Flight dynamics and simulation of a generic aircraft for aeroservoelastic design

Pedro Tomás Marques Martins Margarido
pedro.tomas.margarido@tecnico.ulisboa.pt

Instituto Superior Técnico, Lisboa, Portugal

November 2016

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 of them 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

1. Introduction

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).

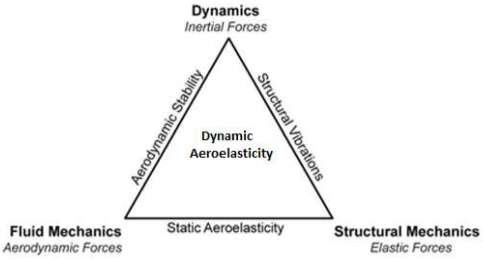

Figure 1: Aeroelasticity (adapted from [4])

As seen in Figure 1, 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.[3]. That is why aeroservoelasticity plays an important role on controlling and preventing these harmful effects from happening.

The goal of the dynamics integration tool is to guarantee future integration with aerodynamics, structures and flight control programs to simulate a generic aircraft dynamics during a time interval (Δt). The grey highlighted boxes in Figure 2 are the modules done and described throughout this thesis.

2. Theoretical Background

Several key disciplines such as flight simulators, aeroelasticity, aeroservoelasticity and mathematical methods used to simulate unsteady aerodynamics and structural dynamics are briefly covered in this section.

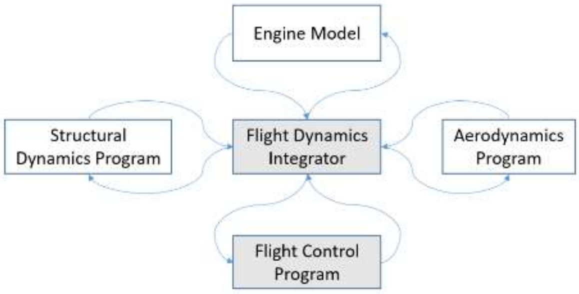

Figure 2: Interactions between modules

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. As seen in Figure 3, 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. 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).

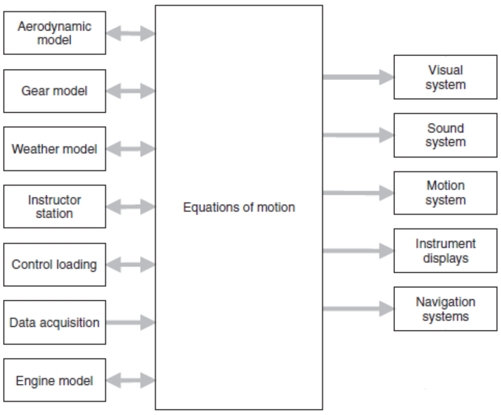

Figure 3: General structure from a flight simulator

2.2. Aeroelasticity

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. 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) [3]. Some of the most catastrophic phenomena are:

- **Flutter**: 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;
- **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.

2.3. Aeroservoelasticity

Aeroservoelasticity (ASE) 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]. A possible block diagram for the aeroservoelasticity is seen in Figure 4.

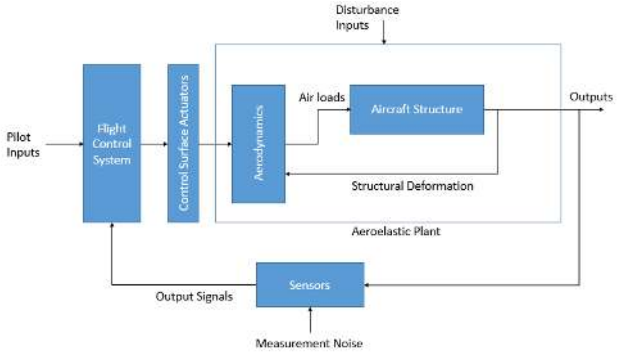

Figure 4: General aeroservoelastic block diagram (adapted from [9])

Deformation happens or is usually increased when there are gusts (disturbance input) or control surface deflection, as seen in the aeroelasticity plant from Figure 4. 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 in the 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. The flutter dynamic pressure (p_{dF}) has an associated speed called flutter velocity

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

The goal of an aeroservoelastic model, is to close the loop in order to increase the open-loop flutter velocity [9].

3. Dynamics Model

In this Section 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.

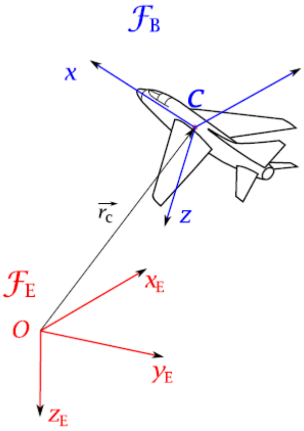

Figure 5: Fixed reference frame, F_E, and aircraft reference frame, F_B [8]

Before advancing to the definition of the equations of motion, two reference frames, as seen in the Figure 5, need to be chosen. The local NED (North-East-Down) coordinate system was chosen for fixed frame (F_E) and the RPY (Roll-Pitch-Yaw) was picked for the body axis system (F_B). Through Euler angles (ϕ,ψ,θ) the transformation from the NED frame to the RPY frame is intuitive [8].

3.2. Rigid Body Flight Dynamics

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

$$\boldsymbol{F} = \frac{d}{dt}[m\boldsymbol{v_E}] \, , \tag{2}$$

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

$$\boldsymbol{M} = \frac{d}{dt}[\boldsymbol{H}]_E = \frac{d}{dt}[\boldsymbol{I}\boldsymbol{w}]_E = \frac{d}{dt}\boldsymbol{I}\boldsymbol{w_E} \, , \tag{3}$$

where $\boldsymbol{M}$ represents the resultant external moment, $\boldsymbol{H}$ is the total moment relative to the aircraft's center of mass. $\boldsymbol{H}$ is equal to the product of $\boldsymbol{w}$, the angular velocity vector, with $\boldsymbol{I}$ being the inertia tensor matrix. The small disturbance theory when, applied to the rigid body flight dynamics in a steady state, as rectilinear flight, is a powerful tool that decouples the motion into variables responsible for the longitudinal and lateral motion. Considering single engine contribution for forces and moments, (2) and (3) expand into two decoupled set of equations, one for longitudinal motion

$$\begin{cases} \dot{u} = X_u u - w_0 q + X_w w - gcos(\theta_0)\theta + X_{\delta_E}\delta_E \\ \qquad + \sum_{i=1}^{N_{eng}} X_{\delta_{T\,i}}\delta_{T\,i} \\ \dot{w} = Z_u u + u_0 q + Z_w w - gsin(\theta_0)\theta + Z_{\delta_E}\delta_E \\ \qquad + \sum_{i=1}^{N_{eng}} Z_{\delta_{T\,i}}\delta_{T\,i} \\ \dot{q} = \tilde{M_u} u + \tilde{M_w} w + \tilde{M_q} q + \tilde{M_\theta}\theta + \tilde{M_{\delta_E}}\delta_E \\ \qquad + \sum_{i=1}^{N_{eng}} \delta_{T\,i}(z_i X_{\delta_{T\,i}} - x_i Z_{\delta_{T\,i}}) \\ \dot{\theta} = q \end{cases} , \tag{4}$$

and other for lateral motion

$$\begin{cases} \dot{\beta} = Y_\beta\beta + p(\frac{Y_p}{u_0} + \alpha_0) + r(\frac{Y_r}{u_0} - 1) + \frac{gcos(\theta_0)}{u_0}\phi \\ \qquad + \frac{Y_{\delta_A}}{u_0}\delta_A + \frac{Y_{\delta_R}}{u_0}\delta_R \\ \dot{p} = L'_\beta\beta + L'_p p + L'_r r + L'_{\delta_A}\delta_A + L'_{\delta_R}\delta_R - \\ \qquad \sum_{i=1}^{N_{eng}} y_i Z_{\delta_{T\,i}}\delta_{T\,i} \\ \dot{r} = N'_\beta\beta + N'_p p + N'_r r + N'_{\delta_A}\delta_A + N'_{\delta_R}\delta_R \\ \qquad + \sum_{i=1}^{N_{eng}} y_i X_{\delta_{T\,i}}\delta_{T\,i} \\ \dot{\phi} = p + tan(\theta_0)r \\ \dot{\psi} = \frac{r}{cos(\theta_0)} \end{cases} , \tag{5}$$

where X_i, Y_i and Z_i represent ith state variable induced force on the x, y and z axis respectively. The L_i, M_i and N_i represent ith state variable induced moment on the x, y and z axis, respectively.

3.3. Elastic Aircraft Consideration

When aeroelastic effects are taken into account, new state variables and their respective equations, representing a set of generalized coordinates associated with the bending modes need to be added to the flight dynamics equations system, (4) and (5). The vibration (bending, torsion, mixed, among others) modes can be represented using generalized coordinates

$$c_{1i}\ddot{q}_i + c_{2i}\dot{q}_i + c_{3i}q_i = F_i \tag{6}$$

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. [5] Although it is possible to represent the vibration mode by two first order, linear, differential equations

$$x_1 = q_i \, , \qquad x_2 = \dot{q}_i \, , \tag{7}$$

where

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= \frac{-c_{2i}}{c_{1i}} x_2 - \frac{c_{3i}}{c_{1i}} x_1 + \frac{1}{c_{1i}} F_i \,. \end{aligned} \tag{8}$$

This pair of first order differential equations 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.

Assuming the conditions of the flight dynamics system (4) and (5) and the state vectors for longitudinal and lateral motions represented in equations (9) and (10)

$$\boldsymbol{x_{long}} = \begin{bmatrix} u & w & q & \theta & \lambda_1 & \sigma_1 & ... & \lambda_n & \sigma_n \end{bmatrix}^T \,, \tag{9}$$

$$\boldsymbol{x_{lat}} = \begin{bmatrix} \beta & p & r & \phi & \psi & \tau_1 & \chi_1 & ... & \tau_n & \chi_n \end{bmatrix}^T \,, \tag{10}$$

the flexibility effects of a general aircraft, for n vibration modes, can be represented as

$$\begin{cases} \dot{u} = X_u u - w_0 q + X_w w - gcos(\theta_0)\theta + X_{\delta_E}\delta_E \\ \quad + \sum_{i=1}^{N_{eng}} X_{\delta_{T\,i}}\delta_{T\,i} + X_{\lambda_1}\lambda_1 + X_{\sigma_1}\sigma_1 + ...+ \\ \quad X_{\lambda_n}\lambda_n + X_{\sigma_n}\sigma_n \\ \dot{w} = Z_u u + u_0 q + Z_w w - gsin(\theta_0)\theta + Z_{\delta_E}\delta_E + \\ \quad \sum_{i=1}^{N_{eng}} Z_{\delta_{T\,i}}\delta_{T\,i} + Z_{\lambda_1}\lambda_1 + Z_{\sigma_1}\sigma_1 + ...+ \\ \quad Z_{\lambda_n}\lambda_n + Z_{\sigma_n}\sigma_n \\ \dot{q} = \tilde{M_u} u + \tilde{M_w} w + \tilde{M_q} q + \tilde{M_\theta}\theta + \tilde{M_{\delta_E}}\delta_E + \\ \quad \sum_{i=1}^{N_{eng}} \delta_{T\,i}(z_i X_{\delta_{T\,i}} - x_i Z_{\delta_{T\,i}}) + M_{\lambda_1}\lambda_1 + \\ \quad M_{\sigma_1}\sigma_1 + ... + M_{\lambda_n}\lambda_n + M_{\sigma_n}\sigma_n \\ \dot{\theta} = q \\ \dot{\lambda_1} = \sigma_1 \\ \dot{\sigma_1} = -(2\xi_1\omega_1 + \eta_{1_{\sigma_1}})\sigma_1 + (-{\omega_1}^2 + \eta_{1_{\lambda_1}})\lambda_1 + \\ \quad \eta_{1_u} u + \eta_{1_w} w + \eta_{1_q} q + \eta_{1_{\delta_E}}\delta_E + \\ \quad \sum_{i=1}^{N_{eng}} \eta_{1_{\delta_{T\,i}}}\delta_{T\,i} + ... + \eta_{1_{\sigma_n}}\sigma_n + \eta_{1_{\lambda_n}}\lambda_n \\ ... \\ \dot{\lambda_n} = \sigma_n \\ \dot{\sigma_n} = -(2\xi_n\omega_n + \eta_{n_{\sigma_n}})\sigma_n + (-{\omega_n}^2 + \eta_{n_{\lambda_n}})\lambda_n \\ \quad + \eta_{n_u} u + \eta_{n_w} w + \eta_{n_q} q + \eta_{n_{\delta_E}}\delta_E + \\ \quad \sum_{i=1}^{N_{eng}} \eta_{n_{\delta_{T\,i}}}\delta_{T\,i} + \eta_{n_{\sigma_1}}\sigma_1 + \eta_{n_{\lambda_1}}\lambda_1 + ...+ \\ \quad \eta_{n_{\sigma_{n-1}}}\sigma_{n-1} + \eta_{n_{\lambda_{n-1}}}\lambda_{n-1} \end{cases} , \tag{11}$$

$$\begin{cases} \dot{\beta} = Y_\beta\beta + p(\frac{Y_p}{u_0} + \alpha_0) + r(\frac{Y_r}{u_0} - 1) + \frac{gcos(\theta_0)}{u_0}\phi \\ \quad + \frac{Y_{\delta_A}}{u_0}\delta_A + \frac{Y_{\delta_R}}{u_0}\delta_R + Y_{\tau_1}\tau_1 + Y_{\chi_1}\chi_1 + ...+ \\ \quad Y_{\tau_n}\tau_n + Y_{\chi_n}\chi_n \\ \dot{p} = L'_\beta\beta + L'_p p + L'_r r + L'_{\delta_A}\delta_A + L'_{\delta_R}\delta_R - \\ \quad \sum_{i=1}^{N_{eng}} y_i Z_{\delta_{T\,i}}\delta_{T\,i} + L_{\tau_1}\tau_1 + L_{\chi_1}\chi_1 + ...+ \\ \quad L_{\tau_n}\tau_n + L_{\chi_n}\chi_n \\ \dot{r} = N'_\beta\beta + N'_p p + N'_r r + N'_{\delta_A}\delta_A + N'_{\delta_R}\delta_R + \\ \quad \sum_{i=1}^{N_{eng}} y_i X_{\delta_{T\,i}}\delta_{T\,i} + N_{\tau_1}\tau_1 + N_{\chi_1}\chi_1 + ... \\ \quad + N_{\tau_n}\tau_n + N_{\chi_n}\chi_n \\ \dot{\phi} = p + tan(\theta_0) r \\ \dot{\psi} = \frac{r}{cos(\theta_0)} \\ \dot{\tau_1} = \chi_1 \\ \dot{\chi_1} = -2\xi_A\omega_A\chi_1 + -{\omega_A}^2\tau_1 + \mu_{1_\beta}\beta + \mu_{1_p} p + \\ \quad \mu_{1_r} r + \mu_{1_{\delta_A}}\delta_A + \mu_{1_{\delta_R}}\delta_R + ... + \mu_{1_{\chi_n}}\chi_n \\ \quad + \mu_{1_{\tau_n}}\tau_n \\ ... \\ \dot{\tau_n} = \chi_n \\ \dot{\chi_n} = -2\xi_Z\omega_Z\chi_n + -{\omega_Z}^2\tau_n + \mu_{n_\beta}\beta + \mu_{n_p} p + \\ \quad \mu_{n_r} r + \mu_{n_{\delta_A}}\delta_A + \mu_{n_{\delta_R}}\delta_R + \mu_{n_{\chi_1}}\chi_1 + \mu_{n_{\tau_1}}\tau_1 \\ \quad + ... + \mu_{n_{\chi_{n-1}}}\chi_{n-1} + \mu_{n_{\tau_{n-1}}}\tau_{n-1} \end{cases} , \tag{12}$$

where λ and τ represent the displacement of symmetrical and asymmetrical bending mode, μ_{n_τ} and η_{n_λ} correspond to the structural derivatives with respect to the n_λ and n_τ bending 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.

4. Dynamics Model Implementation

One of the goals of this thesis, besides the flight controller, is the implementation of a standalone aircraft dynamics client. The dynamics equations for this C++® program were equations (11) and (12).

The center piece of this developed 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 [1].

The integration function was *integrate_adaptive*. 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, a starting state x_0 and the *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.

4.1. Inputs and Outputs

This client will receive information from several modules such as: aerodynamics, structural dynamics, flight controller and propulsion model. Then, it will be possible to simulate the dynamics during a time interval (Δt). The objective is to supply the other modules with the aircraft's trajectory (x_E,y_E,z_E), velocities (u_B,v_B,w_B), accelerations (a_{x_B},a_{y_B},a_{z_B}), euler angles (ϕ,ψ,θ) and angular rates (p,q,r) during that time interval.

4.2. Type of Steppers

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_karp*54;
- **Error steppers**: Steppers that provide an error estimation. Besides also being a basic stepper, *runge_kutta_cash_karp*54 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. The *controlled_runge_kutta* will be the considered controlled stepper.

4.3. Stepper Comparison

For the stepper comparison test, the trim condition stability derivatives data from the three engined Dassault Falcon 7X, was used. The standard integration solver used in SIMULINK$^{®}$, the Dormand-Price, which is a controlled stepper, will be used as a reference.

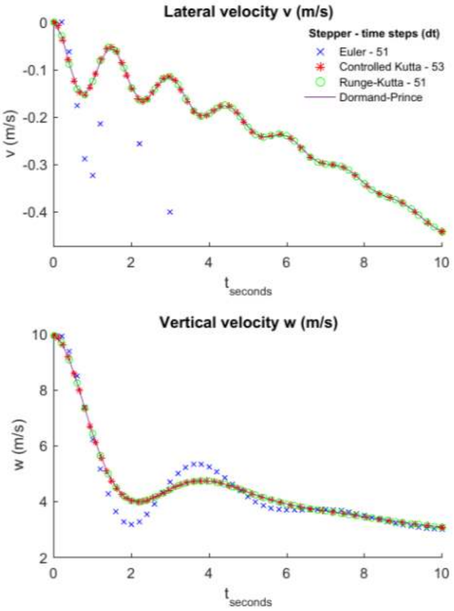

Figure 6: Dynamic response of the Dassault Falcon 7X (v,p,w,q)

In Figure 6, it is exposed the integration results of the Dassault Falcon 7X in a certain flight condition (Dassault Falcon 7X $\delta_A = 5°$, $\delta_R = 0°$, $\delta_E = 5°$, no engine throttle and null initial state conditions). 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 dynamics integration problem.

The Euler method possesses significant accuracy problems as it only corresponds to the two first terms in the Taylor series, these accuracy problems are visible in the w plot. 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 plot.

The 4th order Runge-Kutta integration scheme shows suitable results for the dynamics of this flight condition. Although if the system has certain initial condition or states for the control variables, it can overshoot.

The controlled Kutta stepper internally varies the time step size. However, it has a considerable setback which is that the user has no power in the choice of the controlled stepper time step. As the 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 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 for controlled stepper might appear, especially, for aircraft that have overshooting responses.

5. Flight Control

In this Section the requirements for flight control and the final model state forms will be presented.

5.1. Simulation Domain

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

$$\begin{cases} \dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \\ \boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{D}\boldsymbol{u} \end{cases} . \tag{13}$$

The first equation in (13) 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 (13) 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 [5]. The stability of the system is verified by looking at the eigenvalues of the state coefficient matrix 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. Longitudinal Control

For the lateral mode the control will be done in the flight path angle (γ) and longitudinal speed (u). The flight path angle (γ) needs to be added as a state, and keeping in mind that the vertical velocity (w) can be approximated, for small perturbations, as a function of the angle of attack (α)

$$\gamma = \theta - \alpha \rightarrow \theta = \gamma + \frac{w}{u_0}\,. \tag{14}$$

When transforming the system (4) into state-space and substituting , the pitch angle (θ) for flight path (γ), the longitudinal state-space emerges

$$\dot{\boldsymbol{x}}_{\boldsymbol{Long_\gamma}} = \boldsymbol{A}_{\boldsymbol{Long_\gamma}}\boldsymbol{x}_{\boldsymbol{Long_\gamma}} + \boldsymbol{B}_{\boldsymbol{Long_\gamma}}\boldsymbol{u}_{\boldsymbol{Long}} =$$
$$\begin{bmatrix} X_u & X_w - \frac{gcos(\theta_0)}{u_0} & -w_0 & -gcos(\theta_0) \\ Z_u & Z_w - \frac{gsin(\theta_0)}{u_0} & u_0 & -gsin(\theta_0) \\ \tilde{M}_u & \tilde{M}_w + \frac{\tilde{M}_\theta}{u_0} & \tilde{M}_q & \tilde{M}_\theta \\ -\frac{Z_u}{u_0} & -\frac{Z_w}{u_0} + \frac{gsin(\theta_0)}{u_0^2} & 0 & \frac{gsin(\theta_0)}{u_0} \end{bmatrix} \begin{bmatrix} u \\ w \\ q \\ \gamma \end{bmatrix} +$$
$$\begin{bmatrix} X_{\delta_E} & X_{\delta_{T1}} & ... & X_{\delta_{TN_{eng}}} \\ Z_{\delta_E} & Z_{\delta_{T1}} & ... & Z_{\delta_{TN_{eng}}} \\ \tilde{M_{\delta_E}} & (z_i X_{\delta_{T1}} - x_i Z_{\delta_{T1}}) & ... & (z_{N_{eng}} X_{\delta_{TN_{eng}}} - x_{N_{eng}} Z_{\delta_{TN_{eng}}}) \\ \frac{-Z_{\delta_E}}{u_0} & \frac{-Z_{\delta_{T1}}}{u_0} & ... & \frac{-Z_{\delta_{TN_{eng}}}}{u_0} \end{bmatrix} \boldsymbol{u}_{\boldsymbol{Long}}\,, \tag{15}$$

where

$$\boldsymbol{u}_{\boldsymbol{Long}} = \begin{bmatrix} \delta_E & \delta_{T1} & ... & \delta_{TN_{eng}} \end{bmatrix}^T\,. \tag{16}$$

Two additional states (x_u and x_γ) were added to prevent static error on the longitudinal controllable states, consequently increasing the size of system's (15) $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{x}$.

5.3. Lateral Control

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)

$$\lambda = \beta + \psi\,. \tag{17}$$

This will be the fifth lateral state substituting the yaw angle (ψ). The lateral equations (5) are then transformed into the state-space

$$\dot{\boldsymbol{x}}_{\boldsymbol{Lat_\lambda}} = \boldsymbol{A}_{\boldsymbol{Lat_\lambda}}\boldsymbol{x}_{\boldsymbol{Lat_\lambda}} + \boldsymbol{B}_{\boldsymbol{Lat_\lambda}}\boldsymbol{u}_{\boldsymbol{Lat}} =$$
$$\begin{bmatrix} Y_\beta & \frac{Y_p}{u_0} + \alpha_0 & \frac{Y_r}{u_0} - 1 & \frac{gcos(\theta_0)}{u_0} & 0 \\ L'_v & L'_p & L'_r & 0 & 0 \\ N'_v & N'_p & N'_r & 0 & 0 \\ 0 & 1 & tan(\theta_0) & 0 & 0 \\ Y_\beta & \frac{Y_p}{u_0} + \alpha_0 & \frac{1}{cos(\theta_0)} + \frac{Y_r}{u_0} - 1 & \frac{gcos(\theta_0)}{u_0} & 0 \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \\ \phi \\ \lambda \end{bmatrix}$$
$$+ \begin{bmatrix} \frac{Y_{\delta_A}}{u_0} & 0 & ... & 0 & \frac{Y_{\delta_R}}{u_0} \\ L'_{\delta_A} & -y_1 Z_{\delta_{T1}} & ... & -y_{N_{eng}} Z_{\delta_{TN_{eng}}} & L'_{\delta_R} \\ N'_{\delta_A} & y_1 X_{\delta_{T1}} & ... & y_{N_{eng}} X_{\delta_{TN_{eng}}} & N'_{\delta_R} \\ 0 & 0 & ... & 0 & 0 \\ \frac{Y_{\delta_A}}{u_0} & 0 & ... & 0 & \frac{Y_{\delta_R}}{u_0} \end{bmatrix} \boldsymbol{u}_{\boldsymbol{Lat}}\,. \tag{18}$$

where

$$\boldsymbol{u}_{\boldsymbol{Lat}} = \begin{bmatrix} \delta_A & \delta_{T1} & ... & \delta_{TN_{eng}} & \delta_R \end{bmatrix}^T\,. \tag{19}$$

An additional state (x_λ) was added to prevent static error on the heading angle (λ), consequently increasing the size of system's (18) $\boldsymbol{A}$, $\boldsymbol{B}$ and $\boldsymbol{x}$.

5.4. Flying and Handling Qualities

Aircraft flying qualities are defined by a number of parameters in the complex frequency domain. In equation (20) there are two of these important parameters, damping ratio (ξ) and undamped natural frequency (ω_n).

$$\begin{aligned} \omega_n &= |\kappa| \\ \xi &= -cos(\angle\kappa) \end{aligned} \tag{20}$$

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 depends upon the aircraft class and flight phase [6].

5.5. Disturbances State-space Form

To include the disturbances, a new matrix is added into the aircraft dynamics state-space form (13)

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} + \boldsymbol{E}\boldsymbol{d}\,, \tag{21}$$

where $\boldsymbol{d}$ represents the disturbance states

$$\boldsymbol{d}_{\boldsymbol{coupled}} = \begin{bmatrix} \boldsymbol{d}_{\boldsymbol{Long}} & \boldsymbol{d}_{\boldsymbol{Lat}} \end{bmatrix}^T \tag{22}$$
$$= \begin{bmatrix} u_g & w_g & q_g & v_g & p_g & r_g \end{bmatrix}^T\,. \tag{23}$$

and $\boldsymbol{E}$ the associated disturbance influence matrix.

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 7, satisfies the state-space equation (21) where the $\boldsymbol{A}$ and the $\boldsymbol{B}$ matrices are formed by the matrices of equations (15) and (18).

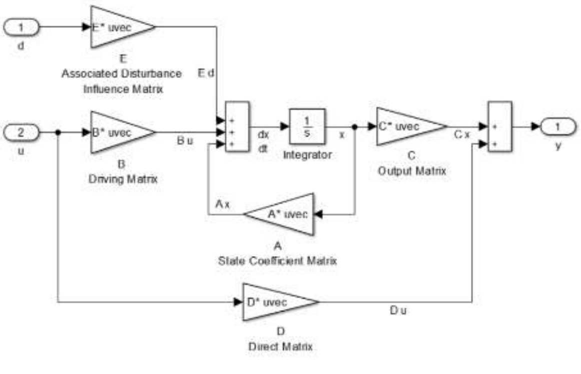

Figure 7: State-Space model implemented in SIMULINK®

6. Optimal Control

The goal is to examine the optimal control technique used for the flight controller. An example of flutter suppression on a two-dimensional aeroelastic airfoil will be used to demonstrate the utility of the linear quadratic regulator.

6.1. Aeroservoelastic Optimal Control

To introduce the linear quadratic regulator (LQR), a flutter suppression controller of a two-dimensional aeroelastic airfoil represented in Figure 8, will be demonstrated.

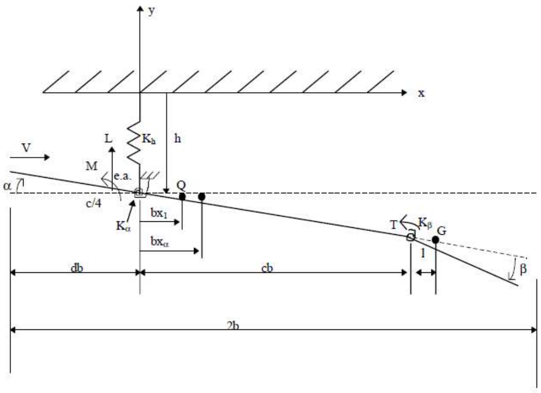

Figure 8: The 2-D cross-section of an airfoil [7]

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. The results of [7] were reproduced using MATLAB® and SIMULINK®.

6.2. 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 ($\boldsymbol{x_0}$).

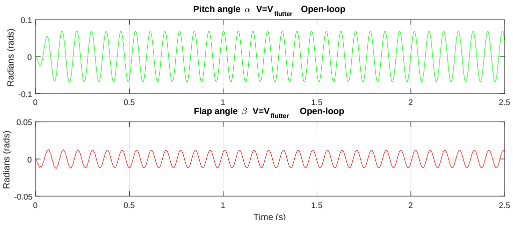

Figure 9: Pitch and flap angle variation over time considering flutter velocity ($V = V_{flutter} = 297.4$ m/s) on open-loop

As seen in Figure 9, if the airspeed is exactly the flutter velocity ($V = V_{Flutter} = 297.4$ m/s), 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. The objective is now to implement a control law to suppress flutter.

6.3. Linear Quadratic Regulator

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

$$\boldsymbol{u} = -\boldsymbol{K}\boldsymbol{x}\,, \tag{24}$$

and secondly, this resulting feedback control law will ensure that the system in closed-loop is stable and robust, but only if the system is controllable and stabilizable [5]. This method is based in the optimization and minimization of the system's performance index J

$$J = \frac{1}{2}\int_0^\infty (\boldsymbol{x^T Q x} + \epsilon \boldsymbol{u^T R u})dt\,. \tag{25}$$

Equation (25) 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. The ϵ is a parameter that determines the relative weights.

6.4. 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 (24)

$$\boldsymbol{u(t)} = -\boldsymbol{K_{LQR}x(t)}\,, \tag{26}$$

and the closed loop system

$$\begin{aligned} \boldsymbol{\dot{x}(t)} &= \boldsymbol{Ax(t)} + \boldsymbol{Bu(t)} \quad \rightarrow \\ \boldsymbol{\dot{x}(t)} &= (\boldsymbol{A} - \boldsymbol{BK_{LQR}})\boldsymbol{x(t)} \quad \rightarrow \\ \boldsymbol{\dot{x}(t)} &= \boldsymbol{A^* x(t)}\,, \end{aligned} \tag{27}$$

where $\boldsymbol{A^*}$ is the the augmented plant matrix.

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.

After choosing the $\boldsymbol{Q}$ and $\boldsymbol{R}$ and considering the flutter velocity ($V_{flutter}$) as 297.4 m/s, the closed loop dynamic response is given in Figure 10.

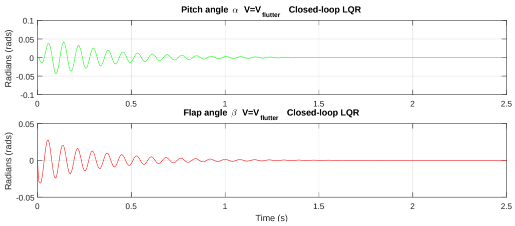

Figure 10: Pitch and flap angle variation over time considering flutter velocity ($V = V_{flutter} = 297.4$ m/s) on closed-loop

Closing the loop stabilized the system. This system that once was marginally stable on the open-loop, now is completely stable having it's 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.

6.5. 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

$$Q = diag(Q_i) \Rightarrow Q_i = \frac{1}{x^2_{i_{max}}}\,, \quad R = diag(R_i) \Rightarrow R_i = \frac{1}{u^2_{i_{max}}}\,. \tag{28}$$

In the flight control system, $u^2_{i_{max}}$ and $x^2_{i_{max}}$ 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. [2]

6.6. Schematic of the Flight Controller Model

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. Controllable states feedback will then be

$$u = -\begin{bmatrix} K_u & K_\gamma & K_\lambda \end{bmatrix} \begin{bmatrix} u - u_{ref} \\ \gamma - \gamma_{ref} \\ \lambda - \lambda_{ref} \end{bmatrix} - K_{LQR_{other}} \begin{bmatrix} w \\ q \\ \beta \\ p \\ r \\ \phi \end{bmatrix} \tag{29}$$

6.7. SIMULINK® Flight Controller Model

The SIMULINK® flight controller model is in Figure 11. It gathers the concepts developed in previous sections, to create a ready-to-use SIMULINK® flight controller model.

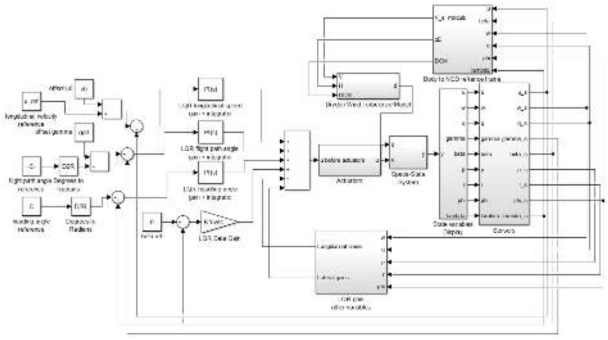

Figure 11: Flight controller model created in SIMULINK®

The dynamics block, named as Space-State System, is defined in Figure 7 and it includes a turbulence model. 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 equation (29). It is assumed a flight with no side slip, hence the reference for β being zero.

6.8. 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. The objective is to use the linear quadratic regulator as a stability augmentation system. However, as this flight controller is designed for a general aircraft, Bryson's method was implemented for an appropriate pole placement.

7. Flight Simulation

Initially, open-loop dynamics of the flight conditions for the Airbus A400M are analysed. Then, the goal is to use the flight controller to follow a reference in the case of engine failure.

7.1. Open-Loop Dynamics

The aircraft used for the engine failure test is the four engine Airbus A400M. From observing Table 7.1, 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. Overall, the longitudinal and lateral motions of the aircraft do not have the level one flying qualities requirement. The LQR script must place the poles of these dynamic modes, in such a way that the level one flying qualities of the aircraft are satisfied.

		Airbus A400M
Longitudinal motion	Phugoid	κ_{phu_1} = -0.066 +0.0883i κ_{phu_2} = -0.066 - 0.0883i
	Short period	κ_{sp_1}= -7.47 + 3.23i κ_{sp_1} = -7.47 - 3.23i
Lateral motion	Spiral	κ_{spi} = 0.0840
	Roll	κ_{roll} = -1.26
	Dutch Roll	κ_{dr_1} = -0.141 + 1.82i κ_{dr_2} = -0.141 - 1.82i
	Heading mode	$\kappa_\lambda = 0$

Table 1: Open-loop dynamic modes eigenvalues

7.2. Engine Failure

In this section, it is assumed that 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

$$\delta_{T_1} = 0\,, \qquad \delta_{T_2} = 0\,. \tag{30}$$

The control surfaces available for the aircraft are then δ_{T3}, δ_{T4}, δ_E, δ_A and δ_R. The goal is to perform a climb ($\gamma_{ref} = 1°$) while maintaining heading ($\lambda_{ref} = 0°$). The reference for longitudinal velocity is maintained at the trim velocity ($u_{ref} = u_0$).

The script is then called and proceeds to find level one flying qualities for a control penalty parameter of $\epsilon = 10$. Table 7.2 contains the closed loop eigenvalues for $\epsilon = 10$ and also for two additional ϵ values.

	Control penalty parameter (ϵ)	10	40	80
Longitudinal motion	Phugoid ($\kappa_{phu_{1,2}}$)	-0.597 ± 0.252i	-0.414 ± 0.36i	-0.369 ± 0.36i
	Short period ($\kappa_{sp_{1,2}}$)	-11.3 ± 4.37i	-6.86 ± 5.23i	-5.39 ± 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 ± 2.65i	-1.33 ± 2.41i	-1.08 ± 2.26i

Table 2: Closed loop poles for $\epsilon = 10$, $\epsilon = 40$ and $\epsilon = 80$

The model's dynamic responses of a thirty five seconds simulation on SIMULINK® are presented in Figures 12, 13 and 14. The position of the aircraft in the ENU frame is defined in Figure 15. Three different control penalty parameters (ϵ) were tested, in order to see its influence on the flight dynamics. Therefore, for each plot there are three sets of curves and the markers for each curve are only to help identification. In Figures 12 and 13, the effects of the control penalty parameter on the control surfaces dynamics response is 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 ($\gamma_{ref} = 1°$), 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, outer engine throttle (δ_{T_4}) tends to decrease because of its higher influence on the yaw moment, and the inner engine (δ_{T_3}) increases its thrust in such a way that both three reference states are achievable and an equilibrium state is found.

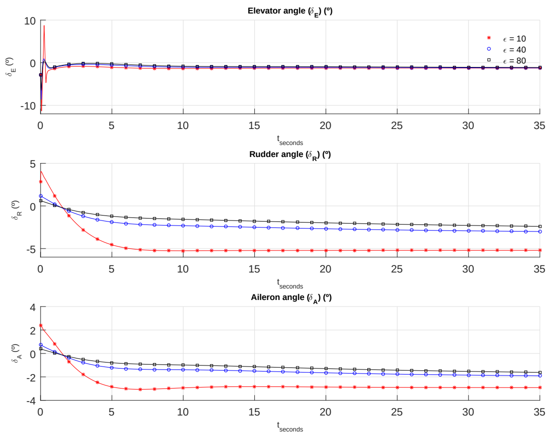

Figure 12: Dynamic responses of the control input variables (δ_E,δ_A,δ_R) for three different control penalty parameters

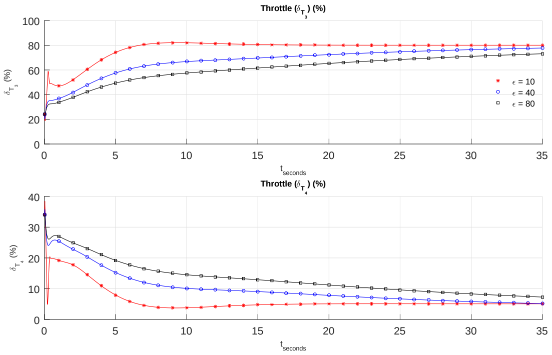

Figure 13: Dynamic responses of the non-malfunctioning engines (δ_{T3},δ_{T4}) for three different control penalty parameters

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

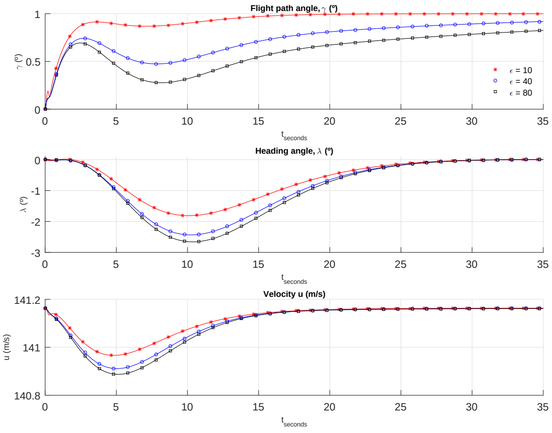

Figure 14: Dynamic responses of the controllable states (u,γ,λ)

In Figure 14, the desirable states dynamic responses are represented. These dynamic responses are the consequence of the control surfaces deflections seen in Figures 12 and 13. When the system allows for higher control deflections ($\epsilon = 10$), the reference state is reached faster than when the control action is more limited ($\epsilon = 80$). However, for reasonable values of ϵ and assuming the system is stable, this final reference state is always reached as seen in Figure 14.

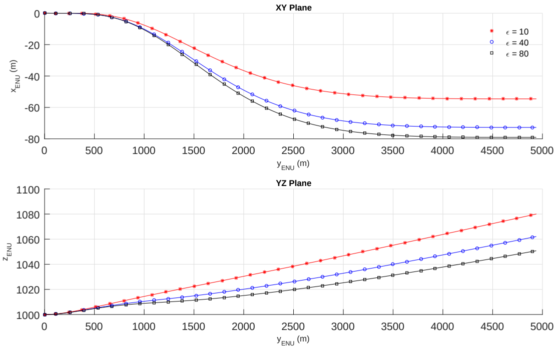

Figure 15: Flight trajectory seen in XY_{ENU} and YZ_{ENU} planes ($\gamma_{ref} = 1°$, $\lambda_{ref} = 0°$ and $u_{ref} = u_0$)

The representation of the states transformed into positions on the XY_{ENU} and YZ_{ENU} planes are shown in 15. 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 Y_{ENU}) but eventually through the control surfaces deflections the heading is stabilized.

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 bending 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 and mathematical model results were as expected, obtaining realistic results on its simulations. 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.

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

References

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

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

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

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

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

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

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

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

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