TÉCNICO LISBOA

Flight dynamics and simulation of a generic aircraft for aeroservoelastic design

Pedro Tomás Marques Martins Margarido

Thesis to obtain the Master of Science Degree in

Aerospace Engineering

Supervisor: Prof. André Calado Marta

Examination Committee

Chairperson: Prof. João Manuel Lage de Miranda Lemos
Supervisor: Prof. André Calado Marta
Member of the Committee: Prof. Agostinho Rui Alves da Fonseca

November 2016

”The dawn mist glowing,
The water flowing,
The endless river,
Forever and ever”
David Gilmour

Acknowledgments

The process of writing a dissertation is exhaustive and arduous - and it is certainly not done single-handedly.

I want to thank my supervisor, Professor André Calado Marta. Thank you Professor, for all your help, support and specially enthusiasm. I believe that I am tremendously fortunate to have worked with an outstanding scholar and excellent person.

I would like to thank my parents, Maria and Fernando that helped me become everyday a better version of myself, and to their sacrifice and dedication over my life, I dedicate them my biggest achievement, the Master's. As the oldest brother, I am also deeply thankful to my brothers, Luis and Inês, whose presence enlightened me and made me more responsible throughout all these years.

I am deeply grateful to my maternal grandparents, Angela and João, who cared for me and played an important role in the development of my identity and shaping the individual that I am today. I was also extremely fortunate in my life to have paternal grandparents who have shown me unconditional love and kindness through their life, and I am sorry that they are not present to see me graduate.

I must thank my girlfriend Sara, who gave me strength in the days when I was not in mood of writing or researching. Through her and her smile I found an easier path for concluding my Master's, and for that I am entirely grateful.

Finally, I want to thank all the friends I made throughout this five year journey. Their help and presence inspired me to reach this goal and without them it would have been certainly been tougher.

Resumo

Com o mercado emergente do sector aeronáutico, começa a aparecer a necessidade de ferramentas que estudem o comportamento aeroservoelastico de uma aeronave. Existem modelos matemáticos para este estudo, porém são complicados e muitos deles usam o domínio da frequência.

Nesta dissertação foram desenvolvidas as equações de movimento aeroservoelasticas de uma aeronave geral para condições de equilíbrio no domínio do tempo. Foi desenvolvido um programa produzido em C++ que integra estas mesmas equações, e que no futuro poderá ser incluído noutros projectos como uma peça de conexão entre diversas áreas da aeronáutica, como aerodinâmica, dinâmicas estruturais e controlo de voo. Para desenvolver esta ferramenta vários métodos de integração foram inspecionados e consequentemente encontrando a utilidade de cada um. Consequências da aeroelasticidade foram também discutidas e utilizadas para introduzir o controlo óptimo.

Foi também realizado um simulador de voo em MATLAB® utilizando controlo óptimo. O comportamento do controlo óptimo, mais consequentemente o regulador quadrático linear, nas dinâmicas de voo foi também estudado. Este simulador de voo permite simular o movimento de uma aeronave geral, adoptando um conjunto de derivadas aerodinâmicas de aeronaves na literatura, em regimes turbulentos e em casos de falha de motor em aeronaves com até cinco motores. O estudo das simulações feito nesta dissertação teve mais em conta que a aeronave consiga garantir equilíbrio e rumo em situações críticas, como as acima referidas.

Palavras-chave: dinâmica de voo, controlo óptimo, aeroservoelasticidade, simulação de voo, integração

Abstract

The emerging market of the aviation sector begins to request the need for tools to study the aeroservoelastic behaviour of an aircraft. There are mathematical models for this kind of study, but its interpretation is not easy and many use the frequency domain.

In this thesis the aeroservoelastic equations of motion of a general aircraft for equilibrium conditions in time domain were developed. A program was also developed, produced in C++®, which integrates these same equations, and that in the future may be included in other projects as an interconnection tool between different fields of aeronautics, such as aerodynamics, structural dynamics and flight control. To develop this tool, various integration methods were inspected and consequently the utility of each one was found. Aeroelasticity consequences were also discussed and used to introduce the optimal control.

It was also carried out a flight simulator in MATLAB® using optimal control. The optimal control behaviour, more specifically the linear quadratic regulator, in the flight dynamics was also studied. This flight simulator allows the simulation of the motion for a general aircraft, adopting a set of aerodynamics derivatives of general aircraft from the literature, on turbulent air flows and in engine failure cases in aircraft up to five engines. The simulation study in this thesis had more in mind to ensure that the aircraft maintains its equilibrium and course in critical situations, as referred above.

Keywords: flight dynamics, optimal control, aeroservoelasticity, flight simulation, integration

Contents

Acknowledgments v
Resumo vii
Abstract ix
List of Tables xv
List of Figures xvii
Nomenclature xix
Glossary xxv

1 Introduction **1**
1.1 Motivation 1
1.2 Topic Overview 2
1.3 Objectives 3
1.4 Thesis Outline 4

2 Theoretical Background **5**
2.1 Flight Simulators 5
2.1.1 Types of Simulators 6
2.1.2 Structure of a Simulator 6
2.2 Aeroelasticity 7
2.2.1 Static Aeroelasticity 7
2.2.2 Dynamic Aeroelasticity 8
2.3 Aeroservoelasticity 9
2.3.1 Flutter Control System 10
2.4 Mathematical Modeling Techniques 12
2.4.1 P-Transform Technique 12
2.4.2 FAMUSS 12
2.4.3 Rational Function Approximation 13

3 Dynamics Model **15**
3.1 Reference Frames and Angles 15
3.1.1 Euler Angles 17
3.1.2 Aerodynamic Angles 17

3.1.3 Angular Velocities . . . 18
3.2 Rigid Body Flight Dynamics . . . 18
3.2.1 Equations of Motion . . . 18
3.2.2 Control Surfaces . . . 20
3.2.3 Applied Forces and Moments . . . 20
3.2.4 Small Disturbance Theory . . . 22
3.2.5 Steady States . . . 22
3.2.6 Decoupled Equations . . . 24
3.2.7 Engine Contribution . . . 26
3.3 Elastic Aircraft Consideration . . . 28
3.3.1 Dynamics of a Flexible Aircraft . . . 28
3.3.2 Mathematical Representation of the Dynamics of a Flexible Aircraft . . . 29

4 Flight Dynamics Model Implementation **31**
4.1 Flight Dynamics Model Integrator . . . 31
4.1.1 Inputs and Outputs . . . 32
4.1.2 Dynamic Equations . . . 33
4.1.3 Integration . . . 33
4.1.4 Type of Steppers . . . 34
4.2 Stepper Comparison . . . 36
4.3 Aircraft Dynamics . . . 37

5 Flight Control and Simulation domain **41**
5.1 Simulation Domain . . . 41
5.2 Sensors and Actuators . . . 41
5.3 State-space Systems . . . 42
5.3.1 Longitudinal State-space Equations . . . 43
5.3.2 Lateral State-space Equations . . . 43
5.4 Variable Control . . . 43
5.4.1 Coupled Motion . . . 44
5.4.2 Longitudinal mode . . . 44
5.4.3 Lateral Mode . . . 45
5.4.4 Heading . . . 46
5.4.5 Flying and Handling Qualities . . . 46
5.5 Atmospheric Perturbations . . . 48
5.5.1 Influence on the Model . . . 49
5.5.2 SIMULINK Wind Turbulence Model . . . 50
5.5.3 Disturbances State-space Form . . . 50
5.6 SIMULINK State-Space model . . . 52

6 Optimal Control and Computational Implementation **53**
6.1 Optimal Control Technique - Linear Quadratic Regulator . . . 53
6.1.1 Aeroservoelastic Optimal Control . . . 53
6.1.2 Open Loop Aeroservoelastic Problem . . . 55
6.1.3 Linear Quadratic Regulator . . . 56
6.1.4 Closed Loop Aeroservoelastic Problem . . . 57
6.1.5 Linear Quadratic Regulator Concerns and Conclusions . . . 59
6.2 Flight Control Coupled Model . . . 60
6.2.1 Schematic of the Flight Controller Model . . . 60
6.2.2 SIMULINK® Flight Controller Model . . . 62
6.3 Linear Quadratic Regulator Script . . . 62
6.3.1 Flying Qualities Evaluator . . . 63
6.3.2 Applicable Bryson's method . . . 63

7 Flight Simulation **67**
7.1 Open-Loop Dynamics . . . 67
7.2 Engine Failure . . . 68
7.3 Turbulence . . . 72

8 Conclusions **77**
8.1 Achievements . . . 77
8.2 Future Work . . . 78

Bibliography **79**

A Reference Frames, Stability Derivatives, Flying Qualities and Trim condition Aircraft data **81**
A.1 Rotation Matrices . . . 81
A.1.1 East North Up Reference Frame . . . 82
A.2 Stability Derivatives . . . 83
A.3 Flying and Handling Qualities . . . 83
A.3.1 Aircraft Classes . . . 83
A.3.2 Flight Phases . . . 84
A.3.3 Aircraft Data . . . 85

B Aircraft Simulation Data **87**
B.1 LQR Q and R Matrices . . . 87
B.2 LQR Gain Matrix . . . 89

List of Tables

4.1 Inputs for the flight dynamics integrator scheme 32
4.2 Outputs of the flight dynamics integrator scheme 33
4.3 Boost© and SIMULINK® types of steppers 35

5.1 Aircraft level of perfomance 48

7.1 Open-loop dynamic modes eigenvalues for Airbus A400M and Dassault Falcon 7X flight conditions 67
7.2 A400M - closed loop poles for ϵ = 10, ϵ = 40 and ϵ = 80 68
7.3 Falcon 7X - closed loop poles for ϵ = 2, ϵ = 20 and ϵ = 200 72

A.1 Aircraft classes 83
A.2 Flight phases 84
A.3 Phugoid flying qualities 84
A.4 Short period flying qualities 84
A.5 Spiral flying qualities 84
A.6 Rolling flying qualities 85
A.7 Dutch roll flying qualities 85
A.8 Aircraft general parameters 86
A.9 Aircraft trim derivatives 86

List of Figures

1.1 Aeroelasticity 1
1.2 Aeroservoelasticity model 2
1.3 Dynamics model purpose 3

2.1 VMS NASA simulator 5
2.2 Flight simulator structure 7
2.3 Divergence graphic 8
2.4 Flutter graphic 8
2.5 Tacoma bridge flutter 8
2.6 Aeroservoelastic interactions 9
2.7 Aeroservoelasticity block diagram 10
2.8 Binary flutter system with a control surface 11

3.1 Fixed and relative reference frame 15
3.2 Euler reference frame rotation 17
3.3 Control surfaces deflection conventions 20
3.4 Applied forces on a general aircraft 20

4.1 Dynamic response of Dassault Falcon 7X flight condition 36
4.2 Embraer E120 dual flight condition trajectory in the ENU reference frame 38
4.3 Embraer E120 dual flight condition trajectory, XY and YZ planes in the ENU reference frame 38
4.4 Dynamic response of Embraer E120 40

5.1 Control surfaces actuators model implemented in SIMULINK® 42
5.2 SIMULINK wind turbulence model 50
5.3 State-Space in SIMULINK 52

6.1 Airfoil model used in the aeroservoelastic simulation 54
6.2 Below flutter velocity $V = 289.6$ m/s on open-loop 55
6.3 Flutter velocity $V = V_{flutter} = 297.4$ m/s on open-loop 55
6.4 Over flutter velocity $V = 304.8$ m/s on open-loop 56
6.5 Flutter velocity $V = V_{flutter} = 297.4$ m/s on closed-loop with LQR 58
6.6 Over flutter velocity $V = 304.8$ m/s on closed-loop with LQR 58

6.7 Schematic of the control law 61
6.8 Flight controller model in SIMULINK 62
6.9 Schematic used for applying Bryson's method 64

7.1 A400M's dynamic responses of the control input variables (δ_E,δ_A,δ_R) for three different control penalty parameters (ϵ = 10, ϵ = 40 and ϵ = 80) 69
7.2 A400M's dynamic responses of the non-malfunctioning engines (δ_{T_3},δ_{T_4}) for three different control penalty parameters (ϵ = 10, ϵ = 40 and ϵ = 80) 70
7.3 A400M's dynamic responses of the controllable states (u,γ,λ) for three different control penalty parameters (ϵ = 10, ϵ = 40 and ϵ = 80) 71
7.4 A400M's flight trajectory seen in XY_{ENU} and YZ_{ENU} planes for three different control penalty parameters (ϵ = 10, ϵ = 40 and ϵ = 80) 71
7.5 Falcon 7X's dynamic responses of the controllable states (u,γ,λ) for three different control penalty parameters (ϵ = 2, ϵ = 20 and ϵ = 200) 73
7.6 Falcon 7X's dynamic responses of the input variables (δ_E,δ_A,δ_R) for three different control penalty parameters (ϵ = 2, ϵ = 20 and ϵ = 200) 74
7.7 Falcon 7X's dynamic responses of the engines (δ_{T_1},δ_{T_2},δ_{T_3}) for three different control penalty parameters (ϵ = 2, ϵ = 20 and ϵ = 200) 75
7.8 Falcon 7X's Turbulent flight trajectory seen in XY_{ENU} and YZ_{ENU} planes for three different control penalty parameters (ϵ = 2, ϵ = 20 and ϵ = 200) 75

A.1 Rotation matrix axis system 81

Nomenclature

Greek symbols

α	Angle of attack
β	Angle of side-slip
χ_i	Velocity of the asymmetrical ith vibration mode
Δt	Time interval
δ_A	Aileron deflection
δ_E	Elevator deflection
δ_R	Rudder deflection
δ_{T_i}	Engine throttle relative to the ith engine
ϵ	Control penalty parameter
η_{i_j}	Structural derivative of the ith symmetrical vibration mode in respect to j
γ	Flight path angle
κ	Eigenvalue
λ	Heading angle
λ_i	Displacement of symmetrical ith vibration mode
$\mathbf{\Omega}$	Angular speed of the fixed reference frame relatively to the body reference frame
μ_{i_j}	Structural derivative of the ith asymmetrical vibration mode in respect to j
ν	Plunge angle
ω	Natural frequency
ω_n	Undamped natural frequency
ϕ	Roll angle
π	Pi constant

ψ Yaw angle

ρ Density

σ_i Velocity of the symmetrical ith vibration mode

τ_i Displacement of asymmetrical ith vibration mode

θ Pitch angle

ξ Damping ratio

Roman symbols

$\tilde{M}_i$ i indirectly induced moment derivative on the y-axis

a_{i_j} Acceleration on the i-axis in the j reference frame

c_g Center of gravity

C_{D_i} Drag coefficient derivative with respect to i

c_{ji} Coefficient j (j = 1, j = 2 and j = 3) of the ith generalized coordinate

C_{L_i} Lift coefficient derivative with respect to i

C_{l_i} Rolling moment coefficient derivative with respect to i

C_{m_i} Pitching moment coefficient derivative with respect to i

C_{n_i} Yawing moment coefficient derivative with respect to i

C_{Y_i} Side slip moment coefficient derivative with respect to i

F_B Body reference frame

F_E Fixed reference frame

F_{a_i} Aerodynamic forces on the reference frame i

F_{g_i} Weight force on the reference frame i

F_{p_i} Propulsive forces on the reference frame i

I_θ Wing section static moment about elastic axis

I_{ij} Moment of inertia on the ij axis

K_δ Linear spring constant for torsion stiffness

K_ν Linear spring constant for wing bending

L'_i i asymmetrical induced moment derivative on the x-axis

L_δ Control induced lift

L_θ Induced aerodynamic lift forces

L_i i symmetrical induced moment derivative on the x-axis

L_{i_j} Rotation matrix from reference frame j to reference frame i

M_δ Control induced moment

M_θ Motion-induced moment

M_i i directly induced moment derivative on the y-axis

N_i' i asymmetrical induced moment derivative on the z-axis

N_i i symmetrical induced moment derivative on the z-axis

N_{eng} Number of engines

N_{steps} Number of integration steps

p_{dF} Flutter dynamic pressure

q_i Generalized coordinate

S_θ Wing static moment

u Longitudinal velocity

u_i Longitudinal velocity in the reference frame i

v Lateral velocity

v_i Lateral velocity in the reference frame i

$V_{Flutter}$ Flutter Velocity

w Vertical velocity

w_i Vertical velocity in the reference frame i

X_i i induced force derivative on the x-axis

x_i x position in the reference frame i

Y_i i induced force derivative on the y-axis

y_i y position in the reference frame i

Z_i i induced force derivative on the z-axis

z_i z position in the reference frame i

$\boldsymbol{A_i}$ Aerodynamic force matrix coefficient with respect to i

$\boldsymbol{A_{Lat}}$ Lateral state coefficient matrix

$\boldsymbol{A_{Long_{\lambda_s}}}$ Lateral state coefficient matrix including λ and integrator states

$\boldsymbol{A_{Long}}$ Longitudinal state coefficient matrix

$\boldsymbol{A_{LQR}}$ LQR state coefficient matrix

$\boldsymbol{A_{Long_{\gamma}}}$ Longitudinal state coefficient matrix including γ state

$\boldsymbol{A_{Long_{\lambda}}}$ Lateral state coefficient matrix including λ state

$\boldsymbol{A_{Long_{u\gamma_s}}}$ Longitudinal state coefficient matrix including γ and integrator states

$\boldsymbol{B_{coupled}}$ Engine coupled driving matrix with the updated states

$\boldsymbol{B_{ini}}$ Initial engine coupled driving matrix

$\boldsymbol{B_{Lat}}$ Lateral driving matrix

$\boldsymbol{B_{Long_{\lambda_s}}}$ Lateral driving matrix including λ and integrator states

$\boldsymbol{B_{Long}}$ Longitudinal driving matrix

$\boldsymbol{B_{LQR}}$ LQR driving matrix

$\boldsymbol{B_{Long_{\gamma}}}$ Longitudinal driving matrix including γ state

$\boldsymbol{B_{Long_{\lambda}}}$ Lateral driving matrix including λ state

$\boldsymbol{B_{Long_{u\gamma_s}}}$ Longitudinal driving matrix including γ and integrator states

$\boldsymbol{B}$ Damping matrix

$\boldsymbol{d_{coupled}}$ Disturbance states vector

$\boldsymbol{e_i}$ Unit vector in the reference frame i

$\boldsymbol{E_{coupled}}$ Associated disturbance influence matrix

$\boldsymbol{F}$ Generalized force; resultant of all external forces

$\boldsymbol{H}$ Total moment

$\boldsymbol{I}$ Inertia tensor matrix; identity matrix

$\boldsymbol{K_s}$ Stiffness matrix

$\boldsymbol{K_i}$ Gain matrix associated with the ith variable

$\boldsymbol{K_{LQR}}$ Gain matrix associated with the LQR method

$\boldsymbol{M}$ Mass matrix; resultant external moment

$\boldsymbol{p_i}$ Position vector in the reference frame i

$\boldsymbol{Q}$ State weighting matrix

$\boldsymbol{q}$ Motion vector

$\boldsymbol{R}$ Control cost matrix

$\boldsymbol{u_{coupled}}$ Longitudinal and Lateral coupled input vector

$\boldsymbol{u_{Lat}}$ Lateral input vector

$\boldsymbol{u_{Long}}$ Longitudinal input vector

$\boldsymbol{v}$ True airspeed

$\boldsymbol{w}$ Angular velocity vector

$\boldsymbol{x_{Lat}}$ Lateral state vector

$\boldsymbol{x_{Long_{\lambda_s}}}$ Lateral state vector including λ and integrator states

$\boldsymbol{x_{Long}}$ Longitudinal state vector

$\boldsymbol{x_{LQR}}$ LQR state vector

$\boldsymbol{x_{Long_{\gamma}}}$ Longitudinal state vector including γ state

$\boldsymbol{x_{Long_{\lambda}}}$ Lateral state vector including λ state

$\boldsymbol{x_{Long_{u\gamma_s}}}$ Longitudinal state vector including γ and integrator states

$\boldsymbol{x}$ Generalized deflection; state vector

A Generalized aerodynamic influence coefficient

b Wing span

c Cord

g Gravitational constant

h Height

L Total moment on the x-axis

M Total moment on the x-axis

M Wing section mass

m Mass

N Total moment on the x-axis

p Roll rate

q Pitch rate

r Yaw rate

S Wing area

V Flight velocity

X Total force on the X-axis

Y Total force on the x-axis

Z Total force on the x-axis

Subscripts

x, y, z Cartesian components.

0 Equilibrium state

a Aerodynamic force

B Body reference frame

dr Dutch roll dynamic mode

E Fixed reference frame

g Gust

i Dummy index

max Maximum

p Propulsive force

phu Phugoid dynamic mode

ref Reference

roll Roll dynamic mode

sp Short period dynamic mode

spi Spiral dynamic mode

W Wind

Superscripts

* Augmented

T Transpose

Glossary

AFC	Automatic Flight Controller
AIC	Aerodynamic influence coefficient
ASE	Aeroservoelasticity
CM	Center of mass
DCM	Direct cosine matrix
ECEF	Earth-Centered Earth-Fixed
ENU	East-North-Up
EOM	Equations of motion
FAMUSS	Flexible Aircraft Modeling Using State Space
FCS	Flight Control System
LLH	Latitude-Longitude-Height
LQR	Linear Quadratic Regulator
NED	North-East-Down
ODE	Ordinary differential equations
RFA	Rational Function Approximation
RPM	Rotor rotation speed
RPY	Roll-Pitch-Yaw
STL	Standard Template Library
TAS	True airspeed

Chapter 1

Introduction

1.1 Motivation

With the increasing growth of high-performance and cheap aircraft, the need for more realistic flight simulators also grows. One of the crucial aspects on making the flight simulator more realistic is the consideration of aircraft's elastic properties (aeroelasticity).

Aeroelasticity has been defined as a science which studies mutual interactions between aerodynamic forces and elastic forces, and the influence of these interactions on airplane design. Some of the most rough phenomena on aircraft's structure happen because of the aircraft's elastic properties. These physical phenomena, as they will be described later, can be, for example, flutter, control reversal and others.[1].

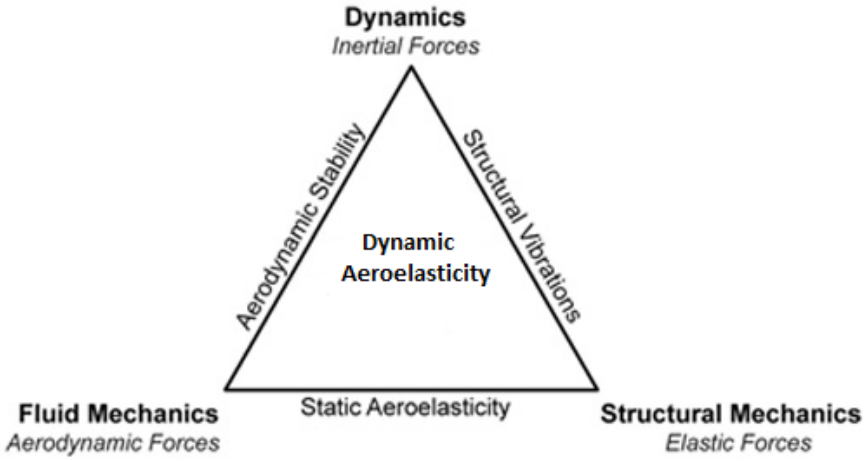

Figure 1.1: Aeroelasticity (adapted from [2])

Traditionally, aeroelasticians utilize frequency domain to model the aeroelastic aircraft, which brings several concerns since the modern control theory is based primarily on the state-space approach, in which it requires the aircraft to be modeled in the time domain. The coordinate system on which each

scientific area works on, also changes. The flight control engineer usually works in a body axis coordinate system while the aeroelastician works in a mean axis system [3].

This dissertation is about reaching a consensus around these two problems by formulating a mathematical numerical model and then implementing a flight dynamics integrator of the equations of motion, for a future use in an aeroservoelastic model of a generic aircraft. A flight controller with the possibility of being embedded in the aeroservoelastic model in the future, was also developed.

1.2 Topic Overview

The focus of this dissertation is firstly the formulation of the equations of motion of a generic elastic aircraft, in which several areas like Flight dynamics, Aerodynamics, Aeroelasticity will be reviewed. Afterwards in order to implement the control laws on the model, Optimal Control laws will also be studied concerning the implemented law, LQR (Linear Quadratic Regulator). On the basis of this model an automatic pilot was created, with the possibility of controlling the longitudinal velocity, the flight path angle and the heading angle.

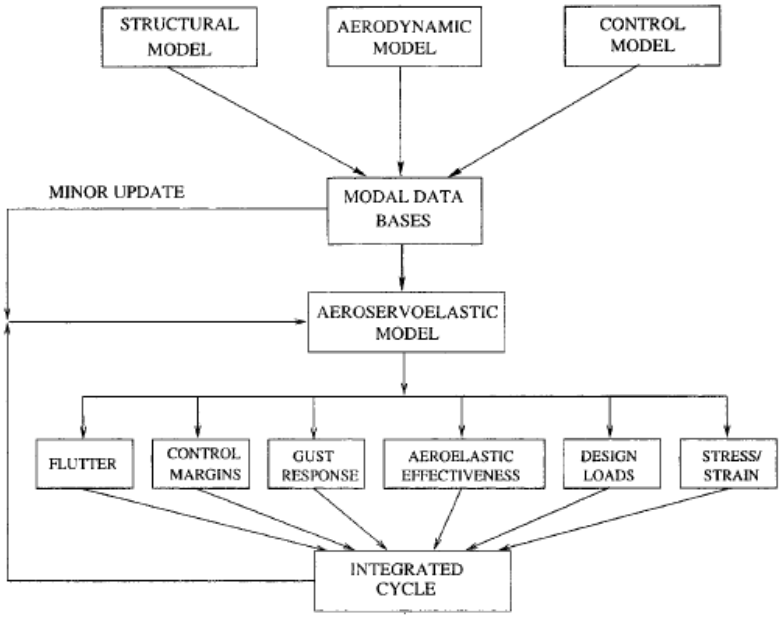

Figure 1.2: Aeroservoelasticity Model (altered from [4])

The goal is developing an aeroservoelastic model, like in Figure 1.2, for a generic aircraft. Initially the model, in order to start the simulation, will use a certain aircraft's aerodynamic and structural model to form a modal database. The integrated cycle of the model will take in consideration several effects, like the ones seen in the Figure 1.2, applying a minor update in the aeroservoelastic model. In this project, only control and gust effects will be taken in consideration since the intention of this dissertation is to reach the elastic aircraft decoupled equations and create a flight controller and a flight dynamics integrator. All the simulation of the real-time aeroelastic effects, to be integrated to the flight controller, could be a possible dissertation work for future students.

1.3 Objectives

The main objective of this thesis is the mathematical formulation and respective numerical implementation of the longitudinal and lateral equations of motion (EOM) of an aircraft, to be embedded in a aeroservoelasticity analysis and to become an optimization tool in the future. To this end, principles of flight dynamics, flight control and aeroelasticity will be used to model the system.

Once the system is defined, it should allow the modeling of the aircraft motion in time domain for simulation purposes. The flight simulator will be done in a MATLAB® environment and the flight dynamics integrator will be written in C/C++® for computational resource optimization and to facilitate integration with other tools.

For the flight simulation tool, the control of flight path angle, the heading angle and the longitudinal velocity, will be applied utilizing the LQR optimal control technique. This flight simulation tool will be of a generic aircraft, adopting a set of aerodynamics derivatives of aircraft of interest from the literature.

As seen in Figure 1.3, the goal of the flight dynamics integration tool is to guarantee future integration with aerodynamics, structures, flight control and engine model programs to simulate a generic aircraft dynamics during a time interval (Δt). The grey highlighted boxes are the programs done and described throughout this thesis.

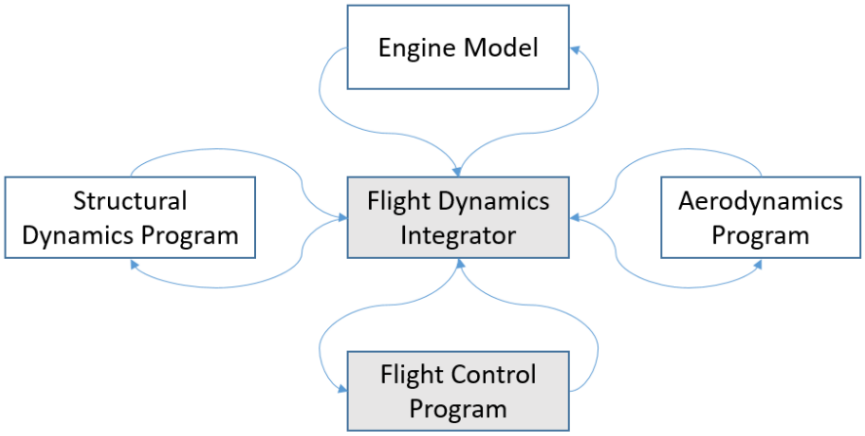

Figure 1.3: Purpose behind the flight dynamics integrator

To create these modules, several steps were carried out, namely:

- formulation mathematical of the equations of motion for an elastic aircraft;
- implementation of the flight dynamics model on C/C++;
- interpretation of the several possible integration types;
- implementation of control laws on the mathematical model;
- implementation of the numerical model on MATLAB® and SIMULINK;
- flight simulation of a generic aircraft on several environments;

1.4 Thesis Outline

This thesis is structured in eight Chapters.

The first Chapter introduces the study that will be developed and it has the purpose of contextualizing. Beyond that, objectives are announced.

The second Chapter contains theoretical background and in it, flight simulators structure and its different types and functionalities are presented. Then the following concepts of aeroelasticity, including its two types and different phenomena, and aeroservoelasticity, the capacity of using control inputs to prevent this phenomena, are introduced. Finally to end the Chapter, different types of mathematical modelling techniques for aeroservoelastic models are discussed.

The flight dynamics model used is demonstrated on the third Chapter. Firstly, the reference frames are described and explained, alongside with the Euler angles, aerodynamic angles and angular velocities. Secondly, rigid body equations of motion are deduced, and by utilizing the small disturbance theory, two set of equations characterizing the aircraft motion emerge. Single engine contribution is specified for these equations, generating a set of equations coupled by engine thrust. To conclude the Chapter, the final trim condition aeroservoelastic set of equations are defined by adding structural bending modes.

The fourth Chapter contains the flight dynamics model implementation, starting by the description of the process that formulates a C++ integrator program that is capable of integrating the final equations of Chapter three. The type of integration, what integration stepper to use, outputs and inputs are discussed throughout this Chapter.

The fifth one is relative to flight control and the simulation domain utilized. The state-space form is introduced as the domain used for the flight controller. So, the set of equations of motion of Chapter three is now defined into state-space form. Then, the desired controllable states are introduced into the state-spaces. Actuators and sensors of the flight controller are discussed here. Flying and handling qualities, as an important concept of defining realistic flight simulations, are here announced. The atmospheric perturbations influence on the model is brought in by the end of the Chapter.

The sixth Chapter refers to optimal control and computational implementation of the problem in question. The optimal control technique - linear quadratic regulator - is here explained by adopting a classic aeroelastic foil. Then, it is adopted to this thesis flight controller by being applied to a general aircraft using the Bryson's method. In the end of this Chapter, the final flight simulator SIMULINK model and the script that is used to guarantee level one flight qualities are shown and deliberated upon.

The seventh Chapter yields the flight simulation results. This Chapter is devoted to discussing the results of two simulations in two different scenarios, engine failure and turbulent flight, for two completely different aircraft, Airbus A400M and Dassault Falcon 7X. The linear quadratic regulator influence on these simulations is the prime discussion factor of this Chapter.

The eighth Chapter is where the final conclusions are drawn and future work to be implemented is defined.

Chapter 2

Theoretical Background

In this Chapter, several topics will be studied, such as: Flight simulation, in order to know how this dissertation project can be embedded in a flight simulator; Aeroelasticity to learn the aeroelastic effects that need to be prevented; Aeroservoelasticity to control those aeroelastic phenomena; Mathematical and numerical methods to simulate unsteady aerodynamics and structural dynamics.

2.1 Flight Simulators

Flight simulation is basically a way to recreate the conditions of a real flight. Several aeronautical areas such as flight dynamics, navigation and aeroelasticity behavior can be studied in an artificial computational environment.

Figure 2.1: VMS, simulator build by NASA in Ames, California USA

From actually large built simulators, as seen in Figure 2.1, to flight simulator games, flight simulation has found its use for both professional and casual purposes. One of the most useful traits of flight simulation is to enable military or civil pilots to train. Being in a simulated environment allows the training of life-treating maneuvers, practicing of complex missions and, of course, the enhancement piloting skills [5]. The benefits of flight simulation are:

1. **Safety**: As said before, flight simulation allows pilots to face dangerous situations that may happen

in a real flight. It also allows inexperienced pilots to gather motor skills and the basic procedures of flight. Simulators are also used as testing devices for new designed aircraft, therefore they may allow foreseeing design failures.

2. **Cost**: Though building a simulator may be expensive, it is very affordable when compared to building a new aircraft. For flight crew training, fuel is preserved when using flight simulators. Finally all of the safety traits are cost beneficial, allowing companies to save millions on potential aircraft flaws or human inexperience errors.

2.1.1 Types of Simulators

Types of flight simulators vary according to their purpose. Several application areas can be recognized as [6]:

- **Engineering Simulators**: Used generally to test an aircraft characteristics. When a new aircraft is being developed, simulators are utilized in the design phase to detect possible design flaws and they also allow a much smoother transition to real flight. One of the crucial characteristics being tested in this type of simulators is the aircraft response to aeroelastic effects, as they might be catastrophic if not corrected in the design phase.

- **Research Simulators**: These kind of simulators are more used to test the human/aircraft interaction and the investigation of human perspective. As the name suggests, it is also used for breakthrough research in several areas as navigation, aeroservoelasticity, flight control, among others. One of the most important uses of this kind of simulators is accident investigation.

- **Training Simulators**: Used for pilot training in either dexterity (manual control) or procedures (flight management). They also permit pilots to transition from one airplane type to another or to have their skills evaluated. Zero flight-time training is begining to emerge in aeronautics and, in the future, companies plan to use solely simulators for the training of their newly-recruited pilots. When completed, these pilots will have the skills to fly a transporting aircraft without real flight-time.

2.1.2 Structure of a Simulator

As seen in Figure 2.2, a flight simulator is composed by several modules. The crucial module of a simulator is the dynamics module and in a general way, all the other modules are inputs or outputs of this major module. The inputs of the central module may be a weather model, engine model, an aerodynamic model, a gear model, among others. The outputs are basically those which interact with the user, as for example: visual system, sound system, motion system, instrument displays and navigation systems.

This dissertation has the objective of creating this significant module, containing the structural dynamics of the aircraft. Then it may be used, when paired with a flight controller, to control the harmful aeroelastic effects that may occur (aeroservoelasticity). Therefore it may be used in the future as a module in a flight simulator.

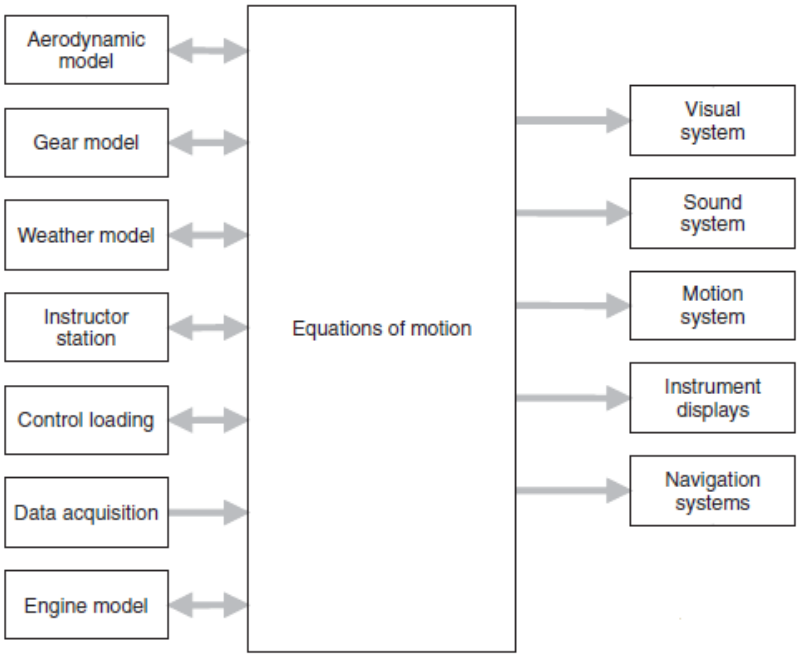

Figure 2.2: General structure from a flight simulator (adapted from [5])

2.2 Aeroelasticity

Aeroelasticity is the mutual interactions of several areas as it is seen in the classic Collar aeroelastic triangle [2], Figure 1.1. Stability and control; structural dynamics and static aeroelasticity - each one of these major disciplines are a product from two of three types of force. When all the three types of force are interacting, dynamic aeroelastic phenomena occur. [7]

Harmful aeroelastic phenomena grow when structure deformation causes additional aerodynamic forces. Eventually, these additional forces may produce more structural deformation, resulting in even greater aerodynamic forces. These adverse phenomena usually occur when there is an interaction between the three forces (dynamic aeroelastic phenomena), and an interaction between aerodynamic and elastic forces (static aeroelastic). [1].

2.2.1 Static Aeroelasticity

Static aeroelasticity phenomena, which can lead potentially to structural failure, is the result of interactions between aerodynamic and elastic forces. Some of the most adverse phenomena of these type are:

- **Divergence**: A static instability of a lifting surface of an aircraft in flight, at a speed called the divergence speed, where the elasticity of the lifting surface plays an essential role in the instability. In Figure 2.3, it is possible to see how the deformation plays out through time.
- **Control reversal**: A condition that occurs in flight, at a speed called the control reversal speed, at which the intended effects of displacing a given component of the control system are completely

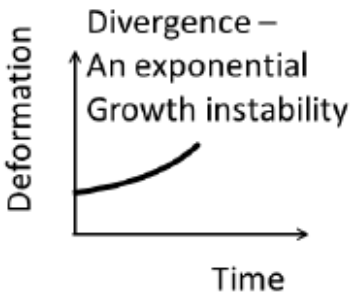

Figure 2.3: Evolution of deformation through time, Divergence (Adapted from [8])

nullified by elastic deformations of the structure.

2.2.2 Dynamic Aeroelasticity

Dynamic aeroelasticity phenomena is the result of interactions amid inertial, aerodynamic and elastic forces. Usually the difference from the static aeroelastic phenomena are the oscillatory effects of the aeroelastic interactions. Static aeroelasticity considers the non oscillatory aerodynamic forces on the flexible aircraft structure [7] . Some of the harmful dynamic phenomena are:

- **Flutter**: As seen in Figure 2.4, flutter is an aeroelastic self-excited unstable vibration in which the airstream energy is absorbed by the lifting surface. The motion involves both bending and torsional components which are basically simple harmonic oscillations with an unique flutter frequency.

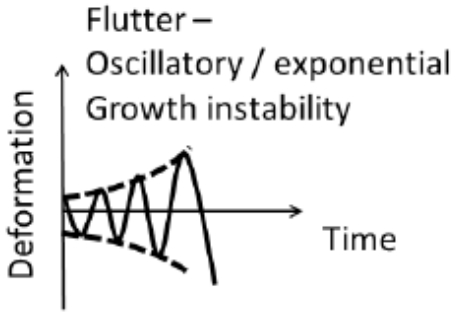

Figure 2.4: Oscillatory effects of flutter through time (adapted from [8])

Flutter effects can be catastrophic, as seen in Figure 2.5:

Figure 2.5: Flutter catastrophic phenomenon in Tacoma Narrows bridge, Washington USA

- **Buffeting**: Transient vibrations of aircraft structural components due to aerodynamic impulses produced by the wake behind wings, nacelles, fuselage pods, or other components of the airplane.
- **Dynamic Response**: Transient response of aircraft structural components produced by rapidly applied loads due to gusts, landing, gun reactions, abrupt control motions, moving shock waves, or other dynamic loads.

2.3 Aeroservoelasticity

Aeroservoelasticity (ASE), as seen in Figure 2.6, is the discipline of the aeronautical science that deals with the interaction of aircraft structural, aerodynamic, and control systems. Though there were early sucesses in creating active flutter suppression systems and load alleviation systems, ASE still remains a vast experimental area and has still not reached operational status on any aircraft [9]. This mainly happens due to the difficulty of designing a control system, which is robust enough for uncertainties in the unsteady aerodynamic model.

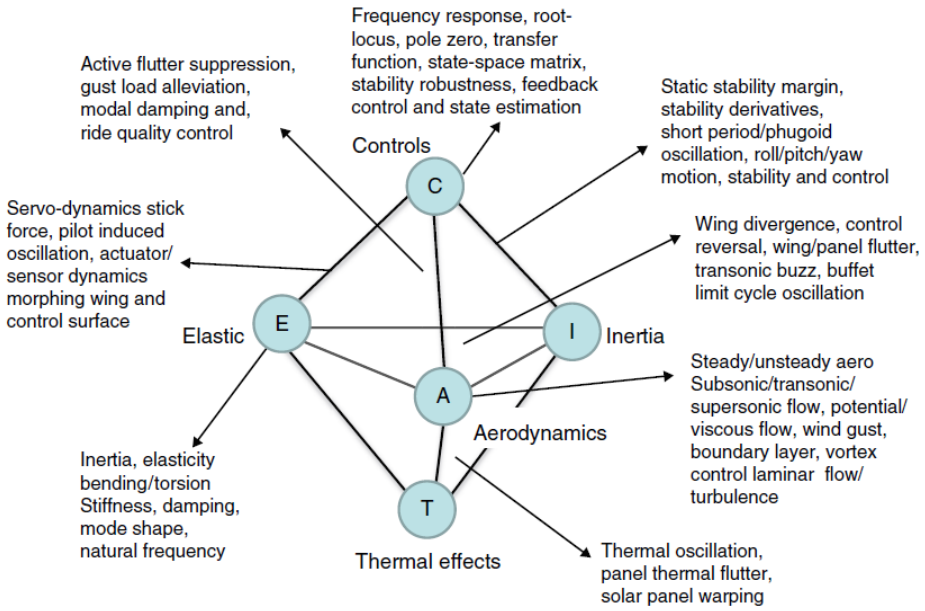

Figure 2.6: Aeroservoelastic interaction between structure, inertia, aerodynamics, control, and thermal effects and associated phenomena. [10]

A possible block diagram for the aeroservoelasticy is seen in Figure 2.7. Deformation happens or is usually increased when there are gusts (disturbance input) or control surface deflection, as seen in the aeroelasticity plant from Figure 2.7. Deformation induces changes on the aerodynamic forces acting on the aircraft, hence the aerodynamic feedback loop. Therefore this cycle needs to be controlled, or in extreme cases, it may lead to one of many catastrophic phenomena as explained Section 2.2. Knowing these deformation rates and the aeroelastic phenomena, it is possible to generate a control model to prevent these phenomena from happening. This control model needs sensors, to calculate relatively important variables and also actuators which induce the changes on the control surfaces according to

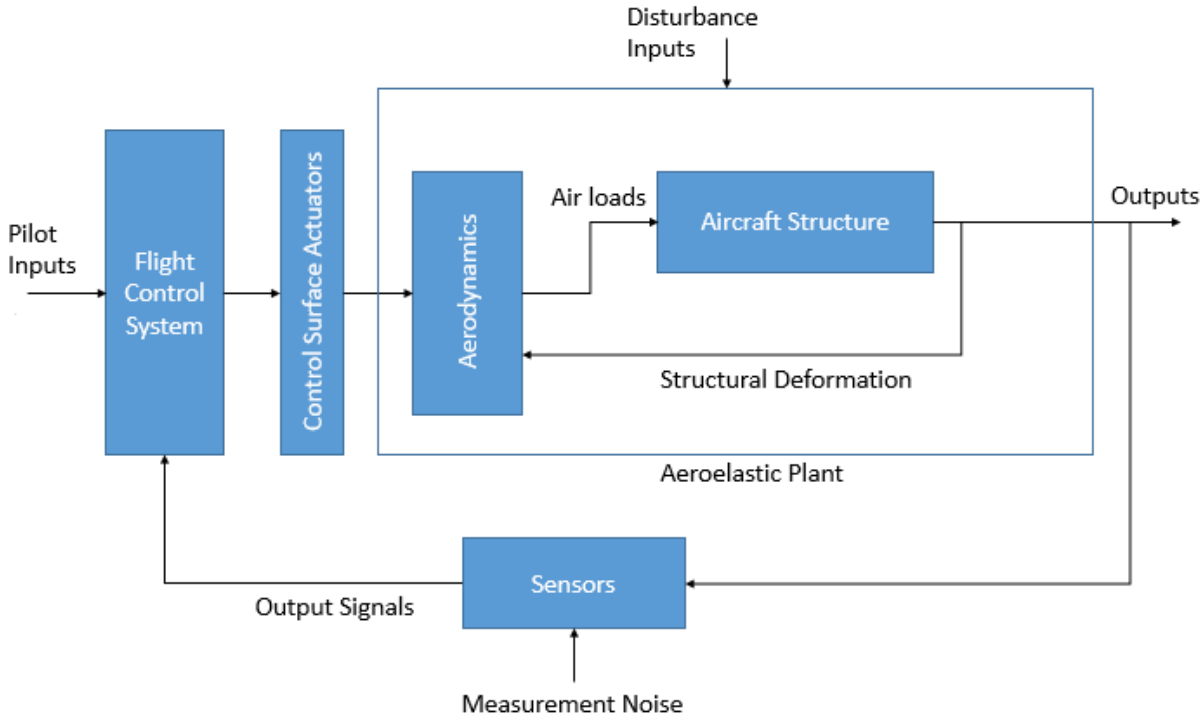

Figure 2.7: General aeroservoelastic block diagram (adapted from [9])

the control laws. The aeroelastic phenomena control are an extra function to the Flight Control System (FCS). Aeroservoelastic models such as active flutter suppression or gust load alleviation must be integrated in the FCS, as a secondary task, in order to prevent catastrophic aeroelastic events from happening and to ensure a more pleasant flight.

2.3.1 Flutter Control System

Of all the phenomena shown, flutter is by far the least desirable effect. Thus keeping this effect from happening is perhaps the most important factor in aeroservoelasticity control systems. A remarkable person who did significant work in the flutter subject was Theodore Theodorsen, a Norwegian-American theoretical physicist. He published his famous Theodorsen's function that laid the foundation for flutter analysis and control [10]. This equation determined a set of complex frequency-response functions, which has as inputs, vertical translation, angle of attack and aileron rotation angle, and as outputs, unsteady lift, pitching moment and aileron hinge moment. However, as his equation is for unsteady aerodynamic forces and its complexity overwhelms the reach of this dissertation, another form of flutter control equation will be demonstrated.

Aeroservoelastic Equation for Flutter Analysis and Control

Let us start by using a two-dimensional airfol section with a control surface embedded in a flowing fluid with velocity V.

This binary aeroelastic system, shown in Figure 2.8, is composed of a uniform rigid rectangular wing with pitch θ and plunge ν motion. It also includes a control surface, that has infinite stiffness but can be

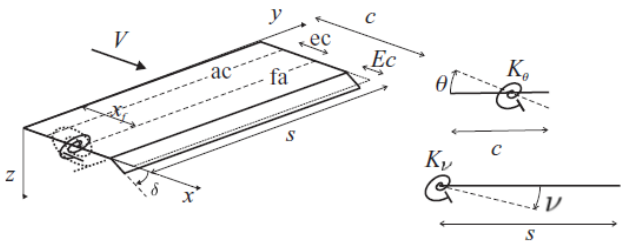

Figure 2.8: Binary flutter system with a control surface (adaptated from [7])

moved angle δ. The control surface basically only acts as a excitation device. θ and ν may also represent (nose-up positive) elastic torsion and elastic bending deflection (downward positive) of elastic axis. The linear equations of motion for small perturbations can be written as the sum of inertia and elastic forces and moments to be equal to the corresponding externally applied forces and moments [11]

$$M\frac{d^2}{dt^2}\nu + S_\theta\frac{d^2}{dt^2}\theta + K_\nu\nu = -L_\theta(t) - L_\delta(t)\,, \tag{2.1}$$

$$S_\theta\frac{d^2}{dt^2}\nu + I_\theta\frac{d^2}{dt^2}\theta + K_\theta\theta = M_\theta(t) + M_\delta(t)\,, \tag{2.2}$$

where, in equation (2.1), M is the wing section mass, S_θ is the wing static moment about elastic axis, the terms $\frac{d^2}{dt^2}\theta$ and $\frac{d^2}{dt^2}\nu$ are pitch and plunge accelerations, $L_\theta(t)$ represents time-dependent induced aerodynamic lift forces, $L_\delta(t)$ time-dependent control induced lift and finally K_ν is the linear spring constant for wing bending ($K_\nu\nu$ represents the linear elastic restoring force). For equation (2.2), I_θ is the wing section static moment about elastic axis, M_θ is motion-induced moment, M_δ control induced moment and K_θ is the linear spring constant for torsion stiffness ($K_\theta\theta$ represents the resisting moment). Using quasi-steady aerodynamics, utilizing the Lagrange's equation across the entire semi-span of the wing and considering motion in the control surface. Equations (2.1) and (2.2), condense to

$$\boldsymbol{M_s}\frac{d^2}{dt^2}\boldsymbol{q} + \rho V\boldsymbol{A_1}\frac{d}{dt}\boldsymbol{q} + (\rho V^2\boldsymbol{A_2} + \boldsymbol{K_s})\boldsymbol{q} + \rho V^2\boldsymbol{A_\delta}\delta = 0\,, \tag{2.3}$$

where $\boldsymbol{K_s}$ is the generalized stiffness matrix, the motion vector $q = [\nu \quad \theta]^T$ has two degrees of freedom, $\boldsymbol{M_s}$ is the matrix containing the terms of second order time derivatives from equations (2.1) and (2.2). $\boldsymbol{A_1}$, $\boldsymbol{A_2}$ and $\boldsymbol{A_\delta}$ are aerodynamic force matrix coefficients and its contribution depends on the air density (ρ) and the flight velocity (V). Equation (2.3) represents the fundamental aeroservoelastic open-loop equation for flutter analysis and control. If $\delta = 0$ above a certain flutter dynamic pressure (p_{dF}), the solution to these equations can become unbounded. The solution of equation (2.3) corresponds then to a non-damped harmonic solution. Basically the binary aeroelastic system undergoes large diverging oscillation that may lead to structural failure. The flutter dynamic pressure (p_{dF}) has an associated speed called flutter speed

$$p_{dF} = \frac{1}{2}\rho {V_{Flutter}}^2\,. \tag{2.4}$$

Goal of an aeroservoelastic model, is to close the loop in order to increase the velocity in which flutter

happens. Closing the loop means implementing a control law that, aided by data provided from sensors, the control actuators change the stability characteristics of the open-loop, therefore increasing the closed-loop flutter speed.

2.4 Mathematical Modeling Techniques

According to [3], there are three main classes of time domain mathematical modeling techniques which are used to model structural dynamics and unsteady aerodynamics. Integrating these models with a non-linear rigid body and static aeroelastic equations of motion, results in an universal aeroelastic simulation model to be used by both aeroelasticians and flight control engineers.

2.4.1 P-Transform Technique

This technique consists of three major steps:

1. Defining the frequency-domain equations of motion of the aircraft,

$$[\boldsymbol{M}s^2 + (\boldsymbol{B} - \frac{\rho c \boldsymbol{v}}{4} A^{\frac{i}{k}}) + (\boldsymbol{K_s} - \frac{\rho \boldsymbol{v}^2}{2} A^r)]\boldsymbol{x}(s) = \boldsymbol{F}(s)\,, \tag{2.5}$$

   where $\boldsymbol{M}$, $\boldsymbol{B}$, and $\boldsymbol{K}$ are the mass, damping and stiffness matrices. A is the generalized Aerodynamic Influence Coefficient (AIC), $\boldsymbol{x}$ is the generalized deflection, $\boldsymbol{F}$ is the generalized force and s is the Laplace variable;

2. Find the eigenvalues and eigenvectors utilizing a flutter solution technique;

3. Using the eigenvalues and eigenvectors to build a time domain state-space model (see Section 5.1).

This method is especially accurate for low-damped modes and can accurately capture the correct mode shapes, frequency, and damping values of rigid and elastic modes. Its roots are consistent with the analytical flutter models. The major setback of this technique is the convergence problems associated with the p-k type of a flutter solution technique.

This technique was used by Boeing to develop aeroelastic models for production aircraft programs. It was used in the development program of several aircraft as the DC-10, Boeing C-13 and MD-11.

2.4.2 FAMUSS

FAMUSS (Flexible Aircraft Modeling Using State Space) technique was developed by Pitt and Goodman at Boeing, in St.Louis. It was used mainly for development programs of vibration control, maneuver and gust load alleviation, and flutter suppression systems.

This tool needed as input polynomials of the frequency response and the aeroelastic roots. Frequency response data was usually generated by traditional frequency-domain tools used for flutter or

dynamic loads, then the FAMUSS input polynomials were created using least mean squares fit. Aeroelastic roots were calculated using a traditional p-k type of flutter solution. This technique was very robust, had a very user friendly code and had a better convergence, in several cases, than the early P-Transform technique, although its accuracy was not as good as P-Transform's.

2.4.3 Rational Function Approximation

Rational Function Approximation (RFA) techniques are used to represent unsteady aerodynamic forces in aeroelastic analytic models. These forces are generated in modal coordinates and represented by a rational function in frequency. Also the aircraft equations of motion can be formulated using a modal approach to represent structural dynamics. Since these functions are in frequency domain, transformation to time domain, using Laplace techniques, is possible. After being in the time domain, these equations can be formulated and cast in state-space form.

The main problem about this aeroelastic model generated by RFA techniques is its size, which affects real-time simulation (control purpose) and quick studies. On the other hand, the accuracy of this model is very high and its robustness is indeed remarkable.

Chapter 3

Dynamics Model

In this Chapter the equations of motion (EOM) of a generic elastic aircraft will be defined.

3.1 Reference Frames and Angles

When working with a flight dynamics' problem it is crucial to choose a proper reference frame that specifies the needs of the problem.

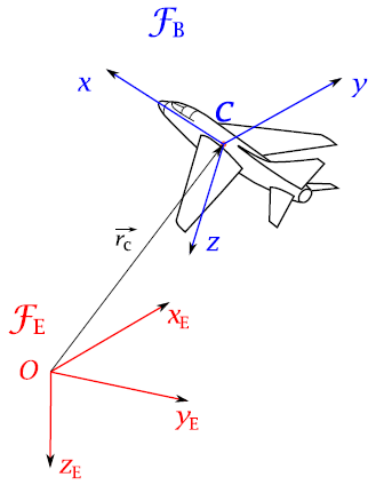

Figure 3.1: Fixed reference frame, F_E, and aircraft reference frame, F_B [12]

In any dynamics' problem there must be an inertial reference frame (F_E), which means that the reference frame must be fixed. On that frame, applying Newton's second law of motion on a particle shouldn't be a problem. The reference frames, or coordinate systems, consist of three mutually orthogonal axes.

Before advancing to the equations of motion (EOM) of this project, several reference frames will be presented. According to [13], the most used systems are:

- **ECEF System** (Earth-Centered, Earth-Fixed): The origin of this frame is in the center of mass of the Earth (Earth-Centered). Its x-axis passes by the intersection between the equator line and the Greenwich meridian (0º Latitude, 0º Longitude) , the z-axis is along the spin axis of the Earth,

pointing to the north pole and finally the y-axis has its direction and orientation defined by the right hand rule. The position vector in the ECEF frame is denoted by

$$\boldsymbol{p_{ECEF}} = [x_{ECEF}[m], y_{ECEF}[m], z_{ECEF}[m]]\,. \tag{3.1}$$

- **LLH System** (Latitude-Longitude-Height): Just like ECEF system, this coordinate system also has its center in the Earth's center of mass. The First Meridian and the Equator are references from which latitude and longitude are defined. Geodetic latitude is characterized by the angle between the equatorial plane and the normal to the surface of the reference ellipsoid. Geodetic longitude is the angle between the plane defined by First Meridian and another meridian. Finally the geodetic height is the distance from the surface ellipsoid to a point in a normal direction of the ellipsoid. The position vector in the LLH frame is denoted by

$$\boldsymbol{p_{LLH}} = [Latitude[^{\circ}], Longitude[^{\circ}], H[m]]\,. \tag{3.2}$$

- **NED System** (North-East-Down): The local NED coordinate system is also known for navigation or ground coordinate system. The origin is arbitrarily fixed to a point on the Earth's surface. The x-axis points towards the ellipsoid north (geodetic north), y-axis points to ellipsoid east (geodetic east) and finally, the z-axis points downward along the ellipsoid normal. Another related coordinate system is ENU (East-North-Up), the transformation from the NED reference frame to the ENU reference frame is in Appendix A.1.1. The position vector in the NED frame is denoted by

$$\boldsymbol{p_{NED}} = [x_{NED}[m], y_{NED}[m], z_{NED}[m]]\,, \tag{3.3}$$

  and in the ENU frame is,

$$\boldsymbol{p_{ENU}} = [x_{ENU}[m], y_{ENU}[m], z_{ENU}[m]]\,, \tag{3.4}$$

- **RPY System** (Roll-Pitch-Yaw): System whose axes are fixed on a vehicle. The origin is located at the center of gravity (c_g) of the flying vehicle. The x-axis (Roll) of the RPY system points in the forward direction of the vehicle movement, the y-axis (Pitch) is starboard (the right side of the flying vehicle) and finally the z-axis (Yaw) is pointed downward. This system is also referred to as the vehicle axis system or body axis system.

For this dissertation two reference frames will be used: one fixed and another one relative. The fixed reference frame utilized is the NED system while in the relative one it is the RPY system. The NED system facilitates the positioning analysis of the aircraft. Also this reference frame can be considered inertial, where Newton's laws are applied , as the Earth's rotational speed can be neglected.

The aircraft positioning can be defined by a combination of rotations and translations, from a reference position on the RPY system. This reference position coincides with the fixed reference frame, the NED system. With this result velocities and accelerations (both linear or angular) relative to the fixed

frame can be expressed on the local reference frame (RPY). Angular orientation of the aircraft is defined by a set of rotations of the fixed reference frame, the Euler angles.

3.1.1 Euler Angles

In flight dynamics the orientation of any reference frame relative to another can be given by three angles, Euler angles (ϕ,θ,ψ). A function of these angles allow the mutual transformation from a fixed reference frame (NED coordinate system, F_E) to the local reference frame (vehicle body frame, F_B) [14]. For small angles, each of the Euler angles has the following designation:

- **Roll angle**: $\phi \in$ [-π,π] (rad) ,
- **Yaw angle**: $\psi \in$ [0,2π] (rad) ,
- **Pitch angle**: $\theta \in$ [-$\frac{\pi}{2}$,$\frac{\pi}{2}$] (rad) .

In order to move from one reference frame to another, a sequence of rotations has to be done.

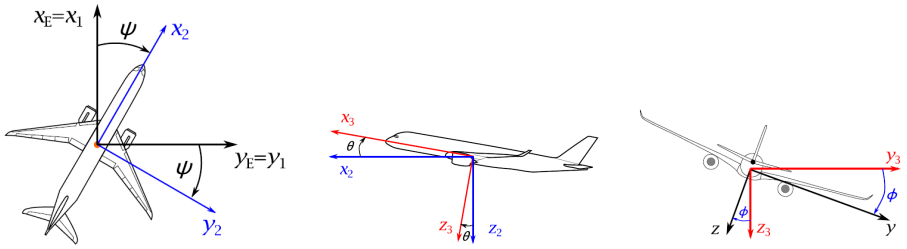

Figure 3.2: Rotations used to define the yaw, pitch and roll angle, respectively (ψ, θ,ϕ) [12]

As seen in Figure 3.2, yaw angle (ψ) is defined by the rotation over the z_E axis. Pitch angle (θ) is then defined by the rotation over the y_1 axis, that appeared from the first rotation. Finally the roll angle (ϕ) is defined by the rotation over the x_2 axis, from the second rotation.

However the Euler angles have a setback, the singularities. These happen for $\psi = \pm\frac{\pi}{2}$ and $\phi = 0, \pi$ [15]. As an alternative to Euler angles, the quaternions have the advantage of not having singularities and having simpler derivatives. Although quaternions' physical meaning is less intuitive than Euler angles. So in this dissertation, as the simulation should not have problems with these singularities, Euler angles will be the choice for reference frames transformation.

3.1.2 Aerodynamic Angles

The vehicle motion relative to the atmosphere $\boldsymbol{v}_B$ (true airspeed, TAS) can be expressed by its three orthogonal components (v_x,v_y,v_z) in the body axis system. Alternatively, two suitable angles can be used to define these velocities. These angles are of the most importance to characterize aerodynamic forces that act on the vehicle, and these angles are:

- **Angle of attack** (α [rad]):

$$\alpha = tan^{-1}\frac{v_z}{v_x}\,, \tag{3.5}$$

- **Sideslip angle** (β [rad]):

$$\beta = sin^{-1}\frac{v_y}{|\boldsymbol{v_B}|}\,. \tag{3.6}$$

Linear velocity components of the aircraft can be expressed using these aerodynamic angles,

$$u = |\boldsymbol{v_B}|cos(\beta)cos(\alpha)\,, \tag{3.7}$$

$$v = |\boldsymbol{v_B}|sin(\beta)\,, \tag{3.8}$$

$$w = |\boldsymbol{v_B}|cos(\beta)sin(\alpha)\,. \tag{3.9}$$

3.1.3 Angular Velocities

From now on until the end of this dissertation variables depending on time will be represented as $\dot{u}$ ($\dot{u} = \frac{d}{dt}u$). Angular velocity of an aircraft is usually defined in the aircraft's reference frame,

$$\boldsymbol{w} = w_x\boldsymbol{e_{x_B}} + w_y\boldsymbol{e_{y_B}} + w_z\boldsymbol{e_{z_B}}\,. \tag{3.10}$$

On the other hand, from the definition of Euler angles it follows that angular velocity can be written as function of the angular rates $\dot{\psi}$, $\dot{\theta}$, $\dot{\phi}$. The relation of these angular rates with the angular speed components comes from the Euler angles definition,

$$\boldsymbol{w} = (\dot{\phi} - \dot{\psi}sin(\theta))\boldsymbol{e_{x_B}} + (\dot{\psi}cos(\theta)sin(\phi) + \dot{\theta}cos(\phi))\boldsymbol{e_{y_B}} + (\dot{\psi}cos(\theta)cos(\phi) - \dot{\theta}sin(\phi))\boldsymbol{e_{z_B}}\,, \tag{3.11}$$

Or,

$$\begin{cases} w_x = \dot{\phi} - \dot{\psi}sin(\theta) \\ w_y = \dot{\psi}cos(\theta)sin(\phi) + \dot{\theta}cos(\phi) \\ w_z = \dot{\psi}cos(\theta)cos(\phi) - \dot{\theta}sin(\phi) \end{cases}\,. \tag{3.12}$$

Now that the project reference frames are determined, development of the equations of motion may begin.

3.2 Rigid Body Flight Dynamics

In order to reach the elastic body lateral and longitudinal equations, first the rigid body ones will be demonstrated, and then further adjustments will be done in pursuance of the elastic body flight dynamics mathematical model.

3.2.1 Equations of Motion

The equations of motion are a result from the application of Newton-Euler formulation in classic mechanics to the flight vehicle, in the fixed reference frame (subscript E). Applying these, and considering for now constant mass and constant inertia throughout time, two crucial equations emerge.

One for linear moment

$$\boldsymbol{F} = m\dot{\boldsymbol{v_E}}\,, \tag{3.13}$$

where $\boldsymbol{F}$ represents the resultant of all external forces applied on the aircraft, m is the aircraft's mass, $\boldsymbol{v_E}$ the vehicle linear motion vector relative to the fixed reference frame.

And finally the angular moment equation

$$\boldsymbol{M} = \dot{\boldsymbol{H}}_E = [\boldsymbol{I}\dot{\boldsymbol{w}}]_E = \boldsymbol{I}\dot{\boldsymbol{w_E}}\,, \tag{3.14}$$

where $\boldsymbol{M}$ represents the resultant external moment, $\boldsymbol{H}$ is the total moment relative to the aircraft's center of mass. The inertia tensor matrix $\boldsymbol{I}$ is defined as

$$\boldsymbol{I} = \begin{bmatrix} I_x & -I_{xy} & -I_{xz} \\ -I_{yx} & I_y & -I_{yz} \\ -I_{zx} & -I_{zy} & I_z \end{bmatrix} = \begin{bmatrix} A & -F & -E \\ -F & B & -D \\ -E & -D & C \end{bmatrix}. \tag{3.15}$$

Both notations for the elements of $\boldsymbol{I}$ given in (3.15) are in current use in flight dynamics, however through this project the American nomenclature will be utilized.

Since writing the equations of motion in the local reference frame is the objective, applying a reference frame transformation [16] from the fixed reference frame to the body axis reference frame, the external force equation, (3.13) is altered to

$$\boldsymbol{F_B} = \frac{d}{dt}[m\boldsymbol{v_E}]_B = m[\dot{v_B} + \boldsymbol{\Omega} \times (\boldsymbol{v_E})_B] = m[\dot{v_B} + \boldsymbol{w_B} \times \boldsymbol{v_B}]\,, \tag{3.16}$$

where $\boldsymbol{\Omega}$ is the angular speed of the fixed reference frame relatively to the body reference frame, in this case $\boldsymbol{w_B}$.

And the angular moment equation (3.14) is now expressed as

$$\boldsymbol{M_B} = \frac{d}{dt}[\boldsymbol{H}_E]_B = \dot{H}_B + \boldsymbol{\Omega} \times \boldsymbol{H_E}_B = \boldsymbol{I}\dot{w}_B + \boldsymbol{w_B} \times \boldsymbol{I}\boldsymbol{w_B}\,, \tag{3.17}$$

and as previously, $\boldsymbol{\Omega}$ has the same meaning as in the (3.16).

The vectors and matrix from equations, (3.16) and (3.17) are now going to be defined following the traditional (American) nomenclature:

- external force: $\boldsymbol{F_B} = [X, Y, Z]_B^T$;
- air velocity (TAS): $\boldsymbol{v_B} = [u, v, w]_B^T$;
- external moment: $\boldsymbol{M_B} = [L, M, N]_B^T$;
- angular speed: $\boldsymbol{w_B} = [p, q, r]_B^T$.

The subscript B indicates that these vectors are defined in the body axis reference frame.

3.2.2 Control Surfaces

Every conventional aircraft have control surfaces which, with the aid of control laws, produce forces and moments required to generate the desired accelerations for the action specified [17].

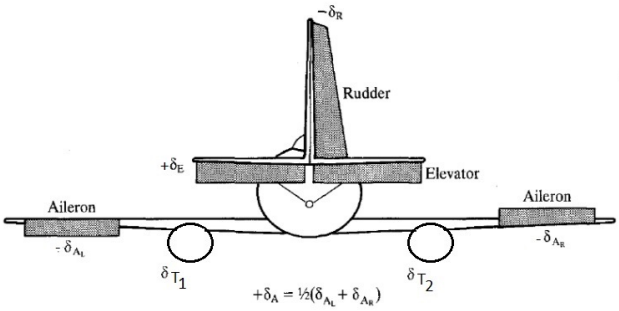

Figure 3.3: Control Surfaces deflection conventions

In Figure 3.3, it is shown the primary control surfaces and the deflection convention used in this project. One of the traits of flight control is the simultaneously use of these to perform actions and compensate external disturbances.

When considering decoupled motion, demonstrated in Section 3.2.6, it is safe to say that two control surfaces are responsible for longitudinal motion and other two for lateral motion.

For longitudinal motion these two are, the elevator deflection (δ_E) and the change in thrust (δ_T). For this particular project it is considered the change in thrust for each engine δ_{T_i}, so the convention used to enumerate the engines is left to right looking from backwards of the aircraft, as it is seen in Figure 3.3. In the lateral motion it is the ailerons deflection (δ_A) and the rudder deflection (δ_R).

3.2.3 Applied Forces and Moments

Before moving to the force and moment body axis equations, it is necessary to characterize the applied forces on the aircraft. As see in Figure 3.4 the external force vector in flight consists of three types of forces:

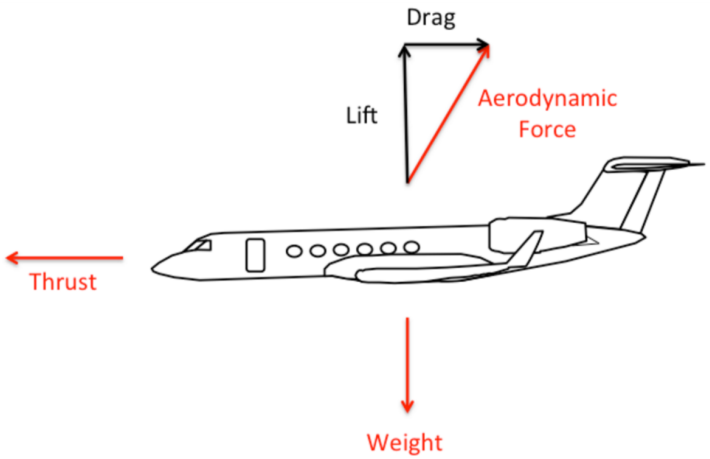

Figure 3.4: Applied forces on a general aircraft (adapted from [18])

- Aerodynamic forces,

$$\boldsymbol{F_{a,B}} = f(\boldsymbol{v}, \alpha, \beta, p, q, r, \delta_E, \delta_A, \delta_R, ...)\,, \tag{3.18}$$

all types of these force are represented in this equation (3.18). Drag, lift and side force are a function of several variables. Principal causes inflicting changes in the aerodynamic forces listed above are movement of the aircraft through air, angular rates and control surfaces positioning;

- Propulsive forces (Thrust),

$$\boldsymbol{F_{p,B}} = f(\boldsymbol{v}, \alpha, \beta, h, \delta_T, RPM, ...)\,, \tag{3.19}$$

depend especially on, rotor rotation speed (RPM), engine throttle (δ_T), velocity of the aircraft ($\boldsymbol{v}$) and height (h);

- Weight force,

$$\boldsymbol{F_{g,B}} = \boldsymbol{L_{BE}}\boldsymbol{F_{g,E}} = \boldsymbol{L_{BE}} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} = mg \begin{bmatrix} -sin\theta \\ cos\theta sin\phi \\ cos\theta cos\phi \end{bmatrix}\,, \tag{3.20}$$

is a constant on the fixed reference frame and it is expressed on the body axis reference frame by a Euler transformation. This expression of the weight on the body axis ($\boldsymbol{F_{g,B}}$), is obtained by the multiplication of the reference frame rotation matrix (Appendix A.1, equation (A.5)) with the weight on the fixed reference frame ($\boldsymbol{F_{g,E}}$).

Having the equations of motion on the body axis reference frame, (3.16) and (3.17), and by knowing the external forces applied on the aircraft, the following six equations of motion for the body reference frame emerge. Expanding equation (3.16) in its components yields

$$\begin{cases} X = m[\dot{u} + qw - rv] \\ Y = m[\dot{v} + ur - pw] \\ Z = m[\dot{w} + vp - uq] \end{cases}\,, \tag{3.21}$$

and by discriminating aerodynamic, propulsive and weight forces

$$\begin{cases} X_a + X_p - mgsin\theta = m[\dot{u} + qw - rv] \\ Y_a + Y_p + mgcos\theta sin\phi = m[\dot{v} + ur - pw] \\ Z_a + Z_p + mgcos\theta cos\phi = m[\dot{w} + vp - uq] \end{cases}\,. \tag{3.22}$$

Expanding equation (3.17) by applying cross product, considering symmetry in the aircraft ($I_{yz} = I_{xy} = 0$) and discriminating aerodynamic and propulsive moments, yields for its components

$$\begin{cases} L_a + L_p = I_x\dot{p} - I_{xz}(\dot{r} + pq) + qr(I_z - I_y) \\ M_a + M_p = I_y\dot{q} + I_{xz}(p^2 - r^2) + pr(I_x - I_z) \\ N_a + N_p = I_z\dot{r} + I_{xz}(qr - \dot{p}) + pq(I_y - I_x) \end{cases}\,. \tag{3.23}$$

3.2.4 Small Disturbance Theory

For control purposes, the resolution of the nine equations of motion (3.22), (3.23) and (3.12), usually passes through its linearization, obtained from the small disturbance theory.

The small disturbance theory is a powerful tool with a wide range of applications, being stability and control response two of those. Two of the reasons why this theory is widely used are: in many cases the aerodynamic effects act as linear functions of the state variables and disturbed flight (gusts) can correspond to relatively small changes in the linear and angular velocities. However it is not suitable for applications that inflict large variations on state variables and for certain flight movements like spinning [19].

A non-linear function $f(X, Y, ...)$ can be approximated, in a certain reference point (X_0,Y_0,...), by the tangent linear function through Taylor's first order expansion [14]

$$f(X, Y, ...) = f(X_0, Y_0, ...) + \frac{\partial f}{\partial X}|_0(X - X_0) + \frac{\partial f}{\partial Y}|_0(Y - Y_0) + ... \,. \tag{3.24}$$

Each variable (X) is then expressed as the sum of a equilibrium state term (X_0), where the subscript 0 means equilibrium state, with a small disturbance term (x)

$$X = X_0 + x \,. \tag{3.25}$$

In flight dynamics, the equilibrium point (X_0) is usually defined as a trim flight with constant air velocity. The linearized equations for small disturbances are now obtained around this equilibrium point, simplifying the representation of aerodynamic and propulsive - forces and moments.

3.2.5 Steady States

Rigid aircraft motion equations are non-linear, therefore in order to get the decoupled longitudinal and lateral motion equations, it is needed to search particular linear solutions of the rigid body equations. These linear solutions happen for steady flight states. In a typical flight mission, an aircraft is usually in a steady flight state. Descent, climb, cruise flight are some of these states. Their importance is immense because they allow the study of dynamic stability for small disturbances.

For the matter of this dissertation only two of these steady flight states are going to be discussed, the rectilinear flight and the coordinated turn flight.

Rectilinear Flight

In this state, angular velocities and angular rates are despised due to rectilinear and trim flight conditions, respectively. Systems (3.16) and (3.17) are altered to

$$\begin{cases} X_0 = mgsin(\theta_0) \\ Y_0 = mgcos(\theta_0)sin(\phi_0) \\ Z_0 = mgcos(\theta_0)cos(\phi_0) \\ L_0 = M_0 = N_0 \\ p_0 = q_0 = r_0 \end{cases} . \tag{3.26}$$

With $\phi_0 = 0$ and implying the non-existence of wind, the flight has null roll and no lateral force, simplifying the system to

$$\begin{cases} X_0 = mgsin(\theta_0) \\ Y_0 = 0 \\ Z_0 = mgcos(\theta_0) \\ L_0 = M_0 = N_0 \\ p_0 = q_0 = r_0 \end{cases} . \tag{3.27}$$

Coordinated Turn Flight

In the coordinated turn flight, all time derivatives are null, except for the yaw rate which is constant,

$$\frac{d\psi}{dt} = const\,. \tag{3.28}$$

Kinetic equations (3.12) change to

$$\begin{cases} p = -\dot{\psi}sin(\theta_0) \\ q = \dot{\psi}cos(\theta_0)sin(\phi_0) \\ r = \dot{\psi}cos(\theta_0)cos(\phi_0) \end{cases} . \tag{3.29}$$

Coordinated turn corresponds to a uniform circular flight where the lateral force is null, altering the system (3.22) to

$$\begin{cases} X_0 = mgsin(\theta_0) \\ Y_0 = m(u_0\dot{\psi}cos(\phi_0) - gsin(\phi_0)) = 0 \\ Z_0 = m(-u_0\dot{\psi}sin(\phi_0) - gcos(\phi_0) \end{cases} \rightarrow \dot{\psi} = \frac{g}{u_0}tan(\phi_0)\,. \tag{3.30}$$

With the expanded general equations of motion, defined on the body axis reference frame, and with a defined steady state, it is possible to begin the decoupling of these equations for lateral and longitudinal movement.

3.2.6 Decoupled Equations

A rectilinear flight state, which has no slideslip, with leveled wings will be utilized for this demonstration. The purpose is to reach the main consequence of the linearization on rectilinear flight which is the decoupled movement equations for longitudinal and lateral motion.

Using the steady state flight described above and applying linearization, several variables can be neglected: linear accelerations ($\dot{u_0} = \dot{v_0} = \dot{w_0} = 0$), angular velocities ($p_0 = q_0 = r_0 = 0$), roll angle ($\phi_0 = 0$), climb and lateral velocity ($v_0 = h_0 = 0$) and angle of attack ($\alpha = 0$). If the linearization process is applied to the angular moment equations (3.23), Euler angles kinetic equations (3.12), to the force equations (3.22) and considering the rectilinear flight, these three systems result in

$$\begin{cases} x = m[\dot{u} + qw_0 + gsin(\theta_0)\theta] \\ y = m[\dot{v} + u_0 r - pw_0 - gcos(\theta_0)\phi] \\ z = m[\dot{w} - u_0 q + gsin(\theta_0)\theta] \end{cases}, \tag{3.31}$$

$$\begin{cases} l = I_x\dot{p} - I_{xz}\dot{r} \\ m = I_y\dot{q} \\ n = I_z\dot{r} - I_{xz}\dot{p} \end{cases}, \tag{3.32}$$

$$\begin{cases} p = \dot{\phi} - \dot{\psi}sin(\theta_0) \\ q = \dot{\theta} \\ r = \dot{\psi}cos(\theta_0) \end{cases}. \tag{3.33}$$

If equations, (3.31), (3.32) and (3.33) are reorganized two decoupled modes emerge:

1. Longitudinal: Depending on the following variables, u, w, q, and θ;

$$\begin{cases} x = m[\dot{u} + qw_0 + gsin(\theta_0)\theta] \\ z = m[\dot{w} - u_0 q + gsin(\theta_0)\theta] \\ m = I_y\dot{q} \\ q = \dot{\theta} \end{cases}. \tag{3.34}$$

2. Lateral: Depending on the following variables, v, p, r, ϕ and ψ.

$$\begin{cases} y = m[\dot{v} + u_0 r - pw_0 - gcos(\theta_0)\phi] \\ l = I_x\dot{p} - I_{xz}\dot{r} \\ n = I_z\dot{r} - I_{xz}\dot{p} \\ p = \dot{\phi} - \dot{\psi}sin(\theta_0) \\ r = \dot{\psi}cos(\theta_0) \end{cases}. \tag{3.35}$$

Now, in order to expand these equations, forces and moments need to be linearized.

Linearized Forces and Moments

When linearizing forces and moments, it is important to keep in mind that longitudinal forces and moments only depend on the longitudinal variables, and the same for lateral forces and moments. In favour of easing the problem, forces and moments only depend linearly with the state variables. The notation used to define forces and moments for partial derivatives is the **stability derivative**. Let F be a scalar representing a force vector component in an axis, and the subscript i a variable. The stability derivative of F with respect to i is

$$F_i = \frac{1}{m}\frac{\partial F}{\partial i}\,. \tag{3.36}$$

Let M be a scalar representing a moment vector component in an axis, and the subscript i a variable. The stability derivative of M with respect to i is

$$M_i = \frac{1}{I_i}\frac{\partial M}{\partial i}\,. \tag{3.37}$$

Therefore after this introduction to stability derivatives, the expansion of longitudinal forces and moments is

$$\begin{cases} \frac{x}{m} = X_u u + X_{\dot{u}}\dot{u} + X_w w + X_{\dot{w}}\dot{w} + X_q q + X_{\dot{q}}\dot{q} + X_{\delta_E}\delta_E + X_{\delta_T}\delta_T \\ \frac{z}{m} = Z_u u + Z_{\dot{u}}\dot{u} + Z_w w + Z_{\dot{w}}\dot{w} + Z_q q + Z_{\dot{q}}\dot{q} + Z_{\delta_E}\delta_E + Z_{\delta_T}\delta_T \\ \frac{m}{I_y} = M_u u + M_{\dot{u}}\dot{u} + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\dot{q}}\dot{q} + M_{\delta_E}\delta_E + M_{\delta_T}\delta_T \end{cases}, \tag{3.38}$$

usually the forces on the x_B and z_B axis, depending on the state variables, such as q and $\dot{w}$, can be neglected. Forces and moments depending on $\dot{u}$ and $\dot{q}$, are not relevant for the longitudinal force and moment system, so they are also neglected. So the system (3.38) transforms into

$$\begin{cases} \frac{x}{m} = X_u u + X_w w + X_{\delta_E}\delta_E + X_{\delta_T}\delta_T \\ \frac{z}{m} = Z_u u + + Z_w w + Z_{\delta_E}\delta_E + Z_{\delta_T}\delta_T \\ \frac{m}{I_y} = M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta_E}\delta_E + M_{\delta_T}\delta_T \end{cases}. \tag{3.39}$$

Lateral forces and moments equations solely depend on state variables v, p and r

$$\begin{cases} \frac{y}{m} = Y_v v + Y_p p + Y_r r + Y_{\delta_A}\delta_A + Y_{\delta_R}\delta_R \\ \frac{l}{I_x} = L_v v + L_p p + L_r r + L_{\delta_A}\delta_A + L_{\delta_R}\delta_R \\ \frac{n}{I_y} = N_v v + N_p p + N_r r + N_{\delta_A}\delta_A + N_{\delta_R}\delta_R \end{cases}. \tag{3.40}$$

Linearized Decoupled Equations

Now, substituting longitudinal forces and moments from the system (3.39) in (3.34). The final decoupled system representing the longitudinal motion for a rigid aircraft is

$$\begin{cases} \dot{u} = X_u u - w_0 q + X_w w - g cos(\theta_0)\theta + X_{\delta_E}\delta_E + X_{\delta_T}\delta_T \\ \dot{w} = Z_u u + u_0 q + Z_w w - g sin(\theta_0)\theta + Z_{\delta_E}\delta_E + Z_{\delta_T}\delta_T \\ \dot{q} = M_u u + M_w w + M_{\dot{w}}\dot{w} + M_q q + M_{\delta_E}\delta_E + M_{\delta_T}\delta_T \\ \dot{\theta} = q \end{cases}. \tag{3.41}$$

Doing the same substitution of the system (3.40) in (3.35), the final decoupled system for lateral motion is

$$\begin{cases} \dot{v} = Y_v v + p(Y_p + w_0) + r(Y_r - u_0) + gcos(\theta_0)\phi + Y_{\delta_A}\delta_A + Y_{\delta_R}\delta_R \\ \dot{p} = \frac{I_{xz}}{I_x}\dot{r} + L_v v + L_p p + L_r r + L_{\delta_A}\delta_A + L_{\delta_R}\delta_R \\ \dot{r} = \frac{I_{xz}}{I_z}\dot{p} + N_v v + N_p p + N_r r + N_{\delta_A}\delta_A + N_{\delta_R}\delta_R \\ \dot{\phi} = p + tan(\theta_0) r \\ \dot{\psi} = \frac{r}{cos(\theta_0)} \end{cases} . \tag{3.42}$$

All stability derivatives of (3.42) and (3.41) are in Appendix A.2. After substituting $\dot{w}$ on the third equation of system (3.41)

$$\begin{cases} \dot{u} = X_u u - w_0 q + X_w w - gcos(\theta_0)\theta + X_{\delta_E}\delta_E + X_{\delta_T}\delta_T \\ \dot{w} = Z_u u + u_0 q + Z_w w - gsin(\theta_0)\theta + Z_{\delta_E}\delta_E + Z_{\delta_T}\delta_T \\ \dot{q} = \tilde{M}_u u + \tilde{M}_w w + \tilde{M}_q q + \tilde{M}_\theta \theta + \tilde{M}_{\delta_E}\delta_E + \tilde{M}_{\delta_T}\delta_T \\ \dot{\theta} = q \end{cases} . \tag{3.43}$$

The new coefficients of the third equation ($\dot{q}$), are defined as

$$\tilde{M}_u = M_u + M_{\dot{w}} Z_u \qquad \tilde{M}_w = M_w + M_{\dot{w}} Z_w \qquad \tilde{M}_q = M_q + M_{\dot{w}} u_0 \,, \tag{3.44}$$

$$\tilde{M}_\theta = -M_{\dot{w}} gsin(\theta_0) \qquad \tilde{M}_{\delta_E} = M_{\delta_E} + M_{\dot{w}} Z_{\delta_E} \qquad \tilde{M}_{\delta_T} = M_{\delta_T} + M_{\dot{w}} Z_{\delta_T} \,. \tag{3.45}$$

As for the lateral mode, the system (3.42) reaches its final form after substituting the $\dot{p}$ and $\dot{r}$ terms of the second and third equations

$$\begin{cases} \dot{v} = Y_v v + p(Y_p + w_0) + r(Y_r - u_0) + gcos(\theta_0)\phi + Y_{\delta_A}\delta_A + Y_{\delta_R}\delta_R \\ \dot{p} = L'_v v + L'_p p + L'_r r + L'_{\delta_A}\delta_A + L'_{\delta_R}\delta_R \\ \dot{r} = N'_v v + N'_p p + N'_r r + N'_{\delta_A}\delta_A + N'_{\delta_R}\delta_R \\ \dot{\phi} = p + tan(\theta_0) r \\ \dot{\psi} = \frac{r}{cos(\theta_0)} \end{cases} . \tag{3.46}$$

The introduced stability derivatives are

$$L'_v = L_v + \frac{I_{xz}}{I_x} N_v \qquad L'_p = L_p + \frac{I_{xz}}{I_x} N_p \qquad L'_r = L_r + \frac{I_{xz}}{I_x} N_r \,, \tag{3.47}$$

$$N'_v = N_v + \frac{I_{xz}}{I_z} L_v \qquad N'_p = N_p + \frac{I_{xz}}{I_z} L_p \qquad N'_r = N_r + \frac{I_{xz}}{I_z} L_r \,. \tag{3.48}$$

3.2.7 Engine Contribution

Engine failure is one of the major causes of accident, especially on take-off and landing [20]. So in this model, instead of considering the engine contribution as a whole, the systems are redefined to have each individual engine contribution. By having the individual engine influence on the dynamics model, it will be possible to simulate adverse situations where an engine is unavailable and therefore observe the

aircraft dynamic behaviour. The individual engine force contribution can be specified as

$$Z_{\delta_T} = \sum_{i=1}^{N_{eng}} Z_{\delta_{T_i}} \Rightarrow Z_{\delta_T} = N_{eng} Z_{\delta_{T_i}} \Rightarrow Z_{\delta_{T_i}} = \frac{Z_{\delta_T}}{N_{eng}}\,, \tag{3.49}$$

$$X_{\delta_T} = \sum_{i=1}^{N_{eng}} X_{\delta_{T_i}} \Rightarrow X_{\delta_T} = N_{eng} X_{\delta_{T_i}} \Rightarrow X_{\delta_{T_i}} = \frac{X_{\delta_T}}{N_{eng}}\,, \tag{3.50}$$

$$Y_{\delta_{T_i}} = 0\,, \tag{3.51}$$

where, N_{eng} represents the total number of engines the aircraft have and the stability derivatives with the subscript i are due to each engine contribution, as see in Figure 3.3. It is considered, in this project, that the engines thrust have only components on the x_B axis and, depending on the case, on the z_B axis. Although it may not be entirely true, since single turboprop engine aircraft maybe have thrust induced lateral force to provide equilibrium. The individual engine moment contribution is

$$L_{\delta_T} = -\sum_{i=1}^{N_{eng}} L_{\delta_{T_i}} \Rightarrow L_{\delta_T} = -\sum_{i=1}^{N_{eng}} y_i Z_{\delta_{T_i}} \Rightarrow L_{\delta_{T_i}} = -y_i Z_{\delta_{T_i}}\,, \tag{3.52}$$

$$\tilde{M_{\delta_T}} = \sum_{i=1}^{N_{eng}} M_{\delta_{T_i}} \Rightarrow M_{\delta_T} = \sum_{i=1}^{N_{eng}} (z_i X_{\delta_{T_i}} - x_i Z_{\delta_{T_i}}) \Rightarrow M_{\delta_{T_i}} = (z_i X_{\delta_{T_i}} - x_i Z_{\delta_{T_i}})\,, \tag{3.53}$$

$$N_{\delta_T} = \sum_{i=1}^{N_{eng}} N_{\delta_{T_i}} \Rightarrow N_{\delta_T} = \sum_{i=1}^{N_{eng}} y_i X_{\delta_{T_i}} \Rightarrow N_{\delta_{T_i}} = y_i X_{\delta_{T_i}}\,, \tag{3.54}$$

where the coordinates (x_i, y_i, z_i) are the engine i coordinates in the body axis system and $(L_{\delta_{T_i}}, M_{\delta_{T_i}}, N_{\delta_{T_i}})$ are the torque induced on the aircraft by that engine. If an aircraft has only one engine, a single engine aircraft it will have $L_{\delta_T} = 0$ and $N_{\delta_T} = 0$, because its position in the y-axis from the body axis system is null. In this project the maximum number of engines considered for an aircraft is five.

The lateral and longitudinal equations that once were completely decoupled, now are coupled by the engine change thrust (δ_{T_i}). Assuming the aircraft has N_{eng} engines, adding the engine moment contributions and using the lateral velocity (v) and vertial velocity (w) approximations

$$v \approx u_0 \beta \qquad w_0 \approx u_0 \alpha_0\,, \tag{3.55}$$

equations (3.43) and (3.46) can be rewritten as

$$\begin{cases} \dot{u} = X_u u - w_0 q + X_w w - g cos(\theta_0)\theta + X_{\delta_E}\delta_E + \sum_{i=1}^{N_{eng}} X_{\delta_{T_i}}\delta_{T_i} \\ \dot{w} = Z_u u + u_0 q + Z_w w - g sin(\theta_0)\theta + Z_{\delta_E}\delta_E + \sum_{i=1}^{N_{eng}} Z_{\delta_{T_i}}\delta_{T_i} \\ \dot{q} = \tilde{M_u} u + \tilde{M_w} w + \tilde{M_q} q + \tilde{M_\theta}\theta + \tilde{M_{\delta_E}}\delta_E + \sum_{i=1}^{N_{eng}} \delta_{T_i}(z_i X_{\delta_{T_i}} - x_i Z_{\delta_{T_i}}) \\ \dot{\theta} = q \end{cases}\,, \tag{3.56}$$

$$\begin{cases} \dot{\beta} = Y_\beta \beta + p(\frac{Y_p}{u_0} + \alpha_0) + r(\frac{Y_r}{u_0} - 1) + \frac{gcos(\theta_0)}{u_0}\phi + \frac{Y_{\delta_A}}{u_0}\delta_A + \frac{Y_{\delta_R}}{u_0}\delta_R \\ \dot{p} = L'_\beta \beta + L'_p p + L'_r r + L'_{\delta_A}\delta_A + L'_{\delta_R}\delta_R - \sum_{i=1}^{N_{eng}} y_i Z_{\delta_{T_i}}\delta_{Ti} \\ \dot{r} = N'_\beta \beta + N'_p p + N'_r r + N'_{\delta_A}\delta_A + N'_{\delta_R}\delta_R + \sum_{i=1}^{N_{eng}} y_i X_{\delta_{T_i}}\delta_{Ti} \\ \dot{\phi} = p + tan(\theta_0) r \\ \dot{\psi} = \frac{r}{cos(\theta_0)} \end{cases} , \tag{3.57}$$

where

$$L'_\beta = L_\beta + \frac{I_{xz}}{I_x} N_\beta \,, \qquad N'_\beta = N_\beta + \frac{I_{xz}}{I_z} L_\beta \,. \tag{3.58}$$

After the demonstration of the longitudinal and lateral motion equations coupled by individual engine contribution, now the main goal is to define the alterations to be made in order to incorporate elastic aircraft behaviour.

3.3 Elastic Aircraft Consideration

The current design adopted for new military and commercial aircraft have one final goal that is preserve fuel. In order to reach that goal such aircraft configurations have required the use of thin lifting surfaces, long and slender fuselages, low mass fraction structures, high stress design levels and low dynamic load factors. The addiction of all these traits result in aircraft which are structurally light and flexible [21]. Such aircraft can develop large values of displacement and acceleration as a result of structural deflection. So, aeroelastic effects can noticeably alter the ”rigid-body” dynamics of a vehicle, especially when the aircraft is susceptible to pilot inputs and/or atmosphere turbulence [22].

3.3.1 Dynamics of a Flexible Aircraft

One of the primary effects of flexibility is that the center of mass (CM) of the aircraft is continuously changing, and for that reason is not advisable to use the CM of the rigid aircraft as the origin of the reference frame. The reference frame used by aeroelasticians is the mean axis system. It decouples the equations of motion for flight dynamics from the equations for structural dynamics for small deformations [23]. Demonstration of this mean axis system is presented in [24].

When aeroelastic effects are taken into account, new state variables and their respective equations, representing a set of generalized coordinates associated with the vibration (bending, torsion, mixed, among others) modes need to be added to the flight dynamics equations system, (3.56) and (3.57). These normal modes of vibration can be represented using generalized coordinates

$$c_{1i}\ddot{q}_i + c_{2i}\dot{q}_i + c_{3i}q_i = F_i \,, \tag{3.59}$$

where F_i is a generalized force, c_{1i}, c_{2i} and c_{3i} are coefficients of the ith generalized coordinate (q_i)

and of its rates [21]. It is possible to represent the vibration mode by two first order, linear, differential equations [25]

$$x_1 = q_i\,, \qquad x_2 = \dot{q}_i\,, \tag{3.60}$$

where

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{-c_{2i}}{c_{1i}} x_2 - \frac{c_{3i}}{c_{1i}} x_1 + \frac{1}{c_{1i}} F_i\,. \end{aligned} \tag{3.61}$$

The pair of first order differential equations (3.61) representing the vibration mode can be used to augment the rigid body dynamics. Usually, the convention for enumerating vibration modes is such that mode 1 corresponds to the mode with lowest vibration frequency. So as the mode number increases, its associated frequency increases. The biggest concern, for flight control designers, is how many structural vibration modes need to be considered to represent the aeroelastic effect in flight dynamics.

There are two methods that approach the problem in different ways [21]:

- **Quasi-static**: the motions of the structure are assumed to be inphase with the rigid mody motion. It is a good method when there is a wide separation between the natural frequencies of the rigid body from those of the elastic motion;

- **Model truncation**: the deleted modes of residual stiffness modes are not represented by any correction factor.

Further discussion and other methods can be found in [21] and [25]. The methods discussed are the most used in aircraft flight control system (AFCS) design.

3.3.2 Mathematical Representation of the Dynamics of a Flexible Aircraft

Assuming the conditions of the flight dynamics system (3.56) and (3.57), the flexibility effects of a general aircraft, for n vibration modes, with the state vectors for longitudinal and lateral motions represented in equation (3.62) and (3.63) respectively

$$\boldsymbol{x_{long}} = \begin{bmatrix} u & w & q & \theta & \lambda_1 & \sigma_1 & \dots & \lambda_n & \sigma_n \end{bmatrix}^T, \tag{3.62}$$

$$\boldsymbol{x_{lat}} = \begin{bmatrix} \beta & p & r & \phi & \psi & \tau_1 & \chi_1 & \dots & \tau_n & \chi_n \end{bmatrix}^T, \tag{3.63}$$

can be represented as

$$
\begin{cases}
\dot{u} = X_u u - w_0 q + X_w w - gcos(\theta_0)\theta + X_{\delta_E}\delta_E + \sum_{i=1}^{N_{eng}} X_{\delta_{T_i}}\delta_{Ti} + X_{\lambda_1}\lambda_1 + X_{\sigma_1}\sigma_1 + ... + X_{\lambda_n}\lambda_n + X_{\sigma_n}\sigma_n \\
\dot{w} = Z_u u + u_0 q + Z_w w - gsin(\theta_0)\theta + Z_{\delta_E}\delta_E + \sum_{i=1}^{N_{eng}} Z_{\delta_{T_i}}\delta_{Ti} + Z_{\lambda_1}\lambda_1 + Z_{\sigma_1}\sigma_1 + ... + Z_{\lambda_n}\lambda_n + Z_{\sigma_n}\sigma_n \\
\dot{q} = \tilde{M}_u u + \tilde{M}_w w + \tilde{M}_q q + \tilde{M}_\theta \theta + \tilde{M}_{\delta_E}\delta_E + \sum_{i=1}^{N_{eng}} \delta_{Ti}(z_i X_{\delta_{T_i}} - x_i Z_{\delta_{T_i}}) + M_{\lambda_1}\lambda_1 + M_{\sigma_1}\sigma_1 \\
\qquad +... + M_{\lambda_n}\lambda_n + M_{\sigma_n}\sigma_n \\
\dot{\theta} = q \\
\dot{\lambda_1} = \sigma_1 \\
\dot{\sigma_1} = -(2\xi_1\omega_1 + \eta_{1_{\sigma_1}})\sigma_1 + (-{\omega_1}^2 + \eta_{1_{\lambda_1}})\lambda_1 + \eta_{1_u} u + \eta_{1_w} w + \eta_{1_q} q + \eta_{1_{\delta_E}}\delta_E + \sum_{i=1}^{N_{eng}} \eta_{1_{\delta_{T_i}}}\delta_{Ti} + ... + \eta_{1_{\sigma_n}}\sigma_n \\
\qquad +\eta_{1_{\lambda_n}}\lambda_n \\
... \\
\dot{\lambda_n} = \sigma_n \\
\dot{\sigma_n} = -(2\xi_n\omega_n + \eta_{n_{\sigma_n}})\sigma_n + (-{\omega_n}^2 + \eta_{n_{\lambda_n}})\lambda_n + \eta_{n_u} u + \eta_{n_w} w + \eta_{n_q} q + \eta_{n_{\delta_E}}\delta_E + \sum_{i=1}^{N_{eng}} \eta_{n_{\delta_{T_i}}}\delta_{Ti} + \eta_{n_{\sigma_1}}\sigma_1 + \\
\qquad \eta_{n_{\lambda_1}}\lambda_1 + ... + \eta_{n_{\sigma_{n-1}}}\sigma_{n-1} + \eta_{n_{\lambda_{n-1}}}\lambda_{n-1}
\end{cases}
, \tag{3.64}
$$

$$
\begin{cases}
\dot{\beta} = Y_\beta\beta + p(\frac{Y_p}{u_0} + \alpha_0) + r(\frac{Y_r}{u_0} - 1) + \frac{gcos(\theta_0)}{u_0}\phi + \frac{Y_{\delta_A}}{u_0}\delta_A + \frac{Y_{\delta_R}}{u_0}\delta_R + Y_{\tau_1}\tau_1 + Y_{\chi_1}\chi_1 + ... + Y_{\tau_n}\tau_n + Y_{\chi_n}\chi_n \\
\dot{p} = L'_\beta\beta + L'_p p + L'_r r + L'_{\delta_A}\delta_A + L'_{\delta_R}\delta_R - \sum_{i=1}^{N_{eng}} y_i Z_{\delta_{T_i}}\delta_{Ti} + L_{\tau_1}\tau_1 + L_{\chi_1}\chi_1 + ... + L_{\tau_n}\tau_n + L_{\chi_n}\chi_n \\
\dot{r} = N'_\beta\beta + N'_p p + N'_r r + N'_{\delta_A}\delta_A + N'_{\delta_R}\delta_R + \sum_{i=1}^{N_{eng}} y_i X_{\delta_{T_i}}\delta_{Ti} + N_{\tau_1}\tau_1 + N_{\chi_1}\chi_1 + ... + N_{\tau_n}\tau_n + N_{\chi_n}\chi_n \\
\dot{\phi} = p + tan(\theta_0) r \\
\dot{\psi} = \frac{r}{cos(\theta_0)} \\
\dot{\tau_1} = \chi_1 \\
\dot{\chi_1} = -2\xi_A\omega_A\chi_1 + -{\omega_A}^2\tau_1 + \mu_{1_\beta}\beta + \mu_{1_p} p + \mu_{1_r} r + \mu_{1_{\delta_A}}\delta_A + \mu_{1_{\delta_R}}\delta_R + ... + \mu_{1_{\chi_n}}\chi_n + \mu_{1_{\tau_n}}\tau_n \\
... \\
\dot{\tau_n} = \chi_n \\
\dot{\chi_n} = -2\xi_Z\omega_Z\chi_n + -{\omega_Z}^2\tau_n + \mu_{n_\beta}\beta + \mu_{n_p} p + \mu_{n_r} r + \mu_{n_{\delta_A}}\delta_A + \mu_{n_{\delta_R}}\delta_R + \mu_{n_{\chi_1}}\chi_1 + \mu_{n_{\tau_1}}\tau_1 \\
\qquad +... + \mu_{n_{\chi_{n-1}}}\chi_{n-1} + \mu_{n_{\tau_{n-1}}}\tau_{n-1}
\end{cases}
, \tag{3.65}
$$

where λ and τ represent the displacement of symmetrical and asymmetrical vibration modes, μ_{n_τ} and η_{n_λ} correspond to the structural derivatives with respect to the n_λ and n_τ vibration modes. Variables χ and σ are used to facilitate interpretation and maintain the system as a first order differential equations system. The variables ξ and ω correlate to the damping ratio and natural frequency.

Chapter 4

Flight Dynamics Model Implementation

One of the goals of this thesis, besides the flight controller, is the implementation of a standalone aircraft flight dynamics client. This client will trade information with other modules and therefore simulate the dynamics during a time interval (Δt). The objective is to supply the other modules with the aircraft's trajectory, velocities and accelerations during that time interval. In this Chapter two simulations, one to test the integration steppers and another to test the flight dynamics equations, were made.

4.1 Flight Dynamics Model Integrator

The idea behind this program is to define the most important piece of a possible flight simulator, as seen in Figure 1.3. The C++® language was the programming language chosen for this program since it has powerful and useful libraries. The extensibility and readability of this programming language is also an additional reason of this choice.

The flight dynamics integrator program receives structural, aerodynamic, control and propulsion data. During a certain time interval (Δt) chosen by the user, it integrates the dynamic equations assuming these received values are constant. Amid the initial and final time (Δt) defined by the user, the program outputs the integration results of the aircraft dynamics system (first order ordinary differential equations (ODE)).

When the other standalone modules receive information about the aircraft's position, velocity and acceleration, data on the structural, aerodynamics and control will be updated. Afterwards this updated data is given to the flight dynamics integrator, closing the cycle and therefore forming a complete aeroservoelastic simulation.

This flight dynamics integrator gives the freedom to test several flight condition possibilities. If this program was to be used by a flight control engineer, he would for example, ignore the structural data and consider the aerodynamic inputs constant throughout the simulation and therefore concentrate on the importance of the control variables. The same applies if this program was to be used by an aeroelastic

engineer, that would focus on the aerodynamic and structural part of the flight dynamics integrator.

4.1.1 Inputs and Outputs

Inputs

In order for the program to work it needs several key variables as inputs from other interconnected modules. These are seen in Table 4.1.

Type	Symbol	Description
General parameters	u_0, α_0, θ_0, c, S, ρ, m, b	Trim condition variables.
Aerodynamics	C_{D_u}, C_{D_0}, C_{D_α}	Drag coefficient derivatives .
	C_{L_0}, C_{L_α}, C_{L_u}, C_{L_q}, $C_{L_{\delta_E}}$	Lift coefficient derivatives .
	C_{m_u}, C_{m_α}, C_{m_q}, $C_{m_{\dot{\alpha}}}$,$C_{m_{\delta_E}}$	Pitching moment coefficient derivatives.
	C_{Y_β}, C_{Y_p}, C_{Y_r}, $C_{Y_{\delta_A}}$, $C_{Y_{\delta_R}}$	Sideslip angle stability derivatives.
	C_{l_β}, C_{l_p}, C_{l_r}, $C_{l_{\delta_A}}$, $C_{l_{\delta_R}}$	Rolling moment coefficient derivatives.
	C_{n_β}, C_{n_p}, C_{n_r},,$C_{n_{\delta_A}}$,$C_{n_{\delta_R}}$	Yaw moment coefficient derivatives.
Structures	I_{xx},I_{yy},I_{zz},I_{xy},I_{xz},I_{yz}	Moments of inertia in all axis .
	μ_{i_j}, η_{i_j}, ξ_i, ω_i	Asymmetric and symmetric vibration modes information and force derivatives.
Control	δ_E, δ_A, δ_R, δ_{T_i}	Control surfaces deflections (elevator, ailerons and rudder, respectively) and each engine throttle.
Engine Model.	x_{e_i},y_{e_i},z_{e_i}, $X_{\delta_{T_i}}$,$Z_{\delta_{T_i}}$	Engine positions and each engine thrust force derivatives for longitudinal and vertical axis (X,Z).

Table 4.1: Inputs for the flight dynamics integrator program

Outputs

The program outputs the integrated state variables that are seen in Table 4.2.

Output Variables	Description
(x_E, y_E, z_E)	Positions of the aircraft
(u_B, v_B, w_B)	Velocities of the aircraft
$(a_{x_B}, a_{y_B}, a_{z_B})$	Accelerations of the aircraft
(ϕ,ψ,θ)	Euler angles (roll,yaw,pitch)
(p, q, r)	Angular rates

Table 4.2: Outputs of the flight dynamics integrator program

4.1.2 Dynamic Equations

The structural problem needs to be tackled in some particular cases in which the data is gathered by a specialized structural simulation program, thus the simulations of Chapters 4, 6 and 7 will not include vibration modes as state variables. The dynamic equations used in the flight dynamics integrator are (3.56) for longitudinal motion and (3.57) for lateral motion. Nevertheless it is possible to augment flight dynamics equations to include more states variables, such as structural vibration modes, just as in (3.64) and (3.65).

4.1.3 Integration

The center piece of the program is its integration function. This integration function comes from the Boost© C++® library, an open-source extensive used library that provides a wide range of platform agnostic functionality that STL (Standard Template Library) missed [26]. The version of Boost© used was the 1.61.

```
integrate(system, x0, t0, t1, dt, observer)
```

Listing 4.1: Boost© $integrate$ function

The integration function of Listing 4.1 is the common integration function of Boost©. It performs the time evolution, for each time step dt, of the ordinary differential system from some starting time t_0 to a given end time t_1 and a starting state x_0.

In the used library, there are five types of integration function, although only two are of the interest of this project's flight dynamics integrator.

```
integrate_const(stepper, system, x0, t0, t1, dt, observer)
```

Listing 4.2: Boost© $integrate\_const$ function

Almost the same as the function in Listing 4.1, but this function has an additional argument, $stepper$, that is nothing more than the mathematical method used during the integration. This function also has the benefit of calling the $observer$ at equidistant times separated by dt.

```
integrate_adaptive(stepper, system, x0, t0, t1, dt, observer)
```

Listing 4.3: Boost© $integrate\_adaptive$ function

Exactly the same of the integration function in Listing 4.2, however it does not necessarily call the observer at equidistant times. It is basically an upgrade to the function described in Listing 4.2, opening the door to controlled steppers which do not act in constant time steps.

4.1.4 Type of Steppers

Ordinary differential equations (ODEs) are usually solved iteratively, that is for a given state of an ordinary differential equation is iterated forward ($x(t) \rightarrow x(t+dt) \rightarrow x(t+2dt)$). The steppers in $odeint$ library, an included library in the Boost© library for numerically solving Ordinary Differential Equations, are responsible for performing each time step (dt) integration. [26]

```
stepper.do_step(system , in , t , out , dt)
```

Listing 4.4: do_step function

In Listing 4.4, there is the function executed by the integration that performs the step. The first parameter is the system function - a function describing the ODE. The second argument is the state of the ODE at time t, the third argument is t, the fourth argument is the approximate solution at time $t + dt$ which is filled by do_step and the fifth argument is the time step, dt. The stepper before the function call is relative to the type of stepper used during the do_step function call. [26]

The stepper is the mathematical model used during the step integration. There are plenty of steppers and each one has its purpose and use. For this type of ODE problem there are three kinds of steppers:

- **Basic steppers**: As the name enunciates, these are the normal steppers. Some of them are $euler$ and $runge\_kutta\_cash\_karp54$;
- **Error steppers**: Steppers that provide an error estimation. This kind of steppers are needed to construct the controlled steppers. Besides also being a basic stepper, $runge\_kutta\_cash\_karp54$ provides also an error estimation;
- **Controlled steppers**: Built on error steppers, this kind of stepper may decide to modify the integration time step if an error criteria finds the suggested time step inadequate. As $runge\_kutta\_cash\_karp54$ will be used in the tests, $controlled\_runge\_kutta$ will be considered, since it is a controlled stepper of Runge Kutta.

Each of the integration steppers announced are exposed with more details in Table 4.3 and will be then experimented in order to find the most appropriate stepper to use in the aircraft dynamics integration. Alongside with these steppers, the usual stepper used for ODEs in SIMULINK® will also be considered as a tool of comparison. Any complementary information about the mathematical schemes used can be found in [27].

In the standard integration solver used in SIMULINK®, the Dormand-Price, the step size varies dynamically based on the local error, just as controlled steppers. Although it is possible to change the time

step in SIMULINK® by switching in the Model Configuration Parameters, $VariableStep$ to $FixedStep$ and determining a value for the time step.

	Algorithm	Stepper Function	Order	Error Estimation	Remarks
Boost© C++® Stepper	Euler	$euler$	1	No	Very Simple. Used only for demonstration purposes.
	Cash-Karp	$runge\_kutta\_cash\_karp54$	5	Yes, 4th order	General scheme used in most applications. Has the peculiarity of also being an error stepper.
	Controlled Runge-Kutta	$controlled\_runge\_kutta$	depends	Yes	Requires an error stepper to work. Changes time steps.
SIMULINK® Solver	Dormand-Prince	-	5	Yes, 4th order	Deviation of the Runge Kutta (4,5). It is the auto solver for SIMULINK®

Table 4.3: Boost© and SIMULINK® Types of steppers

4.2 Stepper Comparison

The SIMULINK® integration method for the results shown in Figure 4.1 are with $VariableStep$ dynamic size, with a 0.2 seconds maximum time step. For the stepper comparison test, trim condition stability derivatives data from the three engine Dassault Falcon 7X present in Appendix A.3.3, was used.

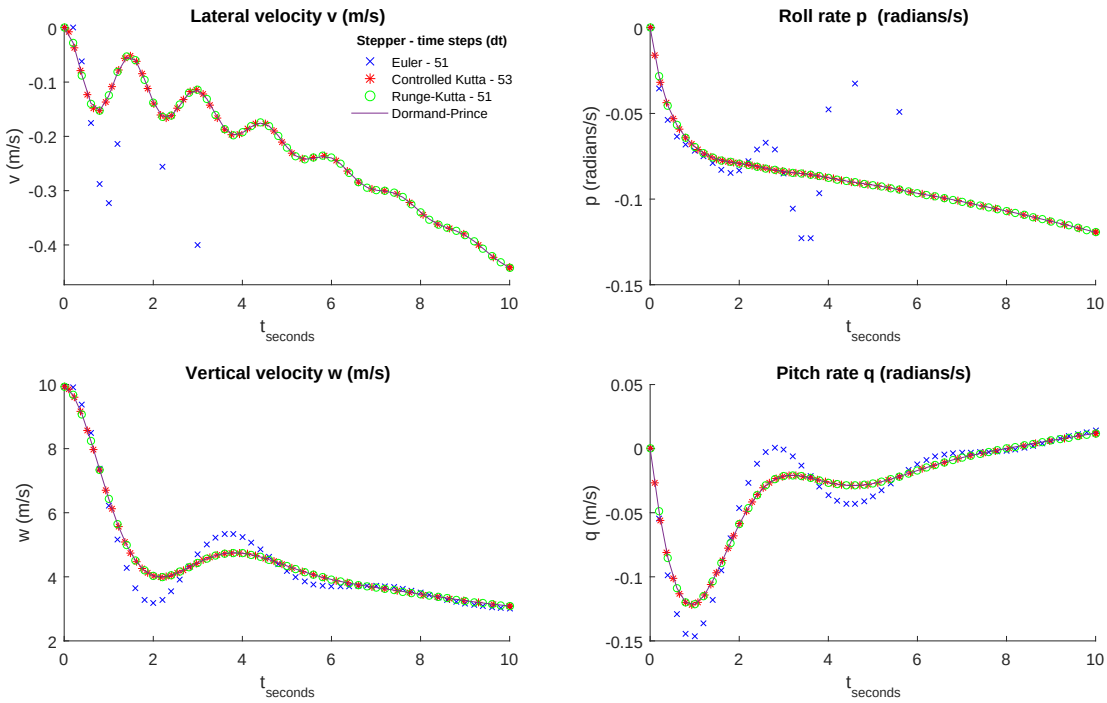

Figure 4.1: Dynamic responses of the Dassault Falcon 7X (v,p,w,q)

In Figure 4.1, it is exposed the integration results of the Dassault Falcon 7X in a certain flight condition (Dassault Falcon 7X δ_A = 5º, δ_R = 0º, δ_E = 5º, no engine throttle and null initial state conditions (x_0=0)). The time of the integration is ten seconds and the time step (dt) for Euler and Runge-Kutta is 0.2 seconds. The plan is to spot the differences between integration steppers and eventually choose one for the flight dynamics integration problem.

The Euler method possesses significant accuracy problems as it only corresponds to the two first terms in the Taylor series, that are visible to see in both w and q plots. As its error propagation grows with the number of time steps and their size, it can become divergent in some cases as in the v and p plots. Euler method is the simplest integration method and in most cases it is only used for learning purposes. It was included in the demonstration of the flight dynamics in order to show that a higher order stepper is needed to have better results. [27]

The 4th order Runge-Kutta integration scheme shows suitable results for the dynamics of this flight condition. Although if the system has certain initial conditions or states for the control variables, it can overshoot. In those cases, if time step of the basic stepper is large, it might have accuracy problems even being a 4th order stepper. As such, a controlled stepper is also considered.

The controlled Kutta stepper internally varies the time step size, adapting the integration to the prob-

lem in question. By doing that, it decreases significantly the integration accuracy errors. On the plots of Figure 4.1, Euler and basic Runge-Kutta steppers have done 51 steps during the integration,

$$N_{steps} = \frac{t_{integration}}{dt} + 1 \Rightarrow \frac{10}{0.2} + 1 = 51\,, \tag{4.1}$$

where plus 1, accounts for the first step at 0 seconds. However, the number of steps the controlled stepper utilized was 53 for all tested variables. As a result of the experimented variables overshooting (q) or having high variations in small time intervals (v), the controlled stepper finds the necessity of decreasing slightly the time step. Appropriately and autonomously it decreases the time step and therefore increases the number of steps, for better results.

All results point to the controlled stepper as being the integration method to choose, however it has a setback to take into account. The considerable setback is that the user has no power in the choice of the controlled stepper time step. As the flight dynamics integrator goal is to interconnect to other software infrastructures, it needs to have a well defined time step in order to synchronize correctly.

In the light of these results, the flight dynamics integrator default stepper is the Runge-Kutta stepper of 4th order. Nevertheless with the usage of the $integrate\_adaptive$ function, the option of using a controlled stepper or another basic stepper is open to the user. The need of a controlled stepper might appear, especially, for aircraft that have overshooting responses.

4.3 Aircraft Dynamics

On the aircraft dynamic analysis, trim condition stability derivatives data, from the twinned engine ($N_{eng} = 2$) Embraer E120 present in Appendix A.3.3, was used.

In Figure 4.4 it is possible to see two different dynamic responses. In both cases the integration time was 10 seconds with a time step of 0.2 seconds and each point corresponds to the integrated state variables for both, longitudinal and lateral modes, on each time step. In order to ease visual interpretation, the results were linearly interpolated.

For both simulations, the initial state conditions (x_0) were null and the control surface deflections were

$$\delta_E = 5^{\text{o}}, \quad \delta_R = 0^{\text{o}}, \quad \delta_A = 0^{\text{o}}\,. \tag{4.2}$$

It is visible in Figure 4.4 that each plot corresponds to a variation in time of a state variable (longitudinal state variables on the left column and lateral state variables on the right column) and that each state variable plot has two sets of points. The blue asterisk curves match to state variables dynamics responses where both engines have the trim condition throttle ($\delta_{T_1} = \delta_{T_2} = \delta_{T_0}$, which in this case is 0.2 or 20%), whereas the red circle curves are also dynamic responses, but with the right engine turned off ($\delta_{T_1} = \delta_{T_0}$, $\delta_{T_2} = 0$). Angular sense for the control surfaces and engine nomenclature is explained in the Section 3.2.2.

For the curves that possess equal trimmed engine throttle (δ_{T_0}), dynamic responses of both modes are consistent. As ailerons and rudder deflections are null ($\delta_A = 0$, $\delta_R = 0$), there is no change in lateral

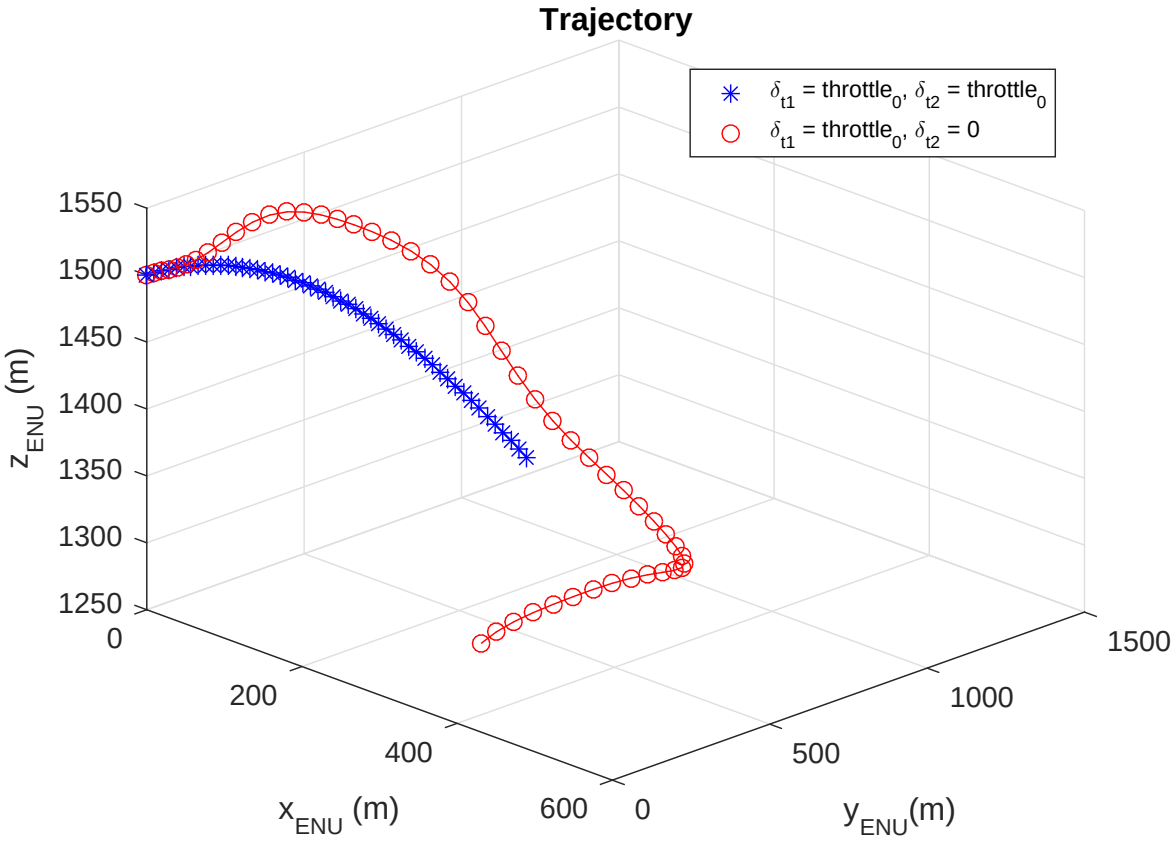

Figure 4.2: Embraer E120 dual flight condition trajectory seen in East-North-Up fixed frame

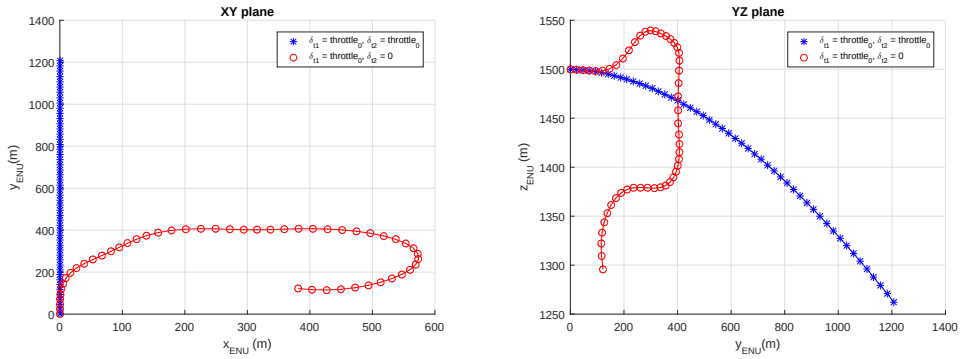

Figure 4.3: Plane XY and YZ cuts from Figure 4.2

state variables. As for longitudinal state variables, the aircraft starts gaining longitudinal speed due to the fact that it is in descent. The descent is the result of the positive deflection on the elevator (δ_E = 5º). The pitch angle (θ), vertical speed (w), the pitch rate (q), seem to be highly dependent on the deflection elevator, and thus have elevated negative values after integration is over.

Considering now the curves of the unequal engine thrust, where an engine is turned off and the other is operating with trimmed throttle ($\delta_{T_1} = \delta_{T_0}$). The longitudinal state variables dynamic responses are adequate, slightly lower especially on the longitudinal speed (u) due to less produced total thrust

by engines. Pitch angle (θ), pitch rate (q) and vertical speed (w) dynamic responses are similar to the case of the equal trimmed engine. These have slightly deviations due to the longitudinal velocity not increasing so fast and having less engine throttle.

The lateral states variables dynamic responses are the purpose of this simulation, since having an engine shut off completely unbalances the state variables. The shut engine influences directly the yaw angular speed (r) by creating a clockwise moment and therefore also highly affecting the other states of the lateral mode (β,p,ϕ). As it is possible to see in Figure 4.4, the proportion in which, for example, the yaw rate (r) and the roll rate (p), are increasing is not viable and unrelated to the results obtained for the longitudinal mode. After all, the mathematical model is still decoupled in the longitudinal and lateral state variables or, in another words, the longitudinal state variables do not affect lateral state variables and vice-versa. From the aircraft's trajectory in Figure 4.2, and the respective YZ and XY plane cuts in Figure 4.3, it is clearly visible the influence of having an engine which provides no thrust on during the simulation.

For this pure mathematical demonstration case, the integration time (Δt) is too large and for better physical results it would need several aerodynamic coefficient updates, in order to refresh the stability derivatives of the dynamic equations. For a better study of the dynamics in an engine loss situation, a non-linear flight dynamics model is preferred since the model utilized above it is for small disturbances. East-North-Up was the reference frame used in Figure 4.2, transformation from the body axis system to this reference frame is in A.1.1.

The need for a controller to send suitable control surface deflections and engine thrust throttle is well presented in the problem of the lateral mode above. With correct deflections it is possible to maintain an aircraft's stability even when there are discrepancies between engines throttles.

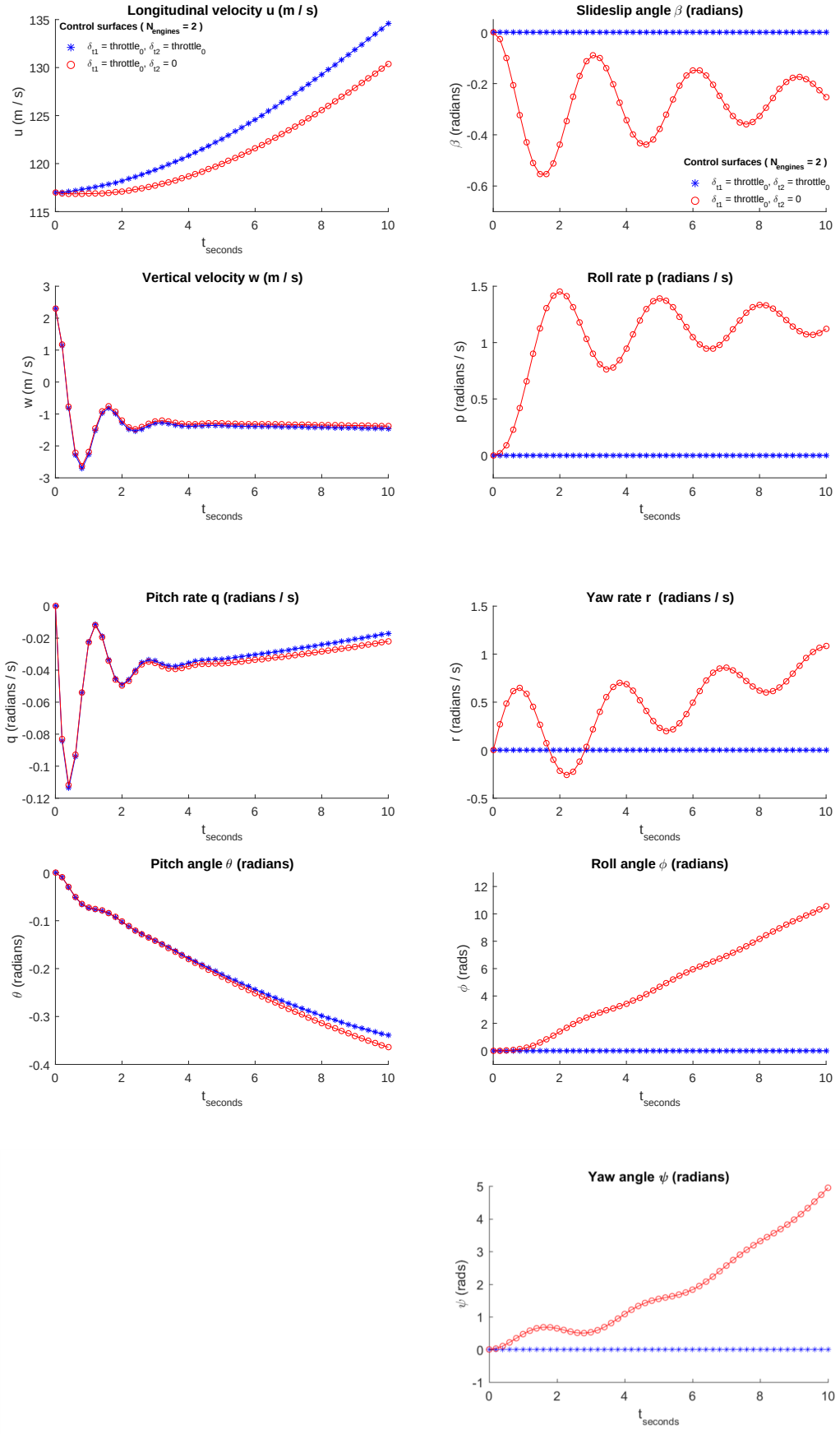

Figure 4.4: Longitudinal (left column) and Lateral (right column) dynamic responses of Embraer E120 ($N_{eng} = 2$, $u_0 = 116.98$ m/s $w_0 = 2.31$ m/s)

Chapter 5

Flight Control and Simulation domain

In this Chapter the requirements for flight control are introduced, disturbances and the desired states for control are added into the state-space form. The final model to be used in the simulation is given. Some of the SIMULINK® modules used, such as the state-space, sensors and actuators are introduced.

5.1 Simulation Domain

The flight controller of this project will be represented in state-space form,

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{cases} . \tag{5.1}$$

The first equation in (5.1) is the state equation. This equation is a first order, vector differential equation, where the $\boldsymbol{x}$ represents the state vector, $\boldsymbol{u}$ the control vector, $\boldsymbol{A}$ the state coefficient matrix and $\boldsymbol{B}$ the driving matrix. The second equation in (5.1) is the output equation, which is merely an algebraic equation that solely depends upon the state vector. Where $\boldsymbol{y}$ is the output vector, and the matrices $\boldsymbol{C}$ and $\boldsymbol{D}$ the output and direct matrix respectively [21]. The stability of the system is verified by looking at the eigenvalues of the state coefficient matrix $\boldsymbol{A}$. If these eigenvalues have negative real part, then it is safe to say that the system, $\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x}(\boldsymbol{t})$ is asymptotically stable.

5.2 Sensors and Actuators

To ensure control of a flying aircraft, there are three main components to consider. First, the sensors which obtain the important flight data and then, through control laws, the automatic flight controller (AFC) sends signals to the actuators with the needed variations on the control surfaces.

Sensors and actuators have an important role in aerospace. They are partly responsible for the growing in size and especially autonomy of aircraft. Sensors are needed to measure unknown signals and parameters of an engineering system and its environment. This knowledge will then be useful to

control and monitor the system in question [17]. Actuators receive the control signal, which usually is a low energy, and then convert this energy into mechanic motion.

In this project, as the flight dynamics integrator and flight controller are designed for dynamics analysis of a general aircraft, for now, realistic models for sensors were not implemented and therefore considered ideal. On the other hand as seen in Figure 5.1, realistic models for actuators, which are

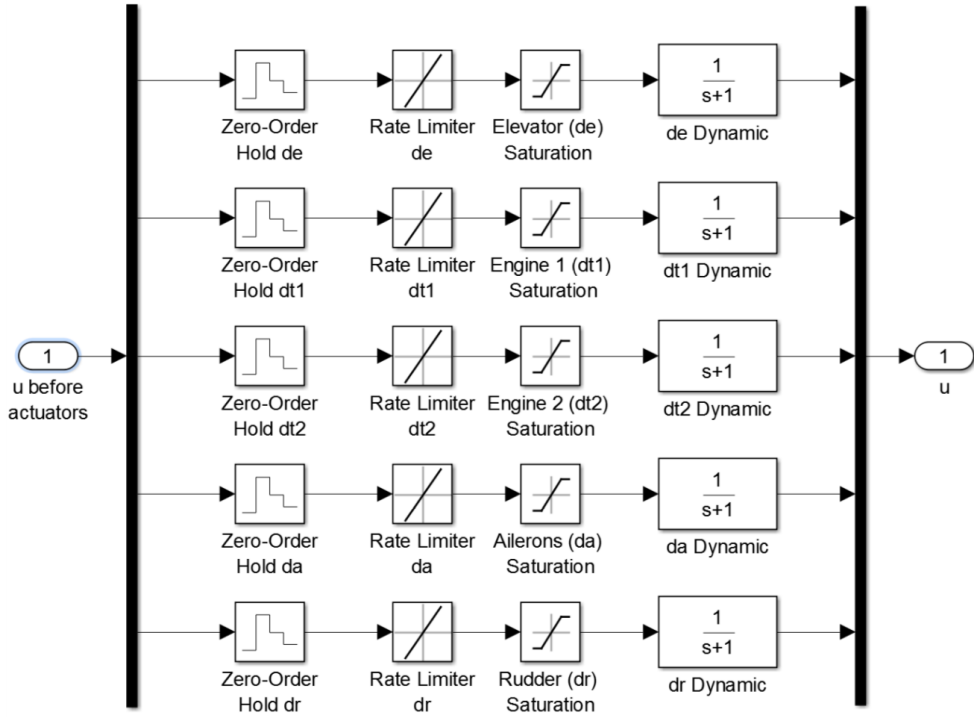

Figure 5.1: Control surfaces actuators model implemented in SIMULINK® for an aircraft with two engines ($N_{eng} = 2$)

the mechanism needed to perform the control action, were implemented. The model is composed of 4 blocks. The first block defines the actuator's frequency, while the second block determines the maximum and minimum allowed velocities to the actuators movement. The third block sets the maximum and minimum deflection the actuator can have, and finally the fourth block is the dynamics block which defines the linear model of the actuator's movement. The saturation parameters, which are considered to be essential in this dissertation, are contained in the '$.mat$' file containing the aircraft trim condition data. Other parameters as the actuator's frequency and movement linear model will be set with a default value.

5.3 State-space Systems

For purposes of control, the longitudinal and lateral equations demonstrated in Chapter 3 need to be in the state-space form.

5.3.1 Longitudinal State-space Equations

The longitudinal set of first order equations from the system (3.56) can be presented in the state-space form, equation (5.1)

$$\dot{\boldsymbol{x}}_{\boldsymbol{Long}} = \boldsymbol{A}_{\boldsymbol{Long}}\boldsymbol{x}_{\boldsymbol{Long}} + \boldsymbol{B}_{\boldsymbol{Long}}\boldsymbol{u}_{\boldsymbol{Long}} =$$

$$\begin{bmatrix} X_u & X_w & -w_0 & -gcos(\theta_0) \\ Z_u & Z_w & u_0 & -gsin(\theta_0) \\ \tilde{M}_u & \tilde{M}_w & \tilde{M}_q & \tilde{M}_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \boldsymbol{x}_{\boldsymbol{Long}} + \begin{bmatrix} X_{\delta_E} & X_{\delta_{T_1}} & ... & X_{\delta_{T_{N_{eng}}}} \\ Z_{\delta_E} & Z_{\delta_{T_1}} & ... & Z_{\delta_{T_{N_{eng}}}} \\ \tilde{M}_{\delta_E} & (z_1 X_{\delta_{T_1}} - x_1 Z_{\delta_{T_1}}) & ... & (z_{N_{eng}} X_{\delta_{T_{N_{eng}}}} - x_{N_{eng}} Z_{\delta_{T_{N_{eng}}}}) \\ 0 & 0 & ... & 0 \end{bmatrix} \boldsymbol{u}_{\boldsymbol{Long}}\,, \tag{5.2}$$

The state and input vector ($\boldsymbol{x}$ and $\boldsymbol{u}$, respectively) are

$$\boldsymbol{x}_{\boldsymbol{Long}} = \begin{bmatrix} u & w & q & \theta \end{bmatrix}^T\,, \tag{5.3}$$

$$\boldsymbol{u}_{\boldsymbol{Long}} = \begin{bmatrix} \delta_E & \delta_{T1} & ... & \delta_{TN_{eng}} \end{bmatrix}^T\,. \tag{5.4}$$

5.3.2 Lateral State-space Equations

The lateral set of first order equations from the system (3.57) can be presented in the state-space form, equation (5.1)

$$\dot{\boldsymbol{x}}_{\boldsymbol{Lat}} = \boldsymbol{A}_{\boldsymbol{Lat}}\boldsymbol{x}_{\boldsymbol{Lat}} + \boldsymbol{B}_{\boldsymbol{Lat}}\boldsymbol{u}_{\boldsymbol{Lat}} =$$

$$\begin{bmatrix} Y_\beta & \frac{Y_p}{u_0} + \alpha_0 & \frac{Y_r}{u_0} - 1 & \frac{gcos(\theta_0)}{u_0} & 0 \\ L'_\beta & L'_p & L'_r & 0 & 0 \\ N'_\beta & N'_p & N'_r & 0 & 0 \\ 0 & 1 & tan(\theta_0) & 0 & 0 \\ 0 & 0 & \frac{1}{cos(\theta_0)} & 0 & 0 \end{bmatrix} \boldsymbol{x}_{\boldsymbol{Lat}} + \begin{bmatrix} \frac{Y_{\delta_A}}{u_0} & 0 & ... & 0 & \frac{Y_{\delta_R}}{u_0} \\ L'_{\delta_A} & -y_1 Z_{\delta_{T_1}} & ... & -y_{N_{eng}} Z_{\delta_{T_{N_{eng}}}} & L'_{\delta_R} \\ N'_{\delta_A} & y_1 X_{\delta_{T_1}} & ... & y_{N_{eng}} X_{\delta_{T_{N_{eng}}}} & N'_{\delta_R} \\ 0 & 0 & ... & 0 & 0 \\ 0 & 0 & ... & 0 & 0 \end{bmatrix} \boldsymbol{u}_{\boldsymbol{Lat}}\,, \tag{5.5}$$

The state and input vector ($\boldsymbol{x}$ and $\boldsymbol{u}$, respectively) are

$$\boldsymbol{x}_{\boldsymbol{Lat}} = \begin{bmatrix} \beta & p & r & \phi & \psi \end{bmatrix}^T\,, \tag{5.6}$$

$$\boldsymbol{u}_{\boldsymbol{Lat}} = \begin{bmatrix} \delta_A & \delta_{T1} & ... & \delta_{TN_{eng}} & \delta_R \end{bmatrix}^T\,. \tag{5.7}$$

5.4 Variable Control

The flight control will be done in both longitudinal and lateral modes.

5.4.1 Coupled Motion

As the systems (5.2) and (5.5) are coupled by single engine thrust contribution, these systems need to be merged into a single state-space form as shown in equation (5.8)

$$\begin{bmatrix} \dot{\boldsymbol{x}}_{\boldsymbol{Long}} \\ \dot{\boldsymbol{x}}_{\boldsymbol{Lat}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{\boldsymbol{Long}} & 0 \\ 0 & \boldsymbol{A}_{\boldsymbol{Lat}} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{Long}} \\ \boldsymbol{x}_{\boldsymbol{Lat}} \end{bmatrix} + \boldsymbol{B}_{\boldsymbol{ini}} \begin{bmatrix} \boldsymbol{u}_{\boldsymbol{coupled}} \end{bmatrix} , \tag{5.8}$$

where

$$\boldsymbol{B}_{\boldsymbol{ini}} = \begin{bmatrix} X_{\delta_E} & X_{\delta_{T_1}} & ... & X_{\delta_{T_{N_{eng}}}} & 0 & 0 \\ Z_{\delta_E} & Z_{\delta_{T_1}} & ... & Z_{\delta_{T_{N_{eng}}}} & 0 & 0 \\ \tilde{M_{\delta_E}} & (z_1 X_{\delta_{T_1}} - x_1 Z_{\delta_{T_1}}) & ... & (z_{N_{eng}} X_{\delta_{T_{N_{eng}}}} - x_{N_{eng}} Z_{\delta_{T_{N_{eng}}}}) & 0 & 0 \\ \\ 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & ... & 0 & \frac{Y_{\delta_A}}{u_0} & \frac{Y_{\delta_R}}{u_0} \\ 0 & -y_1 Z_{\delta_{T_1}} & ... & -y_{N_{eng}} Z_{\delta_{T_{N_{eng}}}} & L'_{\delta_A} & L'_{\delta_R} \\ 0 & y_1 X_{\delta_{T_1}} & ... & y_{N_{eng}} X_{\delta_{T_{N_{eng}}}} & N'_{\delta_A} & N'_{\delta_R} \\ 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & ... & 0 & 0 & 0 \end{bmatrix} , \tag{5.9}$$

$$\boldsymbol{u}_{\boldsymbol{coupled}} = \begin{bmatrix} \delta_E & \delta_{T1} & ... & \delta_{T_{N_{eng}}} & \delta_A & \delta_R \end{bmatrix} . \tag{5.10}$$

The $\boldsymbol{A}$ matrix is purely a diagonal matrix composed of the $\boldsymbol{A}$ matrices from state-space forms (5.2) and (5.5).

The longitudinal mode and lateral modes control variables will be demonstrated in a decoupled way in order to ease the demonstration. In Section 6.2 all schemes and changes in the modes will be gathered to form a final engine coupled state-space form.

5.4.2 Longitudinal mode

For the lateral mode the control will be done in the flight path angle (γ) and longitudinal speed (u).

Flight path angle and longitudinal speed

For the flight path angle control (γ) needs to be added as a state. Keeping in mind that the vertical velocity (w) can be approximated, for small perturbations, as a function of the angle of attack (α), the pitch angle (θ) is substituted for flight path (γ)in system (5.2)

$$\gamma = \theta - \alpha \rightarrow \theta = \gamma + \frac{w}{u_0} . \tag{5.11}$$

The longitudinal space state form transforms now to

$$\dot{\boldsymbol{x}}_{\boldsymbol{Long_\gamma}} = \boldsymbol{A}_{\boldsymbol{Long_\gamma}}\boldsymbol{x}_{\boldsymbol{Long_\gamma}} + \boldsymbol{B}_{\boldsymbol{Long_\gamma}}\boldsymbol{u}_{\boldsymbol{Long}} =$$

$$\begin{bmatrix} X_u & X_w - \frac{gcos(\theta_0)}{u_0} & -w_0 & -gcos(\theta_0) \\ Z_u & Z_w - \frac{gsin(\theta_0)}{u_0} & u_0 & -gsin(\theta_0) \\ \tilde{M}_u & \tilde{M}_w + \frac{\tilde{M}_\theta}{u_0} & \tilde{M}_q & \tilde{M}_\theta \\ -\frac{Z_u}{u_0} & -\frac{Z_w}{u_0} + \frac{gsin(\theta_0)}{u_0^2} & 0 & \frac{gsin(\theta_0)}{u_0} \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \gamma \end{bmatrix}$$

$$+ \begin{bmatrix} X_{\delta_E} & X_{\delta_{T_1}} & ... & X_{\delta_{T_{N_{eng}}}} \\ Z_{\delta_E} & Z_{\delta_{T_1}} & ... & Z_{\delta_{T_{N_{eng}}}} \\ \tilde{M}_{\delta_E} & (z_1 X_{\delta_{T_1}} - x_1 Z_{\delta_{T_1}}) & ... & (z_{N_{eng}} X_{\delta_{T_{N_{eng}}}} - x_{N_{eng}} Z_{\delta_{T_{N_{eng}}}}) \\ \frac{-Z_{\delta_E}}{u_0} & \frac{-Z_{\delta_{T_1}}}{u_0} & ... & \frac{-Z_{\delta_{T_{N_{eng}}}}}{u_0} \end{bmatrix} \boldsymbol{u}_{\boldsymbol{Long}}\,. \tag{5.12}$$

The model in which, longitudinal speed will be controlled is the (5.12), because longitudinal speed is already an incorporated state in that state-space form.

Final Longitudinal Model

With the expectation of preventing static error on the longitudinal controllable states, two new integrator states are added.

$$x_u = \int u \quad dt\,, \qquad x_\gamma = \int \gamma \quad dt\,. \tag{5.13}$$

These new states, x_u and x_γ, are respectively integrator states of the longitudinal velocity (u) and flight path angle (γ). So, the space state (5.12) is now

$$\dot{\boldsymbol{x}}_{\boldsymbol{Long_{u\gamma_s}}} = \boldsymbol{A}_{\boldsymbol{Long_{u\gamma_s}}}\boldsymbol{x}_{\boldsymbol{Long_{u\gamma_s}}} + \boldsymbol{B}_{\boldsymbol{Long_{u\gamma_s}}}\boldsymbol{u}_{\boldsymbol{Long}} =$$

$$\begin{bmatrix} X_u & X_w - \frac{gcos(\theta_0)}{u_0} & -w_0 & -gcos(\theta_0) & 0 & 0 \\ Z_u & Z_w - \frac{gsin(\theta_0)}{u_0} & u_0 & -gsin(\theta_0) & 0 & 0 \\ \tilde{M}_u & \tilde{M}_w + \frac{\tilde{M}_\theta}{u_0} & \tilde{M}_q & \tilde{M}_\theta & 0 & 0 \\ -\frac{Z_u}{u_0} & -\frac{Z_w}{u_0} + \frac{gsin(\theta_0)}{u_0^2} & 0 & \frac{gsin(\theta_0)}{u_0} & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \gamma \\ x_u \\ x_\gamma \end{bmatrix}$$

$$+ \begin{bmatrix} X_{\delta_E} & X_{\delta_{T_1}} & ... & X_{\delta_{T_{N_{eng}}}} \\ Z_{\delta_E} & Z_{\delta_{T_1}} & ... & Z_{\delta_{T_{N_{eng}}}} \\ \tilde{M}_{\delta_E} & (z_1 X_{\delta_{T_1}} - x_1 Z_{\delta_{T_1}}) & ... & (z_{N_{eng}} X_{\delta_{T_{N_{eng}}}} - x_{N_{eng}} Z_{\delta_{T_{N_{eng}}}}) \\ \frac{-Z_{\delta_E}}{u_0} & \frac{-Z_{\delta_{T_1}}}{u_0} & ... & \frac{-Z_{\delta_{T_{N_{eng}}}}}{u_0} \\ 0 & 0 & ... & 0 \\ 0 & 0 & ... & 0 \end{bmatrix} \boldsymbol{u}_{\boldsymbol{Long}}\,. \tag{5.14}$$

5.4.3 Lateral Mode

Finally, for the lateral mode the control will be done in heading angle (λ).

5.4.4 Heading

Only one variable will be controlled in the lateral mode, the heading angle (λ). It can be defined as the sum of slide slip angle (β) with the yaw angle (ψ)

$$\lambda = \beta + \psi \rightarrow \psi = \lambda - \beta\,. \tag{5.15}$$

This will be the fifth lateral state substituting the yaw angle (ψ). The lateral state-space form (5.5) is now

$$\boldsymbol{\dot{x}_{Lat_\lambda}} = \boldsymbol{A_{Lat_\lambda}}\boldsymbol{x_{Lat_\lambda}} + \boldsymbol{B_{Lat_\lambda}}\boldsymbol{u_{Lat}} =$$

$$\begin{bmatrix} Y_\beta & \frac{Y_p}{u_0}+\alpha_0 & \frac{Y_r}{u_0}-1 & \frac{gcos(\theta_0)}{u_0} & 0 \\ L'_v & L'_p & L'_r & 0 & 0 \\ N'_v & N'_p & N'_r & 0 & 0 \\ 0 & 1 & tan(\theta_0) & 0 & 0 \\ Y_\beta & \frac{Y_p}{u_0}+\alpha_0 & \frac{1}{cos(\theta_0)}+\frac{Y_r}{u_0}-1 & \frac{gcos(\theta_0)}{u_0} & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \\ \lambda \end{bmatrix} + \begin{bmatrix} \frac{Y_{\delta_A}}{u_0} & 0 & ... & 0 & \frac{Y_{\delta_R}}{u_0} \\ L'_{\delta_A} & -y_1 Z_{\delta_{T_1}} & ... & -y_{N_{eng}} Z_{\delta_{T N_{eng}}} & L'_{\delta_R} \\ N'_{\delta_A} & y_1 X_{\delta_{T_1}} & ... & y_{N_{eng}} X_{\delta_{T N_{eng}}} & N'_{\delta_R} \\ 0 & 0 & ... & 0 & 0 \\ \frac{Y_{\delta_A}}{u_0} & 0 & ... & 0 & \frac{Y_{\delta_R}}{u_0} \end{bmatrix} \boldsymbol{u_{Lat}}\,. \tag{5.16}$$

As in the longitudinal state-space (5.14), a new integrator state is added in order to prevent static error

$$x_\lambda = \int \lambda \quad dt\,. \tag{5.17}$$

So state-space form (5.16) transforms into

$$\boldsymbol{\dot{x}_{Lat_{\lambda_s}}} = \boldsymbol{A_{Lat_{\lambda_s}}}\boldsymbol{x_{Lat_{\lambda_s}}} + \boldsymbol{B_{Lat_{\lambda_s}}}\boldsymbol{u_{Lat}} =$$

$$\begin{bmatrix} Y_\beta & \frac{Y_p}{u_0}+\alpha_0 & \frac{Y_r}{u_0}-1 & \frac{gcos(\theta_0)}{u_0} & 0 & 0 \\ L'_v & L'_p & L'_r & 0 & 0 & 0 \\ N'_v & N'_p & N'_r & 0 & 0 & 0 \\ 0 & 1 & tan(\theta_0) & 0 & 0 & 0 \\ Y_\beta & \frac{Y_p}{u_0}+\alpha_0 & \frac{1}{cos(\theta_0)}+\frac{Y_r}{u_0}-1 & \frac{gcos(\theta_0)}{u_0} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \\ \lambda \\ x_\lambda \end{bmatrix} + \begin{bmatrix} \frac{Y_{\delta_A}}{u_0} & 0 & ... & 0 & \frac{Y_{\delta_R}}{u_0} \\ L'_{\delta_A} & -y_1 Z_{\delta_{T_1}} & ... & -y_{N_{eng}} Z_{\delta_{T N_{eng}}} & L'_{\delta_R} \\ N'_{\delta_A} & y_1 X_{\delta_{T_1}} & ... & y_{N_{eng}} X_{\delta_{T N_{eng}}} & N'_{\delta_R} \\ 0 & 0 & ... & 0 & 0 \\ \frac{Y_{\delta_A}}{u_0} & 0 & ... & 0 & \frac{Y_{\delta_R}}{u_0} \\ 0 & 0 & ... & 0 & 0 \end{bmatrix} \boldsymbol{u_{Lat}}\,. \tag{5.18}$$

5.4.5 Flying and Handling Qualities

Aircraft flying qualities are defined by a number of parameters in the complex frequency domain. In equation (5.19) there are two of these important parameters, damping ratio (ξ) and undamped natural frequency (ω_n). These parameters depend on the eigenvalue (κ) and having knowledge of these allows a flight control engineer to imagine the nature of the aircraft's response to any command or disturbance [21]. The damping ratio is a dimensionless variable and the natural frequency has its units expressed in

radians per second

$$\begin{aligned} \omega_n &= |\kappa| \\ \xi &= -cos(\angle\kappa) \end{aligned} \tag{5.19}$$

On the other hand handling qualities describe the ease with which a pilot can manage an aircraft with a particular set of flying qualities on a certain mission. It is possible to imagine that the pilot handling qualities not only depend upon flight qualities but also upon flying controls, the display of flight information available in the cockpit and among others.

In this dissertation project it is needed to ensure that for a certain mission, the aircraft has the best flying qualities. The specification used is the MIL-F-8785, Military Specification - Flying Qualities of Piloted Airplanes published in 1980. The level of flying qualities on this specification depend upon the aircraft class and flight phase [28].

From the $\boldsymbol{A}$ matrix of state-space systems defined above for both longitudinal and lateral modes ((5.2) and (5.5)), it is possible to establish knowledge of the dynamic stability through its eigenvalues [21]

$$det[\kappa\boldsymbol{I} - \boldsymbol{A}] = 0 \rightarrow |\kappa\boldsymbol{I} - \boldsymbol{A}| = 0\,. \tag{5.20}$$

The $\boldsymbol{I}$ is an identity matrix of the same size of $\boldsymbol{A}$, the coefficient matrix.

Longitudinal Dynamic Modes

If the eigenvalue equation is applied to the longitudinal coefficient matrix $\boldsymbol{A}$, a fourth degree polynomial eigenvalue emerges

$$\kappa^4 + \kappa^3 a_1 + \kappa^2 a_2 + \kappa a_3 = 0\,. \tag{5.21}$$

In order to test for the longitudinal dynamic stability of an aircraft, its eigenvalues (κ_i) must have negative real parts. If the real part of an eigenvalue is zero or positive, the system's response to a perturbation is unstable due to growing oscillations. The equation (5.21) can be factorized into two quadratic factors such as

$$(\kappa^2 + 2\xi_{phu}\omega_{phu}\kappa + 2\omega_{phu})(\kappa^2 + 2\xi_{sp}\omega_{sp}\kappa + 2\omega_{sp}) = 0\,. \tag{5.22}$$

The first factor is relative to a mode of motion which is characterized by an oscillation of long period. This mode of motion is referred to as the phugoid. It has low frequency (ω_{phu}) and its damping (ξ_{phu}) it is usually low. The second factor is relative to a mode of motion, which is rapid and well-damped. It is nominated as the short period.

Lateral Dynamic Modes

In the same way of the longitudinal motion, by applying the eigenvalue equation (5.20) to the lateral coefficient matrix $\boldsymbol{A}$

$$\kappa^5 + \kappa^4 d_1 + \kappa^3 d_2 + \kappa^2 d_3 + \kappa d_4 = 0\,, \tag{5.23}$$

this stability fifth degree equation can be factorized into

$$\kappa(\kappa+e)(\kappa+f)(\kappa^2+2\xi_{dr}w_{dr}\kappa+2w_{dr})=0\,. \tag{5.24}$$

As it is possible to see, $\kappa=0$ is a root of the characteristic equation and that term corresponds to the heading mode. This means that if the heading is perturbed, there would be no natural tendency for the aircraft to restore its equilibrium heading. By this is said that an aircraft has neutral heading stability.

The quadratic term in the equation (5.24) is denoted as dutch roll. It usually has a low damping ratio (ξ_{dr}) so that its motion is oscillatory.

The term $(\kappa+e)$ is referred to spiral mode, which is usually a very slow motion corresponding to a long term tendency either to maintain the wings level [21].

Finally, the term $(\kappa+f)$ is related to the rolling mode which usually is fast.

Level of Flying Qualities

As seen in Table 5.1, there are three levels of performance in the MIL-F-8785, these levels categorize if the flight qualities are adequate for the mission flight phase,

Level	
1	Flying qualities clearly adequate for the mission flight phase
2	Flying qualities adequate to accomplish the mission flight phase, with some increase in pilot workload or degradation of mission effectiveness
3	Flying qualities such that the aircraft can be controlled safely, but pilot workload is excessive or mission effectiveness is inadequate

Table 5.1: Aircraft level of perfomance [21]

For this project, it is stipulated that any aircraft for a certain mission must have level 1 of performance for any mode. In the Chapter 6 it is explained how this goal is achieved.

5.5 Atmospheric Perturbations

An aircraft flying in the atmosphere behaves as a stochastic system due to the system's noise, external disturbances and control orders. [12]

Control orders either from the manual pilot or from the automatic flight controller (AFC), noise from control elements, like sensors, make the system unpredictable and therefore, it is considered stochastic.

Gust perturbations are the external disturbances included in this project. When the air stream is flowing through the aircraft it's movements affect the aircraft movement. These disturbances are non-deterministic and frequently unpredictable. Wind shears, abrupt unpredictable variations of wind speed through the aircraft, are not having an use in this essay.

5.5.1 Influence on the Model

Considering an aircraft entering a zone where there is wind with a given distribution , that is constant through time but varies in space

$$u_g = u_g(\boldsymbol{x})\,, \tag{5.25}$$

$$v_g = v_g(\boldsymbol{x})\,, \tag{5.26}$$

$$w_g = w_g(\boldsymbol{x})\,. \tag{5.27}$$

The space distribution is transformed through the aircraft displacement in a time distribution

$$\frac{\partial v}{\partial x} = \frac{\frac{\partial v}{\partial t}}{\frac{\partial x}{\partial t}} = \frac{1}{u_0}\frac{\partial v}{\partial t}\,. \tag{5.28}$$

The perturbed motion of the aircraft will result from aerodynamic forces induced by the relative velocity. Now the air speed ($\boldsymbol{v_B}$), defined in Chapter 3 has to take into account the effect of the wind speed ($\boldsymbol{v_W}$)

$$\boldsymbol{v_B} = \boldsymbol{v} - \boldsymbol{v_W} \quad \rightarrow \quad \boldsymbol{v_B} = \begin{bmatrix} u - u_g \\ v - v_g \\ w - w_g \end{bmatrix}\,. \tag{5.29}$$

In the same way, velocity distribution throughout the body of the aircraft also induces an angular velocity distribution

$$\boldsymbol{w_B} = \begin{bmatrix} p - p_g \\ q - q_g \\ r - r_g \end{bmatrix}\,. \tag{5.30}$$

5.5.2 SIMULINK Wind Turbulence Model

The SIMULINK block used to generated wind turbulence is in Figure 5.2. This model generates con-

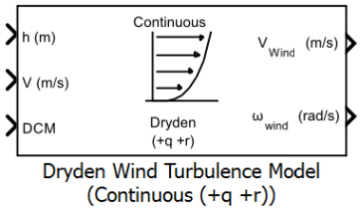

Figure 5.2: SIMULINK Wind Turbulence Model

tinuous wind turbulence with Dryden velocity spectra. The Dryden turbulence model is mathematically simplier than von Kármán model and it is extensively used in the simulation environment. More information about this model is in [29] or in [14]. Inputs for this function are the DCM (NED Direct Cosine Matrix, A.5), aircraft's altitude (m) and the magnitude of the velocity vector in the fixed frame. The function also requires wingspan parameter (b) and allows the user to define the sign of the turbulence angular rates (q,r). The model outputs the disturbance states for lateral and longitudinal motions (equations (5.32) and (5.33)).

5.5.3 Disturbances State-space Form

To include the disturbances described above a new state matrix is added into the aircraft dynamics state-space form (5.1)

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{d}\,, \tag{5.31}$$

where $\boldsymbol{d}$ represents the disturbance states and $\boldsymbol{E}$ the associated disturbance influence matrix. Disturbance states are distributed according to the mode they belong to, for longitudinal

$$\boldsymbol{d_{Long}} = \begin{bmatrix} u_g & w_g & q_g \end{bmatrix}^T\,, \tag{5.32}$$

and for lateral

$$\boldsymbol{d_{Lat}} = \begin{bmatrix} v_g & p_g & r_g \end{bmatrix}^T\,. \tag{5.33}$$

For the coupled matrix the disturbance states will be

$$\boldsymbol{d_{coupled}} = \begin{bmatrix} \boldsymbol{d_{Long}} & \boldsymbol{d_{Lat}} \end{bmatrix}^T = \begin{bmatrix} u_g & w_g & q_g & v_g & p_g & r_g \end{bmatrix}^T\,. \tag{5.34}$$

The final flight controller state-space with gust disturbances will then be

$$\dot{\boldsymbol{x}}_{\boldsymbol{coupled}} = \boldsymbol{A}_{\boldsymbol{coupled}}\boldsymbol{x}_{\boldsymbol{coupled}} + \boldsymbol{B}_{\boldsymbol{coupled}}\boldsymbol{u}_{\boldsymbol{coupled}} + \boldsymbol{E}_{\boldsymbol{coupled}}\boldsymbol{d}_{\boldsymbol{coupled}}\,, \tag{5.35}$$

where

$$\boldsymbol{x_{coupled}} = \begin{bmatrix} \boldsymbol{x_{Long_\gamma}} \\ \boldsymbol{x_{Lat_\lambda}} \end{bmatrix}, \qquad \boldsymbol{A_{coupled}} = \begin{bmatrix} \boldsymbol{A_{Long_\gamma}} & 0 \\ 0 & \boldsymbol{A_{Lat_\lambda}} \end{bmatrix}. \tag{5.36}$$

The $\boldsymbol{B_{coupled}}$ represents now a $9 \times (N_{eng} + 3)$ driving matrix, being basically a variation of $\boldsymbol{B_{ini}}$ that includes the driving's matrix substituted states contribution (ψ for λ and θ for γ)

$$\boldsymbol{B_{coupled}} = \begin{bmatrix} X_{\delta_E} & X_{\delta_{T_1}} & ... & X_{\delta_{T_{N_{eng}}}} & 0 & 0 \\ Z_{\delta_E} & Z_{\delta_{T_1}} & ... & Z_{\delta_{T_{N_{eng}}}} & 0 & 0 \\ \tilde{M}_{\delta_E} & (z_1 X_{\delta_{T_1}} - x_1 Z_{\delta_{T_1}}) & ... & (z_{N_{eng}} X_{\delta_{T_{N_{eng}}}} - x_{N_{eng}} Z_{\delta_{T_{N_{eng}}}}) & 0 & 0 \\ \frac{-Z_{\delta_E}}{u_0} & \frac{-Z_{\delta_{T_1}}}{u_0} & ... & \frac{-Z_{\delta_{T_{N_{eng}}}}}{u_0} & 0 & 0 \\ 0 & 0 & ... & 0 & \frac{Y_{\delta_A}}{u_0} & \frac{Y_{\delta_R}}{u_0} \\ 0 & -y_1 Z_{\delta_{T_1}} & ... & -y_{N_{eng}} Z_{\delta_{T_{N_{eng}}}} & L'_{\delta_A} & L'_{\delta_R} \\ 0 & y_1 X_{\delta_{T_1}} & ... & y_{N_{eng}} X_{\delta_{T_{N_{eng}}}} & N'_{\delta_A} & N'_{\delta_R} \\ 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & ... & 0 & \frac{Y_{\delta_A}}{u_0} & \frac{Y_{\delta_R}}{u_0} \end{bmatrix}. \tag{5.37}$$

The associated disturbance influence 9×6 matrix for coupled motion, $\boldsymbol{E_{coupled}}$, is

$$\boldsymbol{E_{coupled}} = \begin{bmatrix} -X_u & -X_w & -X_q & 0 & 0 & 0 \\ -Z_u & -Z_w & -Z_q & 0 & 0 & 0 \\ -\tilde{M}_u & -\tilde{M}_w & -M_{\dot{w}} Z_q & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -Y_\beta & -Y_p & -Y_r \\ 0 & 0 & 0 & -L'_\beta & -L'_p & -L'_r \\ 0 & 0 & 0 & -N'_\beta & -N'_p & -N'_r \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \tag{5.38}$$

5.6 SIMULINK State-Space model

The common state-space SIMULINK model could not be used for this project because it does not include the associated disturbance influence matrix, $\boldsymbol{E}$. The model in Figure 5.3, satisfies the state-space equation (5.31). The matrices in Figure 5.3 are the matrices representing the coupled state-space in

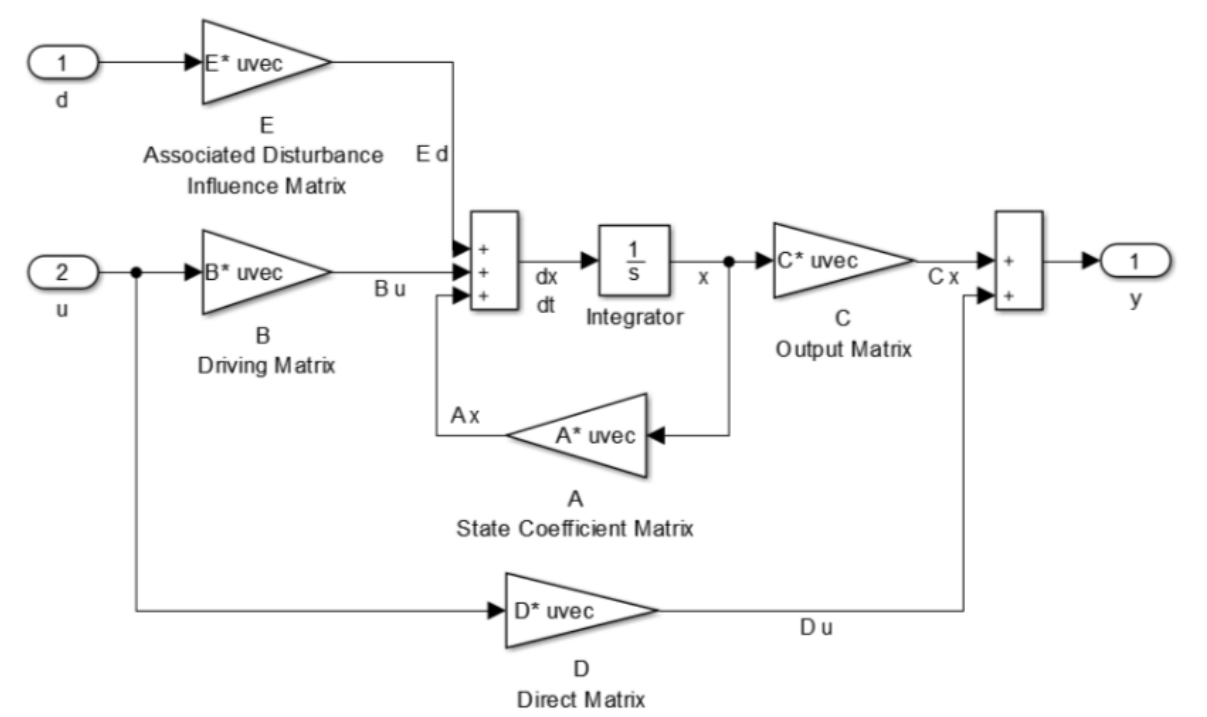

Figure 5.3: State-Space model implemented in SIMULINK

equation 5.35. The output matrix, $\boldsymbol{C}$, is an identity matrix of the same size as the state coefficient matrix, $\boldsymbol{A}$, and the direct matrix, $\boldsymbol{D}$, is a zero matrix of the same size as $\boldsymbol{B}$.

Chapter 6

Optimal Control and Computational Implementation

In this chapter, the optimal control technique used for the flight controller will be introduced by using an example of flutter suppression on a two-dimensional aeroelastic airfoil. Then, the approach in which LQR is going to be utilized in the flight controller is explained. The flight controller module in SIMULINK® is demonstrated. At the end of this chapter two important functions that evaluate the level of flying qualities, are interpreted. The process of choosing an appropriate optimal gain matrix for a general aircraft to be used in the flight simulation is also described.

6.1 Optimal Control Technique - Linear Quadratic Regulator

Since the flight control problem will be addressed as a linear quadratic problem, an aeroelastic plant with a LQR (Linear Quadratic Regulator) will be studied. With this study, LQR will be presented and ready for use on the final flight control problem.

6.1.1 Aeroservoelastic Optimal Control

To introduce the linear quadratic regulator (LQR), a flutter suppression controller of a two-dimensional aeroelastic airfoil represented in Figure 6.1, will be demonstrated.

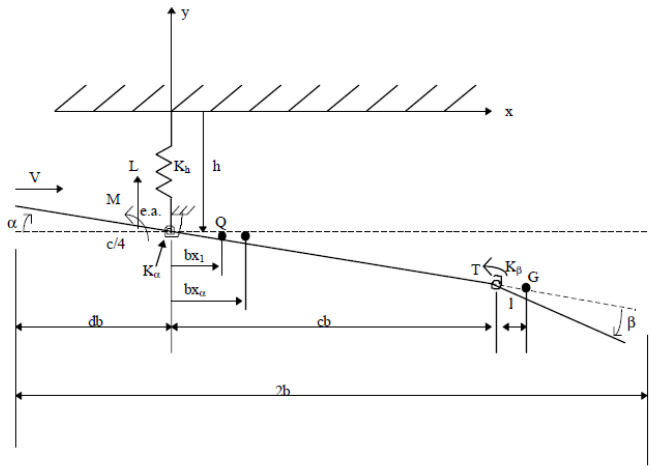

Figure 6.1: The 2-D cross-section of a airfoil [30]

Though there's a problem concerning the two-dimensional aeroelastic airfoil equations shown in Chapter 2. The fundamental equations of this problem are second order, and state-space form first equation is a first order equation. The mathematical transformation and demonstration of this model is explained in [30]. The final state space first equation of this model is

$$\dot{\boldsymbol{x}} = \begin{bmatrix} \boldsymbol{A_{11}} & \boldsymbol{A_{12}} & \boldsymbol{A_{13}} \\ \boldsymbol{A_{21}} & \boldsymbol{A_{22}} & \boldsymbol{A_{23}} \\ \boldsymbol{A_{31}} & \boldsymbol{A_{32}} & \boldsymbol{A_{33}} \end{bmatrix} \begin{bmatrix} \dot{\boldsymbol{Y}}(t) \\ \boldsymbol{Y}(t) \\ \boldsymbol{x_A}(t) \end{bmatrix} + \boldsymbol{B}\boldsymbol{u}(t)\,,$$

where the state matrix $\boldsymbol{A}$ is composed of smaller matrices that when complemented form a 10×10 matrix. The driving vector $\boldsymbol{B}$ is a 10×1 matrix, and the state vector is

$$\boldsymbol{x} = \begin{bmatrix} \dot{\boldsymbol{Y}}(t) & \boldsymbol{Y}(t) & \boldsymbol{x_A}(t) \end{bmatrix}^T\,, \tag{6.1}$$

where

$$\boldsymbol{Y}(t) = \begin{bmatrix} \frac{h(t)}{b} & \alpha(t) & \beta(t) \end{bmatrix}^T\,, \qquad \boldsymbol{x_A}(t) = \begin{bmatrix} B_1(t) & B_2(t) & A_1(t) & A_2(t) \end{bmatrix}^T\,. \tag{6.2}$$

The states are, plunge (h) over time divided by a normalizing constant (b), the pitch over time (α), the flap angle over time (β), and the aerodynamic lag states (A_1(t), A_2(t), B_1(t), B_2(t)). All demonstration of the sub matrices (A_{ij}) contained in the state coefficient matrix $\boldsymbol{A}$, aerodynamic lag states and driving matrix $\boldsymbol{B}$ are in [30], the results were reproduced using the SIMULINK® tool.

In order to show the effectiveness of the LQR, only the results of the variations over time of the pitch and the flap angle are going to be considered.

This is a typical aeroservoelastic problem, when the airspeed increases the elastic airfoil starts deflecting and therefore increasing the aerodynamic forces acting on it, leading to bigger deflections and as, a result, bigger oscillations. There is an airspeed limit, called flutter velocity, for marginally stable oscillations. Depending on the airfoil material, it may lead to a catastrophic structural failure.

6.1.2 Open Loop Aeroservoelastic Problem

In this example the flutter velocity ($V_{Flutter}$) for this airfoil is 297.4 m/s. In order to begin the simulation, it is considered that there is an initial condition for the state variables ($\boldsymbol{x_0}$).

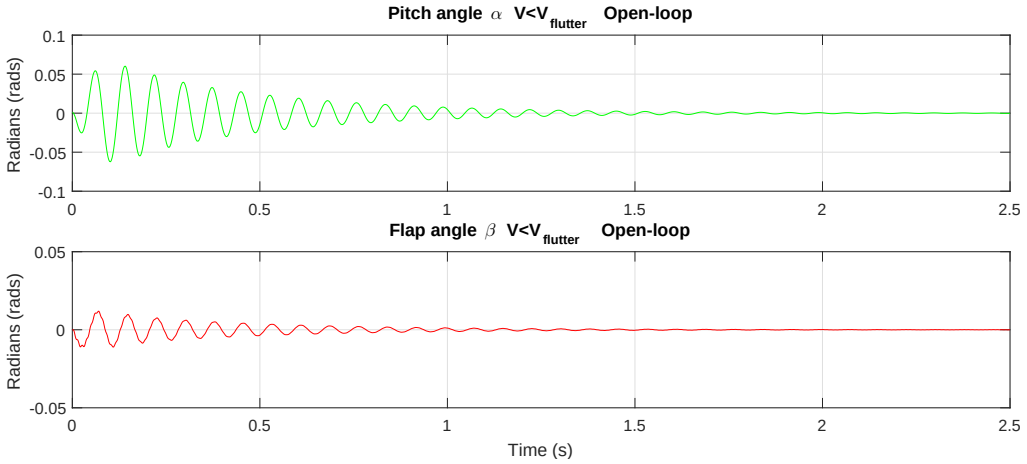

Figure 6.2: Pitch and flap angle variation over time considering ($V = 289.6$ m/s) on open-loop

Below the flutter velocity, Figure 6.2, it is visible that the larger amplitude oscillations last for about one second, the system is stable for this airspeed (V) of 289.6 m/s, all $\boldsymbol{A}$ eigenvalues have real negative part.

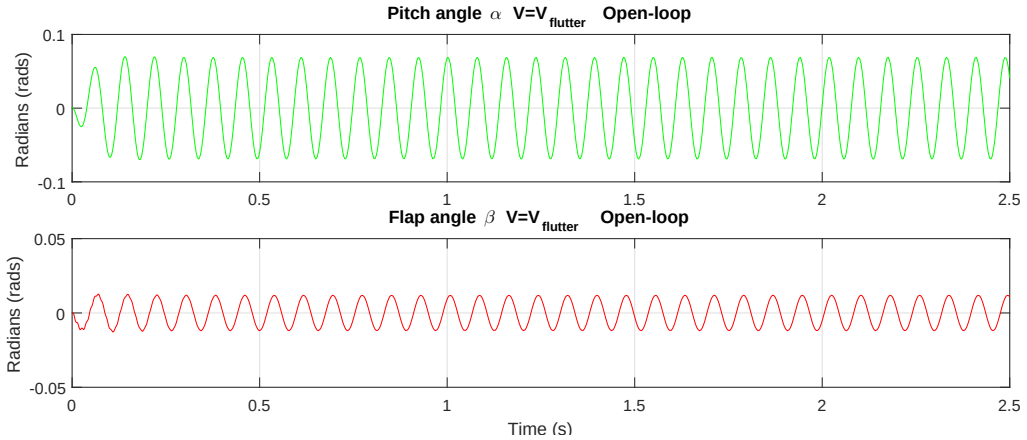

Figure 6.3: Pitch and flap angle variation over time considering flutter velocity ($V = V_{flutter} = 297.4$ m/s) on open-loop

If the airspeed is exactly the flutter velocity ($V = V_{Flutter} = 297.4$ m/s) as in Figure 6.3, the system remains marginally stable throughout all simulation. Marginally stable means that the matrix $\boldsymbol{A}$ has at least one eigenvalue with zero real part.

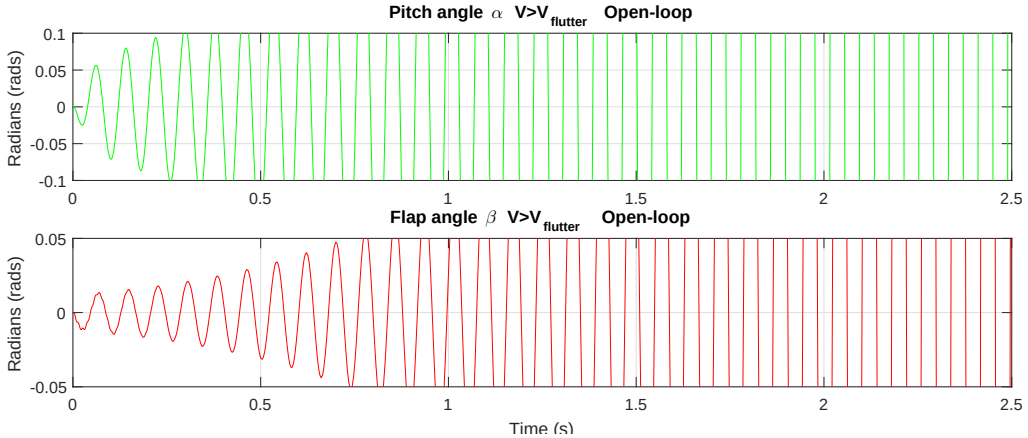

Figure 6.4: Pitch and flap angle variation over time considering ($V = 304.8$ m/s) on open-loop

Over the flutter velocity ($V = 304.8$) m/s, Figure 6.4, the system is clearly unstable, leading the oscillations to grow bigger through time. The main cause of the instability of the system is the matrix $\boldsymbol{A}$ having real positive part eigenvalues. Now the goal is to implement a control law, with the objective to suppress flutter. This will be done by inducing a feedback gain, calculated by the linear quadratic regulator (LQR), into the system and therefore increasing the system's flutter velocity. The linear quadratic regular is used because of its efficient application and utility. There are other more robust control laws, such as H_∞ [31].

6.1.3 Linear Quadratic Regulator

There are two great advantages when solving a linear quadratic problem. Firstly, the control is a full state linear feedback law

$$\boldsymbol{u}^{\circ} = -\boldsymbol{K}\boldsymbol{x}\,, \tag{6.3}$$

and secondly, this resulting feedback control law will ensure the system in closed-loop is stable and robust, but only if the system is controllable and stabilizable.[21]. This method is based in the optimization and minimization of the system's performance index J

$$J = \frac{1}{2}\int_0^\infty (\boldsymbol{x}^{\boldsymbol{T}}\boldsymbol{Q}\boldsymbol{x} + \epsilon \boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{R}\boldsymbol{u}) \quad dt\,. \tag{6.4}$$

The equation (6.4) represents a trade-off between, $\boldsymbol{x}$, $\boldsymbol{u}$ and two matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. The state vector $\boldsymbol{x}$ behaves as a constrain to the minimization of the performance index, J.

Matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ are square and symmetric matrices and they can be time-dependent. The state weighting matrix $\boldsymbol{Q}$ is a positive definite matrix and the control cost matrix $\boldsymbol{R}$ is a positive semi-definite matrix. The objective of these matrices is to regulate the importance of states and inputs variables in the considered problem. The ϵ is a parameter that determines the relative weights given to control action and perturbations in the state variable. It is designated as control penalty parameter.

The resolution of the performance index minimization leads to an optimal gain matrix given by equation (6.3). This gain affects all states and is considered optimal, given certain matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$. The

demonstration and resolution of the minimization can be seen in [32] or in [14].

In this dissertation, both in this flutter suppression problem and in the final flight controller, MATLAB® will be used to generate and solve this linear quadratic problem, using the function *lqr*. This function uses the matrix $\boldsymbol{A}$ and $\boldsymbol{B}$ from the state-space form's first equation, the state weighting matrix $\boldsymbol{Q}$, the control cost matrix $\boldsymbol{R}$ and then generates the optimal feedback gain $\boldsymbol{K}$, known as the gain matrix. This will be the control method utilized in this project for stabilizing the model and closing the loop.

6.1.4 Closed Loop Aeroservoelastic Problem

In this particular aeroservoelastic case, LQR was applied as a control method in the pursuance of finding a control function $\boldsymbol{u}$(t) to stabilize the system. This control function will have the form presented in equation (6.3)

$$\boldsymbol{u(t)} = -\boldsymbol{K_{LQR}}\boldsymbol{x(t)}\,, \tag{6.5}$$

and the closed loop system

$$\dot{\boldsymbol{x}}\boldsymbol{(t)} = \boldsymbol{A}\boldsymbol{x(t)} + \boldsymbol{B}\boldsymbol{u(t)} \quad \rightarrow \quad \dot{\boldsymbol{x}}\boldsymbol{(t)} = (\boldsymbol{A} - \boldsymbol{B}\boldsymbol{K_{LQR}})\boldsymbol{x(t)} \quad \rightarrow \quad \dot{\boldsymbol{x}}\boldsymbol{(t)} = \boldsymbol{A^*}\boldsymbol{x(t)}\,, \tag{6.6}$$

where $\boldsymbol{A^*}$ is the the augmented plant matrix. The values of the entries of matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ ((6.7) and (6.8), respectively) are given through trial and error. The *lqr* MATLAB® function retrieves the associated full state feedback gain accordingly to the system matrices ($\boldsymbol{A}$ and $\boldsymbol{B}$).

$$\boldsymbol{Q} = \begin{bmatrix} 1000 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1000 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.1 \end{bmatrix}\,, \tag{6.7}$$

$$R = \frac{V^2}{1000000}\,. \tag{6.8}$$

Considering the flutter velocity ($V_{flutter}$) as 297.4 m/s, linear quadratic regulator feedback gain for this arguments will be

$$\boldsymbol{K_{LQR}} = \begin{bmatrix} 0.547 & -1.529 & 0.6858 & -38.6409 & -39.4681 & -27.4179 & 0.0017 & 0.0349 & 0.0001 & 0.0173 \end{bmatrix}\,, \tag{6.9}$$

and the closed loop pitch and flap angle dynamic response is presented in Figure 6.5.

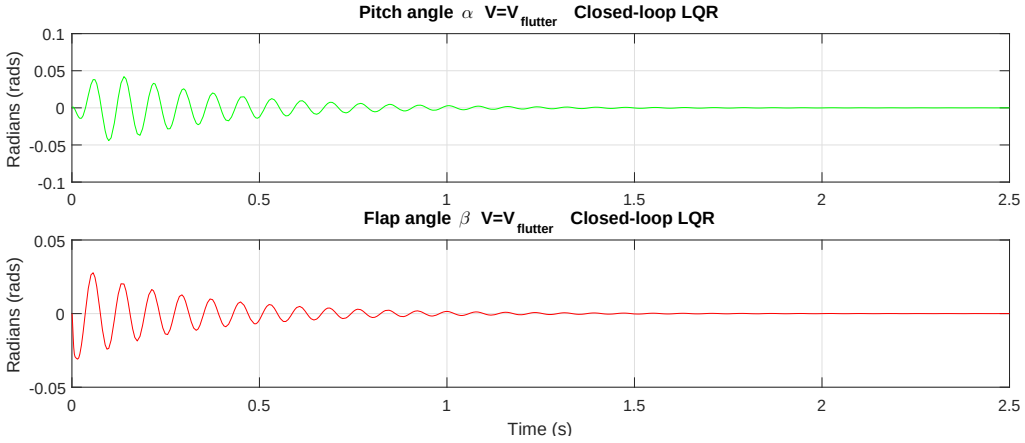

Figure 6.5: Pitch and flap angle variation over time considering flutter velocity ($V = V_{flutter} = 297.4$ m/s) on closed-loop

It is possible to see that closing the loop stabilized the system. This system that once was marginally stable on the open-loop, now is completely stable having its oscillations ending after about one and a half seconds. The full state feedback gain made the zero real part eigenvalues of open-loop, translate to real negative part eigenvalues in the closed loop.

Applying the same method but for an airspeed of 304.8 m/s, that in the open loop would make the system unstable. The full state feedback gain will be

$$\boldsymbol{K_{LQR}} = \begin{bmatrix} 9.8 & -39.9 & -1.6 & -1279.7 & -804 & -650.2 & 0.1 & 0.9 & 0 & 0.1 \end{bmatrix}, \tag{6.10}$$

and the following closed loop pitch and flap angle dynamic response is presented in Figure 6.6.

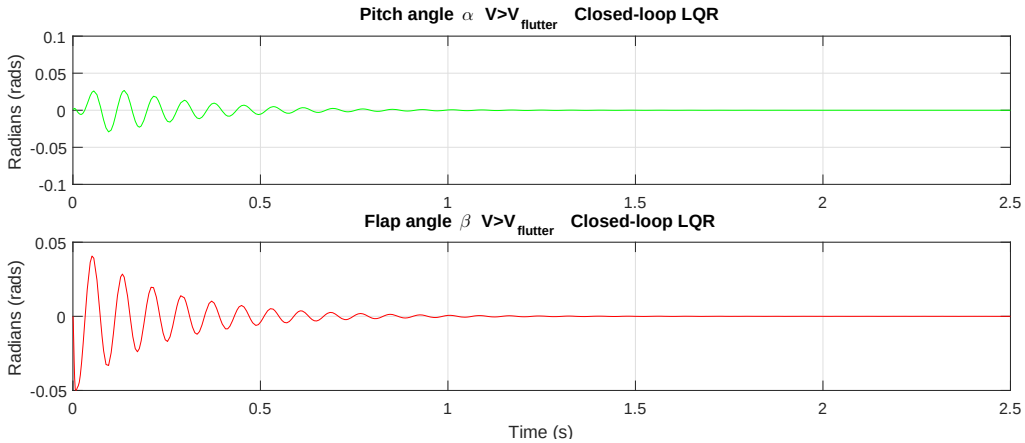

Figure 6.6: Pitch and flap angle variation over time considering ($V = 304.8$ m/s) on closed-loop

This open-loop unstable system is now stable in the closed loop having its oscillations end after about one second. The closed loop eigenvalues both in this case and on the case above, all have real negative parts.

6.1.5 Linear Quadratic Regulator Concerns and Conclusions

However through these two examples it is possible to see that suppressing flutter for higher airspeed using the LQR method implies higher gain matrices, or in this case, gain vectors. There is also one final thought to add to this particular problem, there is not a control variable in it. This is a merely mathematical demonstration of the LQR virtues, the control vector ($\boldsymbol{u}$) in this problem is always the feedback gain from LQR.

Summing up, LQR control is tempting but unrealistic. Firstly because it needs all states to be known at all times in order to use state feedback. In this particular case it would be impractical since some of the states, as aerodynamic lag states (A_1(t), A_2(t), B_1(t) and B_2(t)), are not physical and therefore cannot be detected by sensors. Secondly, for higher airspeed, LQR produces higher gains, and in the real world, actuators may not be able to deal with these higher gains, resulting in saturation. However, it is an easy implementable and robust control tool, and it will be the control method adopted in this dissertation's flight controller.

One of the problems for this thesis project is the definition of the LQR weight matrices ($\boldsymbol{Q}$ and $\boldsymbol{R}$). In the flutter suppression problem, these matrices were defined arbitrarily, through trial and error, with the objective to have the best possible full state feedback gain [30]. Although in the flight controller, as it will be for a generic aircraft on a given flight condition, a method for obtaining appropriate state weighting ($\boldsymbol{Q}$) and control cost ($\boldsymbol{R}$) matrices is required.

Bryson's Method

Since the definition of matrices $\boldsymbol{Q}$ and $\boldsymbol{R}$ can be arbitrary, there is a method called Bryson's method in which it suggests that each term of the diagonal matrices, $\boldsymbol{Q}$ and $\boldsymbol{R}$, is the inverse square of the maximum value expected for the variable on the simulation time. As it follows

$$Q = diag(Q_i) \Rightarrow Q_i = \frac{1}{x_{i_{max}}^2}\,, \qquad R = diag(R_i) \Rightarrow R_i = \frac{1}{u_{i_{max}}^2}\,. \tag{6.11}$$

In the flight control system, $u_{i_{max}}^2$ and $x_{i_{max}}^2$ are the values indicating the extreme of the perturbations wanted for u_i or x_i for the closed loop during a maneuver. This method is a good starting point to define these matrices and will be used in the flight control problem. [14]

6.2 Flight Control Coupled Model

The model of the equations of motion to be used by the linear quadratic regulator script is the combination of both lateral and longitudinal models defined in equations (5.18) and (5.14)

$$\dot{\boldsymbol{x}}_{\boldsymbol{LQR}} = \boldsymbol{A}_{\boldsymbol{LQR}}\boldsymbol{x}_{\boldsymbol{LQR}} + \boldsymbol{B}_{\boldsymbol{LQR}}\boldsymbol{u}_{\boldsymbol{coupled}}\,, \tag{6.12}$$

where

$$\boldsymbol{A}_{\boldsymbol{LQR}} = \begin{bmatrix} \boldsymbol{A}_{\boldsymbol{Long}_{u\gamma_s}} & 0 \\ 0 & \boldsymbol{A}_{\boldsymbol{Lat}_{\lambda_s}} \end{bmatrix}, \qquad \boldsymbol{x}_{\boldsymbol{LQR}} = \begin{bmatrix} \boldsymbol{x}_{\boldsymbol{Long}_{u\gamma_s}} & \boldsymbol{x}_{\boldsymbol{Lat}_{\lambda_s}} \end{bmatrix}. \tag{6.13}$$

$\boldsymbol{A}_{\boldsymbol{LQR}}$ is a 12×12 diagonal state matrix since it is assumed that lateral states do not influence longitudinal states and vice-versa or, in another words, they are decoupled from each other. The states are

$$\boldsymbol{x}_{\boldsymbol{LQR}} = \begin{bmatrix} u & w & q & \gamma & x_u & x_\gamma & \beta & p & r & \phi & \lambda & x_\lambda \end{bmatrix}^T \tag{6.14}$$

The $\boldsymbol{B}_{\boldsymbol{LQR}}$ is a $12 \times (N_{eng} + 3)$ driving matrix containing both longitudinal and lateral driving matrices from (5.18) and (5.14),

$$\boldsymbol{B}_{\boldsymbol{LQR}} = \begin{bmatrix} X_{\delta_E} & X_{\delta_{T_1}} & ... & X_{\delta_{T_{N_{eng}}}} & 0 & 0 \\ Z_{\delta_E} & Z_{\delta_{T_1}} & ... & Z_{\delta_{T_{N_{eng}}}} & 0 & 0 \\ \tilde{M_{\delta_E}} & (z_1 X_{\delta_{T_1}} - x_1 Z_{\delta_{T_1}}) & ... & (z_{N_{eng}} X_{\delta_{T_{N_{eng}}}} - x_{N_{eng}} Z_{\delta_{T_{N_{eng}}}}) & 0 & 0 \\ -\frac{Z_{\delta_E}}{u_0} & -\frac{Z_{\delta_{T_1}}}{u_0} & ... & -\frac{Z_{\delta_{T_{N_{eng}}}}}{u_0} & 0 & 0 \\ 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & ... & 0 & \frac{Y_{\delta_A}}{u_0} & \frac{Y_{\delta_R}}{u_0} \\ 0 & -y_1 Z_{\delta_{T_1}} & ... & -y_{N_{eng}} Z_{\delta_{T_{N_{eng}}}} & L'_{\delta_A} & L'_{\delta_R} \\ 0 & y_1 X_{\delta_{T_1}} & ... & y_{N_{eng}} X_{\delta_{T_{N_{eng}}}} & N'_{\delta_A} & N'_{\delta_R} \\ 0 & 0 & ... & 0 & 0 & 0 \\ 0 & 0 & ... & 0 & \frac{Y_{\delta_A}}{u_0} & \frac{Y_{\delta_R}}{u_0} \\ 0 & 0 & ... & 0 & 0 & 0 \end{bmatrix}. \tag{6.15}$$

6.2.1 Schematic of the Flight Controller Model

The general schematic of the flight control model implemented in this project is in the Figure 6.7.

The desired control states are longitudinal speed, flight path angle and heading angle. The flight controller or the pilot inserts references for these states and the model follows them, aided by the linear quadratic regulator feedback gains. The integrator gains obtained from the linear quadratic regulator are included within the gain of the desired variable

$$K_\gamma = K_{LQR_\gamma} + \frac{1}{s} K_{LQR_{x_\gamma}}\,. \tag{6.16}$$

Longitudinal speed (u) and heading angle (λ) as desired control states, have the same structure

$$K_u = K_{LQR_u} + \frac{1}{s} K_{LQR_{x_u}}\,, \qquad K_\lambda = K_{LQR_\lambda} + \frac{1}{s} K_{LQR_{x_\lambda}}\,. \tag{6.17}$$

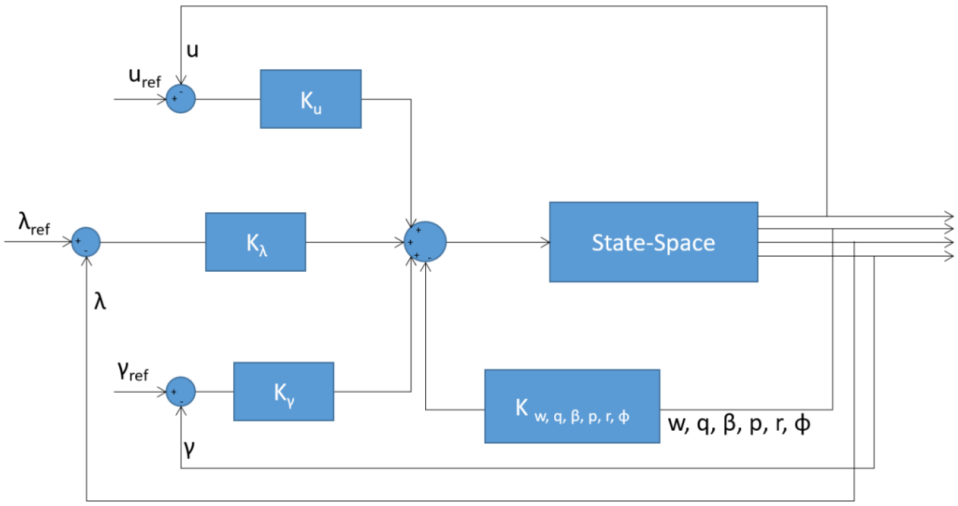

Figure 6.7: Schematic of the control law

The integrator states variables eliminate constant input values and also helps to stabilize the system. All of these gains are optimal and are calculated using the Bryson's method explained in Section 6.1.5. The non-desired control state gains, $K_{w,q,\beta,p,r,\phi}$, are included in the outer feedback loop, as seen in Figure 6.7

$$K_{w,q,\beta,p,r,\phi} = \begin{bmatrix} K_{LQR_w} & K_{LQR_q} & K_{LQR_\beta} & K_{LQR_p} & K_{LQR_r} & K_{LQR_\phi} \end{bmatrix}^T . \tag{6.18}$$

The input vector will then be

$$u = -\begin{bmatrix} K_u & K_\gamma & K_\lambda \end{bmatrix} \begin{bmatrix} u - u_{ref} \\ \gamma - \gamma_{ref} \\ \lambda - \lambda_{ref} \end{bmatrix} - \begin{bmatrix} K_{LQR_w} & K_{LQR_q} & K_{LQR_\beta} & K_{LQR_p} & K_{LQR_r} & K_{LQR_\phi} \end{bmatrix} \begin{bmatrix} w \\ q \\ \beta \\ p \\ r \\ \phi \end{bmatrix} \tag{6.19}$$

6.2.2 SIMULINK® Flight Controller Model

The SIMULINK® flight controller model is in Figure 6.8. It gathers the concepts developed in previous sections, to create a ready-to-use SIMULINK® flight controller model.

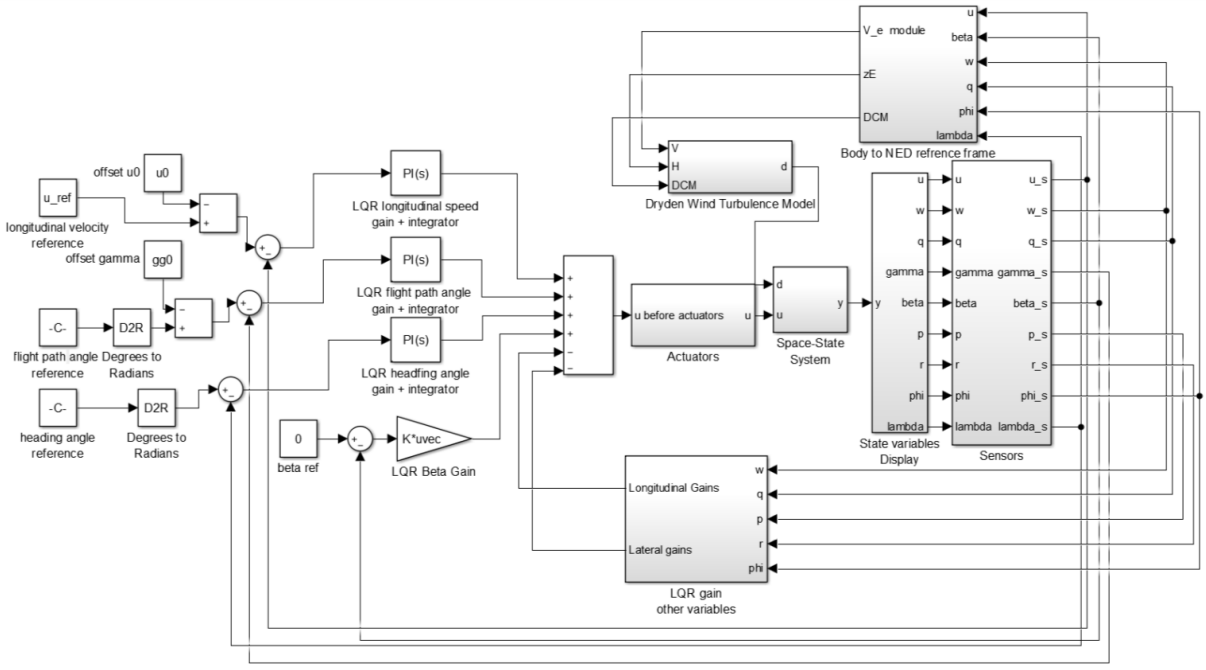

Figure 6.8: Flight controller model created in SIMULINK®

The dynamics block, named as Space-State System is defined in Figure 5.3, includes the turbulence model defined in Section 5.5.2. This turbulence model can be toggled off by simply unchecking the 'Turbulence on' box, and thus having a non-turbulent simulation. The control part of the model was built based in Schematic 6.7 and Section 6.2.1. For the follow through simulations of Chapter 7, as most of the steady states require null side slip angle, there is a constant reference for the side slip angle (β=0^o so that $\lambda \approx \psi$). The actuators block is presented in Figure 5.1 and the sensors block, for now, is considered ideal.

6.3 Linear Quadratic Regulator Script

The first step of the flight controller is to assure the aircraft dynamic modes have level 1 flying qualities. In order to reach that goal, several scripts and functions were created in MATLAB® applying the concepts explained in this Chapter and Chapter 5. The objective is to use the linear quadratic regulator as a stability augmentation system. It is possible to place poles utilizing a linear quadratic regulator weight selection algorithm [33], however as this flight controller is designed for a general aircraft, Bryson's method was implemented for an appropriate pole placement.

6.3.1 Flying Qualities Evaluator

For evaluating the aircraft's longitudinal and lateral modes two functions were created.

```
function Long_Flying_qual = flyingqual_long(a_long, Flight_phase, plane_level
    )
```

Listing 6.1: Longitudinal flight qualities evaluator

The goal of this implemented function, in Listing 6.1, is to evaluate the longitudinal dynamic modes, phugoid and short period, and to output its levels of flying qualities for both dynamic modes.

It takes as arguments the aircraft's, flight phase ($Flight\_phase$) and class ($plane\_level$), which are defined in the $.mat$ flight condition data base of the aircraft. It also receives the longitudinal state-space state coefficient matrix ($\boldsymbol{A_{Long_{u\gamma_s}}}$) as the a_long parameter.

First by applying the function $damp$, it discovers the eigenvalues (poles) of the state coefficient matrix (a_long). Then, after identifying what egeinvalues correspond to which dynamic mode, the function correlates the phugoid's and short period's damping (ξ) and natural frequencies (ω_n), with Tables A.3 and A.4. After the correlation is done, it outputs a column matrix ($Long\_Flying\_qual$) in which the first row contains the phugoid's level (1 to 3) and the second row contains the short period's level (1 to 3).

```
function Lat_Flying_qual = flyingqual_lat(a_lat, Flight_phase, plane_level)
```

Listing 6.2: Lateral flight qualities evaluator

The function in Listing 6.2 works in the same way of the function in Listing 6.1. The input variables are the same, but the inserted matrix is the lateral state-space state coefficient matrix ($\boldsymbol{A_{Lat_{\lambda_s}}}$). It does the same processing as the function listed in Listing 6.1, but associating Tables A.5, A.6 and A.7 to the spiral, roll and dutch roll dynamic modes. It outputs a column matrix (Lat_Flying_qual) in which the first row contains the spiral's level (1 to 3), second row contains the roll's level (1 to 3) and the final row has dutch roll's level (1 to 3).

6.3.2 Applicable Bryson's method

The linear quadratic regulator gains are generated from the $\boldsymbol{Q}$ and $\boldsymbol{R}$ obtained by applying the Bryson's method to a general aircraft. As it was mentioned in Section 6.2, the aircraft dynamics is represented in state-space form where $\boldsymbol{A_{LQR}}$ is the state coefficient matrix and $\boldsymbol{B_{LQR}}$ the driving matrix.

For the most part, longitudinal and lateral models of aircraft dynamics are controllable [34]. So, as $\boldsymbol{A_{LQR}}$ is a diagonal matrix formed by the longitudinal and lateral state coefficient matrix, as seen in equation (6.13), it is assumed to be controllable. As a result of that, even if an open-loop system is unstable, when closing the loop with the linear quadratic regulator gains it becomes stable.

```
K = lqr(A,B,Q_ini,R_ini)
```

Listing 6.3: Linear quadratic regulator function

Thus, the first step of this algorithm was to use the function in Listing 6.3, with $\boldsymbol{A_{LQR}}$ as first argument, $\boldsymbol{B_{LQR}}$ as second, a same dimension as A , identity matrix (Q_ini) and an identity matrix R_ini

with dimension equal to the columns of $\boldsymbol{B_{LQR}}$ (number of control variables). The augmented plant matrix (closed loop) of equation (6.20)

$$\boldsymbol{A^*_{LQR}} = \boldsymbol{A_{LQR}} - \boldsymbol{B_{LQR}}\boldsymbol{K}\,, \tag{6.20}$$

is guaranteed to be stable. The $\boldsymbol{A^*_{LQR}}$ and $\boldsymbol{B_{LQR}}$ matrices are then used as the state coefficient matrix and driving matrix, respectively, in the state-space of Figure 6.9. It was chosen for the impulses of the control surfaces in Figure 6.9, 10% of the maximum surface deflections, as it represents, for a general aircraft, the control surface deflection used in a typical manoeuvre. The Bryson's method $\boldsymbol{R_{final}}$ matrix will be a diagonal matrix composed of the inverse-square of these values ($\frac{1}{u^2_{i_{max}}}$).

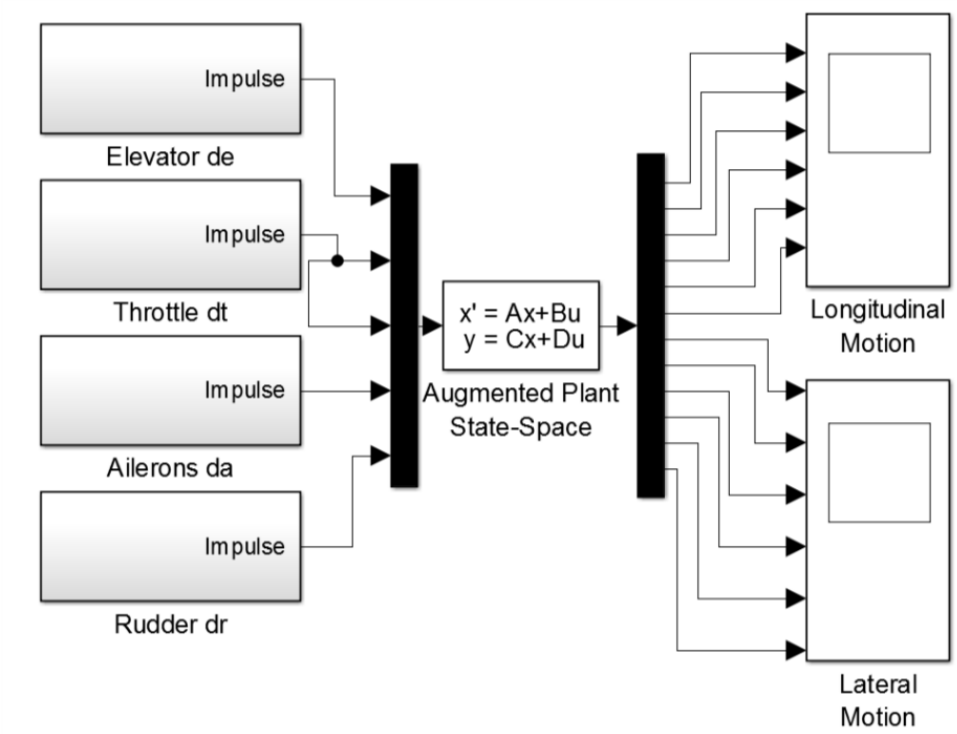

Figure 6.9: SIMULINK® schematic used for applying Bryson's Method ($N_{eng} = 2$)

Covering all combinations of impulses on the control surfaces, the script saves the state maximum values obtained by the various simulations for each state variable. Those values ($x_{i_{max}}$) are then used to create the Bryson's method $\boldsymbol{Q_{final}}$ matrix, which will be a diagonal state matrix with entries, $\frac{1}{x^2_{i_{max}}}$, that are relative to each state.

With these new $\boldsymbol{Q_{final}}$ and $\boldsymbol{R_{final}}$ matrices, there's a cycle, that tests for each iteration the flying qualities of the aircraft in the closed loop. In each iteration the $\boldsymbol{R_{final}}$ matrix is multiplied by a certain control penalty parameter (ϵ varies from 0.1 to 1000) and then submitted to the lqr function of Listing 6.4. The cycle breaks when the script finds an augmented matrix

$$\boldsymbol{A^*_{LQR}} = \boldsymbol{A_{LQR}} - \boldsymbol{B_{LQR}}\boldsymbol{K_{LQR}}\,, \tag{6.21}$$

that has level 1 qualities in all dynamic modes. Therefore for each iteration, functions in Listings 6.2 and 6.1 are consecutively called to check the augmented matrix lateral and longitudinal dynamics' modes levels. Briefly, when the cycle breaks it means the optimal gain matrix ($\boldsymbol{K_{LQR}}$) to use in the flight controller is found.

```
K_LQR = lqr(A_LQR,B_LQR, Q_final , R_final*rho)
```

Listing 6.4: Linear quadratic regulator function cycle

Chapter 7

Flight Simulation

Besides the flight dynamics integrator program created in C++®, an optimal flight controller using LQR on MATLAB® was also developed. This program made of the MATLAB® scripts and SIMULINK modules, seen over Chapters 6 and 5, will allow the study of certain flight conditions from a flight engineer point of view.

Initially, open-loop dynamics of the flight conditions for the Airbus A400M and the Dassault Falcon 7X aircraft will be analysed. Then, the goal is to use the flight controller to follow a reference in the case of engine failure and turbulence.

7.1 Open-Loop Dynamics

The aircraft used for the engine failure test is the four engines Airbus A400M and the aircraft to be tested in turbulent flow is the Dassault Falcon 7X . These aircraft trim data is in Appendix A.3.3.

		Aircraft	
		Dassault Falcon 7X	Airbus A400M
Longitudinal motion	Phugoid	κ_{phu_1} = -0.0579 + 0.231i κ_{phu_2} =-0.0579 - 0.231i	κ_{phu_1} = -0.066 +0.0883i κ_{phu_2} = -0.066 - 0.0883i
	Short period	κ_{sp_1} = -0.7270 + 1.5i κ_{sp_2} = -0.7270 + 1.5i	κ_{sp_1}= -7.47 + 3.23i κ_{sp_1} = -7.47 - 3.23i
Lateral motion	Spiral	κ_{spi} =-0.0138	κ_{spi} = 0.0840
	Roll	κ_{roll} = -2.35	κ_{roll} = -1.26
	Dutch Roll	κ_{dr_1} = -0.314 + 4.18i κ_{dr_2} = -0.314 - 4.18i	κ_{dr_1} = -0.141 + 1.82i κ_{dr_2} = -0.141 - 1.82i
	Heading mode	κ_λ = 0	κ_λ = 0

Table 7.1: Open-loop dynamic modes eigenvalues

By first applying equation (5.19) and then correlating the damping and the natural frequency of A400M dynamic modes, contained in Table 7.1, with Tables A.3, A.4, A.6, A.5 and A.7. It is possible to conclude that the phugoid has level three and the short period has level two. In the lateral motion, the spiral mode is stable and thus level one, roll is also level one and dutch roll is categorized as level two. The pole at the origin is due to the heading mode (λ), seen in Section 5.4.5. Overall, the longitudinal and lateral motions of the aircraft do not have the level one flying qualities requirement.

When categorizing the dynamic modes of the Dassault Falcon 7X, utilizing the same method as above, phugoid is level one and the short period is also level one. As for the lateral dynamic modes, spiral is a positive real number making it level three, however roll is level one and dutch roll is level two. Dynamic modes of this Falcon 7X's flight condition are unstable.

The LQR script of Chapter 6 must place the poles of these dynamic modes, in order that the level one flying qualities of the aircraft are satisfied.

7.2 Engine Failure

In this section, the left engines of the four engine Airbus A400M (δ_{T_1} and δ_{T_2}) are malfunctioning and therefore providing no thrust to the aircraft

$$\delta_{T_1} = 0\,, \qquad \delta_{T_2} = 0\,. \tag{7.1}$$

The control surfaces available for the aircraft are then δ_{T_3}, δ_{T_4}, δ_E, δ_A and δ_R. The goal is to perform a climb ($\gamma_{ref} = 1^o$) while maintaining heading ($\lambda_{ref} = 0^o$). The reference for longitudinal velocity is maintained at the trim velocity ($u_{ref} = u_0$).

First, the state coefficient matrices ($\boldsymbol{A_{coupled}}$ to be used in the state-space and $\boldsymbol{A_{LQR}}$ in the linear quadratic regulator script) and driving matrices ($\boldsymbol{B_{coupled}}$ to be used in the state-space and $\boldsymbol{B_{LQR}}$ in the linear quadratic regulator script) need to be defined by utilizing the A400M trim condition data contained in Appendix A.3.3.

The script is then called and proceeds to find level one flying qualities for a control penalty parameter of $\epsilon = 10$. In the gain matrix associated with this value (B.9), the two rows composed of zeros correspond to engine one (δ_{T_1}) and engine two (δ_{T_2}), which are not functioning.

	Control penalty parameter (ϵ)	10	40	80
Longitudinal motion	Phugoid ($\kappa_{phu_{1,2}}$)	-0.597 $\pm$ 0.252i	-0.414 $\pm$ 0.36i	-0.369 $\pm$ 0.36i
	Short period ($\kappa_{sp_{1,2}}$)	-11.3 $\pm$ 4.37i	-6.86 $\pm$ 5.23i	-5.39 $\pm$ 4.92i
Lateral motion	Spiral (κ_{spi})	-0.631	-0.603	-0.569
	Roll (κ_{roll})	-3.36	-2.54	-2.28
	Dutch roll ($\kappa_{dr_{1,2}}$)	-1.98 $\pm$ 2.65i	-1.33 $\pm$ 2.41i	-1.08 $\pm$ 2.26i

Table 7.2: Closed loop poles for ϵ = 10, ϵ = 40 and ϵ = 80

The closed-loop dynamic mode poles are represented in Table 7.2. Upon initial inspection, the poles are moving to the right side of the complex plane as the control penalty paramenter (ϵ) is increasing.

Thus, this parameter, as it is penalizing the control input, when assigned high values can make the open-loop and closed-loop dynamic responses be virtually identical.

The model's dynamic responses of a thirty five seconds simulation on SIMULINK are presented in Figures 7.1, 7.2 and 7.3. The position of the aircraft in the ENU frame is defined in Figure 7.4. Three different control penalty parameters were tested, in order to see its influence in the aircraft dynamics. Therefore, for each plot there are three sets of curves and the markers for each curve are only to ease identification. In Figures 7.1 and 7.2, the effects of the control penalty parameter on the control

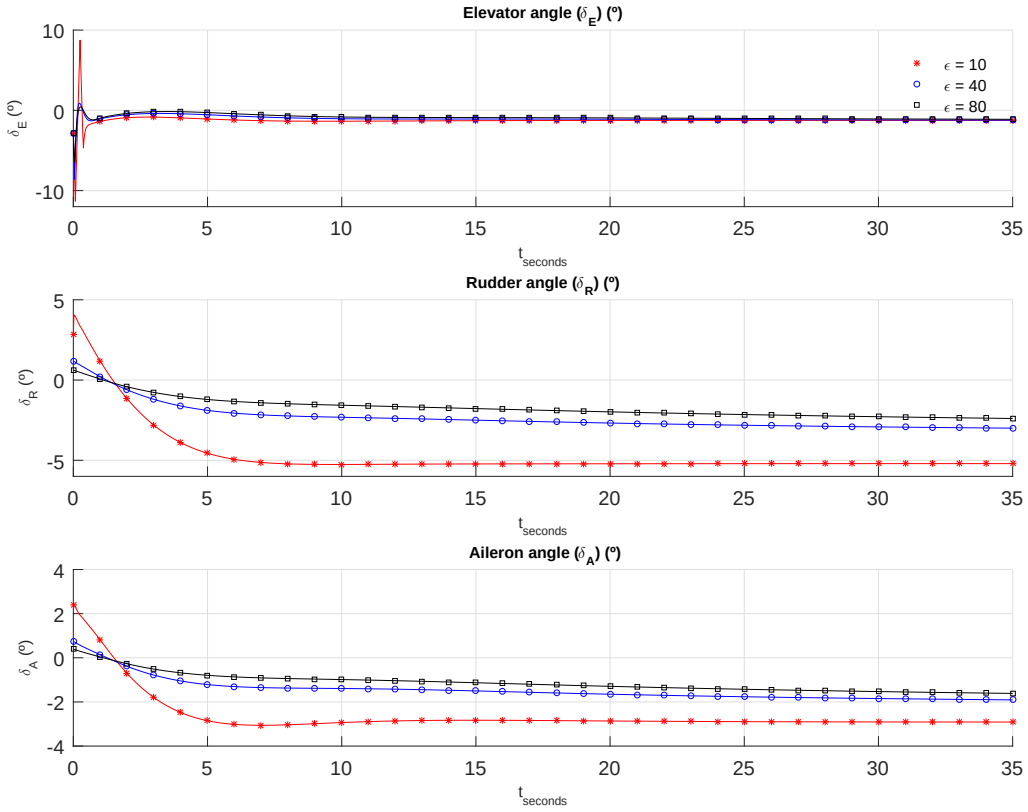

Figure 7.1: Dynamic responses of the control input variables (δ_E,δ_A,δ_R) for three different control penalty parameters

surfaces dynamic responses are evident. Dynamic responses of engines 1 and 2 (δ_{T_1} and δ_{T_2}) are not represented as their response is zero over the whole simulation. When the system has reached an equilibrium state, the constant negative deflections in the rudder and ailerons generate a positive yaw and clockwise roll moments used to stabilize the aircraft due to the loss of the two left engines. As the aircraft also needs to climb (γ_{ref} = 1º), after the initial variations, the elevator deflection finds its equilibrium state at a negative value. As the functioning engines have influence in the longitudinal and lateral motion, the outer engine throttle (δ_{T_4}) tends to decrease because of its higher influence in the yaw moment, and the inner engine (δ_{T_3}) increases its thrust in such a way that the three reference states are achievable and an equilibrium state is found.

Increasing the control penalty parameter makes the system utilize less the control surfaces, therefore

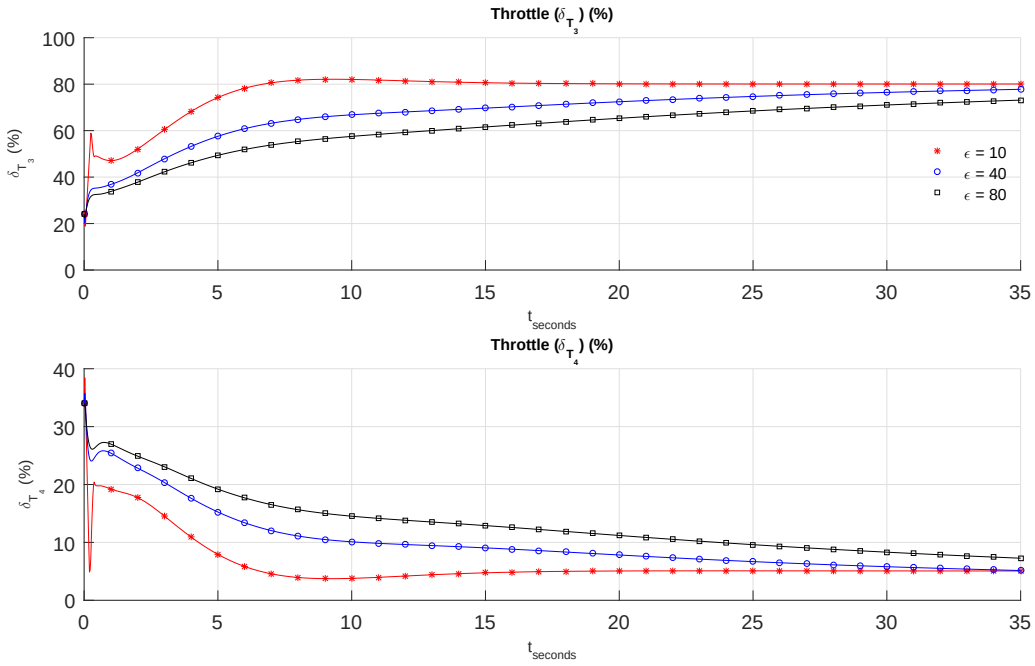

Figure 7.2: Dynamic responses of the non-malfunctioning engines (δ_{T_3},δ_{T_4}) for three different control penalty parameters

reaching the desirable state in a higher time. However, higher values also attenuates the overshooting deflections when requested to follow a reference, making the system more realistic, because high overshooting deflections in control surfaces in short periods of time may have rough effects in the aircraft structural dynamics. These overshooting control surfaces deflections also may not be physically possible due to the actuator dynamics.

When the system allows for higher control deflections (ϵ = 10), the reference state is reached faster than when the control action is more limited (ϵ = 80). However, for reasonable values of ϵ and assuming the system is stable, this final reference state is eventually reached as seen in Figure 7.3.

The representation of the states transformed into positions on the XY_{ENU} and YZ_{ENU} planes are shown in 7.4. The aircraft's trajectory is represented in the ENU reference frame, to facilitate visual interpretation of the results. Initially the lack of thrust in the left engines make the aircraft deviate to the left (negative x_{ENU}) but eventually through the control surfaces deflections the heading is stabilized.

For this case in specific the handling qualities might arise due to $\frac{\omega_{phu}}{\omega_{sp}}$ being always less than 0.1, however the reason for this is the malfunctioning of the two engines. For this trim condition data after level one flying qualities is achieved (ϵ = 10), the result of that division is always decreasing for higher ϵ values, making it impossible to reach.

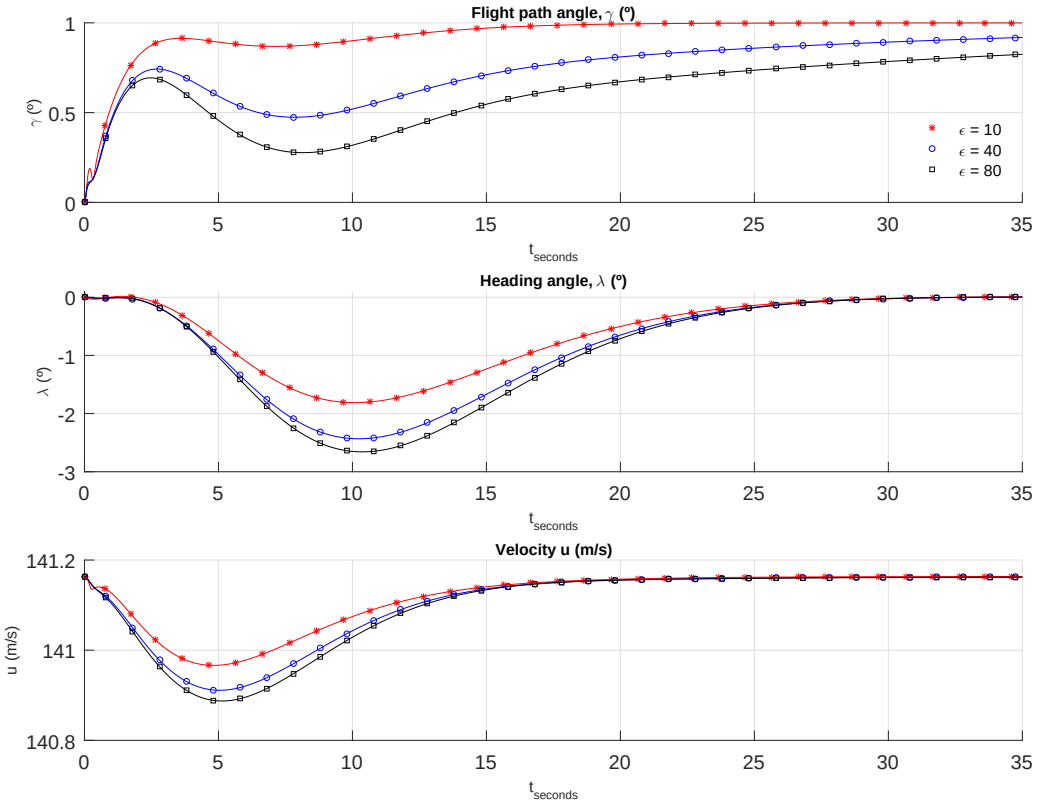

Figure 7.3: Dynamic responses of the controllable states (u,γ,λ)

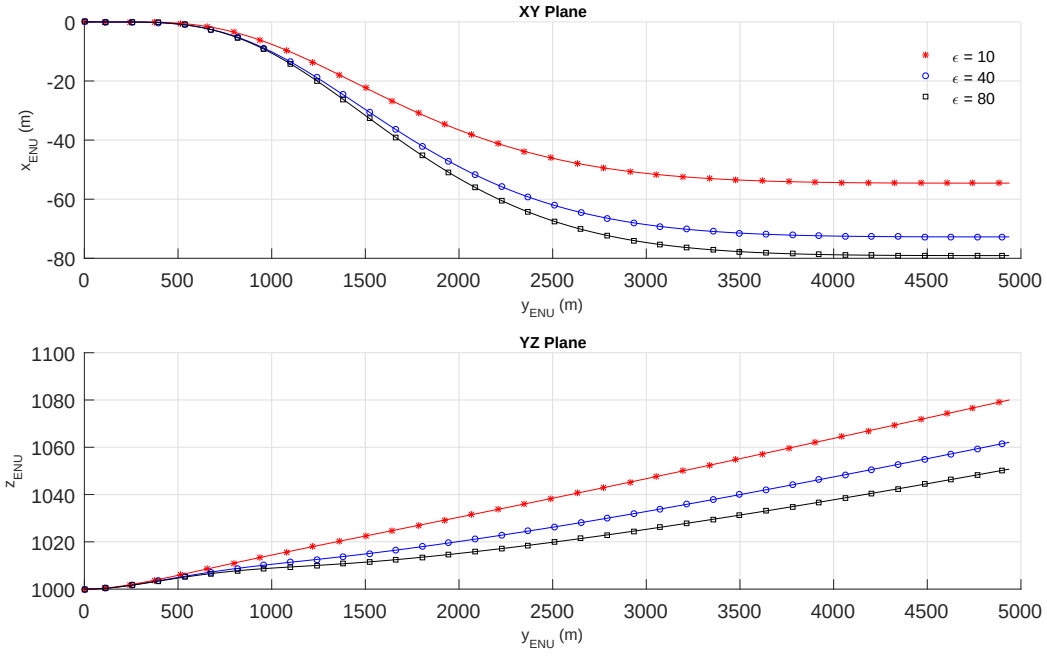

Figure 7.4: Flight trajectory seen in XY_{ENU} and YZ_{ENU} planes ($\gamma_{ref} = 1$º, $\lambda_{ref} = 0$º and $u_{ref} = u_0$)

7.3 Turbulence

In this Section the three engined Dassault Falcon 7X aircraft is flying in a turbulent flow. The aircraft is required to maintain its heading and flight path angle ($\lambda_{ref} = \gamma_{ref} = 0^{o}$), all of the three engines are functioning. The state coefficient ($\boldsymbol{A}$) and driving ($\boldsymbol{B}$) matrices are determined by utilizing the aircraft's trim condition data contained in Appendix A.3.3.

The first control penalty parameter for which the script found level one flying qualities was for $\epsilon = 20$. The gain matrix represented in equation B.10, has the particularity of its middle engine only affecting the longitudinal mode, as it is situated under the elevator and thus, has no influence in the yaw and roll moment.

Alongside with the control penalty parameter found by the script, two more dynamic responses produced by other two control parameters are analysed. This time, a higher and lower order of magnitude values were used. The high order value ($\epsilon = 200$) has level one flying qualities and the lower order value does not have level one flying qualities.

	Control penalty (ϵ)	2	20	200
Longitudinal motion	Phugoid ($\kappa_{phu_{1,2}}$)	κ_{phu_1}=-2.34 κ_{phu_2} =-0.591	-0.306 $\pm$ 0.241i	-0.266 $\pm$ 0.424i
	Short period ($\kappa_{sp_{1,2}}$)	-8.28 $\pm$ 0.65i	-2.82 $\pm$ 3.52i	-1.71 $\pm$ 1.99i
Lateral motion	Spiral (κ_{spi})	-0.792	-0.620	-0.435
	Roll (κ_{roll})	-4.40	-2.75	-2.45
	Dutch roll ($\kappa_{dr_{1,2}}$)	κ_{dr_1} =-14.7 κ_{dr_2} =-2.54	-2.83 $\pm$ 3.51i	-0.951 $\pm$ 4.11i

Table 7.3: Closed loop poles for ϵ = 2, ϵ = 20 and ϵ = 200

In Table 7.3, the poles for different control penalty parameters are represented. For the lower value of control penalty the phugoid and dutch roll dynamic modes are critically damped, such value is not adequate for a realistic and practical system. Overall the poles behave the same way with the control penalty parameter as in Table 7.2.

Just as in Section 7.2, there are three curves for each plot, each representing a control penalty value. In Figures 7.5, 7.6 and 7.7 there are the dynamic responses and Figure 7.8 has the positions during the thirty five seconds simulation.

In Figure 7.5, as the objective is the aircraft maintaining its altitude ($\gamma_{ref} = 0^{o}$) and heading ($\lambda_{ref} = 0^{o}$), for the values of control penalty parameter computed, the goal is achieved. The control penalty analysis is somewhat similar to the engine failure situation of Section 7.2, for higher values its control system is weighting heavily in the performance index, making the aircraft having larger deviations due to turbulence.

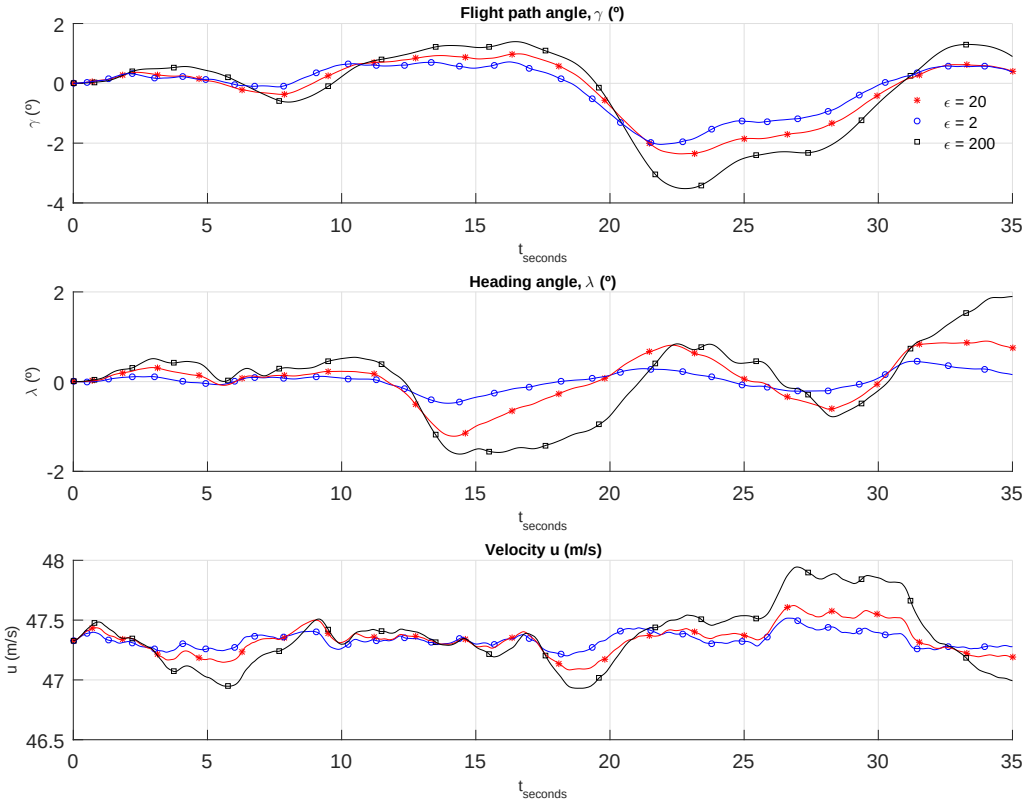

Figure 7.5: Dynamic responses of the controllable states (u,γ,λ)

The interpretation of Figures 7.6 and 7.7 is also identical to the control surfaces deflections in Section 7.2. With higher values of control penalty parameter, the control surfaces are susceptible of having higher deflections to reach its reference state. On the other hand, for lower values it is more sensitive and tends to saturate its control inputs more often (δ_{T_1} and δ_{T_3}).

The level of turbulence used in the Dryden model was categorized as "Moderate" and its turbulence angular rates (q,r) are characterized as positive. The reference frame used was also the ENU. During the flight trajectory, the aircraft is seriously effected by turbulence, having its route oscillate significant values. For this aircraft's flight condition and for $\epsilon = 20$ or $\epsilon = 200$, there should not be a problem with handling qualities as $\frac{\omega_{phu}}{\omega_{sp}}$ is greater than 0.1 for both cases. The lower order of magnitude $\epsilon = 2$ has critically damped modes and so it was concluded that for lower control penalty parameter values (ϵ) the dynamic responses are impractical and unrealistic.

Summing up, it is safe to say that for the less penalized control system, as it weighs less in the performance index, it has a better overall dynamic response but at a higher control cost. As this script is supposed to work for any general engined aircraft, it finds the less penalized, by performance index, control system which has level one flying qualities. Sometimes it can be adequate to the aircraft flight condition, sometimes it cannot.

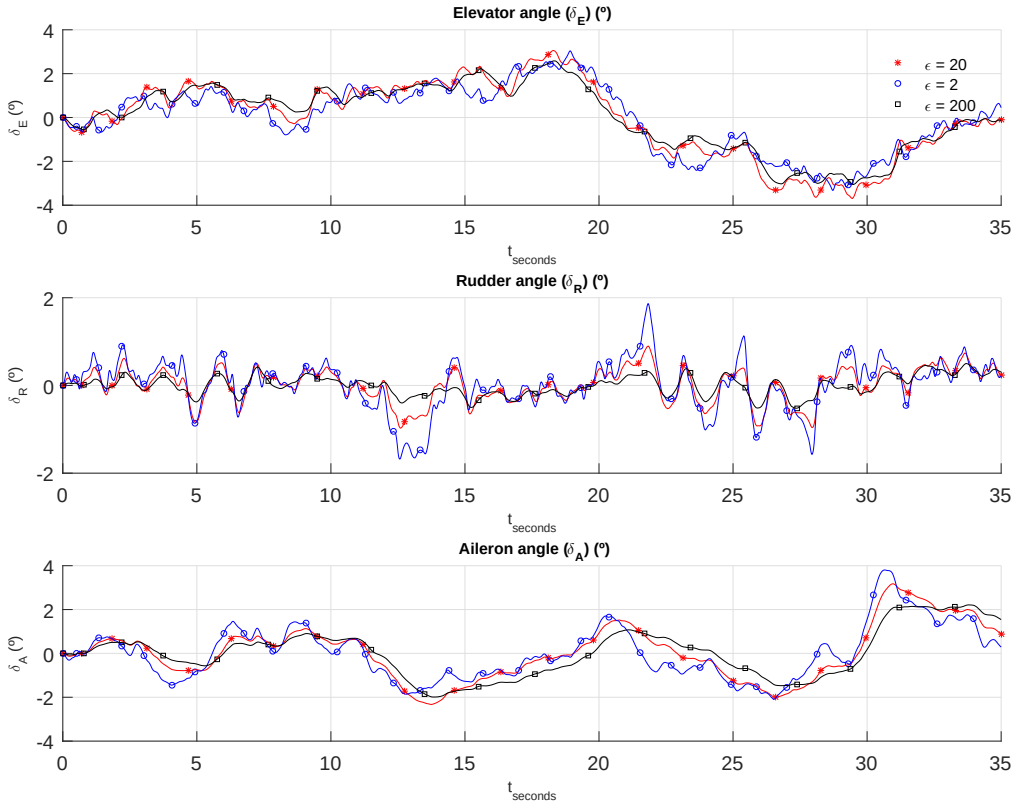

Figure 7.6: Dynamic responses of the input variables (δ_E,δ_A,δ_R)

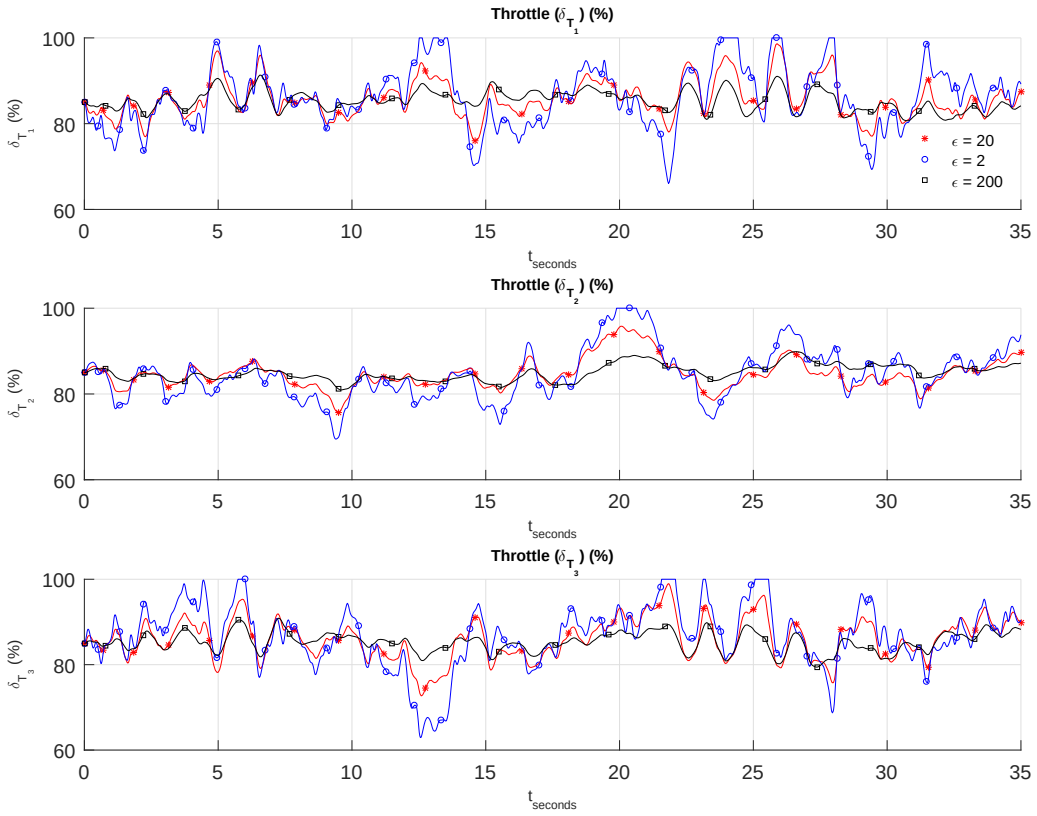

Figure 7.7: Dynamic responses of the engines (δ_{T_1},δ_{T_2},δ_{T_3})

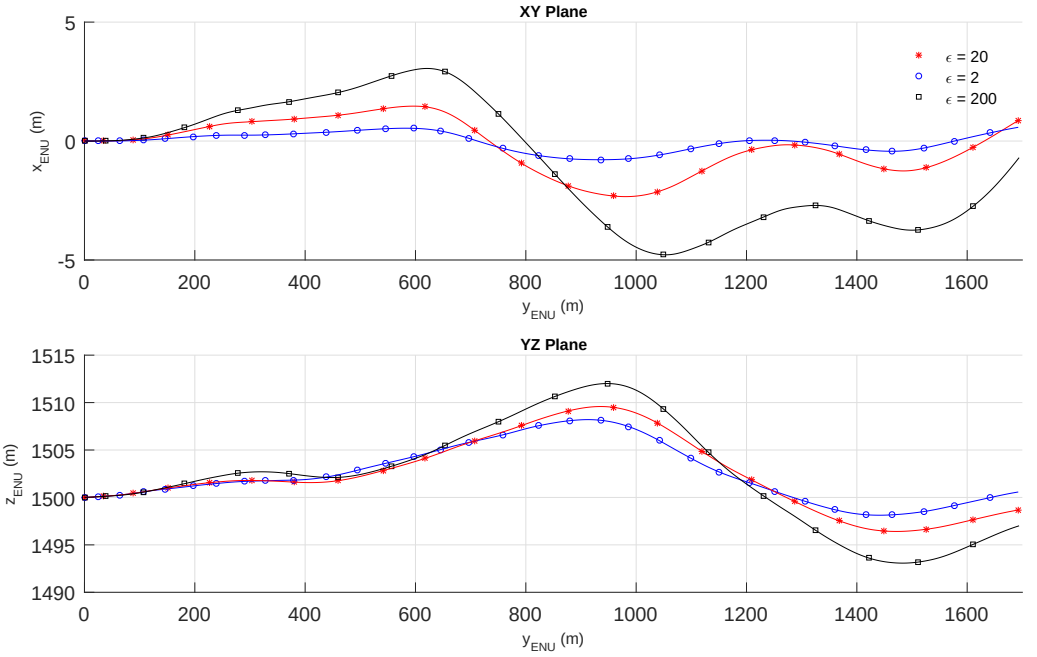

Figure 7.8: Turbulent flight trajectory seen in XY_{ENU} and YZ_{ENU} planes ($\gamma_{ref} = 1^o$, $\lambda_{ref} = 0^o$ and $u_{ref} = u_0$)

Chapter 8

Conclusions

This dissertation not only developed a crucial piece for a future aeroservoelastic tool but also a flight controller that can possibly be embedded into it.

The integrator results depend highly on the stepper used. For aircraft with low and medium manoeuvrability, an error stepper is recommended, although for high manoeuvrability aircraft a controlled stepper is the one to use as a result of overshooting responses and high variations in short periods of time.

The defined trim condition aeroservoelastic mathematical equations of motion are also left in a general state, as it is difficult to define the number of vibration modes required. This number of vibration modes rely upon not only on the approximation needed to define the structural influence on flight dynamics, but also on the aircraft to be studied.

The flight controller simulations and mathematical model results were as expected. However, the linear quadratic regulator as the control law is not always practical, serving nevertheless good use when designing a control tool for a general aircraft as in this dissertation. The flight simulator realizes the simulation based on the first control penalty parameter (ϵ) found that has level one flying qualities. For certain flight conditions it might not be the most appropriate, as it might cause overshooting and therefore create unrealistic simulations.

8.1 Achievements

During the present work I was able to accomplish the majority of the proposed objectives. The mathematical elastic equations of motion were successfully demonstrated for a equilibrium condition of a general aircraft.

Then, these equations were implemented as a C++ ® program which integrates these equations for a time interval (Δ_t), different integration techniques were also analysed for a practical case.

The flight simulator was successfully built, the implementation of control laws on the dynamics model. However, it was not possible to simulate using structural vibration modes due to lack of data about these. Two different simulations for two different aircraft were carried out in MATLAB® and SIMULINK®

environments, and with these simulations significant results, about the control method used and flight dynamics, were obtained.

8.2 Future Work

There is no doubt that the purpose of this work is to have continuation. The flight dynamics integrator is going to work as a fundamental role by functioning as a bridge between several disciplines such as aerodynamics, structures and control. The way these connections are going to be made is open to the user. The flight dynamics integrator observer argument has many possible diverse uses, in the manner that it can be used to write on a file, for interprocess communication, serial port writing, among others.

The flight dynamics integrator is ready to receive the necessitated values of other programs, however it is only doing one integration. The future use that is given to the flight dynamics integrator needs to take into account the recursivity, the program is left in its simplest state, only realizing one integration, to allow the future user to change it according to his objectives.

The flight dynamics integrator stepper was used based on one flight condition integration. When using it, it is important to take in consideration the stepper and the integration time interval, in order to gather adequate results.

The mathematical trim equations of motion for an aeroservoelastic model were also deduced, giving the possibility to be included in other projects or developed even more.

The idea of building a flight controller and a flight dynamics integrator for a general aircraft, made this dissertation a bit generalist and hard to, for instance, include sensors and structural data into the simulations. So, a future idea is also to concentrate on a certain aircraft, and build a full aeroservoelastic flight controller with different missions and manoeuvres by embedding the aeroelasticity characteristics in this flight control simulation tool.

Another interesting concept is create general structural dynamics and aerodynamics models and through the flight dynamics integrator test new aircraft designs and possibilities, in such way, the aeroservoelastic simulator would work as a research simulator.

Bibliography

[1] R. Bisplinghoff and H. Ashley. *Principles of Aeroelasticity*. Dover Phoenix Editions, 2002.

[2] A. R. Collar. The first fifty years of aeroelasticity, 1978.

[3] M. L. Baker, P. J. Goggin, and B. A. Winther. Aeroservoelastic modeling, analysis, and design techniques for transport aircraft. In *Structural Aspects of Flexible Aircraft Control*, 2000.

[4] M. Karpel. Procedures and models for aeroservoelastic analysis and design. *ZAMM - Journal of Applied Mathematics and Mechanics*, 2001.

[5] D. Allerton. *Principles of flight simulation*. Wiley, 2009.

[6] S. K. Advani. *The Kinematic Design of Flight Simulator Motion-Bases*. Delf University Press, 1998.

[7] J. R. Wright and J. E. Cooper. *Introduction to Aircraft Aeroelasticiy and Loads*. Wiley, 2007.

[8] D. L. York. Analysis of flutter and flutter suppression via an energy method. Master's thesis, Virgina Polytechnic Institute and State University, 1980.

[9] A. Tewari. *Aeroservoelasticity Modeling and Control*. Springer, 2015.

[10] R. Blockley and W. Shyy. *Encyclopedia of Aerospace Engineering*. Wiley, 2010.

[11] Y. C. Fung. *An Introduction to the theory of aeroelasticity*. John Wiley and Sons, 1955.

[12] J. Oliveira. Apontamentos de estabilidade de voo. Instituto Superior Técnico, 2013.

[13] H. Farrell and M. Barth. *The Global Positioning System and Inertial Navigation.* McGraw, 1999.

[14] J. Azinheira. Apontamentos de controlo de voo. Instituto Superior Técnico, 2013.

[15] P. Singla, D. Mortari, and J. Junkins. How to avoid singularity when using euler angles? Available online: http://lairs.eng.buffalo.edu/pdffiles/pconf/C10.pdf.

[16] F. P. Beer and E. R. Johnston. *Vector Mechanics for Engineers Dynamics*. McGraw-Hill, 2006.

[17] C. W. de Silva. *Sensors and Actuators: Engineering System Instrumentation*. CRC Press, 2015.

[18] E. Haskel. http://code7700.com/drag.html. Code 7700.

[19] B. Etkin. *Dynamics of Atmosferic Flight*. Dover Phoenix Editions, 2005.

[20] E. D. Achtmann. A critical review of world jet transport safety. Master's thesis, Massachusetts Institute of Technology, 1995.

[21] D. McClean. *Automatic Flight Control Systems*. Granada Publishing Limited, 1979.

[22] M. R. Waszak, J. B. Davidson, and D. K. Schmidt. *A Simulation Study of the Flight Dynamics of Elastic Aircraft*. NASA, 1987.

[23] R. D. Milne. Dynamics of the deformable aeroplane. Technical report, British Aeronautical Research Council, 1964.

[24] B. Raghavan. *Flight Dynamics and Control of Highly Flexible Flying-Wings*. PhD thesis, Faculty of the Virginia Polytechnic Institute and State University, 2009.

[25] R. C. Schwanz. Formulations of the equations of motion of an elastic aircraft for stability and flight control applications. Technical report, Airforce Flight Dynamics Laboratory, 1972.

[26] K. Ahnert and M. Mulansky. http://www.boost.org/doc/libs/. Boost C++ libraries.

[27] A. Klein and A. Godunov. *Introductory Computational Physics*. Cambridge University Press, 2006.

[28] MIL-F-8785. *Military Specification - Flying Qualities of Piloted Airplanes*. American Department of Defense, 1980.

[29] M. V. Cook. *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*. Butterworth-Heinemann, 2013.

[30] S. D. Olds. Modeling and lqr control of a two-dimensional airfoil. Master's thesis, Virgina Polytechnic Institute and State University, 1997.

[31] J. Theis. Robust control design for active flutter suppression. Master's thesis, University of Minnesota, 2016.

[32] A. Tewari. *Modern Control Design With MATLAB and SIMULINK*. Wiley, 2002.

[33] J. D. Robinson. A linear quadratic regulator weight selection algorithm. Master's thesis, Air Force Institute of Technology, 1990.

[34] R. Stengel. *Flight Dynamics*. Princeton University Press, 2004.

[35] J. Azinheira. Controlo de voo, enunciado dos projectos 2013-2014. Controlo de Voo, 2014.

Appendix A

Reference Frames, Stability Derivatives, Flying Qualities and Trim condition Aircraft data

A.1 Rotation Matrices

In order to be able to relate vectors from one reference frame to another, rotation matrices have to be introduced. Basically rotation matrices transform the components of a vector from a axis system to another.

In this section A.1, first the rotation matrices of three generalized axis sytem, and three angles will be defined. After this step is done the goal is reaching the final rotation matrix.

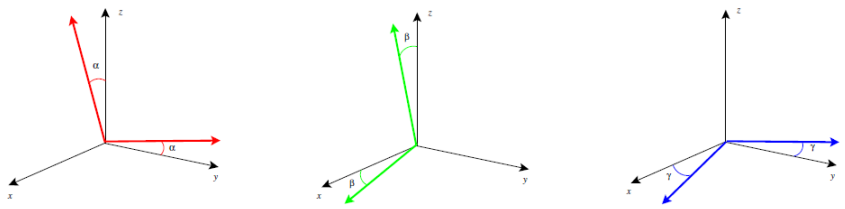

Figure A.1: Rotations around each of the generalized axis systems [12]

For one rotation of an angle α over the x-axis, the rotation matrix $\boldsymbol{L_x}$ is,

$$\boldsymbol{L_x}(\alpha) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos(\alpha) & sin(\alpha) \\ 0 & -sin(\alpha) & cos(\alpha) \end{bmatrix}, \tag{A.1}$$

rotation of an angle β over the y-axis, the rotation matrix $\boldsymbol{L_y}$ is,

$$\boldsymbol{L_y}(\beta) = \begin{bmatrix} cos(\beta) & 0 & -sin(\beta) \\ 0 & 1 & 0 \\ sin(\beta) & 0 & cos(\beta) \end{bmatrix}, \tag{A.2}$$

and finally the rotation of an angle γ over the z-axis, the rotation matrix $\boldsymbol{L_z}$ is,

$$\boldsymbol{L_z}(\gamma) = \begin{bmatrix} cos(\gamma) & sin(\gamma) & 0 \\ -sin(\gamma) & cos(\gamma) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \tag{A.3}$$

The rotation matrix of the fixed reference frame,F_E, to the body reference frame, F_B, is the multiplication (right to left) of the single rotation matrices corresponding to the Euler angles ($\boldsymbol{L_x}(\phi)$,$\boldsymbol{L_y}(\theta)$,$\boldsymbol{L_z}(\psi)$).

$$\boldsymbol{L_{BE}} = \boldsymbol{L_x}(\phi)\boldsymbol{L_y}(\theta)\boldsymbol{L_z}(\psi) \tag{A.4}$$

$$\boldsymbol{L_{BE}} = \begin{bmatrix} cos(\theta)cos(\psi) & cos(\theta)sin(\psi) & -sin(\theta) \\ sin(\phi)sin(\theta)cos(\psi) - cos(\phi)sin(\psi) & sin(\phi)sin(\theta)sin(\psi) + cos(\phi)cos(\psi) & sin(\phi)cos(\theta) \\ cos(\phi)cos(\theta)cos(\psi) + sin(\phi)sin(\psi) & cos(\phi)sin(\theta)sin(\psi) - sin(\phi)cos(\psi) & cos(\phi)cos(\theta) \end{bmatrix} \tag{A.5}$$

To obtain a vector in the earth fixed frame (NED) from the body axis system,

$$\boldsymbol{v_{EB}} = \boldsymbol{L_{EB}}\boldsymbol{v_{BE}} = \boldsymbol{L_{BE}}^{-1}\boldsymbol{v_{BE}}, \tag{A.6}$$

where,

$$\boldsymbol{L_{EB}} = \boldsymbol{L_{BE}}^{-1} = \begin{bmatrix} cos(\theta)cos(\psi) & sin(\theta)sin(\psi)cos(\psi) - cos(\phi)sin(\psi) & sin(\theta)cos(\psi)cos(\phi) + sin(\phi)sin(\psi) \\ sin(\psi)cos(\theta) & sin(\phi)sin(\theta)sin(\psi) + cos(\phi)cos(\psi) & cos(\phi)sin(\theta)sin(\psi) - sin(\phi)\cos(\psi) \\ -sin(\theta) & sin(\phi)cos(\theta) & cos(\phi)cos(\theta) \end{bmatrix} \tag{A.7}$$

A.1.1 East North Up Reference Frame

To facilitate the visualization of data, East-North-Up (ENU) reference frame is better than North-East-Down (NED). The ENU reference frame is achievable when multiplying the body axis frame vector by,

$$\boldsymbol{L_{E_{ENU}B}} = \begin{bmatrix} sin(\psi)cos(\theta) & sin(\phi)sin(\theta)sin(\psi) + cos(\phi)cos(\psi) & cos(\phi)sin(\theta)sin(\psi) - sin(\phi)\cos(\psi) \\ cos(\theta)cos(\psi) & sin(\theta)sin(\psi)cos(\psi) - cos(\phi)sin(\psi) & sin(\theta)cos(\psi)cos(\phi) + sin(\phi)sin(\psi) \\ +sin(\theta) & -sin(\phi)cos(\theta) & -cos(\phi)cos(\theta) \end{bmatrix} \tag{A.8}$$

A.2 Stability Derivatives

In this Section, lateral and longitudinal stability derivatives will be exposed according to [14]. The following lists resume the relation between stability derivatives and dimensionless coefficients. (Parameter, $\mu = \frac{m}{\rho S V_0}$, is introduced to simplify the representation. μ units is seconds.)

Longitudinal derivatives:

$$X_u = -(C_{D_u} + 2C_{D_0})\frac{1}{2\mu} \qquad Z_u = -(C_{L_u} + 2C_{L_0})\frac{1}{2\mu} \qquad M_u = C_{m_u}\frac{m\bar{c}}{2\mu I_y} \tag{A.9}$$

$$X_w = -(C_{D_\alpha} - C_{L_0})\frac{1}{2\mu} \qquad Z_w = -(C_{L_\alpha} + C_{D_0})\frac{1}{2\mu} \qquad M_w = C_{m_\alpha}\frac{m\bar{c}}{2\mu I_y} \tag{A.10}$$

$$Z_{\dot{w}} = -C_{L_\alpha}\frac{\bar{c}}{4\mu U_0} \qquad Z_q = -C_{L_q}\frac{\bar{c}}{4\mu U_0} \qquad M_q = C_{m_q}\frac{m\bar{c}^2}{4\mu I_y} \tag{A.11}$$

$$M_{\dot{w}} = C_{m_{\dot{\alpha}}}\frac{m\bar{c}^2}{4\mu U_0 I_y} \qquad Z_{\delta_E} = -C_{L_{\delta_E}}\frac{U_0}{2\mu} \qquad M_{\delta_E} = -C_{m_{\delta_E}}\frac{mU_0\bar{c}}{2\mu I_y} \tag{A.12}$$

Lateral derivatives:

$$Y_v = C_{Y_\beta}\frac{1}{2\mu} \qquad L_\beta = C_{l_\beta}\frac{mU_0 b}{2uI_x} \qquad N_\beta = C_{n_\beta}\frac{mU_0 b}{2\mu I_z} \tag{A.13}$$

$$Y_p = C_{Y_p}\frac{b}{4\mu} \qquad L_p = C_{l_p}\frac{mb^2}{4\mu I_x} \qquad N_p = C_{n_p}\frac{mb^2}{4\mu I_z} \tag{A.14}$$

$$Y_r = C_{Y_r}\frac{b}{4\mu} \qquad L_r = C_{l_r}\frac{mb^2}{4\mu I_x} \qquad N_r = C_{n_r}\frac{mb^2}{4\mu I_z} \tag{A.15}$$

$$Y_\delta = C_{Y_\delta}\frac{U_0}{2\mu} \qquad L_\delta = C_{l_\delta}\frac{mU_0 b}{2\mu I_x} \qquad N_\delta = C_{n_\delta}\frac{mU_0 b}{2\mu I_z} \tag{A.16}$$

A.3 Flying and Handling Qualities

In order to evaluate the level of flying qualities of a general aircraft's longitudinal and lateral dynamics modes, there are several factors to take in mind.

A.3.1 Aircraft Classes

There are four aircraft classes that an aircraft is considered to belong to, as seen in Table A.1.

Class	Aircraft characteristics
I	Small, light aircraft (max. weight $<$5 000 kg)
II	Aircraft of medium weight and moderate manoeuvrability (weight between 5 000 kg and 30 000 kg)
III	Large, heavy aircraft with moderate manoeuvrability (weight $>$30 000 kg)
IV	Aircraft with high manoeuvrability

Table A.1: Aircraft classes [21]

A.3.2 Flight Phases

There are three flight phases in which an aircraft can be during a mission. These phases are in Table A.2.

Flight phase	
A	All the non-terminal phases of flight, such require rapid manoeuvring, precision tracking and control of the flight path. (e.g., air-to-air combat, close reconnaissance, close formation flying)
B	Non-terminal phases of flight with gradual manoeuvres which do not require precision tracking. (e.g., climb, cruise, descent)
C	Terminal phases of flight, usually accomplished by gradual manoeuvres, but requiring accurate flight path control. (e.g., take-off, landing, approach)

Table A.2: Flight phases [21]

The simplified tables defining the necessary damping ratio and/or natural frequencies for every level of performance according to the MIL-F-8785 [21] are the following:

For longitudinal motion:

Level	ξ_{ph}
1	>0.04
2	>0
3	Period $>55s$

Table A.3: Phugoid flying qualities

Flight phase	Level 1	Level 2	Level 3
A	0.35 $<\xi<$ 1.3	0.25 $<\xi<$2	$\xi>$ 0.1
B	0.3 $<\xi<$ 2.0	0.2 $<\xi<$ 2	$\xi>0.1$
C	0.35 $<\xi<$ 1.3	0.35 $<\xi<$ 2	$\xi>$ 0.25

Table A.4: Short period flying qualities

For lateral motion:

Flight phase	Level 1	Level 2	Level 3
A,C	$>$12 s	$>$8 s	$>$5 s
B	$>$20 s	$>$8 2	$>$5 s

Table A.5: Spiral flying qualities

Although if the spiral has a negative real part, it is automatically considered stable and therefore the mode level 1.

Flight phase	Level 1	Level 2	Level 3
A I,IV	1.0	1.4	10
A II,III	1.4	3.0	10
B	1.4	3.0	10
C I,IV	1.0	1.4	10
C II,III	1.4	3.0	10

Table A.6: Rolling flying qualities

	Level 1			Level 2			Level 3		
Flight phase	ξ	ξw_n	w_n	ξ	ξw_n	w_n	ξ	ξw_n	w_n
A I,IV	0.19	0.35	1.0						
A II,III	0.19	0.35	0.5						
B	0.08	0.15	0.5	0.02	0.05	0.5	0.02	-	0.4
C I,IV	0.08	0.15	1.0						
C II,III	0.08	0.1	0.5						

Table A.7: Dutch roll flying qualities

A.3.3 Aircraft Data

These trim condition stability derivatives were taken from [35]. Some of the stability data for three aircraft are presented here. These aircraft are:

- **Airbus A400M**: a very large, four-engined, cargo jet aircraft;
- **Dassault Falcon 7X**: a three-engined, executive jet aircraft;
- **Embraer E120**: a two-engined, small passenger jet aircraft;

Table A.8: Aircraft general parameters

	Aircraft		
Parameters	Dassault Falcon 7X	Embraer E120	Airbus A400M
Altitude (m)	500	1500	1000
u_0 (m/s)	47.33	116.98	141.16
α_0 (º)	12.02	1.13	0.72
δ_{T_0}	85%	22%	29%
m (kgs)	116315	6501	111377
I_{xx} (kgm^2)	5152111	87466	7842288
I_{yy} (kgm^2)	1700195	74625	4655576
I_{zz} (kgm^2)	1965940	130077	10554844
I_{xz} (kgm^2)	50000	1566	0
b (m)	51.66	17.42	42.39
$\delta_{E_{min/max}}$ (º)	-21/28	-15/17	-20/20
$\delta_{A_{min/max}}$ (º)	-19/19	-18/18	-20/20
$\delta_{R_{min/max}}$ (º)	-23/23	-30/30	-20/20
N_{eng}	3	2	4
engine positions relative to c.m. $(x, y, z)_i$ (m) (estimatives)	$(-5, 1.5, -0.0644)_1$ $(-5, 0, 1)_2$ $(-5, -1.5, -0.0644)_3$	$(0, 2.5, -0.0132)_1$ $(0, -2.5, -0.0132)_2$	$(0, 12.86, 0.26)_1$ $(0, 6.7, -0.24)_2$ $(0, -6.73, -0.24)_3$ $(0, -12.86, 0.26)_4$
γ_0 (º)	0	0	0

Table A.9: Aircraft trim derivatives

	Aircraft		
Parameters	Dassault Falcon 7X	Embraer E120	Airbus A400M
X_u	-0.0990	-0.0285	-0.0133
X_w	0.1370	0.0277	0.0609
X_{δ_E}	0	0	0
X_{δ_T}	3.616	5.679	2.452
Z_u	-0.3995	-0.1683	-0.1389
Z_w	-0.5014	-0.0168	-0.9069
Z_q	-7.3078	-4.7616	-3.3658
Z_{δ_E}	-0.838	-12.498	22.933
Z_{δ_T}	0	0	0
M_u	0	0	0
M_w	-0.0752	-0.1366	-0.0634
M_q	-1.5326	-0.6368	-2.2674
$M_{\dot{w}}$	-0.0436	-0.0018	0.0119
M_{δ_E}	-3.128	-5.977	-3.577
M_{δ_T}	0.105	-0.075	-0.098
Y_β	-0.0416	-0.1571	-0.1213
Y_p	0.0012	0	0
Y_r	0.0169	0.0071	0.0081
Y_{δ_A}	0	0	0
Y_{δ_R}	-0.013	-0.025	-0.029
N_β	17.6847	4.0971	3.2053
N_p	-0.0622	-0.0631	0
N_r	-0.4575	-0.2543	-0.2130
N_{δ_A}	0	0	0.182
N_{δ_R}	-1.623	-1.407	-1.710
L_β	-0.8563	-4.0621	-5.4558
L_p	-2.4016	-1.1486	-1.7601
L_r	0.4745	0.4538	-0.0292
L_{δ_A}	-2.029	-7.353	-2.547
L_{δ_R}	-0.043	-0.630	0.041

Appendix B

Aircraft Simulation Data

B.1 LQR Q and R Matrices

The engine failure LQR matrices are submitted to the linear quadratic script of 6.3 providing the x_{max} and u_{max} values, and therefore the Q_{final} matrix,

$$\boldsymbol{x_{max}} = \begin{bmatrix} 0.0169 & 0.0330 & 0.0025 & 0.0015 & 0.0650 & 0.0071 & 0.0470 & 0.0433 & 0.0923 & 0.0622 & 0.0181 & 0.0826 \end{bmatrix}^T, \tag{B.1}$$

$$Q_{final} = \begin{bmatrix} 3510 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 920 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 163220 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 455760 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 240 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 19580 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 450 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 530 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 120 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 260 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 3050 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 150 \end{bmatrix}, \tag{B.2}$$

and the R_{final} matrix,

$$\boldsymbol{u_{max}} = \begin{bmatrix} 0.0349 & 0.1000 & 0.1000 & 0.1000 & 0.1000 & 0.0349 & 0.0349 \end{bmatrix}^T, \tag{B.3}$$

$$R_{final} = \begin{bmatrix} 820.5410 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 820.5410 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 820.5410 \end{bmatrix}. \tag{B.4}$$

For the turbulence simulation, the x_{max} and u_{max} values, and the Q_{final} and R_{final} matrices, found by the script are,

$$\boldsymbol{x_{max}} = \begin{bmatrix} 0.0876 & 0.0994 & 0.0122 & 0.0137 & 0.1497 & 0.0631 & 0.0057 & 0.01440.0116 & 0.0206 & 0.0077 & 0.0326 \end{bmatrix}^T, \tag{B.5}$$

$$\boldsymbol{u_{max}} = \begin{bmatrix} 0.0489 & 0.1000 & 0.1000 & 0.100 & 0.0332 & 0.0401 \end{bmatrix}^T, \tag{B.6}$$

$$Q_{final} = \begin{bmatrix} 130 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 101 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 6765 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 5367 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 45 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 251 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 30560 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 4792 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 7390 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 2355 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 16747 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 942 \end{bmatrix}, \tag{B.7}$$

$$R_{final} = \begin{bmatrix} 418.72 & 0 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 & 0 \\ 0 & 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 909.37 & 0 \\ 0 & 0 & 0 & 0 & 0 & 620.57 \end{bmatrix}. \tag{B.8}$$

B.2 LQR Gain Matrix

The engine failure gain matrix associated with the value the script found ($\rho = 10$) utilizing the matrices (B.2) and (B.2) is,

$$K_{LQR} = \begin{bmatrix} 0.4015 & -0.2383 & -6.4192 & -22.2477 & 0.0664 & -1.6801 & 0.0313 & 0.0104 & 0.0220 & 0.0243 & 0.0559 & 0.0023 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 2.2125 & -0.1379 & -2.7448 & -15.1224 & 0.4291 & 1.9008 & 0.7640 & -0.1407 & -0.1431 & -0.1801 & -1.0060 & -0.1232 \\ -0.7070 & 0.1320 & 2.5032 & 9.7677 & -0.1128 & -0.7113 & 1.5885 & -0.4390 & -0.6906 & -0.6607 & -3.1044 & -0.3490 \\ -0.1092 & 0.0008 & -0.0394 & 1.8940 & -0.0181 & -0.0582 & 0.1099 & -0.0700 & -0.0246 & -0.0788 & -0.2115 & -0.0221 \\ -0.2127 & 0.0010 & -0.0658 & 3.1405 & -0.0392 & -0.0330 & 0.0501 & -0.0192 & -0.0365 & -0.0370 & -0.2408 & -0.0278 \end{bmatrix}, \tag{B.9}$$

The control penalty parameter for which the script first found level one flying qualities, in the turbulence simulation, was for $\rho = 20$. The computed gain matrix for this case is, taking in account matrices (B.7) and (B.8),

$$K_{LQR} = \begin{bmatrix} 0.1311 & -0.0311 & -1.1820 & -1.9993 & 0.0578 & -0.1048 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1704 & 0.0059 & -0.3148 & -0.1317 & 0.0644 & 0.1929 & -0.4609 & 0.0833 & 1.3167 & 0.2392 & 1.1593 & 0.1684 \\ 0.0078 & 0.1384 & 2.3626 & -0.0199 & 0.0070 & 0.0723 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0.1704 & 0.0059 & -0.3148 & -0.1317 & 0.0644 & 0.1929 & 0.4609 & -0.0833 & -1.3167 & -0.2392 & -1.1593 & -0.1684 \\ 0 & 0 & 0 & 0 & 0 & 0 & -0.4194 & -0.5271 & -0.0144 & -0.8739 & -1.5349 & -0.2092 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0.0069 & -0.0527 & -0.1918 & -0.1024 & -0.3518 & -0.0516 \end{bmatrix}. \tag{B.10}$$