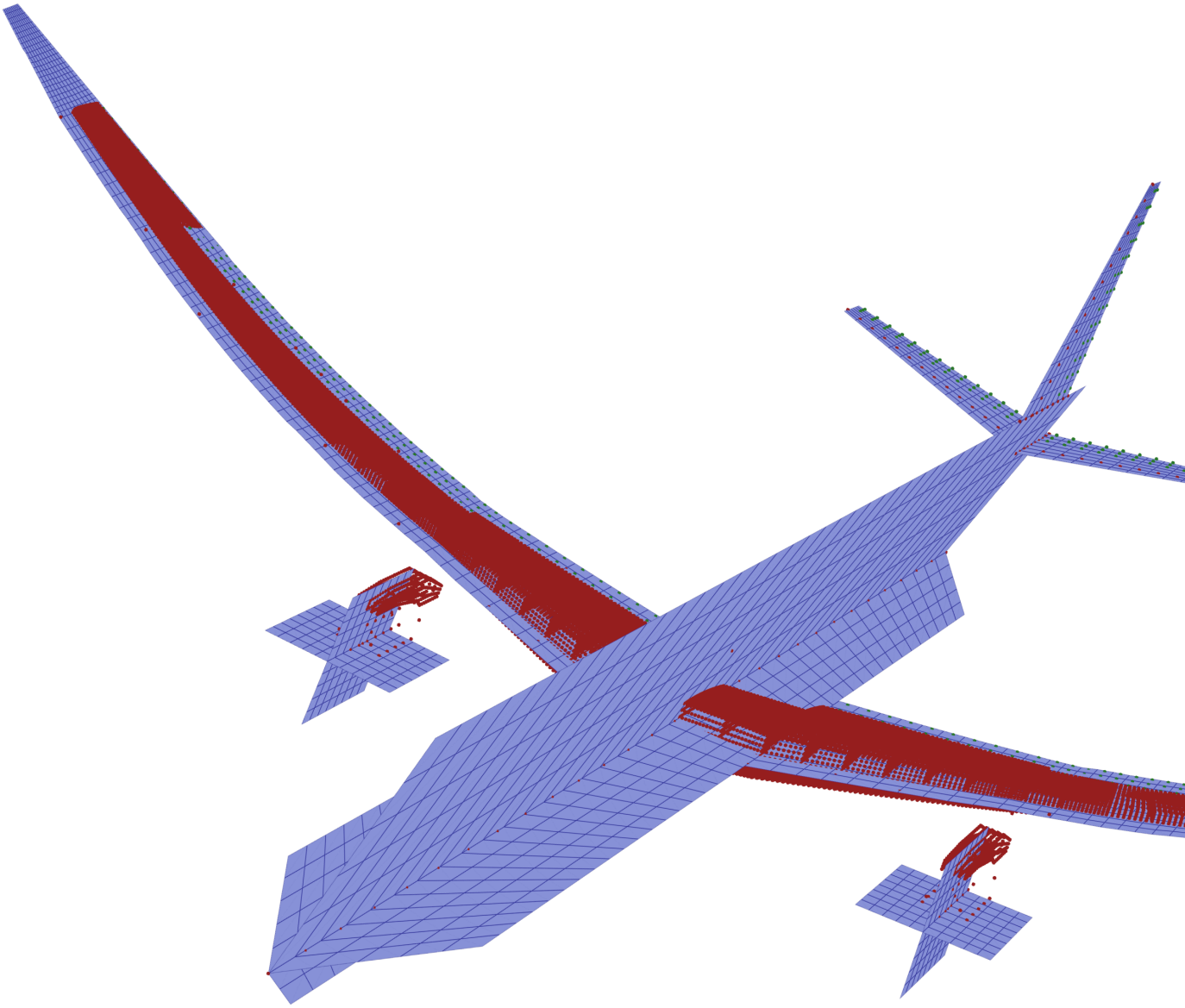
Free-Flying Aeroservoelastic State-Space Modeling

Bridging Loads Analysis and Aeroservoelasticity

O.F.H. Rots

Delft University of Technology



TU Delft Delft University of Technology

Challenge the future

FREE-FLYING AEROSERVOELASTIC STATE-SPACE MODELING

BRIDGING LOADS ANALYSIS AND AEROSERVOELASTICITY

by

O.F.H. Rots

in partial fulfillment of the requirements for the degree of

Master of Science
in Aerospace Engineering

at the Delft University of Technology,
to be defended publicly on Wednesday March 11, 2026 at 14:00.

Supervisors:	dr. X. Wang,	TU Delft
	dr. J. Sodja,	TU Delft
Committee:	prof. dr. ir. R. De Breuker,	TU Delft
	dr. C. Varriale,	TU Delft
	dr. X. Wang,	TU Delft
	dr. J. Sodja,	TU Delft

TU Delft Delft University of Technology

PREFACE

When flying, the bending wings and shudder of turbulence always fascinated me, and hence, I became interested in aeroelasticity during my master's degree. I wanted to work on a research topic that was methodological in nature, and the research area of this thesis, free-flying aeroservoelastic state-space modeling, was therefore an excellent choice. I would like to thank Assistant Professors Xuerui Wang and Jurij Sodja for their supervision of this master thesis: I enjoyed working with you and under your supervision. I am also thankful for the chance to present my research to Embraer, which encouraged the feeling of working on a topic of relevance for the next-generation of aircraft. I would also like to thank PhD Candidates Stefan De Boer and Hauke Maathuis for their support with DLR Loads Kernel and the tailoring of the Embraer Benchmark Wing respectively. Lastly, I would like to thank Professor Roeland De Breuker and Assistant Professor Carmine Varriale for finding the time to join the thesis committee.

This master thesis also concludes my education in Delft: 6,5 years that have shaped me profoundly. I vividly remember my excitement waking up in the middle of the night, and seeing an email which congratulated me on being selected for a place to study Aerospace Engineering at Delft University of Technology. As I spent much time in high school planespotting at Schiphol, simulating flights on my flight simulator, building remote-controlled model aircraft, and visiting the air shows in Le Bourget and Farnborough, I was delighted to pursue a degree in Delft and follow my passion: a feeling that remains with me today, and far into the future.

O.F.H. Rots
Delft, March 2026

List of Figures

List of Tables

LIST OF SYMBOLS

Table 1: General definitions and constants nomenclature

Index / subscript	Description
AR	Aspect ratio
$\boldsymbol{C}$	Damping matrix
C_D	Drag coefficient
C_L	Lift coefficient
$\ddot{(\cdot)}$	Second time derivative
$\dot{(\cdot)}$	First time derivative
$\boldsymbol{E}$	Selection matrix
e	Oswald efficiency factor
$\boldsymbol{F}$	Forces only
g_0	Gravitational acceleration
$\boldsymbol{K}$	Stiffness matrix
$\boldsymbol{L}$	Cut loads
$\boldsymbol{M}$	Mass matrix
$\boldsymbol{M}$	Moments only
MAC	Mean aerodynamic chord
$\boldsymbol{P}$	Forces and moments
$\boldsymbol{\Phi}$	Splining matrix
$\boldsymbol{\Pi}$	Expansion operator
$\boldsymbol{T}$	Transformation matrix
$\tilde{(\cdot)}$	Expanded aerodynamic matrix

Table 2: Reference frames and grids nomenclature

Subscript	Description
b	Body frame
c	Control surface coordinates
g	Structural/nodal grid
h	Modal/generalized coordinates
j	Aerodynamic collocation points, j-grid
k	Aerodynamic half-chord points index, k-grid
l	Aerodynamic quarter-chord points index, l-grid
m	Monitoring station grid
n	Flight physical frame
r	Rotational body DOFs
s	Accelerometer grid
t	Translational body DOFs

Table 3: State-space model order nomenclature

Index / subscript	Description
c	Number of actuators/control surfaces
m	Number of flexible modes
n	Number of aerodynamic panels
o	Number of engines
p	Number of RFA poles
q	Index of RFA pole
s	Number of cut loads
z	Number of gust zones

Table 4: State-space formulation nomenclature

Symbol	Description
$\boldsymbol{A}$	State matrix
$\boldsymbol{a}_r$	Rotational accelerations at accelerometers
$\boldsymbol{a}_t$	Translational accelerations at accelerometers
α	Angle of attack
$\boldsymbol{B}$	Input matrix
β	Sideslip angle
$\boldsymbol{C}$	Output matrix
$\boldsymbol{C}(\Theta)$	Direction cosine matrix
$\boldsymbol{D}$	Input feedthrough matrix
$\boldsymbol{\delta}$	Actuator/control surface deflection vector $\boldsymbol{\delta} = [\delta_1, \ldots, \delta_c]^T$
$\dot{\boldsymbol{\delta}}$	Actuator/control surface rate vector $\dot{\boldsymbol{\delta}} = [\dot{\delta}_1, \ldots, \dot{\delta}_c]^T$
$\boldsymbol{E}$	Descriptor matrix
$\boldsymbol{E}(\Theta)$	Euler transformation matrix
$\boldsymbol{\eta}$	Modal displacement vector $\boldsymbol{\eta} = [\eta_1, \ldots, \eta_m]^T$
$\dot{\boldsymbol{\eta}}$	Modal velocity vector $\dot{\boldsymbol{\eta}} = [\dot{\eta}_1, \ldots, \dot{\eta}_m]^T$
$\boldsymbol{F}$	Disturbance matrix
$\boldsymbol{F}_l$	Aerodynamic body force lag $\boldsymbol{F}_l = [\boldsymbol{F}_l^1, \boldsymbol{F}_l^2, \ldots, \boldsymbol{F}_l^p]^T$
$\boldsymbol{F}_l^q$	Aerodynamic body force lag per pole $\boldsymbol{F}_l^q = [F_x^q, F_y^q, F_z^q]^T$
$\boldsymbol{f}_0$	State equation bias terms
$\boldsymbol{f}_{nl}(\boldsymbol{x})$	State equation nonlinearities
$\boldsymbol{G}$	Disturbance feedthrough matrix
$\boldsymbol{g}$	Gust state vector $\boldsymbol{g} = [g_{1,1}, g_{1,2}, \ldots, g_{z,1}, g_{z,2}]^T$
$\boldsymbol{g}_b(\boldsymbol{\Theta})$	Body gravity vector
γ	Flight path angle
$\boldsymbol{H}$	Output descriptor matrix
$\boldsymbol{h}_0$	Output equation bias terms
$\boldsymbol{h}_{nl}(\boldsymbol{x})$	Output equation nonlinearities
$\boldsymbol{L}_l$	Aerodynamic cut load lag $\boldsymbol{L}_l = [\boldsymbol{L}_l^1, \boldsymbol{L}_l^2, \ldots, \boldsymbol{L}_l^p]^T$
$\boldsymbol{L}_l^q$	Aerodynamic cut load lag per pole $\boldsymbol{L}_l^q = [L_1^q, L_2^q, \ldots, L_s^q]^T$
$\boldsymbol{L}_m$	Cut loads at monitoring stations
$\boldsymbol{M}_l$	Aerodynamic body moment lag per pole $\boldsymbol{M}_l = [\boldsymbol{M}_l^1, \boldsymbol{M}_l^2, \ldots, \boldsymbol{M}_l^p]^T$
$\boldsymbol{M}_l^q$	Aerodynamic body moment lag $\boldsymbol{M}_l^q = [M_x^q, M_y^q, M_z^q]^T$

Symbol	Description
$\boldsymbol{n}_b$	Load factor vector
$\boldsymbol{\omega}$	Angular velocity vector $\boldsymbol{\omega} = [p, q, r]^T$
$\boldsymbol{P}$	Inertial position vector $\boldsymbol{P} = [X, Y, Z]^T$
$\boldsymbol{P}_l$	Aerodynamic generalized force lag $\boldsymbol{P}_l = [\boldsymbol{P}_l^1, \boldsymbol{P}_l^2, \ldots, \boldsymbol{P}_l^p]^T$
$\boldsymbol{P}_l^q$	Aerodynamic generalized force lag per pole $\boldsymbol{P}_l^q = [P_1^q, P_2^q, \ldots, P_m^q]^T$
p	Body roll rate
ϕ	Aircraft roll angle
ψ	Aircraft yaw angle
q	Body pitch rate
r	Body yaw rate
$\boldsymbol{\Theta}$	Euler angle vector $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$
θ	Aircraft pitch angle
U	Body forward velocity
$U_{g,0}$	Gust velocity input
$\dot{U}_{g,0}$	Gust acceleration input
$\boldsymbol{u}$	Input vector
$\boldsymbol{u}_c$	Control surface command vector $\boldsymbol{u}_c = [c_1, \ldots, c_c]^T$
$\boldsymbol{u}_t$	Thrust command vector $\boldsymbol{u}_t = [e_1, \ldots, e_o]^T$
V	Body lateral velocity
V_∞	Freestream velocity
$\boldsymbol{V}$	Translational velocity vector $\boldsymbol{V} = [U, V, W]^T$
W	Body vertical velocity
$\boldsymbol{w}$	Disturbance vector $\boldsymbol{w} = [U_{g,0}, \dot{U}_{g,0}]^T$
X	Aircraft longitudinal position
$\boldsymbol{x}$	State vector $\boldsymbol{x} = [\boldsymbol{P}, \boldsymbol{\Theta}, \boldsymbol{V}, \boldsymbol{\omega}, \boldsymbol{\eta}, \dot{\boldsymbol{\eta}}, \boldsymbol{\delta}, \dot{\boldsymbol{\delta}}, \boldsymbol{g}, \boldsymbol{F}_l, \boldsymbol{M}_l, \boldsymbol{P}_l, \boldsymbol{L}_l]^T$
Y	Aircraft lateral position
$\boldsymbol{y}$	Output vector $\boldsymbol{x} = [V_\infty, \alpha, \beta, \gamma, \boldsymbol{n}_b, \boldsymbol{P}, \boldsymbol{\Theta}, \boldsymbol{\omega}, \boldsymbol{L}_m, \boldsymbol{a}_t, \boldsymbol{a}_r]^T$
Z	Aircraft vertical position

Table 5: Structural modeling nomenclature

Symbol	Description
m	Aircraft mass
$\boldsymbol{J}$	Aircraft inertia matrix
$\boldsymbol{M}_{hh}$	Modal mass matrix
$\boldsymbol{C}_{hh}$	Modal damping matrix
$\boldsymbol{K}_{hh}$	Modal stiffness matrix
$\boldsymbol{M}_{ff}$	Free DOFs mass matrix
$\boldsymbol{C}_{ff}$	Free DOFs damping matrix
$\boldsymbol{K}_{ff}$	Free DOFs stiffness matrix
$\boldsymbol{M}_{mt}$	Mass matrix translational DOFs → monitoring stations
$\boldsymbol{M}_{mr}$	Mass matrix rotational DOFs → monitoring stations
$\boldsymbol{M}_{mh}$	Mass matrix generalized DOFs → monitoring stations

Table 6: Aerodynamic modeling nomenclature

Symbol	Description
$\boldsymbol{A}_0$	RFA quasi-steady matrix
$\boldsymbol{A}_1$	RFA unsteady matrix
$\boldsymbol{A}_{3+p}$	RFA lag matrices
$\boldsymbol{AIC}$	Aerodynamic influence matrix
$\boldsymbol{B}_F$	Aerodynamic body force lag decay matrix
$\boldsymbol{B}_L$	Aerodynamic body cut load decay matrix
$\boldsymbol{B}_M$	Aerodynamic body moment lag decay matrix
$\boldsymbol{B}_P$	Aerodynamic body generalized lag decay matrix
$\boldsymbol{C}_a$	Aerodynamic quasi-steady damping matrix
$\boldsymbol{C}_q$	Aerodynamic lag damping matrix
k	Reduced frequency
$\boldsymbol{K}_a$	Aerodynamic stiffness matrix
$\boldsymbol{K}_q$	Aerodynamic lag stiffness matrix
$\boldsymbol{l}$	Aerodynamic lag vector
$\boldsymbol{l}_q$	Aerodynamic lag vector per pole q
M	Mach number
$\boldsymbol{n}_i$	Aerodynamic panel normal vector
$\boldsymbol{N}_j$	Aerodynamic panel normal matrix
$\boldsymbol{P}_a$	Aerodynamic forces and moments, at l-grid, size $6n$
$\Delta\boldsymbol{c}_p$	Pressure coefficient increment vector
ρ	Air density
$\boldsymbol{S}$	Aerodynamic panel area matrix
$\boldsymbol{t}_i$	Aerodynamic panel tangent vector
$\boldsymbol{T}_j$	Aerodynamic panel tangent matrix
V_∞	Freestream velocity
$\boldsymbol{w}_j$	Panel downwash vector, at collocation points
$\boldsymbol{W}_{cd}$	Control surface deflection downwash matrix
$\boldsymbol{W}_{cm}$	Control surface motion downwash matrix
$\boldsymbol{W}_{ga}$	Gust velocity states downwash matrix
$\boldsymbol{W}_{gv}$	Gust acceleration states downwash matrix
$\boldsymbol{W}_{ro}$	Rotational body downwash matrix
$\boldsymbol{W}_{sd}$	Flexible displacement downwash matrix
$\boldsymbol{W}_{sm}$	Flexible motion downwash matrix
$\boldsymbol{W}_{tr}$	Translational body downwash matrix
$\boldsymbol{W}_{wa}$	Gust acceleration input downwash matrix
$\boldsymbol{W}_{wv}$	Gust velocity input downwash matrix
$\boldsymbol{W}_{ga}$	Gust velocity states downwash matrix
$\boldsymbol{W}_{gv}$	Gust acceleration states downwash matrix
$\boldsymbol{W}_{wv}$	Gust velocity input downwash matrix
$\boldsymbol{W}_{wa}$	Gust acceleration input downwash matrix
$\boldsymbol{\theta}_c$	Downwash due to camber and twist
$\boldsymbol{\beta}$	RFA poles, size p, index q

Table 7: Splining transformations nomenclature

Symbol	Description
$\boldsymbol{E}_F$	Selection matrix to extract forces from 6 DOF force vector
$\boldsymbol{E}_M$	Selection matrix to extract moments from 6 DOF force vector
$\boldsymbol{E}_\theta$	Selection matrix to extract induced rotations from 6 DOF spline
$\boldsymbol{E}_v$	Selection matrix to extract induced velocities from 6 DOF spline
$\boldsymbol{\Phi}_{be}$	Thrust grids → body frame
$\boldsymbol{\Phi}_{bl}$	Aerodynamic l-grid → body frame
$\boldsymbol{\Phi}_{he}$	Thrust grids → generalized coordinates
$\boldsymbol{\Phi}_{hl}$	Aerodynamic l-grid → generalized coordinates
$\boldsymbol{\Phi}_{jb}$	Body frame → aerodynamic j-grid spline
$\boldsymbol{\Phi}_{jc}$	Control surface deflections → aerodynamic j-grid spline
$\boldsymbol{\Phi}_{jh}$	Generalized coordinates → aerodynamic j-grid spline
$\boldsymbol{\Phi}_{ml}$	Aerodynamic l-grid → monitoring stations
$\boldsymbol{\Phi}_{sb}$	Body frame → accelerometer stations
$\boldsymbol{\Phi}_{sh}$	Generalized coordinates → accelerometer stations

Table 8: Gust modeling nomenclature

Symbol	Description
$\boldsymbol{A}_g$	Gust state matrix
$\boldsymbol{B}_g$	Gust input matrix
$\boldsymbol{C}_g$	Gust output matrix
$\boldsymbol{D}_g$	Gust feedthrough matrix
H	Gust gradient
$\boldsymbol{M}_g$	Gust zone mapping matrix
s	Laplace variable
τ	Time delay
U_{ds}	Gust design velocity
x_g	Gust longitudinal onset

Table 9: Accelerometer modeling nomenclature

Symbol	Description
$\boldsymbol{E}_R$	Selection matrix to extract rotational DOFs from 6 DOF spline
$\boldsymbol{E}_T$	Selection matrix to extract translational DOFs from 6 DOF spline
$\boldsymbol{R}$	Repeat operator
$\boldsymbol{r}_{bs}$	Distance sensors to CG
$\boldsymbol{S}(\boldsymbol{v})$	Stack operator

Table 10: Actuator modeling nomenclature

Symbol	Description
$\boldsymbol{D}_a$	Actuator damping matrix

Symbol	Description
$\boldsymbol{K}_a$	Actuator stiffness matrix
ω_c	Actuator angular frequency of actuator c
ξ_c	Actuator damping ratio of actuator c

List of Abbreviations

AFS Active Flutter Suppression. xvii, 2, 3, 23, 46, 62, 83, 95, 97, 101

AIC Aerodynamic Influence Coefficient. 8–10, 32–34, 74, 91–93, 95

ALA Active Load Alleviation. xvii, 2, 3, 46, 95

APM Airport Planning Manual. vii, 61, 63, 65

BDF Bulk Data File. 14, 59, 62, 63

CCD Control Co-Design. xvii, 97, 101

CFD Computational Fluid Dynamics. 7, 10, 12–15, 18, 91, 97

CG Center of Gravity. vii, 6, 61, 62, 64

DLM Doublet Lattice Method. 3, 8–10, 12, 14, 15, 18, 22, 25, 32, 74, 91, 92, 95, 97, 99

DOF Degree of Freedom. 7, 10–12, 18, 32, 57

EBW Embraer Benchmark Wing. xvii, 59

EOM Equations of Motion. 6, 7, 18, 92

FEM Finite-Element Model. xvii, 3, 10–15, 18, 59, 60, 62–65, 92, 95, 97

FSM Force Summation Method. 10, 11, 17, 43, 50

GAF Generalized Aerodynamic Forces. 10, 97

GLA Gust Load Alleviation. 2, 17, 23, 46, 62, 83, 93, 96, 97, 101

HALE High-Altitude Long Endurance. 10, 13, 92

HARW High Aspect Ratio Wing. 1

INDI Incremental Nonlinear Dynamic Inversion. 2, 14, 96

LE Leading Edge. 62, 65, 68, 69

LEMAC Leading Edge Mean Aerodynamic Chord. 67

LFD Linearized Frequency-Domain CFD. 10, 97, 101

LRA Loads Reference Axis. 71

LTI Linear Time-Invariant. 25

MAC Modal Assurance Criterion. v, 84

MAC Mean Aerodynamic Chord. 61, 67, 84

MDM Mode Displacement Method. 10, 11, 14, 17

MLA Maneuver Load Alleviation. 2, 23, 46, 62, 83, 96, 97, 101

MOW Minimum Operating Weight. 61, 62

MPC Model Predictive Control. 2

MST Modified Strip Theory. 7, 13

MTOW Maximum Take-Off Weight. 2

NASTRAN NASA Structural Analysis. xvii, 3, 12–18, 22, 25, 32, 57–59, 62–64, 71, 72, 95, 99–101

OEW Operating Empty Weight. 61

RANS Reynolds Averaged Navier-Stokes. 10

RFA Rational Function Approximation. vi, vii, xvii, 8, 9, 16, 18, 22, 26, 33, 73, 74, 80, 88, 89, 92, 97, 99, 100

ROM Reduced-Order Model. 10

TE Trailing Edge. 62, 65, 68, 69

UHARW Ultra High Aspect Ratio Wing. 1, 10, 16

UVLM Unsteady Vortex Lattice Method. 8–10, 13, 14

VLM Vortex Lattice Method. 3, 8, 9, 14, 18, 25, 32, 91, 92, 95, 97, 99

WRBM Wing Root Bending Moment. vi, 1, 26, 80, 85, 88, 94, 96

WRTM Wing Root Torsional Moment. 26, 80, 88, 94

ABSTRACT

The urgent need to further reduce fuel burn and climate impact for next-generation aircraft drives wings to become lighter and with higher aspect ratios to reduce induced drag. The resulting increase in wing flexibility presents both an opportunity and numerous challenges, as such wings experience increased structural loads at the wing root and are more prone to aeroelastic instabilities such as flutter. Passive and active flutter suppression and load alleviation techniques therefore provide a promising solution to enable these wings without the subsequent weight increase. For active flutter suppression and load alleviation, it is particularly important to express the free-flying aeroservoelastic model as an interpretable state-space model with manageable order.

In this regard, the aircraft loads model in the DLR Loads Kernel is formally derived and refined to make it suitable for free-flying aeroservoelasticity, constituting several improvements over the original model. The model is strictly expressed as a closed-form, symbolic, monolithic state-space representation, rigorously derived in this work. Besides improved interpretability and extensibility, this also allows for easily porting the nonlinear state-space model from Python to MATLAB/Simulink, and therefore effectively bridges NASTRAN and Simulink, the industry standards for finite-element modeling and control design respectively. To further improve suitability for free-flying aeroservoelasticity, the spiral nature of the Sears function is approximated using cascaded gust zones using a Padé approximation, significantly reducing the number of disturbance inputs. Physical Rational Function Approximation (RFA) is employed, with the resulting aerodynamic lag dynamics projected to lag force and moment dynamics, achieving a substantial reduction in lag states. Additionally, the model is augmented with actuator dynamics and an accelerometer sensor model.

As the state-space model requires a representative aircraft, the Embraer Benchmark Wing (EBW), used for research on aeroelastic tailoring, is systematically transformed into a free-flying Finite-Element Model (FEM), enabling the generation of free-flying aeroservoelastic state-space models with different wing mass and stiffness distributions at various mass cases and in varying flight conditions. Using this aircraft, the derived model achieves a 100–400x improvement in simulation time and a 458x reduction in input file size for the considered cases, and thus eliminate the need to save model output. Gust inputs are reduced from n to 2, and lag states from $n \times p$ to $(6 + m + s) \times p$, with n aerodynamic panels, p RFA poles, m flexible modes, and s monitoring forces and/or moments. At the same time, these refinements are shown to not compromise model behaviour, as results show excellent agreement. These improvements, in conjunction with the added actuator dynamics and sensor model, make the state-space model well-suited for future work in free-flying Active Flutter Suppression (AFS), Active Load Alleviation (ALA), and on the longer-term, Control Co-Design (CCD).

CONTENTS

1

INTRODUCTION

With the advance of global warming and the continuing growth of the energy-intensive and hard-to-abate aviation industry, all technological, albeit incremental, improvements in the design of next-generation commercial aircraft can meaningfully reduce the climate impact of aviation [1], and therefore aerospace research has focused on reducing the energy consumption of next-generation aircraft. An obvious, but often nontrivial, approach to reducing the fuel burn of aircraft is to reduce drag. Aircraft drag consists of parasitic drag, due to viscous effects, and of induced drag, due to lift generation, which can be up to 40% in cruise, and even higher in climb [2]. The induced drag in Equation 1.1 of an aircraft can be reduced by means of increasing the aspect ratio AR of the wings. Most often, this increase in aspect ratio constitutes increasing the wingspan. The increase in aspect ratio has brought about High Aspect Ratio Wing (HARW) aircraft, with ARs roughly in the range 11-18, and Ultra High Aspect Ratio Wing (UHARW) aircraft, with ARs at roughly 18+.

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

$$R = \frac{V}{g_0}\frac{1}{SFC}\left(\frac{C_L}{C_D}\right)\ln\left(\frac{W_I}{W_F}\right) \tag{1.2}$$

However, increasing the aspect ratio causes the wing structural mass to rise rapidly, if no further design measures are taken [3]. As the wings become more slender, the Wing Root Bending Moment (WRBM) correspondingly increases due to stronger aeroelastic coupling in gusts and maneuvers. Additionally, slender wings are more prone to aeroelastic instabilities such as flutter. Consequently, the aircraft must be designed to withstand the higher limit and ultimate loads and while not exceeding flutter speed requirements [4], requiring a strengthened airframe, which is often heavier. Therefore, there is a certain wingspan increase after which the drag reduction benefits are offset by the heavier structure. The Breguet range equation in Equation 1.2 illustrates this trade-off: an increase in wingspan improves aerodynamic efficiency, but the subsequently heavier structure reduces the weight fraction.

Whereas aeroelasticity considers the interaction between aerodynamics, inertia, and structures [5], flight dynamics concerns itself with the motion, stability, and control of aircraft [6]. Besides the trade-off in wing slenderness and weight, wing flexibility introduces coupling between the disciplines of aeroelasticity and flight dynamics, as the frequency separation between the flight dynamic and aeroelastic modes narrows [7]. Moreover, this can additionally affect handling qualities and ride quality. Consequently, aircraft exhibiting significant structural flexibility require treatment of aeroelasticity and flight dynamics in a fully coupled manner in design [8]. This convergence has led to the emergence of the field known as free-flying aeroelasticity, or flexible aircraft flight dynamics, which concerns itself with the analysis of the open-loop characteristics of flexible aircraft. Due to the aforementioned considerations, embodying wing flexibility into the aircraft therefore presents both a challenge and an opportunity in the design process.

Research has therefore focused on enabling high aspect ratio aircraft by managing the aeroelastic coupling and instabilities without the subsequent weight increase. For HARW, this can be achieved through both active and passive methods. For UHARW, often a strut is additionally required to further support the wing. Passive methods often focus on tailoring the wing mass and stiffness distribution, known as aeroelastic tailoring [9], to increase the flutter speed, and/or to alleviate wing loads, but concerns any technique to suppress flutter

and loads that does not require a control system. In active methods, on the contrary, control effectors are actuatued to increase the flutter speed, known as Active Flutter Suppression (AFS), and/or to alleviate wing loads, known as Active Load Alleviation (ALA). The load alleviation problem is typically divided into Maneuver Load Alleviation (MLA), which focuses on shifting the lift distribution more inboard to achieve the same lift, and Gust Load Alleviation (GLA), which focuses on minimizing the peak loads with minimal perturbation of the aircraft trajectory [10]. Figure 1.1 shows the wing lift distribution of an aircraft with and without load alleviation.

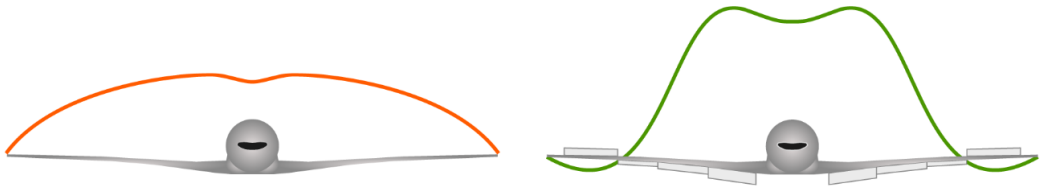


Figure 1.1: Aircraft without and with load alleviation. Taken from Xu and Kroo [11].

The challenges and subsequent solutions related to wing flexibility of aircraft are not exclusive to new generations of aircraft, but become even more pressing as aspect ratios are pushed higher and higher. Engineers discovered late in the design of the Boeing XB-47, the prototype that paved the way for today's commercial transonic aircraft, that the aircraft might be unstable in pitch due to the flexibility of the swept-back wings. However, it was subsequently discovered that this effect was negated by the change in horizontal tail incidence angle because the flexibility of the fuselage [8]. The Voyager, the first aircraft to fly around the world nonstop, suffered from body-freedom flutter, where the rigid-body motion affected wing bending and vice versa [8]. The in-flight disintegration of the AeroVironment Helios was caused by turbulence that caused excessive wing dihedral, which lead to divergent pitch oscillations, which in turn eventually caused critical high dynamic pressure on the vehicle leading to structural failure [12]. Although these examples have been cherrypicked, it is certain that innumerable additional real-world instances of coupled aeroelastic and flight dynamic issues exist, given the widespread presence of wing flexibility in both past and present aircraft, such as the B787, B777, and E2. Moreover, active load alleviation has also been applied on both past and present aircraft. Regan and Jutte [13] provide an overview of aircraft featuring active load control for gust alleviation, which include the Lockheed C-5A, Lockheed L-1011-500 TriStar, B-1, B-2, A320, A330, A340, A380, and B787. Interesting notes include that the wingspan of the TriStar could be increased by 5.8% and the drag subsequently reduced by 3% due this system, and that the Maximum Take-Off Weight (MTOW) of the A320 could be increased by 1.3% on currently operational aircraft due to the implemented load alleviation. The report also indicates these systems had varying objectives, such as fatigue life extension for the C-5A, ride quality for the B-1 and B-2, and load alleviation and ride quality for most of the modern commercial aircraft. Xu and Kroo [11] notes that aircraft with MLA can have up to 15% higher wingspan when keeping the weight fixed, which can lead to a 8-10% reduction in drag. Besides span extensions, the authors note that ALA could also enable laminar flow if thinner profiles can be used because of the technology.

As active control methods inherently require control design, it is desirable to have the free-flying aeroelastic model expressed as a state-space model, as this is preferred or required for many modern control methods, such as LQR [14], H_∞ [15], μ-Synthesis [16], MPC [17] and INDI [18]. When free-flying aeroelastic models are studied in closed-loop systems, and with sensing and actuator systems involved, the field is referred to as free-flying aeroservoelasticity. At the heart of a state-space model is the state equation of Equation 1.3, which expresses the free-flying aeroservoelastic dynamics as a first-order differential equation as a function of the state vector, input vector, and disturbance vector. Whereas the state equation models the system dynamics, the output equation of Equation 1.4 models the measurement process, which is inherently required in control feedback systems, such as in AFS and ALA. As such, free-flying aeroservoelastic state-space models form the cornerstone for the modeling and analysis of coupled aeroelasticity and flight dynamics, as well as passive and active flutter suppression and load alleviation techniques, and therefore constitutes the research field of this thesis.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{1.3}$$

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{1.4}$$

This report is structured as follows. First, Chapter 2 reviews the contemporary modeling approaches, examines existing tools and frameworks, highlights identified field-level challenges, and identifies a reference framework, DLR Loads Kernel [19], [20], in which limitations are defined with respect to aeroservoelasticity, and formulates thesis objectives and research questions. The overarching thesis objective is therefore to build a pipeline for generating free-flying aeroservoelastic state-space models given an aircraft geometry, mass distribution, structural model, and aerodynamic model, given the set aeroservoelastic model requirements in Chapter 2.

Subsequently, Chapter 3 aims to formulate a closed-form, symbolic, monolithic state-space model describing free-flying aeroservoelasticity. The chapter is set up top-down, starting with the final model and then gradually deriving the building blocks in order to pierce together the model. This model is expressed using nonlinear flight dynamics using mean-body axes, linear panel aerodynamics using VLM/DLM, and an industrial linear FEM from NASTRAN. Special attention is paid to improving the computational efficiency of the model, and keeping the order of the state-space model manageable. To this end, gust penetration is modeled using cascaded gust zones with a Padé approximation, and the aerodynamic lag states are reduced through projection to rigid-body force and moment lag, generalized force lag, and monitoring station load lag. The model is also augmented with actuator dynamics and an accelerometer sensor model.

Whereas the first part of this thesis addresses the theoretical modeling aspects of free-flying aeroservoelasticity, the matrices and parameters that serve as input to this framework are critically required, as the state-space framework is useless without a representative aircraft to be fed into it. This process, however, requires a more practical framework. Therefore, Chapter 4 devises an end-to-end modeling pipeline for creating a suitable commercial aircraft that can be used in the state-space model. In order to achieve this, the Embraer Benchmark Wing, a clamped-wing finite-element model that has been used for research on aeroelastic tailoring in past research [21], [22], is systematically transformed to a representative free-flying finite-element model. This pipeline subsequently enables the generation of free-flying aeroservoelastic state-space models for various tailored wings, mass cases, and flight conditions, which can be used for the design of AFS and ALA on aircraft with aeroelastically tailored wings.

The results of the model constitute the model verification, which is discussed in Chapter 5. In this chapter, the computational efficiency of the model is verified, the coupled aeroelastic and flight dynamic behaviour as well as gust and maneuver responses are compared with the DLR Loads Kernel reference framework, and the lag state order reduction is verified. Chapter 6 subsequently discusses the adopted modeling approaches, assumptions and limitations, as well as addressing applications and future work. Lastly, Chapter 7 draws conclusions regarding this work.

2

Literature Review

This chapter presents an overview of the recent work performed in the field of free-flying aeroservoelastic state-space modeling. Section 2.1 reviews the state-of-the-art in free-flying aeroservoelastic state-space modeling by addressing what the contemporary modeling approaches are, and what existing tools and frameworks have been built for similar objectives. Subsequently, Section 2.2 identifies practical as well as methodological field-level challenges. However, the specific challenges and gaps that are addressed in this thesis work are a small, specific subset of these, and are introduced in Section 2.3. An overall thesis objective is formulated in Section 2.4, based on these research gaps. Lastly, based on the thesis objectives, research questions are formulated that are to be addressed in order to reach this objective. The considered literature is recent, with the considered literature stemming mostly from the past 15 years, excluding seminal papers published before that.

2.1. State-of-the-Art Review

The review of the state-of-the-art in this section follows a two-pronged approach. First, Subsection 2.1.1 presents the multitude of possible modeling approaches in free-flying aeroservoelasticity. Secondly, Subsection 2.1.2 presents an overview of the existing tools and frameworks with similar objectives as this work.

2.1.1. Modeling Approaches

Modeling fundamentally requires simplifying the true characteristics of what is modeled. However, the degree to which the aeroservoelastic system is simplified can be controlled, most often at the cost of computational resources and model complexity [23]. In free-flying aeroservoelastic modeling, the modeling choices that control the fidelity of the model can be divided into four major modeling choices: equations of motion, aerodynamic model, structural model, and aerostructural coupling. While control methods and actuator dynamics are also important for closed-loop free-flying aeroelasticity, this review and thesis work focus on the fundamental modeling framework, leaving controller synthesis and actuator specification as future work, depending on the desired application. Additionally, a small section is dedicated to the state-space modeling aspects. The modeling approaches discussed here serve as an overview, but not a comprehensive analysis of each model, as each subject can be dedicated multiple literature reviews to.

Equations of Motion

At the heart of free-flying models are the equations of motion. The flight dynamic equations of motion consist of six equations: three describing linear, or translational, dynamics and three describing angular, or rotational, dynamics. In contrast, the aeroelastic equations of motion include as many equations as there are structural degrees of freedom in the aeroelastic model. In free-flying aeroelastic models, these flight dynamic equations and aeroelastic equations are aerodynamically coupled, and may additionally be inertially coupled, resulting in a model that captures both rigid-body and elastic dynamics, as well as their interactions.

Aerodynamic coupling arises from the aeroelastic feedback loop in which aerodynamic loads induce structural deformation, and the resulting deformation alters the aerodynamic loads. Inertial coupling occurs as

the structural flexibility of the airframe changes the mass distribution of the structure, leading to a time-varying CG position, which causes additional inertial coupling terms in the equations of motion.

To understand when the EOMs are aerodynamically-coupled, or also inertially-coupled, one must first understand the two reference frames that are most commonly used to describe the EOMs in: fixed-body axes and mean-body axes. Waszak and Schmidt [7] detailed the derivation of EOMs based on the Euler-Lagrange equations and applied the mean-axes constraint to arrive at EOMs defined in mean-body axes, distinguishing between exact and practical mean-axes constraints. Exact mean-axes constraints constrain the axes such that the net linear and net angular momenta due to elastic displacements are zero, which drops out additional inertial coupling terms in the kinetic and potential energy expressions, simplifying the EOMs, but require the axes to be both moving and rotating as the aircraft deforms. The exact mean-axes constraints are as follows with $\boldsymbol{p}$ the position vector, $\boldsymbol{d}$ the elastic displacement vector, which are defined continuously. The first constrains the reference frame to the instantaneous CG to not produce net linear momentum, and the second constrains the orientation of the reference frame to not produce net angular momentum.

$$\int_V \rho \dot{\boldsymbol{d}}\, dV = 0 \quad \int_V \rho (\boldsymbol{p} \times \dot{\boldsymbol{d}})\, dV = 0 \tag{2.1}$$

These exact mean-axes constraints are hard to apply as one needs to define the EOMs in a continuously moving and rotating reference frame so as not to produce net linear and angular momenta due to elastic displacement. Therefore, these mean-axes constraints are often linearized instead, resulting in the practical mean-axes constraints. As the constraints are linearized, the deformations are assumed small, which implies that the instantaneous CG position is equal to the undeformed CG position and that the orientation of the axes is fixed, as the elastic displacement does not contribute to net angular momentum, as can be observed from the equation below, with $\boldsymbol{s}$ the undeformed position vector.

$$\int_V \rho \dot{\boldsymbol{d}}\, dV = 0 \quad \int_V \rho (\boldsymbol{s} \times \dot{\boldsymbol{d}})\, dV = 0 \tag{2.2}$$

The resulting EOMs take the following form in case of practical mean-axes and the elastic displacements represented using vibrational modes. It is important to note that the inertia tensor also changes with structural deformation, but this is often ignored in practical mean-axes.

$$m\dot{\boldsymbol{V}} = -m\boldsymbol{\omega} \times \boldsymbol{V} + \boldsymbol{F}_b \tag{2.3}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{M}_b \tag{2.4}$$

$$\boldsymbol{M}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}\dot{\boldsymbol{\eta}} + \boldsymbol{K}\boldsymbol{\eta} = \boldsymbol{\Phi}_{hf}\boldsymbol{F}_f \tag{2.5}$$

On the contrary, fixed-body axes are attached to a fixed point in the undeformed aircraft, which brings about additional inertial coupling terms in the EOMs. The derivation of EOMs using fixed-body axes has been detailed extensively by Meirovitch and Tuzcu [24], who heavily criticized the mean-body axes approach. It is important to note, however, that any choice of reference frame should result in exactly the same motion of the vehicle, as rightly noted by Neto, Silva, Paglione, *et al.* [25], who formulated the EOMs for flexible aircraft flight dynamics in terms of general body axes, making no assumptions about the position and orientation of the reference frame during the derivation, and showed their equivalence in terms of motion, as long as the implementation is consistent. Meirovitch and Tuzcu argued that preventing additional coupling in the EOMs is pointless as the equations are already aerodynamically coupled, that expressing forces and moments along the time-varying mean axes is tedious, and that invoking the use of practical mean-axes without enforcing the constraints raises questions on the validity of the results. These have been refuted by Neto, Silva, Paglione, *et al.*, who points that when free-free vibration modes are used, the elastic modes are mass-orthogonal to the rigid-body modes, meaning that the elastic deformation does not generate net linear momentum. As a result, the instantaneous CG coincides with the undeformed CG to first order, and the practical mean-axes constraints are automatically satisfied. Therefore, it does simplify the equations significantly, it is not tedious to express them in these axes, and it is in fact valid.

The previously discussed derivations [7], [24] are not tailored towards finite-element models, with lumped masses, yet structural models in the industry most often come in the form of such finite-element models.

Reschke [26] derived the EOMs using practical mean-axes for finite-element models with lumped masses, including nodal rotational DOFs and offset inertia terms, which introduce residual inertial coupling not eliminated by the practical mean-axes. Reschke highlighted the need to include these residual terms in case of high angular rates and high accelerations. It has been a debate in literature whether EOMs should model inertial coupling in addition to aerodynamic coupling. Saltari, Riso, Matteis, *et al.* [27] derived an aerodynamically- and inertially-coupled set of EOMs based on practical mean-axes, and showed that aerodynamic coupling is the primary effect, and that residual inertial coupling is negligible for their considered models and assumptions, even during highly-loaded maneuvers. Therefore, as long as small deformations can be appropriately assumed and free-free vibrational modes are used, use of the practical mean-axes constraint is valid and the resulting EOMs are easier to work with. This is also reflected by the amount of work in recent literature using mean-body axes [10], [18], [19], [28]–[40], as opposed to fixed-body axes [41]–[46].

AERODYNAMIC MODELING

The focus of this thesis is on next-generation commercial aircraft, which operate at most in the high-subsonic and transonic regime, and thus only aerodynamic models up to this regime are discussed. Aerodynamic models most often used for free-flying aeroelasticity can be broadly divided into three domains: aerodynamic strip theory, also known as lifting-line methods, panel aerodynamics, and CFD-based models [47]. An important requirement on the aerodynamic model when analyzing aircraft in in flutter and gusts, is to model unsteady aerodynamics. Aerodynamic phase lag is critical in predicting flutter onset, as the wing oscillates at high frequencies, and the aerodynamics do not adjust instantly to these fast oscillations. Similarly, for the same reason, unsteady aerodynamics is also important for gust loads. For maneuver loads, it is less important, unless the maneuvers are extremely quick.

The simplest unsteady aerodynamics models are based on aerodynamic strip theory. These are also computationally the least resourceful models [34]. The lifting surfaces are divided into two-dimensional strips that describe the local aerodynamics and are integrated along the span to obtain the aerodynamic forces. Representative 2D aerodynamic strip models Theodorsen's theory [48], the seminal theory on two-dimensional unsteady aerodynamics, Peters' finite-state theory [49], which yields a compact state-space representation convenient for time-domain aeroelastic simulation, and Modified Strip Theory (MST), which was developed to account for higher angles of attack as well as stall [50]. Most studies in flexible aircraft flight dynamics employ some sort of strip theory [28], [32], [41], [43]–[45], [51]–[57]. An advantage of this method is that it can easily be corrected with empirical corrections, such as stall models and viscous drag [58]. On the other hand, a major disadvantage of strip theory is the lack of spanwise and chordwise variations in lift, which may be significant even in high aspect ratio aircraft [58]. In literature, there seems to be the trend that strip theory is more popular when applied in conjunction with geometric nonlinear structural models or when used in aeroservoelastic studies. The former is because large deformations are more easily accounted for with strip theory as aerodynamic mesh deformation is less involved. The latter is most likely due to the smaller computational time required to construct an aerodynamic model based on strip theory.

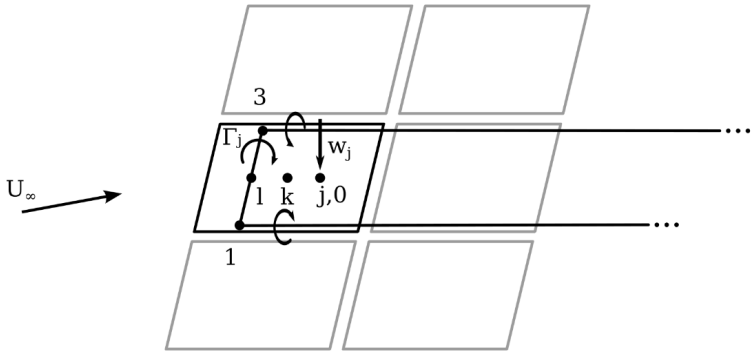


Figure 2.1: Aerodynamic panel, horseshoe vortex and nomenclature. Taken from Voß [19].

An upgrade in aerodynamics fidelity can be obtained when using panel aerodynamics, as these account for both spanwise and chordwise lift variation. In panel aerodynamics, the aircraft is discretized into aerody-

namic panels. Each of the aerodynamic panels has an aerodynamic potential flow element associated with it of unknown strength. Based on aerodynamic boundary conditions, this strength is solved for and the lift distribution can be obtained. This aerodynamic boundary condition entails the flow-tangency condition. These boundary conditions arise from both any type of motion and any type of displacement affecting the aerodynamics, and are typically computed using splining matrices, which are discussed later in this review. At the heart of panel aerodynamics lies the aerodynamic panels themselves. Figure 2.1 shows an aerodynamic panel, the associated horseshoe vortex, used in VLM, and the symbols used. The aerodynamic quarter-chord points are designated with subscript l, the half-chord points are designated with subscript l, and the three quarter-chord points, also known as the collocation points, are designated with subscript j. The aerodynamic force acts perpendicular to the aerodynamic panel. The downwash acting on the panel is what produces the aerodynamic force. It is important to note that this downwash is in fact a normalwash on the panel, and thus acts normal to it.

The Vortex Lattice Method (VLM) uses vortex elements and is often used for quasi-steady aerodynamics. However, it is thus less suitable for unsteady aerodynamics, as aerodynamic lag is not modeled. The outcome of the VLM is the Aerodynamic Influence Coefficient (AIC) matrix that relates the pressure coefficient differential over the quarter-chord point l of each aerodynamic panel to the downwash induced at the collocation point j of each aerodynamic panel, as indicated by Equation 2.6. This AIC matrix is constructed based on evaluating the Biot-Savart law for each pair of aerodynamic panels by computing the aerodynamic influence of the horseshoe vortex of one panel at the collocation point of the other panel, which results in an $n \times n$ matrix with n the number of aerodynamic panels. The exact implementation details of VLM are not covered in this review, but Katz and Plotkin [59] gives a comprehensive treatment of the relevant theory. Additionally, Voß [20] provides a more practical treatment and a publicly available implementation.

$$\Delta \boldsymbol{c}_{p_l} = \boldsymbol{AIC}\boldsymbol{w}_j \tag{2.6}$$

The desire to model unsteady aerodynamics using aerodynamic panels led to the development of the Doublet Lattice Method (DLM) by Albano and Rodden [60] and the Unsteady Vortex Lattice Method (UVLM) [59]. For DLM, the same aerodynamic panel discretization is used as for VLM. Similar to VLM, an AIC matrix is constructed that relates the pressure coefficient differential to the downwash on the panel. Whereas VLM uses the Biot-Savart law, DLM evaluates the downwash induced at the receiving point due to an oscillating pressure doublet at the sending point using so-called kernel functions, which are derived from linearized unsteady potential flow theory, and depend on the panel geometry, Mach number, and reduced frequency [61]. This kernel function is evaluated for each pair panels, and thus the AIC matrix from DLM also has size $n \times n$. The outcome of the DLM is similar to the VLM, being an AIC matrix, with a major difference, however, that the AIC matrix from DLM is valid only for the reduced frequency and Mach number at which it was evaluated. Another distinction is that the AIC matrix from DLM is complex-valued, which model the oscillatory motion the model aims to capture. In order to describe the unsteady aerodynamics at multiple reduced frequencies, AIC matrices are computed for the set of reduced frequencies that are of interest, resulting in a formulation in the discrete frequency-domain. The pressure coefficient differential is related to the panel downwash in a similar fashion as for the steady AIC matrix, shown in Equation 2.7. The theoretical background of DLM is highly mathematical and out of scope for this review and work, but Demasi [61] provides a mathematical, yet accessible, and probably the best explanation of the model.

$$\Delta \boldsymbol{c}_{p_l} = \boldsymbol{AIC}(k)\boldsymbol{w}_j \tag{2.7}$$

DLM uses oscillating doublets as opposed to vortex elements and the wake is therefore effectively modeled as flat. Additionally, as the AIC matrix is only valid for the reduced frequency at which it was evaluated, DLM is expressed in the discrete frequency-domain, and therefore, in order to express it in continuous time-domain, which is required for state-space modeling of free-flying aeroservoelasticity, it needs to be converted, which is done using a Rational Function Approximation (RFA) [62], most often with Roger's approximation [63], which has the form of Equation 2.8. Once an RFA model is obtained it can be easily converted to the continuous time-domain using an inverse Laplace transform. It must be noted, however, that the resulting RFA aerodynamic model is an approximation, due to the least-squares fit used to fit the RFA.

$$\boldsymbol{AIC}(k) = \boldsymbol{A}_0 + jk\boldsymbol{A}_1 - k^2\boldsymbol{A}_2 + \sum_{q=1}^{p} \boldsymbol{A}_{q+2}\frac{jk}{jk+\beta_q} \tag{2.8}$$

Several RFA variants are used: physical RFA, generalized RFA, and half-generalized RFA. In physical RFA, the RFA is performed on the AIC matrices that map the downwash at each panel to the strength of the potential element [39], hence the name physical. The upside is that the aerodynamics can be exactly resolved, and the steady RFA matrix can be substituted with the more accurate AIC from VLM, but the downside is the huge number of aerodynamic lag states. In generalized RFA, the RFA is performed on AIC matrices which are first expressed in modal coordinates. Hence, the aerodynamic forces are expressed in the generalized coordinates, and the aerodynamic forces cannot be reconstructed exactly, as they are expressed using the truncated modal basis. Another downside is that the aerodynamic forces are mass-dependent, as they are expressed in generalized coordinates, and the steady VLM AIC cannot be used. In half-generalized RFA [34], the RFA is performed on AIC matrices which map the generalized coordinates to nodal displacements and rotations. Compared to the generalized approach, this allows recovering the nodal loads. It is interesting that there is a discussion on the right RFA approach, as seemingly the physical RFA approach can be transformed to generalized or half-generalized representations, also after the RFA, which provides much more flexibility. It could be though that doing these transformations before the RFA is computationally less resourceful.

Related to the RFA and aerodynamic modeling, is the modeling of gusts entering the aeroservoelastic system. Gusts affect the aircraft differently from other perturbations, as the gust travels downstream the aircraft, known as the gust penetration effect [6]. This gust penetration effect brings about the spiral nature of the Sears function. RFAs as-is struggle to capture this spiral nature as RFAs cannot capture time delays. Several workarounds to this issue have been proposed. First, the aircraft can be divided into gust zones [64]. Additionally, the delays can be approximated using a Padé approximation [10], [65]. Recently, Quero, Vuillemin, and Poussot-Vassal [66] formulated a different approach, doing away with the RFA completely, and using tangential interpolation instead, which is a type of interpolation along certain directions, which makes sense in the case of the spiraling nature of the Sears function. The method achieves this through the so-called Loewner framework, which is a data-driven model order reduction technique that outputs a state-space model based on tangential interpolation constraints. The method has not been further deeply scrutinized in this review, besides its high-level working principles, although it is rather promising technique for modeling gusts.

Although formulated using potential flow theory, compressible effects can be accounted for with DLM, using the Prandtl-Glauert correction. DLM remains the contemporary aerodynamic model for unsteady compressible aerodynamics in aircraft flutter and loads analysis as it has proven to be reliable for aircraft gust and maneuver loads certification [4], [67]. To build the aerodynamic influences in DLM, numerical approximation is used as evaluate the kernel functions as these cannot be integrated analytically, and therefore the AIC matrices themselves are an approximation. It is for this reason that the steady aerodynamic influence matrix of DLM is often replaced with that from VLM to improve accuracy [19], [20], [39], [68]. DLM is often used with linear boundary conditions, which Van Zyl and Mathews [69] enhanced with nonlinear boundary conditions to model sideslip/dihedral and yaw/dihedral coupling, calling it enhanced DLM. Neto [30] applied a similar approach for modeling nonlinear boundary conditions, by updating the panel normals at every timestep in the simulation. The issue at the root here is that the aerodynamic mesh used in DLM is often 3D but constructed using horizontal and vertical 2D panels, which struggle to capture the true pressure distribution as the pressure forces always act perpendicular to the panels. For flight dynamics, this can be problematic as flight dynamic effects such as yaw-roll coupling may not be appropriately captured. Kier and Looye [39] proposed using the vectorial Kutta-Jouwkowski law to capture yaw-roll coupling, or to explicitly add induced drag based on Trefftz plane analysis. Further work by Kier [36], [38], [40] focused on building a 3D panel method to further alleviate this problem. Another issue with the planar representation, is that the fuselage aerodynamics are not resolved properly. To this end, recently, Dimitriadis, Kilimtzidis, Kostopoulos, *et al.* [70] created an unsteady source and doublet panel method, which can model the fuselage using source elements, and can resolve more complex 3D geometries, and is publicly available. Recent publications in free-flying aeroelasticity using DLM include [10], [29], [31], [33], [68], [71]–[74].

UVLM uses vortex elements, but models the wake explicitly in order to capture unsteady aerodynamics, compared with VLM. This wake is modeled using an additional set of aerodynamic panels. The wake is also allowed to deform in UVLM. An advantage of UVLM over DLM is that it is formulated in the discrete time-domain. A formulation in the continuous time-domain was derived by Werter, De Breuker, and Abdalla [75], which makes this implementation extremely well suited for state-space modeling of free-flying aircraft for loads analysis or control design. UVLM has been shown to be a good alternative for flexible aircraft flight

dynamics when the flat wake assumption does not hold, for example, in large and complex wing deformations [58]. UVLM is gaining ground as the aerodynamic model in free-flying aeroelasticity, as indicated by the number of recent work using UVLM [25], [30], [42], [47], [76]–[78]. A limitation of models based on panel aerodynamics is their fundamental assumptions of inviscid and incompressible flow, which make them incapable of modeling viscous drag, shocks and stall.

Lastly, CFD-based models can be used, although this comes at an insurmountable computational cost and is therefore not applied when analyzing flexible aircraft in control feedback systems, especially in research and preliminary design. CFD-based models are used, however, for high-fidelity aircraft loads modeling, especially at later design stages [79]. Reimer, Ritter, Heinrich, *et al.* [80] compared DLM and CFD results on a free-flying aircraft in a discrete gust and concluded that DLM overestimates the gust loads quite significantly in the transonic regime, but more noticeably also slightly in the subsonic regime. An alternative is the development of Reduced-Order Model (ROM)-based CFD models, or the AIC matrices can be corrected using CFD data to improve accuracy [81]. Additionally, the linear AIC matrices can also be replaced by computing the Generalized Aerodynamic Forces (GAF) using Linearized Frequency-Domain CFD (LFD) [82]–[85], which is an emerging, higher fidelity aerodynamic modeling technique for loads analysis. In LFD, a Navier-Stokes RANS CFD solver is used to compute a steady RANS flow around an aircraft, and subsequently, a linearized CFD solver is used to compute the response due to small harmonic perturbations around this steady base flow, which is then used to compute the GAF.

STRUCTURAL MODELING

Two distinctions in structural models are often made in free-flying aeroelastic modeling. First, structural models can be linear or nonlinear. Nonlinearities in the structural model can be due to material, contact, and geometric nonlinearity. For Ultra High Aspect Ratio Wing (UHARW) and High-Altitude Long Endurance (HALE) aircraft, geometric nonlinearities dominate, as the large wing deformations significantly alter both the structural stiffness and mass distribution. Simply said, geometric nonlinearities arise due to tip shortening effects. Secondly, another major distinction is that of beam models versus finite-element models.

Linear beam models can hardly be described as state-of-the-art, and are thus not further discussed in this review. Linear FEM is most commonly used for free-flying aeroelasticity, particularly for conventional transport aircraft when small deformations can be assumed. Essentially, a finite-element model is a large mass-spring-damper system. When using linear FEMs, the structural dynamics are typically expressed using free-free vibrational modes using modal analysis, with the resulting equation shown in Equation 2.9. This improves interpretability, reduces the model order, and reduces computational cost when working with the aeroelastic equation. Modal analysis is discussed in this thesis in Section 3.4. Often, the number of modes is truncated to further reduce the model order, and care must be taken to truncate it to roughly match the bandwidth of the excitation.

$$\boldsymbol{M}_{ff}\ddot{\boldsymbol{x}}_f + \boldsymbol{C}_{ff}\dot{\boldsymbol{x}}_f + \boldsymbol{K}_{ff}\boldsymbol{x}_f = \boldsymbol{P}_f \tag{2.9}$$

Alternatively, a Guyan reduction [86], also known as static condensation, can be applied, which retains active DOFs to which loads are applied and reduces the system into a condensed mass and stiffness matrix by taking into account the effect of the slave DOFs. Compared to modal analysis, a Guyan reduction is computationally less expensive and retains physical DOFs, but neglects inertia and is therefore less well suited to higher frequency dynamic analysis, such as in free-flying aeroelasticity. The partitioning of the stiffness matrix is shown in Equation 2.10, and the reduced stiffness matrix in Equation 2.11.

$$\begin{bmatrix} \boldsymbol{K}_{aa} & \boldsymbol{K}_{ao} \\ \boldsymbol{K}_{oa} & \boldsymbol{K}_{oo} \end{bmatrix} \begin{pmatrix} \boldsymbol{u}_a \\ \boldsymbol{u}_o \end{pmatrix} = \begin{pmatrix} \boldsymbol{p}_a \\ \boldsymbol{p}_o \end{pmatrix} \tag{2.10}$$

$$\boldsymbol{K}_{aa,r} = \boldsymbol{K}_{aa} - \boldsymbol{K}_{ao}\boldsymbol{K}_{oo}^{-1}\boldsymbol{K}_{oa} \tag{2.11}$$

Another important aspect in aircraft loads analysis is that of structural loads recovery. As the finite-element model essentially describes the nodal displacements or element strains, the structural loads, or cutting forces and moments, on the finite-element need to be recovered. Typically, either the Force Summation Method (FSM) or Mode Displacement Method (MDM) is used. The MDM directly uses the truncated modal basis to recover the nodal lies, which is exactly where its main issue lies.

$$\boldsymbol{P}_g = \boldsymbol{K}_{gg}\boldsymbol{\Phi}_{gf}\boldsymbol{u}_f \tag{2.12}$$

The FSM on the other than reconstructs the nodal loads by subtracting the inertial nodal loads from the external loads applied at the nodes. The external loads at the nodes are expressed by summing the force contributions due to each state and input, hence the summation name.

$$\boldsymbol{P}_g = \boldsymbol{P}_g^{ext} - \boldsymbol{M}_{gg}\left[\boldsymbol{\Phi}_{gb}\ddot{\boldsymbol{u}}_b + \boldsymbol{\Phi}_{gf}\ddot{\boldsymbol{u}}_b f\right] \tag{2.13}$$

As MDM uses a truncated modal basis for the recovery of the loads, whereas FSM only uses the truncated basis for part of the inertial loads, and for the external loads which directly depend on the modal displacements and velocities. Hence, when using a truncated modal basis FSM converges better than MDM [39]. Also, FSM allows discrening between external and inertial loads [39].

Nonlinear FEM in free-flying aeroservoelasticity for conventional transport aircraft seems to be relatively unexplored, while geometrically nonlinear beam models are well established for very flexible aircraft [41], [51], [87]–[90]. Recently, Cea and Palacios [91] built FENIAX, a free-flying framework that augments linear FEM with geometric nonlinear effects for loads analysis, allowing to take into account these effects in loads and aeroelastic analysis. Gray and Martins [92] compared the effect of geometric linear versus geometric nonlinear FEM, noting an increase of 5-10% in bending stress for the highly flexible uCRM 13-5. Additionally, the difference in lift and drag is on the order of 1-2%, indicating that the effect of geometric nonlinearities on the aggregated aerodynamics are second-order effects, even in highly flexible wings.

AEROSTRUCTURAL COUPLING

As the aerodynamic mesh and the structural mesh are non-coincident, the aerodynamic loads need to be transferred to the structural mesh and applied to the structure. Simultaneously, the structural deformations need to be modeled in the aerodynamic mesh to model aeroelastic effects. How this is achieved is known as aerostructural coupling. Aerostructural coupling is achieved most often with interpolation methods, also known as splining methods, which allows expressing quantities in different reference frames and grids. Important when doing aerostructural coupling is that the interpolation method should be conservative, preserving virtual work during the transfer of loads and displacements between meshes. Although the aerostructural coupling method choice can influence model accuracy, the methods are considered relatively mature. Consequently, research has mostly focused on improving the aerodynamic models and structural models instead, as these offer greater potential improvements in model accuracy. Common splining methods include beam spline, rigid-body spline, thin plate spline, and radial basis functions [93]. The result is a splining matrix that relates the DOFs in one grid to the DOFs in another grid. To illustrate the construction and idea of a splining matrix, as this concept appears repeatedly throughout this thesis, the rigid-body spline is taken as an example. The core of this spline is Equation 2.17, which allows expressing the displacement and rotation at the dependent grid in terms of the displacement and rotation at the independent grid. These matrices are assembled for each node pair of the independent grid and the dependent grid and placed into one big matrix, resulting in an $6d \times 6i$ sized matrix. Not all equations are shown here to limit the size of this discussion. In fact, the distance vector $\boldsymbol{r}$ is expressed in the global frame, and thus the global spline matrix must be expressed as a local spline matrix that maps the DOFs of the independent and dependent grids. The splining approach is rather versatile, and can also be used to transfer forces and moments between different reference frames. Additionally, the induced downwash is also computed using these splining relationships.

$$\boldsymbol{u} = \begin{bmatrix} u_x & u_y & u_z \end{bmatrix}^T \tag{2.14}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_x & \theta_y & \theta_z \end{bmatrix}^T \tag{2.15}$$

$$\boldsymbol{u}_d = \boldsymbol{u}_i + \boldsymbol{\theta}_i \times \boldsymbol{r}_{di} = \boldsymbol{u}_i - \tilde{\boldsymbol{r}}_{di}\boldsymbol{\theta}_i \tag{2.16}$$

$$\begin{pmatrix} \boldsymbol{u}_d \\ \boldsymbol{\theta}_d \end{pmatrix} = \begin{bmatrix} \boldsymbol{I} & -\tilde{\boldsymbol{r}}_{di} \\ \boldsymbol{0} & \boldsymbol{I} \end{bmatrix} \begin{pmatrix} \boldsymbol{u}_i \\ \boldsymbol{\theta}_i \end{pmatrix} \tag{2.17}$$

2.1.2. EXISTING TOOLS AND FRAMEWORKS

Building a free-flying aeroservoelastic modeling framework is an involved effort, and consequently, several tools and frameworks have already been developed. This section reviews the available tools mentioned in prior research. Although additional modeling approaches are presented in literature, many are tailored for a single research application and lack the generality to be described as a framework. A practical distinction in this regard is whether the model is recognized and named as a tool or framework. Additionally, aircraft manufacturers often maintain proprietary in-house tools and frameworks, which are therefore also excluded from this review, as there are no details available on these tools. This overview merely serves as a list of tools

available, and not an in-depth analysis of each. This was done to later, in Subsection 2.3.2, select one of these frameworks to further refine and build upon.

NASTRAN

NASA Structural Analysis (NASTRAN) is the aerospace industry-standard for finite-element analysis and aeroelastic analysis. NASTRAN is capable of free-free analysis thereby including the rigid-body DOFs. However, NASTRAN does not implement the flight dynamic equations with the Coriolis, and gyroscopic term, but rather linear rigid-body equations of motion. Additionally, state-space models cannot be extracted from NASTRAN. However, it is possible to extract matrices from NASTRAN that can be used to construct such models, by using a DMAP ALTER to extract the mass and stiffness matrices, as well as the aerostructural splines, for instance. NASTRAN provides dedicated solution sequences for aeroelastic analysis: SOL144 for static aeroelasticity, SOL145 for flutter analysis, SOL146 for dynamic aeroelastic response, and SOL200 for design optimization with aeroelastic constraints. NASTRAN has various aerodynamic models built in, with DLM most often used in aeroelastic analysis. Additionally, it implements the Mach box method, strip theory, and piston theory. It can also be coupled with external CFD software for higher-fidelity aerodynamics. Typically, linear FEM is used in NASTRAN, although SOL400 provides nonlinear FEM as well. The most popular NASTRAN implementations are MSC NASTRAN [93] and Simcenter NASTRAN [94]. Extensive documentation of NASTRAN is publicly available.

MSC FLIGHTLOADS

MSC FlightLoads [95] is a commercial software tool that wraps MSC NASTRAN and MSC Patran and simplifies loads analysis by streamlining the entire loads modeling process. FlightLoads provides an environment for creating aerodynamic models, performing aeroelastic analyses, and generating external loads for structural design across multiple load cases. The structural FEM is assumed to be pre-existing in FlightLoads. FlightLoads also provides a tool for visualizing the load cases to identify critical load cases. Extensive documentation of FlightLoads is publicly available. As FlightLoads wraps around NASTRAN, it is limited by the same constraints.

ZONA ZAERO/SIMCENTER ZONA

Another major software package for aeroservoelastic analysis is ZONA ZAERO, which is now part of Siemens PLM, and continues as Simcenter Zona. ZAERO is essentially similar to NASTRAN and FlightLoads, but is more advanced in every regard. It integrates all disciplines required for advanced aeroelastic analysis such as aeroservoelasticity, aerothermoelasticity, structural optimization, multidisciplinary design analysis and optimization, and flight loads. It implements a so-called unified higher-order panel method that can describe subsonic, transonic, supersonic, and hypersonic aerodynamics, using a single model. For higher-fidelity analysis, it can also be coupled with CFD based on Euler equations. Structural nonlinearities can also be included to perform nonlinear aeroelastic analysis. Although highly capable, it has several limitations that make it less suitable, especially in academia and research. First, it is commercial and thus closed source, although papers and documentation describing the underlying models are available.

ASWING (1999)

ASWING, developed by Drela [96] at MIT in 1999, is a flexible aircraft flight dynamics tool for the analysis of very flexible aircraft in the preliminary design phase. The aerodynamic model used is an unsteady lifting-line model with wind-aligned trailing vortices. The structural model is a nonlinear Euler-Bernoulli beam and is therefore well suited for large displacements. ASWING also implements linear state feedback control. Although an older software, ASWING remains relevant. Recently, Avoni, Bronz, Condomines, *et al.* [97] introduced WingLoop, a framework built around ASWING that enables running ASWING simulations directly from Python and extends the control capabilities of ASWING with nonlinear control in addition to the linear control capabilities already present in ASWING. Therefore, any control law can be implemented on any aircraft defined using ASWING.

VARLOADS (2003)

VarLoads, developed by Kier, Hofstee, Cerulli, *et al.* [37] at Airbus and DLR in 2003 is an in-house tool developed in MATLAB/Simulink for loads analysis and special investigations, ranging from preliminary design to in-flight loads analysis. Although an in-house tool, its general implementation is described in literature. At its heart are equations of motion based on mean-body axes with the practical mean-axes constraints applied.

The structural model used is based on linear FEM. The distinguishing innovation at the time of creation was its modular and object-oriented structure. Because of this modular structure, VarLoads provides an extensible platform that can be integrated with other disciplines. Due to its modular nature, it can be integrated with different aerodynamic models, although the paper is tailored towards aerodynamic strip theory.

NATASHA (2006)

Nonlinear Aeroelastic Trim and Stability of HALE Aircraft, abbreviated as NATASHA, was developed by Patil and Hodges [98], is a flexible aircraft flight dynamics program for the analysis of very flexible aircraft. It therefore implements a geometrically-exact structural model in the intrinsic form and a subsequent aerodynamic model using strip theory, allowing for straightforward aerodynamic modeling under large deformations.The paper highlights the need for such models, as the phugoid of the considered HALE aircraft becomes unstable, potentially leading to catastrophy. The program is not publicly available, although the methodology is described in detail.

NEOCASS (2009)

At Politecnico di Milano, Cavagna, Ricci, and Riccobene [99] developed NeoCASS. NeoCASS is a broader toolsuite written in MATLAB aimed to bringing aeroelasticity earlier into the preliminary design phase, to prevent costly weight estimation errors early on. The software comprises three modules: WB, GUESS, and SMARTCAD. WB provides mass and inertia estimates as well as their distribution. GUESS performs a loads analysis and structural sizing, and upon convergence, outputs an aeroelastic stick model that is fed to SMARTCAD. SMARTCAD is the aerostructural analysis and optimization tool, capable of performing aeroelastic calculations such as flutter analysis, aeroelastic trim calculations, and aeroelastic stability and control derivatives. The program can be accessed upon request. As the focus is on bringing aeroelasticity earlier into the design phase, it does not seem to be focused on free-flying aeroservoelastic time simulation.

UM/NAST (2010)

The University of Michigan Nonlinear Aeroelastic Simulation Toolbox, abbreviated as UM/NAST was developed by Su and Cesnik [51]. It presents a methodology for coupling aeroelasticity and flight dynamics using a geometrically nonlinear strain-based finite-element model coupled with an unsteady aerodynamic strip model. The tool is therefore, like NATASHA, well-suited for very flexible aircraft undergoing large displacements. The paper considers a very flexible blended-wing body aircraft, which showed that the unconstrained vehicle suffered from body-freedom flutter, highlighting the importance of including rigid-body degrees of freedom when determining the flutter boundary.

AEROFLEX (2012)

AeroFlex, developed by Ribeiro, Paglione, Annes da Silva, *et al.* [100] at Instituto Tecnológico de Aeronáutica, is a publicly available MATLAB program for the analysis of very flexible aircraft, and is mostly based on the work by Shearer and Cesnik [101] and Su and Cesnik [51].

AMLOAD (2016)

Also at NLR, a dedicated aeroelasticity and loads tool was developed by Timmermans and Prananta [102], similar in its objective to NeoCASS, and also written in MATLAB. However, in terms of methodology, it is more similar to FlightLoads, as it acts as a pre- and post-processor for MSC NASTRAN, rather than implementing its own coupled aeroelastic and flight dynamic equations. It generates an aeroelastic stick model based on linear FEM with panel aerodynamics that can be consumed by NASTRAN. The post-processor then builds loads envelopes and flutter diagrams. AMLoad also provides the option to incorporate higher-fidelity FEM using DMIG matrices and apply aerodynamic corrections from CFD if desired in later design stages.

CA2LM (2016)

The Cranfield Aircraft Accelerated Loads Model (CA2LM) was developed by Dussart, Portapas, Pontillo, *et al.* [47] and Portapas, Cooke, and Lone [103] was conceived to investigate the flight dynamics and handling qualities of flexible aircraft. CA2LM is implemented in MATLAB/Simulink, but is not publicly available. It uses Modified Strip Theory (MST) or Unsteady Vortex Lattice Method to model unsteady aerodynamics. Linear FEM from NASTRAN is used as the structural model. The equations of motion are formulated using mean-body axes with the practical mean-axes constraints applied.

SHARPy (2019)

SHARPy, developed by Del Carre, Muñoz-Simón, Goizueta, *et al.* [78] at Imperial College London in, is a popular Python-based aeroelastic package tailored towards the aeroelastic and flight dynamic analysis of high aspect ratio wings and aircraft as well as wind turbines. SHARPy uses UVLM as the underlying aerodynamic model and a geometrically-exact beam formulation as the structural model. Although SHARPy is highly capable, its usability is limited by the need to define models within the SHARPy environment, which makes integration with existing aeroelastic workflows challenging. SHARPy enables full nonlinear coupled aeroelastic and flight dynamics analyses, although the model can also be linearized and used for control system design.

DLR Loads Kernel (2020)

Voß [19] developed DLR Loads Kernel at the DLR Institute of Aeroelasticity, which is a comprehensive Python package for loads analysis. In terms of implementation, it is similar to VarLoads. It is the first free-flying aeroelasticity modeling tool that is both publicly available and whose inputs are NASTRAN-based. All inputs to Loads Kernel are defined in a Python script called the Job Control File (JCL). Loads Kernel comprises three modules: the preprocessor, main module, and post-processor. The preprocessor consumes a NASTRAN structural mesh as defined using a Bulk Data File (BDF) as well as an aerodynamic mesh as defined using CAERO, AELIST, and AESURF cards. The global mass and stiffness matrices as output by NASTRAN are also required as input. The preprocessor transforms these inputs into matrices to be consumed by the main module that runs the desired simulations. Loads Kernel also comes with a useful GUI to physically inspect the subsequent model. Loads Kernel uses mean-body axes with the practical mean-axes constraint applied. The aerodynamic model uses VLM for quasi-steady aerodynamics and DLM for unsteady aerodynamics, which can be imported from NASTRAN or generated using the implementation by Voß [20]. Loads Kernel is also capable of being coupled to integrate with CFD. The structural model is linear FEM. The main module supports multiple simulation types: linear quasi-steady, linear unsteady, and nonlinear quasi-steady time-domain, frequency-domain analysis, and both quasi-steady and unsteady CFD simulation. The post-processor computes the desired section loads and enables inspection of them in a dedicated GUI.

UltraFLoads (2023)

More recently, Feldwisch and Bauer [104] developed UltraFLoads at the DLR Institute of Aeroelasticity and DLR Institute of Aerodynamics and Flow Technology, tailored towards loads analysis using high-fidelity CFD and CSM. It integrates with FlowSimulator, an ecosystem that simplifies data exchange in CFD workflows, allowing integration with state-of-the-art CFD tools such as DLR TAU and, in the future, DLR/ONERA/Airbus CODA. Of all listed frameworks therefore, UltraFLoads allows the most advanced and accurate analysis. However, it is not accessible outside of DLR, requires access to proprietory CFD software, and is computationally resourceful because of the high-fidelity underlying models.

Braunschweig SE2A Flexible Model (2024)

At TU Braunschweig, Beyer, Steen, and Hecker [18] developed an open-source simulation environment in MATLAB/Simulink, although no particular name has been assigned to the tool, but the full implementation used in their paper is publicly available. The implementation is based on mean-body axes with practical mean-axes constraint. Aerodynamic strip theory is used for the unsteady aerodynamics, and linear FEM is used for the structural model, with Mode Displacement Method used for loads recovery. The model was used to demonstrate an Incremental Nonlinear Dynamic Inversion (INDI) control law, which is extended to allow faster actuator response. Although the model is publicly available in MATLAB/Simulink, it unnecessarily uses indicial functions for the unsteady aerodynamics, MDM for loads recovery, and a condensed FEM. Although the source code is not studied in great detail, the emphasis of this paper is on the INDI law, and likely the software is not architected as to support different free-flying models, limiting its capabilities to be considered a general modeling framework.

FENIAX (2025)

Another recent addition to the suite of free-flying tools is FENIAX, developed by Cea and Palacios [91] at Imperial College London. FENIAX is novel in that it augments linear NASTRAN models with geometric nonlinear effects and its use of automatic differentiation using JAX [105]. Automatic differentiation is a computational method to compute exact derivatives by applying the chain rule to each numerical operation in numerical computation. This allows to add geometric nonlinear effects to current industrial aeroelastic

workflows at no additional computational cost. Moreover, the automatic differentiation capabilities provide promising opportunities in multidisciplinary design optimization. The aerodynamic model used in FENIAX is DLM. The structural model is linear FEM that is condensed to the primary load paths, and enhanced with a geometrically-exact intrinsic beam formulation to capture nonlinear effects, resulting in what the authors refer to as a Nonlinear Modal Reduced Order Model (NMROM). The software is implemented in Python and publicly available.

2.2. FIELD-LEVEL CHALLENGES

Several field-level challenges were identified during the literature review. This discussion is structured to first highlight the practical challenges that hinder progress in the field, followed by the identified theoretical challenges that, if addressed, hold the potential to advance the field. Surely, this is not an exhaustive list, but a compilation of challenges that were identified.

PRACTICAL CHALLENGES

- The first practical challenge that anyone entering this research field will inevitably encounter is the highly multidisciplinary nature of free-flying aeroservoelasticity, which requires substantial theoretical knowledge across multiple domains, such as flexible body dynamics, aeroelasticity, structural dynamics, unsteady aerodynamics, state-space modeling and controller design, and actuator modeling, as well as practical experience with industry-standard tools such as NASTRAN and MATLAB/Simulink. Essentially, the field spans the entire aerospace engineering domain of fundamental disciplines. It is fair to consider this a field-level challenge, as it can definitely hinder advancement of knowledge, and generation of new knowledge as one requires understanding of all disciplines before getting to useful insights.
- The second practical challenge, which was underscored in the survey of free-flying aeroelastic tools and frameworks in Subsection 2.1.2, is that the implementation of many free-flying frameworks are not publicly available, further complicating contributing or augmenting their capabilities. This second challenge ties in with the first as more time as much time is spent on understanding and building such frameworks, before being able to apply them. It was observed that the publications of each framework often do not provide sufficient insight into the exact workings of the models.
- Another practical challenge is the segmentation of workflows for free-flying aeroelastic modeling. Although some software tools such as SHARPy and ASWING allow defining the aircraft model inside their environment, this is precisely what limits their applicability outside of research, as industrial NASTRAN models cannot be used. The steps required to build these industrial models and convert them to free-flying aeroelastic models are typically divided over several software tools. This complicates fast iteration as users must first become skilled in this before being able to contribute. Moreover, design optimization is difficult as the model is not built and analyzed in the same software.
- The last practical challenge is primarily in the academic sphere. Industrial NASTRAN finite-element models of appropriate fidelity are proprietary information of aircraft manufacturers, such as the Airbus XRF1, or otherwise developed in-house. Both are typically controlled in terms of distribution, and it can thus be a challenge in finding a suitable model for a certain application in this field. Clamped-wing finite-element models, such as the uCRM Benchmark Wings [106] are more widely available.
- It seems as if higher-fidelity methods for aerodynamic modeling are mostly exclusive to the industry, such as aircraft manufacturers and industrial research institutes. Although completely understandable, this is a bit unfortunate, as it means the true state-of-the-art cannot be easily used within this thesis, for instance. On the other hand, high-fidelity is also not used in preliminary design to enable quick design iteration, and thus low-fidelity is used instead, so this only holds for more advanced design stages in general.

THEORETICAL CHALLENGES

- First, although not necessarily a challenge, it is remarkable that the aerodynamic models currently employed for free-flying aeroelastic models largely originate from the 1960s and 1970s, such as DLM [60], particularly given the significant advances in computational power and numerical methods that would seemingly enable wider use of higher-fidelity models, altough this transition seems to have started, with the wider use of linearized CFD for these applications.

- Similarly, these aerodynamic models are often linear, which raises doubts on their validity in considerably flexible aircraft and nonlinear aerodynamics in general, which in case UHARW aircraft will definitely become important to account for.
- These linear aerodynamic models are computationally quite heavy to construct, as the aerodynamic influence between each panel must be evaluated. The nature of these panel aerodynamic models is such that recomputing the aerodynamic influences is computationally feasible only for offline simulation while remaining impractical for applications that require repeated model evaluations, such as controller design, unless not updating the aerodynamic influence matrix, which is common practice.
- Because the aerodynamic model is only valid at the reference flight condition, in order to construct the flutter boundary, or consider many different flight conditions in general, a lot of aerodynamic models need to be constructed. This is not necessarily a challenge, but rather a limitation. Due to the nature in which compressibility is taken into account, it would be difficult to express the model continuous in the Mach number. The same applies for the reduced frequency, which is why an RFA is needed.
- As was noted before, geometric nonlinearities in free-flying aeroservoelasticity has been relatively unexplored and would be the next advancement in terms of fidelity. It is interesting that the structural model is relatively more advanced compared to the aerodynamic model, although the aerodynamics define the loads, which is seemingly of equal importance.
- Another challenge is a seeming lack of consensus on what is the most appropriate modeling choice for what type of aircraft and flight condition. There are simply no clear cut answers to questions such as when should geometric nonlinearities be taken into account, when should inertial coupling be accounted for, when should aerodynamic nonlinearities be accounted for, among others.
- Lastly, in the future, control co-design optimization or integrated aeroservoelastic optimization likely could future squeeze more performance out of aircraft designs. All the above issues, primarily those relating to computational efficiency, model fidelity and segmentation of workflows, render higher-fidelity optimization simply unfeasible. Take for instance, the high computational cost of recomputing the aerodynamic influences in panel aerodynamics. If one desires aeroservoelastic optimization of the wing design including the planform, the problem becomes computationally too demanding in a realistic amount of time.

2.3. RESEARCH GAPS

The presented field-level challenges are much broader in scope than what is tackled within the scope of this thesis. This section presents the specific gaps that will be addressed in this thesis. A reference modeling framework is selected that will be modified to suit the needs of free-flying aeroservoelastic models required for analyzing both passive and active load alleviation. However, requirements on what a free-flying aeroservoelastic model for load alleviation should adhere to, are first formulated in Subsection 2.3.1. Subsection 2.3.2 selects a suitable reference framework from the surveyed tools in Subsection 2.1.2 to further refine. Subsequently, Subsection 2.3.3 shortly summarizes the framework, critically analyzes this framework in terms of its suitability for free-flying aeroservoelasticity and identifies the research gaps addressed in this thesis.

2.3.1. AEROSERVOELASTIC MODEL REQUIREMENTS

As was highlighted, aeroservoelasticity concerns the interaction between unsteady aerodynamics, structural dynamics, and control systems. Typically, aircraft loads models are purely used for simulation of loads. However, the requirement to simultaneously use loads models in a control feedback system for active load alleviation drives several requirements on this aeroservoelastic model. These requirements have been summarized below, and are used to refine a suitable reference modeling framework selected in Subsection 2.3.2.

- As NASTRAN is the industry-standard for finite-element modeling, structural models often come in the form of a NASTRAN model. Therefore, an important requirement on the aeroservoelastic model the ability to consume NASTRAN models. As aeroelastic tailoring workflows, for passive load alleviation, therefore often integrate with NASTRAN this is a important requirement for free-flying aeroservoelastic models to be used for both passive and active load alleviation. Moreover, compatibility with NASTRAN reduces dependency of the model on one particular aircraft configuration.

- In order to apply most modern control techniques, a state-space model of the aircraft loads model is required, which is neatly formulated into a state equation and output equation, with explicit definition of the state, input, output and disturbance vectors. Casting the model into a state-space representation and linearizing around a trim point yields a linear time-invariant state-space model with which modern control techniques can be applied.

- Preferably, the model should be expressed in closed-form and symbolically solely using matrices and parameters, such that model dependencies between the states, inputs, disturbances and outputs are revealed, in addition to the nonlinearities in the model. Although control methods can be model-dependent or independent, expressing the model as a closed-form, symbolic model is therefore more versatile than having a model more akin to a black box. Additionally, this comes with the added benefit that the model itself is not dependent on the programming environment it was made in. As MATLAB/Simulink is the industry-standard tool for controller design, such a model can be imported into this environment easily.

- In Gust Load Alleviation, gusts are fed as disturbances into the aeroservoelastic state-space model. How this is achieved, however, can be challenging as was highlighted in Subsection 2.1.1. Particularly, it is important that the gust modeling does not increase the order of the state-space model, either in terms of state variables or input variables. Additionally, the time delay nature of gusts, also known as the gust penetration effect, which brings about the spiral nature of the Sears function should be appropriately modeled.

- Synthesizing controllers can be computationally expensive and therefore the order of the state vector should be as low as possible while preserving the input-output behaviour. Although model order reduction methods exist for further reducing the state size, often the state interpretability is reduced, additional approximation errors are introduced, and the computational effort scales with the state order, which motivates reducing the model order as early as possible.

- It is desirable to be able to recover the structural loads using the Force Summation Method as opposed to the Mode Displacement Method as this yields more accurate loads recovery when using a truncated modal basis. A truncated modal basis is often used in aeroservoelasticity to reduce the model order and therefore FSM is preferred over MDM.

- As the timescales in maneuvers, and, most importantly, gusts, are quite small, it is important to include actuator dynamics in order to model the actuator bandwidth, time delays, and saturation due to rate and deflection limits. Additionally, depending on the type of sensor used, most often strain gauges or accelerometers, an appropriate sensor model should be included in the output equation.

- The model shall be computationally efficient. Controller tuning is an iterative process, and it is therefore preferable to have a computationally efficient simulation model to speed up the design process.

2.3.2. DLR LOADS KERNEL RATIONALE & SELECTION

Deriving and building a framework for aircraft loads analysis and free-flying aeroservoelasticity is an involved effort, and it would be a waste of time to not reuse past work. Therefore, rather than building an entire framework from scratch, an existing framework is taken as a baseline which is further refined to make it suitable for free-flying aeroservoelasticity in line with the requirements set in Subsection 2.3.1. The requirements on this reference framework are twofold. First, its implementation should be publicly available in order to integrate with it. Moreover, the survey of existing tools and frameworks in Subsection 2.1.2 identified that documentation is often insufficient while the tools are complicated, and therefore an public implementation is essential in order to critically analyze the framework. Secondly, the framework should be NASTRAN-based as this is the industry-standard tool for finite-element modeling and seamlessly integrates with existing aeroelastic workflows, and also to make it suitable for passive load alleviation in free flight. Lastly, the framework should be representative of the state-of-the-practice rather than state-of-the-art, as a reference should be an established baseline rather than something pushing the boundaries from the outset. Out of the tools and frameworks presented in Subsection 2.1.2, DLR Loads Kernel is therefore selected as the most suitable baseline framework to build upon.

2.3.3. DLR LOADS KERNEL CRITIQUE & GAPS

DLR Loads Kernel comprises a preprocessor, a main processor, and a post-processor. Loads Kernel is based on linear FEM underpinning the structural model, practical mean-axes EOMs, aerodynamic model based on VLM for steady aerodynamics and DLM for unsteady aerodynamics or integration with CFD for higher-fidelity aerodynamics, and aerostructural coupling using radial basis functions or rigid-body splines. All inputs to Loads Kernel are defined in a separate Python script called the Job Control File (JCL). A problem with the setup of the JCL is that Loads Kernel supports a wide range of settings, each often coming with its own input requirements, yet the JCL does not enforce these inputs for the selected settings. As a result, it is the responsibility of the user to determine the required inputs for their desired mode of operation, which makes the JCL less user-friendly and increases the learning curve. It is therefore difficult to clearly summarize what inputs it requires, as this requires some research and is case-dependent, but broadly the inputs it requires are as follows. It is important to note that this merely serves as an overview of the type of inputs Loads Kernel requires, and is anything but exhaustive and complete. Additionally, Loads Kernel also allows varying levels of control. For instance, one can compute AIC matrices based on the imported NASTRAN aerodynamic mesh using the implementation by Voß [20], but can also directly import AIC matrices as generated by NASTRAN. Loads Kernel also integrates with CFD, which is not covered here. Integration with other aerodynamic models is not implemented.

- NASTRAN-based structural model:
  - Structural mesh definition using GRIDs, CORDs, CQUADs, and CTRIAs
  - Monitoring stations definition using GRIDs or MONPNTs
  - Categorization of all structural DOFs using USET table
  - Global mass and stiffness matrices as exported by a NASTRAN-based FEM solver
- NASTRAN-based aerodynamic model
  - Aerodynamic panel mesh definition using CAEROs
  - Aerodynamic surfaces definition using AELISTs and AESURFs
  - Number of poles required for the RFA
  - Reduced frequencies at which to compute AICs

The preprocessor performs several steps to convert NASTRAN-based inputs into the data required by the main module that performs the actual analysis and simulation. Although the precise workflow depends on the JCL settings, core tasks typically include:

- Importing and processing the NASTRAN structural and aerodynamic mesh
- Defining grids and reference frames such as the aerodynamic grid, structural grid, control surface grids, monitoring grids, thrust grids, sensor grids, body frame, flight physical frame, and inertial frame
- Importing global mass and stiffness matrices and removing the non-free DOFs, performing a modal analyis and/or static condensation
- Building the AICs, or importing them, and compute RFA matrices
- Building the splining matrices for interpolation between the different grids and reference frames

A more detailed description of the processes performed by the preprocessor is shown in Figure 2.2. The preprocessor, although it can come across as clunky in terms of software architecture, gets the job done and is therefore a decent tool to reduce the redundant work of importing and processing NASTRAN input decks as well as generating the required building blocks for free-flying state-space models. Loads Kernel also comes with a useful GUI to physically inspect the model generated by the preprocessor, which is useful for verification of the aircraft model. The preprocessor therefore is deemed satisfactory for the purposes of this thesis. If one desires to perform multidisciplinary design optimization and/or control co-design optimization, it could be worth, however, to critically examine its capabilities with these applications in mind, as it is not particularly quick. An additional issue in such applications would be that NASTRAN needs to rebuild global mass

and stiffness matrices in each iteration, which is arguably a larger problem.

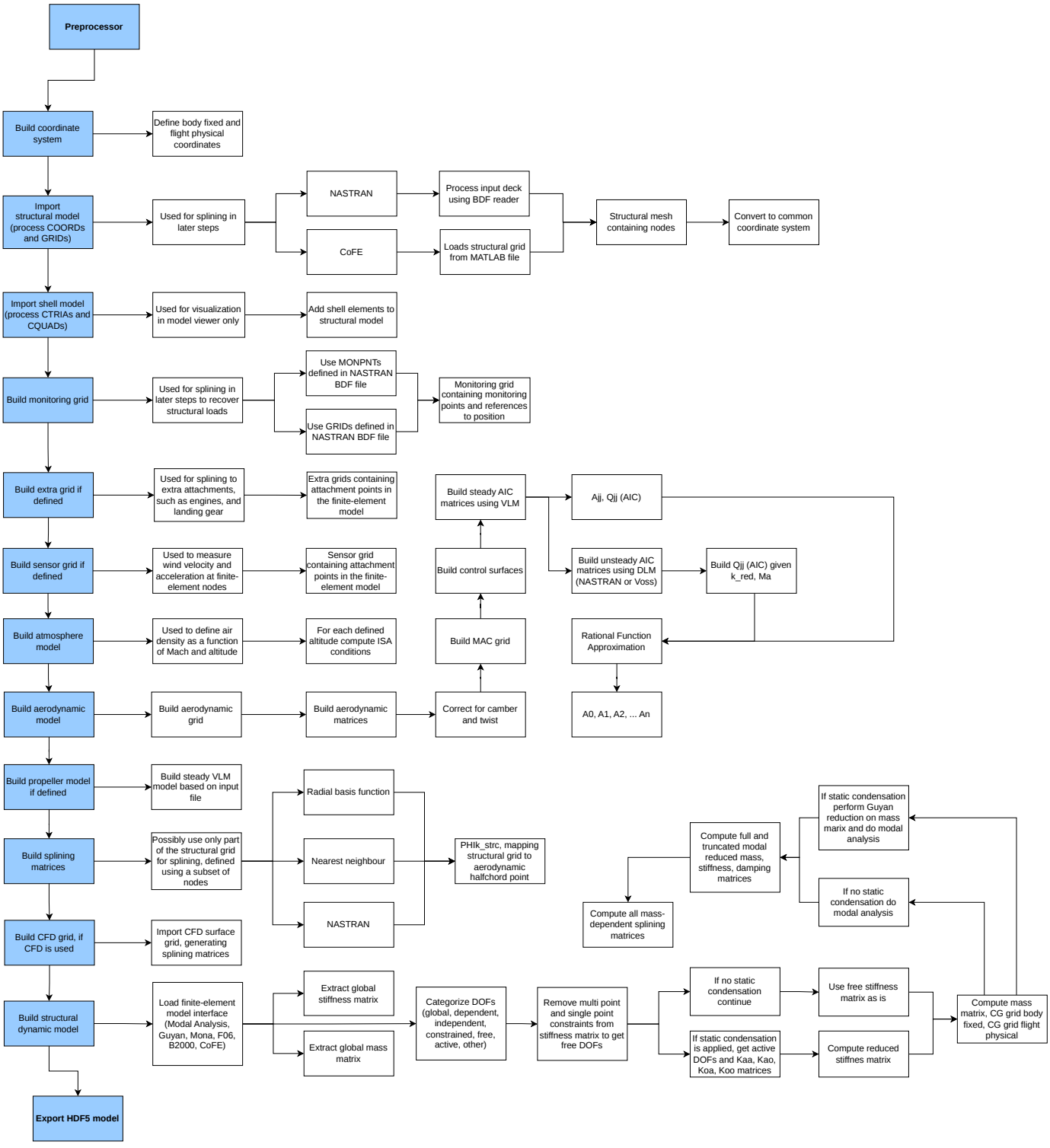


Figure 2.2: DLR Loads Kernel preprocessor flowchart

Although Loads Kernel is is an impressive, all-encompassing and comprehensive tool, the main simulation module has issues that make it less suitable free-flying aeroservoelasticity, of which the main steps are de-

picted in the flowchart of Figure 2.3. The documentation provided for Loads Kernel consists of a tutorial, user guide [19], and technical report on the aerodynamic model [20], and, logically, the full Python implementation. Based on reverse-engineering and critical analysis of the framework, several issues with Loads Kernel have been identified that do not align with the requirements set in Subsection 2.3.1.

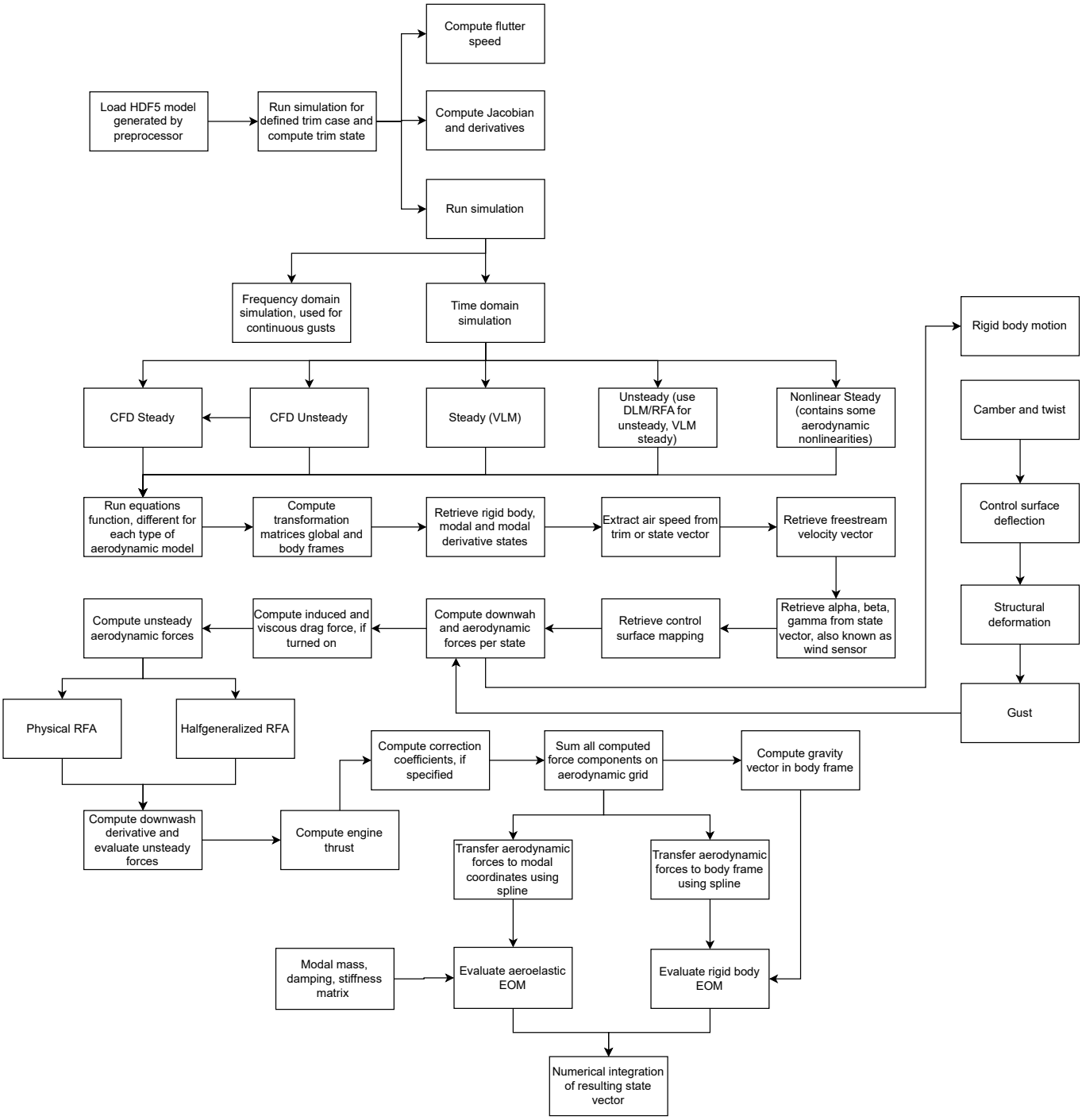


Figure 2.3: DLR Loads Kernel main module flowchart

These issues can be traced back to the following root cause. This root causes propagates into several issues classified as relating to either 1) interpretability and traceability, 2) implementation, efficiency and documen-

tation, and 3) aeroservoelastic applications suitability.

> The fundamental issue is that DLR Loads Kernel lacks an explicit, continuous-time state-space formulation with formally defined state and output equations using matrices and parameters, as well as the corresponding input, output, state, and disturbance vectors. The model is therefore almost a black box unless the implementation is deeply scrutizined.

This issue also becomes evident looking at the flowchart of Figure 2.3, as the model is numerically built in a sequential fashion, rather than in a mathematical, monolithic approach.

MODEL INTERPRETABILITY & TRACEABILITY

The interpretability and traceability of the model concerns understanding how the model functions, how it is derived, and understanding the model interdependencies. The lack of a formal state-space model introduces the following issues:

- Model dependencies between the simulation model inputs and outputs are opaque and lead to black-box behaviour. Using a documented state-space model, the model interdependencies can be understood with a single glance. It is therefore also unknown what the nonlinearities in the equations are, unless one peeks deep into the numerical model, as they have not been documented explicitly. In hindsight, this only concerns the flight dynamic nonlinearities and nonlinearities in the output equation. However, as it has not been explicitly mentioned, it would be hard to tell for inexperienced users of the software.
- The model is insufficiently derived, and modeling choices are mostly undocumented. On the other hand, an explicit state-space model that is rigorously derived and formulated is self-documenting. As such, the underlying assumptions are also not explicitly stated.
- Due to the lack of a mathematical derivation of the model, there are unaccounted for model simplifications or in the model. Although these could be intended, the modeling choices have not been documented. These include:
  - The panel downwash derivative is computed using backward finite-difference. This is a bit negligent, as the equations are continuous in nature. This makes the model formulated in discrete-time rather than continuous-time.
  - It seems as if the downwash due to control surface motion and acceleration is not captured fully, which should arise naturally when deriving the equations. The control surface motion should cause motion-induced downwash in addition to displacement-induced downwash. In the Loads Kernel formulation, the control surface acceleration should then appear through the panel downwash derivative.
  - The formula for computing displacement-induced downwash is not grounded in first principles and is inconsistent. The formula for the motion-induced downwash makes sense: the induced velocities are projected along the panel normals.

$$w_j = -\frac{1}{V_\infty}\begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} \cdot \begin{pmatrix} U_j \\ V_j \\ W_j \end{pmatrix} \tag{2.18}$$

  However, the downwash computation due to displacement is inconsistent. Downwash due to wing deformation is computed using Equation 2.19, and due to control surface deflection using Equation 2.20. Both control surface deflection and flexible deformation both deflect the flow, and it is therefore expected that the resulting equations should be equal. Moreover, it is difficult to understand how these equations were formulated, as their derivation is never stated.

$$w_j = \frac{1}{V_\infty}\begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} \cdot \left[\begin{pmatrix} \phi_j \\ \theta_j \\ \psi_j \end{pmatrix} \times \begin{pmatrix} U \\ V \\ W \end{pmatrix} \right] \tag{2.19}$$

$$w_j = \begin{pmatrix} N_x \\ N_y \\ N_z \end{pmatrix} \cdot \left[\begin{pmatrix} \phi_j \\ \theta_j \\ \psi_j \end{pmatrix} \times \begin{pmatrix} -1 \\ 0 \\ 0 \end{pmatrix} \right] \tag{2.20}$$

The original NASTRAN implementation uses differentiation matrices [93], also adopted in the work of Kier and Looye [39]. These matrices take the relevant components of the induced velocities and rotations at the half-chord of each panel. The differentiation matrix for displacement-induced downwash considers the panel rotation around the body y-direction, but this does not generalize to panels with inclination and deflection of vertical panels.

$$w_j = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{u}_k \qquad w_j = -\frac{1}{V_\infty} \begin{bmatrix} 0 & 0 & 1 & 0 & -c_j/4 & 0 \end{bmatrix} \boldsymbol{u}_k \tag{2.21}$$

Interestingly, as the original NASTRAN implementation uses this linearized approach, and DLM in general assumes small, linear perturbations around a reference flight condition, it raises doubt whether using nonlinear aerodynamic boundary conditions does not break underlying assumptions in the DLM aerodynamic model, as DLM models a fixed, prescribed wake. Further investigation into the DLM model should uncover whether the boundary conditions may be arbitrarily specified, or whether this is introduces inconsistencies in the model, but this is considered out of scope for this thesis.

– In the aeroelastic equation of motion it seems that an inertial coupling term is added as a contribution to the generalized forces. However, the origin of this term remains unclear as it is not documented how it is brought about. Additionally, if it is an inertial coupling term, a corresponding term should also be added to the rigid-body equations.

IMPLEMENTATION, EFFICIENCY & DOCUMENTATION

The implementation issues concern those related to the computational efficiency, ease of use, learning curve, and extensibility of the software.

- In DLR Loads Kernel the simulation model is assembled online in each timestep, allowing to view and extract the values of intermediate matrices and variables. However, this leads to a low computational efficiency due to redundant computations. This would be an issue when having to perform multiple time domain simulations, as was identified as a requirement in Subsection 2.3.1.
- The software architecture could be improved. For example, the JCL configuration makes the user responsible for enforcing the right inputs, the codebase contains duplicated logic, long functions with multiple responsibilities, functions with side effects, and limited vectorization where this would likely be possible. Although the software is validated and works, the learning curve is quite steep due to this architecture. This is a common issue in software in academia.
- There is insufficient guidance on importing non-default models into DLR Loads Kernel. The user guide explicitly states that the Loads Kernel preprocessor is set up to directly interface with the data and files generated by DLR ModGen, a proprietory tool for generating aeroelastic NASTRAN-compatible models, complicating the use of Loads Kernel outside of DLR.

AEROSERVOELASTIC APPLICATIONS SUITABILITY

The application issues mostly relate to its suitability for aseroservoelastic applications, for which the requirements of Subsection 2.3.1 should be met. Additional deeper analysis of Loads Kernel with respect to these requirements identified the following issues.

- By default Loads Kernel uses physical RFA which leads to a massive number of aerodynamic lag sates, equal to $n \times p$, with n aerodynamic panels and p the number of RFA poles. Loads Kernel also implements a half-generalized approach leading to $2 \times m \times p$ lag states, but there is no explanation available on how this method is supposed to work. It appears to only use the modal velocity and acceleration to compute the aerodynamic lag forces. Regardless of its implementation details, it is therefore an approximation and no data is available on whether it is suitable.
- The downwash on the aerodynamic panels is adjusted per panel in order to model the spiral nature of the Sears function, also known as the gust penetration effect. This leads to the same number of gust inputs as aerodynamic panels, yielding n disturbance inputs. For a typical aircraft, the number of panels is at least 1000, and often a lot more. This large number of disturbance inputs makes it challenging to design modern controllers.

- The model can be linearized around a trim point to get an LTI state-space model with A, B, C, D matrices, which can be exported to MATLAB/Simulink. However, it would be desirable to be able to import the full model, including the state and output nonlinearities, in MATLAB/Simulink as well as this is the industry-standard software suite for control design.

- DLR Loads Kernel does not model actuator dynamics, which is important to include for Active Flutter Suppression (AFS), Maneuver Load Alleviation (MLA), and Gust Load Alleviation (GLA) as the actuator bandwidth, delays, and saturation limits can affect the performance of these control systems. This should be straightforward matter of augmenting the state-space model once the state-space model is realized.

- The sensor model in Loads Kernel is used to measure the local wind at the sensor positions. However, in Active Flutter Suppression (AFS), Gust Load Alleviation (GLA), and Maneuver Load Alleviation (MLA) accelerometers are often placed on the wing, which need to be modeled appropriately as an output equation of the state-space model. This should also be quite straightforward once there is a state-space model.

2.4. THESIS OBJECTIVES

The overarching aim of this thesis is to complement DLR Loads Kernel to derive state-space models that can be used for aeroservoelastic applications such as the design of Active Flutter Suppression (AFS), Gust Load Alleviation (GLA), and Maneuver Load Alleviation (MLA) controllers. This involves two primary contributions, in line with addressing most of the research gaps in Subsection 2.3.3.

1. **Core, Theoretical Framework** The first aim of this thesis is to derive and define an analytical free-flying aeroservoelastic state-space framework, expressed symbolically using matrices and parameters, with explicit expressions for the state and output equations as well as the corresponding state, input, disturbance, and output vectors. The model shall be derived using a mathematically rigorous formulation, providing a theoretical framework that enables clear interpretation and facilitates refinement and extension of the model. The model shall be backward compatible with DLR Loads Kernel for verification of this model, but it shall be formulated as a general mathematical framework that ingests the relevant matrices and parameters and spits out state-space models in nonlinear form. Specific attention will be paid to meeting the aeroservoelastic model requirements set in Subsection 2.3.1, and missing in Loads Kernel. Moreover, most of the issues highlighted in Subsection 2.3.3 should be addressed in this state-space model derivation, except for one, which leads to the second thesis objective.

2. **Complementary, Practical Framework** The second aim of this thesis is to address the more practical process of building a free-flying aeroelastic model that can be fed to DLR Loads Kernel, as a demonstration on how to get the required matrices and parameters that are fed into the state-space framework. To this end, a clamped-wing finite-element model from Embraer is available, which is used for research on aeroelastic tailoring. This clamped-wing model is to be converted to a free-flying model and appropriately processed to be imported into DLR Loads Kernel. It shall be possible to generate aircraft with different levels of wing tailoring, mass distribution, and flight conditions. Although this is applied to a specific clamped-wing model, using the Embraer wing brings two additional advantages. First, clamped-wing models are more widely available, and the presented method can be applied also to a different wing model. Secondly, it directly enables AFS, GLA, and MLA control design for aircraft with aeroelastically tailored wings, as past research has focused on the aeroelastic tailoring of this particular wing model.

2.5. RESEARCH QUESTIONS

In order to achieve the two objectives formulated in Section 2.4, four research questions are formulated, which are:

1. How can the contemporary aircraft loads model, based on matrix and parameter inputs from DLR Loads Kernel, be formally rederived into an explicit, closed-form, symbolic continuous-time state-space model?

2. How can this aircraft loads state-space model be further modified and/or augmented to be converted to become free-flying aeroservoelastic state-space model, in line with the aeroservoelastic model requirements set in Subsection 2.3.1?

3. How can clamped-wing finite-element models be systematically extended to a free-flying finite-element model that allows generation of free-flying aeroservoelastic state-space models for different wing mass and stiffness distributions, mass cases, and flight conditions?

4. To what extent does the derived state-space model replicate the dynamics of DLR Loads Kernel, how can discrepancies be explained, and how high is the computational efficiency improvement through this formulation?

3

Part I: Free-Flying Aeroservoelastic State-Space Framework

The objective of this chapter is to derive and develop an analytical free-flying aeroservoelastic state-space model, that is, a model describing the dynamics of a free-flying aeroservoelastic aircraft in the following form, with the state equation of Equation 3.1 describing the state dynamics, and the output equation of Equation 3.2 describing the outputs used for feedback. This chapter describes the derivation of this free-flying aeroservoelastic state-space model in a top-down fashion, starting with the mathematical description of the final model in Section 3.1. This approach is adopted because an overview of the final state and output equations, as well as the corresponding input, output, and disturbance vectors, provides a clearer understanding of the state-space derivation. As this section is only intended to show the final state-space model, the equations are presented without detailed explanation, with references to the corresponding sections that contain the full derivations.

$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{3.1}$$

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{3.2}$$

The model derivation starts with Section 3.3 presenting the equations of motion, which describe the nonlinear flight dynamics. Section 3.4 describes the structural model equations, which presupposes an industrial linear finite-model based on NASTRAN, and applies modal analysis. Section 3.5 subsequently describes the aerodynamic model equations, which is based on linear panel aerodynamics using VLM for steady aerodynamics and DLM for unsteady aerodynamics. Section 3.6 describes the aerodynamic coupling between the aerodynamic equations and the equations of motion. As additional forces and moments act on the aircraft, Section 3.7 is dedicated to deriving these contributions, which include the aircraft weight and engine thrust. Section 3.8 describes the rigid-body kinematics in the inertial reference frame. Section 3.9 augments the model with actuator states and dynamics. Section 3.10 adds an accelerometer sensor model to the state-space model. Section 3.11 derives a state-space model for the gust dynamics and inputs. Section 3.12 is devoted to recovering the structural loads from the states, inputs, and disturbances. Section 3.13 realizes the state-space model. The numerical integration approach is explained in Section 3.14. Subsequently, once the model is realized, Section 3.15 explains the trimming procedure for steady, level flight, and, lastly, Section 3.16 explains the numerical linearization to get an LTI state-space model.

3.1. Final State-Space Model

The state-space model consists of the state and output equations, of which the mathematical formulations are described in Subsection 3.1.5 and Subsection 3.1.6 respectively. The mathematical descriptions of the input vector, disturbance vector, state vector, and output vector are presented in Subsection 3.1.2, Subsection 3.1.3, Subsection 3.1.1, and Subsection 3.1.4, respectively.

3.1.1. State Vector

The state vector is composed of several constituent state vectors, which are described in this section. The body linear velocity vector is defined by Equation 3.3. The body angular velocity vector is defined by Equation 3.4. The aircraft position vector in the inertial reference frame is defined by Equation 3.5. The aircraft

orientation vector in the inertial reference frame is defined by Equation 3.6. The modal displacement vector and modal velocity vector are defined by Equation 3.7 and Equation 3.8 respectively, with m the number of modes. The actuator state vectors are defined by Equation 3.9 and Equation 3.10, representing the control surface deflection and rate respectively, with n the number of control surfaces, as each control surface has an associated actuator. The gust state vector is defined using Equation 3.19, with z the number of gust zones. The aerodynamic rigid-body lag force and moment vectors are defined as Equation 3.12 and Equation 3.14 respectively, with p the number of poles for the RFA. The aerodynamic generalized lag force vector is given by Equation 3.16. The aerodynamic lag forces are also separately defined for the cutting forces and moments with Equation 3.18, with s indicating the number of cutting forces and/or moments to be monitored. These are only used, however, in the output equations. Note that each monitoring station has 6 components: F_x, F_y, F_z, M_x, M_y , M_z. Finally, the state vector is represented with Equation 3.20, with order $12+2n+2m+2z+(6+m+s)p$. As a representative example, with 10 modes, 20 monitoring stations, 10 gust zones, 5 control surfaces, 4 RFA poles, this leads to an order of 606. Now, this is too large for aeroservoelasticity without further model order reduction. However, the largest number of states is due to the load lag states, which can be reduced by removing the deemed irrelevant monitoring stations from the state-space model, and focusing only on specific force and moment components, such as the WRBM and WRTM. This can be done as the load lag dynamics are uncoupled and do not affect the states, as is shown in Subsection 3.6.2. Only keeping the WRBM and WRTM, the order of the system becomes 134, which is good. If the modes are further truncated, which does affect the model behaviour, however, the order can be reduced to 104 for 5 modes.

$$\boldsymbol{V} = \begin{bmatrix} U \\ V \\ W \end{bmatrix} \tag{3.3}$$

$$\boldsymbol{\omega} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \tag{3.4}$$

$$\boldsymbol{P} = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \tag{3.5}$$

$$\boldsymbol{\Theta} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \tag{3.6}$$

$$\boldsymbol{\eta} = \begin{bmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_m \end{bmatrix} \tag{3.7}$$

$$\dot{\boldsymbol{\eta}} = \begin{bmatrix} \dot{\eta}_1 \\ \dot{\eta}_2 \\ \vdots \\ \dot{\eta}_m \end{bmatrix} \tag{3.8}$$

$$\boldsymbol{\delta} = \begin{bmatrix} \delta_1 \\ \delta_2 \\ \vdots \\ \delta_n \end{bmatrix} \tag{3.9}$$

$$\dot{\boldsymbol{\delta}} = \begin{bmatrix} \dot{\delta}_1 \\ \dot{\delta}_2 \\ \vdots \\ \dot{\delta}_n \end{bmatrix} \tag{3.10}$$

$$\boldsymbol{F}_l^q = \begin{bmatrix} F_x^q \\ F_y^q \\ F_z^q \end{bmatrix} \tag{3.11}$$

$$\boldsymbol{F}_l = \begin{bmatrix} \boldsymbol{F}_l^1 \\ \boldsymbol{F}_l^2 \\ \vdots \\ \boldsymbol{F}_l^p \end{bmatrix} \tag{3.12}$$

$$\boldsymbol{M}_l^q = \begin{bmatrix} M_x^q \\ M_y^q \\ M_z^q \end{bmatrix} \tag{3.13}$$

$$\boldsymbol{M}_l = \begin{bmatrix} \boldsymbol{M}_l^1 \\ \boldsymbol{M}_l^2 \\ \vdots \\ \boldsymbol{M}_l^p \end{bmatrix} \tag{3.14}$$

$$\boldsymbol{P}_l^q = \begin{bmatrix} P_1^q \\ P_2^q \\ \vdots \\ P_m^q \end{bmatrix} \tag{3.15}$$

$$\boldsymbol{P}_l = \begin{bmatrix} \boldsymbol{P}_l^1 \\ \boldsymbol{P}_l^2 \\ \vdots \\ \boldsymbol{P}_l^p \end{bmatrix} \tag{3.16}$$

$$\boldsymbol{L}_l^q = \begin{bmatrix} L_1^q \\ L_2^q \\ \vdots \\ L_s^q \end{bmatrix} \tag{3.17}$$

$$\boldsymbol{L}_l = \begin{bmatrix} \boldsymbol{L}_l^1 \\ \boldsymbol{L}_l^2 \\ \vdots \\ \boldsymbol{L}_l^p \end{bmatrix} \tag{3.18}$$

$$\boldsymbol{g} = \begin{bmatrix} g_{1,1} \\ g_{1,2} \\ g_{2,1} \\ g_{2,2} \\ \vdots \\ g_{z,1} \\ g_{z,2} \end{bmatrix} \tag{3.19}$$

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{\Theta} \\ \boldsymbol{V} \\ \boldsymbol{\omega} \\ \boldsymbol{\eta} \\ \dot{\boldsymbol{\eta}} \\ \boldsymbol{\delta} \\ \dot{\boldsymbol{\delta}} \\ \boldsymbol{g} \\ \boldsymbol{F}_l \\ \boldsymbol{M}_l \\ \boldsymbol{P}_l \\ \boldsymbol{L}_l \end{bmatrix} \tag{3.20}$$

3.1.2. INPUT VECTOR

The input vector is defined by Equation 3.23 and consists of two inputs: control surface inputs and thrust command inputs. The former is described with the control surface command vector Equation 3.21, again with c denoting the number of control surfaces, which feeds into the actuator equations, which is discussed in Section 3.9. The thrust command vector is equal to Equation 3.22, with o denoting the number of engines. Thrust is added to the model in Subsection 3.7.1. The order of the input vector is $c + o$.

$$\boldsymbol{u}_c = \begin{bmatrix} c_1 \\ \vdots \\ c_c \end{bmatrix} \tag{3.21}$$

$$\boldsymbol{u}_t = \begin{bmatrix} e_1 \\ \vdots \\ e_o \end{bmatrix} \tag{3.22}$$

$$\boldsymbol{u} = \begin{bmatrix} \boldsymbol{u}_c \\ \boldsymbol{u}_t \end{bmatrix} \tag{3.23}$$

3.1.3. DISTURBANCE VECTOR

The disturbance vector is represented by Equation 3.24 and contains two gust inputs channels: one for the gust velocity at the nose of the aircraft, and one for the gust acceleration, also defined at the nose. Gust modeling is discussed in Section 3.11. The order of the disturbance vector is 2, which is important to be kept on the low side for modern control design.

$$\boldsymbol{w} = \begin{bmatrix} U_{g,0} \\ \dot{U}_{g,0} \end{bmatrix} \tag{3.24}$$

3.1.4. OUTPUT VECTOR

The output vector is defined as Equation 3.25. The output vector contains scalar quantities such as the freestream velocity, angle of attack, sideslip angle, and flight path angle. Additionally, it contains the load factor in the body frame, body angular rates, aircraft position and orientation, as well as the angular and linear accelerations at the sensor positions, and the cutting forces and moments at the monitoring stations.

$$\boldsymbol{y} = \begin{bmatrix} V_\infty \\ \alpha \\ \beta \\ \gamma \\ \boldsymbol{n}_b \\ \boldsymbol{P} \\ \boldsymbol{\Theta} \\ \boldsymbol{\omega} \\ \boldsymbol{L}_m \\ \boldsymbol{a}_t \\ \boldsymbol{a}_r \end{bmatrix} \tag{3.25}$$

3.1.5. STATE EQUATION

The state equations realized in Section 3.13 can be cast into one nonlinear vector state equation in descriptor form given by Equation 3.26, with $\boldsymbol{E}$ the descriptor/mass matrix, $\boldsymbol{A}$ the state matrix, $\boldsymbol{B}$ the input matrix, and $\boldsymbol{F}$ the disturbance/gust matrix. Additionally, $\boldsymbol{f}(\boldsymbol{x})$ contains the nonlinear terms, and $\boldsymbol{f}_0$ contains the state bias terms. Note that the terms inside the matrices are constant, as they are evaluated at the reference flight condition. Additionally, due to the linearized nature of the boundary conditions, the effect of camber and twist is added separately to the model, which brings about the state bias term.

$$\boldsymbol{E}\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{F}\boldsymbol{w} + \boldsymbol{f}_0 \tag{3.26}$$

$$\boldsymbol{E} = \begin{bmatrix}
\boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \left(m\boldsymbol{I} - \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{V}}}\right) & -\frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\omega}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\eta}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\delta}}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & -\frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{V}}} & \left(\boldsymbol{J} - \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\omega}}}\right) & \boldsymbol{0} & -\frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\eta}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\delta}}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & -\frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{V}}} & -\frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\omega}}} & \boldsymbol{0} & \left(\boldsymbol{M}_{hh} - \frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\eta}}}\right) & \boldsymbol{0} & -\frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\delta}}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & -\frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{V}}} & -\frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\omega}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{F}_l}{\partial \ddot{\boldsymbol{\eta}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{F}_l}{\partial \ddot{\boldsymbol{\delta}}} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & -\frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{V}}} & -\frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\omega}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{M}_l}{\partial \ddot{\boldsymbol{\eta}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{M}_l}{\partial \ddot{\boldsymbol{\delta}}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & -\frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{V}}} & -\frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\omega}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{P}_l}{\partial \ddot{\boldsymbol{\eta}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{P}_l}{\partial \ddot{\boldsymbol{\delta}}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & -\frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{V}}} & -\frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\omega}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{L}_l}{\partial \ddot{\boldsymbol{\eta}}} & \boldsymbol{0} & -\frac{\partial \boldsymbol{L}_l}{\partial \ddot{\boldsymbol{\delta}}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I}
\end{bmatrix} \tag{3.27}$$

$$\boldsymbol{A} = \begin{bmatrix}
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{V}} & \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\eta}} & \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{g}} & \boldsymbol{E}_F & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{V}} & \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\eta}} & \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{g}} & \boldsymbol{0} & \boldsymbol{E}_M & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{V}} & \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\omega}} & \left(\frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\eta}} - \boldsymbol{K}_{hh}\right) & \left(\frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\eta}}} - \boldsymbol{C}_{hh}\right) & \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{g}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E}_P & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & -\boldsymbol{K}_a & -\boldsymbol{D}_a & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{A}_g & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\eta}}} & \boldsymbol{0} & \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{F}_l}{\partial \boldsymbol{g}} & \boldsymbol{B}_F & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\eta}}} & \boldsymbol{0} & \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{M}_l}{\partial \boldsymbol{g}} & \boldsymbol{0} & \boldsymbol{B}_M & \boldsymbol{0} & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\eta}}} & \boldsymbol{0} & \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{P}_l}{\partial \boldsymbol{g}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_P & \boldsymbol{0} \\
\boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\eta}}} & \boldsymbol{0} & \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{L}_l}{\partial \boldsymbol{g}} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{B}_L
\end{bmatrix} \tag{3.28}$$

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} \boldsymbol{C}(\boldsymbol{\Theta})^T \boldsymbol{V} \\ \boldsymbol{E}(\boldsymbol{\Theta})^{-1} \boldsymbol{\omega} \\ -m\left(\boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})\right) \\ -\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.29}$$

$$\boldsymbol{B} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{u}_t} \\ \boldsymbol{0} & \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{u}_t} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{u}_t} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{K}_a & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{3.30}$$

$$\boldsymbol{F} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{w}} \\ \boldsymbol{0} \\ \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{w}} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{B}_g \\ \frac{\partial \boldsymbol{F}_l}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{M}_l}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{P}_l}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{L}_l}{\partial \boldsymbol{w}} \end{bmatrix} \tag{3.31}$$

$$\boldsymbol{f}_0 = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{F}_C \\ \boldsymbol{M}_C \\ \boldsymbol{0} \\ \boldsymbol{P}_C \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.32}$$

3.1.6. OUTPUT EQUATION

The output equations define all quantities of interest that are not to be integrated over time but rather derived from the states, inputs, and disturbances and used for feedback. The output equations realized in Section 3.13 can be cast into the nonlinear output equation of Equation 3.33, with $\boldsymbol{C}$ the output matrix, $\boldsymbol{D}$ the input feedthrough matrix, $\boldsymbol{G}$ the disturbance feedthrough matrix, the nonlinear terms lumped into $\boldsymbol{h}(\boldsymbol{x})$, and the output bias terms lumped into $\boldsymbol{h}_0$. This equation has a slightly unconventional form, as there is also a $\boldsymbol{H}$ matrix, which can be viewed as the output mass matrix. It is chosen to specifically not back-substitute the state equation into the state derivative here, as this requires explicit inversion of the state descriptor matrix, which is often badly conditioned.

$$\boldsymbol{y} = \boldsymbol{h}(\boldsymbol{x}) + \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} + \boldsymbol{G}\boldsymbol{w} + \boldsymbol{H}\dot{\boldsymbol{x}} + \boldsymbol{h}_0 \tag{3.33}$$

$$\boldsymbol{D} = \begin{bmatrix} \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{0} \end{bmatrix} \tag{3.34}$$

$$\boldsymbol{G} = \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{w}} \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.35}$$

$$\boldsymbol{h}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{0} \\ \boldsymbol{L}_C \\ \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix} \tag{3.36}$$

$$\boldsymbol{C} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{V}} & \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\omega}} & \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\eta}} & \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\delta}} & \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{g}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{E}_L \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.37}$$

$$\boldsymbol{H} = \begin{bmatrix} \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{1}{g}\boldsymbol{I} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \left(\frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{V}}} - \boldsymbol{M}_{mt}\right) & \left(\frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\omega}}} - \boldsymbol{M}_{mr}\right) & \mathbf{0} & \left(\frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\eta}}} - \boldsymbol{M}_{mh}\right) & \mathbf{0} & \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\delta}}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{R} & \mathbf{0} & \mathbf{0} & \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \boldsymbol{R} & \mathbf{0} & \frac{\partial \boldsymbol{a}_r}{\partial \boldsymbol{\eta}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix} \tag{3.38}$$

$$\boldsymbol{h}(\boldsymbol{x}) = \begin{bmatrix} \sqrt{U^2 + V^2 + W^2} \\ \arctan\left(\frac{W}{U}\right) \\ \arctan\left(\frac{V}{\sqrt{U^2+W^2}}\right) \\ \theta - \arctan\left(\frac{W}{U}\right) \\ \frac{\boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})}{g} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ -\boldsymbol{M}_{mt}\left(\boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})\right) - \boldsymbol{M}_{mr}\boldsymbol{J}^{-1}\boldsymbol{\omega} \times (\boldsymbol{J}\boldsymbol{\omega}) \\ \boldsymbol{R}(\boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})) + 2\boldsymbol{S}(\boldsymbol{\omega})\frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} + \boldsymbol{S}(\dot{\boldsymbol{\omega}})\left(\boldsymbol{r}_{bs} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\boldsymbol{\eta}\right) + \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{S}(\boldsymbol{\omega})\left(\boldsymbol{r}_{bs} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\boldsymbol{\eta}\right) \\ \mathbf{0} \end{bmatrix} \tag{3.39}$$

3.2. MODEL PROCESSING & INPUTS

The model processing sequence is presented in Figure 3.1. This figure does not show the model inputs required into each of the computational steps, but rather the sequence in which the model is constructed in the processing script. The actual derivation fo each of these terms is discussed throughout this chapter.

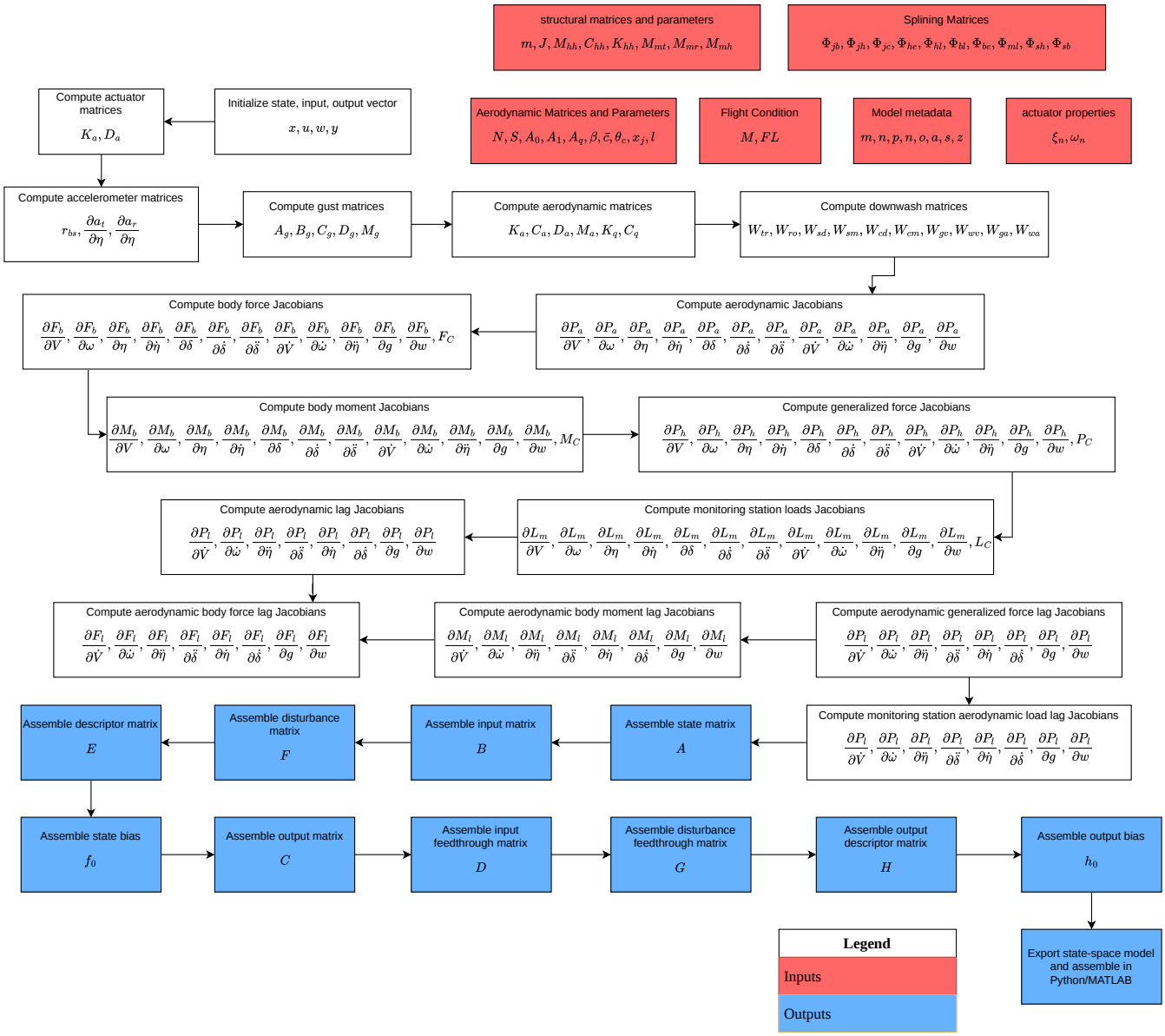


Figure 3.1: State-space model processing sequence flowchart

3.3. EQUATIONS OF MOTION

The equations of motion are based on mean-body axes. The derivation of the equations of motion based on mean-body axes is provided by Waszak and Schmidt [7]. Equation 3.40 describes the translational rigid-body motion, Equation 3.41 describes the rotational rigid-body motion, and lastly, Equation 3.42 describes the aeroelastic equation. A major objective of this chapter is to couple these equations by expressing the externally applied forces in terms of the states, inputs and disturbances to form a state-space model.

$$m\dot{\boldsymbol{V}} = -m\boldsymbol{\omega} \times \boldsymbol{V} + \boldsymbol{F}_b \tag{3.40}$$

$$\boldsymbol{J}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \boldsymbol{M}_b \tag{3.41}$$

$$\boldsymbol{M}_{ff}\ddot{\boldsymbol{x}}_f + \boldsymbol{C}_{ff}\dot{\boldsymbol{x}}_f + \boldsymbol{K}_{ff}\boldsymbol{x}_f = \boldsymbol{P}_f \tag{3.42}$$

3.4. MODAL ANALYSIS

The mass and stiffness matrices are presupposed, as well as the eigenmodes constituting the modal matrix. Equation 3.42 describes the aeroelastic dynamics in terms of the free DOFs of the finite-element model. However, it is rather convenient to express this equation using modal analysis to reduce the model order as well as make the system more interpretable. Using the modal matrix the free DOFs can be expressed as a linear combination of the mode shapes, or eigenmodes, which gives:

$$\boldsymbol{x}_f = \boldsymbol{\Phi}_{fh}\boldsymbol{\eta} \tag{3.43}$$

$$\dot{\boldsymbol{x}}_f = \boldsymbol{\Phi}_{fh}\dot{\boldsymbol{\eta}} \tag{3.44}$$

$$\ddot{\boldsymbol{x}}_f = \boldsymbol{\Phi}_{fh}\ddot{\boldsymbol{\eta}} \tag{3.45}$$

Accordingly, the modal mass, stiffness, and damping matrices are defined by right-multiplying with the modal matrix and left-multiplying with the transpose of the modal matrix:

$$\boldsymbol{K}_{hh} = \boldsymbol{\Phi}_{fh}^T \boldsymbol{K}_{ff}\boldsymbol{\Phi}_{fh} \tag{3.46}$$

$$\boldsymbol{C}_{hh} = \boldsymbol{\Phi}_{fh}^T \boldsymbol{C}_{ff}\boldsymbol{\Phi}_{fh} \tag{3.47}$$

$$\boldsymbol{M}_{hh} = \boldsymbol{\Phi}_{fh}^T \boldsymbol{M}_{ff}\boldsymbol{\Phi}_{fh} \tag{3.48}$$

Substituting Equation 3.43, Equation 3.44, and Equation 3.44 into the aeroelastic equation yields Equation 3.49. Subsequently, left-multiplying the resulting equation with the transpose of the modal matrix and recognizing the modal mass of Equation 3.48, modal stiffness of Equation 3.48, and modal damping of Equation 3.48 then gives the modal representation of the structural dynamics with Equation 3.51.

$$\boldsymbol{M}_{ff}\boldsymbol{\Phi}_{fh}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_{ff}\boldsymbol{\Phi}_{fh}\dot{\boldsymbol{\eta}} + \boldsymbol{K}_{ff}\boldsymbol{\Phi}_{fh}\dot{\boldsymbol{\eta}} = \boldsymbol{P}_f \tag{3.49}$$

$$\boldsymbol{\Phi}_{fh}^T\boldsymbol{M}_{hh}\boldsymbol{\Phi}_{fh}\ddot{\boldsymbol{\eta}} + \boldsymbol{\Phi}_{fh}^T\boldsymbol{C}_{hh}\boldsymbol{\Phi}_{fh}\dot{\boldsymbol{\eta}} + \boldsymbol{\Phi}_{fh}^T\boldsymbol{K}_{hh}\boldsymbol{\Phi}_{fh}\dot{\boldsymbol{\eta}} = \boldsymbol{\Phi}_{fh}^T\boldsymbol{P}_f = \boldsymbol{P}_h \tag{3.50}$$

$$\boldsymbol{M}_{hh}\ddot{\boldsymbol{\eta}} + \boldsymbol{C}_{hh}\dot{\boldsymbol{\eta}} + \boldsymbol{K}_{hh}\boldsymbol{\eta} = \boldsymbol{P}_h \tag{3.51}$$

Lastly, this equation can be cast in state-space representation as follows:

$$\dot{\boldsymbol{\eta}} = \dot{\boldsymbol{\eta}} \tag{3.52}$$

$$\ddot{\boldsymbol{\eta}} = \boldsymbol{M}_{hh}^{-1}\left(-\boldsymbol{C}_{hh}\dot{\boldsymbol{\eta}} - \boldsymbol{K}_{hh}\boldsymbol{\eta} + \boldsymbol{P}_h\right) \tag{3.53}$$

3.5. PANEL AERODYNAMICS

The aerodynamic model uses linear panel aerodynamics. For the steady aerodynamics, the Vortex Lattice Method (VLM) is used, of which the required equations to define the state-space model are presented in Subsection 3.5.1. For the unsteady aerodynamics, the Doublet Lattice Method (DLM) is used, of which the required mathematical formulation and equations are detailed in Subsection 3.5.2.

3.5.1. STEADY AERODYNAMICS

The Vortex Lattice Method (VLM) is used for quasi-steady aerodynamics. Using the Biot-Savart law the aerodynamic influences of each panel on one another can be obtained and assembled into an AIC matrix, which relates the panel downwash at the collocation point to the pressure coefficient differential at the quarter-chord points, as shown in Equation 3.54. In this thesis, the implementation by Voß [20] is used.

$$\Delta\boldsymbol{c}_{p_l} = \boldsymbol{AIC}\boldsymbol{w}_j \tag{3.54}$$

3.5.2. UNSTEADY AERODYNAMICS

For unsteady aerodynamics, the Doublet Lattice Method (DLM) is used. The result is similar to that when using VLM, but the resulting AIC matrix is complex-valued and only valid at the reduced frequency at which it was evaluated, as shown by Equation 3.55. Also for DLM, the implementation by Voß [20] is used.

$$\Delta\boldsymbol{c}_{p_l} = \boldsymbol{AIC}(k)\boldsymbol{w}_j \tag{3.55}$$

The reduced frequency in the formulation by Voß [20] uses the NASTRAN definition, with $\bar{c}$ being the reference chord length, most often taken as the mean aerodynamic chord of the wing. This reduced frequency equals:

$$k = \frac{\omega \bar{c}}{2V_\infty} \tag{3.56}$$

To form a state-space model, the unsteady aerodynamics should be expressed in the continuous-time rather than the discrete-frequency domain. The Rational Function Approximation (RFA) allows to convert the discrete frequency-domain description of the set of AIC matrices into a continuous-frequency domain description, which can be easily converted to the continuous-time domain by means of an inverse Laplace transform. Specifically, Roger's approximation is used [63] as the RFA. Physical RFA is used, meaning that the RFA is performed on the physical AIC matrices, as opposed to the generalized AIC matrices, as this allows separation of quasi-steady and unsteady aerodynamics, and therefore enables using the AIC matrix from VLM for the quasi-steady aerodynamics [39] and DLM for the unsteady aerodynamics, which is more accurate as the steady AIC matrix from DLM uses numerical approximation, whereas the steady AIC matrix from VLM is exact. Additionally, using physical RFA also allows for more accurate loads recovery [39], as is discussed in Section 3.12. Roger's approximation takes the following form:

$$\boldsymbol{AIC}(k) = \boldsymbol{A}_0 + jk\boldsymbol{A}_1 - k^2\boldsymbol{A}_2 + \sum_{q=1}^{p} \boldsymbol{A}_{q+2}\frac{jk}{jk+\beta_q} \tag{3.57}$$

The poles of Roger's approximation are computed as the maximum of the reduced frequency k divided by the pole number m. The poles can be seen as similar to knots in a spline, or the number of terms that the approximation can use to improve the fit. As usual with fitting data to basis functions, care must be taken to not overfit the approximation but also not to underfit it, which can be controlled with the number of poles.

$$\beta_q = \frac{k_{max}}{q} \tag{3.58}$$

The reduced frequency complex variable can be rewritten in terms of the Laplace variable s, which simplifies converting the expression to the time-domain later on:

$$jk = j\left(\frac{\bar{c}\omega}{2V_\infty}\right) = \left(\frac{\bar{c}}{2V_\infty}\right) j\omega = \left(\frac{\bar{c}}{2V_\infty}\right) s \tag{3.59}$$

Substituting the result into Equation 3.57 and neglecting the acceleration term yields the AIC expression in terms of the Laplace variable.

$$\boldsymbol{AIC}(s) = \boldsymbol{A}_0 + \left(\frac{\bar{c}}{2V_\infty}\right)\boldsymbol{A}_1 s + \sum_{q=1}^{p} \boldsymbol{A}_{q+2}\frac{\left(\frac{\bar{c}}{2V_\infty}\right)s}{\left(\frac{\bar{c}}{2V_\infty}\right)s+\beta_q} \tag{3.60}$$

Multiplying this expression by the downwash at the collocation point $\boldsymbol{w}_j$ yields the pressure coefficient acting on each panel:

$$\Delta\bar{\boldsymbol{c}}_{p_l} = \boldsymbol{A}_0\boldsymbol{w}_j + \left(\frac{\bar{c}}{2V_\infty}\right)\boldsymbol{A}_1 s\boldsymbol{w}_j + \sum_{q=1}^{p} \boldsymbol{A}_{q+2}\frac{\left(\frac{\bar{c}}{2V_\infty}\right)s}{\left(\frac{\bar{c}}{2V_\infty}\right)s+\beta_q}\boldsymbol{w}_j \tag{3.61}$$

This expression as is cannot be converted into the time-domain due to the rational term. Therefore, an aerodynamic lag vector for each pole $\boldsymbol{l}_q = \begin{bmatrix} l_1 & l_2 & \cdots & l_p \end{bmatrix}^T$ is introduced, which describes the aerodynamic lag states associated with each pole number m:

$$\boldsymbol{l}_q = \frac{\left(\frac{\bar{c}}{2V_\infty}\right)s}{\left(\frac{\bar{c}}{2V_\infty}\right)s+\beta_q}\boldsymbol{w} = \frac{s}{s+\left(\frac{2V_\infty}{\bar{c}}\right)\beta_q}\boldsymbol{w} \tag{3.62}$$

Right-multiplying with the Laplace variable yields:

$$\boldsymbol{l}_q s = \frac{s^2}{s+\left(\frac{2V_\infty}{\bar{c}}\right)\beta_q}\boldsymbol{w} = \frac{s\left(s+\left(\frac{2V_\infty}{\bar{c}}\right)\beta_q\right) - s\left(\frac{2V_\infty}{\bar{c}}\right)\beta_q}{s+\left(\frac{2V_\infty}{\bar{c}}\right)\beta_q}\boldsymbol{w} = s\boldsymbol{w} - \left(\frac{2V_\infty}{\bar{c}}\right)\beta_q\boldsymbol{l} \tag{3.63}$$

Converting Equation 3.63 to the time-domain yields the lag dynamics associated with each pole:

$$\dot{\boldsymbol{l}}_m = \dot{\boldsymbol{w}}_j - \left(\frac{2V_\infty}{\bar{c}}\right)\beta_q \boldsymbol{l}_q \tag{3.64}$$

Substituting Equation 3.62 into Equation 3.60 yields the AIC expressed in terms of the lag vectors per pole.

$$\Delta \boldsymbol{c}_{p_l} = \boldsymbol{A}_0 \boldsymbol{w}_j + \left(\frac{\bar{c}}{2V_\infty}\right)\boldsymbol{A}_1 s\boldsymbol{w}_j + \sum_{q=1}^{p} \boldsymbol{A}_{q+2}\boldsymbol{l}_q \tag{3.65}$$

This equation can subsequently be converted into the time-domain, which yields the following equation relating the pressure coefficient differential to the panel downwash, panel downwash derivative and the aerodynamic lag states. It is important to note that the freestream velocity is assumed constant at the reference flight condition, as making it a function of the body velocity introduces the following issues. First, making the freestream velocity a function of the system states complicates computing the inverse Laplace transform when converting to the continuous-time domain. Secondly, the expression for the downwash derivative becomes nonlinear as well, reducing the model interpretability as the governing equations become more intricate. Thirdly, the AIC matrices are only valid for the Mach number at which they were evaluated. As air density and Mach number determine the freestream velocity, it would be inconsistent to make the freestream velocity variable while keeping the Mach number fixed. Lastly, as an important objective is to keep the model explicit and interpretable, keeping the freestream velocity as a parameter retains the linearity of the equations. If one however desires a model that is continuously valid for a range of flight speeds, it would be a more consistent approach to build a parameter-varying model for various flight speeds instead. One could argue, that for slowly-varying flight speeds the error would be minimal and would be offset by the improved accuracy of using the actual freestream velocity, but for the sake of mathematical correctness and the above reasons the freestream velocity is taken constant at the reference flight condition. This implies that a new aerodynamic model has to be built for each desired flight condition, determined by the flight speed and air density.

$$\Delta \boldsymbol{c}_{p_l} = \boldsymbol{A}_0 \boldsymbol{w}_j + \left(\frac{\bar{c}}{2V_\infty}\right)\boldsymbol{A}_1 \dot{\boldsymbol{w}}_j + \sum_{q=1}^{p} \boldsymbol{A}_{q+2}\boldsymbol{l}_q \tag{3.66}$$

The lag vector per pole at the current step in the derivation has the same size as the number of aerodynamic panels n, leading to $n \times p$ lag states, with p the number of poles. This is not in line with the requirements for an aeroservoelastic model as determined in Chapter 2 and moreover, as will be shown, can be described using less states while retaining the same input-output behaviour. This solution is addressed, however, in Subsection 3.6.2.

3.6. AERODYNAMIC COUPLING

Equation 3.66 relates the pressure coefficient differential, panel downwash, panel downwash derivative, and the aerodynamic lag states. However, the objective is to express these in terms of the states, inputs, and disturbances of the state-space model, which requires several transformations. First, the pressure coefficient differential is multiplied by the dynamic pressure and the panel area matrix to obtain the aerodynamic force normal to each panel. In the resulting aerodynamic force expression, quasi-steady aerodynamic forces are represented by the first term, added mass aerodynamic forces are represented by the second term, and aerodynamic lag forces are represented by the last term:

$$\boldsymbol{F}_a = \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\Delta\bar{\boldsymbol{c}}_{p_j} = \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{w}_j + \frac{1}{4}\rho c V_\infty \boldsymbol{S}\boldsymbol{A}_1\dot{\boldsymbol{w}}_j + \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\sum_{q=3}^{p+2}\boldsymbol{A}_q\boldsymbol{l}_q \tag{3.67}$$

The final result should be the aerodynamic force expressed as body forces, body moments, and generalized forces in terms of the states, inputs, and disturbances of the state-space model. First, the quasi-steady and added mass aerodynamic forces are expressed as body forces, body moments and generalized forces, which is discussed in Subsection 3.6.1. Secondly, the aerodynamic lag forces require separate treatment, as the order of the aerodynamic lag vectors is reduced through projection, which is explained in Subsection 3.6.2.

3.6.1. QUASI-STEADY AND ADDED MASS FORCES

To express the panel downwash in terms of the model states, control inputs, and external disturbances, we distinguish between displacement-induced downwash as well as motion-induced downwash, both of which affect the downwash differently. This approach implicitly assumes that there is no coupling between the

displacements $\boldsymbol{\eta}, \boldsymbol{\delta}, \boldsymbol{\theta}_c$ and motions $\boldsymbol{V}, \boldsymbol{\omega}, \dot{\boldsymbol{\eta}}, \dot{\boldsymbol{\delta}}$, and accelerations $\dot{\boldsymbol{V}}, \dot{\boldsymbol{\omega}}, \ddot{\boldsymbol{\eta}}, \ddot{\boldsymbol{\delta}}$, keeping the panel aerodynamics linear. An alternative method of expressing the panel downwash and its derivative is possible using the dot product of the panel normal and induced velocity at each panel, but is not adopted in this derivation for reasons discussed in Section 6.2. This section first discusses the motion-induced downwash, followed by the displacement-induced downwash, and finishes with the downwash due to camber and gusts, which are treated separately.

DOWNWASH: MOTION-INDUCED

Motion-induced downwash is due to linear rigid-body velocity $\boldsymbol{V}$, angular rigid-body velocity $\boldsymbol{\omega}$, control surface motion $\dot{\boldsymbol{\delta}}$, and structural/flexible motion $\dot{\boldsymbol{\eta}}$. The induced velocities at the collocation point j can be expressed as a function of these states through the relevant splining matrix. As the splining matrices map 6 DOFs to 6 DOFs, but for motion-induced downwash, only the linear velocities are used, the selection matrix $\boldsymbol{E}_v$ is introduced, which extracts only the linear velocities from the relevant splining matrix. This selection matrix is computed using the following Kronecker product with n being the number of aerodynamic panels:

$$\boldsymbol{E}_v = \boldsymbol{I}_{n\times n} \otimes \begin{bmatrix} \boldsymbol{I}_{3x3} & \boldsymbol{0}_{3x3} \end{bmatrix} \tag{3.68}$$

The induced velocities due to flexible motion is obtained using the splining matrix $\boldsymbol{\Phi}_{jh}$ mapping generalized coordinates to the collocation points, and subsequent selection of the linear velocities:

$$\boldsymbol{v}_j = \boldsymbol{E}_v \boldsymbol{\Phi}_{jh} \dot{\boldsymbol{\eta}} \tag{3.69}$$

The induced velocities at the collocation points due to rigid-body motion is expressed similarly. The splining matrix $\boldsymbol{\Phi}_{jb}$ which maps rigid-body degrees of freedom to the aerodynamic collocation points, is split into two matrices in order to express the induced velocities due to the state vectors $\boldsymbol{V}$ and $\boldsymbol{\omega}$ individually, followed by selection of the linear velocities using the selection matrix:

$$\boldsymbol{v}_j = \boldsymbol{E}_v \boldsymbol{\Phi}_{jb} \begin{pmatrix} \boldsymbol{V} \\ \boldsymbol{\omega} \end{pmatrix} = \boldsymbol{E}_v \begin{bmatrix} \boldsymbol{\Phi}_{jt} & \boldsymbol{\Phi}_{jr} \end{bmatrix} \begin{pmatrix} \boldsymbol{V} \\ \boldsymbol{\omega} \end{pmatrix} = \boldsymbol{E}_v \boldsymbol{\Phi}_{jt} \boldsymbol{V} + \boldsymbol{E}_v \boldsymbol{\Phi}_{jr} \boldsymbol{\omega} \tag{3.70}$$

Lastly, the induced velocities at the collocation points due to control surface motion is expressed using the splining matrix $\boldsymbol{\Phi}_{jc}$ and also selection of the linear velocities:

$$\boldsymbol{v}_j = \boldsymbol{E}_v \boldsymbol{\Phi}_{jc} \dot{\boldsymbol{\delta}} \tag{3.71}$$

The motion-induced downwash at the collocation point j of each aerodynamic panel can then be computed as the dot product of each panel normal vector and the induced velocities at the collocation points. Additionally, it must be divided by the freestream velocity in order to non-dimensionalize the result, and therefore allow multiplication with the dynamic pressure later in the derivation, when converting the pressure coefficient differential to an aerodynamic force. Moreover, they are negated to make the downwash have the correct sign, as the downwash is opposite to the induced motion. Mathematically, the panel downwash at the collocation points can be expressed as:

$$\boldsymbol{w}_j = -\frac{1}{V_\infty} \begin{bmatrix} \boldsymbol{n}_0^T & \boldsymbol{0}_{1\times3} & \cdots & \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{1\times3} & \boldsymbol{n}_1^T & \cdots & \boldsymbol{0}_{1\times3} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \cdots & \boldsymbol{n}_n^T \end{bmatrix} \begin{pmatrix} \boldsymbol{v}_0 \\ \boldsymbol{v}_1 \\ \vdots \\ \boldsymbol{v}_n \end{pmatrix} = -\frac{1}{V_\infty} \boldsymbol{N}_j \boldsymbol{v}_j \tag{3.72}$$

The motion-induced downwash can now be fully expressed, by summing all contributions:

$$\boldsymbol{w}_j = -\boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jh} \left(\frac{\dot{\boldsymbol{\eta}}}{V_\infty} \right) - \boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jt} \left(\frac{\boldsymbol{V}}{V_\infty} \right) - \boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jr} \left(\frac{\boldsymbol{\omega}}{V_\infty} \right) - \boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jc} \left(\frac{\dot{\boldsymbol{\delta}}}{V_\infty} \right) \tag{3.73}$$

However, this can be further simplified as all matrix multiplications can be lumped into matrices referred to here as the downwash matrices, which relate system states to the induced panel downwash. $\boldsymbol{W}_{tr}$ is the definition of the downwash matrix for translational body motion, $\boldsymbol{W}_{ro}$ is the downwash matrix for rotational body motion, $\boldsymbol{W}_{sm}$ is the downwash matrix for structural/flexible motion, and $\boldsymbol{W}_{cm}$ is the downwash matrix for control surface deflection.

$$\boldsymbol{W}_{tr} = -\boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jt} \tag{3.74}$$

$$\boldsymbol{W}_{ro} = -\boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jr} \tag{3.75}$$

$$\boldsymbol{W}_{sm} = -\boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jh} \tag{3.76}$$

$$\boldsymbol{W}_{cm} = -\boldsymbol{N}_j \boldsymbol{E}_v \boldsymbol{\Phi}_{jc} \tag{3.77}$$

Finally, this yields the motion-induced panel downwash vector in a much simplified form:

$$\boldsymbol{w}_j = \boldsymbol{W}_{tr}\left(\frac{\boldsymbol{V}}{V_\infty}\right) + \boldsymbol{W}_{ro}\left(\frac{\boldsymbol{\omega}}{V_\infty}\right) + \boldsymbol{W}_{sm}\left(\frac{\dot{\boldsymbol{\eta}}}{V_\infty}\right) + \boldsymbol{W}_{cm}\left(\frac{\dot{\boldsymbol{\delta}}}{V_\infty}\right) \tag{3.78}$$

DOWNWASH: DISPLACEMENT-INDUCED

Displacement-induced downwash, on the contrary, is due to structural/flexible deformation $\boldsymbol{\eta}$ and control surface deflection $\boldsymbol{\delta}$. A similar approach is adopted here, with one major difference. For the displacement-induced downwash, the induced rotations in the local panel y-direction is what creates downwash, as this alters the local angle of attack. Therefore, instead of taking the dot product of the normal vector, the dot product of the tangential vector is computed instead. This tangential vector is defined as the cross product of the normal vector and the unit vector in the x-direction, resulting in a vector that can be used to project the rotational components in the correct direction:

$$\boldsymbol{t}_i = \boldsymbol{n}_i \times \boldsymbol{i} \tag{3.79}$$

As opposed to motion-induced downwash, for the displacement-induced downwash the induced rotations are required. Therefore, the selection matrix $\boldsymbol{E}_\theta$ is introduced that selects the induced rotations from the relevant splining matrix. This selection matrix is computed using the following Kronecker product with n being the number of aerodynamic panels:

$$\boldsymbol{E}_\theta = \boldsymbol{I}_{n\times n} \otimes \begin{bmatrix} \boldsymbol{0}_{3x3} & \boldsymbol{I}_{3x3} \end{bmatrix} \tag{3.80}$$

The induced rotations due the flexible deformation is computed using the splining matrix $\boldsymbol{\Phi}_{jh}$ and selection of the angular displacements:

$$\boldsymbol{\theta}_j = \boldsymbol{E}_\theta \boldsymbol{\Phi}_{jh} \boldsymbol{\eta} \tag{3.81}$$

The induced rotations due to the control surface deflection is evaluated using the splining matrix $\boldsymbol{\Phi}_{jc}$, and also selecting the angular displacements:

$$\boldsymbol{\theta}_j = \boldsymbol{E}_\theta \boldsymbol{\Phi}_{jc} \boldsymbol{\delta} \tag{3.82}$$

The displacement-induced downwash at the collocation point j of each aerodynamic panel can be computed as the dot product of the tangential vector and the induced rotation at each collocation point, which can be expressed mathematically as follows:

$$\boldsymbol{w}_j = \begin{bmatrix} \boldsymbol{t}_0^T & \boldsymbol{0}_{1\times3} & \cdots & \boldsymbol{0}_{1\times3} \\ \boldsymbol{0}_{1\times3} & \boldsymbol{t}_1^T & \cdots & \boldsymbol{0}_{1\times3} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_{1\times3} & \boldsymbol{0}_{1\times3} & \cdots & \boldsymbol{t}_n^T \end{bmatrix} \begin{pmatrix} \boldsymbol{\theta}_0 \\ \boldsymbol{\theta}_1 \\ \vdots \\ \boldsymbol{\theta}_n \end{pmatrix} = \boldsymbol{T}_j \boldsymbol{\theta}_j \tag{3.83}$$

The displacement-induced downwash can now be fully expressed, by summing the two contributions:

$$\boldsymbol{w}_j = \boldsymbol{T}_j \boldsymbol{E}_\theta \boldsymbol{\Phi}_{jc} \boldsymbol{\delta} + \boldsymbol{T}_j \boldsymbol{E}_\theta \boldsymbol{\Phi}_{jh} \boldsymbol{\eta} \tag{3.84}$$

Again, the matrix multiplication can be lumped into downwash matrices with $\boldsymbol{W}_{cd}$ being the downwash matrix for control surface deflection and $\boldsymbol{W}_{sd}$ being the downwash matrix for structural/flexible deformation:

$$\boldsymbol{W}_{cd} = \boldsymbol{T}_j \boldsymbol{E}_\theta \boldsymbol{\Phi}_{jc} \tag{3.85}$$

$$\boldsymbol{W}_{sd} = \boldsymbol{T}_j \boldsymbol{E}_\theta \boldsymbol{\Phi}_{jh} \tag{3.86}$$

Lastly, this then gives the simplified form of the displacement-induced downwash:

$$\boldsymbol{w}_j = \boldsymbol{W}_{cd} \boldsymbol{\delta} + \boldsymbol{W}_{sd} \boldsymbol{\eta} \tag{3.87}$$

DOWNWASH: CAMBER AND GUSTS

Additional downwash components must also be considered that are not related to the dynamics of the aircraft. These downwash components are the downwash due to camber and twist distribution and the downwash due to gusts. The downwash due to camber and twist is defined in a NASTRAN card called DMI W2GJ, which specifies an angle in radians at the half-chord of each aerodynamic panel along the local panel y-direction. A more practical example of how this camber and twist distribution is computed is given in Subsection 4.3.7. As the small angle approximation is applied for the camber and twist distribution, $\boldsymbol{\theta}_c$, the vector containing the rotation angle around the y-direction of each panel at the half-chord, as specified by the DMI W2GJ card, directly acts as the downwash at the collocation point of each panel:

$$\boldsymbol{w}_c = \boldsymbol{\theta}_c \tag{3.88}$$

The downwash due to gusts is handled slightly differently. Gusts require special consideration as the downwash induced at each panel varies as the aircraft penetrates the gust field. Section 3.11 is dedicated to gust modeling and accounting for this gust penetration effect. The result of that section is a linear state-space model that augments the free-flying aeroservoelastic state-space model. The output equations from this state-space model constitute two equations: Equation 3.89 relating the panel downwash to the gust states and the gust inputs and Equation 3.90 relating the panel downwash derivative also to the gust states and gust inputs. In the panel downwash equation, $\boldsymbol{W}_{gv}$ can be identified as relating the gust states to the gust downwash velocities, and $\boldsymbol{W}_{wv}$ as the downwash matrix relating the gust inputs to the gust downwash velocities. For the panel downwash derivative, $\boldsymbol{W}_{ga}$ can be identified as the downwash matrix relating the gust states to the gust downwash accelerations, and $\boldsymbol{W}_{wa}$ the downwash matrix due to the gust inputs to the gust downwash accelerations.

$$\boldsymbol{w}_g = \boldsymbol{GC}_1 \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \begin{bmatrix}\boldsymbol{GD}_1 & \boldsymbol{0}\end{bmatrix} \left(\frac{\boldsymbol{w}}{V_\infty}\right) = \boldsymbol{W}_{gv} \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \boldsymbol{W}_{wv} \left(\frac{\boldsymbol{w}}{V_\infty}\right) \tag{3.89}$$

$$\dot{\boldsymbol{w}}_g = \boldsymbol{GC}_1 \boldsymbol{A}_g \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \begin{bmatrix}\boldsymbol{GC}_1 \boldsymbol{B}_g & \boldsymbol{GD}_1\end{bmatrix} \left(\frac{\boldsymbol{w}}{V_\infty}\right) = \boldsymbol{W}_{ga} \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \boldsymbol{W}_{wa} \left(\frac{\boldsymbol{w}}{V_\infty}\right) \tag{3.90}$$

Summing the contributions for the panel downwash yields the following equation:

$$\boldsymbol{w}_j = \boldsymbol{\theta}_c + \boldsymbol{W}_{gv} \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \boldsymbol{W}_{wv} \left(\frac{\boldsymbol{w}}{V_\infty}\right) \tag{3.91}$$

Similarly, for the panel downwash derivative:

$$\dot{\boldsymbol{w}}_j = \dot{\boldsymbol{w}}_g = \boldsymbol{W}_{ga} \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \boldsymbol{W}_{wa} \left(\frac{\boldsymbol{w}}{V_\infty}\right) \tag{3.92}$$

PANEL DOWNWASH AND DERIVATIVE

Next, the panel downwash vector can be expressed by summing the motion-induced, displacement-induced, camber and twist, and gust contributions derived earlier:

$$\boldsymbol{w}_j = \boldsymbol{\theta}_c + \boldsymbol{W}_{tr} \left(\frac{\boldsymbol{V}}{V_\infty}\right) + \boldsymbol{W}_{ro} \left(\frac{\boldsymbol{\omega}}{V_\infty}\right) + \boldsymbol{W}_{cd}\boldsymbol{\delta} + \boldsymbol{W}_{cm} \left(\frac{\dot{\boldsymbol{\delta}}}{V_\infty}\right) + \boldsymbol{W}_{sd}\boldsymbol{\eta} + \boldsymbol{W}_{sm} \left(\frac{\dot{\boldsymbol{\eta}}}{V_\infty}\right) + \boldsymbol{W}_{gv} \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \boldsymbol{W}_{wv} \left(\frac{\boldsymbol{w}}{V_\infty}\right) \tag{3.93}$$

The derivative of the panel downwash vector can be obtained by taking the derivative of Equation 3.93, resulting in the expression below. The downwash matrices are constant, the freestream velocity is equal to speed at the reference flight condition, and the camber and twist is also constant, and thus need to be appropriately taken into account when computing the derivative.

$$\dot{\boldsymbol{w}}_j = \boldsymbol{W}_{tr} \left(\frac{\dot{\boldsymbol{V}}}{V_\infty}\right) + \boldsymbol{W}_{ro} \left(\frac{\dot{\boldsymbol{\omega}}}{V_\infty}\right) + \boldsymbol{W}_{cd}\dot{\boldsymbol{\delta}} + \boldsymbol{W}_{cm} \left(\frac{\ddot{\boldsymbol{\delta}}}{V_\infty}\right) + \boldsymbol{W}_{sd}\dot{\boldsymbol{\eta}} + \boldsymbol{W}_{sm} \left(\frac{\ddot{\boldsymbol{\eta}}}{V_\infty}\right) + \boldsymbol{W}_{ga} \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \boldsymbol{W}_{wa} \left(\frac{\boldsymbol{w}}{V_\infty}\right) \tag{3.94}$$

AERODYNAMIC FORCES

Substituting the downwash and downwash derivative into the first two terms of Equation 3.67 gives the following. Note that the lag terms are neglected as these are separately handled in Subsection 3.6.2.

$$\begin{aligned}\boldsymbol{F}_a = &\frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{\theta}_c + \frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{tr}\boldsymbol{V} + \frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{ro}\boldsymbol{\omega} + \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{cd}\boldsymbol{\delta} \\ &+\frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{cm}\dot{\boldsymbol{\delta}} + \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{sd}\boldsymbol{\eta} + \frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{sm}\dot{\boldsymbol{\eta}} + \frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{gv}\boldsymbol{g} \\ &+\frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_0\boldsymbol{W}_{wv}\boldsymbol{w} + \frac{1}{4}\rho c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{tr}\dot{\boldsymbol{V}} + \frac{1}{4}\rho c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{ro}\dot{\boldsymbol{\omega}} + \frac{1}{4}\rho V_\infty c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{cd}\dot{\boldsymbol{\delta}} \\ &+\frac{1}{4}\rho c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{cm}\ddot{\boldsymbol{\delta}} + \frac{1}{4}\rho V_\infty c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{sd}\dot{\boldsymbol{\eta}} + \frac{1}{4}\rho c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{sm}\ddot{\boldsymbol{\eta}} + \frac{1}{4}\rho c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{ga}\boldsymbol{g} \\ &+\frac{1}{4}\rho c\boldsymbol{S}\boldsymbol{A}_1\boldsymbol{W}_{wa}\boldsymbol{w}\end{aligned} \tag{3.95}$$

From the above expression stiffness-like, damping-like, and mass-like terms can be distinguished and lumped into an aerodynamic stiffness matrix $\boldsymbol{K}_a$, an aerodynamic quasi-steady damping matrix $\boldsymbol{C}_a$, an aerodynamic unsteady damping matrix $\boldsymbol{D}_a$, and an aerodynamic apparent mass matrix $\boldsymbol{M}_a$:

$$\boldsymbol{K}_a = \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{A}_0 \tag{3.96}$$

$$\boldsymbol{C}_a = \frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_0 \tag{3.97}$$

$$\boldsymbol{D}_a = \frac{1}{4}\rho V_\infty c\boldsymbol{S}\boldsymbol{A}_1 \tag{3.98}$$

$$\boldsymbol{M}_a = \frac{1}{4}\rho c\boldsymbol{S}\boldsymbol{A}_1 \tag{3.99}$$

Substituting these shorthand matrix expressions results in the following expression for the aerodynamic forces due to stiffness, damping and, apparent mass:

$$\begin{aligned}\boldsymbol{F}_a = &\boldsymbol{K}_a\boldsymbol{\theta}_c + \boldsymbol{C}_a\boldsymbol{W}_{tr}\boldsymbol{V} + \boldsymbol{C}_a\boldsymbol{W}_{ro}\boldsymbol{\omega} + \boldsymbol{K}_a\boldsymbol{W}_{cd}\boldsymbol{\delta} + \boldsymbol{C}_a\boldsymbol{W}_{cm}\dot{\boldsymbol{\delta}} \\ &+\boldsymbol{K}_a\boldsymbol{W}_{sd}\boldsymbol{\eta} + \boldsymbol{C}_a\boldsymbol{W}_{sm}\dot{\boldsymbol{\eta}} + \boldsymbol{C}_a\boldsymbol{W}_{gv}\boldsymbol{g} + \boldsymbol{C}_a\boldsymbol{W}_{wv}\boldsymbol{w} \\ &+\boldsymbol{M}_a\boldsymbol{W}_{tr}\dot{\boldsymbol{V}} + \boldsymbol{M}_a\boldsymbol{W}_{ro}\dot{\boldsymbol{\omega}} + \boldsymbol{D}_a\boldsymbol{W}_{cd}\dot{\boldsymbol{\delta}} + \boldsymbol{M}_a\boldsymbol{W}_{cm}\ddot{\boldsymbol{\delta}} \\ &+\boldsymbol{D}_a\boldsymbol{W}_{sd}\dot{\boldsymbol{\eta}} + \boldsymbol{M}_a\boldsymbol{W}_{sm}\ddot{\boldsymbol{\eta}} + \boldsymbol{M}_a\boldsymbol{W}_{ga}\boldsymbol{g} + \boldsymbol{M}_a\boldsymbol{W}_{wa}\boldsymbol{w}\end{aligned} \tag{3.100}$$

The input to these matrices is a downwash vector or the derivative thereof, and output the resulting aerodynamic force normal to the aerodynamic panel. However, in Equation 3.6.1 and Equation 3.6.1 the forces are interpolated to body forces/moments and generalized forces respectively, which is performed by left-multiplying with the relevant splining matrix. As again, the splining matrix maps 6 DOFs to 6 DOFs, the aerodynamic force matrices need to be decomposed along the panel normals and expanded to 6 DOFs, which is achieved using matrix $\boldsymbol{\Pi}$. The expanded aerodynamic matrices are subsequently denoted with a tilde.

$$\boldsymbol{\Pi} = \begin{bmatrix} \begin{bmatrix}\boldsymbol{n}_0 \\ \boldsymbol{0}_{3\times1}\end{bmatrix} & \boldsymbol{0}_{6\times1} & \cdots & \boldsymbol{0}_{6\times1} \\ \boldsymbol{0}_{6\times1} & \begin{bmatrix}\boldsymbol{n}_1 \\ \boldsymbol{0}_{3\times1}\end{bmatrix} & \cdots & \boldsymbol{0}_{6\times1} \\ \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{0}_{6\times1} & \boldsymbol{0}_{6\times1} & \cdots & \begin{bmatrix}\boldsymbol{n}_n \\ \boldsymbol{0}_{3\times1}\end{bmatrix} \end{bmatrix} \tag{3.101}$$

$$\tilde{\boldsymbol{K}}_a = \boldsymbol{\Pi}\boldsymbol{K}_a \tag{3.102}$$

$$\tilde{\boldsymbol{C}}_a = \boldsymbol{\Pi}\boldsymbol{C}_a \tag{3.103}$$

$$\tilde{\boldsymbol{D}}_a = \boldsymbol{\Pi}\boldsymbol{D}_a \tag{3.104}$$

$$\tilde{\boldsymbol{M}}_a = \boldsymbol{\Pi}\boldsymbol{M}_a \tag{3.105}$$

As this expression will be substituted several times, it is convenient to introduce shorthand notation for the premultiplying terms. The matrices multiplying each of the states can be identified as the Jacobians of the aerodynamic forces with respect to each of the states, inputs, and disturbances. The aerodynamic forces due to camber and twist are constant and is thus a bias term.

$$\begin{aligned}
&\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{V}} = \tilde{\boldsymbol{C}}_a \boldsymbol{W}_{tr} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\omega}} = \tilde{\boldsymbol{C}}_a \boldsymbol{W}_{ro} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\eta}} = \tilde{\boldsymbol{K}}_a \boldsymbol{W}_{sd} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\delta}} = \tilde{\boldsymbol{K}}_a \boldsymbol{W}_{cd} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{V}}} = \tilde{\boldsymbol{M}}_a \boldsymbol{W}_{tr} \\
&\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\omega}}} = \tilde{\boldsymbol{M}}_a \boldsymbol{W}_{ro} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\eta}}} = \tilde{\boldsymbol{C}}_a \boldsymbol{W}_{sm} + \tilde{\boldsymbol{D}}_a \boldsymbol{W}_{sd} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\delta}}} = \tilde{\boldsymbol{C}}_a \boldsymbol{W}_{cm} + \tilde{\boldsymbol{D}}_a \boldsymbol{W}_{cd} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\eta}}} = \tilde{\boldsymbol{M}}_a \boldsymbol{W}_{sm} \\
&\frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\delta}}} = \tilde{\boldsymbol{M}}_a \boldsymbol{W}_{cm} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{g}} = \tilde{\boldsymbol{C}}_a \boldsymbol{W}_{gv} + \tilde{\boldsymbol{M}}_a \boldsymbol{W}_{ga} \qquad \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{w}} = \tilde{\boldsymbol{C}}_a \boldsymbol{W}_{wv} + \tilde{\boldsymbol{M}}_a \boldsymbol{W}_{wa} \qquad \boldsymbol{P}_{a,c} = \tilde{\boldsymbol{K}}_a \boldsymbol{\theta}_c
\end{aligned} \tag{3.106}$$

Finally, this then yields the expression for the aerodynamic force expressed using 6 DOFs at the quarter-chord points of each aerodynamic panel, resulting in a vector of size $6n$:

$$\begin{aligned}
\boldsymbol{P}_a = &\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{V}} \boldsymbol{V} + \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\omega}} \boldsymbol{\omega} + \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\eta}} \boldsymbol{\eta} + \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\delta}} \boldsymbol{\delta} + \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{V}}} \dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\omega}}} \dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\eta}}} \dot{\boldsymbol{\eta}} \\
&+ \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\delta}}} \dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\eta}}} \ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\delta}}} \ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{g}} \boldsymbol{g} + \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{w}} \boldsymbol{w} + \boldsymbol{P}_{a,c}
\end{aligned} \tag{3.107}$$

BODY FORCES AND MOMENTS

To express the aerodynamic force in the body frame and in generalized coordinates, the splining matrices are used, which interpolate the aerodynamic forces and moments in the body frame and generalized coordinates. The aerodynamic forces act on the quarter-chord point of each aerodynamic panel defined at the aerodynamic l-grid. For the rigid-body forces and moments, the splining matrix $\boldsymbol{\Phi}_{bl}$ is used, and multiplied with selection matrix $\boldsymbol{E}_F$ to extract the force components:

$$\boldsymbol{E}_F = \begin{bmatrix} \boldsymbol{I}_{3\times 3} & \boldsymbol{0}_{3\times 3} \end{bmatrix} \tag{3.108}$$

$$\boldsymbol{F}_b = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \boldsymbol{P}_a \tag{3.109}$$

However, the states, inputs and disturbances should be abstracted out and therefore the Jacobians expressing the body forces with respect to the states, inputs, and disturbances are expressed as:

$$\begin{aligned}
&\frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{V}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{V}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\omega}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\omega}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\eta}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\eta}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\delta}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\delta}} \\
&\frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{V}}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{V}}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\omega}}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\omega}}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\eta}}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\eta}}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\delta}}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\delta}}} \\
&\frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\eta}}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\eta}}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\delta}}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\delta}}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{g}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{g}} \qquad \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{w}} = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{w}} \\
&\boldsymbol{F}_C = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \boldsymbol{P}_{a,c}
\end{aligned} \tag{3.110}$$

Lastly, this gives the full expression for the body forces in terms of the states, inputs, and disturbances:

$$\begin{aligned}
\boldsymbol{F}_b = &\frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{V}} \boldsymbol{V} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\omega}} \boldsymbol{\omega} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\eta}} \boldsymbol{\eta} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\delta}} \boldsymbol{\delta} + \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{V}}} \dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\omega}}} \dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\eta}}} \dot{\boldsymbol{\eta}} \\
&+ \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\delta}}} \dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\eta}}} \ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\delta}}} \ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{g}} \boldsymbol{g} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{w}} \boldsymbol{w} + \boldsymbol{F}_C
\end{aligned} \tag{3.111}$$

The body moments are expressed similarly, using the same splining matrix $\boldsymbol{\Phi}_{bl}$, but selecting the moment components using selection matrix $\boldsymbol{E}_M$:

$$\boldsymbol{E}_M = \begin{bmatrix} \boldsymbol{0}_{3\times 3} & \boldsymbol{I}_{3\times 3} \end{bmatrix} \tag{3.112}$$

$$\boldsymbol{M}_b = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \boldsymbol{P}_a \tag{3.113}$$

With the corresponding Jacobians being equal to:

$$\begin{aligned}
&\frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{V}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{V}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\omega}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\omega}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\eta}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\eta}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\delta}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\delta}} \\
&\frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{V}}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{V}}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\omega}}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\omega}}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\eta}}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\eta}}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\delta}}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\delta}}} \\
&\frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\eta}}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\eta}}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\delta}}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\delta}}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{g}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{g}} \qquad \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{w}} = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{w}} \\
&\boldsymbol{M}_C = \boldsymbol{E}_M \boldsymbol{\Phi}_{bl} \boldsymbol{P}_{a,c}
\end{aligned} \tag{3.114}$$

And the expression for the body moments lastly becomes:

$$\begin{aligned} \boldsymbol{M}_b = &\frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\eta}}\boldsymbol{\eta} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} \\ &+ \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{M}_C \end{aligned} \tag{3.115}$$

GENERALIZED FORCES

The generalized forces are obtained using splining matrix $\boldsymbol{\Phi}_{hl}$, which interpolates the aerodynamic forces to modal coordinates:

$$\boldsymbol{P}_h = \boldsymbol{\Phi}_{hl}\boldsymbol{P}_a \tag{3.116}$$

The Jacobians can be identified as:

$$\begin{aligned} &\frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{V}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{V}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\omega}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\omega}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\eta}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\eta}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\delta}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\delta}} \\ &\frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{V}}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{V}}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\omega}}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\omega}}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\eta}}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\eta}}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\delta}}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\delta}}} \\ &\frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\eta}}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\eta}}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\delta}}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\delta}}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{g}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{g}} \quad \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{w}} = \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{w}} \\ &\boldsymbol{P}_C = \boldsymbol{\Phi}_{hl}\boldsymbol{P}_{a,c} \end{aligned} \tag{3.117}$$

The generalized force expression in terms of the states, inputs, and disturbances becomes:

$$\begin{aligned} \boldsymbol{P}_h = &\frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\eta}}\boldsymbol{\eta} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} \\ &+ \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{P}_C \end{aligned} \tag{3.118}$$

3.6.2. AERODYNAMIC LAG FORCES

A problem with the physical RFA, as has been noted earlier, is the large number of lag states which in total equals $p \times n$ with p the number of poles and n the number of aerodynamic panels. Typically, the number of poles is at minimum 4 and the number of panels anywhere between 1000 and 4000, meaning that the number of states of the state-space model explodes, which is undesirable for aeroservoelasticity, as the order of the system should be kept to a minimum for controller synthesis, preferably while retaining the same input-output behaviour. In Section 3.6 the aerodynamics are expressed in terms of the rigid-body forces and moments, as well as generalized forces and moments in the aeroelastic equation, which also means that the lag vector associated with each pole is reduced to 3 rigid-body force contributions, 3 rigid-body moment contributions, and m generalized force contributions, with m the number of modes. Therefore, the order of the lag states can be reduced substantially by introducing rigid-body lag force, rigid-body lag moment, and generalized lag forces for each pole as the states. This can be achieved by introducing $(6+m) \times p$ scalar equations/states with m the number of modes and p the number of poles, as opposed to $n \times p$ vector equations for the lag dynamics. By left-multiplying the lag vector of each pole by the lag matrix corresponding to each pole q and the respective splining matrix. In order to express the lag states, the aerodynamic lag stiffness matrix and the aerodynamic lag damping matrix for each pole q are introduced:

$$\boldsymbol{K}_q = \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{A}_q \tag{3.119}$$

$$\boldsymbol{C}_q = \frac{1}{2}\rho V_\infty \boldsymbol{S}\boldsymbol{A}_q \tag{3.120}$$

Again, the expanded form of these matrices is obtained by multiplying with the projection matrix $\boldsymbol{\Pi}$, and also denoted with a tilde:

$$\tilde{\boldsymbol{K}}_q = \boldsymbol{\Pi}\boldsymbol{K}_q \tag{3.121}$$

$$\tilde{\boldsymbol{C}}_q = \boldsymbol{\Pi}\boldsymbol{C}_q \tag{3.122}$$

The following Jacobians will be reused in the subsequent subsections, and are therefore introduced here.

$$\begin{aligned}
\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{V}}} &= \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{tr} & \frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\omega}}} &= \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{ro} & \frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\delta}}} &= \tilde{\boldsymbol{K}}_q \boldsymbol{W}_{cd} & \frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\delta}}} &= \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{cm} \\
\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\eta}}} &= \tilde{\boldsymbol{K}}_q \boldsymbol{W}_{sd} & \frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\eta}}} &= \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{sm} & \frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{g}} &= \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{ga} & \frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{w}} &= \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{wa}
\end{aligned} \tag{3.123}$$

AERODYNAMIC BODY LAG FORCES

First, the lag dynamics for the aerodynamic body force lag are derived. This is achieved by left-multiplying Equation 3.64, which represents the lag dynamics per pole, with the terms required to express the lag states per pole in terms of body forces:

$$\boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{K}}_q \dot{\boldsymbol{l}}_m = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{K}}_q \dot{\boldsymbol{w}}_j - \left(\frac{2V_\infty}{\bar{c}}\right) \beta_q \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{K}}_q \boldsymbol{l}_q \tag{3.124}$$

Subsequently, the aerodynamic rigid-body lag force per pole is introduced as a state vector:

$$\boldsymbol{F}_l^q = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{K}}_q \boldsymbol{l}_q \tag{3.125}$$

$$\begin{aligned}
\dot{\boldsymbol{F}}_l^q = \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{K}}_q \Bigg[& \boldsymbol{W}_{tr} \left(\frac{\dot{\boldsymbol{V}}}{V_\infty}\right) + \boldsymbol{W}_{ro} \left(\frac{\dot{\boldsymbol{\omega}}}{V_\infty}\right) + \boldsymbol{W}_{cd} \dot{\boldsymbol{\delta}} + \boldsymbol{W}_{cm} \left(\frac{\ddot{\boldsymbol{\delta}}}{V_\infty}\right) \\
& + \boldsymbol{W}_{sd} \dot{\boldsymbol{\eta}} + \boldsymbol{W}_{sm} \left(\frac{\ddot{\boldsymbol{\eta}}}{V_\infty}\right) + \boldsymbol{W}_{ga} \left(\frac{\boldsymbol{g}}{V_\infty}\right) + \boldsymbol{W}_{wa} \left(\frac{\boldsymbol{w}}{V_\infty}\right) \Bigg] - \left(\frac{2V_\infty}{\bar{c}}\right) \beta_q \boldsymbol{F}_l^q
\end{aligned} \tag{3.126}$$

Writing out this equation in full gives, and recognizing that $\tilde{\boldsymbol{C}}_q = \tilde{\boldsymbol{K}}_q / V_\infty$:

$$\begin{aligned}
\dot{\boldsymbol{F}}_l^q = & \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{tr} \dot{\boldsymbol{V}} + \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{ro} \dot{\boldsymbol{\omega}} + \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{K}}_q \boldsymbol{W}_{cd} \dot{\boldsymbol{\delta}} + \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{cm} \ddot{\boldsymbol{\delta}} + \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{K}}_q \boldsymbol{W}_{sd} \dot{\boldsymbol{\eta}} \\
& + \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{sm} \ddot{\boldsymbol{\eta}} + \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{ga} \boldsymbol{g} + \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \tilde{\boldsymbol{C}}_q \boldsymbol{W}_{wa} \boldsymbol{w} - \left(\frac{2V_\infty}{\bar{c}}\right) \beta_q \boldsymbol{F}_l^q
\end{aligned} \tag{3.127}$$

The aerodynamic body force lag Jacobians per pole can then be identified as follows, by recognizing the terms in Equation 3.127 as those of Equation 3.123.

$$\begin{aligned}
\frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{V}}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{V}}} & \frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{\omega}}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\omega}}} & \frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{\delta}}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{F}_l^q}{\partial \ddot{\boldsymbol{\delta}}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\delta}}} \\
\frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{\eta}}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{F}_l^q}{\partial \ddot{\boldsymbol{\eta}}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{F}_l^q}{\partial \boldsymbol{g}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{g}} & \frac{\partial \boldsymbol{F}_l^q}{\partial \boldsymbol{w}} &= \boldsymbol{E}_F \boldsymbol{\Phi}_{bl} \frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{w}}
\end{aligned} \tag{3.128}$$

This then gives the aerodynamic body lag force dynamics per pole, expressed in terms of the model states, inputs, and disturbances:

$$\dot{\boldsymbol{F}}_l^q = \frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{V}}} \dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{\omega}}} \dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{F}_l^q}{\partial \ddot{\boldsymbol{\delta}}} \ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_l^q}{\partial \ddot{\boldsymbol{\eta}}} \ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{F}_l^q}{\partial \boldsymbol{g}} \boldsymbol{g} + \frac{\partial \boldsymbol{F}_l^q}{\partial \boldsymbol{w}} \boldsymbol{w} + \frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{\delta}}} \dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_l^q}{\partial \dot{\boldsymbol{\eta}}} \dot{\boldsymbol{\eta}} - \frac{2V_\infty}{\bar{c}} \beta_q \boldsymbol{F}_l^q \tag{3.129}$$

By gathering the aerodynamic body lag force dynamic equations of all poles into one equation, the aerodynamic body lag force dynamics result:

$$\dot{\boldsymbol{F}}_l = \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \dot{\boldsymbol{V}}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \dot{\boldsymbol{V}}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \dot{\boldsymbol{V}}} \end{bmatrix} \dot{\boldsymbol{V}} + \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \dot{\boldsymbol{\omega}}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \dot{\boldsymbol{\omega}}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \dot{\boldsymbol{\omega}}} \end{bmatrix} \dot{\boldsymbol{\omega}} + \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \ddot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \ddot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \ddot{\boldsymbol{\delta}}} \end{bmatrix} \ddot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \ddot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \ddot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \ddot{\boldsymbol{\eta}}} \end{bmatrix} \ddot{\boldsymbol{\eta}} + \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \boldsymbol{g}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \boldsymbol{g}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \boldsymbol{g}} \end{bmatrix} \boldsymbol{g} + \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \boldsymbol{w}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \boldsymbol{w}} \end{bmatrix} \boldsymbol{w} + \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \dot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \dot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \dot{\boldsymbol{\delta}}} \end{bmatrix} \dot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{F}_l^1}{\partial \dot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{F}_l^2}{\partial \dot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{F}_l^p}{\partial \dot{\boldsymbol{\eta}}} \end{bmatrix} \dot{\boldsymbol{\eta}} \tag{3.130}$$
$$- \begin{bmatrix} \frac{2V_\infty}{\bar{c}}\beta_0 & 0 & \cdots & 0 \\ 0 & \frac{2V_\infty}{\bar{c}}\beta_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{2V_\infty}{\bar{c}}\beta_q \end{bmatrix} \boldsymbol{F}_l$$

Which can be simplified into the following, final equation:

$$\dot{\boldsymbol{F}}_l = \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{F}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{F}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{F}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_F\boldsymbol{F}_l \tag{3.131}$$

AERODYNAMIC BODY LAG MOMENTS

The aerodynamic body lag moment is derived in a similar fashion, but with the following state vector definition of the aerodynamic body lag moment per pole:

$$\boldsymbol{M}_l^q = \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\tilde{\boldsymbol{K}}_q\boldsymbol{l}_q \tag{3.132}$$

The aerodynamic body lag moments per pole can be described similar to how Equation 3.128 is derived using Equation 3.126 and Equation 3.127 for the aerodynamic body lag force, resulting in the Jacobians below:

$$\begin{aligned} \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{V}}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{V}}} & \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{\omega}}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\omega}}} & \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{\delta}}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{M}_l^q}{\partial \ddot{\boldsymbol{\delta}}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{\eta}}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{M}_l^q}{\partial \ddot{\boldsymbol{\eta}}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{M}_l^q}{\partial \boldsymbol{g}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{g}} & \frac{\partial \boldsymbol{M}_l^q}{\partial \boldsymbol{w}} &= \boldsymbol{E}_M\boldsymbol{\Phi}_{bl}\frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{w}} \end{aligned} \tag{3.133}$$

Which gives the aerodynamic body moment lag dynamics per pole:

$$\dot{\boldsymbol{M}}_l^q = \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{M}_l^q}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_l^q}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{M}_l^q}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{M}_l^q}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_l^q}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} - \frac{2V_\infty}{\bar{c}}\beta_q\boldsymbol{M}_l^q \tag{3.134}$$

Again, the aerodynamic body lag moment of all poles can be described by gathering the individual body lag moment equations of each pole:

$$\dot{\boldsymbol{M}}_l = \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \dot{\boldsymbol{V}}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \dot{\boldsymbol{V}}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \dot{\boldsymbol{V}}} \end{bmatrix} \dot{\boldsymbol{V}} + \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \dot{\boldsymbol{\omega}}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \dot{\boldsymbol{\omega}}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \dot{\boldsymbol{\omega}}} \end{bmatrix} \dot{\boldsymbol{\omega}} + \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \ddot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \ddot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \ddot{\boldsymbol{\delta}}} \end{bmatrix} \ddot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \ddot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \ddot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \ddot{\boldsymbol{\eta}}} \end{bmatrix} \ddot{\boldsymbol{\eta}} + \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \boldsymbol{g}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \boldsymbol{g}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \boldsymbol{g}} \end{bmatrix} \boldsymbol{g} + \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \boldsymbol{w}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \boldsymbol{w}} \end{bmatrix} \boldsymbol{w} + \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \dot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \dot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \dot{\boldsymbol{\delta}}} \end{bmatrix} \dot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{M}_l^1}{\partial \dot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{M}_l^2}{\partial \dot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{M}_l^p}{\partial \dot{\boldsymbol{\eta}}} \end{bmatrix} \dot{\boldsymbol{\eta}}$$
$$- \begin{bmatrix} \frac{2V_\infty}{\bar{c}}\beta_0 & 0 & \cdots & 0 \\ 0 & \frac{2V_\infty}{\bar{c}}\beta_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{2V_\infty}{\bar{c}}\beta_q \end{bmatrix} \boldsymbol{M}_l \tag{3.135}$$

Which can also be further simplified as:

$$\dot{\boldsymbol{M}}_l = \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{M}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{M}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{M}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_M \boldsymbol{M}_l \tag{3.136}$$

AERODYNAMIC GENERALIZED LAG FORCES

For the aerodynamic generalized lag force the same procedure is applied, but with aerodynamic generalized lag force state vector per pole defined as:

$$\boldsymbol{P}_l^q = \boldsymbol{\Phi}_{hl}\tilde{\boldsymbol{K}}_q \boldsymbol{l}_q \tag{3.137}$$

The aerodynamic generalized lag forces per pole can be described, again, similar to how Equation 3.128 is derived using Equation 3.126 and Equation 3.127 for the aerodynamic body lag force, resulting in the Jacobians below:

$$\begin{aligned}
\frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{V}}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{V}}} & \frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{\omega}}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\omega}}} & \frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{\delta}}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{P}_l^q}{\partial \ddot{\boldsymbol{\delta}}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\delta}}} \\
\frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{\eta}}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{P}_l^q}{\partial \ddot{\boldsymbol{\eta}}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{P}_l^q}{\partial \boldsymbol{g}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{g}} & \frac{\partial \boldsymbol{P}_l^q}{\partial \boldsymbol{w}} &= \boldsymbol{\Phi}_{hl}\frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{w}}
\end{aligned} \tag{3.138}$$

The above Jacobians are again assembled into the aerodynamic generalized lag force dynamics per pole:

$$\dot{\boldsymbol{P}}_l^q = \frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{P}_l^q}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_l^q}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{P}_l^q}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{P}_l^q}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_l^q}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} - \frac{2V_\infty}{\bar{c}}\beta_q \boldsymbol{P}_l^q \tag{3.139}$$

The contributions of each pole are combined to form one equation representing the aerodynamic generalized lag force dynamics of all poles:

$$\begin{aligned}
\dot{\boldsymbol{P}}_l = &\begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \dot{\boldsymbol{V}}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \dot{\boldsymbol{V}}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \dot{\boldsymbol{V}}} \end{bmatrix}\dot{\boldsymbol{V}} + \begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \dot{\boldsymbol{\omega}}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \dot{\boldsymbol{\omega}}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \dot{\boldsymbol{\omega}}} \end{bmatrix}\dot{\boldsymbol{\omega}} + \begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \ddot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \ddot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \ddot{\boldsymbol{\delta}}} \end{bmatrix}\ddot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \ddot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \ddot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \ddot{\boldsymbol{\eta}}} \end{bmatrix}\ddot{\boldsymbol{\eta}} + \begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \boldsymbol{g}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \boldsymbol{g}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \boldsymbol{g}} \end{bmatrix}\boldsymbol{g} + \begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \boldsymbol{w}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \boldsymbol{w}} \end{bmatrix}\boldsymbol{w} + \begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \dot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \dot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \dot{\boldsymbol{\delta}}} \end{bmatrix}\dot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{P}_l^1}{\partial \dot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{P}_l^2}{\partial \dot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{P}_l^p}{\partial \dot{\boldsymbol{\eta}}} \end{bmatrix}\dot{\boldsymbol{\eta}} \\
&- \begin{bmatrix} \frac{2V_\infty}{\bar{c}}\beta_0 & 0 & \cdots & 0 \\ 0 & \frac{2V_\infty}{\bar{c}}\beta_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{2V_\infty}{\bar{c}}\beta_q \end{bmatrix}\boldsymbol{P}_l
\end{aligned} \tag{3.140}$$

Which can again be simplified to the following:

$$\dot{\boldsymbol{P}}_l = \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{P}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{P}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{P}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_P \boldsymbol{P}_l \tag{3.141}$$

AERODYNAMIC MONITORING LAG FORCES

A downside to this approach of expressing the lag states is that in Section 3.12, the aerodynamic lag force in terms of the forces and moments at the monitoring stations cannot be reconstructed with the FSM from the body lag forces/moments and generalized lag forces. Therefore, extra lag states need to be introduced which can be used in the output equations for the cutting forces and moments at the monitoring stations. The same projection is used as for body lag forces, body lag moments, and generalized lag forces, but with the splining matrix $\boldsymbol{\Phi}_{ml}$, which maps the aerodynamic forces at the quarter-chord point to the cutting forces and moments at the monitoring stations:

$$\boldsymbol{L}_l^q = \boldsymbol{\Phi}_{ml}\tilde{\boldsymbol{K}}_q \boldsymbol{l}_q \tag{3.142}$$

$$\begin{aligned}
\frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{V}}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{V}}} & \frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{\omega}}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\omega}}} & \frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{\delta}}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\delta}}} & \frac{\partial \boldsymbol{L}_l^q}{\partial \ddot{\boldsymbol{\delta}}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\delta}}} \\
\frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{\eta}}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \dot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{L}_l^q}{\partial \ddot{\boldsymbol{\eta}}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \ddot{\boldsymbol{\eta}}} & \frac{\partial \boldsymbol{L}_l^q}{\partial \boldsymbol{g}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{g}} & \frac{\partial \boldsymbol{L}_l^q}{\partial \boldsymbol{w}} &= \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a^q}{\partial \boldsymbol{w}}
\end{aligned} \tag{3.143}$$

Which is again assembled to:

$$\dot{\boldsymbol{L}}_l^q = \frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{L}_l^q}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_l^q}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{L}_l^q}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{L}_l^q}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_l^q}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} - \frac{2V_\infty}{\bar{c}}\beta_q \boldsymbol{L}_l^q \tag{3.144}$$

The equations pertaining to each pole are stacked again to form the final equation:

$$\begin{aligned}
\dot{\boldsymbol{L}}_l = &\begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \dot{\boldsymbol{V}}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \dot{\boldsymbol{V}}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \dot{\boldsymbol{V}}} \end{bmatrix}\dot{\boldsymbol{V}} + \begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \dot{\boldsymbol{\omega}}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \dot{\boldsymbol{\omega}}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \dot{\boldsymbol{\omega}}} \end{bmatrix}\dot{\boldsymbol{\omega}} + \begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \ddot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \ddot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \ddot{\boldsymbol{\delta}}} \end{bmatrix}\ddot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \ddot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \ddot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \ddot{\boldsymbol{\eta}}} \end{bmatrix}\ddot{\boldsymbol{\eta}} + \begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \boldsymbol{g}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \boldsymbol{g}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \boldsymbol{g}} \end{bmatrix}\boldsymbol{g} + \begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \boldsymbol{w}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \boldsymbol{w}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \boldsymbol{w}} \end{bmatrix}\boldsymbol{w} + \begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \dot{\boldsymbol{\delta}}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \dot{\boldsymbol{\delta}}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \dot{\boldsymbol{\delta}}} \end{bmatrix}\dot{\boldsymbol{\delta}} + \begin{bmatrix} \frac{\partial \boldsymbol{L}_l^1}{\partial \dot{\boldsymbol{\eta}}} \\ \frac{\partial \boldsymbol{L}_l^2}{\partial \dot{\boldsymbol{\eta}}} \\ \vdots \\ \frac{\partial \boldsymbol{L}_l^p}{\partial \dot{\boldsymbol{\eta}}} \end{bmatrix}\dot{\boldsymbol{\eta}} \\
&- \begin{bmatrix} \frac{2V_\infty}{\bar{c}}\beta_0 & 0 & \cdots & 0 \\ 0 & \frac{2V_\infty}{\bar{c}}\beta_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \frac{2V_\infty}{\bar{c}}\beta_q \end{bmatrix}\boldsymbol{L}_l
\end{aligned} \tag{3.145}$$

Which is also further simplified:

$$\dot{\boldsymbol{L}}_l = \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{L}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{L}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{L}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_L\boldsymbol{L}_l \tag{3.146}$$

3.7. ADDITIONAL FORCES AND MOMENTS

Next, the non-aerodynamic forces and moments are integrated into the model. These forces and moments constitute the thrust produced by engines and the weight of the aircraft. Additional forces and moments also include viscous drag and induced drag, but these are not added, and could be added as future work, as discussed in Section 6.2.

3.7.1. ENGINE THRUST

The engine thrust is defined in the thrust grids. The thrust vector is subsequently defined in this reference frame. The splining matrix $\boldsymbol{\Phi}_{bt}$ maps thrust grid to the body frame. The splining matrix $\boldsymbol{\Phi}_{ht}$ does the same, but to generalized coordinates instead. The thrust commands are given by input vector $\boldsymbol{u}_t$. However, these solely specify the thrust level, not the thrust direction, which is why the thrust mapping matrix $\boldsymbol{M}_t$ maps the thrust input vector to forces and moments in the thrust grid. Assembling these yield the thrust in the body frame and the generalized thrust:

$$\begin{pmatrix} \boldsymbol{F}_T \\ \boldsymbol{M}_T \end{pmatrix} = \boldsymbol{\Phi}_{bt}\boldsymbol{M}_t\boldsymbol{u}_t = \begin{pmatrix} \boldsymbol{E}_F\boldsymbol{\Phi}_{bt}\boldsymbol{M}_t \\ \boldsymbol{E}_M\boldsymbol{\Phi}_{bt}\boldsymbol{M}_t \end{pmatrix}\boldsymbol{u}_t \tag{3.147}$$

$$\boldsymbol{P}_T = \boldsymbol{\Phi}_{ht}\boldsymbol{M}_t\boldsymbol{u}_t \tag{3.148}$$

$$\boldsymbol{F}_T = \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{u}_t}\boldsymbol{u}_t \tag{3.149}$$

$$\boldsymbol{M}_T = \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{u}_t}\boldsymbol{u}_t \tag{3.150}$$

$$\boldsymbol{P}_T = \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{u}_t}\boldsymbol{u}_t \tag{3.151}$$

3.7.2. AIRCRAFT WEIGHT

The gravitational vector is defined in the inertial reference frame as $\boldsymbol{g}_0 = \begin{bmatrix} 0 & 0 & g_0 \end{bmatrix}$. Therefore, it needs to be transformed to the flight physical frame and subsequently converted to the body frame, giving the body gravitational vector:

$$\boldsymbol{g}_b(\boldsymbol{\theta}) = \boldsymbol{C}(\boldsymbol{\Theta})\boldsymbol{g}_0 \tag{3.152}$$

Multiplication with the aircraft mass gives the weight vector in the body frame. As the weight acts on the CG, there is no moment produced in the body frame.

$$\boldsymbol{W}_b = m\boldsymbol{g}_b(\boldsymbol{\Theta}) \tag{3.153}$$

3.8. RIGID-BODY KINEMATICS

The rotation from the inertial reference frame to the flight physical frame is performed using three consecutive rotations using the Euler angles. First, a rotation around the z-axis of the inertial frame by angle ψ. Secondly, a rotation around the y-axis by angle θ. Subsequently, a rotation around the x-axis by angle ϕ to obtain the flight physical frame. Lastly, the x-axis and z-axis are inverted to go from the flight physical frame to the body frame, in which the EOMs are expressed. This is represented using matrix notation as follows:

$$\begin{aligned} \boldsymbol{C}(\phi,\theta,\psi) &= \boldsymbol{T}_{bn}\boldsymbol{C}_3(\phi)\boldsymbol{C}_2(\theta)\boldsymbol{C}_1(\psi) \\ &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix} \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix} \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{aligned} \tag{3.154}$$

$$\boldsymbol{C}(\boldsymbol{\Theta}) = \boldsymbol{C}(\phi,\theta,\psi) = \begin{bmatrix} -\cos\psi\cos\theta & -\sin\psi\cos\theta & \sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ -\sin\phi\sin\psi - \cos\phi\sin\theta\cos\psi & \sin\phi\cos\psi - \cos\phi\sin\theta\sin\psi & -\cos\phi\cos\theta \end{bmatrix} \tag{3.155}$$

The velocity in the flight physical frame can be converted back to the velocity in the inertial reference frame $\dot{\boldsymbol{P}} = \begin{bmatrix} \dot{X} & \dot{Y} & \dot{Z} \end{bmatrix}^T$ using the inverse of the transformation matrix $\boldsymbol{C}(\boldsymbol{\Theta})$. As $\boldsymbol{C}(\boldsymbol{\Theta})$ is a rotation matrix, its inverse equals its transpose. Therefore, the state equation for the position of the aircraft is:

$$\dot{\boldsymbol{P}} = \boldsymbol{C}(\boldsymbol{\Theta})^T\boldsymbol{V} \tag{3.156}$$

The angular velocity expressed in the inertial reference frame $\dot{\boldsymbol{\Theta}} = \begin{bmatrix} \dot{\phi} & \dot{\theta} & \dot{\psi} \end{bmatrix}^T$ can be obtained from the angular velocity expressed in the flight physical frame similarly. Again, the body velocity must be expressed in the flight physical frame using $\boldsymbol{T}_{nb}$, as the state-space model expresses aircraft angular velocity in the body frame $\boldsymbol{\omega} = \begin{bmatrix} p & q & r \end{bmatrix}^T$ instead. From Etkin and Reid [6], the transformation matrix $\boldsymbol{R}$ is taken, which allows transforming Euler angle rates to the flight physical angular rates. The transformation matrix $\boldsymbol{E}(\boldsymbol{\Theta})$ is obtained when left-multiplying this matrix with matrix $\boldsymbol{T}_{bn}$:

$$\begin{aligned} \boldsymbol{E}(\boldsymbol{\Theta}) = \boldsymbol{E}(\phi,\theta,\psi) = \boldsymbol{T}_{bn}\boldsymbol{R} &= \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \\ &= \begin{bmatrix} -1 & 0 & \sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & \sin\phi & -\cos\theta\cos\phi \end{bmatrix} \end{aligned} \tag{3.157}$$

The inverse of this $\boldsymbol{E}(\boldsymbol{\Theta})$ matrix yields the transformation from the body angular rates to the Euler angle rates:

$$\boldsymbol{E}(\boldsymbol{\Theta})^{-1} = \boldsymbol{E}(\phi,\theta,\psi)^{-1} = \begin{bmatrix} -1 & \sin\phi\tan\theta & -\cos\phi\tan\theta \\ 0 & \cos\phi & \sin\phi \\ 0 & \sin\phi\sec\theta & -\cos\phi\sec\theta \end{bmatrix} \tag{3.158}$$

Therefore, the state equation for the orientation of the aircraft is:

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{E}(\boldsymbol{\Theta})^{-1}\boldsymbol{\omega} \tag{3.159}$$

3.9. ACTUATOR DYNAMICS

The model currently has the control surface deflections and control surface rates as states in the model. However, their dynamics have not been specified yet, which is achieved by modeling the actuator dynamics. As actuator models are specific to the aircraft at hand, the exact actuator model implemented may have to be adjusted. In order to include any actuator model however, the state-space model needs to be augmented with actuator states. A linear second-order actuator model is implemented here, which is capable of modeling a finite control surface speed, phase lag between commanded deflection and actual deflection, and overshoot and oscillation, as well as modeling limited bandwith. When moving to a nonlinear actuator model, the adjustments to the state-space model are relatively minor as only the state equations for the actuators have to be changed. Interesting nonlinearities for GLA, MLA, and AFS could be deflection and rate limits, which can be modeled by capping $\boldsymbol{\delta}$ and $\dot{\boldsymbol{\delta}}$, respectively.

$$\ddot{\delta} + 2\xi\omega_c\dot{\delta} + \omega_c^2\delta = \omega_c^2 u_c \tag{3.160}$$

$$\ddot{\delta} = -2\xi\omega_c\dot{\delta} - \omega_c^2\delta + \omega_c^2 u_c \tag{3.161}$$

The above second-order equation can be converted to a linear state-space model using the standard form with state vector $\boldsymbol{x} = \begin{bmatrix}\delta & \dot{\delta}\end{bmatrix}^T$ as shown below.

$$\begin{pmatrix}\dot{\delta}\\ \ddot{\delta}\end{pmatrix} = \begin{bmatrix}0 & 1\\ -\omega_c^2 & -2\xi\omega_c\end{bmatrix}\begin{pmatrix}\delta\\ \dot{\delta}\end{pmatrix} + \begin{bmatrix}0\\ \omega_c^2\end{bmatrix} u_c \tag{3.162}$$

Each control surface of the state-space model should have an actuator associated with it. Moreover, the panel downwash contributions defined in Section 3.6 require the deflection, velocity and acceleration of all control surfaces expressed as the derivative of one vector in order to compute the associated aerodynamic forces due to control surface, deflection, velocity and acceleration. Thus, the control surface deflection state is introduced as $\boldsymbol{\delta} = \begin{bmatrix}\delta_1 & \delta_2 & \cdots & \delta_c\end{bmatrix}^T$, and the control surface velocity state is introduced as $\dot{\boldsymbol{\delta}} = \begin{bmatrix}\dot{\delta}_1 & \dot{\delta}_2 & \cdots & \dot{\delta}_c\end{bmatrix}^T$, which results in the following state-space formulation:

$$\dot{\boldsymbol{\delta}} = \dot{\boldsymbol{\delta}} \tag{3.163}$$

$$\ddot{\boldsymbol{\delta}} = -\boldsymbol{D}_a\dot{\boldsymbol{\delta}} - \boldsymbol{K}_a\boldsymbol{\delta} + \boldsymbol{K}_a\boldsymbol{u}_c \tag{3.164}$$

The actuator damping is represented by matrix $\boldsymbol{D}_a$ in Equation 3.165, and the actuator stiffness is represented with $\boldsymbol{K}_a$ in Equation 3.166.

$$\boldsymbol{D}_a = \begin{bmatrix}2\xi_1\omega_1 & 0 & \cdots & 0\\ 0 & 2\xi_2\omega_2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & 2\xi_c\omega_c\end{bmatrix} \tag{3.165}$$

$$\boldsymbol{K}_a = \begin{bmatrix}\omega_1^2 & 0 & \cdots & 0\\ 0 & \omega_2^2 & \cdots & 0\\ \vdots & \vdots & \ddots & \vdots\\ 0 & 0 & \cdots & \omega_c^2\end{bmatrix} \tag{3.166}$$

3.10. ACCELEROMETER SENSOR MODEL

For Active Flutter Suppression and Active Load Alleviation, accelerometers are placed along stations on the aircraft structure. In this model these accelerometers are attached to nodes of the finite-element model. As such, the accelerations measured by these accelerometers can be expressed in terms of modal acceleration and rigid-body velocities and accelerations. The derivation for these is quite intricate, and out of scope for this thesis. However, Grauer and Boucher [107] has dedicated a technical report on output equations for flexible aircraft, which includes the derivation for the acceleration measurements. It is noted that accelerometers typically only output the acceleration along one axes. However, the purpose of this thesis is to solely model

the physics of the system, leaving such details to the control designer. For linear accelerations, the equation is as follows:

$$\frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}} = \boldsymbol{E}_T \boldsymbol{\Phi}_{sh} \tag{3.167}$$

$$\frac{\partial \boldsymbol{a}_r}{\partial \boldsymbol{\eta}} = \boldsymbol{E}_R \boldsymbol{\Phi}_{sh} \tag{3.168}$$

In order to define the accelerations purely using one vector-matrix expression, several matrix operators need to be defined, as the spline operations map to 3 DOFs per accelerometer. Equation 3.169 allows replicating/-expanding body accelerations to size $3 \times s$. Equation 3.170 essentially stacks the skew-symmetric matrix of $\boldsymbol{\nu}$, allowing matrix multiplication with the spline multiplication outputs.

$$\boldsymbol{R} = \mathbf{1}_a \otimes \boldsymbol{I}_{3\times 3} \tag{3.169}$$

$$\boldsymbol{S}(\nu) = \boldsymbol{I}_a \otimes \tilde{\boldsymbol{\nu}} \tag{3.170}$$

Subsequently, the translational acceleration sensor equation can be expressed as the following matrix-vector expression, as was done with all other equations in this model:

$$\boldsymbol{a}_t = \boldsymbol{R}(\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})) + 2\boldsymbol{S}(\boldsymbol{\omega})\frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\ddot{\boldsymbol{\eta}} + \boldsymbol{S}(\dot{\boldsymbol{\omega}})\left(\boldsymbol{r}_{bs} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\boldsymbol{\eta}\right) + \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{S}(\boldsymbol{\omega})\left(\boldsymbol{r}_{bs} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\boldsymbol{\eta}\right) \tag{3.171}$$

The first three terms are the accelerations experienced inside the body frame, the second term is the Coriolis term, the third is the modal acceleration, the fourth is the tangential acceleration, and the fifth the centrifugal acceleration [10]. The report also notes that angular accelerometers are not widely used. However, for the sake of completeness, angular accelerometers are also implemented in this aeroservoelastic model, with the equation being equal to:

$$\boldsymbol{a}_r = \boldsymbol{R}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{a}_r}{\partial \boldsymbol{\eta}}\ddot{\boldsymbol{\eta}} \tag{3.172}$$

3.11. CERTIFICATION GUST MODELING

Gusts are distinguished into two categories: discrete gusts and continuous gusts. Discrete gusts are isolated, deterministic gusts. Continuous gusts on the other hand model the atmospheric turbulence as a stochastic process using the statistical properties of turbulence. For certification gust loads, the discrete 1-cosine gust is typically the critical design case and is therefore the only gust implemented in this model, although other discrete gusts as well as continuous gusts, such as the Von Dryden or Von Karman turbulence model could be integrated easily by further augmenting the state-space model. The 1-cosine gust profile is defined using Equation 3.173, and its derivative is defined by Equation 3.174. The gust design velocity U_{ds} and the gust gradient H are defined by certification authorities as specified in the dedicated certification specifications.

$$U_g(t) = \begin{cases} \frac{U_{ds}}{2}\left(1 - \cos\left(\frac{\pi}{H}\left(V_\infty t - x_g\right)\right)\right) & \frac{x_g}{V_\infty} \leq t \leq \frac{2H+x_g}{V_\infty} \\ 0 & \text{otherwise} \end{cases} \tag{3.173}$$

$$\dot{U}_g(t) = \begin{cases} \frac{U_{ds}\pi V_\infty}{2H}\sin\left(\frac{\pi}{H}\left(V_\infty t - x_g\right)\right) & \frac{x_g}{V_\infty} \leq t \leq \frac{2H+x_g}{V_\infty} \\ 0 & \text{otherwise} \end{cases} \tag{3.174}$$

Gusts require special treatment as an external disturbance due to the nature in which they affect the aircraft, as the gust impacts the nose first and then travels downstream. This gust penetration effect is important to take into account when modeling gust loads [4]. The approach implemented here is similar to Wuestenhagen [10]. The gust signal is input as $U_{g,0}(t)$, which denotes the gust velocity at time t at the nose of the aircraft. The gust velocity at aerodynamic panel i at time t is equal to the gust velocity that impacted the nose τ_i seconds earlier, with τ_i defined as the time delay defined as the distance of panel i from the aircraft nose x_i over the freestream velocity V_∞:

$$\tau_i = \frac{x_i}{V_\infty} \tag{3.175}$$

$$U_{g,i}(t) = U_{g,0}(t - \tau_i) \tag{3.176}$$

Although this could be directly implemented into the model, where the downwash on each panel is computed as the aircraft penetrates the gust field, this leads to the same number of disturbance inputs as aerodynamic

panels can be troublesome for robust controller synthesis. Therefore, another approach is adopted here. Converting this gust equivalence relation to the Laplace domain gives:

$$U_{g,i}(s) = e^{-\tau_i s} U_{g,0}(s) \tag{3.177}$$

Therefore, the transfer function from the gust velocity at the nose to the gust velocity at panel i is defined by $G(s)$:

$$G(s) = \frac{U_{g,i}(s)}{U_{g,0}(s)} = e^{-\tau_i s} \tag{3.178}$$

However, because this transfer function is a transcendental term, it cannot be converted directly to the time-domain. This time delay transfer function is therefore approximated using the following second-order Padé approximation:

$$G(s) = \frac{s^2 - \frac{6}{\tau_i}s + \frac{12}{\tau_i^2}}{s^2 + \frac{6}{\tau_i}s + \frac{12}{\tau_i^2}} \tag{3.179}$$

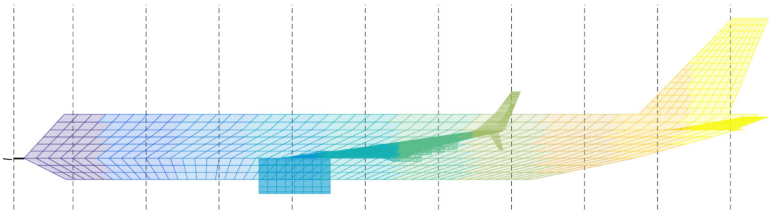


Figure 3.2: Illustration of aircraft with 10 gust zones. Taken from Wuestenhagen [10].

Now, another problem presents itself. The second-order Padé approximation is a good approximation to the exact time delay when this time delay is small. Thus, this approximation may not be valid when modeling the time delay between the nose and panels further aft. Therefore, gust zones are introduced and the time delay between each gust zone is modeled in a cascaded fashion, with subscript z indicating the gust zone, and thus $\Delta\tau_z$ being the time delay with respect to the previous zone $z-1$. An illustration of this gust zone discretization is shown in Figure 3.2. Utilising this approach allows further refining the number of zones to achieve a good approximation depending on the gust frequency content. Before detailing the resulting equations, first the Padé approximation needs to be converted to a state-space system, which the state-space model can be augmented with. First, the terms are split into numerator and denominator:

$$G(s) = \frac{N(s)}{D(s)} \tag{3.180}$$

$$N(s) = s^2 - \frac{6}{\Delta\tau_z}s + \frac{12}{\Delta\tau_z^2} \tag{3.181}$$

$$D(s) = s^2 + \frac{6}{\Delta\tau_z}s + \frac{12}{\Delta\tau_z^2} \tag{3.182}$$

The numerator can be rewritten in terms of the denominator:

$$N(s) = D(s) - \frac{12}{\Delta\tau_z}s \tag{3.183}$$

The transfer function $G(s)$ then becomes as follows. The transfer function H(s) subsequently results as Equation 3.185.

$$G(s) = \frac{D(s) - \frac{12}{\Delta\tau_z}s}{D(s)} = 1 - \frac{12}{\Delta\tau_z}\frac{s}{s^2 + \frac{6}{\Delta\tau_z}s + \frac{12}{\Delta\tau_z^2}} = 1 + H(s) \tag{3.184}$$

$$H(s) = \frac{-\frac{12}{\Delta\tau_z}s}{s^2 + \frac{6}{\Delta\tau_z}s + \frac{12}{\Delta\tau_z^2}} \tag{3.185}$$

Equation 3.185 is of the following general form:

$$H(s) = \frac{\beta_1 s + \beta_0}{s^2 + a_1 s + a_0} \tag{3.186}$$

This transfer function can be directly expressed in controllable canonical form:

$$\begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -a_0 & -a_1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \tag{3.187}$$

$$z = \begin{bmatrix} \beta_0 & \beta_1 \end{bmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \tag{3.188}$$

$$y(t) = z(t) + u(t) \tag{3.189}$$

Substituting the coefficients back in yields the linear state-space system approximating the gust delays. Equation 3.190 is the resulting state equation for the gust delay dynamics, and Equation 3.191 the output equation for the gust velocity of the zone.

$$\begin{pmatrix} \dot{g}_{z,1} \\ \dot{g}_{z,2} \end{pmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{12}{\Delta\tau_z^2} & -\frac{6}{\Delta\tau_z} \end{bmatrix} \begin{pmatrix} g_{z,1} \\ g_{z,2} \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \end{pmatrix} U_{g,z-1}(t) \tag{3.190}$$

$$U_{g,z}(t) = -\frac{12}{\Delta\tau_z} g_{z,2} + U_{g,z-1}(t) \tag{3.191}$$

The derivative of the velocity of each gust zone is also required explicitly as was shown in Section 3.6, which is obtained by differentiating the output equation of each zone, but this is addressed later in this section. As can be observed, the above equations depend on the gust velocity of the previous zone. Therefore, to realize one state-space model describing the gust delay dynamics of all gust zones, these need to be back-substituted. For the gust zone velocity of each zone z this can be represented using the following sum:

$$U_{g,z}(t) = U_{g,0}(t) - \sum_{k=1}^{z} \frac{12}{\Delta\tau_k} g_{k,2} \tag{3.192}$$

Subsequently, the following state-space model can be assembled, describing the gust delay dynamics, with $\boldsymbol{g} = \begin{bmatrix} g_{1,1} & g_{1,2} & g_{2,1} & g_{2,1} & \cdots & g_{z,1} & g_{z,2} \end{bmatrix}^T$.

$$\dot{\boldsymbol{g}} = \boldsymbol{A}_g \boldsymbol{g} + \boldsymbol{B}_g U_{g,0} \tag{3.193}$$

With the following definition of the gust state matrix $\boldsymbol{A}_g$ and gust input matrix $\boldsymbol{B}_g$:

$$\boldsymbol{A}_g = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ -\frac{12}{\Delta\tau_1^2} & -\frac{6}{\Delta\tau_1} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\frac{12}{\Delta\tau_1} & -\frac{12}{\Delta\tau_2^2} & -\frac{6}{\Delta\tau_2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & \cdots & 0 & 0 \\ 0 & -\frac{12}{\Delta\tau_1} & 0 & -\frac{12}{\Delta\tau_2} & -\frac{12}{\Delta\tau_3^2} & -\frac{6}{\Delta\tau_3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 \\ 0 & -\frac{12}{\Delta\tau_1} & 0 & -\frac{12}{\Delta\tau_2} & 0 & -\frac{12}{\Delta\tau_3} & \cdots & -\frac{12}{\Delta\tau_z^2} & -\frac{6}{\Delta\tau_z} \end{bmatrix} \tag{3.194}$$

$$\boldsymbol{B}_g = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 1 \\ 0 \\ 1 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \tag{3.195}$$

Two output equations result: one for the gust zone velocities and one for the gust zone accelerations. The output equation for the gust zone velocities becomes:

$$\boldsymbol{U}_g = \boldsymbol{C}_g \boldsymbol{g} + \boldsymbol{D}_g U_{g,0} \tag{3.196}$$

$$\mathbf{C}_g = \begin{bmatrix} 0 & -\frac{12}{\Delta\tau_1} & 0 & 0 & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\frac{12}{\Delta\tau_1} & 0 & -\frac{12}{\Delta\tau_2} & 0 & 0 & \cdots & 0 & 0 \\ 0 & -\frac{12}{\Delta\tau_1} & 0 & -\frac{12}{\Delta\tau_2} & 0 & -\frac{12}{\Delta\tau_3} & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & -\frac{12}{\Delta\tau_1} & 0 & -\frac{12}{\Delta\tau_2} & 0 & -\frac{12}{\Delta\tau_3} & \cdots & 0 & -\frac{12}{\Delta\tau_z} \end{bmatrix} \tag{3.197}$$

$$\boldsymbol{D}_g = \begin{bmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \tag{3.198}$$

The resulting output equation is multiplied with the gust mapping matrix $\boldsymbol{M}_g$, which maps the gust zone velocities to the downwash at each aerodynamic panel. This mapping is built by dividing the aircraft length by the number of gust zones, and assigning the z-component of the normal vector of the corresponding panel to it. Additionally, the equation is divided by the freestream velocity to result in an equation relating the panel downwash to the gust states and input:

$$\boldsymbol{w}_g = \frac{\boldsymbol{M}_g \boldsymbol{U}_g}{V_\infty} = \boldsymbol{M}_g \boldsymbol{C}_g \left(\frac{\boldsymbol{g}}{V_\infty} \right) + \boldsymbol{M}_g \boldsymbol{D}_g \left(\frac{U_{g,0}}{V_\infty} \right) \tag{3.199}$$

Taking the derivative of Equation 3.192 yields the output equation for the gust zone acceleration, which is expressed using the gust states by back-substituting the gust state equation:

$$\dot{\boldsymbol{w}}_g = \frac{\boldsymbol{M}_g \dot{\boldsymbol{U}}_g}{V_\infty} = \boldsymbol{M}_g \boldsymbol{C}_g \left(\frac{\dot{\boldsymbol{g}}}{V_\infty} \right) + \boldsymbol{M}_g \boldsymbol{D}_g \left(\frac{\dot{U}_{g,0}}{V_\infty} \right) = \boldsymbol{M}_g \boldsymbol{C}_g \left[\boldsymbol{A}_g \left(\frac{\boldsymbol{g}}{V_\infty} \right) + \boldsymbol{B}_g \left(\frac{U_{g,0}}{V_\infty} \right) \right] + \boldsymbol{M}_g \boldsymbol{D}_g \left(\frac{\dot{U}_{g,0}}{V_\infty} \right) \tag{3.200}$$

$$\dot{\boldsymbol{w}}_g = \boldsymbol{M}_g \boldsymbol{C}_g \boldsymbol{A}_g \left(\frac{\boldsymbol{g}}{V_\infty} \right) + \boldsymbol{M}_g \boldsymbol{C}_g \boldsymbol{B}_g \left(\frac{U_{g,0}}{V_\infty} \right) + \boldsymbol{M}_g \boldsymbol{D}_g \left(\frac{\dot{U}_{g,0}}{V_\infty} \right) \tag{3.201}$$

Introducing the gust disturbance vector $\boldsymbol{w} = \begin{bmatrix} U_{g,0} & \dot{U}_{g,0} \end{bmatrix}^T$, the state and output equations become:

$$\dot{\boldsymbol{g}} = \boldsymbol{A}_g \boldsymbol{g} + \begin{bmatrix} \boldsymbol{B}_g & \boldsymbol{0} \end{bmatrix} \boldsymbol{w} \tag{3.202}$$

$$\boldsymbol{w}_g = \boldsymbol{M}_g \boldsymbol{C}_g \left(\frac{\boldsymbol{g}}{V_\infty} \right) + \begin{bmatrix} \boldsymbol{M}_g \boldsymbol{D}_g & \boldsymbol{0} \end{bmatrix} \left(\frac{\boldsymbol{w}}{V_\infty} \right) \tag{3.203}$$

$$\dot{\boldsymbol{w}}_g = \boldsymbol{M}_g \boldsymbol{C}_g \boldsymbol{A}_g \left(\frac{\boldsymbol{g}}{V_\infty} \right) + \begin{bmatrix} \boldsymbol{M}_g \boldsymbol{C}_g \boldsymbol{B}_g & \boldsymbol{M}_g \boldsymbol{D}_g \end{bmatrix} \left(\frac{\boldsymbol{w}}{V_\infty} \right) \tag{3.204}$$

3.12. STRUCTURAL LOADS RECOVERY

As the state equations do not directly describe the actual loads on the structure, these need to be retrieved from the model, a process known as structural loads recovery. As discussed in Section 2.1, the Force Summation Method (FSM) is superior for loads analysis and therefore applied in this model as well. The FSM works by summing all force components on the structural grid in order to reconstruct the internal structural loads. These internal structural loads at the structural nodes are the externally applied loads minus the inertial loads experienced by the structure [39]:

$$\boldsymbol{L}_{g,int} = \boldsymbol{L}_{g,ext} - \boldsymbol{L}_{g,iner} \tag{3.205}$$

These internal structural loads can be used to compute the internal shear forces, bending moments, and torsional moments. Using the splining matrix $\boldsymbol{\Phi}_{mg}$ the internal shear forces, bending moments, and torsional moments at the defined monitoring stations can be obtained. Most often, these monitoring stations are placed along the span of the wing. This then allows to extract the WRBM and WRTM at various stations along the span, for instance. The forces and moments at the monitoring stations are indicated using vector $\boldsymbol{L}_m$, with $\boldsymbol{L}$ signifying the loads, and is computed as follows with m indicating the monitoring stations and g indicating the structural nodes:

$$\boldsymbol{L}_m = \boldsymbol{\Phi}_{mg} \boldsymbol{L}_{g,int} = \boldsymbol{\Phi}_{mg} \left(\boldsymbol{L}_{g,ext} - \boldsymbol{L}_{g,iner} \right) = \boldsymbol{L}_{m,ext} - \boldsymbol{L}_{m,iner} \tag{3.206}$$

The goal of this section is to express $\boldsymbol{L}_{m,ext}$, the external loads at the monitoring stations, and $\boldsymbol{L}_{m,iner}$, the inertial loads at the monitoring stations. First, the externally applied loads are derived, which requires establishing the relationship between the aerodynamic forces at the aerodynamic l-grid and the monitoring stations, which is achieved using splining matrix $\boldsymbol{\Phi}_{ml}$, which can be obtained by matrix-multiplying the splining matrix $\boldsymbol{\Phi}_{ms}$, mapping structural DOFs to the monitoring station DOFs, with splining matrix $\boldsymbol{\Phi}_{sl}$, mapping the aerodynamic l-grid to structural DOFs:

$$\boldsymbol{\Phi}_{ml} = \boldsymbol{\Phi}_{ms}\boldsymbol{\Phi}_{sl} \tag{3.207}$$

Then, the following Jacobians can be defined, which is similar to previous results in Section 3.6:

$$\begin{aligned}
&\frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{V}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{V}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\omega}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\omega}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\eta}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\eta}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\delta}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{\delta}} \\
&\frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{V}}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{V}}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\omega}}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\omega}}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\eta}}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\eta}}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\delta}}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \dot{\boldsymbol{\delta}}} \\
&\frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\eta}}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\eta}}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\delta}}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \ddot{\boldsymbol{\delta}}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{g}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{g}} \quad \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{w}} = \boldsymbol{\Phi}_{ml}\frac{\partial \boldsymbol{P}_a}{\partial \boldsymbol{w}} \\
&\boldsymbol{L}_C = \boldsymbol{\Phi}_{ml}\boldsymbol{L}_{a,c}
\end{aligned} \tag{3.208}$$

The external loads at the monitoring stations can then be expanded using the above Jacobians, giving Equation 3.210. Additionally, the aerodynamic lag force at the monitoring stations, which evolves separately as a state in the state-space model as discussed in Subsection 3.6.2, is also added, with the selection matrix $\boldsymbol{E}_L$ defined by Equation 3.209, with s the number of monitoring stations.

$$\boldsymbol{E}_L = \begin{bmatrix} \boldsymbol{I}_{6s\times 6s} & \boldsymbol{I}_{6s\times 6s} & \cdots & \boldsymbol{I}_{6s\times 6s} \end{bmatrix} \tag{3.209}$$

$$\begin{aligned}
\boldsymbol{L}_{m,ext} = &\frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\eta}}\boldsymbol{\eta} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} \\
&+ \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{L}_C + \boldsymbol{E}_L\boldsymbol{L}_l
\end{aligned} \tag{3.210}$$

The inertial nodal loads are computed by applying Equation 3.211 with the acceleration experienced inside the body frame, which consists of two components, the linear and angular acceleration, inside the body frame, as shown by Equation 3.212.

$$\boldsymbol{L}_{g,iner} = \boldsymbol{M}_{gg}\begin{bmatrix} \boldsymbol{\Phi}_{gb} & \boldsymbol{\Phi}_{gh} \end{bmatrix}\begin{pmatrix} \ddot{\boldsymbol{u}}_b \\ \ddot{\boldsymbol{u}}_h \end{pmatrix} \tag{3.211}$$

$$\ddot{\boldsymbol{u}}_b = \begin{pmatrix} \ddot{\boldsymbol{u}}_t \\ \ddot{\boldsymbol{u}}_r \end{pmatrix} \tag{3.212}$$

Equation 3.211 also needs to be expressed in terms of the state-space model states, inputs, and disturbances. First, the inertial loads equation is expanded and the splining matrix $\boldsymbol{\Phi}_{gb}$ is split into two block matrices, $\boldsymbol{\Phi}_{gt}$ for the linear acceleration and $\boldsymbol{\Phi}_{gr}$ for the angular acceleration. The inertial loads equation then becomes:

$$\boldsymbol{L}_{g,iner} = \boldsymbol{M}_{gg}\begin{bmatrix} \boldsymbol{\Phi}_{gt} & \boldsymbol{\Phi}_{gr} & \boldsymbol{\Phi}_{gh} \end{bmatrix}\begin{pmatrix} \ddot{\boldsymbol{u}}_t \\ \ddot{\boldsymbol{u}}_r \\ \ddot{\boldsymbol{u}}_h \end{pmatrix} = \boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gt}\ddot{\boldsymbol{u}}_t + \boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gr}\ddot{\boldsymbol{u}}_r + \boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gh}\ddot{\boldsymbol{u}}_h \tag{3.213}$$

The inertial loads at the monitoring stations are subsequently:

$$\boldsymbol{L}_{m,iner} = \boldsymbol{\Phi}_{mg}\boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gt}\ddot{\boldsymbol{u}}_t + \boldsymbol{\Phi}_{mg}\boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gr}\ddot{\boldsymbol{u}}_r + \boldsymbol{\Phi}_{mg}\boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gh}\ddot{\boldsymbol{u}}_h \tag{3.214}$$

This expression can be simplified by introducing mass matrices $\boldsymbol{M}_{mt}$, $\boldsymbol{M}_{mr}$, and $\boldsymbol{M}_{mh}$:

$$\boldsymbol{M}_{mt} = \boldsymbol{\Phi}_{mg}\boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gt} \qquad \boldsymbol{M}_{mr} = \boldsymbol{\Phi}_{mg}\boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gr} \qquad \boldsymbol{M}_{mh} = \boldsymbol{\Phi}_{mg}\boldsymbol{M}_{gg}\boldsymbol{\Phi}_{gh} \tag{3.215}$$

$$\boldsymbol{L}_{m,iner} = \boldsymbol{M}_{mt}\ddot{\boldsymbol{u}}_t + \boldsymbol{M}_{mr}\ddot{\boldsymbol{u}}_r + \boldsymbol{M}_{mh}\ddot{\boldsymbol{u}}_h \tag{3.216}$$

The linear and angular body acceleration are retrieved as follows. For the translational acceleration, this equals the translational body acceleration with the Coriolis force added and the body gravitational vector subtracted, resulting purely in acceleration due to external forces.

$$\ddot{\boldsymbol{u}}_t = \dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta}) \tag{3.217}$$

The rotational acceleration inside the body frame is extracted similarly, by adding the inertia-normalized gyroscopic term to the rotational body acceleration.

$$\ddot{\boldsymbol{u}}_r = \dot{\boldsymbol{\omega}} + \boldsymbol{J}^{-1}\boldsymbol{\omega} \times (\boldsymbol{J}\boldsymbol{\omega}) \tag{3.218}$$

The flexible acceleration is equal to the modal acceleration vector $\ddot{\boldsymbol{\eta}}$.

$$\ddot{\boldsymbol{u}}_h = \ddot{\boldsymbol{\eta}} \tag{3.219}$$

Finally, by assembling the external and inertial loads, this yields the expression for the loads at the monitoring stations:

$$\begin{aligned}\boldsymbol{L}_m = &\frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\eta}}\boldsymbol{\eta} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} \\ &+ \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{E}_L\boldsymbol{L}_l + \boldsymbol{L}_C \\ &- \boldsymbol{M}_{mt}\left(\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})\right) - \boldsymbol{M}_{mr}\left(\dot{\boldsymbol{\omega}} + \boldsymbol{J}^{-1}\boldsymbol{\omega} \times (\boldsymbol{J}\boldsymbol{\omega})\right) - \boldsymbol{M}_{mh}\ddot{\boldsymbol{\eta}}\end{aligned} \tag{3.220}$$

3.13. STATE-SPACE REALIZATION

As all governing equations have been derived the model can lastly be cast into state-space representation. Subsection 3.13.1 realizes the vector state equation. Subsection 3.13.2 realizes the vector output equation.

3.13.1. STATE EQUATIONS

From Section 3.8, Equation 3.221 and Equation 3.222 describe the position and orientation of the aircraft, respectively. These equations do not require further processing.

$$\dot{\boldsymbol{P}} = \boldsymbol{C}(\boldsymbol{\Theta})^T\boldsymbol{V} \tag{3.221}$$

$$\dot{\boldsymbol{\Theta}} = \boldsymbol{E}(\boldsymbol{\Theta})^{-1}\boldsymbol{\omega} \tag{3.222}$$

From Section 3.6, Equation 3.111 is substituted into the rigid-body translational equation of Equation 3.40. Additionally, the aerodynamic body force lag states derived in Subsection 3.6.2 are also added to the equations by means of summing the force contributions of each pole using selection matrix $\boldsymbol{E}_F$. Lastly, the additional forces and moments derived in Section 3.7 are added, specifically Equation 3.149 and Equation 3.153. Lastly, the state derivatives are brought from the RHS to the LHS.

$$\boldsymbol{E}_F = \begin{bmatrix}\boldsymbol{I}_{3\times3} & \boldsymbol{I}_{3\times3} & \cdots & \boldsymbol{I}_{3\times3}\end{bmatrix} \tag{3.223}$$

$$\begin{aligned}-\frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} - \frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} - \frac{\partial \boldsymbol{F}_b}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \left(m\boldsymbol{I} - \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{V}}}\right)\dot{\boldsymbol{V}} = &-m\boldsymbol{\omega} \times \boldsymbol{V} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\eta}}\boldsymbol{\eta} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} \\ &+ \frac{\partial \boldsymbol{F}_b}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{E}_F\boldsymbol{F}_l + \boldsymbol{F}_C + \frac{\partial \boldsymbol{F}_b}{\partial \boldsymbol{u}_t}\boldsymbol{u}_t + m\boldsymbol{g}_b(\boldsymbol{\Theta})\end{aligned} \tag{3.224}$$

From Section 3.6, Equation 3.115 is substituted into the rigid-body rotational equation of Equation 3.41. Additionally, the aerodynamic body moment lag states derived in Subsection 3.6.2 are also added to the equations by means of summing the moment contributions of each pole using selection matrix $\boldsymbol{E}_M$. Lastly, the moment due to engine thrust is added as per Equation 3.150. Lastly, the state derivatives are brought from the RHS to the LHS.

$$\boldsymbol{E}_M = \begin{bmatrix}\boldsymbol{I}_{3\times3} & \boldsymbol{I}_{3\times3} & \cdots & \boldsymbol{I}_{3\times3}\end{bmatrix} \tag{3.225}$$

$$\begin{aligned}-\frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} - \frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} - \frac{\partial \boldsymbol{M}_b}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \left(\boldsymbol{J} - \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{V}}}\right)\dot{\boldsymbol{\omega}} = &-\boldsymbol{\omega} \times \boldsymbol{J}\boldsymbol{\omega} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\eta}}\boldsymbol{\eta} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} \\ &+ \frac{\partial \boldsymbol{M}_b}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{E}_M\boldsymbol{M}_l + \boldsymbol{M}_C + \frac{\partial \boldsymbol{M}_b}{\partial \boldsymbol{u}_t}\boldsymbol{u}_t\end{aligned} \tag{3.226}$$

The aeroelastic equation is slightly different as the state-space model derived from modal analysis in Section 3.4 is used. Equation 3.227 shows the identify relation from Equation 3.51 that relates the derivative of the modal displacement state vector to the modal velocity vector. From Section 3.6, Equation 3.118 is substituted into the second equation of the modal state-space equation, Equation 3.53. Additionally, the aerodynamic generalized force lag states derived in Subsection 3.6.2 are also added to the equations by means of summing the contributions of each pole using selection matrix $\boldsymbol{E}_P$. Lastly, the generalized forces due to engine thrust are added as per Equation 3.151. Lastly, the state derivatives are brought from the RHS to the LHS.

$$\dot{\boldsymbol{\eta}} = \dot{\boldsymbol{\eta}} \tag{3.227}$$

$$\boldsymbol{E}_P = \begin{bmatrix} \boldsymbol{I}_{m\times m} & \boldsymbol{I}_{m\times m} & \cdots & \boldsymbol{I}_{m\times m} \end{bmatrix} \tag{3.228}$$

$$-\frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} - \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} - \frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \left(\boldsymbol{M}_{hh} - \frac{\partial \boldsymbol{P}_h}{\partial \ddot{\boldsymbol{\eta}}}\right)\ddot{\boldsymbol{\eta}} = \left(\frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\eta}}} - \boldsymbol{C}_{hh}\right)\dot{\boldsymbol{\eta}} + \left(\frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\eta}} - \boldsymbol{K}_{hh}\right)\boldsymbol{\eta} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} \tag{3.229}$$

$$+\frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{P}_h}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{E}_P\boldsymbol{P}_l + \frac{\partial \boldsymbol{P}_h}{\partial \boldsymbol{u}_t}\boldsymbol{u}_t \tag{3.230}$$

The state equations for the actuator dynamics originate from Section 3.9 and are shown again for completeness:

$$\dot{\boldsymbol{\delta}} = \dot{\boldsymbol{\delta}} \tag{3.231}$$

$$\ddot{\boldsymbol{\delta}} = -\boldsymbol{D}_a\dot{\boldsymbol{\delta}} - \boldsymbol{K}_a\boldsymbol{\delta} + \boldsymbol{K}_a\boldsymbol{u}_c \tag{3.232}$$

The gust dynamics were derived in Section 3.11 and is also repeated here for completeness:

$$\dot{\boldsymbol{g}} = \boldsymbol{A}_g\boldsymbol{g} + \boldsymbol{B}_g\boldsymbol{w} \tag{3.233}$$

Lastly, the aerodynamic body force lag, body moment lag, generalized force lag, and monitoring force lag dynamics are transformed by bringing the state derivatives from the RHS to the LHS of Equation 3.131, Equation 3.136, Equation 3.131, and Equation 3.146 respectively.

$$-\frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} - \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} - \frac{\partial \boldsymbol{F}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} - \frac{\partial \boldsymbol{F}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \dot{\boldsymbol{F}}_l = \frac{\partial \boldsymbol{F}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{F}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{F}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_F\boldsymbol{F}_l \tag{3.234}$$

$$-\frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} - \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} - \frac{\partial \boldsymbol{M}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} - \frac{\partial \boldsymbol{M}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \dot{\boldsymbol{M}}_l = \frac{\partial \boldsymbol{M}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{M}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{M}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_M\boldsymbol{M}_l \tag{3.235}$$

$$-\frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} - \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} - \frac{\partial \boldsymbol{P}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} - \frac{\partial \boldsymbol{P}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \dot{\boldsymbol{P}}_l = \frac{\partial \boldsymbol{P}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{P}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{P}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_P\boldsymbol{P}_l \tag{3.236}$$

$$-\frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} - \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} - \frac{\partial \boldsymbol{L}_l}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} - \frac{\partial \boldsymbol{L}_l}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \dot{\boldsymbol{L}}_l = \frac{\partial \boldsymbol{L}_l}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{L}_l}{\partial \boldsymbol{w}}\boldsymbol{w} + \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_l}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} + \boldsymbol{B}_L\boldsymbol{L}_l \tag{3.237}$$

The above equations can be neatly arranged into the state-space representation of Equation 3.26, with the nonlinearities represented by Equation 3.29, the descriptor/mass matrix equal to Equation 3.27, the state matrix equal to Equation 3.28, the input matrix equal to Equation 3.30, the disturbance matrix equal to Equation 3.31, and a bias term equal to Equation 3.32.

3.13.2. OUTPUT EQUATIONS

The state-space model implements a multitude of output equations. Although this depends on the specific feedback used, the most common output equations for flight dynamics have been implemented. However, missing output equations can be easily added my augmenting the state-space model. Equation 3.238 is the freestream velocity, Equation 3.239 is the angle of attack, Equation 3.240 is the sideslip angle, Equation 3.241 is the flight path angle, and Equation 3.242 is the load factor. The identity relationships of Equation 3.243, Equation 3.244, and Equation 3.245 output the aircraft position, aircraft orientation, and the body angular rates, respectively. Equation 3.246 describes the cutting forces and moments at the monitoring stations,

as was derived in Section 3.12. Equation 3.247 outputs the translational accelerations as measured by the accelerometers on the wing, and Equation 3.248 outputs the angular accelerations as measured by the accelerometers on the wing.

$$V_\infty(\boldsymbol{V}) = \sqrt{U^2 + V^2 + W^2} \tag{3.238}$$

$$\alpha(\boldsymbol{V}) = \arctan\left(-\frac{W}{U}\right) \tag{3.239}$$

$$\beta(\boldsymbol{V}) = \arctan\left(\frac{V}{\sqrt{U^2 + W^2}}\right) \tag{3.240}$$

$$\gamma(\boldsymbol{\Theta}, \boldsymbol{V}) = \theta - \alpha \tag{3.241}$$

$$\boldsymbol{n}_b = \frac{\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})}{g} \tag{3.242}$$

$$\boldsymbol{P} = \boldsymbol{P} \tag{3.243}$$

$$\boldsymbol{\Theta} = \boldsymbol{\Theta} \tag{3.244}$$

$$\boldsymbol{\omega} = \boldsymbol{\omega} \tag{3.245}$$

$$\begin{aligned}\boldsymbol{L}_m = &\frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{V}}\boldsymbol{V} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\omega}}\boldsymbol{\omega} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\eta}}\boldsymbol{\eta} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{\delta}}\boldsymbol{\delta} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{V}}}\dot{\boldsymbol{V}} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\omega}}}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\eta}}}\dot{\boldsymbol{\eta}} \\ &+ \frac{\partial \boldsymbol{L}_m}{\partial \dot{\boldsymbol{\delta}}}\dot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\eta}}}\ddot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{L}_m}{\partial \ddot{\boldsymbol{\delta}}}\ddot{\boldsymbol{\delta}} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{g}}\boldsymbol{g} + \frac{\partial \boldsymbol{L}_m}{\partial \boldsymbol{w}}\boldsymbol{w} + \boldsymbol{E}_L\boldsymbol{L}_l + \boldsymbol{L}_C \\ &- \boldsymbol{M}_{mt}\left(\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})\right) - \boldsymbol{M}_{mr}\left(\dot{\boldsymbol{\omega}} + \boldsymbol{J}^{-1}\boldsymbol{\omega} \times (\boldsymbol{J}\boldsymbol{\omega})\right) - \boldsymbol{M}_{mh}\ddot{\boldsymbol{\eta}}\end{aligned} \tag{3.246}$$

$$\boldsymbol{a}_t = \boldsymbol{R}(\dot{\boldsymbol{V}} + \boldsymbol{\omega} \times \boldsymbol{V} - \boldsymbol{g}_b(\boldsymbol{\Theta})) + 2\boldsymbol{S}(\boldsymbol{\omega})\frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\dot{\boldsymbol{\eta}} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\ddot{\boldsymbol{\eta}} + \boldsymbol{S}(\dot{\boldsymbol{\omega}})\left(\boldsymbol{r}_{bs} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\boldsymbol{\eta}\right) + \boldsymbol{S}(\boldsymbol{\omega})\boldsymbol{S}(\boldsymbol{\omega})\left(\boldsymbol{r}_{bs} + \frac{\partial \boldsymbol{a}_t}{\partial \boldsymbol{\eta}}\boldsymbol{\eta}\right) \tag{3.247}$$

$$\boldsymbol{a}_r = \boldsymbol{R}\dot{\boldsymbol{\omega}} + \frac{\partial \boldsymbol{a}_r}{\partial \boldsymbol{\eta}}\ddot{\boldsymbol{\eta}} \tag{3.248}$$

The above output equations can be put neatly into the vector output equation of Equation 3.33, with the nonlinear terms lumped into Equation 3.39, the output matrix is given by Equation 3.37, the input feedthrough matrix is equal to Equation 3.34, the disturbance feedthrough matrix is equal to Equation 3.35, and the bias terms are lumped into Equation 3.36. Although against standard conventions for state-space modeling, the equation also includes a matrix that multiplies the state derivative vector, which is referred to here as the output descriptor matrix, which is equal to Equation 3.38. The state equation is not substituted back in as the state descriptor matrix might be badly conditioned due to the small lag state entries.

3.14. NUMERICAL INTEGRATION

During the experience of this thesis it was observed that the descriptor matrix was problematic to invert as the numerical scales in this matrix vastly differ, a problem known as an ill-conditioned matrix. This problem was alleviated by utilising the known structure of the equations to solve the set of coupled equations in a partitioned manner, by dividing the state vector up into components related to the implicit variables, denoted with I, the components related to the actuators, denoted with A, and the state components related to the explicit states, denoted with E:

$$\boldsymbol{x} = \begin{bmatrix} \boldsymbol{x}_I \\ \boldsymbol{x}_A \\ \boldsymbol{x}_E \end{bmatrix} \quad (3.249) \qquad \boldsymbol{x}_I = \begin{bmatrix} \boldsymbol{V} \\ \boldsymbol{\omega} \\ \ddot{\boldsymbol{\eta}} \end{bmatrix} \quad (3.250) \qquad \boldsymbol{x}_A = \begin{bmatrix} \ddot{\boldsymbol{\delta}} \end{bmatrix} \quad (3.251) \qquad \boldsymbol{x}_E = \begin{bmatrix} \boldsymbol{P} \\ \boldsymbol{\Theta} \\ \boldsymbol{\delta} \\ \boldsymbol{\eta} \\ \boldsymbol{g} \\ \boldsymbol{F}_l \\ \boldsymbol{M}_l \\ \boldsymbol{P}_l \\ \boldsymbol{L}_l \end{bmatrix} \quad (3.252)$$

Equation 3.27 can be similarly partitioned, and several of these blocks can be identified as zero or identity:

$$\boldsymbol{E} = \begin{bmatrix} \boldsymbol{II} & \boldsymbol{IA} & \boldsymbol{IE} \\ \boldsymbol{AI} & \boldsymbol{AA} & \boldsymbol{AE} \\ \boldsymbol{EI} & \boldsymbol{EA} & \boldsymbol{EE} \end{bmatrix} = \begin{bmatrix} \boldsymbol{II} & \boldsymbol{IA} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{EI} & \boldsymbol{EA} & \boldsymbol{I} \end{bmatrix} \quad (3.253) \qquad \begin{bmatrix} \boldsymbol{II} & \boldsymbol{IA} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{I} & \boldsymbol{0} \\ \boldsymbol{EI} & \boldsymbol{EA} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{x}}_I \\ \dot{\boldsymbol{x}}_A \\ \dot{\boldsymbol{x}}_E \end{bmatrix} = \begin{bmatrix} \boldsymbol{R}_I \\ \boldsymbol{R}_A \\ \boldsymbol{R}_E \end{bmatrix} \quad (3.254)$$

First, the actuator state derivatives are computed using Equation 3.255. The implicit state derivatives are computed using a precomputed LU factorization using Equation 3.256. The conditioning of this matrix is much better, and thus it can be inverted. Lastly, the state derivatives of the explicit states can be evaluated using Equation 3.257.

$$\dot{\boldsymbol{x}}_A = \boldsymbol{R}_A \quad (3.255)$$

$$\boldsymbol{II}\dot{\boldsymbol{x}}_I = \boldsymbol{R}_I - \boldsymbol{IA}\dot{\boldsymbol{x}}_A \quad (3.256)$$

$$\dot{\boldsymbol{x}}_E = \boldsymbol{R}_E - \boldsymbol{EI}\dot{\boldsymbol{x}}_I - \boldsymbol{EA}\dot{\boldsymbol{x}}_A \quad (3.257)$$

3.15. AIRCRAFT TRIMMING

The aircraft trim condition is defined as the set of states and inputs that result in constant aircraft motion at a prescribed airspeed. Important in trimming is that the number of free variables equals the number of constraints, as otherwise the problem is underconstrained or overconstrained. For aircraft with multiple control surfaces, this requires allocating the control surfaces.

TRIM VARIABLES

As the state-space model contains nonlinearities, the trim problem is a nonlinear root-finding problem. The trim free variables, which are the degrees of freedom, in this problem are the control surface deflections $\boldsymbol{\delta}$, the modal displacement $\boldsymbol{\eta}$, the angle of attack α, the pitch angle θ, the elevator command u_e, the roll command u_r, the yaw command u_y, and, lastly, the thrust command u_t:

$$\boldsymbol{\xi} = \begin{bmatrix} \begin{bmatrix} \alpha & \theta & u_e & u_r & u_y & u_t \end{bmatrix} & \boldsymbol{\delta} & \boldsymbol{\eta} \end{bmatrix} \quad (3.258)$$

Although the aircraft is trimmed primarily along the pitch axis, the roll and yaw axes are also accounted for during trimming as the free-flying aircraft might have residual moments caused by the asymmetries during the modeling of the aircraft.

TRIM CONSTRAINTS

The trim condition should enforce steady, level flight. Therefore, the following subset of the state vector is constrained to 0 during the trimming. These constrain the acceleration along the body x- and z-axis through $\dot{U}$ and $\dot{W}$, ensure that the aircraft is not climbing nor descending through $\dot{Z}$, ensure no rolling of the aircraft through $\dot{\boldsymbol{\omega}}$, and ensure the control surfaces are not moving through $\ddot{\boldsymbol{\delta}}$, and that the modal acceleration is also zero through $\ddot{\boldsymbol{\eta}}$:

$$\boldsymbol{c}(\boldsymbol{\xi}) = \begin{bmatrix} \begin{bmatrix} \dot{U} & \dot{W} & \dot{Z} \end{bmatrix} & \dot{\boldsymbol{\omega}} & \ddot{\boldsymbol{\delta}} & \ddot{\boldsymbol{\eta}} \end{bmatrix}^T \quad (3.259)$$

The aircraft speed is constrained indirectly by prescribing the trim speed through the following relationships, which depend on the free α variable. The z-coordinate in the inertial frame is set equal to minus the aircraft altitude $Z = -h_0$. Additionally, all other states and inputs are initialized and kept to 0.

$$U = -V_\infty \cos(\alpha) \qquad W = -V_\infty \sin(\alpha) \tag{3.260}$$

NUMERICAL SOLUTION

Finding the trim solution is implemented numerically which attempts to find $\boldsymbol{\xi}$ such that the following condition is met. Using the final trim variables, the trim state vector $\boldsymbol{x}_0$ and the trim input vector $\boldsymbol{u}_0$ can be reconstructed.

$$\boldsymbol{c}(\boldsymbol{\xi}) = \boldsymbol{0} \tag{3.261}$$

3.16. MODEL LINEARIZATION

The resulting state-space model has nonlinear flight dynamics using mean-body axes, linear panel aerodynamics using VLM for quasi-steady aerodynamics, and DLM for unsteady aerodynamics, and linear finite-elements based on industrial finite-element models, output from NASTRAN. Therefore, the model is linear with exception of the nonlinearities introduced by the flight dynamics. A linear state-space representation with A, B, C, D, F, G matrices can be obtained numerically at the trim condition computed using Section 3.15, with trim state vector $\boldsymbol{x}_0$, trim input vector $\boldsymbol{u}_0$, and trim disturbance vector $\boldsymbol{w}_0$. The model could also be linearized analytically, but this is not executed here. Starting from the general state-space model:

$$\dot{\boldsymbol{x}} = f(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \qquad \boldsymbol{y} = h(\boldsymbol{x}, \boldsymbol{u}, \boldsymbol{w}) \tag{3.262}$$

The A, B, C, D, F, G matrices are computed using numerical linearization by applying small perturbations around the trim condition:

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0)} \qquad \boldsymbol{B} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0)} \qquad \boldsymbol{F} = \left.\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{w}}\right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0)} \tag{3.263}$$

$$\boldsymbol{C} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{x}}\right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0)} \qquad \boldsymbol{D} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{u}}\right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0)} \qquad \boldsymbol{G} = \left.\frac{\partial \boldsymbol{h}}{\partial \boldsymbol{w}}\right|_{(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0)} \tag{3.264}$$

The state, input, output, and disturbance vectors become:

$$\Delta \boldsymbol{x} = \boldsymbol{x} - \boldsymbol{x}_0 \qquad \Delta \boldsymbol{u} = \boldsymbol{u} - \boldsymbol{u}_0 \qquad \Delta \boldsymbol{y} = \boldsymbol{y} - \boldsymbol{y}_0 \qquad \Delta \boldsymbol{w} = \boldsymbol{w} - \boldsymbol{w}_0 \tag{3.265}$$

$$\boldsymbol{x}_0 = \boldsymbol{f}(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0) \qquad \boldsymbol{y}_0 = \boldsymbol{h}(\boldsymbol{x}_0, \boldsymbol{u}_0, \boldsymbol{w}_0) \tag{3.266}$$

The LTI state-space model then becomes:

$$\Delta \dot{\boldsymbol{x}} = \boldsymbol{A}\Delta \boldsymbol{x} + \boldsymbol{B}\Delta \boldsymbol{u} + \boldsymbol{F}\Delta \boldsymbol{w} \qquad \Delta \boldsymbol{y} = \boldsymbol{C}\Delta \boldsymbol{x} + \boldsymbol{D}\Delta \boldsymbol{u} + \boldsymbol{G}\Delta \boldsymbol{w} \tag{3.267}$$

4

Part II: Free-Flying Commercial Aircraft Pipeline

Although the mathematical state-space model derived in Chapter 3 is crucial for modeling free-flying flexible aircraft, it is useless without a corresponding representative aircraft to use as input as this determines what the matrices look like that are fed into it. Industrial free-flying finite-element models are scarcely available in academia and often tightly controlled within the industry. Therefore, this section presents a methodology and pipeline for converting clamped-wing NASTRAN finite-element models, which are more readily available, to a free-flying aeroservoelastic state-space model as defined in Chapter 3. This process is applied to the Embraer Benchmark Wing, which is provided by Embraer. This clamped-wing model is used for research on aeroelastic tailoring, and thus the resulting pipeline is moreover capable of generating free-flying aeroservoelastic state-space models of aircraft with aeroelastically tailored wings.

This chapter is structured similarly to the state-space model chapter, with this chapter also described in a top-down fashion, starting with an overview of the end-to-end modeling pipeline for converting a clamped-wing industrial finite-element model into a free-flying aeroservoelastic state-space model of the form derived in Chapter 3. Section 4.2 dives deeper into the modeling steps applied to convert the clamped-wing model into a free-flying model, the Embraer Benchmark Wing, integrating aeroelastically tailored wings, and mass case generation. Section 4.3 describes the aerodynamic mesh parameterization and generation for the lifting surfaces, as well as monitoring stations and camber and twist modeling. Lastly, Section 4.4 shows the resulting free-flying aeroelastic model and performs validation on the model.

4.1. Aircraft Modeling Pipeline Overview

Whereas Chapter 3 provides a mathematical framework for deriving free-flying aeroservoelastic state-space models based on the matrices and parameters used as input, these matrices and parameters do need to be derived for a particular aircraft configuration, mass model, aerodynamic model, and structural model, which requires a modeling pipeline, which is described in this section. At the outset of this work no representative free-flying NASTRAN model was available as input to the framework, highlighting the scarcity of these models within academia. Moreover, if such a model would be available, it is unfortunately not a matter of plug-and-play either, as systematic processing is required to get the matrices and parameters, further underscoring the need for a practical processing pipeline. Most often, however, only the wing box is modeled, assuming negligible interaction between rigid-body DOFs and elastic DOFs, and these models are subsequently more widely available, such as the uCRM Benchmark Models, created by Brooks, Kenway, and Martins [106]. Therefore, the processing pipeline presented here presupposes a clamped-wing finite-element model. To this end, the Embraer Benchmark Wing was made available, which is is briefly discussed in Subsection 4.2.1.

The end-to-end modeling pipeline is shown in Figure 4.1. DLR Loads Kernel [19], [20] is used to convert the free-flying aeroelastic NASTRAN model into the matrices and parameters that serve as input to the state-space framework derived in Chapter 3. This process is indicated in yellow in the flowchart. Using DLR Loads Kernel has two major advantages. First, it avoids the redundant work of building these matrices, which is

time-consuming to build from scratch and of less scientific value. Secondly, it allows backward compatibility with DLR Loads Kernel, allowing the use of their main- and postprocessor, for instance, for visual inspection of the aircraft model in their model viewer and for verification of the state-space model by running an equivalent analysis, while still reaping the benefits of the state-space model derived in Chapter 3. As such, most 3D model views in this chapter were created using this model viewer. The structural processing steps for converting a clamped-wing baseline NASTRAN finite-element model into a (non)tailored free-flying NASTRAN finite-element model is indicated in blue, and is the subject of Section 4.2, in which all steps are further detailed. The aeroelastic tailoring process is out of scope of this work, but the general steps are highlighted in red and discussed briefly in Subsection 4.2.3. Similarly, mass case generation is indicated in purple and discussed in Subsection 4.2.4. Lastly, an aerodynamic mesh must be generated, as finite-element models most often do not come with such a mesh. This is indicated in green in the pipeline, and the parameterization and generation is discussed in Section 4.3.

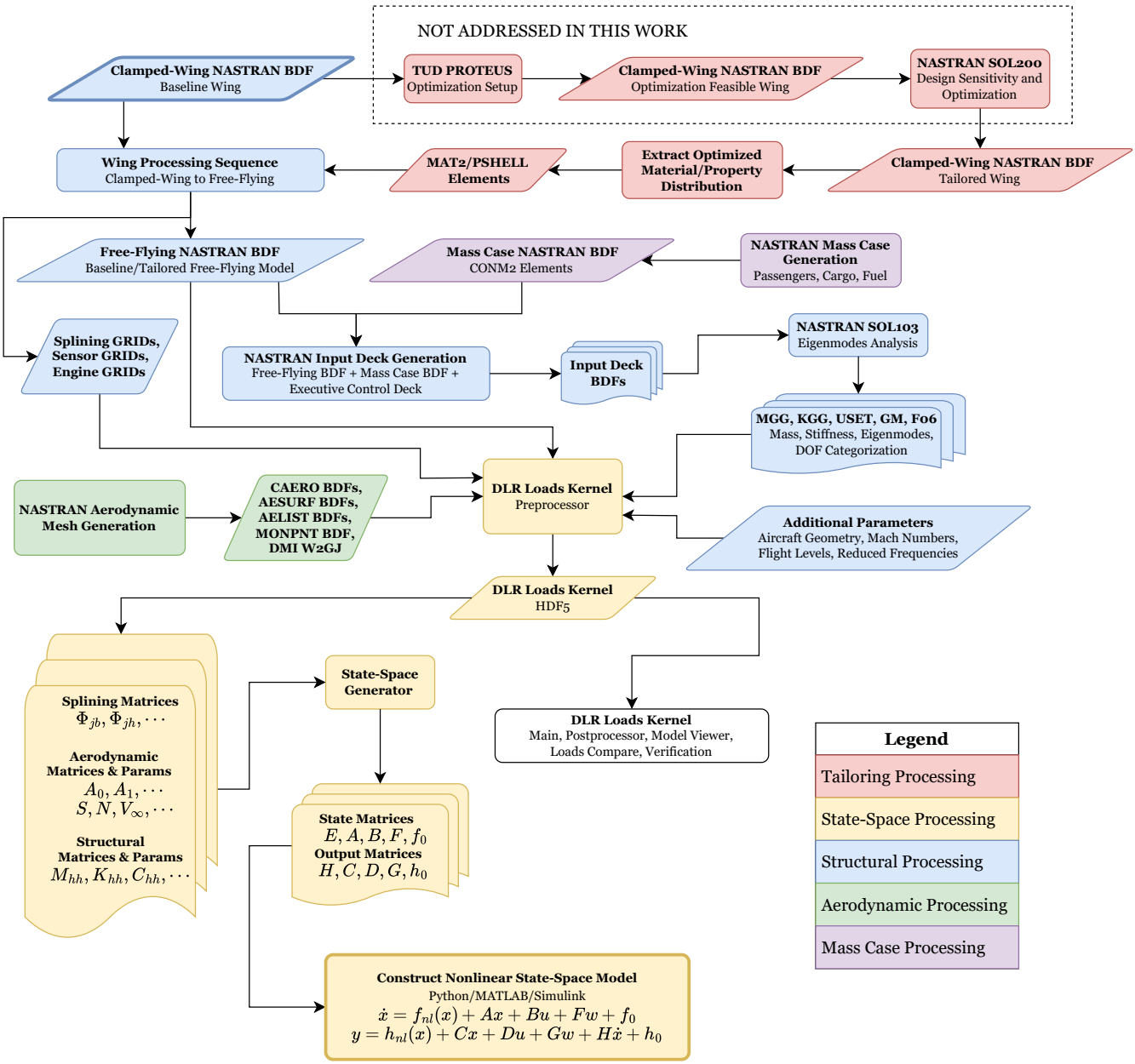


Figure 4.1: Aircraft modeling pipeline flowchart

4.2. CLAMPED-WING FEM TO FREE-FLYING FEM

This section outlines the process for extending clamped-wing finite-element models to free-flying finite-element models. First, the clamped-wing model to which this process is applied, the Embraer Benchmark Wing (EBW) is briefly discussed. Secondly, Section 4.2 discusses the actual methodology for extending clamped-wing models.

4.2.1. EMBRAER BENCHMARK WING

The Embraer Benchmark Wing, shown in Figure 4.2, is a research model provided by Embraer that is used for research on aeroelastic tailoring [21] within an international consortium led by Embraer. It is representative of a regional jet wing similar that on an Embraer E-Jet. The model, which consists of a right clamped-wing FEM, has close to 40k nodes and 50k elements, and is representative of the wing models used in the preliminary design of a regional jet. The elements consist mostly of shell elements, represented using CQUAD4 elements in NASTRAN, as well as beam elements, represented using CBAR elements in NASTRAN. Considering the number of nodes and wide usage of shell elements, it can be considered a high-fidelity industrial model. Aerodynamic loads are applied at the master nodes of RBE3 elements, which distribute the applied loads to the surrounding nodes. The wing can be clamped by means of the RBE2 element at the root, which connects to the nodes at the wing root. Additionally, fuel mass CONM2 elements are available, which can be detached and reattached to the wing structure depending on the desired mass case considered for analysis. Additionally, all relevant masses attached to the wing are modeled as well, such as the engines and the landing gear, also using CONM2 elements. As past research [21], [22] has focused on the aeroelastic tailoring of this wing model, aeroelastically tailored versions of this wing can be generated using tailoring workflows at TU Delft.

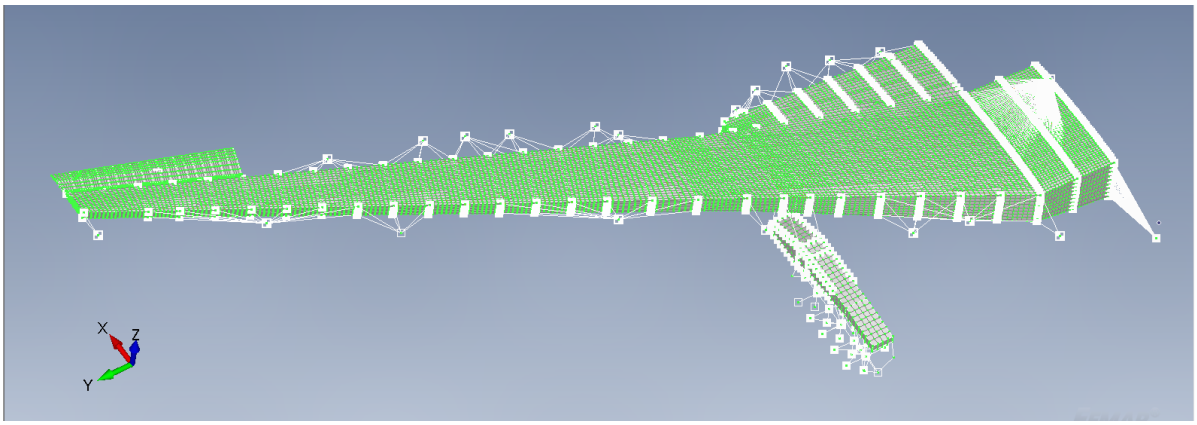


Figure 4.2: Embraer Benchmark Wing provided by Embraer

4.2.2. WING PROCESSING SEQUENCE

The first step in converting a clamped-wing FEM to a free-flying FEM is to process the clamped-wing model. Depending on the clamped-wing model used, some of these steps might be redundant or additional processing steps are required. Additionally, if one uses a full-aircraft FEM instead, most of these steps can be skipped entirely, possibly only having to run a NASTRAN SOL103 to extract the mass matrices, stiffness matrices, and eigenmodes, as well as parameterize and generate an aerodynamic mesh.

The major processing sequence is to mirror the wing, join the wing halves, model the fuselage and empennage as rigid-bodies, and add representative mass and inertia to this model to make it free-flying. Important are two different approaches, both of which were attempted initially, to joining the wings: merging the individual nodes and adjusting the element connectivity, or joining each corresponding node at the root using rigid-body elements. However, both approaches were tried, yielding the same results. In the end, the node merging was chosen as this does not require distinguishing between two sets of nodes at the root. The Bulk Data File (BDF) contains the entire definition of the structural NASTRAN mesh. This BDF file was chosen to be modified programmatically, using PyNastran [108] in Python, as opposed to using a FEM preprocessor, such as MSC Patran and Simcenter Femap. This wields greater control over modifications if one has programming experience and moreover does not require familiarizing with a particular graphical user interface.

Most importantly, an automated workflow allows to generate new free-flying FEMs sharing the same FEM, but with tailoring stiffness and mass properties, with relative ease, as shown in the pipeline and discussed in Subsection 4.2.3. The processing steps are globally defined as follows:

1. **Clean and preprocess the finite-element model.** It is recommended to run an SOL 103 normal modes analysis to identify and resolve critical problems in the clamped-wing model before building the free-flying model in order to identify solver issues before propagating them when mirroring the wing. Additionally, even though this may not cause trouble for the solver, unconnected entities must be either attached, or removed.

2. **Define and enforce a consistent wing root interface.** To enable a physically meaningful connection between the wing halves, the structural nodes defining the wing root interface must overlap once the wing is mirrored in the next step. However, in the clamped-wing model the GRIDs may not be exactly on the symmetry plane yet, as there may be some offset. Therefore, the GRIDs belonging to the root interface are identified based on their proximity to the symmetry plane based on a specified tolerance ($y<0.025$) and are shifted to the symmetry plane.

3. **Generate a full-wing model by mirroring and establish mapping.** The full structural mesh is mirrored about the symmetry xz-plane to generate the other wing half. In PyNastran [108], a dedicated method is included for this purpose, which was used in this process. This results in a full-wing model, yet unconnected. A problem specifically with the PyNastran [108] mirroring function is that only nodes and elements are mirrored. As a result, when the tailored mass and stiffness distribution is fed into the model in later on, the shell CQUAD4 elements of the mirrored wing still reference the right wing material coordinate system, known as MCIDs, which is incorrect as the tailored wing has anisotropic material properties, and thus differ between the left and right wing in terms of direction. Therefore, the mirroring step should also copy and mirror the MCIDs. Additionally, PyNastran [108] has an issue where instead of flipping the element normals on the mirrored wing, it flips the normals on the original wing, which also needs to be corrected. Similarly, the node ordering on the mirrored wing needs to be adjusted to ensure that elements which do not have an associated MCID, have the material angle in the correct direction. Additionally, the MAT2 and PSHELL cards also need to be copied, as the G13 and G23 of the MAT2 need to be negated in order to properly represent stress-shear coupling in the mirrored wing.

4. **Join the wing halves to form a continuous structure.** Now, the wing halves can be joined by merging each of the nodes at the root. The node mapping can be obtained by adding the node ID offset between the original and mirrored wing, and adding it to each of the root node IDs of the original wing. The wings can be merged by replacing all references pointing to the root node IDs of the mirrored wing to the root node IDs of the original wing. This must be done for all elements, such as CQUAD4s, RBE2s and RBE3s.

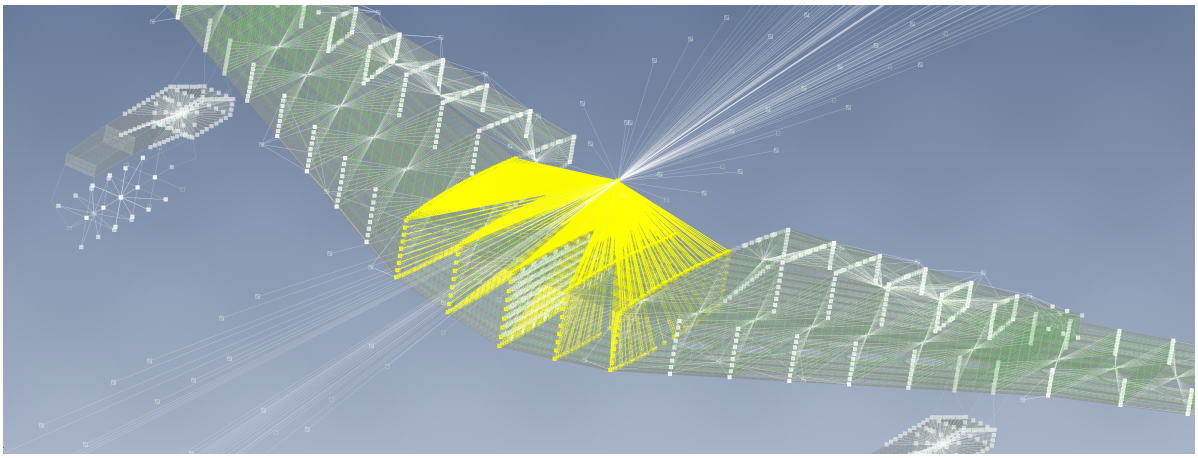


Figure 4.3: Root RBE3 element connecting wing, fuselage, tail and corresponding mass/inertia

5. **Define an attachment reference for non-wing components.** An attachment reference is required to connect the wings and the fuselage and tail. This serves two purposes. First, in the second next step the fuselage and tail are modeled as GRIDs connected using rigid RBE2 elements, which need to be attached to a reference. Secondly, to represent the free-flying dynamics correctly, representative mass and inertia in the third next step is added, which also needs to be attached to a reference GRID. Therefore, the root RBE3 element is redefined, connecting the master node not only to the root region of the right wing, but also to the root region of the left wing, as shown in Figure 4.3.

6. **Align the structural mesh with the reference geometry.** Next, the structural mesh is aligned with the aerodynamic mesh. Initially, this requires some iteration to translate the wingbox to the desired position, but also the aerodynamic mesh also has to be adjusted slighty to better match with the wing structure. It is important that not only the GRIDs are translated, but also the COORDs, for correct definition of the MCIDs.

7. **Represent the fuselage and empennage using rigid elements.** GRIDs are placed along the fuselage centreline and the root and quarter-chord line of the vertical tail and the horizontal tail, which are connected to the master node of the root RBE3 element using RBE2 elements, effectively modeling the fuselage and empennage as rigid bodies. It is explicitly required to place GRIDs along the fuselage centreline and along root and quarter-chord line of the tail planforms, as the nearest neighbour splining method used in Subsection 4.4.2 otherwise yields out-of-plane connections between the aerodynamic and structural mesh.

8. **Augment the model with representative mass and inertia.** To enable proper representation of the flight dynamics of the free-flying aircraft, the wing mass and inertia must be augmented by representative mass and inertia of all masses not present in the wing model. As the Embraer Benchmark Wing already includes all mass that is directly attached to the wings, the to-be-added mass should represent the mass of the fuselage and tail including all inside of it. Fuel, cargo, and passenger mass are handled separately ensuring separation between the structural mesh and mass case definitions, and is discussed in Subsection 4.2.4. The following method was used to model the non-wing mass and inertia. The Operating Empty Weight (OEW) is assumed equal to the Minimum Operating Weight (MOW) from the E195-E2 Airport Planning Manual (APM) [109]. The longitudinal range of the CG as a fraction of the MAC at the Minimum Operating Weight (MOW) is also taken from the APM. From this range, CG at MOW is assumed to at the middle of this range. The vertical CG position is assumed at the middle fuselage, $-h_l + r$, defined in Subsection 4.3.5. Note that this is with respect to the fuselage centreline of the aerodynamic mesh.

MOW	34 700 kg	$OEW \approx MOW$
$x_{CG,MOW}/MAC$	0.24	0.18-0.30
$z_{CG,MOW}$	0.7875 m	assumed at $-h_l + r$

Table 4.1: Approximated CG and mass data for the E195-E2 from the APM [109]

The CG at MOW is expressed as a fraction of the MAC, which therefore requires the longitudinal position of the MAC as well as the MAC length, which have been computed in Table 4.6. The longitudinal CG position is computed as:

$$x_{CG,MOW} = x_{LEMAC} + (x_{CG,MOW}/MAC) \times MAC \tag{4.1}$$

The CG of the aircraft can be computed as being determined by the nonwing mass and arm and wing mass and arm:

$$\boldsymbol{x}_{CG} = \frac{m_{NW}\boldsymbol{x}_{CG,NW} + m_W\boldsymbol{x}_{CG,W}}{m_{NW} + m_W} = \frac{m_{NW}\boldsymbol{x}_{CG,NW} + m_W\boldsymbol{x}_{CG,W}}{m_{MOW}} \tag{4.2}$$

From the which the nonwing CG can be deduced:

$$x_{CG,NW} = \frac{m_{MOW}x_{CG,MOW} - m_W x_{CG,W}}{m_{NW}} \tag{4.3}$$

The nonwing mass is estimated as follows using the wing mass of the wing model, which is computed by using the mass properties functionality in PyNastran [108], or any other FEM preprocessor. This wing mass does not include the fuel masses as these have been removed in the first processing step.

$$m_{NW} = m_{MOW} - m_W \tag{4.4}$$

Lastly, the nonwing inertia is approximated using a thin-walled hollow tube with a radius at 2/3 of the fuselage radius to account for the mass being distributed closer to the fuselage centreline and with a length equal to the fuselage length. Additionally, the parallel-axis theorem is applied to compute the inertia of the hollow tube around the CG of the MOW, with the hollow tube coinciding with the fuselage.

$$\boldsymbol{I} = \begin{bmatrix} \frac{4}{9} m_{NW} r_f^2 & 0 & 0 \\ 0 & m_{NW}\left[\frac{2r_f^2}{9} + \frac{l_f^2}{12} + \left(\frac{l_f}{2} - x_{CG,NW}\right)^2\right] & 0 \\ 0 & 0 & m_{NW}\left[\frac{2r_f^2}{9} + \frac{l_f^2}{12} + \left(\frac{l_f}{2} - x_{CG,NW}\right)^2\right] \end{bmatrix} \tag{4.5}$$

9. **Identify structural nodes for aerostructural coupling.** The aerodynamic forces which are located at the quarter-chord points of the aerodynamic mesh need to be applied to the structure, which brings about the question on how to introduce the loads into the structure: which aerodynamic force should be applied to which structural node for realistic introduction of the loads onto the structure? In aeroelastic modeling in NASTRAN, RBE3 elements, also known as spiders, are used to distribute forces and moments applied at a master node to the slave nodes to which the element is attached, and these elements are also present in the Embraer Benchmark Wing. These RBE3 elements are identified programmatically in PyNastran [108]. Additionally, the aerodynamic mesh should also be properly splined to the fuselage and tail GRIDs, and therefore the GRIDs used for this are also used for splining.

10. **Identify structural nodes for sensing and control.** To design MLA, GLA, and AFS controllers, accelerometers are placed along the wing to measure the local acceleration, which are used as control feedback. In the model, these accelerometers are attached to structural nodes, as this allows describing the local acceleration in terms of the eigenmodes. As such, these nodes are identified in the final step. This step requires one-off manual processing by locating appropriate nodes in a FEM preprocessor. For the Embraer Benchmark Wing, LE and TE pairs of nodes on the top side of the wing are identified for 7 spanwise stations, as shown in Figure 4.4.

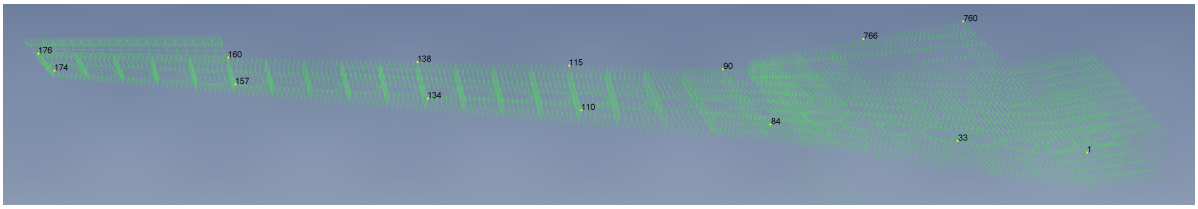


Figure 4.4: Structural nodes used as positions for accelerometers

11. **Copy the baseline wing and read in the aeroelastically tailored wing.** The wing generated in the above steps is from hereon referred to as the baseline wing. The last step is to read in the tailored wing BDF and extract the MAT2 and PSHELL cards containing the optimized material and property distribution and overwrite the same MAT2/PSHELL cards in the baseline wing. This should be done for both the MAT2s and PSHELLs on the original wing as well as on the mirrored wing. Additionally, the G13 and G23 terms should be negated to correctly model the stress-shear coupling in the mirrored wing. This approach is adopted rather than applying the above processing steps directly to the tailored wing BDF, as this ensures that the tailored wing does not pose issues for the NASTRAN solver and is overall easier to implement.

The final step is to create input decks that are consumed by the NASTRAN solver. The generation of input decks is fully automated as well. This consists of generating BDF files with INCLUDEs for the desired free-flying FEM and mass case BDF, in addition the Executive Control Deck, which contains the DMAP ALTERs

for extracting the global mass and stiffness matrix, degree of freedom categorization, and the eigenmodes. On a practical note, DLR Loads Kernel struggles to compute the eigenmodes and eigenfrequencies for the Embraer Benchmark Wing presumably because of the large number of nodes, and therefore the eigenmodes and eigenfrequencies are taken directly from NASTRAN instead.

4.2.3. AEROELASTICALLY TAILORED WINGS

Aeroelastically tailored wings models have been produced in the work of Maathuis, Castro, and De Breuker [22] and used as input in this framework. The tailoring process is out of scope of this work, but is briefly described here. The clamped-wing model is fed into PROTEUS, which generates a feasible wing design that can be fed as input to the NASTRAN SOL200 Design Sensitivity and Optimization sequence, which performs a higher fidelity tailoring process. Each iteration step outputs a feasible wing design BDF, meaning it is within the specified optimization constraints, with further iteration steps providing more optimized wing BDFs. The objective in this optimization is to minimize the wing weight while meeting constraints arising from static strength, local buckling analysis, aeroelastic flutter stability, aileron effectiveness, and angle of attack bounds. The design variables consist of lamination parameters and panel thicknesses for each structural panel, which are optimized using gradient-based methods. Lamination parameters provide a continuous representation of the discrete composite layup, enabling the use of gradient-based methods. The optimized lamination parameters and panel thicknesses are embodied in the wing BDF file in MAT2 and PSHELL cards. These are subsequently extracted from the optimized wing BDF and used to overwrite the original values of the baseline free-flying FEM, yielding a free-flying aeroelastically tailored aircraft. This approach integrates more seamlessly with the wing processing sequence in Section 4.2, as the optimized wing BDF also contains the optimation setup and is quite different from the baseline wing BDF, except for the GRIDs and CQUADs, as these define the wingbox FEM.

4.2.4. MASS CASE GENERATION

Mass cases are generated separately from the structural mesh to separate the mesh definition from the mass case. Masses are modeled as lumped masses using CONM2 elements, which are attached to the master GRID node of the wing root RBE3 element, discussed in Section 4.2. The objective is to be able to demonstrate the effect of mass cases on the load alleviation performance, and only a few representative mass cases are required for this, as opposed to the typically large number of mass cases considered in actual design. In terms of the mass cases, 4 fuel mass cases are defined, and 4 passenger/cargo mass cases are defined, leading to 16 mass cases. A dedicated Python script was built to generate BDF files with the desired mass case, which are included in the BDF file fed to NASTRAN. These are discussed below.

FUEL MASS CASES

The tank capacities were already specified in the Embraer Benchmark Wing and therefore modeling is solely a matter of specifying the fuel mass as a fraction of this capacity. Fuel mass cases can be generated by specifying the fuel loading factor and the fuel loading strategy, which attaches CONM2 to the corresponding GRID of the free-flying model. Implemented loading strategies in the Python script include inboard-outboard, outboard-inboard, and evenly distributed. Four fuel mass cases are used in this work that represent different extremes of fuel loading, displayed in Table 4.2.

Loading	Strategy	Total
100%	evenly distributed	14 630 kg
40%	inboard-outboard	5852 kg
40%	outboard-inboard	5852 kg
0%	evenly distributed	0 kg

Table 4.2: Chosen fuel mass cases for the free-flying aircraft

PAX/CARGO MASS CASES

Passengers are assumed to weigh 84 kilograms in line with EASA standard passenger weights [110]. This approximation includes the cabin bag. Similar to the E195-E2, two cargo holds are implemented. The forward cargo hold has a capacity of 2375 kg and the aft hold of 2555 kg. The position of the cargo holds and the seat position and pitch were estimated from the E195-E2 APM as well [109]. The passenger and cargo parameters have been summarized in Table 4.3. The same is summarized for the cargo compartments in Table 4.4. The implemented Python script allows specifying the passenger loading factor and the loading strategy: randomly allocated, front-to-back, and back-to-front. The cargo loading factor can also be defined, in addition to a

longitudinal shift of the cargo with respect to the centre of each cargo hold. Four passenger/cargo mass cases are used in this thesis, which represent full and empty passenger/cargo conditions, as well as forward CG and aft CG passenger/cargo loadings. These are tabulated in Table 4.5. The aft CG and forward CG cases have been illustrated in Figure 4.5.

Description	Symbol	Value
Mass 1 PAX	m_{PAX}	84 kg
Seat rows	n_r	136
Seat pitch	Δx_s	0.737 m
Seat x	x_s	0.737 m
Seat z	z_s	0.9 m
Aisle seat y	y_{as}	0.45 m
Window seat y	y_{ws}	1.02 m

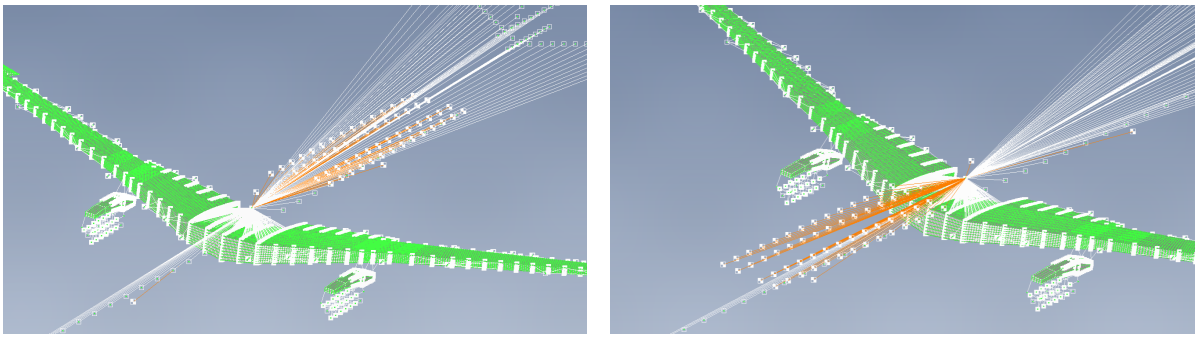
Table 4.3: Seating arrangement and parameters for the free-flying aircraft

Description	Symbol	Value
Aft compartment	m_{aft}	2555 kg
Forward compartment	m_{fwd}	2375 kg
Compartment z	z_c	−0.4 m
Forward compartment x	x_{fwd}	9.75 m
Aft compartment x	x_{aft}	27.87 m

Table 4.4: Cargo hold parameters for the free-flying aircraft

PAX Loading	PAX Strategy	Cargo Loading	Cargo Strategy
100%	random allocation	100%	forward and aft
40%	front-to-back	80%	forward only
40%	back-to-front	80%	aft only
0%	random allocation	0%	forward and aft

Table 4.5: Chosen passenger and cargo mass cases for the free-flying aircraft



(a) Passengers loaded back-to-front

(b) Passengers loaded front-to-back

Figure 4.5: Passengers loading strategies in the free-flying aircraft

4.3. AERODYNAMIC MESH PARAMETERIZATION

The first step is to generate an aerodynamic mesh, as FEMs often do not come with the aerodynamic mesh. In NASTRAN, the panel aerodynamic mesh is defined using CAERO1 cards for lifting surfaces. Control surfaces are defined using AESURF cards that reference a set of aerodynamic boxes using AELIST cards. Additionally, the hinge line of the control surface is specified using a local coordinate system, for which the CORD2R card is used. It is recommended for new aircraft configurations to generate a custom parameterization. However, the particular parameterization presented in this section can be taken as a guiding example. The parameterization of the wings is presented in Subsection 4.3.1, in which also the wing reference quantities are computed. The same is outlined for the horizontal and vertical tail in Figure 4.7 and Subsection 4.3.3, respectively. The engines/nacelles are defined in Subsection 4.3.4. The cruciform fuselage is parameterized in Subsection 4.3.5. The definition of the monitoring stations, which are used to extract the cuting forces and moments at these locations, is explained in Subsection 4.3.6. Lastly, the method for adding a camber distribution to the wings is highlighted in Subsection 4.3.7. The geometry of the free-flying aeroservoelastic regional jet is based on the Embraer E195-E2. Although not extremely accurate, digital measurements and appropriate scaling from the

Airport Planning Manual (APM) of the E195-E2 [109] provide a decent estimate of the aircraft geometry. As at the same time, the mesh should also be made to fit to the Embraer Benchmark Wing, which lies at the basis of the free-flying FEM generated in Section 4.2, the parameter values were iteratively adjusted until a good balance was achieved between the measured E195-E2 aircraft geometry from the APM, the fit with the Embraer Benchmark Wing FEM, and the visual similarity with the actual E195-E2 aircraft. The final aerodynamic mesh of the regional jet is presented in Subsection 4.4.1.

4.3.1. WINGS, FLAPS, AND AILERONS

Table 4.6 shows the parameters and final values of the wing parameterization. The defining points and lengths required to define the CAERO, AESURF, and AELIST cards are computed with Equation 4.9 till Equation 4.20. Four trailing edge control surfaces are defined to allow effective maneuver load alleviation on this aircraft in future work.

Description	Symbol	Value
Wing root lateral position	y_w	$(w_f/2)$
Wing root longitudinal position	x_w	15.36 m
Wing pylon/yehudi lateral position	y_p	4.93 m
Wing tip lateral position	y_t	15.21 m
Wing tip-to-tip span	b	35.12 m
Wing root chord length	c_r	5.37 m
Wing tip chord length	c_t	1.44 m
Wing LE main sweep angle	$\Lambda_{w,LE}$	26.71 °
Wing LE tip sweep angle	$\Lambda_{t,LE}$	34.21 °
Inboard wing dihedral angle	Γ_{in}	6.90 °
Outboard wing dihedral angle	Γ_{out}	3.88 °
Wing chordwise panels	$n_{w,c}$	11
Inboard spanwise panels	$n_{i,s}$	10
Outboard spanwise panels	$n_{o,s}$	42
Tip spanwise panels	$n_{t,s}$	22
Flap/aileron chordwise panels	$n_{a,c}$	3
Inboard flap spanwise region	$n_{a1,s}$	(0,10)
Outboard flap spanwise region	$n_{a2,s}$	(0,14)
Inboard aileron spanwise region	$n_{a3,s}$	(14,28)
Outboard aileron spanwise region	$n_{a4,s}$	(28,42)

Table 4.6: Wing, flap, and aileron geometric parameters and final values

First, some intermediate values are defined as these are used repeatedly. Equation 4.6 defines the span of the tip section, Equation 4.7 defines the span of the mid/outboard section, and Equation 4.8 defines the span of the yehudi/inboard section.

$$b_t = \frac{b}{2} - y_t \tag{4.6}$$

$$b_m = y_t - y_p \tag{4.7}$$

$$b_y = y_p - y_w \tag{4.8}$$

The defining points are numbered by spanwise station from inboard to outboard. At each spanwise station, the LE point is numbered first, followed by the corresponding TE point. Thus, the defining point pairs $(\boldsymbol{p}_1, \boldsymbol{p}_2)$, $(\boldsymbol{p}_3, \boldsymbol{p}_4)$, $(\boldsymbol{p}_5, \boldsymbol{p}_6)$, and $(\boldsymbol{p}_7, \boldsymbol{p}_8)$ represent the leading- and TE points at successive spanwise stations from root to tip. The coordinates of these points are parameterized as follows:

$$\boldsymbol{p}_1 = \begin{bmatrix} x_w & y_w & 0 \end{bmatrix}^T \tag{4.9}$$

$$\boldsymbol{p}_2 = \boldsymbol{p}_1 + \begin{bmatrix} c_r & 0 & 0 \end{bmatrix}^T \tag{4.10}$$

$$\boldsymbol{p}_3 = \boldsymbol{p}_1 + \begin{bmatrix} b_y \tan\Lambda_{w,LE} & b_y & b_y \tan\Gamma_{in} \end{bmatrix}^T \tag{4.11}$$

$$\boldsymbol{p}_4 = \boldsymbol{p}_2 + \begin{bmatrix} 0 & b_y & b_y \tan\Gamma_{in} \end{bmatrix}^T \tag{4.12}$$

$$\boldsymbol{p}_5 = \boldsymbol{p}_3 + \begin{bmatrix} b_m \tan\Lambda_{w,LE} & b_m & b_m \tan\Gamma_{out} \end{bmatrix}^T \quad (4.13)$$

$$\boldsymbol{p}_6 = \boldsymbol{p}_5 + \begin{bmatrix} c_t & 0 & 0 \end{bmatrix}^T \quad (4.14)$$

$$\boldsymbol{p}_7 = \boldsymbol{p}_5 + \begin{bmatrix} b_t \tan\Lambda_{t,LE} & b_t & b_t \tan\Gamma_{out} \end{bmatrix}^T \quad (4.15)$$

$$\boldsymbol{p}_8 = \boldsymbol{p}_6 + \begin{bmatrix} \left(\frac{x_6 - x_4}{b_m}\right) b_t & b_t & b_t \tan\Gamma_{out} \end{bmatrix}^T \quad (4.16)$$

In order to define control surfaces, the aerodynamic boxes that are to be used need to be identified. Additionally, the hinge line needs to be specified. Thus, the three points defining the leading edges of all flaps/ailerons need to be computed. Only three are required, as all trailing edge surfaces in the outboard section share the same hinge line. The hinge points are defined right at the chordwise positions where the aerodynamic boxes defined as surfaces start.

$$\boldsymbol{p}_{a1} = \boldsymbol{p}_1 + \begin{bmatrix} l_{12}\left(1 - \frac{n_{a,c}}{n_{w,c}}\right) & 0 & 0 \end{bmatrix}^T \quad (4.17)$$

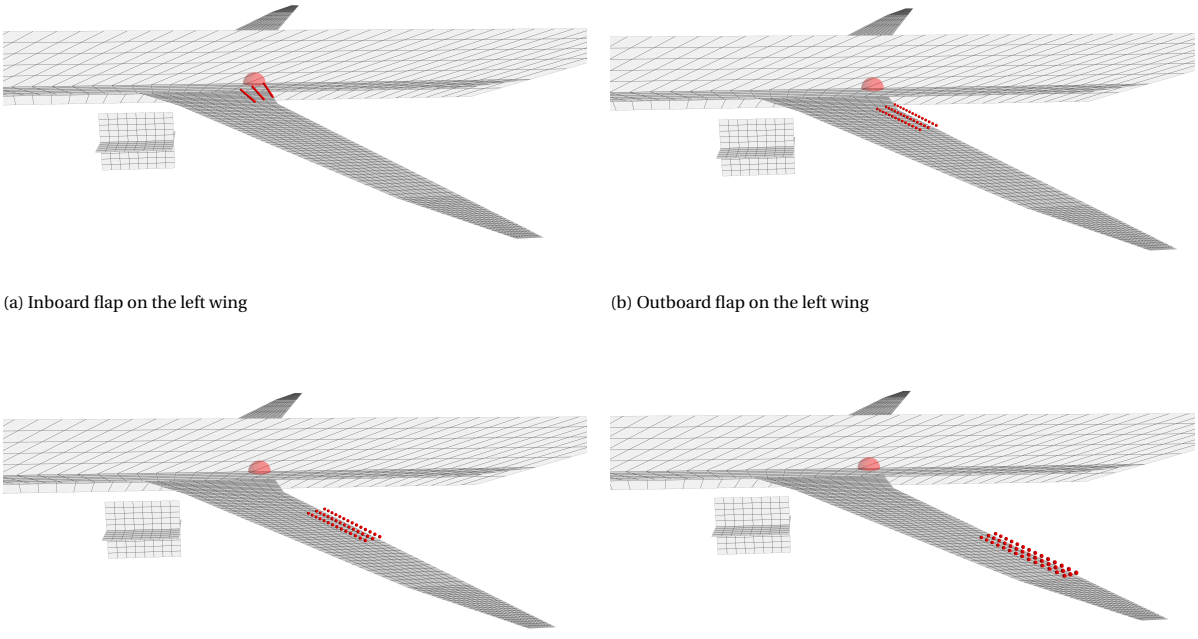
$$\boldsymbol{p}_{a2} = \boldsymbol{p}_3 + \begin{bmatrix} l_{34}\left(1 - \frac{n_{a,c}}{n_{w,c}}\right) & 0 & 0 \end{bmatrix}^T \quad (4.18)$$

$$\boldsymbol{p}_{a3} = \boldsymbol{p}_5 + \begin{bmatrix} l_{56}\left(1 - \frac{n_{a,c}}{n_{w,c}}\right) & 0 & 0 \end{bmatrix}^T \quad (4.19)$$

The chord lengths required for the CAERO definitions are defined as the difference in x-positions of the defining outer points of the wing parameterization:

$$l_{12} = x_2 - x_1 \qquad l_{34} = x_4 - x_3 \qquad l_{56} = x_6 - x_5 \qquad l_{78} = x_8 - x_7 \quad (4.20)$$

The resulting four flap/aileron control surfaces on the wing are shown in Figure 4.6. Naturally, identical control surfaces are present on the right wing. The surfaces in these figures are shown for a deflection angle of -10 degrees.



(a) Inboard flap on the left wing

(b) Outboard flap on the left wing

(c) Inboard aileron on the left wing

(d) Outboard aileron on the left wing

Figure 4.6: Flap/aileron control surfaces on the trailing edge of the left wing

WING REFERENCE QUANTITIES

The area of a panel p is computed with Equation 4.21. The taper ratio of panel p is computed with Equation 4.22. The mean aerodynamic chord of panel p is computed with Equation 4.23. The spanwise position of the leading edge of the mean aerodynamic chord with respect to the panel inboard edge is computed with Equation 4.24.

$$S_p = \frac{1}{2}(c_r + c_t)\, b_p \tag{4.21}$$

$$\lambda_p = \frac{c_{t_p}}{c_{r_p}} \tag{4.22}$$

$$MAC_p = \frac{2}{3} c_r \frac{1+\lambda_p+\lambda_p^2}{1+\lambda_p} \tag{4.23}$$

$$y_{LEMAC_p} = \frac{b_p}{3}\frac{1+2\lambda_p}{1+\lambda_p} \tag{4.24}$$

By computing the above quantities for each panel on one wing, the mean aerodynamic chord of the wing, the longitudinal position of the leading edge of the mean aerodynamic chord, as well as the reference area of the wing can be computed. The MAC is the area-weighted average of the panel MACs. The spanwise position of the leading edge of the MAC is the weighted average of the panel LEMAC locations, with weights $MAC_j S_j$. Note that the fuselage portion of the wing must also be accounted for in the wing reference quantities.

$$S = S_i + S_o + S_t \tag{4.25}$$

$$MAC = \frac{MAC_i S_i + MAC_o S_o + MAC_t S_t}{S_i + S_o + S_t} \tag{4.26}$$

$$y_{LEMAC} = \frac{MAC_i S_i y_{i,LEMAC} + MAC_o S_o y_{o,LEMAC} + MAC_t S_t y_{t,LEMAC}}{MAC_i S_i + MAC_o S_o + MAC_t S_t} \tag{4.27}$$

The longitudinal position of the LEMAC is computed with Equation 4.29, which also requires accounting for the fuselage portion of the wing reference area using Δx. The final wing reference quantities for the free-flying aircraft are tabulated in Table 4.7.

$$\Delta x = y_w \tan \Lambda_{w,LE} \tag{4.28}$$

$$x_{LEMAC} = x_w - \Delta x + y_{LEMAC} \tan(\Lambda) \tag{4.29}$$

$$AR = \frac{b^2}{2S} \tag{4.30}$$

Reference	Value
AR	11.87
S	51.95 m^2
MAC	3.63 m
x_{LEMAC}	17.19 m

Table 4.7: Wing reference quantities of the free-flying aircraft

4.3.2. HORIZONTAL TAIL AND ELEVATOR

The horizontal tail parameterization and final values are presented in Table 4.8.

Parameter	Symbol	Value
Horizontal tail longitudinal position	x_h	36.39 m
Horizontal tail vertical position	z_h	1.57 m
Horizontal tail root chord	$c_{h,r}$	3.02 m
Horizontal tail tip chord	$c_{h,t}$	1.16 m
Horizontal tail span	b_h	9.80 m
Horizontal tail dihedral angle	Γ_h	6.90 °
Horizontal tail sweep angle	Λ_h	34.5 °
Horizontal tail chordwise panels	$n_{h,c}$	7
Horizontal tail spanwise panels	$n_{h,s}$	15
Elevator chordwise panels	$n_{e,c}$	3

Table 4.8: Horizontal tail and elevator parameters and final values

The horizontal tail is also defined using points on the outer perimeter of the horizontal tail. Similar to the wings, these are numbered by spanwise station from root to tip, and at each spanwise station the LE point is numbered first, followed by the TE point. This results in two pair points $(\boldsymbol{p}_1, \boldsymbol{p}_2), (\boldsymbol{p}_3, \boldsymbol{p}_4)$ which define the root LE and TE points and the tip LE and TE points respectively. The parameterization of these points is:

$$\boldsymbol{p}_1 = \begin{bmatrix} x_h & 0 & z_h \end{bmatrix}^T \tag{4.31}$$

$$\boldsymbol{p}_2 = \boldsymbol{p}_1 + \begin{bmatrix} c_{h,r} & 0 & 0 \end{bmatrix}^T \tag{4.32}$$

$$\boldsymbol{p}_3 = \boldsymbol{p}_1 + \begin{bmatrix} \frac{b_h}{2}\tan\Lambda_h & \frac{b_h}{2} & \frac{b_h}{2}\tan\Gamma_h \end{bmatrix}^T \tag{4.33}$$

$$\boldsymbol{p}_4 = \boldsymbol{p}_3 + \begin{bmatrix} c_{h,t} & 0 & 0 \end{bmatrix}^T \tag{4.34}$$

The lengths required for the CAERO definition are defined as the difference in x-position of each LE and TE point pair for both the root and the tip, which are naturally equal to the horizontal tail root and tip chords:

$$l_{12} = x_2 - x_1 = c_{h,r} \tag{4.35}$$

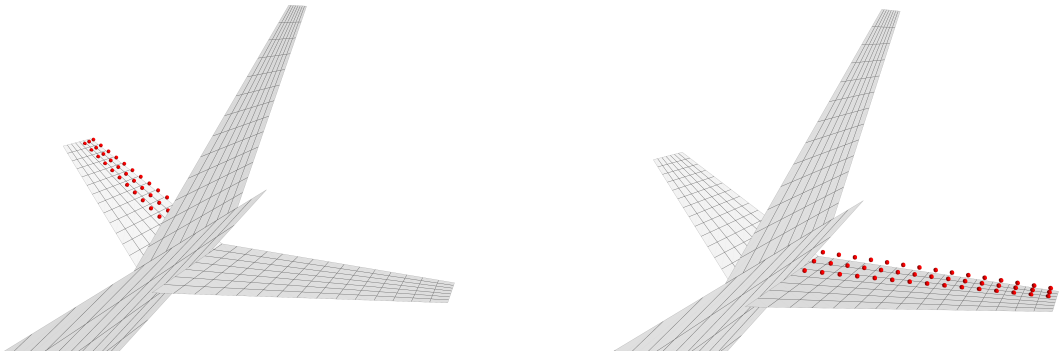
$$l_{34} = x_4 - x_3 = c_{h,t} \tag{4.36}$$

For the elevator control surface, the elevator hinge line must be specified besides the aerodynamic boxes that are used as elevator. The hinge line is defined again by two hinge points, one at the root and one at the tip of the horizontal tail. These hinge point are defined at the chordwise positions where the aerodynamic boxes start that are used as elevator surface:

$$\boldsymbol{p}_{e,r} = \boldsymbol{p}_1 + \begin{bmatrix} c_{h,r}\left(1 - \frac{n_{e,c}}{n_{h,c}}\right) & 0 & 0 \end{bmatrix}^T \tag{4.37}$$

$$\boldsymbol{p}_{e,t} = \boldsymbol{p}_3 + \begin{bmatrix} c_{h,t}\left(1 - \frac{n_{e,c}}{n_{h,c}}\right) & 0 & 0 \end{bmatrix}^T \tag{4.38}$$

Lastly, Figure 4.7 shows the resulting horizontal tail with the left and right elevators shown deflected to -10 degrees.



(a) Right elevator surface

(b) Left elevator surface

Figure 4.7: Horizontal tail aerodynamic panel and elevator control surfaces

4.3.3. VERTICAL TAIL AND RUDDER

The vertical tail is mounted on top of the fuselage. The parameters and final values of the vertical tail and rudder are described in Table 4.9. The vertical tail is defined similarly to the horizontal tail.

Parameter	Symbol	Value
Vertical tail vertical position	z_v	$z_{f,u}$
Vertical tail longitudinal position	x_v	35.52 m
Vertical tail root chord	$c_{v,r}$	4.29 m
Vertical tail tip chord	$c_{v,t}$	1.16 m
Vertical tail span	h_v	4.58 m
Vertical tail sweep angle	Λ_v	42.8 °
Vertical tail chordwise panels	$n_{v,c}$	8
Vertical tail spanwise panels	$n_{v,s}$	13
Rudder chordwise panels	n_r	3

Table 4.9: Vertical tail and rudder parameters and final values

Similar to the other lifting surfaces, the defining points are numbered by spanwise station from inboard to outboard. At each spanwise station, the LE point is numbered first, followed by the corresponding TE point. This results in the point pairs $(\boldsymbol{p}_1, \boldsymbol{p}_2), (\boldsymbol{p}_3, \boldsymbol{p}_4)$ which represent the LE and TE points of the root and the tip of the vertical tail. The parameterization of these points is expressed as:

$$\boldsymbol{p}_1 = \begin{bmatrix} x_v & 0 & z_v \end{bmatrix}^T \tag{4.39}$$

$$\boldsymbol{p}_2 = \boldsymbol{p}_1 + \begin{bmatrix} c_{v,r} & 0 & 0 \end{bmatrix}^T \tag{4.40}$$

$$\boldsymbol{p}_3 = \boldsymbol{p}_1 + \begin{bmatrix} h_v \tan\Lambda_v & 0 & h_v \end{bmatrix}^T \tag{4.41}$$

$$\boldsymbol{p}_4 = \boldsymbol{p}_3 + \begin{bmatrix} c_{v,t} & 0 & 0 \end{bmatrix}^T \tag{4.42}$$

The chord lengths required for the CAERO card definition are defined as the difference in x-position of each of the LE and TE point pairs, which are equal to the root and tip chord of the vertical tail:

$$l_{12} = x_2 - x_1 = c_{v,r} \tag{4.43}$$

$$l_{34} = x_4 - x_3 = c_{v,t} \tag{4.44}$$

The hinge line of the rudder also needs to be defined. In a similar fashion as for the trailing edge surfaces on the wing and the elevator surfaces, the points which are used to define the hinge line start right at the chordwise positions where the aerodynamic boxes start that are defined as belonging to the rudder surface. The root and tip points of the hinge line are respectively:

$$\boldsymbol{p}_{r,r} = \boldsymbol{p}_1 + \begin{bmatrix} c_{v,r}\left(1 - \dfrac{n_{rud}}{n_{v,c}}\right) & 0 & 0 \end{bmatrix}^T \tag{4.45}$$

$$\boldsymbol{p}_{r,t} = \boldsymbol{p}_3 + \begin{bmatrix} c_{v,t}\left(1 - \dfrac{n_{rud}}{n_{v,c}}\right) & 0 & 0 \end{bmatrix}^T \tag{4.46}$$

Finally, Figure 4.8 shows the resulting vertical tail and the rudder control surface, which in the figure is deflected to -10 degrees.

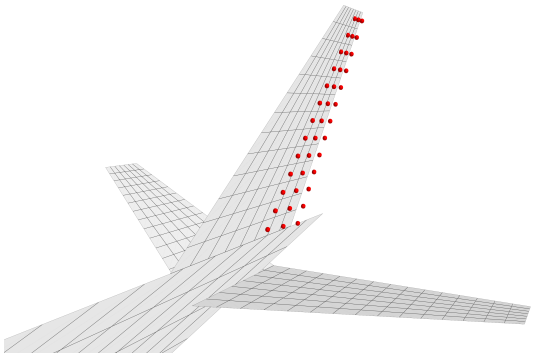


Figure 4.8: Vertical tail aerodynamic panel and rudder control surface

4.3.4. ENGINES/NACELLES

Table 4.10 shows the parameters and final values of the engine/nacelle parameterization. The engine/nacelle resulting aerodynamic panels are shown in Figure 4.9.

Parameter	Symbol	Value
Engine lateral position	y_e	y_p
Engine longitudinal position	x_e	14.3 m
Engine vertical position	z_e	0.186 m
Engine length	l_e	2.62 m
Engine radius	r_e	1.08 m
Engine chordwise panels	$n_{n,c}$	8
Engine spanwise panels	$n_{n,s}$	5

Table 4.10: Engine/nacelles parameters and final values

The engine reference position is defined by Equation 4.47. The intersection point of the engine panels is given by Equation 4.48. The inboard and outboard positions are given by Equation 4.49 and Equation 4.50. The upper point is equal to the reference position and the lower point is equal to Equation 4.51.

$$\boldsymbol{p}_e = \begin{bmatrix} x_e & y_e & z_e \end{bmatrix}^T \quad (4.47) \qquad \boldsymbol{p}_c = \boldsymbol{p}_e - \begin{bmatrix} 0 & 0 & r_e \end{bmatrix}^T \quad (4.48)$$

$$\boldsymbol{p}_i = \boldsymbol{p}_c - \begin{bmatrix} 0 & r_e & 0 \end{bmatrix}^T \quad (4.49) \qquad \boldsymbol{p}_o = \boldsymbol{p}_c + \begin{bmatrix} 0 & r_e & 0 \end{bmatrix}^T \quad (4.50) \qquad \boldsymbol{p}_l = \boldsymbol{p}_c - \begin{bmatrix} 0 & 0 & r_e \end{bmatrix}^T \quad (4.51)$$

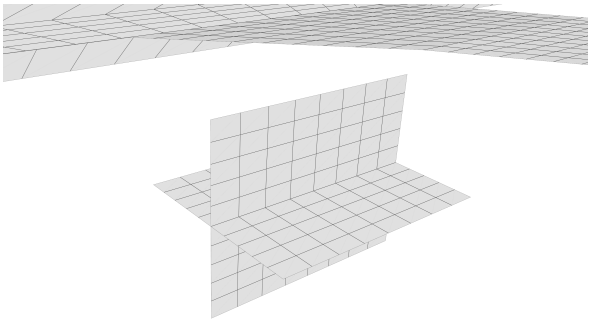


Figure 4.9: Engine/nacelle aerodynamic panels

4.3.5. CRUCIFORM FUSELAGE

Table 4.11 shows the parameters used to describe the cruciform fuselage and the final values. It is a common convention to approximate the fuselage using a cruciform shape when using DLM as most DLM implementations are not able to properly model non-lifting bodies in the flow field.

Parameter	Symbol	Value
Fuselage length	l_f	41.60 m
Fuselage width	w_f	2.77 m
Fuselage height	h_f	3.15 m
Upper/lower fuselage ratio	λ_f	0.25
Vertical tail cone angle w.r.t centreline	$\kappa_{t,v}$	15.5 °
Horizontal tail cone angle w.r.t centreline	$\kappa_{t,h}$	20 °
Upper nose cone angle w.r.t centreline	$\kappa_{n,uv}$	35 °
Lower nose cone angle w.r.t centreline	$\kappa_{n,lv}$	30 °
Side nose cone angle w.r.t centreline	$\kappa_{n,h}$	20 °
Fuselage chordwise panels	n_f	40
Lower fuselage spanwise panels	n_l	2
Upper fuselage spanwise panels	n_u	5
Fuselage sides spanwise panels	n_s	5

Table 4.11: Cruciform fuselage parameters and final values

The lower and upper fuselage heights are defined using the upper/lower fuselage ratio λ_f:

$$h_l = \lambda_f h_f \quad (4.52) \qquad h_u = \left(1 - \lambda_f\right) h_f \quad (4.53)$$

The nose is linearly tapered from the fuselage centerline to the full fuselage dimensions. The longitudinal extents of the nose for the lower fuselage, upper fuselage, and fuselage sides are defined directly from the nose cone angles:

$$x_{n,l} = \frac{h_l}{\tan\left(\kappa_{n,lv}\right)} \quad (4.54) \qquad x_{n,u} = \frac{h_u}{\tan\left(\kappa_{n,uv}\right)} \quad (4.55) \qquad x_{n,s} = \frac{w_f/2}{\tan\left(\kappa_{n,h}\right)} \quad (4.56)$$

The tail geometry is defined analogously. The longitudinal extents of the tail are computed using the tail cone angles, which for the lower tail, upper tail, and sides of the tail are, respectively:

$$x_{t,l} = \frac{h_l}{\tan(\kappa_{t,v})} \quad (4.57) \qquad x_{t,s} = \frac{w_f/2}{\tan(\kappa_{t,h})} \quad (4.58) \qquad x_{t,u} = \frac{h_u}{\tan(\kappa_{t,v})} \quad (4.59)$$

Using these dimensions, the defining points of the cruciform fuselage geometry are computed. The nose apex is located at the origin, Equation 4.60. The lower fuselage corner point is given by. The fuselage side corner point is defined as. The upper fuselage corner point is defined by

$$\boldsymbol{p}_0 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T \quad (4.60) \qquad \boldsymbol{p}_1 = \begin{bmatrix} x_{n,l} & 0 & -h_l \end{bmatrix}^T \quad (4.61)$$

$$\boldsymbol{p}_2 = \begin{bmatrix} x_{n,s} & w_f/2 & 0 \end{bmatrix}^T \quad (4.62) \qquad \boldsymbol{p}_3 = \begin{bmatrix} x_{n,u} & 0 & h_u \end{bmatrix}^T \quad (4.63)$$

The effective fuselage reference length l_{123} is computed by subtracting the upper tail cone projection from the total fuselage length. Figure 4.10 shows the resulting cruciform fuselage.

$$l_{123} = l_f - x_{t,u} \quad (4.64) \qquad l_1 = l_{123} - x_{n,l} - x_{t,l} \quad (4.65)$$

$$l_2 = l_{123} - x_{n,s} - x_{t,s} \quad (4.66) \qquad l_3 = l_f - x_{n,u} \quad (4.67)$$

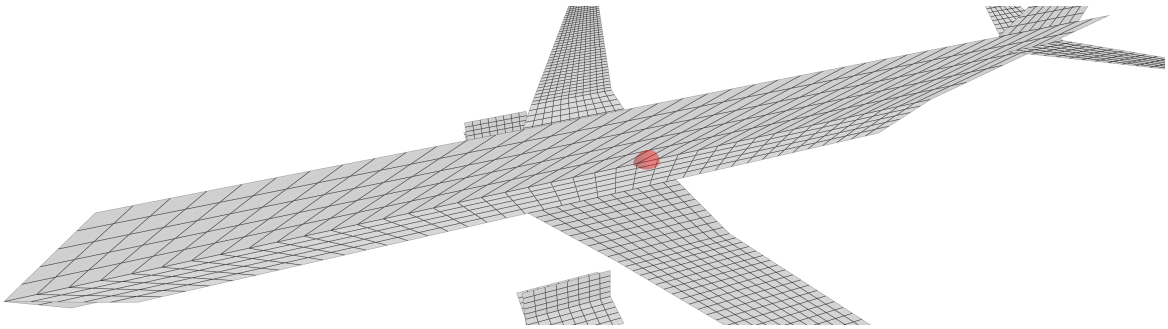


Figure 4.10: Cruciform fuselage aerodynamic panels

4.3.6. MONITORING STATIONS

In gust and maneuver load alleviation, the cutting forces and moments are required to be monitored in order to evaluate the effectiveness of passive and active load alleviation. In pursuit of this objective, monitoring stations are placed along the span of the wing, at which one can extract the WRBM and WRTM. Particularly, these monitoring stations are placed along the Loads Reference Axis (LRA). This LRA is defined here as the line running along the span of the wing at 50% of the local chord, starting at the wing-fuselage intersection. Although this step could be viewed as more closely related to Section 4.2, it requires the definition of the aerodynamic mesh in order to define the LRA along the span at a certain percent of the local chord. The LRA is defined using linear interpolation between the following three points of the wing parameterization:

$$\boldsymbol{LRA_1} = \boldsymbol{p}_1 + \begin{bmatrix} l_{12}\left(\frac{x}{c}\right)_{LRA} & 0 & 0 \end{bmatrix}^T \quad (4.68) \qquad \boldsymbol{LRA_2} = \boldsymbol{p}_3 + \begin{bmatrix} l_{34}\left(\frac{x}{c}\right)_{LRA} & 0 & 0 \end{bmatrix}^T \quad (4.69)$$

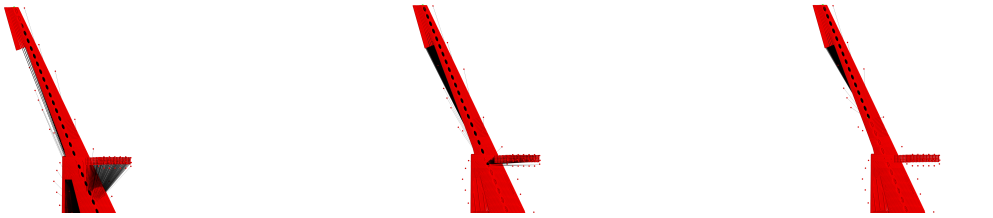
$$\boldsymbol{LRA_2} = \boldsymbol{p}_5 + \begin{bmatrix} l_{56}\left(\frac{x}{c}\right)_{LRA} & 0 & 0 \end{bmatrix}^T \quad (4.70)$$

The longitudinal and vertical position of the load reference axis can then be obtained based on the spanwise position. For this particular aircraft, this equation is evaluated at 10 equally-spaced spanwise stations along the span of each wing.

$$\begin{pmatrix} x \\ y \end{pmatrix} = LRA(y) \quad (4.71)$$

At each of these spanwise points along the LRA a monitoring station is placed using the MONPNT card in NASTRAN. This MONPNT card in turn references an AECOMP card which references a SET1 card. This SET1

card then lists all GRIDs, which define the structural nodes, that are associated with that monitoring station. For the right wing this entails gathering all nodes to the right of that monitoring station, and vice versa for the left wing. Care must be taken to accidentally not include nodes that do not belong to the wing. Figure 4.11 shows for monitoring stations 0, 2, and 4 on the right wing what nodes these stations reference.



(a) Monitoring station 0 connections (b) Monitoring station 2 connections (c) Monitoring station 4 connections

Figure 4.11: Right wing monitoring station connections to structural grid

4.3.7. CAMBER AND TWIST

Camber and twist can be specified through a DMI W2GJ input in NASTRAN, which requires specifying the flow deflection angle due to camber and twist at each aerodynamic box. The flow deflection angle is in fact equal to the negative of the derivative of the camber line:

$$\theta = -\frac{dz}{dx} \tag{4.72}$$

$$\frac{dz}{dx} = \begin{cases} \frac{2m}{p^2}\left(p - \frac{x}{c}\right) & \frac{x}{c} \leq p \\ \frac{2m}{(1-p)^2}\left(p - \frac{x}{c}\right) & \frac{x}{c} > p \end{cases} \tag{4.73}$$

As the mean camber line, as well as its derivative with respect to the chordwise position, of NACA 4-digit airfoils can be expressed using closed-form equations, these airfoils enable an elegant way of adding camber to each aerodynamic box. Other airfoils whose camber line and derivative can also be expressed using closed-form equations do exist. However, in panel aerodynamics, the desired aerodynamic characteristics of more complicated airfoil shapes cannot be resolved and are therefore unnecessary. The camber line derivative of NACA 4-digit airfoils is computed using Equation 4.73 with m the chord-normalized maximum camber and p the chord-normalized position of maximum camber. For the free-flying regional jet, the NACA 24XX is used, and thus $m = 0.02$ and $p = 0.4$, with XX indicating that the thickness is not modeled in panel aerodynamics. The regional jet with the resulting camber distribution is shown in Figure 4.12.

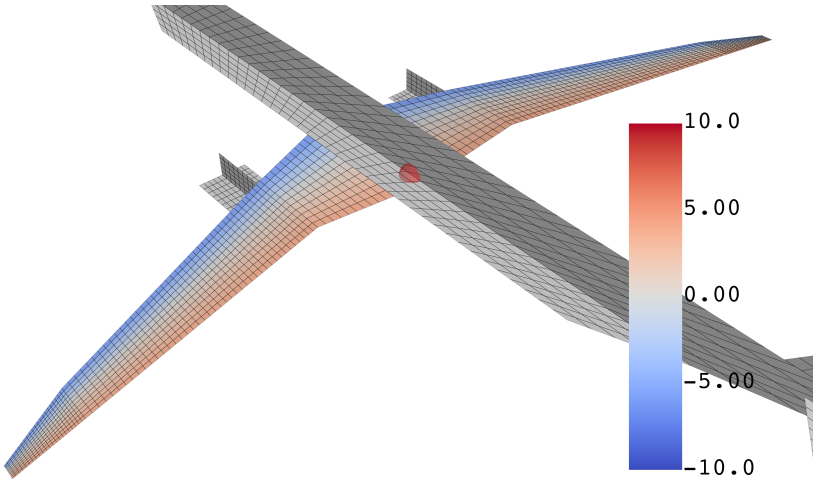


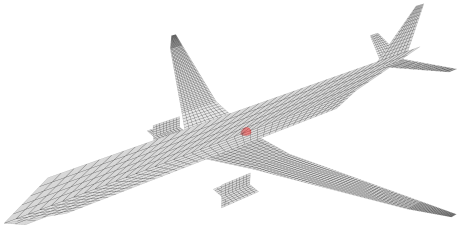
Figure 4.12: Camber distribution along the wing

4.4. RESULTING FREE-FLYING AEROELASTIC MODELS & VALIDATION

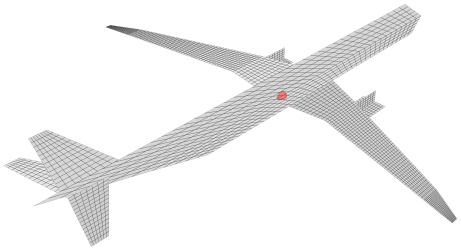
This section discusses the resulting free-flying aeroelastic model that can be generated using the modeling pipeline, for both the baseline aircraft and the tailored aircraft. Subsection 4.4.1 describes the aerodynamic model. Subsection 4.4.2 displays the splining model. Subsection 4.4.3 shows the mass model, the structural model, as well as the eigenmodes, and eigenfrequencies.

4.4.1. PANEL AERODYNAMIC MODEL

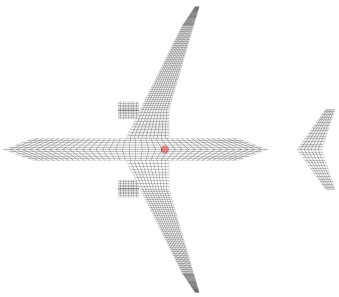
The final aerodynamic mesh is shown from several angles in Figure 4.13. The aerodynamic matrices were generated using Loads Kernel [20] for three flight conditions in the cruise: a low cruise Mach number of 0.70, a typical cruise Mach number of 0.78, and the maximum cruise Mach number of the E195-E2 of 0.82. At these flight conditions, unsteady AIC matrices were computed for 8 reduced frequencies: 0.001, 0.1, 0.3, 0.6, 1.0, 1.5, 2.0, 3.0. As load alleviation typically concerns relatively low reduced frequencies, this range is acceptable. The RFA was performed by the Loads Kernel preprocessor as well using 4 RFA poles, in line with the equations specified in Subsection 3.5.2.



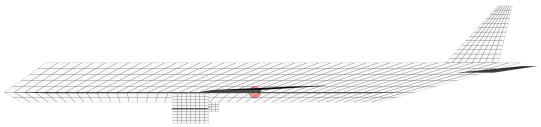
(a) Aerodynamic mesh view from the front left and above



(b) Aerodynamic mesh view from the aft right and above



(c) Aerodynamic mesh view from the top



(d) Aerodynamic mesh view from the side

Figure 4.13: Aerodynamic mesh of the free-flying aircraft

DLR Loads Kernel [19], [20], which is used in the modeling pipeline as discussed in Section 4.1, requires the panel normals to be defined from left to right for horizontal surfaces and bottom to top for vertical surfaces. Adherence to this requirement can be verified by the direction of the panel normals, which are shown in Figure 4.14, which shows the aerodynamic mesh is correctly implemented. Another important consideration for the aerodynamic mesh discretization concerns the aspect ratio of the individual aerodynamic boxes, which should not be larger than 3 or 4. Exceeding this aspect ratio has been shown to undermine the numerical reliability of the Doublet Lattice Method [20]. Moreover, practical experience in this work has shown that a symptom of high panel aspect ratios subsequently degrades the quality of the RFA fit, leading to spurious looping behavior instead of a smooth and accurate fit. Another important requirement is that near-planar aerodynamic panels which have an overlapping wake with other panels should have the same spanwise discretization, meaning that the wake essentially aligns with the other panel. This aircraft does have near-planar panels, but not ones meeting this condition and this requirement was adhered to automatically.

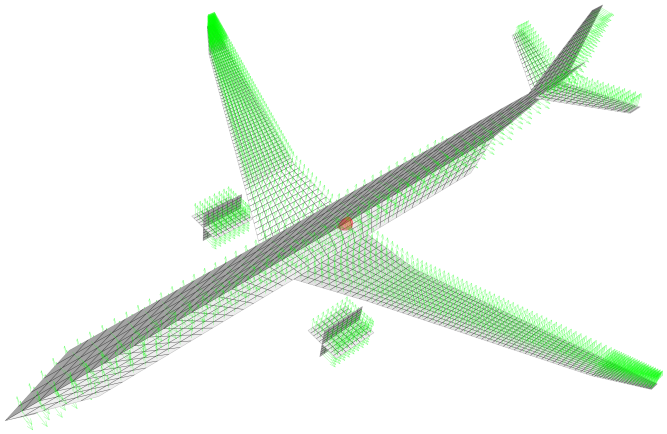
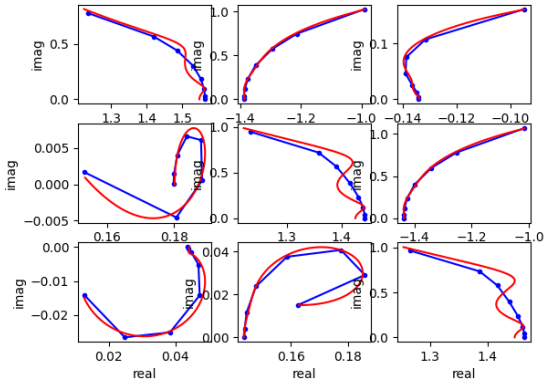
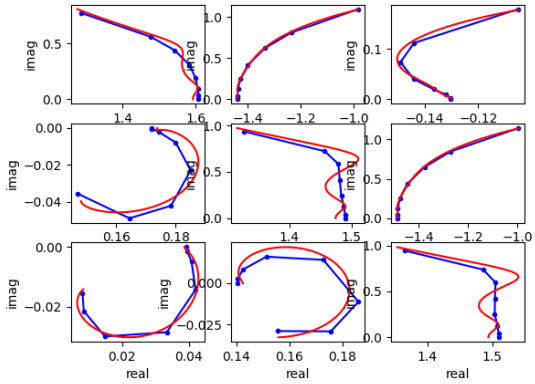


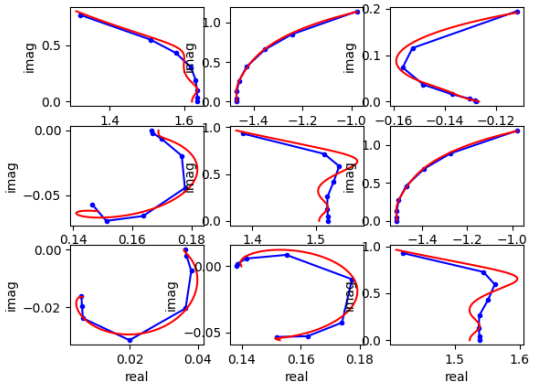
Figure 4.14: Panel normal vectors of the aerodynamic mesh of the free-flying aircraft



(a) Exact and approximated first 3x3 entries AIC Matrix M=0.70



(b) Exact and approximated first 3x3 entries AIC Matrix M=0.78



(c) Exact and approximated first 3x3 entries AIC matrix M=0.82

Figure 4.15: Exact (=blue) and approximated/fit (=red) first 3x3 entries AIC matrices

The RFA performed by Loads Kernel achieves an acceptable fit as can be observed from Figure 4.15a, Figure 4.15b, and Figure 4.15c for M=0.70, M=0.78, and M=0.82 respectively. These plots show the real and imaginary part of the first 3x3 entries of the AIC matrix. There are real and imaginary parts associated with these matrices as the AIC matrices computed with DLM are complex-valued due to the oscillatory nature of the aerodynamics. Hence, the real and imaginary parts are fit separately, of which the fit is shown here.

The fit is not perfect, however, as indicated by the S-shaped nature in some areas. Increasing the number of poles only made this worse. Additionally, more reduced frequencies to interpolate on could help, but was not attempted.

4.4.2. AEROSTRUCTURAL SPLINING MODEL

The aerostructural coupling was achieved using the DLR Loads Kernel preprocessor. As was highlighted in Section 4.2, the master nodes of the RBE3 elements of the wings and the nodes representing the rigid fuselage and tail are used for aerostructural coupling. The nearest neighbour mapping was used to connect the meshes. The resulting splining model is shown in Figure 4.16. The pylon RBE3 master nodes have been excluded from the nodes for splining as inclusion causes connections of collocation points on the wing to the pylon nodes, yielding out-of-plane connections.

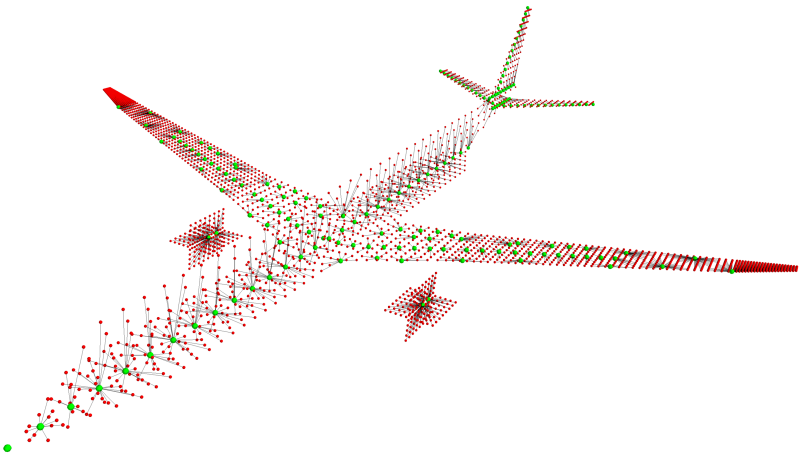


Figure 4.16: Aerostructural splining model of the free-flying aircraft

4.4.3. STRUCTURAL & MASS MODEL

This section convers the structural model, the mass model, and the eigenmodes and eigenfrequencies of the structure. Two versions of the aircraft are presented here: the baseline version and a tailored version. The tailored version stems from the last iteration step of the tailoring process defined in Subsection 4.2.3.

STRUCTURAL MODEL

The final structural model is shown in Figure 4.17. It contains roughly 75k nodes and 100k elements. The time to build the mass and stiffness matrices using SOL103 varies, but typically does not exceed 1-2 minutes.

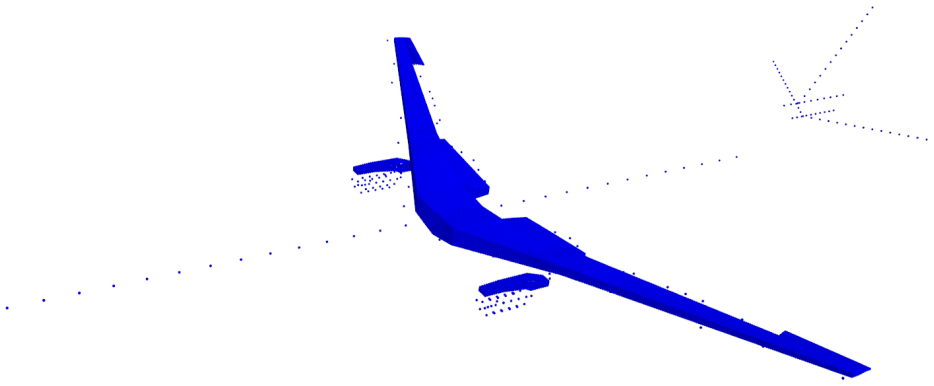


Figure 4.17: Structural finite-element model of the free-flying aircraft

MASS MODELS

As was discussed in Subsection 4.2.4, mass cases can be generated for fuel, passengers, and cargo. Below these have been depicted. Figure 4.18 shows the empty configuration, Figure 4.19 shows the full configuration, and Figure 4.20 and Figure 4.21 show the 40% loaded case for inboard-outboard and outboard-inboard loaded respectively. The passenger and cargo masses are lumped to a big mass that can be seen in the middle, which therefore changes in size and position depending on the mass case.

Figure 4.18: Mass model of the free-flying regional jet with no fuel inside the tanks

Figure 4.19: Mass model of the free-flying regional jet with the tanks at full capacity

Figure 4.20: Mass model of the free-flying regional jet with 40% fuel inboard out

Figure 4.21: Mass model of the free-flying regional jet with 40% fuel outboard in

MODE SHAPES

The figures below show the mode shapes of the baseline wing in the empty configuration, meaning no passengers, cargo, or fuel in the tanks. Figure 4.22 shows the first mode, Figure 4.23 the second mode, Figure 4.24 the third mode, Figure 4.25 the fourth mode, Figure 4.26 the fifth mode, Figure 4.27 the sixth mode, Figure 4.28 the seventh mode, and lastly, Figure 4.29 the eighth mode. The eigenfrequencies of the modes are shown in Table 4.12 for the baseline wing in both the empty and full configuration, to demonstrate the effort of mass on the eigenfrequencies.

Mode	Empty	Full
1	2.82 Hz	2.36 Hz
2	3.34 Hz	3.06 Hz
3	3.51 Hz	3.50 Hz
4	3.95 Hz	3.72 Hz
5	5.32 Hz	5.11 Hz
6	6.29 Hz	5.59 Hz
7	7.18 Hz	5.93 Hz
8	7.24 Hz	6.05 Hz

Table 4.12: Eigenfrequencies of mode shapes for various configurations

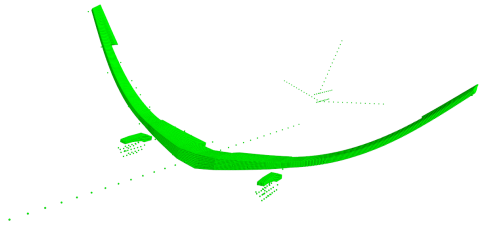


Figure 4.22: 1st eigenmode, symmetric wing bending

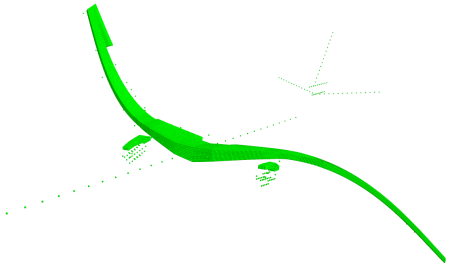


Figure 4.23: 2nd eigenmode, antisymmetric wing bending

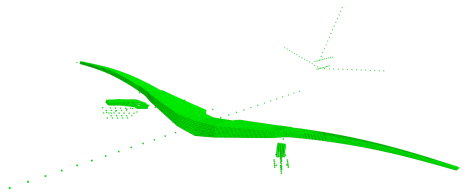


Figure 4.24: 3rd eigenmode, symmetric lateral bending

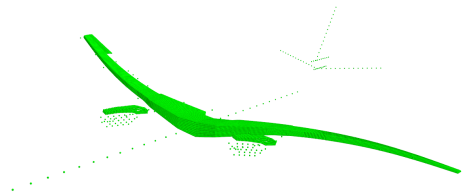


Figure 4.25: 4th eigenmode, antisymmetric lateral bending

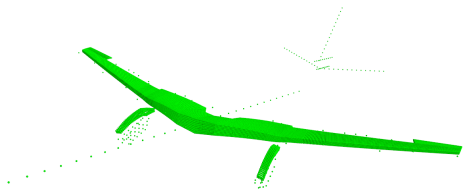


Figure 4.26: 5th eigenmode, symmetric wing torsion

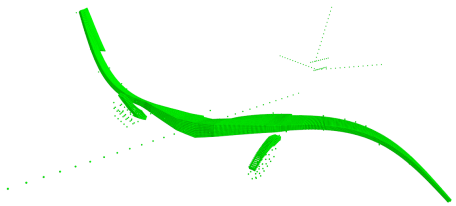


Figure 4.27: 6th eigenmode, antisymmetric wing torsion

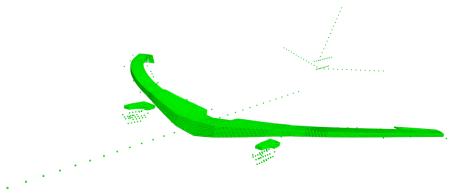


Figure 4.28: 7th eigenmode, symmetric lateral bending

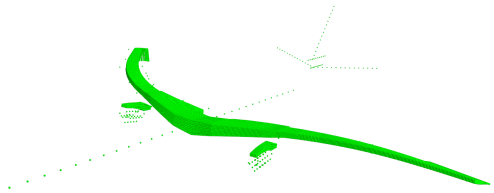


Figure 4.29: 8th eigenmode, antisymmetric lateral bending

5

RESULTS

The state-space model inputs are derived from DLR Loads Kernel and is therefore backward compatible with Loads Kernel, as is explained in Section 4.1. Thus, DLR Loads Kernel can be used to verify the model. Verification of the model is performed through three avenues. First, the performance of DLR Loads Kernel is compared to the state-space model in Section 5.1. Secondly, an eigenvalue analysis is executed in Section 5.2 on the state matrix of both Loads Kernel and the state-space model to compare the aeroelastic and flight dynamics coupling. Additionally, gust and maneuver time responses are compared as a final verification in Section 5.3. Lastly, Section 5.4 compares the lag state reduction approach of this thesis to the methods in Loads Kernel. The aircraft used in this verification is the regional jet generated in Chapter 4. The flight condition at which the verification was performed corresponds to $M = 0.70$ and $FL = 250$. However, before analyzing the models, it is important to list the differences in which the model in DLR Loads Kernel differs from the derived state-space model, and what the expected differences in results are.

1. **Improved model interpretability, extensibility, and traceability.** In this work, the model has been mathematically derived and formulated into an explicit, closed-form, symbolic, continuous-time state-space model, cleanly separated into the state and output equation, in conjunction with the corresponding input, state, output, and, disturbance vectors, which is preassembled offline. DLR Loads Kernel, on the other hand, expresses the model as a first-order differential equation, but not strictly in state-space form, and most of the computational processing steps are performed online. Preassembling the model should yield improvements in computational efficiency, which are shown in Section 5.1. Additionally, a rigorous mathematical derivation of the Loads Kernel model is absent. The lack of a mathematical derivation has caused unaccounted for model simplifications in Loads Kernel. For instance, the downwash due to control surface motion is not modeled and the equation for displacement-induced downwash is not derived from first principles. Moreover, the Loads Kernel model approximates the downwash derivative using backward finite-difference, making the model formulated in discrete-time. In the state-space model, these terms are explicitly computed to keep the equations in continuous-time. The effect of this could be a slight discretization error, although this is expected to be small. Using the state-space model derived in this thesis, the model becomes more interpretable. It is rigorously derived, with all terms accounted for, and formulated in continuous-time. Simultaneously, these improvements should not affect the core model behaviour, or minimally so, which is verified in Section 5.2 and Section 5.3.

2. **Reduction of the number of gust disturbance inputs.** DLR Loads Kernel adjusts the downwash of each panel individually as the aircraft penetrates the gust field, leading to a huge number of disturbance inputs. On the contrary, in this work, the aircraft is discretized into gust zones [64], and the time delay between each zone is modeled using a Padé approximation, leading to only two gust inputs. This is adopted similarly to Schulz and Ossmann [65] and Wuestenhagen [10]. This approach could introduce a discretization error, and approximating the time delay could introduce unphysical ringing or oscillatory behaviour in the response, which is a known issue with the Padé approximation, as the transfer function has complex conjugate poles. The gust modeling was covered in Section 3.11, and the gust response is compared in Section 5.3.

3. **Reduction of the number of aerodynamic lag states.** The aerodynamic lag states of the physical RFA have been replaced by a mathematically equivalent formulation where the lag states are projected to rigid-body force lag, rigid-body moment lag, generalized force lag, and monitoring stations load lag. The advantage in this approach is that the number of lag states is significantly reduced from $n \times p$ to $(6+m+s) \times p$, with n aerodynamic panels, p RFA poles, m structural modes, and s the number of forces and moments to be output from the monitoring stations. Therefore, if only the WRBM and WRTM are monitored, the number of lag states becomes $(8+m) \times p$. The advantage is a massive reduction in states while still enabling the use of the superior, physical RFA. The downside is that the aerodynamics at the panel level cannot be resolved from the state-space model, and that each monitoring station subsequently also needs a lag state, but this was deemed not an issue from an aeroservoelastic point of view. This approach is verified in Section 5.4.

4. **Model augmented with actuator dynamics and accelerometer model.** Lastly, in the model of this thesis, actuator dynamics have been added to the model, although for verification of the model, the actuator bandwidth and damping have been set to sufficiently high values to not have an effect on the response. The actuator dynamics have been modeled in Section 3.9. Sensor dynamics have also been added but this is merely an output equation, which does not affect the model dynamics. These are merely missing functionalities and are not interesting to analyze, and are therefore not further discussed nor verified.

5.1. COMPUTATIONAL EFFICIENCY

As the most distinctive difference between Loads Kernel and the model in this work is the formulation of a state-space model which is preassembled, this should come with a subsequent increase in computational efficiency. As the main goal of the free-flying aeroservoelastic state-space model is to perform time domain simulations, the performance of Loads Kernel and the state-space model is compared in this regard. The simulation cases for this analysis are the same as the maneuvers and gust considered in Section 5.3: an elevator step input, aileron step input, rudder step input, and a 1-cos discrete gust. Table 5.1 shows the difference in simulation times between Loads Kernel and the state-space model. The state-space model consistently achieves a staggering decrease in simulation time, with the decrease being at least two orders of magnitudes faster. It must be admitted though that the simulation times vary, and depend on computer configuration, and available computational resources. In the experience of this work, the simulation times for Loads Kernel have never dropped below 700 seconds, indicating that two orders of magnitude faster is good ballpark number for a consistent quantification of the increase in computational efficiency. Additionally, the state-space model is run with an explicit Runge-Kutta integration scheme of order 45, and Loads Kernel with an explicit Adams-Bashforth method. As the Adams-Bashforth uses a fixed timestep, and the RK45 an adaptive step size, the RK45 is computationally more expensive. So in fact, the state-space model would likely be even faster when also using the Adams-Bashforth integration scheme. For the simulations in this section, it was not necessary to use an implicit solver with the state-space model, although this could easily increase the computational time by an order of magnitude. The results also show that the implemented MATLAB model achieves similar improvements in computational efficiency.

Simulation	DLR Loads Kernel	State-Space (Python)	State-Space (MATLAB)
Elevator Step Input	806.45 s	2.53 s (319x faster)	3.06 s (264x faster)
Aileron Step Input	797.58 s	2.55 s (313x faster)	4.14 s (193x faster)
Rudder Step Input	1156.16 s	2.68 s (431x faster)	3.76 s (307x faster)
1-Cos Gust Disturbance	865.48 s	6.27 s (138x faster)	8.58 s (101x faster)

Table 5.1: Comparison simulation times DLR Loads Kernel versus state-space model of this thesis

Table 5.3 shows the comparison in the file sizes of the input files. Also in this regard is the computational efficiency shown to be improved due to the preassembly of the state-space model, achieving a massive decrease in input file size. However, in this work, the state-space model was assembled using the input file of Loads Kernel, effectively negating this particular benefit as the Loads Kernel input file is still required. The inputs to the state-space framework can also be generated without Loads Kernel, and therefore, the decrease in file size is still an achievement which reinforces that the state-space model is more computationally efficient.

Simulation	DLR Loads Kernel	State-Space Model
Elevator Step Input	4.37 GB	9.54 MB (458x smaller)
Aileron Step Input	4.37 GB	9.54 MB (458x smaller)
Rudder Step Input	4.37 GB	9.54 MB (458x smaller)
1-Cos Gust Disturbance	4.37 GB	9.54 MB (458x smaller)

Table 5.2: Comparison input file sizes DLR Loads Kernel and state-space model of this thesis

The biggest upside of the state-space model in terms of file sizes is related enabled due to the increase in simulation time. In Loads Kernel, the evolution of the state vector is saved to a separate file, which is understandable given the large simulation time. These files, however, for the verification of the time responses in Section 5.3, become absolutely massive for just a 5 second simulation, as shown in Table 5.2. As the state-space model runs very quickly, it is not necessary to save the results into a separate output file, and one can head directly to analysis instead. However, the Loads Kernel output file contains a breadth of information that cannot be retrieved from the state-space model, such as the nodal loads. With an aeroservoelastic application in mind, the most important information is captured in the state and output vector and the additional information in this output file is not required.

Simulation	DLR Loads Kernel	State-Space Model
Elevator Step Input	23.31 GB	-
Aileron Step Input	23.31 GB	-
Rudder Step Input	23.31 GB	-
1-Cos Gust Disturbance	23.31 GB	-

Table 5.3: Comparison output file sizes DLR Loads Kernel and state-space model of this thesis

It must be added that the simulation times in Loads Kernel heavily depend on the number of aerodynamic panels the model uses, which in this verification corresponds to about 2942 panels, which is quite significant. In this remark also lies the crux of the derived model, which decouples the simulation time from the number of aerodynamic panels. Additionally, it must also be noted that the state-space model naturally requires preassembling the state-space model, which can be computationally demanding due to the explicit precomputation of the state-space matrices. Yet, in the experience of this work, this has not been an issue.

5.2. FLIGHT DYNAMIC AND AEROELASTIC MODES

In order to assess whether the aeroelastic and flight dynamic coupling is implemented correctly, the eigenvalues of the linearized state matrix from the state-space model, as well as the linearized state matrix from Loads Kernel are compared. This requires some manual adjustments in Loads Kernel to make the results from the state-space model correspond to that of Loads Kernel. Although the original implementation of Loads Kernel is therefore modified, the emphasis of this section is to ensure that the state-space model is correctly implemented, not whether the modeling choices, in which Loads Kernel and the model of this thesis differ, are representative of reality, as that is regarded as validation. Therefore, the Loads Kernel implementation must match closely with the state-space model, in areas where the coupled aeroelastic and flight dynamics dynamics may be affected. First, Loads Kernel uses the steady aerodynamic equations for computing the trim condition. In these steady aerodynamic equations of Loads Kernel, the freestream velocity is a function of the body states, whereas the state-space model fixes the dynamic pressure at the reference flight condition, just as is done in Loads Kernel when using unsteady aerodynamics. Therefore, in the Loads Kernel implementation for steady aerodynamics the freestream velocity is fixed to that of the reference flight condition. Secondly, Loads Kernel uses a different downwash formulation, which was discussed in Subsection 2.3.3. This is also changed to the formulation that is used in the model, described in Section 3.6. Thirdly, Loads Kernel flips some of the states in the state vector, which are therefore negated, as this also yielded issues in the eigenvalue analysis. Regarding the state-space model, the gust and lag state derivatives are set to 0 in order to ensure that the state matrices share a common set of states, and to prevent residual coupling from polluting the eigenvalue analysis. Once these changes are brought about, the eigenvalues of both matrices are computed, resulting in 32 eigenvalues, 12 due to the rigid body states, and 20 relating to the structural dynamics, as 10

modes are used for this verification. Subsequently, the eigenfrequencies are computed with Equation 5.1 and the damping ratios with Equation 5.2.

$$f_n = \frac{\mathrm{Im}(\lambda_n)}{2\pi} = \frac{\omega_n}{2\pi} \tag{5.1}$$

$$\zeta_n = -\frac{\mathrm{Re}(\lambda_n)}{|\lambda_n|} = -\frac{\sigma_n}{\sqrt{\sigma_n^2 + \omega_n^2}} \tag{5.2}$$

Before diving further into the analysis, it is important to identify what the modes represent. This is primarily of importance for the flight dynamic modes and the first aeroelastic modes, as these have a relatively low eigenfrequency and therefore dominate the response. The modes were identified by looking at the dominating states in each of the eigenvector in conjunction with the corresponding eigenvalue of that eigenvector, as this characterizes the flight dynamic modes. For the subsequent analysis, the first 5 modes were removed as these represent free modes. Additionally, only one of the conjugate pairs is kept. This results in 5 lower frequency flight dynamic modes, and 10 aeroelastic modes, which are shown in Table 5.4. This table is sorted from low to high absolute eigenvalue.

Mode Number	Eigenvalue	Mode
1	0.001034	Phugoid-like
2	−0.014439	Spiral
3	$-0.321364 \pm 1.558017\,j$	Dutch Roll
4	$-0.831149 \pm 1.972255\,j$	Short Period
5	−3.389520	Aperiodic Roll
6	$-0.661551 \pm 21.364667\,j$	3rd Structural Mode
7	$-0.691818 \pm 24.143616\,j$	4th Structural Mode
8	$-8.739379 \pm 23.295261\,j$	1st Structural Mode
9	$-7.865481 \pm 26.501992\,j$	2nd Structural Mode
10	$-0.695842 + 32.937882\,j$	5th Structural Mode
11	$-2.888201 \pm 39.067441\,j$	6th Structural Mode
12	$-0.907105 \pm 45.095876\,j$	7th Structural Mode
13	$-0.955719 \pm 45.470755\,j$	8th Structural Mode
14	$-6.605966 + 57.839117\,j$	9th Structural Mode
15	$-1.349000 + 62.251069\,j$	10th Structural Mode

Table 5.4: Eigenvalues and mode identification of flight dynamic and aeroelastic modes

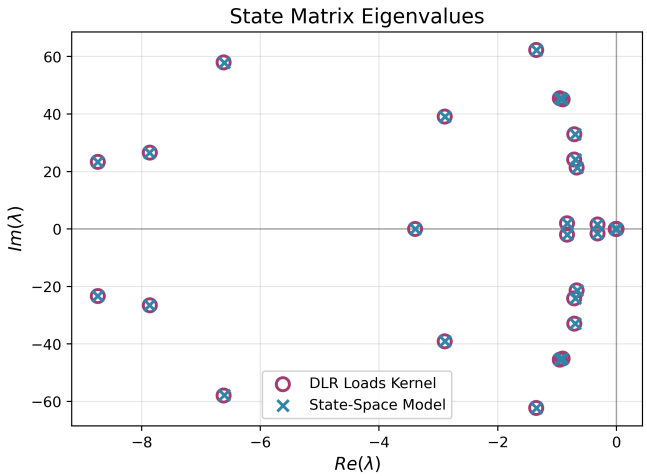


Figure 5.1: Eigenvalues of LK and SS in the complex plane

The flight dynamic modes are identified based on the participating states in the corresponding eigenvector, and on the eigenvalue. In this particular model, there is no phugoid, but rather what would be the phugoid if

the freestream velocity was not fixed at the reference flight condition and the downwash formula considered camber and twist. As a result, the phugoid-like mode 1 is non-oscillatory, and drifts very slowly. This implies that any perturbation in terms of pitch will continue to keep growing, although very slowly. As the timescale at which this takes place is much larger than typically analysed when doing AFS, MLA, and GLA, the usability of the model for these purposes is not affected. When the Loads Kernel implementation is switched back to using the non-fixed freestream velocity and the original downwash implementation, this mode becomes oscillatory, yet still unstable, which is in line with expectations as the dynamic pressure becomes varying. The phugoid likely becomes stable when induced drag and viscous drag are implemented into the model, as this creates a mechanism to dissipate energy. The second mode shows strong participation of the lateral velocity and the yaw angle, and is lightly damped, indicating the spiral mode. Mode 3 shows participation of lateral velocity, roll rate, roll angle, and yaw rate, shows oscillatory motion, is mildly damped, and therefore represents the Dutch roll. Mode 4 shows participation of the vertical body velocity, pitch rate, and forward velocity, is oscillatory, and relatively heavily damped, and therefore represents the short period mode. Lastly, mode 5 shows participation of roll rate and roll angle, and is quite strongly damped, indicating it is the aperiodic roll. The aeroelastic modes can be identified based on the strongest participating structural mode for each, as is shown. The aeroelastic modes do not follow the same ordering as that expected from purely looking at the structural dynamics. As can be seen from both the eigenvalues in Figure 5.1 and the natural frequencies in Figure 5.2, mode 6 has the lowest frequency, mode 8 second lowest, mode 7 third lowest, and mode 9 fourth lowest. This means the modes dominated by the 3rd and 4th structural modes have lower frequencies compared to the modes dominated by the 1st and 2nd structural modes, respectively. This can be explained by looking at the respective structural modes. The 3rd and 4th structural modes are dominated by lateral bending, in which there is less aerodynamic mass, stiffness, or damping. Therefore, the frequencies do not increase as much as for the other modes, which are subject to stronger aerodynamic stiffness, mass and damping, and explains this peculiar ordering. Lastly, Figure 5.1 shows the eigenvalues of both Loads Kernel and the state-space model in the complex plane, showing that the eigenvalues correspond exceptionally well with the adjusted Loads Kernel model. Figure 5.2 shows the natural frequencies of the flight dynamic and aeroelastic modes. These plots show exceptional agreement, with the difference in natural frequencies all below 0.5% if not less, as shown by Figure 5.3. No further analysis was done as to what causes this slight difference in natural frequencies, although it is likely due to a slightly different trim condition, or due to any of the additional model differences noted at the start of this chapter.

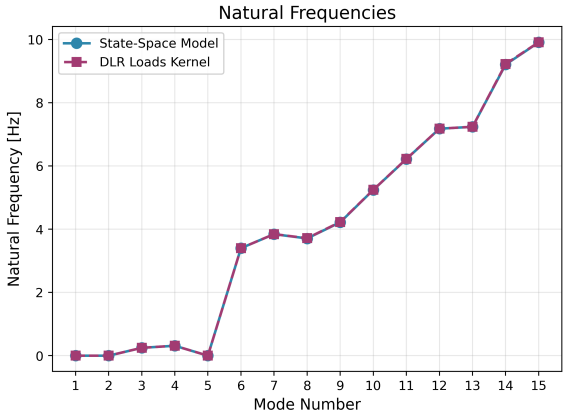


Figure 5.2: Natural frequencies from LK and SS

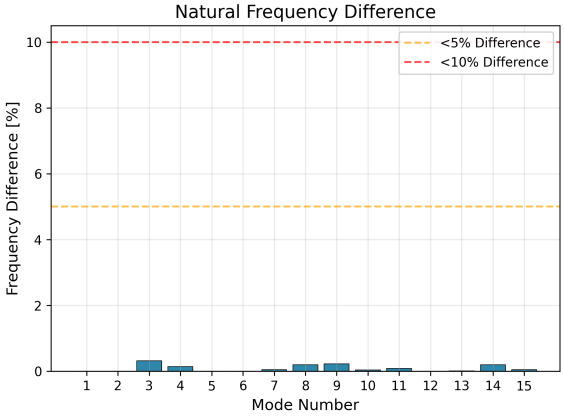


Figure 5.3: Natural frequencies difference between LK and SS

The corresponding damping ratios are shown in Figure 5.4, which also agree well between the state-space model and the adjusted Loads Kernel model. Figure 5.5 shows the percentual differences in damping ratios between the adjusted Loads Kernel model and the state-space model of this thesis. The difference in damping ratios is at most 3%, with most differences below 1%. It is likely that these differences stem from a slightly different trim condition. Additionally, although the state-space model was effectively converted to represent steady aerodynamics, differences could still arise in the state matrix if the trim condition is not perfectly in trim, for instance. In any case, these differences are of such small magnitude, that it was deemed irrelevant to further investigate the cause, especially considering that the model in this thesis is more scrupulously derived.

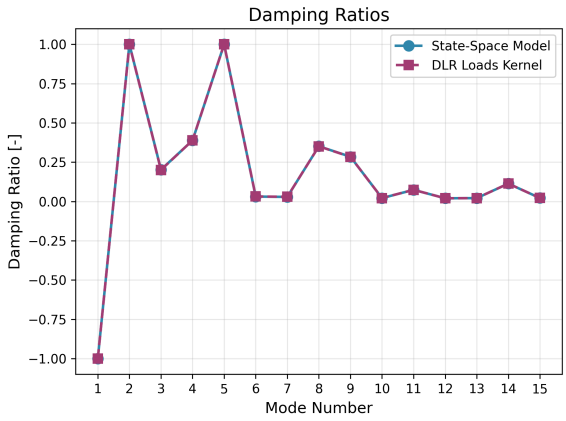


Figure 5.4: Damping ratios from LK and SS

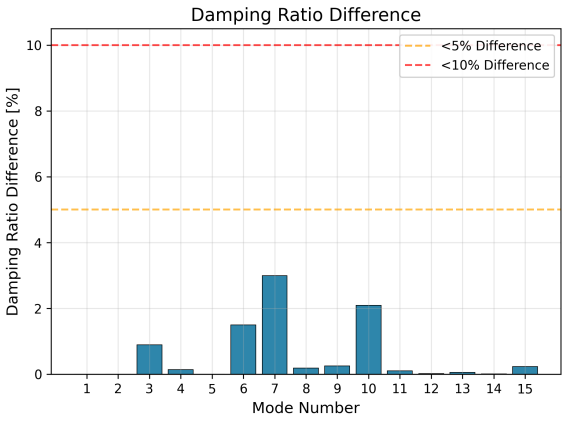


Figure 5.5: Damping ratios difference between LK and SS

The eigenvalues have now been verified, however, the mode shapes should also be checked for their correspondence. In order to compare these eigenmodes, the Modal Assurance Criterion (MAC) is used, which indicates how well two modes correlate, and is computed using Equation 5.3. If MAC is 1 the modes correlate perfectly, if 0 they are entirely different. Figure 5.8 shows the MAC matrix. Before comparing the MAC across Loads Kernel and the state-space model it is important to first look at how the modes of each model are correlated in itself, as this may otherwise lead to the conclusion that the modes do not correspond, while this is in fact the way the modes should appear. Figure 5.6 shows the MAC matrix for the eigenmodes from Loads Kernel, and Figure 5.7 shows the eigenmodes from the state-space model. Indeed, the modes do correlate slightly. Figure 5.8 lastly shows the MAC matrix of the modes from the state-space model and those from the adjusted Loads Kernel model, which show the exact same pattern, and therefore the modes are equivalent.

$$MAC = \frac{\left|\boldsymbol{\phi}_i^T \boldsymbol{\phi}_j\right|^2}{(\boldsymbol{\phi}_i^T \boldsymbol{\phi}_i)(\boldsymbol{\phi}_j^T \boldsymbol{\phi}_j)} \tag{5.3}$$

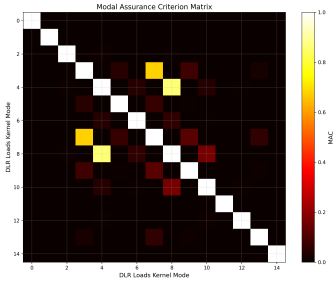


Figure 5.6: Eigenmodes MAC matrix of LK against LK

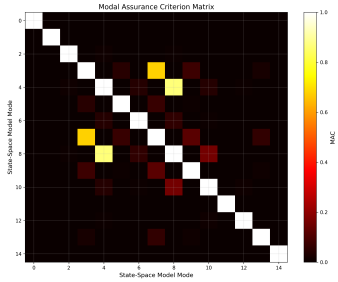


Figure 5.7: Eigenmodes MAC matrix of SS against SS

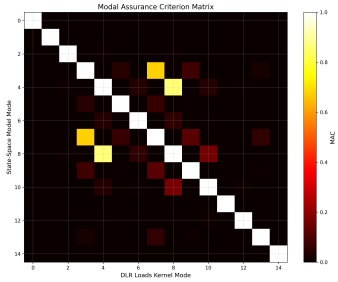


Figure 5.8: Eigenmodes MAC matrix of SS against LK

5.3. MANEUVER AND GUST RESPONSES

As the model has nonlinearities and the state matrix only considers perturbations of the model around the trim condition, the actual responses are also compared. The considered cases include three maneuvers and one gust. First, the results for an elevator step, aileron step, and rudder step are shown, and at last, the results for a 1-cos gust are shown. In Section 5.2, the Loads Kernel model and state-space model were slightly modified in order to be able to verify the implementation of equal models. The time responses of this section, however, use the state-space model and the unsteady model in Loads Kernel as is. The response of the state-space model is shown for both the Python and MATLAB model, in order to verify that the MATLAB implementation is also correctly implemented.

ELEVATOR STEP INPUT

The figures below display the time response due to an elevator step input, for both DLR Loads Kernel and the state-space model derived in this work. Figure 5.9 shows the pitch angle, Figure 5.10 shows the pitch

rate, Figure 5.11 shows the wing root bending moment, and Figure 5.12 shows the load factor in the body z-direction, Figure 5.13 shows the first wing bending mode response, Figure 5.14 shows the angle of attack response, and, lastly, Figure 5.15 shows the altitude deviation. Overall, the responses match exceptionally well.

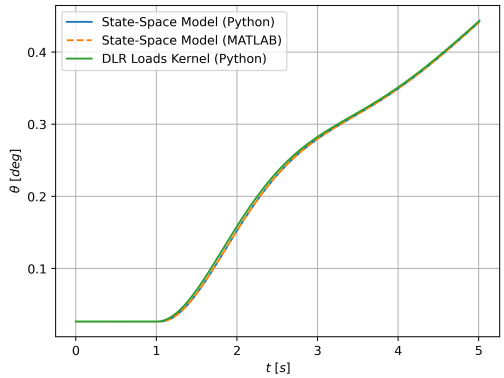


Figure 5.9: Pitch angle response to an elevator step input

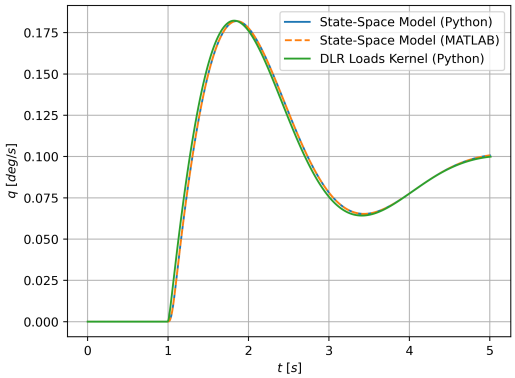


Figure 5.10: Pitch rate response to an elevator step input

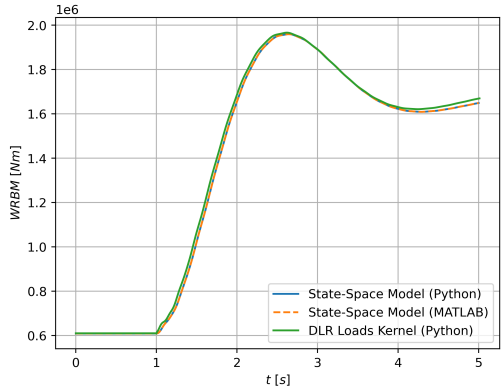


Figure 5.11: WRBM response to an elevator step input

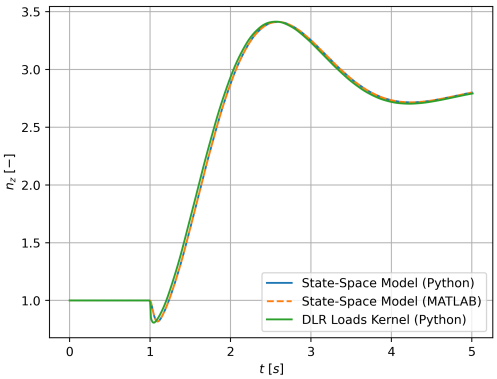


Figure 5.12: Load factor response to an elevator step input

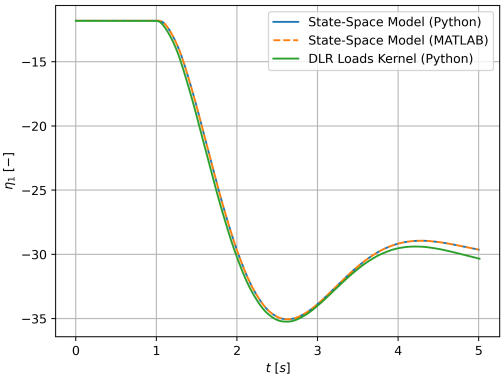


Figure 5.13: First bending mode response to an elevator step input

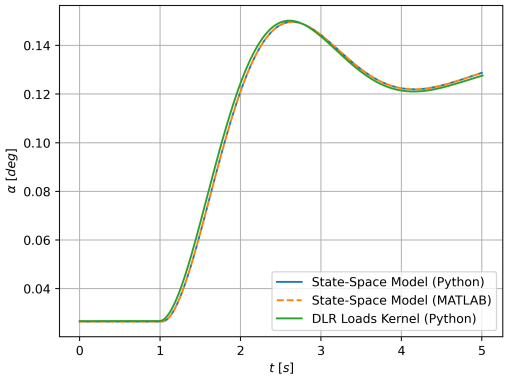


Figure 5.14: Angle of attack response to an elevator step input

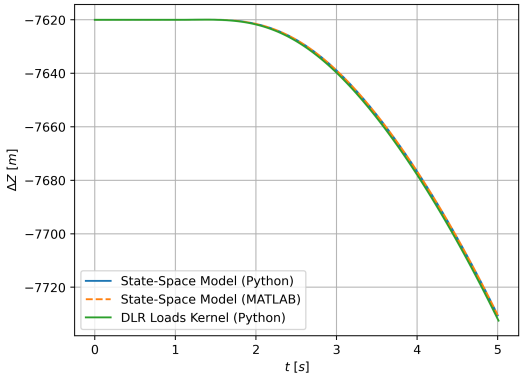


Figure 5.15: Altitude deviation response to an elevator step input

AILERONS STEP INPUT

Additionally, the model is fed an aileron step input. For this aileron step input, the outboard ailerons of the aircraft are deflected asymmetrically. Figure 5.16 shows the roll angle, Figure 5.17 shows the roll rate, Figure 5.18 shows the sideslip angle, and Figure 5.19. Except for the roll rate response, are the responses nearly identical. The difference in roll rate is most likely due to the downwash in the state-space model being linear.

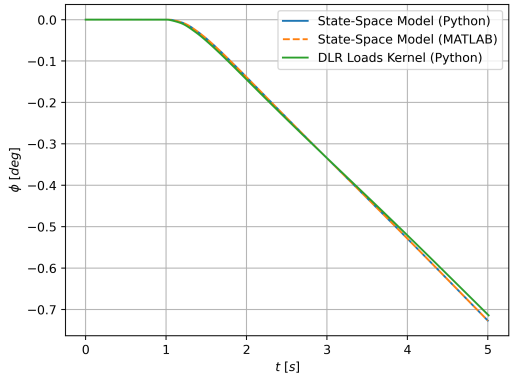


Figure 5.16: Roll angle response to an aileron step input

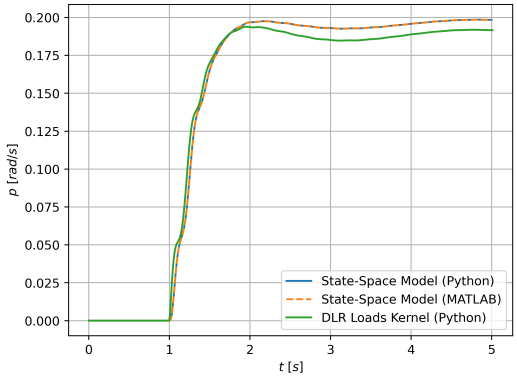


Figure 5.17: Roll rate response to an aileron step input

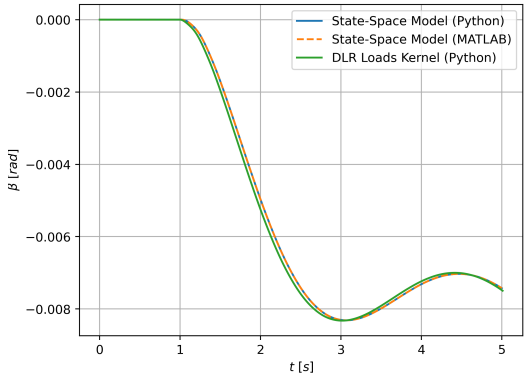


Figure 5.18: Sideslip angle response to an aileron step input

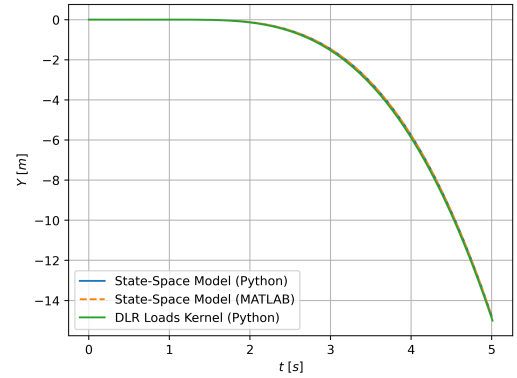


Figure 5.19: Lateral deviation response to an aileron step input

RUDDER STEP INPUT

Figure 5.21 shows the time response due to a rudder step input, with Figure 5.20 showing the yaw angle, and Figure 5.21 showing the yaw rate. The match in these plots, although still acceptable, is worse. The difference must also be due to the downwash formula being linear in the state-space model, as there would be no other plausible explanation.

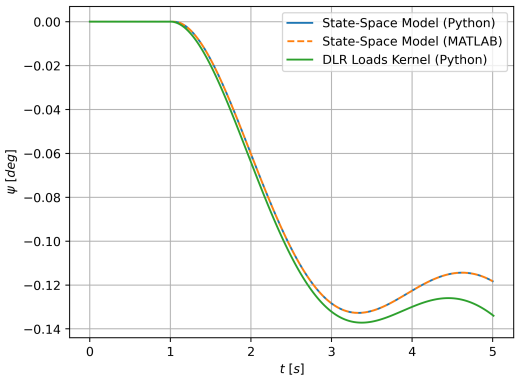


Figure 5.20: Yaw angle response to a rudder step input

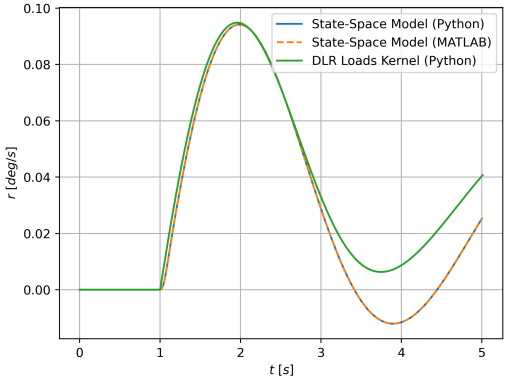


Figure 5.21: Yaw rate response to a rudder step input

1-COS GUST

The plots below show the response due to a 1-cos gust with gust design velocity $U_{ds} = 35\,\text{m/s}$ and gust gradient $H = 20\,\text{m}$. Figure 5.22 shows the angle of attack variation, Figure 5.23 shows the pitch angle, Figure 5.24 shows the first wing bending mode, Figure 5.25 shows the altitude deviation due to the gust, Figure 5.26 shows the load factor in the body z-direction, and, lastly, Figure 5.27 shows the wing root bending moment response. Although the Padé approximation is prone to unphysical oscillatory motion in the gust response, this does not seem to affect the gust response in this simulation. This is likely due to these modes being damped out relatively quickly due to the aerodynamic damping, or because the 1-cos gust can be resolved quite well using the Padé approximations, with the oscillatory motion only present in extremely steep gust gradients. Further investigation should be performed to pinpoint when these effects start to creep in. Considering that the match for this already short, transient gust is very good, such an analysis was not performed in this work. Overall, the gust response plots match exceptionally well.

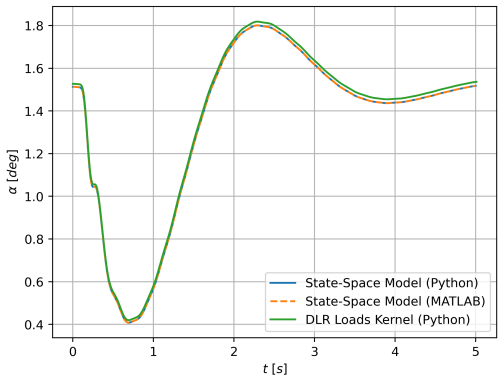


Figure 5.22: Angle of attack response to a discrete 1-cos gust

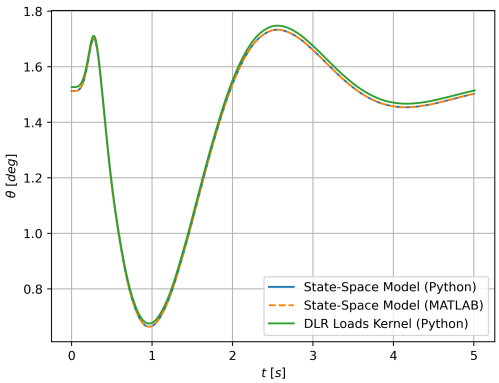


Figure 5.23: Pitch angle response to a discrete 1-cos gust

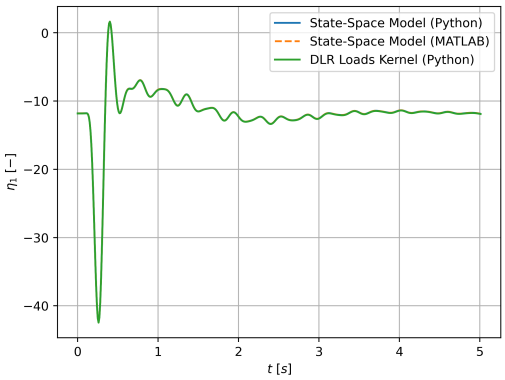


Figure 5.24: First bending mode response to a discrete 1-cos gust

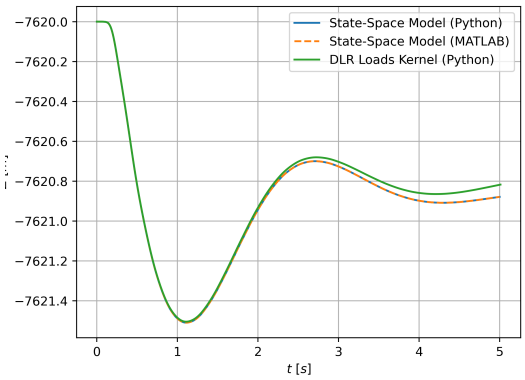


Figure 5.25: Altitude deviation response to a discrete 1-cos gust

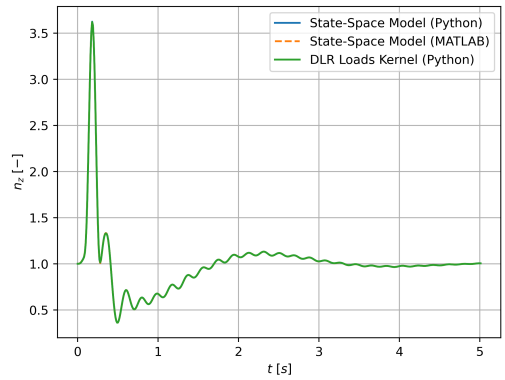


Figure 5.26: Load factor response to a discrete 1-cos gust

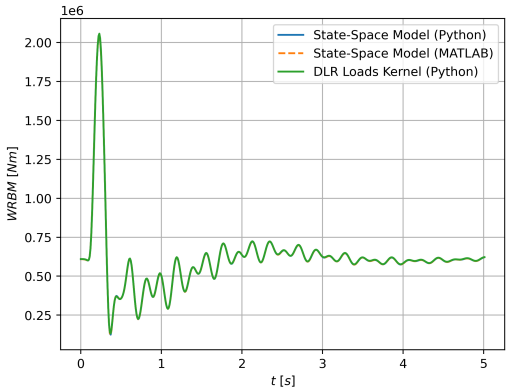


Figure 5.27: WRBM response to a discrete 1-cos gust

5.4. LAG STATE COMPARISON

An interesting further verification is to compare how the full physical RFA and the half-generalized RFA in Loads Kernel compare to the projected physical RFA presented in Equation 3.64. Table 5.5 summarizes the important differences between these approaches.

Method	States	Nature
Full Physical RFA	$n \times p$ (11768)	Exact
Half-Generalized RFA	$2 \times m \times p$ (80)	Approximate
Projected Physical RFA	$(6+m+s) \times p$ (544 [1])	Exact

Table 5.5: Comparison number of lag states and RFA method

At the same flight condition as was considered earlier in this chapter, FL250 and Mach 0.70, two gusts were fed into the aeroservoelastic system with gust design velocity 35 m/s. Figure 5.28 shows the WRBM for a gust gradient of 20 m and Figure 5.29 for a gust gradient of 10 m. WRBM was chosen to demonstrate this as all dynamics are aggregated in this metric. Although at this particular flight condition the lag states do not dominate the response, the effect of lag on the aerodynamic response is not what is to be shown. The goal of these plots is to prove the equivalence and superiority of the projected physical RFA method of this thesis. These plots indeed show that 1) the full physical RFA method and the projected physical RFA method presented in this thesis are shown to produce equivalent results with a massive reduction in the number of lag

[1] Note that this value is particularly high as there are 6 times 20 load lag states included. If only the WRBM and WRTM are used, this reduces down to 72 states, which is more reasonable.

states and 2) the half-generalized approach in Loads Kernel is indeed an approximation which also requires more lag states than the exact, projected RFA and this half-generalized approach is therefore inferior. In flight conditions where the lag states dominate, such as in flutter, this projected physical RFA could be an important, but incremental contribution as it allows an exact reduced-order representation of the aeroservoelastic state-space model.

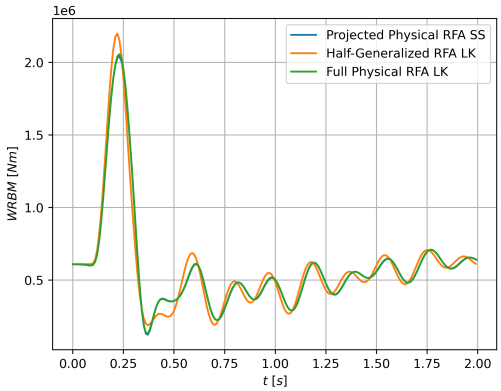


Figure 5.28: Comparison RFA method for 1-cos gust with $H = 20m$

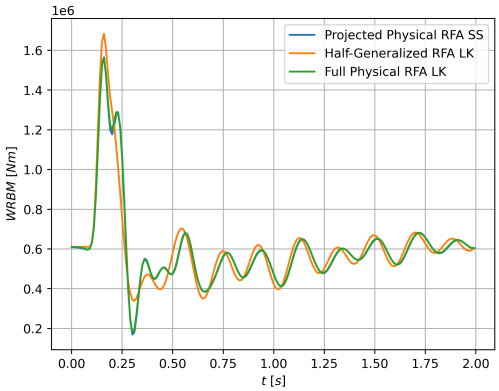


Figure 5.29: Comparison RFA method for 1-cos gust with $H = 10m$

6

DISCUSSION

The discussion is divided up into four sections. As this work is mostly modeling work, the discussion revolves around alternative modeling approaches and justification of the ones chosen, modifications and improvements, as well as applications of the model and future work. The results of the model verification are not discussed further here, as they do not require further discussion due to their agreement. Section 6.1 discusses the most important model assumptions, the limitation imposed by them as well as possible alternative modeling approaches to take out those assumptions. This section primarily covers issues that would require an entirely new model to be developed and derived. Secondly, due to the time constraints in this work, certain extensions or modifications to the model have not been added or were chosen to not be added to the model, which are detailed in Section 6.2. These extensions and modifications do not require different underlying models, and can thus be integrated into the framework as is. Section 6.3 discusses improvements in the architecture of the end-to-end modeling pipeline of Chapter 4, as well as specific improvements to the specific aircraft model that is generated using the pipeline. Lastly, Section 6.4 discusses applications of the model.

6.1. CORE ASSUMPTIONS & LIMITATIONS

The model hinges on several important assumptions, which are listed here. Although these do not provide a comprehensive list of assumptions made at each step in the derivation, the most important assumptions are highlighted here. It is important to question the validity of these in case of making changes to the model, or when importing new aircraft, or generating models at different flight conditions.

- The panel aerodynamics using VLM/DLM are based on linear potential flow, meaning the model assumes incompressible, inviscid, irrotational flow. Consequently, viscous effects and flow separation are not captured. Compressibility corrections can be applied to VLM/DLM. As the VLM/DLM implementation by Voß [20] is used, the Prandtl-Glauert correction is applied. However, these corrections are empirical and thus merely provide an approximation. Alternatively, the accuracy of these AIC matrices can also be enhanced with CFD[80], which is not performed in this work. The author also notes that in general VLM/DLM overestimates the loads compared to CFD , and is thus a conservative aerodynamic model.

- The aerodynamic forces are normal to the aerodynamic panels. This does mean however, that when using an aerodynamic mesh consisting of (near-)horizontal and (near-)vertical panels, such as the aircraft modeled in Subsection 4.4.1, insufficiently captures the true pressure distribution around the aircraft. When using such a mesh, more complex aerodynamic and flight dynamic behaviour, such as yaw-roll coupling, is insufficiently captured. This also means that wing-fuselage interference is also not modeled properly. A solution could be to integrate the model with a 3D panel method for VLM/DLM, such as the unsteady source and doublet panel method developed by Dimitriadis, Kilimtzidis, Kostopoulos, *et al.* [70]. This software is publicly available, allows modeling the fuselage as sources, and can resolve complex 3D geometries.

- Using linear finite-element models assumes that the structural deformations are in the linear regime, which is valid for small to moderate deformations. Geometric nonlinear effects, such as tip shortening

and geometric stiffness are not captured. It was noted in the literature review however, that this primarily affects the structural loads, and not so much the aggregated aerodynamics that are relevant for free-flying aeroelasticity. Yet, for highly flexible wings and subsequent accurate loads calculation, this should be the next step in the development of these models.

- As modal analysis and truncation is applied, the model implicitly assumes that the structural dynamics are captured sufficiently accurate using a truncated modal basis. This assumptions holds if the forces exciting the structure are in the same frequency band, which should be verified depending on the desired application.

- The aerodynamic coupling can be assumed linear, yielding a mostly linear state-space model. Thus, nonlinearities between displacement states and motion states are not captured. This means that, for example, that the effect of increased wing dihedral due to wing flexibility in sideslip on the aircraft response is not captured. A solution to this would be to use nonlinear aerodynamic boundary conditions. A method to implement nonlinear aerodynamic boundary conditions is presented in Section 6.2, but for reasons also discussed in that section, is not implemented in this model.

- The model is only aerodynamically coupled, as inertial coupling is assumed negligible. As was discussed in the literature review, inertial coupling does not have a significant impact on the aircraft behaviour unless the deformations are significant such as in HALE aircraft, when large accelerations are experienced, or if large concentrated masses are attached on the flexible structure. Surely, inertial coupling would have an impact on the response, if implemented, as the aircraft model used in this work has heavy engines attached to the pylon, and because configurations can be generated with fuel in the outer tanks. However, this can be considered a second-order effect, and as such was not derived. As Reschke [26] has derived inertially-coupled EOMs that directly integrate with industrial FEMs, this can be added to the model, although this will require a deep understanding of this paper, which was a lower priority for this work.

- The aircraft loads model typically assumes a fixed aerodynamic reference frame [39] due to the linearized nature of this model, as is also the case in the model of this thesis and in Loads Kernel. This frame is aligned with the body axes. However, when the angle of attack becomes large this assumption becomes invalid. However, as the linearity of the aerodynamic model is tied to the same assumption, the aerodynamic frame being fixed is a valid assumption as long as the linearity of the aerodynamic model holds. This also means that gusts are applied effectively in the body frame. In flutter, this is a valid assumption as the angle of attack remains limited. In maneuvers and gusts, it is recommended to check the angle of attack, to see if it becomes significant during the considered simulation.

- The aerodynamics of the model are evaluated at the reference flight condition. This implies that if the flight condition becomes significantly different, the aerodynamics are not modeled correctly any longer. This stems from the constant velocity and density at which the equations are evaluated, as well as the AIC matrices from DLM being valid for one Mach number. A solution would be to interpolate the matrices at the desired Mach numbers. This is not a limitation per se, as models can be built for different flight conditions, but it is a slight inconvenience as opposed to having one model valid for all flight conditions.

- On a similar note, the AIC matrices are fit using the RFA and therefore the fit must be verified. Additionally, it must be ensured that the band of reduced frequencies at which are interpolated are representative for the encountered frequencies. Moreover, one should mind that the resulting unsteady aerodynamics are an approximation because of the RFA and because numerical approximation is used to build the unsteady AICs. As VLM is used for steady aerodynamics, the steady AIC matrix is exact.

- The model in its current representation has limited ability to accurately capture the flight dynamic modes as was observed in Chapter 5, as viscous drag and induced drag have not been implemented. This can be added by making some adjustments to the model, as discussed in Section 6.2. The effects of neglecting viscous drag primarily presents itself in the phugoid mode being off, as there is no damping mechanism in the motion. The absence of induced drag primarily affects the Dutch roll mode being off, as there is less strong yaw-roll coupling.

- The measurement process is only partially modeled by only including the physics. However, in order to obtain a realistic model of the measurement process, measurement delay, noise and estimation must also be modeled. Examples on how to best do this for flexible aircraft are described in [10], [107].

6.2. MODEL EXTENSIONS & MODIFICATIONS

This section covers model extensions and modifications, or alternatives modeling approaches that could also be attempted, that might be better suited for specific situations.

- The actuator model implemented in Section 3.9 represents linear actuator dynamics. However, the nonlinearities in actuators, such as deflection and rate limits, are what typically drives the requirement for including actuators in the first place, as they can drive GLA performance [11], and this actuator model may be replaced by a suitable nonlinear model. This would not require the states-space system to be further augmented, but only require different, nonlinear equations that relate the actuator states to each other, which would be a fairly straightforward modification, but deemed out of scope within this work, as actuator specification is out of scope.
- As the 1-cos gust is often the critical gust loads in aircraft loads certification only this discrete gust was implemented. However, it could be interesting to add additional discrete gusts, as well as implement continuous gusts using Von Dryden or Von Karman turbulence. The latter would require augmenting the state-space model. Care must be taken however as ringing may be worse for sharply varying gusts like a step gust, or continuous turbulence with high-frequency content, and as such the number of gust zones may have to be increased.
- Alternatively for the modeling of deterministic gusts like the 1-cos gust, the Padé approximation can be replaced by directly using the gust zones as disturbance inputs. Logically, this does come at an increase in the number of gust inputs. However, this was not implemented as this would complicate adding continuous turbulence in the future, which is more difficult as these gusts are inherently not deterministic and the delay of this signal would have to be modeled eventually.
- Induced drag is neglected in this model, as it was deemed irrelevant in load alleviation or flutter suppression. However, for flight dynamics it is important to include as it is critical in modeling yaw-roll coupling. The model can be augmented to include induced drag by using a Trefftz plane analysis [39]. In the derivation of the equations, the following drag term should be added, and the resulting updated matrix expressions can be substituted into the model. Using the nomenclature from Loads Kernel [19], [20], Equation 6.1 are the induced velocities due to the bound vortex and trailing vortices, Equation 6.2 only accounts for the trailing vortices of each panel, which is what contributes to induced drag. Equation 6.3 expresses the pressure coefficient differential in terms of the AIC matrix, and Equation 6.4 expresses the induced velocities at the quarter-chord point, which directly causes the induced drag, as shown in Equation 6.5. The panel downwash can be substituted with Equation 3.93, and this term should then be added as an aerodynamic force in the x-direction at each panel in the derivation in Section 3.6.

$$\boldsymbol{A}_{jj} = \frac{A}{2b}(\boldsymbol{D}_1 + \boldsymbol{D}_2 + \boldsymbol{D}_3) \tag{6.1}$$

$$\boldsymbol{B}_{jj} = \frac{A}{2b}(\boldsymbol{D}_2 + \boldsymbol{D}_3) \tag{6.2}$$

$$\Delta \boldsymbol{c}_p = \boldsymbol{Q}_{jj}\boldsymbol{w}_j = -\boldsymbol{A}_{jj}^{-1}\boldsymbol{w}_j \tag{6.3}$$

$$\boldsymbol{w}_{l,ind} = -\boldsymbol{\Phi}_{lj}\boldsymbol{B}_{jj}\Delta\boldsymbol{c}_p = -\boldsymbol{\Phi}_{lj}\boldsymbol{B}_{jj}\boldsymbol{Q}_{jj}\boldsymbol{w}_j \tag{6.4}$$

$$\boldsymbol{D_i} = \frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{w}_{l,ind} = -\frac{1}{2}\rho V_\infty^2 \boldsymbol{S}\boldsymbol{\Phi}_{lj}\boldsymbol{B}_{jj}\boldsymbol{Q}_{jj}\boldsymbol{w}_j \tag{6.5}$$

- Additionally, viscous drag was also deemed irrelevant in load alleviation or flutter suppression and therefore also neglected. For flight dynamics it is important to include as it causes the phugoid to be damped out. Viscous drag can be added straightforwardly, by adding the drag formula in the aerodynamic reference frame in the x-direction. Two choices can be made in this regard. For a more accurate trim solution, it can be added as a bias term, as shown in Equation 6.6. For more accurate flight dynamics, it needs to compute the dynamic pressure based on the body velocity and also apply it in the x-axis of the aerodynamic reference frame, as shown in Equation 6.7.

$$\boldsymbol{D}_v = \begin{bmatrix} \frac{1}{2}\rho V_\infty^2 S C_{D_0} \\ 0 \\ 0 \end{bmatrix} \tag{6.6}$$

$$\boldsymbol{D}_v = \begin{bmatrix} \frac{1}{2}\rho S C_{D_0} (\boldsymbol{V}\cdot\boldsymbol{V}) \\ 0 \\ 0 \end{bmatrix} \tag{6.7}$$

- The gust modeling currently assumes a gust acting in the z-direction of the body frame. Lateral gusts could also be modeled by including two extra gust channels in the model with the corresponding gust states. Head-on gusts, or rather gust components in the longitudinal direction, are not possible to be modeled as the aircraft is modeled using flat vertical and horizontal panels. The former was not included as the 1-cos gust in the vertical direction is critical, and the head-on gust is often not the sizing gust, and therefore also deemed unimportant.

- The Padé approximation approach for modeling the gusts comes with two downwsides. First, the approximation can introduce so-called ringing or oscillatory behaviour in the response, which is unphysical. As was observed in the model verification, this has not been an issue in this work in the performed analyses. Secondly, it must be checked that the number of gust zones is sufficient for modeling the gust penetration effect. An alternative to the gust zones with a Padé approximation, which completely bypasses these nuisances is application of the Loewner framework, as derived by Quero, Vuillemin, and Poussot-Vassal [66]. As opposed to approximating the unsteady aerodynamics using an RFA, the authors used tangential interpolation through the Loewner framework. Although the details are out of scope of this thesis, using this approach, a minimal order realization is obtained, and no poles and resulting lag states have to be selected, and the fit is much better. It could be an excellent modification of the model to implement this methodology. It was not implemented in this thesis due the time constraints in this work as this method appears quite demanding to implement. Moreover, although this is not a particularly strong argument, the method in this thesis builds upon established methods, while also yielding a considerably smaller state-space model.

- The approach to reducing the order of the lag states comes at the cost of losing the ability to extract the lift distribution at the panel level. As one can still extract the cutting forces and moments, this is deemed not a problem as relatively few cut loads are monitored in an aeroservoelastic control problem, like the WRBM or the WRTM, but it could be a slight nuisance in interpretation of the aerodynamic behaviour. A potential solution would be to build a state-space model without reducing the lag states and use that as an analysis tool. The focus of the model in this thesis is on interpretability. However, for the aforementioned reasons one could argue that in fact interpretability is reduced.

- The downwash computation used in Section 3.6 results in linear boundary conditions in order to represent the flow-tangency condition, which was also stated in the model assumptions. However, this may not be a realistic assumption, depending on the desired maneuver and gust, as nonlinearities between the displacement and motion states are not captured. For example, the increased dihedral of the wings due to wing bending in trim is not captured in this model. Additionally, the effect of camber and twist is added as a separate downwash contribution, while in fact this should affect the panel normal vector in computing the downwash. This camber and twist therefore becomes a bias term in the state and output equation, which feels slightly off as it is an aerodynamic term. The following method could be used to capture nonlinearities in the boundary conditions. Initially, it was thought that this had not been applied yet in these models. However, Van Zyl and Mathews [69] and Neto [30] compute the downwash similarly. However, this downwash method was deemed too risky to implement considering the time constraint in this thesis work, as it would have required a major overhaul of state-space model. Additionally, the effect on the response may be quite limited. Especially during maneuvers and gusts, which result in relatively simple motion of the aircraft, this approach may not result in a different response. It is expected to cause a difference in flight dynamic behaviour, lateral gusts, and combined gusts and maneuvers. The main idea is shown below. The downwash is computed as the dot product between the panel normal vector and the induced velocities at the panel. The derivative can also be computed by applying the chain rule. The panel normal vector can be rotated for example using Rodrigues' rotation formula using the displacement states, and the induced velocities can be computed similarly as done in the linear model, using the corresponding splines and motion states.

$$w_j = \boldsymbol{n}_j \cdot \boldsymbol{v}_j \tag{6.8}$$

$$\dot{w}_j = \dot{\boldsymbol{n}}_j \cdot \boldsymbol{w}_j + \boldsymbol{n}_j \cdot \dot{\boldsymbol{w}}_j \tag{6.9}$$

$$\boldsymbol{n}_r = \boldsymbol{n}_0 \cos\theta + (\boldsymbol{k} \times \boldsymbol{n}_0)\sin\theta + \boldsymbol{k}(\boldsymbol{k} \cdot \boldsymbol{n}_0)(1 - \cos\theta) \tag{6.10}$$

6.3. PIPELINE REVISION & GENERALIZATION

The core contribution of this thesis is in the derivation of the mathematical model, the state-space representation while simultaneously meeting the aeroservoelastic model requirements in Subsection 2.3.1. The pipeline for generating free-flying aeroelastic aircraft models based on NASTRAN was an added contribution, which is of practical rather than academic relevance. However, the state-space model is useless without an aircraft to use as input to the framework, and therefore this pipeline is a critical cornerstone in the modeling process. At the start of this work it was chosen to reuse the preprocessor of DLR Loads Kernel to save the redundant work of importing and processing NASTRAN inputs, as well as generating the other matrices and parameters that serve as input to the framework. However, the setup of the pipeline as shown in Section 4.1 can be improved in several regards, of which most aim to reduce the time to build the model.

- The preprocessor performs all steps required to build the DLR Loads Kernel input file sequentially. Additionally, all functions used to build up this file have many side effects. It would be much better if this program more strictly adheres to object-oriented principles, such that each function can be executed individually without breaking the software. This is advantageous as then the software does not have to rebuild the aerodynamics model, if only changes are made to the structural model, for instance. Generally, the preprocessor of Loads kernel suffers from the same software architecture issues identified in Subsection 2.3.3 for the main module, with long functions having many responsibilities, functions with side effects, and global variables.

- Lastly, it might be possible to completely do away with the Loads Kernel preprocessor if one generates all required matrices and parameters directly from NASTRAN, and extract them using DMAP ALTERs. Further research into this possibility is required, but at the current moment there is nothing to believe NASTRAN is any less capable of building these matrices than Loads Kernel. Additionally, Loads Kernel contains a lot of bells and whistles that are not required to generate the inputs to this framework, so only writing the code that is required to generate the inputs to this model is also entirely feasible. After all, the state-space model is formulated using mathematics, and therefore in a programming language-agnostic manner. Additionally, solely using NASTRAN allows verification with industry-standard software that is well validated, and allows use of the advanced model checks built into FEM preprocessors such as FEMAP.

- Additionally, it could be interesting to further test integration with Simcenter Zona to see how this software can be used to offload some of the Loads Kernel functionalities, such as building the AIC matrices for VLM/DLM, and perhaps using a more intricate aerodynamic mesh. The downwash formulation may have to be changed in this case, however, which needs to be verified.

- The aerodynamic mesh generation merely generates the mesh, although it would be useful to also build tools that can automatically verify the appropriateness of the mesh for use with DLM, as described in Voß [20] and Demasi [61].

- The wing mirroring code in PyNastran contains errors, such as flipping the wrong element normals, and does not mirror the MCIDs, MAT2s, and PSHELLs. In future versions of this pipeline it would be better wise to only rely on PyNastran for creating, updating, reading, and writing cards, and not relying on the processing capabilities of PyNastran, and writing these parts separately.

Improvements can also be made to the aircraft that was generated, which uses the Embraer Benchmark Wing and publicly available data of the E195-E2. Most of these are outside of control and scope, however.

- First, the resulting aircraft has an aspect ratio of $AR = 11.87$. Although on the higher side compared to current generation aircraft, the potential for Active Flutter Suppression and Active Load Alleviation controllers is likely greater with higher aspect ratios. This would require modifying the Embraer Benchmark Wing finite-element model, which would be feasible, but nontrivial. An alternative would be to

apply the same method for generating the free-flying aircraft using the uCRM benchmarks wings by Brooks, Kenway, and Martins [106].

- Although multiple trailing-edge surfaces are implemented for effective MLA, the finite-element model only models outboard ailerons in a structural sense. Additionally, nearest neighbour splining is used, such that it might be that the spline mapping near the ailerons is not fully physical. The first was deemed not an issue as the goal of the model is to properly represent the dynamics, not computing the most accurate load paths.

- Although the aim was to make the aircraft as realistic as possible, it is unknown to what degree this has been achieved, as there is no real world data to validate the aircraft on, such as from the E195-E2, on which the aircraft geometry was based.

- On a similar note, one of the most obvious improvements that can be made in this aircraft, is to conceive or use a more intricate method for estimating the non-wing inertia of the aircraft. Currently, a thin-walled hollow tube at two thirds of the fuselage radius is used, while this may not be an accurate assumption.

- The wing currently has limited RBE3 elements towards the tips of the wings, which can be increased for more accurate aerostructural coupling, although the effect this will have may be limited. Nevertheless, it is an improvement that can be made.

- Much effort was put into ensuring that tailored wings can be input to the framework to output tailored aircraft. However, it turned out at the last phase of the project that the tailored wing in fact gave an unfavourable gust response, showing wash-in rather than wash-out, compared to the baseline wing. Therefore, no results regarding untailored versus tailored are presented. Further investigation into the tailoring framework should uncover whether this is due to the gust loads not being used as a constraint or whether there is some kind of sign error describing the bend-twist coupling. It was found that negating the G13 and G23 terms of the MAT2 card did yield wash-out, reducing the WRBM in gust responses. As the tailoring was performed outside of this work, no further investigation into it was performed.

6.4. APPLICATIONS & FUTURE WORK

The free-flying aeroservoelastic state-space model can be used for several applications, which is discussed in this section. For some applications, further additions to the framework are required, which are also highlighted. The model derived in this thesis can be considered the aeroservoelastic plant, which is exact, yet expressed using the smallest order possible. As such measurement delay, estimation, and noise are not added to this model, and recommended additions to the model when designing a controller. All of the mentioned applications below can be performed on any aircraft that can be imported into Loads Kernel. Additionally, using the aircraft generated in Chapter 4 specifically, the controllers can be designed on free-flying aircraft with aeroelastically tailored wings.

- A logical first application of the model is to design a Gust Load Alleviation (GLA) controller, which was intended as a third part of this work, but not started on due to unforeseen delays. It would be recommended to set up the state-space model in MATLAB/Simulink in order to use the control tools of MATLAB. The original research plan included building GLA conroller using Incremental Nonlinear Dynamic Inversion (INDI) in the inner loop and a robust controller, such as H_∞ or μ-Synthesis in the outer loop, using feedback from the acceleration sensors that are placed on the wing. This should require minimal modifications and extensions to the state-space model, as it was intended to be used for this purpose. One of these additions would to model sensor noise and delay, as well as air data measurement modeling, similar to the implementation of the model by Wuestenhagen [10], which was mentioned earlier in this chapter as a model addition.

- A second interesting application could be to design a complementary Maneuver Load Alleviation (MLA) controller, although this was not further investigated in this thesis, although also for this application no further modifications to the model should be required. The aircraft modeled in Chapter 4 features four trailing-edge surfaces to specifically enable effective MLA in future work with this aircraft and state-space model.

- If one wants to design an Active Flutter Suppression (AFS) controller, this would require further augmentation of the framework. First, state-space models at a certain Mach number can be obtained, and by sweeping over either speed or altitude for this given Mach number the flutter point can be obtained, by computing the eigenvalues. This sweeping code has not been implemented, however. In order to achieve this, it would be recommended to reformulate the model such that the velocity and air density terms are only multiplied at the last step, when assembling the model. For instance, the state matrix could be divided up into Equation 6.11, with F denoting fixed terms, K denoting stiffness terms, C denoting damping terms, and M denoting mass terms. Secondly, it would be useful to interpolate the state-space models as a function of the Mach number, such that a continuous flutter boundary can be constructed. To be able to cast the model into the form below, the gust states should be removed as they are not required and complicate casting the model into this form.

$$\boldsymbol{A} = \boldsymbol{A}_F + \rho V_\infty^2 \boldsymbol{A}_K + \rho V_\infty \boldsymbol{A}_C + \rho \boldsymbol{A}_M \tag{6.11}$$

- Aeroelastic tailoring is a high-dimensional optimization problem in which objectives such as weight minimization or bend-twist coupling maximization are pursued subject to multiple constraints, such aeroelastic stability, aileron effectiveness, and strength requirements. Incorporating control laws of GLA, MLA, and AFS into this optimization problem, introduces the possibility of satisfying these constraints while further improving the performance objective, as the feasible design space expands. In this approach, the control laws are therefore designed in conjunction with the wing structural properties. This methodology is referred to as Control Co-Design (CCD), or integrated aeroservoelastic optimization. CCD with a shell-based FEM is computationally infeasible and thus an alternative approach is required. Therefore, it is recommended to generate state-space models using lower-fidelity structural models, such as stick or beam models. The framework lends itself also to using these models, as long as the splines are properly computed. Additionally, if no modal analysis is applied, the derivation should be slightly adjusted, but given the extensive derivation in this thesis, that should not be pose difficulties. Furthermore, an additional interesting area of potential future work in this regard is the integration with JAX, like FENIAX [91], which integrates well with the matrix-based nature of the state-space model of this work. No further review of JAX was performed in this thesis, such that this is merely an idea. It is known however that the learning curve of JAX is steep, and GPU acceleration is not available on Windows, complicating its application.

- The model currently uses panel aerodynamics, with VLM for steady aerodynamics and DLM for unsteady aerodynamics. Both of these are mature methods, and not particularly state-of-the-art. Linearized Frequency-Domain CFD (LFD) [82]–[85] provides more accurate aerodynamics, as CFDcan resolve more complex geometry, viscous effects, and transonic effects, and would thus be an upgrade in aerodynamic fidelity, even though LFD linearizes the flow around a steady RANS flow. The model would have to be slightly rederived to generate state-space models using linearized frequency-domain CFDaerodynamics. Although not comprehensive by any means, the steps should be as follows. First, a suitable CFDplatform needs to be identified, which is capable of linearized CFD , which is likely the primary bottleneck in this area of future work, unless access is available to CFD software such as DLR TAU or CODA. Subsequently, around the steady base flow small-amplitude harmonic perturbations must be applied in each of the model states and inputs that affect the aerodynamics. From this, the GAF matrix can be constructed, on which an RFA could be applied, for instance, are another interpolation method. This would naturally require some adjustments in the framework, although the global structure of the model would be retained, and it would likely thus be merely a matter of reorganizing the aerodynamic matrices, but this would no doubt be an interesting and challenging undertaking.

7

CONCLUSION

Future commercial transport aircraft will feature increasingly higher aspect ratios as well reduced wing weight, which increases wing flexibility. This wing flexibility increases structural loads at the root and such wings are more prone to aeroelastic instabilities. Control surfaces can be actuated as to suppress flutter and alleviate wing loads. In order to synthesize such controllers, state-space models describing the free-flying aeroservoelasticity of these aircraft are required. Chapter 2 surveyed the modeling approaches and existing tools and frameworks with similar objectives, such as tools for modeling flexible aircraft flight dynamics and aircraft loads models. DLR Loads Kernel [19] was chosen as a reference aircraft loads model framework to further refine to make it suitable for free-flying aeroservoelasticity. DLR Loads Kernel is an excellent tool for loads analysis, but several issues were identified that make it less suitable for aeroservoelasticity. Most importantly, DLR Loads Kernel lacks a model explicitly formulated in state-space representation, and a preceding rigorous derivation of this model. Moreover, the computational efficiency of the software is low as analyses can take on the order of minutes. Because the formal state-space model is lacking, it is only possible to import an LTI state-space model into MATLAB/Simulink, whereas it would be desirable to also have the nonlinear model in this environment. Additionally, the number of gust disturbance inputs is too high and physical RFA is used, which leads to an excessive number of lag states, while the implemented half-generalized RFA only yields an approximation. Additionally, actuator dynamics and an accelerometer sensor model is missing. A gap less academic in nature but a practical necessity, is that DLR Loads Kernel tailored for use with the in-house DLR ModGen software, which complicates the use of the tool outside of DLR. Based on these limitations, the following research questions were formulated, and executed in this work.

1. **How can the contemporary aircraft loads model be formally rederived into an explicit, closed-form, symbolic continuous-time state-space model, based on matrix and parameter inputs from DLR Loads Kernel?**

   Chapter 3 addresses this question by formally rederiving and refining the aircraft loads model, for which the model implemented in DLR Loads Kernel is taken as a guiding example, into a monolithic continuous-time state-space model. The model in this work uses inputs from DLR Loads Kernel. However, the model is formulated as a general framework such that the required matrices and parameters could also be sourced elsewhere.

   The final model is based on nonlinear flight dynamics using mean-body axes, linear finite-element model based on NASTRAN, expressed using modal analysis, and panel aerodynamics using VLM for steady aerodynamics and DLM for unsteady aerodynamics. Formal rederivation of the model first required defining the state, input, output, and disturbance vectors. The derivation then starts from the governing equations of motion, which are expressed in vector–matrix form.

   The major objective of the state-space derivation was the aerodynamical coupling of the three governing equations as to create a monolithic state-space model. In contrast to Loads Kernel and most other aircraft loads models, the model is derived exclusively using matrix operations, which enabled lumping intermediate matrix products directly into the matrices defining the state-space model, resulting in a clean state-space formulation. As the model is expressed purely in vector–matrix form, it could be cast directly into an explicit, closed-form, symbolic, continuous-time state-space model.

This approach has translated itself into five primary benefits. First, the mathematical derivation and state-space formulation improves model interpretability, traceability, and also extensibility. Secondly, as the model was mathematically derived, the unaccounted-for model simplifications in Loads Kernel, such as the downwash derivative being computed using backward-difference, the displacement-downwash formula not derived using first principles, the neglected control surface motion, among others highlighted in Subsection 2.3.3, were naturally resolved by this approach. Thirdly, lumping the matrix operations into equivalent matrices also breaks the dependency of the model performance on the number of aerodynamic panels or the size of the structural mesh. Fourth, as the model was formulated using vector equations and matrices, the underlying structure of the equations became evident which proved instrumental in observing that the physical RFA could be cast into an exact, but reduced-order representation, as is discussed in the conclusion of the second research question. Lastly, the nonlinear model can be easily ported to MATLAB/Simulink. The downside of this modeling approach is the lost ability to extract intermediate computational results, such as the nodal loads.

2. **How can this aircraft loads state-space model be further modified and/or augmented to be converted to become a free-flying aeroservoelastic state-space model, in line with the aeroservoelastic model requirements set in Subsection 2.3.1?**

   In Chapter 3, simultaneously with the derivation of the state-space model, four additions to the model were made in this regard, directly addressing this question. First, spiral nature of the Sears function was approximated using cascaded gust zones [64] using a Padé approximation [10], [65]. This required discretizing the aircraft longitudinally into gust zones. This approach has reduced the number of gust disturbance inputs from n to 2, but also increases the number of states by $2z$, with n aerodynamic panels and z gust zones. The massive number of lag states, $n \times p$, with p RFA poles, due to the usage of physical RFA was solved through projecting the lag dynamics to lag force and moment dynamics. As a result the lag states were replaced by aerodynamic rigid-body force lag, aerodynamic rigid-body moment lag, aerodynamic generalized force lag, and aerodynamic monitoring load lag. Through this exact transformation the lag states were reduced to $(6 + m + s) \times p$, with m the flexible modes, and s the number of cutting forces and moments to be output, while enabling the use of the superior RFA. Although this leads to a massive reduction in lag states, the downside is that for each monitored cutting force or moment, a new lag state needs to be introduced. As the number of monitored cutting forces and moments is typically low for aeroservoelastic control, as opposed to in loads analysis, this was deemed not an issue. Lastly, the state-space model was also augmented with second-order actuator dynamics, and a sensor model for accelerometers placed on the nodes of the finite-element model was also added.

3. **How can clamped-wing finite-element models be systematically extended to a free-flying finite-element model that allows generation of free-flying aeroservoelastic state-space models for different wing mass and stiffness distributions, mass cases, and flight conditions?**

   Chapter 4 was devoted to devising an end-to-end modeling pipeline, starting with a clamped-wing finite-element model, the Embraer Benchmark Wing, up to the generation of a free-flying aeroservoelastic state-space model of Chapter 3. The model was based on publicly available data of the Embraer E195-E2. The devised modeling pipeline is depicted in Figure 4.1, which directly addresses this question. The structural processing step, as discussed in Section 4.2 included extending the clamped-wing model into a free-flying finite-element model, a seemingly simple, yet nontrivial task, as it turned out. Essential steps included mirroring the wing, joining them with rigid elements per node, modeling the fuselage and tail as rigid elements, and adding representative mass and inertia to the model. Additionally, nodes used as sensors, engine attachment points, and for aerostructural coupling are identified. Mass cases, as discussed in Subsection 4.2.4, were defined separately, for which the passenger seating and cargo holds were parameterized as well as the fuel in the tanks, enabling the generation of different mass cases. An aerodynamic mesh generator was also built, which requires parameterization of the aircraft geometry, as exemplified in Section 4.3. The capability was also added to generate free-flying finite-element models with aeroelastically tailored wings, which were generated using PROTEUS and NASTRAN SOL200, which was covered briefly in Subsection 4.2.3. The resulting aircraft model was discussed in Section 4.4. Executing this pipeline, once the models are set up, is almost fully automated, with the result being input files for the state-space model containing the system matrices, which can be obtained for different mass cases, flight conditions, and for different wing mass and stiffness distributions.

4. **To what extent does the derived state-space model replicate the dynamics of DLR Loads Kernel, how can discrepancies be explained, and how high is the computational efficiency improvement through this formulation?**

   As the model improvements should not degrade the model behaviour, this is a critical question, which is answered in Chapter 5. Overall, the state-space model derived in this thesis delivers on the expected computational efficiency improvements due the monolithic state-space formulation, yielding a 100-400x improvement in simulation time and a 458x reduction in model input size, using the aircraft generated in Chapter 4 and the considered simulations. As this only applies to the specific simulations performed, these figures may be even higher or lower in other simulations, but it definitely constitutes at least two orders of magnitude of an improvement. Two additional remarks regarding these figures apply, however. First, the aircraft of Chapter 4 has a relatively high number of aerodynamic panels at about 3000 and a high number of structural nodes at around 75000, for which the state-space model of this thesis is particularly advantageous as the simulation performance is decoupled from the size of the aerodynamic and structural meshes, while the relative improvement in performance is expected to be lower for lower-fidelity models. Secondly, the DLR Loads Kernel input file is still required to generate the state-space model, but this file is only accessed during an offline preprocessing step, after which all subsequent analyses rely solely on the significantly reduced input file. In terms of the model behaviour, the coupled aeroelastic and flight dynamic behaviour agree very well, with nearly identical eigenvalues, frequencies and damping ratios. Moreover, time-domain simulations for elevator, aileron, and rudder step inputs as well as a 1-cos gust further demonstrate that the models are nearly identical. The state-space model therefore provides an excellent model for future applied work in control design for free-flying flexible aircraft with active flutter suppression and/or load alleviation, and a mathematical basis on which to further augment or refine this model.

Finally, Chapter 6 discusses the limitations of the model imposed by the core assumptions, as well as modifications to the model that would improve it. Critical assumptions and limitations are related to the linearity of the panel aerodynamic and finite-element model. Tangible improvements in terms of the state-space model include adding viscous drag, induced drag, using the Loewner framework for gust modeling, adjusting the downwash calculation method, adding continuous turbulence, modeling inertial coupling, making the actuator dynamics nonlinear, and modeling measurement delay, noise and estimation. In terms of the modeling pipeline, the biggest improvement can be attained by generating the input matrices to the state-space model without DLR Loads Kernel, and solely using NASTRAN and/or ZAERO instead, which should be entirely feasible. In terms of the aircraft modeled in Chapter 4, improvements would be to make the wing finite-element model of higher aspect ratio, using a more accurate method for modeling nonwing inertia, and validation of the aircraft with respect to the E195-E2. However, the last would be difficult considering that data is proprietary. In terms of applications and future work, natural applications include designing Active Flutter Suppression, Maneuver Load Alleviation, and Gust Load Alleviation controllers. With knowledge from those applications, a model revision can be made to make it suitable for Control Co-Design and using higher fidelity aerodynamics using Linearized Frequency-Domain CFD.

BIBLIOGRAPHY

[1] V. Grewe, A. Gangoli Rao, T. Grönstedt, *et al.*, "Evaluating the Climate Impact of Aviation Emission Scenarios Towards the Paris Agreement Including COVID-19 Effects," *Nature Communications*, vol. 12, no. 1, p. 3841, Jun. 22, 2021, ISSN: 2041-1723. DOI: 10.1038/s41467-021-24091-y.

[2] N. Beck, T. Landa, A. Seitz, L. Boermans, Y. Liu, and R. Radespiel, "Drag Reduction by Laminar Flow Control," *Energies*, vol. 11, no. 1, p. 252, Jan. 20, 2018, ISSN: 1996-1073. DOI: 10.3390/en11010252.

[3] Y. Ma and A. Elham, "Designing High Aspect Ratio Wings: A Review of Concepts and Approaches," *Progress in Aerospace Sciences*, vol. 145, p. 100 983, Feb. 2024, ISSN: 03760421. DOI: 10.1016/j.paerosci.2024.100983.

[4] "Certification Specifications and Acceptable Means of Compliance for Large Aeroplanes CS-25," European Aviation Safety Agency, 2023.

[5] A. R. Collar, "The Expanding Domain of Aeroelasticity," *The Journal of the Royal Aeronautical Society*, vol. 50, no. 428, pp. 613–636, 1946. DOI: 10.1017/S0368393100120358.

[6] B. Etkin and L. D. Reid, *Dynamics of Flight*, 3rd ed. John Wiley & Sons, 1996, ISBN: 0-471-03418-5.

[7] M. R. Waszak and D. K. Schmidt, "Flight Dynamics of Aeroelastic Vehicles," *Journal of Aircraft*, vol. 25, no. 6, pp. 563–571, Jun. 1988, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/3.45623.

[8] E. Livne and T. A. Weisshaar, "Aeroelasticity of Nonconventional Airplane Configurations–Past and Future," *Journal of Aircraft*, vol. 40, no. 6, pp. 1047–1065, Nov. 2003, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/2.7217.

[9] M. Natella, "Aeroelastic Tailoring of Composite Aircraft," Ph.D. dissertation, Delft University of Technology, Delft, Netherlands, 2020. DOI: 10.4233/uuid:48af4e9b-1487-4402-a1aa-19e302b0eb97.

[10] M. Wuestenhagen, "Gust Load Alleviation Control of Aircraft with Varying Mass Distribution," presented at the AIAA SciTech Forum, National Harbor, Maryland: American Institute of Aeronautics and Astronautics, Jan. 23, 2023, ISBN: 978-1-62410-699-6. DOI: 10.2514/6.2023-0371.

[11] J. Xu and I. Kroo, "Aircraft Design with Active Load Alleviation and Natural Laminar Flow," *Journal of Aircraft*, vol. 51, no. 5, pp. 1532–1545, Sep. 2014, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C032402.

[12] T. E. Noll, J. M. Brown, M. E. Perez-Davis, S. D. Ishmael, G. C. Tiffany, and M. Gaier, "Investigation of the Helios Prototype Aircraft Mishap," National Aeronautics and Space Administration, Jan. 2004.

[13] C. D. Regan and C. V. Jutte, "Survey of Applications of Active Control Technology for Gust Alleviation and New Challenges for Lighter-weight Aircraft," NASA Dryden Flight Research Center, Technical Memorandum 216008, 2012.

[14] D. McLean and R. Prasad, "A Structure Load Alleviation Control Sytem for a Large Aircraft," *Transactions of the Institute of Measurement and Control*, vol. 2, no. 1, pp. 25–37, Jan. 1980, ISSN: 0142-3312, 1477-0369. DOI: 10.1177/014233128000200104.

[15] A. D. R. De Souza, P. Vuillemin, C. Poussot-Vassal, and D. Quero, "Gust Load Alleviation Using Reduced-Order Aeroelastic Models and Observer-Based Robust Control," *Journal of Guidance, Control, and Dynamics*, vol. 46, no. 5, pp. 949–957, May 2023, ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.G007153.

[16] H. Fournier, P. Massioni, M. Tu Pham, L. Bako, R. Vernay, and M. Colombo, "Robust Gust Load Alleviation of Flexible Aircraft Equipped with LiDAR," *Journal of Guidance, Control, and Dynamics*, vol. 45, no. 1, pp. 58–72, Jan. 2022, ISSN: 1533-3884. DOI: 10.2514/1.G006084.

[17] H.-G. Giesseler, M. Kopf, P. Varutti, T. Faulwasser, and R. Findeisen, "Model Predictive Control for Gust Load Alleviation," *IFAC Proceedings Volumes*, vol. 45, no. 17, pp. 27–32, 2012, ISSN: 14746670. DOI: 10.3182/20120823-5-NL-3013.00049.

[18] Y. Beyer, M. Steen, and P. Hecker, "Boosted Incremental Nonlinear Dynamic Inversion for Flexible Airplane Gust Load Alleviation," *Journal of Guidance, Control, and Dynamics*, vol. 47, no. 7, pp. 1394–1413, Jul. 2024, ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.G007984.

[19] A. Voß, "Loads Kernel User Guide," DLR Institute of Aeroelasticity, DLR-IB-AE-GO-2020-136, Oct. 7, 2020.

[20] A. Voß, "An Implementation of the Vortex Lattice and the Doublet Lattice Method," DLR Institute of Aeroelasticity, DLR-IB-AE-GO-2020-137, Oct. 7, 2020.

[21] G. H. C. Silva, A. P. Do Prado, P. H. Cabral, R. De Breuker, and J. K. S. Dillinger, "Tailoring of a Composite Regional Jet Wing Using the Slice and Swap Method," *Journal of Aircraft*, vol. 56, no. 3, pp. 990–1004, May 2019, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C035094.

[22] H. Maathuis, S. G. P. Castro, and R. De Breuker. "Exploring Multi-Fidelity Aeroelastic Tailoring: Prospect and Model Assessment." arXiv: 2411.03247 [cs]. (Nov. 5, 2024), pre-published.

[23] F. Afonso, J. Vale, É. Oliveira, F. Lau, and A. Suleman, "A Review on Non-Linear Aeroelasticity of High Aspect-Ratio Wings," *Progress in Aerospace Sciences*, vol. 89, pp. 40–57, Feb. 2017, ISSN: 03760421. DOI: 10.1016/j.paerosci.2016.12.004.

[24] L. Meirovitch and I. Tuzcu, "Unified Theory for the Dynamics and Control of Maneuvering Flexible Aircraft," *AIAA Journal*, vol. 42, no. 4, pp. 714–727, Apr. 2004, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.1489.

[25] A. B. G. Neto, R. G. A. Silva, P. Paglione, and F. J. Silvestre, "Formulation of the Flight Dynamics of Flexible Aircraft Using General Body Axes," *AIAA Journal*, vol. 54, no. 11, pp. 3516–3534, Nov. 2016, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J054752.

[26] C. Reschke, "Flight Loads Analysis with Inertially Coupled Equations of Motion," presented at the AIAA Atmospheric Flight Mechanics Conference and Exhibit, San Francisco, California: American Institute of Aeronautics and Astronautics, Aug. 15, 2005, ISBN: 978-1-62410-055-0. DOI: 10.2514/6.2005-6026.

[27] F. Saltari, C. Riso, G. D. Matteis, and F. Mastroddi, "Finite-Element-Based Modeling for Flight Dynamics and Aeroelasticity of Flexible Aircraft," *Journal of Aircraft*, vol. 54, no. 6, pp. 2350–2366, Nov. 2017, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C034159.

[28] F. J. Silvestre and R. Luckner, "Experimental Validation of a Flight Simulation Model for Slightly Flexible Aircraft," *AIAA Journal*, vol. 53, no. 12, pp. 3620–3636, Dec. 2015, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J054023.

[29] T. M. Kier, M. Wüstenhagen, Ö. Süelözgen, *et al.*, "Aeroservoelastic Models for Design, Testing, Flight Test Clearance and Validation of Active Flutter Suppression Control Laws," presented at the International Forum on Aeroelasticity and Structural Dynamics, The Hague, Netherlands, Jun. 17, 2024.

[30] A. B. G. Neto, "Simplified Integrated Model of the Flight Dynamics of Flexible Aircraft," *Aerospace Science and Technology*, vol. 161, p. 110 115, Jun. 2025, ISSN: 12709638. DOI: 10.1016/j.ast.2025.110115.

[31] A. Castrichini, T. Wilson, F. Saltari, F. Mastroddi, N. Viceconti, and J. E. Cooper, "Aeroelastics Flight Dynamics Coupling Effects of the Semi-Aeroelastic Hinge Device," *Journal of Aircraft*, vol. 57, no. 2, pp. 333–341, Mar. 2020, ISSN: 1533-3868. DOI: 10.2514/1.C035602.

[32] D. Castillo Zúñiga, A. Souza, and L. Góes, "Flight Dynamics Modeling of a Flexible Wing Unmanned Aerial Vehicle," *Mechanical Systems and Signal Processing*, vol. 145, p. 106 900, Nov. 2020, ISSN: 08883270. DOI: 10.1016/j.ymssp.2020.106900.

[33] F. J. Silvestre, A. B. G. Neto, R. M. Bertolin, R. G. A. Da Silva, and P. Paglione, "Aircraft Control Based on Flexible Aircraft Dynamics," *Journal of Aircraft*, vol. 54, no. 1, pp. 262–271, Jan. 2017, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C033834.

[34] T. Kier, "Comparison of Unsteady Aerodynamic Modelling Methodologies with Respect to Flight Loads Analysis," presented at the AIAA Atmospheric Flight Mechanics Conference and Exhibit, San Francisco, California: American Institute of Aeronautics and Astronautics, Aug. 15, 2005, ISBN: 978-1-62410-055-0. DOI: 10.2514/6.2005-6027.

[35] T. M. Kier, "An Integrated Flexible Aircraft Model for Optimal Control Surface Scheduling of Manoeuvre Load Alleviation and Wing Shape Control Functions," presented at the International Forum on Aeroelasticity and Structural Dynamics, Madrid, Spain, 2022.

[36] T. M. Kier, "An Integrated Model for Lateral Gust Loads Analysis and Dutch Roll Flight Dynamics Using a 3D Panel Method," presented at the International Forum on Aeroelasticity and Structural Dynamics, Como, Italy, 2017.

[37] T. Kier, J. Hofstee, C. Cerulli, and G. Looye, "A Variable, Fully Flexible Dynamic Response Tool for Special Investigations VarLoads," presented at the International Forum on Aeroelasticity and Structural Dynamics, Amsterdam, Netherlands, 2003.

[38] T. M. Kier, "Integrated Flexible Dynamic Maneuver Loads Models based on Aerodynamic Influence Coefficients of a 3D Panel Method," presented at the 56th AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 5, 2015, ISBN: 978-1-62410-342-1. DOI: 10.2514/6.2015-0185.

[39] T. Kier and G. Looye, "Unifying Manoeuvre and Gust Loads Analysis Models," presented at the International Forum on Aeroelasticity and Structural Dynamics, Seattle, Washington, 2009.

[40] T. M. Kier, "An Integrated Modelling Approach for Flight Dynamics, Manoeuvre- and Gust-Loads Analysis," presented at the 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, Florida: American Institute of Aeronautics and Astronautics, Jan. 8, 2018, ISBN: 978-1-62410-532-6. DOI: 10.2514/6.2018-2209.

[41] W. Su and C. E. S. Cesnik, "Dynamic Response of Highly Flexible Flying Wings," *AIAA Journal*, vol. 49, no. 2, pp. 324–339, Feb. 2011, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J050496.

[42] S. Haghighat, J. R. R. A. Martins, and H. H. T. Liu, "Aeroservoelastic Design Optimization of a Flexible Wing," *Journal of Aircraft*, vol. 49, no. 2, pp. 432–443, Mar. 2012, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C031344.

[43] X. Wang, T. Mkhoyan, and R. De Breuker, "Nonlinear Incremental Control for Flexible Aircraft Trajectory Tracking and Load Alleviation," in *AIAA SciTech Forum*, Virtual Event: American Institute of Aeronautics and Astronautics, Jan. 11, 2021, ISBN: 978-1-62410-609-5. DOI: 10.2514/6.2021-0503.

[44] X. Wang, E. Van Kampen, Q. P. Chu, and R. De Breuker, "Flexible Aircraft Gust Load Alleviation with Incremental Nonlinear Dynamic Inversion," *Journal of Guidance, Control, and Dynamics*, vol. 42, no. 7, pp. 1519–1536, Jul. 2019, ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.G003980.

[45] G. Avanzini, F. Nicassio, and G. Scarselli, "Reduced-Order Short-Period Model of Flexible Aircraft," *Journal of Guidance, Control, and Dynamics*, vol. 40, no. 8, pp. 2017–2029, Aug. 2017, ISSN: 0731-5090, 1533-3884. DOI: 10.2514/1.G002387.

[46] E. A. M. Pinto, F. L. Cardoso-Ribeiro, and F. J. O. Moreira, "Comparative Analysis of Flight Maneuver Loads Between Flexible and Rigid Aircraft," presented at the International Forum on Aeroelasticity and Structural Dynamics, The Hague, 2024.

[47] G. Dussart, V. Portapas, A. Pontillo, and M. Lone, "Flight Dynamic Modelling and Simulation of Large Flexible Aircraft," in *Flight Physics - Models, Techniques and Technologies*, K. Volkov, Ed., InTech, Feb. 14, 2018, ISBN: 978-953-51-3807-5 978-953-51-3808-2. DOI: 10.5772/intechopen.71050.

[48] T. Theodorsen, "General Theory of Aerodynamic Instability and the Mechanism of Flutter," National Advisory Committee for Aeronautics, 496, 1949.

[49] D. A. Peters, S. Karunamoorthy, and W.-M. Cao, "Finite-State Induced Flow Models Part I: Two-Dimensional Thin Airfoil," *Journal of Aircraft*, vol. 32, no. 2, pp. 313–322, Mar. 1995, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/3.46718.

[50] J.-W. Lee, J.-S. Lee, J.-H. Han, and H.-K. Shin, "Aeroelastic Analysis of Wind Turbine Blades Based on Modified Strip Theory," *Journal of Wind Engineering and Industrial Aerodynamics*, vol. 110, pp. 62–69, Nov. 2012, ISSN: 01676105. DOI: 10.1016/j.jweia.2012.07.007.

[51] W. Su and C. E. S. Cesnik, "Nonlinear Aeroelasticity of a Very Flexible Blended-Wing-Body Aircraft," *Journal of Aircraft*, vol. 47, no. 5, pp. 1539–1553, Sep. 2010, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.47317.

[52] Z. Zhao and G. Ren, "Multibody Dynamic Approach of Flight Dynamics and Nonlinear Aeroelasticity of Flexible Aircraft," *AIAA Journal*, vol. 49, no. 1, pp. 41–54, Jan. 2011, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.45334.

[53] G. Avanzini, E. Capello, and I. A. Piacenza, "Mixed Newtonian–Lagrangian Approach for the Analysis of Flexible Aircraft Dynamics," *Journal of Aircraft*, vol. 51, no. 5, pp. 1410–1421, Sep. 2014, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C032235.

[54] W. Su, S. S.-M. Swei, and G. G. Zhu, "Optimum Wing Shape of Highly Flexible Morphing Aircraft for Improved Flight Performance," *Journal of Aircraft*, vol. 53, no. 5, pp. 1305–1316, Sep. 2016, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C033490.

[55] Y. Wang, A. Wynn, and R. Palacios, "Nonlinear Modal Aeroservoelastic Analysis Framework for Flexible Aircraft," *AIAA Journal*, vol. 54, no. 10, pp. 3075–3090, Oct. 2016, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J054537.

[56] Y. Wang, A. Wynn, and R. Palacios, "Nonlinear Aeroelastic Control of Very Flexible Aircraft Using Model Updating," *Journal of Aircraft*, vol. 55, no. 4, pp. 1551–1563, Jul. 2018, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C034684.

[57] C. Zhang, Z. Zhou, X. Zhu, and L. Qiao, "A Comprehensive Framework for Coupled Nonlinear Aeroelasticity and Flight Dynamics of Highly Flexible Aircrafts," *Applied Sciences*, vol. 10, no. 3, p. 949, Feb. 1, 2020, ISSN: 2076-3417. DOI: 10.3390/app10030949.

[58] J. Murua, R. Palacios, and J. M. R. Graham, "Applications of the Unsteady Vortex-Lattice Method in Aircraft Aeroelasticity and Flight Dynamics," *Progress in Aerospace Sciences*, vol. 55, pp. 46–72, Nov. 2012, ISSN: 03760421. DOI: 10.1016/j.paerosci.2012.06.001.

[59] J. Katz and A. Plotkin, *Low Speed Aerodynamics*. Mc-GrawHill, 1991, ISBN: 0-07-050446-6.

[60] E. Albano and P. Rodden, "A Doublet-Lattice Method for Calculating Lift Distributions on Oscillating Surfaces in Subsonic Flows.," *AIAA Journal*, vol. 7, no. 2, p. 279, 1969.

[61] L. Demasi, *Introduction to Unsteady Aerodynamics and Dynamic Aeroelasticity*. Cham, Switzerland: Springer Nature Switzerland, 2024, ISBN: 978-3-031-50053-4 978-3-031-50054-1. DOI: 10.1007/978-3-031-50054-1.

[62] D. H. Baldelli, P. C. Chen, and J. Panza, "Unified Aeroelastic and Flight Dynamic Formulation via Rational Function Approximations," *Journal of Aircraft*, vol. 43, no. 3, pp. 763–772, May 2006, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.16620.

[63] K. L. Roger, "Airplane Math Modeling Methods for Active Control Design," AGARD, AGARD/CP-228, Apr. 1977.

[64] M. Karpel, B. Moulin, and P. C. Chen, "Dynamic Response of Aeroservoelastic Systems to Gust Excitation," *Journal of Aircraft*, vol. 42, no. 5, pp. 1264–1272, Sep. 2005, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.6678.

[65] S. Schulz and D. Ossmann, "Estimation of Global Structural Aircraft Loads due to Atmospheric Disturbances for Structural Fatigue Estimation," presented at the International Forum on Aeroelasticity and Structural Dynamics, Madrid, Spain, Jun. 13, 22.

[66] D. Quero, P. Vuillemin, and C. Poussot-Vassal, "A Generalized State-Space Aeroservoelastic Model Based on Tangential Interpolation," *Aerospace*, vol. 6, no. 1, p. 9, Jan. 15, 2019, ISSN: 2226-4310. DOI: 10.3390/aerospace6010009.

[67] T. M. Kier, "Comparing Different Potential Flow Methods for Unsteady Aerodynamic Modelling of a Flutter Demonstrator Aircraft," presented at the AIAA SciTech Forum, National Harbor, Maryland: American Institute of Aeronautics and Astronautics, Jan. 23, 2023, ISBN: 978-1-62410-699-6. DOI: 10.2514/6.2023-0177.

[68] A. B. G. Neto, G. C. Barbosa, J. A. Paulino, *et al.*, "Flexible Aircraft Simulation Validation with Flight Test Data," *AIAA Journal*, vol. 61, no. 1, pp. 285–304, Jan. 2023, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J060960.

[69] L. H. Van Zyl and E. H. Mathews, "Aeroelastic Analysis of T-Tails Using an Enhanced Doublet Lattice Method," *Journal of Aircraft*, vol. 48, no. 3, pp. 823–831, May 2011, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C001000.

[70] G. Dimitriadis, S. Kilimtzidis, V. Kostopoulos, V. Laraspata, and L. Soria, "Flutter Calculations Using the Unsteady Source and Doublet Panel Method," *Journal of Aircraft*, vol. 62, no. 1, pp. 117–131, Jan. 2025, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C037891.

[71] A. Kotikalpudi, H. Pfifer, and G. J. Balas, "Unsteady Aerodynamics Modeling for a Flexible Unmanned Air Vehicle," presented at the AIAA Atmospheric Flight Mechanics Conference, Dallas, Texas: American Institute of Aeronautics and Astronautics, Jun. 22, 2015, ISBN: 978-1-62410-358-2. DOI: 10.2514/6.2015-2854.

[72] X. Changchuan, Y. Lan, L. Yi, and Y. Chao, "Stability of Very Flexible Aircraft with Coupled Nonlinear Aeroelasticity and Flight Dynamics," *Journal of Aircraft*, vol. 55, no. 2, pp. 862–874, Mar. 2018, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.C034162.

[73] Y. Liu and C. Xie, "Aeroservoelastic Stability Analysis for Flexible Aircraft Based on a Nonlinear Coupled Dynamic Model," *Chinese Journal of Aeronautics*, vol. 31, no. 12, pp. 2185–2198, Dec. 2018, ISSN: 10009361. DOI: 10.1016/j.cja.2018.08.019.

[74] Y. M. Meddaikar, J. K. Dillinger, T. Klimmek, *et al.*, "Aircraft Aeroservoelastic Modelling of the FLEXOP Unmanned Flying Demonstrator," presented at the AIAA SciTech Forum, San Diego, California: American Institute of Aeronautics and Astronautics, Jan. 7, 2019, ISBN: 978-1-62410-578-4. DOI: 10.2514/6.2019-1815.

[75] N. P. M. Werter, R. De Breuker, and M. M. Abdalla, "Continuous-Time State-Space Unsteady Aerodynamic Modeling for Efficient Loads Analysis," *AIAA Journal*, vol. 56, no. 3, pp. 905–916, Mar. 2018, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.j056068.

[76] J. Murua, R. Palacios, and J. Graham, "Open-Loop Stability and Closed-Loop Gust Alleviation on Flexible Aircraft Including Wake Modeling," presented at the 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, Hawaii: American Institute of Aeronautics and Astronautics, Apr. 23, 2012, ISBN: 978-1-60086-937-2. DOI: 10.2514/6.2012-1484.

[77] S. Maraniello and R. Palacios, "Optimal Rolling Maneuvers with Very Flexible Wings," *AIAA Journal*, vol. 55, no. 9, pp. 2964–2979, Sep. 2017, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J055721.

[78] A. Del Carre, A. Muñoz-Simón, N. Goizueta, and R. Palacios, "SHARPy: A Dynamic Aeroelastic Simulation Toolbox for Very Flexible Aircraft and Wind Turbines," *Journal of Open Source Software*, vol. 4, no. 44, p. 1885, Dec. 13, 2019, ISSN: 2475-9066. DOI: 10.21105/joss.01885.

[79] A. Voß and T. Klimmek, "Parametric Aeroelastic Modeling, Maneuver Loads Analysis using CFD Methods and Structural Design of a Fighter Aircraft," *Aerospace Science and Technology*, vol. 136, p. 108 231, May 2023, ISSN: 12709638. DOI: 10.1016/j.ast.2023.108231.

[80] L. Reimer, M. Ritter, R. Heinrich, and W. Krüger, "CFD-Based Gust Load Analysis for a Free-Flying Flexible Passenger Aircraft in Comparison to a DLM-Based Approach," presented at the 22nd AIAA Computational Fluid Dynamics Conference, Dallas, Texas: American Institute of Aeronautics and Astronautics, Jun. 22, 2015, ISBN: 978-1-62410-366-7. DOI: 10.2514/6.2015-2455.

[81] M. Ripepi, M. J. Verveld, N. W. Karcher, *et al.*, "Reduced-Order Models for Aerodynamic Applications, Loads and MDO," *CEAS Aeronautical Journal*, vol. 9, no. 1, pp. 171–193, Mar. 2018, ISSN: 1869-5582, 1869-5590. DOI: 10.1007/s13272-018-0283-6.

[82] K. Pausch, W. Weigold, B. Stickan, and D. Cantiani, "Correction of Linearized CFD-Based Unsteady Aerodynamic Models," presented at the AIAA SciTech Forum, Orlando, Florida: American Institute of Aeronautics and Astronautics, Jan. 12, 2026, ISBN: 978-1-62410-765-8. DOI: 10.2514/6.2026-1338.

[83] C. Kaiser, D. Quero, J. Nitzsche, and B. Stickan, "Analysis of Linearized Motion- and Gust-Induced Airloads with a Next-Generation Computational Fluid Dynamics Solver," presented at the International Forum on Aeroelasticity and Structural Dynamics, The Hague, Netherlands, Jun. 17, 2024.

[84] C. Kaiser, D. Friedewald, and J. Nitzsche, "Comparison of Nonlinear CFD with Time-Linearized CFD and CFD-Corrected DLM for Gust Encounter Simulations," presented at the International Forum on Aeroelasticity and Structural Dynamics, Como, Italy, Jun. 5, 2017.

[85] L. Daumas, N. Forestier, A. Bissuel, *et al.*, "Industrial Frequency-Domain Linearized Navier-Stokes Calculations for Aeroelastic Problems in the Transonic Flow Regime," presented at the International Forum on Aeroelasticity and Structural Dynamics, Como, Italy, Jun. 25, 2017.

[86] R. J. Guyan, "Reduction of Stiffness and Mass Matrices," *AIAA Journal*, vol. 3, no. 2, pp. 380–380, Feb. 1965, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/3.2874.

[87] W. Su and C. E. Cesnik, "Strain-Based Geometrically Nonlinear Beam Formulation for Modeling Very Flexible Aircraft," *International Journal of Solids and Structures*, vol. 48, no. 16–17, pp. 2349–2360, Aug. 2011, ISSN: 00207683. DOI: 10.1016/j.ijsolstr.2011.04.012.

[88] C. E. Cesnik and W. Su, "Nonlinear Aeroelastic Modeling and Analysis of Fully Flexible Aircraft," presented at the 46th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Austin, Texas: American Institute of Aeronautics and Astronautics, Apr. 18, 2005, ISBN: 978-1-62410-065-9. DOI: 10.2514/6.2005-2169.

[89] R. Palacios, J. Murua, and R. Cook, "Structural and Aerodynamic Models in Nonlinear Flight Dynamics of Very Flexible Aircraft," *AIAA Journal*, vol. 48, no. 11, pp. 2648–2659, Nov. 2010, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J050513.

[90] C.-S. Chang, D. H. Hodges, and M. J. Patil, "Flight Dynamics of Highly Flexible Aircraft," *Journal of Aircraft*, vol. 45, no. 2, pp. 538–545, Mar. 2008, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.30890.

[91] A. Cea and R. Palacios, "JAX-Based Aeroelastic Simulation Engine for Differentiable Aircraft Dynamics," *Computer Physics Communications*, vol. 311, p. 109 547, Jun. 2025, ISSN: 00104655. DOI: 10.1016/j.cpc.2025.109547.

[92] A. C. Gray and J. R. Martins, "Geometrically Nonlinear High-fidelity Aerostructural Optimization for Highly Flexible Wings," presented at the AIAA SciTech Forum, Virtual Event: American Institute of Aeronautics and Astronautics, Jan. 11, 2021, ISBN: 978-1-62410-609-5. DOI: 10.2514/6.2021-0283.

[93] *MSC Nastran Aeroelastic Analysis User's Guide*, Hexagon, 2023.

[94] *Simcenter Nastran Aeroelastic Analysis User's Guide*, Siemens, 2020.

[95] *MSC FlightLoads User's Guide*, Hexagon, 2021.

[96] M. Drela, "Integrated Simulation Model for Preliminary Aerodynamic, Structural, and Control-Law Design of Aircraft," presented at the 40th Structures, Structural Dynamics, and Materials Conference and Exhibit, St. Louis, Missouri: American Institute of Aeronautics and Astronautics, Apr. 12, 1999. DOI: 10.2514/6.1999-1394.

[97] L. Avoni, M. Bronz, J.-P. Condomines, and J.-M. Moschetta, "Enhancing ASWING Flight Dynamics Simulations with Closed-Loop Control for Flexible Aircraft," presented at the AIAA AVIATION Forum and ASCEND, Las Vegas, Nevada: American Institute of Aeronautics and Astronautics, Jul. 29, 2025, ISBN: 978-1-62410-738-2. DOI: 10.2514/6.2025-3425.

[98] M. J. Patil and D. H. Hodges, "Flight Dynamics of Highly Flexible Flying Wings," *Journal of Aircraft*, vol. 43, no. 6, pp. 1790–1799, Nov. 2006, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.17640.

[99] L. Cavagna, S. Ricci, and L. Riccobene, "A Fast Tool for Structural Sizing, Aeroelastic Analysis and Optimization in Aircraft Conceptual Design," presented at the 50th AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Palm Springs, California, May 4, 2009, ISBN: 978-1-60086-975-4. DOI: 10.2514/6.2009-2571.

[100] F. L. C. Ribeiro, P. Paglione, R. G. Annes da Silva, and M. Santiago De Sousa, "AeroFlex: A Toolbox for Studying the Flight Dynamics of Highly Flexible Airplanes," presented at the VII Congresso Nacional de Engenharia Mecânica, São Luís, Maranhão, Brasil, Aug. 3, 2012.

[101] C. M. Shearer and C. E. S. Cesnik, "Nonlinear Flight Dynamics of Very Flexible Aircraft," *Journal of Aircraft*, vol. 44, no. 5, pp. 1528–1545, Sep. 2007, ISSN: 0021-8669, 1533-3868. DOI: 10.2514/1.27606.

[102] H. Timmermans and B. Prananta, "Aeroelastic Challenges in the Aircraft Design Process," presented at the READ & SCAD Conference, Warsaw, Poland, Sep. 2016.

[103] V. Portapas, A. Cooke, and M. Lone, "Modelling Framework for Flight Dynamics of Flexible Aircraft," *Aviation*, vol. 20, no. 4, pp. 173–182, Dec. 20, 2016, ISSN: 1648-7788, 1822-4180. DOI: 10.3846/16487788.2016.1264719.

[104] J. Feldwisch and M. Bauer, "UltraFLoads: A Simulation Suite and Framework for High-Fidelity Flight Loads," *Aerospace*, vol. 10, no. 3, p. 273, Mar. 10, 2023, ISSN: 2226-4310. DOI: 10.3390/aerospace10030273.

[105] J. Bradbury, R. Frostig, P. Hawkins, *et al.*, *JAX: Composable transformations of Python+NumPy programs*, 2018.

[106] T. R. Brooks, G. K. W. Kenway, and J. R. R. A. Martins, "Benchmark Aerostructural Models for the Study of Transonic Aircraft Wings," *AIAA Journal*, vol. 56, no. 7, pp. 2840–2855, Jul. 2018, ISSN: 0001-1452, 1533-385X. DOI: 10.2514/1.J056603.

[107] J. A. Grauer and M. J. Boucher, "Output Measurement Equations for Flexible Aircraft Flight Dynamics," NASA Langley Research Center, NASA/TM–2018–220102, Oct. 2018.

[108] S. Doyle, *PyNastran: A Python-Based Interface Tool for NASTRAN File Formats*, version 1.4.1.

[109] "Airport Planning Manual E-Jets E2," Embraer, 2024.

[110] "Review of Standard Passenger Weights," European Aviation Safety Agency, 2022.