
- [2] F. M. Hoblit. *Gust Loads on Aircraft: Concepts and Applications*. AIAA Education Series, AIAA, Washington, DC, 1988.
- [3] John C. Houbolt. Atmospheric Turbulence. *AIAA Journal*, 11 (04):421–437, 1973.
- [4] L. Bensch, H. Henrichfreise, J. Gusset, and L. Merz. Method for reconstructing gusts and structural loads at aircraft, in particular passenger aircraft. World Intellectual Property Organization, June 2007.
- [5] Airbus S.A.S. Global Market Forecast: Growing Horizons 2017/2036. *Art & Caractère*, April(4):1–127, 2017.
- [6] Simone Simeone, Christian Agostinelli, Thomas Rendall, and Abdul Rampurawala. Gust reconstruction from digital flight data recorder via numerical optimisation. In *57th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 1–19, Reston, Virginia, Jan 2016. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-392-6. doi: 10.2514/6.2016-1484.
- [7] S. Simeone, T. Rendall, and A. Da Ronch. A gust reconstruction framework applied to a nonlinear reduced order model of a wing typical section. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, pages 1–16, Grapevine, Texas, USA, Jan 2017. American Institute of Aeronautics and Astronautics. ISBN 978-1-62410-392-6. doi: 10.2514/6.2017-0634.
- [8] S. Simeone, T. Rendall, S. Williams, C. Wales, J.E. Cooper, D. Jones, and A.L. Gaitonde. Reconstruction of gust velocity profiles via potential flow, cfd and rom techniques. In

- International Forum on Aeroelasticity and Structural Dynamics*, pages 1–18, Como, Italy, Jun 2017. IFASD. ISBN IFASD-2017-182.
- [9] M. C. Jackson. *Systems Thinking: Creative Holism for Managers*. John Wiley and Sons, Ltd, 2003.
- [10] P. Checkland. O. r. and the Systems Movement: Mappings and Conflicts. *The Journal of Operational Research Society*, 34(8):661–675, 1983.
- [11] Certification specifications for large aeroplanes. EASA, Amendment 3, September 2003.
- [12] Kermit G.Pratt and Walter G.Walker. A revised Gust Load Formulats and re-evaluation of V-G data taken on civil transport airplanes from 1933 to 1950. NACA TR1206. Technical report, 1954.
- [13] Edwin Bidwell Wilson. Theory of an Aeroplane Encountering Gusts , II. Technical Report 3, American Philosophical Society, 1917. URL <http://www.jstor.org/stable/983947>.
- [14] J. Fuller. Evolution and future development of airplane gust loads design requirements. *1997 World Aviation Congress*, 32(2):235–246, 1997. ISSN 0021-8669. doi: 10.2514/6.1997-5577.
- [15] An Introduction to Rigid Aeroplane Response to Gusts and Atmospheric Turbulence. ESDU 04024, November 2004.
- [16] H. G. Kussner. Stresses Produced in Airplane Wings by Gusts. Technical report, 1932. URL <http://go.nasa.gov/2cw66GC>.
- [17] Harry Press and Bernard Mazelsky. A study of the application of power-spectral methods of generalized harmonic analysis to gust loads on airplanes. Technical report, 1953. URL <http://ntrs.nasa.gov/search.jsp?R=19930090975>.
- [18] Hubert I. Flomenhoft. Brief history of gust models for aircraft design. *Journal of Aircraft*, 31(5):1225–1227, 1994. ISSN 0021-8669. doi: 10.2514/3.46637.
- [19] Joint aviation requirements - jar-25 large aeroplanes. Federal Aviation Regulations, May 1994. Change 14 - Book 1 - Subpart B.
- [20] L. Usik. *Spectral Analysis of Signals*. 2009. ISBN 9780470823743. doi: 10.1002/9780470823767.ch2. URL <http://doi.wiley.com/10.1002/9780470823767.ch2>.

- [21] T. von Kármán. Progress in the Statistical Theory of Turbulence. In *Proceedings of the National Academy of Sciences of the United States of America*, number 0027-8424, June 1948. doi: 10.1073/pnas.34.11.530.
- [22] J. C. Houbolt, K. G. Pratt, and R. Steiner. Dynamic response of airplanes to atmospheric turbulence including flight data on input and response. Technical Report TR R-199, NASA Langley Research Center, 1964. URL <http://hdl.handle.net/2027/uiug.30112106585620>.
- [23] Raymond L. Bisplinghoff, Holt Ashley, and Robert L. Halfman. *Aeroelasticity*. 1983. ISBN 978-0-486-69189-3.
- [24] D. R. Jones, M. Schonlau, and W. Welch. Efficient global optimization of expensive black-box functions. *Journal of Global Optimization*, 13:455–492, 1998.
- [25] John C. Houbolt. Design Manual for Vertical Gusts Based on Power Spectral Techniques. *Air Force Flight Dynamics Lab (AFFDL-TR-70-106)*, 1970.
- [26] J. K. Zbrozek. The Relationship between the Discrete Gust and Power Spectra Presentations of Atmospheric turbulence, with a suggested Model of Low-Altitude Turbulence. *Aeronautical Research Council Reports and Memoranda*, R. & M. No. 3216, 1961.
- [27] M. C. Kim, A. M. Kabe, and S. S. Lee. Atmospheric Flight Gust Loads Analysis. *Space and Missile Systems Center Air Force Materiel Command*, TR-99(1534)-5, 1999.
- [28] B.-T. Wang. Prediction of Impact and Harmonic Forces Acting on Arbitrary Structures: Theoretical Formulation. *Mechanical Systems and Signal Processing*, 16(6):935–953, 2002. ISSN 08883270. doi: 10.1006/mssp.2002.1505. URL <http://www.sciencedirect.com/science/article/pii/S0888327002915050>.
- [29] Patrik Nordberg. *Time Domain Methods for Load Identification of Linear and Nonlinear Systems*. Doktorsavhandlingar vid Chalmers tekniska. Ny serie, no: 2170. Department of Applied Mechanics, Chalmers University of Technology,, 2004. ISBN 91-7291-489-0.
- [30] H Henrichfreise, L Bensch, J Jusseit, L Merz, and M Gojny. Estimation of gusts and structural loads for commercial aircraft. In *Int. Forum on Aeroelasticity and Structural Dynamics (IFASD)*, pages 1–11, Seattle, USA, 2009. IFASD.

- [31] Mike Montel and Frank Thielecke. Validation of a Model-Based Structural Loads Monitoring System using the Flight Test Aircraft UW-9 Sprint. (June):1–17, 2015. doi: 10.2514/6.2015-2237.
- [32] T. Theodorsen. General theory of aerodynamic instability and the mechanism of flutter. Technical report, 1935.
- [33] A. Da Ronch, K. Badcock, Y. Wang, A. Wynn, and R. Palacios. Nonlinear Model Reduction for Flexible Aircraft Control Design. *AIAA Atmospheric Flight Mechanics Conference*, pages 1–23, 2012. doi: 10.2514/6.2012-4404. URL <http://arc.aiaa.org/doi/abs/10.2514/6.2012-4404>.
- [34] L. Meirovitch. *Dynamics and Control of Structures*. Wiley, New York, 1989.
- [35] Chris L Pettit and Philip S Beran. Effects of Parametric Uncertainty on Airfoil Limit Cycle Oscillation. *Journal of Aircraft*, 40(5):1004–1006, sep 2003. ISSN 0021-8669. doi: 10.2514/2.6889. URL <http://arc.aiaa.org/doi/abs/10.2514/2.6889>.
- [36] Hamed Haddad Khodaparast, Simon Coggon, Michael Ian Friswell, and Je Cooper. The Effects of Structural Nonlinearity on the Dynamic Response to Aeroelastic The effects of structural nonlinearity on the dynamic response to aeroelastic gust models. In *27th International Conference on Noise and Vibration Engineering (ISMA 2016)*, number September, 2016.
- [37] J. Katz and A. Plotkin. *Low-Speed Aerodynamics*. Cambridge Aerospace Series. Cambridge University Press, 2001. ISBN 9780521665520. URL <http://books.google.co.uk/books?id=rAS1DmBRL08C>.
- [38] J. Gordon Leishman. *Principles of Helicopter Aerodynamics*. Cambridge Aerospace Series, 2006. ISBN 978-0-521-85860-1. doi: 10.1002/1521-3773(20010316).
- [39] Steven L. Brunton and Clarence W. Rowley. Empirical state-space representations for Theodorsen’s lift model. *Journal of Fluids and Structures*, 38(April):174–186, 2013. ISSN 08899746. doi: 10.1016/j.jfluidstructs.2012.10.005.
- [40] Robert L Halfman. Experimental Aerodynamic Derivatives of a Sinusoidally Oscillating Airfoil in Two-Dimensional Flow. Technical Report NACA-TN-2465, National Advisory Committee for Aeronautics (NACA), November 1951.

- [41] Y.C. Fung. *An Introduction to the Theory of Aeroelasticity*. Dover Phoenix Edition: Engineering. Dover Publications, 2002. ISBN 9780486495057. URL <https://books.google.co.uk/books?id=KetYrthKfEgC>.
- [42] T. H. Von Karman. Airfoil Theory for Non-Uniform Motion. *Journal of the Aeronautical Sciences*, 5(10):379–390, aug 1938. doi: 10.2514/8.674. URL <https://doi.org/10.2514/8.674>.
- [43] C. Wales, D. Jones, and A. Gaitonde. Prescribed Velocity Method for Simulation of Aerofoil Gust Responses. *Journal of Aircraft*, 52(1):1–13, 2015. ISSN 0021-8669. doi: 10.2514/1.C032597. URL <http://arc.aiaa.org/doi/abs/10.2514/1.C032597>.
- [44] D. Jones and A. Gaitonde. Future Fast Methods for Loads Calculations: The FFAST Project. In *Innovation for Sustainable Aviation in a Global Environment Proceedings of Aerodays*, pages 110–115, Madrid, Spain, 2011. IOS Press BV.
- [45] Stephen P. Williams, Dorian Jones, Ann Gaitonde, Chris Wales, and Samantha J. Huntley. Application of Reduced Order Models in Aircraft Gust Response Studies. *46th AIAA Fluid Dynamics Conference*, (June):1–23, 2016. doi: 10.2514/6.2016-4261. URL <http://arc.aiaa.org/doi/10.2514/6.2016-4261>.
- [46] Stephen P. Williams, D. Jones, A. Gaitonde, C. Wales, and S. J. Huntley. Reduced order modelling of aircraft gust response for use in early design stages. In *35th AIAA Applied Aerodynamics Conference, American Institute of Aeronautics and Astronautics*, Denver, USA, June 2017. American Institute of Aeronautics and Astronautics.
- [47] J.-N. Juang and R. S. Pappa. An eigensystem realization algorithm for modal parameter identification and model reduction. *Journal of Guidance Control and Dynamics*, 8:620–627, 1985.
- [48] C. Wales, D. Jones, and A. Gaitonde. Reduced order modelling for aeroelastic aerofoil response to a gust. In *51st AIAA Aerospace Sciences Meeting*, pages 1–16, Denver, USA, 2013. American Institute of Aeronautics and Astronautics.
- [49] C. Wales, A. Gaitonde, and D. Jones. Stabilisation of reduced order models via restarting. *International Journal for Numerical Methods in Fluids*, 73(6):578–599, 2013. ISSN 1097-0363. doi: 10.1002/flid.3814. URL <http://dx.doi.org/10.1002/flid.3814>.

- [50] T. McKelvey, H. Akcay, and L. Ljung. Subspace-based multivariable system identification from frequency response data. *IEEE Transactions on Automatic Control*, 41(7):960–979, Jul 1996. ISSN 0018-9286. doi: 10.1109/9.508900.
- [51] H.H. Khodaparast and J.E. Cooper. Rapid prediction of worst-case gust loads following structural modification. *AIAA Journal*, 52:242–254, 01 2014. doi: 10.2514/1.J052031.
- [52] W.P. Rodden and E.H. Johnson. Msc/nastran aeroelastic analysis user’s guide. Technical report, MSC Software, 1994.
- [53] R. Palacios, H. Climent, A. Karlsson, and B. Winzell. Assessment of strategies for correcting linear unsteady aerodynamics using cfd or experimental results. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Madrid, Spain, 2001.
- [54] R.G.A. Silva, O.A.F. Mello, J.L.F. Azevedo, P.C. Chen, and D.D. Liu. Investigation on transonic correction methods for unsteady aerodynamics and aeroelastic analyses. *Journal of Aircraft*, 45:1890–1903, 2008. doi: 10.2514/1.33406.
- [55] R. Moreno, F. Knoblauch, R. Narisetti, and P.F. Taylor. A modification to the enhanced correction factor technique to correlate with experimental data. In *56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, Kissimmee, Florida, USA, 2015. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2015-1421.
- [56] R. Cook, D. Calderon, M.H. Lowenberg, S. Nield, J.E. Cooper, and E. Coetzee. Worst case gust prediction of highly flexible wings. In *58th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, AIAA SciTech Forum*, Grapevine, Texas, USA, 2017. American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2017-1355.
- [57] K. L Roger. Airplane math modeling methods for active control design. Technical Report AGARD-CP-228, August 1977.
- [58] W. Eversman and A. Tewari. Consistent rational-function approximation for unsteady aerodynamics. *Journal of Aircraft*, 28:545–552, 1991. doi: 10.2514/3.46062.
- [59] M. Karpel and S.H. Tiffany. Physically weighted approximations of unsteady aerodynamic forces using the minimum-state method. Technical Report NASA 3025, NASA, 1991.

- [60] L. Morino, F. Mastroddi, R. De Troia, G. L. Ghiringhelli, and P. Mantegazza. Matrix fraction approach for finite-state aerodynamic modelling. *AIAA Journal*, 33:703–711, 1995. doi: 10.2514/3.12381.
- [61] M. Mor and E. Livne. Sensitivities and approximations for aeroservoelastic shape optimization with gust response constraints. *Journal of Aircraft*, 43:1516–1527, 2006. doi: 10.2514/1.17467.
- [62] B. Moulin and M. Karpel. Gust loads alleviation using special control surfaces. *Journal of Aircraft*, 44:17–25, 2007. doi: 10.2514/1.19876.
- [63] M. Karpel, B. Moulin, and P. C. Chen. Dynamic response of aeroservoelastic systems to gust excitation. *Journal of Aircraft*, 42:1264–1272, 2005. doi: 10.2514/1.6678.
- [64] T. Kier and G. Looye. Unifying manoeuvre and gust loads analysis. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Seattle, WA, USA, 2009.
- [65] T. Kier. An integrated loads analysis model including unsteady aerodynamic effects for position and attitude dependent gust fields. In *International Forum on Aeroelasticity and Structural Dynamics (IFASD)*, Paris, France, 2011.
- [66] Andrea Castrichini. *On the Design of a Folding Wing-Tip Device for Aircraft Loads Alleviation*. PhD thesis, University of Bristol, Department of Aerospace Engineering, October 2016.
- [67] ChihKao Ma and ChihCherng Ho. An inverse method for the estimation of input forces acting on non-linear structural systems. *Journal of Sound and Vibration*, 275:953–971, 06 2003. doi: 10.1016/S0022-460X(03)00797-1.
- [68] E.J. Oosterhuis, W.B. Eindhoven, P.J.M. van der Hoogt, and A. de Boer. Force prediction via the inverse frf using experimental and numerical data from a demonstrator with tuneable nonlinearities. In *The Thirteenth International Congress on Sound and Vibration*, pages 1–8, Vienna, Austria, 2006. American Institute of Aeronautics and Astronautics.
- [69] M. Hockenhuell. Flight Incident Analysis Tool - User Guide. Airbus confidential report, November 2014.
- [70] Hirotsugu Inoue, John J Harrigan, and Stephen R Reid. Review of inverse analysis for indirect measurement of impact force. *Applied Mechanics Reviews*, 54:503–524, 11 2001. doi: 10.1115/1.1420194.

- [71] M. Cerna and A. F. Harvey. The fundamentals of fft-based signal analysis and measurement. Technical Report Application Note 041, National Instruments Corporation, 2000. URL https://www.sjsu.edu/people/burford.furman/docs/me120/FFT_tutorial_NI.pdf.
- [72] R.L. Easton. *Fourier Methods in Imaging*. The Wiley-IS&T Series in Imaging Science and Technology. Wiley, 2010. ISBN 9781119991861. URL <https://books.google.co.uk/books?id=QuIHjnXQqM8C>.
- [73] O. Dababneh, T. Kipouros, and J.F. Whidborne. Application of an Efficient Gradient-Based Optimization Strategy for Aircraft Wing Structures. *Aerospace*, 5(3):615–640, 2018. doi: 10.3390/aerospace5010003. URL <https://www.mdpi.com/2226-4310/5/1/3/pdf>.
- [74] A. D. Belegundu and T. R. Chandrupatla. *Optimization Concepts and Applications in Engineering*. Cambridge University Press, 2019. ISBN 9781108424882.
- [75] Dominic Masters, Daniel Poole, Nigel Taylor, Thomas Rendall, and Christian Allen. Influence of shape parameterisation on a benchmark aerodynamic optimisation problem. *Journal of Aircraft*, 3 2017. ISSN 0021-8869.
- [76] Raymond M. Hicks and Preston A. Henne. Wing Design by Numerical Optimization. *Journal of Aircraft*, 15(7):407–412, July 1978. doi: 10.2514/3.58379. URL <https://arc.aiaa.org/doi/10.2514/3.58379>.
- [77] Hsiao-Yuan Wu, Shuchi Yang, Feng Liu, and Her-Mann HM Tsai. Comparison of three geometric representations of airfoils for aerodynamic optimization. *16th AIAA Computational Fluid Dynamics Conference, Orlando, Florida, (June), 2003*. doi: 10.2514/6.2003-4095.
- [78] J. Hicken, J. Alonso, and C. Farhat. Introduction to Multidisciplinary Design Optimization. Stanford University Lecture Notes, 2012.
- [79] Jack Sherman and Winifred J. Morrison. Adjustment of an Inverse Matrix Corresponding to a Change in One Element of a Given Matrix. *The Annals of Mathematical Statistics*, 1950. ISSN 0003-4851. doi: 10.1214/aoms/1177729893.
- [80] William W. Hager. Updating the Inverse of a Matrix. *SIAM Review*, 1989. ISSN 0036-1445. doi: 10.1137/1031049.
- [81] Charles L. Karr, Thomas A. Zeiler, and Rajiv Mehrotra. Determining worst-case gust loads on aircraft structures using an evolutionary algorithm. *Applied Intelligence*, 20(2):135–145, Mar 2004. ISSN 1573-7497. doi: 10.1023/B:APIN.0000013336.08029.0c.

- [82] Anthony S. Pototzky, Thomas A. Zeiler, and Boyd Perry. Calculating time-correlated gust loads using matched filter and random process theories. *Journal of Aircraft*, 28:346–352, 05 1991. doi: 10.2514/3.46033.
- [83] H.H. Khodaparast, G. Georgiou, J.E. Cooper, L. Travaglini, S. Ricci, G.A. Vio, and P. Denner. Rapid prediction of worst case gust loads. In *Structures, Structural Dynamics and Materials Conference*, pages 1–22, Denver, Colorado, 2011. American Institute of Aeronautics and Astronautics.
- [84] J.G. Jones. Documentation of the linear statistical discrete gust method. Technical Report DOT/FAA/AR-04/20, Stirling Dynamics Inc. AAR-480, 2004.
- [85] S.H.M. McGuinness. *Improving Aircraft Stress-Loads Interface and Evaluation Procedures*. PhD thesis, Queen’s University Belfast, School of Mechanical and Aerospace Engineering, May 2011.
- [86] Irene Tartaruga, Jonathan E. Cooper, Mark H. Lowenberg, P.N. Sartor, Simon Coggon, and Y. Lemmens. Prediction and uncertainty propagation of correlated time-varying quantities using surrogate models. *CEAS Aeronautical Journal*, 7:29–42, 03 2016. doi: 10.1007/s13272-015-0172-1.
- [87] H. G. Daellenbach. Hard or, soft or problem structuring methods, critical systems thinking: A primer. In *Proceedings of the ORSNZ Conference Twenty Naught One*, University of Cantenbury, Christchurch, New Zealand., 2001. ORSNZ Operational Research Society of New Zealand.
- [88] NASA Systems Engineering Handbook. NASA/SP-2007-6105, Rev 1., 2007.
- [89] Neil Stolovitsky. Taking project document management beyond paper pushing. Project Management Times website. Accessed: July 2015. URL <http://www.projecttimes.com/articles/taking-project-document-management-beyond-paper-pushing.html>.
- [90] S. Ambler. Agile/lean documentation: Strategies for agile software development. Agile modeling website. Accessed: July 2015. URL <http://agilemodeling.com/essays/agileDocumentation.htm>.
- [91] Marta Bertelè, Carlo L Bottasso, Stefano Cacciola, Fabiano Daher Adegas, and Sara Delpont. Wind inflow observation from load harmonics. *Wind Energy Science*, 2(2):

- 615–640, 2017. ISSN 2366-7451. doi: 10.5194/wes-2-615-2017. URL <https://www.wind-energ-sci.net/2/615/2017/>.
- [92] Marta Bertelè, Carlo L Bottasso, and Stefano Cacciola. Wind inflow observation from load harmonics : wind tunnel validation of the rotationally symmetric formulation. *Wind Energy Science Discuss.*, (4):89–97, September 2019.
- [93] James Martin. Rapid application development. 1:5–8, 1991.
- [94] Margaret Rouse. Rapid Application Development (RAD). TechTarget website. Accessed: July 2015. URL <http://searchsoftwarequality.techtarget.com/definition/rapid-application-development>.
- [95] B. Hambling. Testing in a RAD environment. In *Post-Conference tutorial at the Sixth International Conference on Practical Software Quality Techniques.*, Austin, Texas., March 2000.