DEVELOPMENT OF A CONTROL SYSTEM BASED ON DYNAMIC INVERSION FOR A SUBSCALE AIRCRAFT

Author

Ángel González de la Vara

A thesis presented for the degree of
Master of Science in Aeronautical Engineering

Linköpings Universitet

Linköping - May 2019

Linköpings universitet
IEI - Department of Management and Engineering
FLUMES - Division of Fluid and Mechatronic Systems
Master's Thesis 2019

Development of a Control System Based on Dynamic Inversion for a Subscale Aircraft

Ángel González de la Vara

Academic supervisor: Alejandro Sobrón Rueda
Examiner: David Lundström

Linköping universitet
SE-581 83 Linköping, Sverige
013-28 10 00, www.liu.se

Abstract

In this thesis work, a simulator for the longitudinal dynamics of an aircraft has been programmed in *MATLAB* and *Simulink* with the objective of analyzing the response of the *Generic Future Fighter* subscale model when different control systems are applied and a series of adverse conditions affect the nominal flight. The controllers studied are based on model reference adaptive control MRAC and dynamic inversion. Two different adaptive algorithms have been explored: simple adaptation and neural networks. The robustness of these controllers has been tested in the event of control surface destruction, actuator jam, the presence of noise and a constant bias in the measurement of the states, the presence of errors in the dynamics model and for different levels of relaxed longitudinal stability of the aircraft. The results obtained from the simulations show that the adaptive control systems are reliable for all the cases studied while the non-adaptive ones do not perform satisfactorily in the event of actuator failures or the presence of model error. Also, the sensor failures have been demonstrated to be the most detrimental phenomena for the controllers performance. Finally, the linear dynamic inversion controller based on simple adaptation is proposed for future implementation in the subscale model due to its simplicity, good performance and low computational cost.

Acknowledgements

First of all, I would like to thank my supervisor Alejandro Sobrón Rueda for the help, orientation and advise that he gave me to conduct this work as well as the freedom that he offered me to explore research ideas that were not in the initial outline of the work.

In the second place, I want to thank all my friends and colleagues that have shared with me these exciting last six years, especially Anastasiya Rybak, Juan Antonio López and Javier Robles. Thank you for making the hard-work days less hard and the celebration days even more memorable. Thank you for being my life partners wherever we live from this moment on. I also want to thank my little Erasmus family that has made of this last semester an unforgettable and unique experience, always keeping me away from all the parties so that I could focus on this work.

Last but not least, I thank my whole family for their unconditional support, but especially my father, Ángel González, my mother, Consuelo de la Vara and my brother, Carlos González. Thank you for literally giving me the wings that I needed to begin this adventure and always making me feel that I could achieve everything I set my mind to. Thank you for the freedom and confidence that you have always given me. Obviously, without them I would not be the person I am today and all my past, present and future achievements have been, are and will be possible thanks to them.

Contents

List of Figures

List of Tables

List of Symbols

Symbol	IS Units	Description
α	[rad]	Angle of attack
γ	[-]	Learning rate matrix
$\gamma_{s_{ini}}$	[rad]	Initial slope of the trajectory
Γ_V	[-]	Learning rate matrix
Γ_W	[-]	Learning rate matrix
Δ	[-]	Vector of disturbances
Δ_c	[-]	Maximum uncertainty in the model
Δ_q	[rad/s^2]	Disturbance in the time derivative of the pith rate
δ_C	[rad]	Canard deflection
δ_E	[rad]	Elevator deflection
ε	[rad]	Downwash
η_C	[-]	Dynamic pressure efficiency
θ	[rad]	Pitch angle
ϑ	[-]	Parameter vector
$\hat{\vartheta}$	[-]	Estimation of the parameter vector
λ	[-]	Error modification scalar
ξ_{sp}	[-]	Short period damping ratio
σ	[-]	Squashing function
σ_b	[-]	Step bias
σ_d	[-]	Drifting bias
σ_n	[-]	Measurement noise
σ_s	[rad/s]	Standard deviation
τ_{δ_C}	[-]	Scaling factor of the canard
τ_{δ_E}	[-]	Scaling factor of the elevator
φ	[-]	Regressor
$\omega_{n_{sp}}$	[rad/s]	Short period natural frequency
A	[-]	System matrix
$\hat{A}$	[-]	Estimated system matrix
a	[-]	Activation potential
AR	[-]	Aspect Ratio
B	[-]	Control matrix
$\hat{B}$	[-]	Estimated control matrix
C	[-]	Output matrix
CD	[-]	Drag coefficient
CD_{trim}	[-]	Drag coefficient for trim conditions
CL	[-]	Lift coefficient
CL_{trim}	[-]	Lift coefficient for trim conditions
CM	[-]	Pitch moment coefficient
CM_{trim}	[-]	Pitch moment coefficient for trim conditions
c_w	[m]	Reference wing chord

Symbol	IS Units	Description
D	[N]	Drag
e	[-]	Tracking error
e	[-]	Oswald coefficient
g	[m/s^2]	Gravitational acceleration
h_{ini}	[m]	Initial altitude
I_{yy}	[Kg m^2]	Moment of inertia in the y direction
K	[-]	Gain matrix
K_D	[-]	Induced drag coefficient
K_d	[-]	Slope of the drift function
L	[N]	Lift
M	[N m]	Pitch moment
m	[Kg]	Mass
mse	[-]	Mean squared error
N	[-]	Number of hidden layer neurons
n	[-]	Load factor
q	[rad/s]	Pitch rate
q_d	[N/m^2]	Dynamic pressure
q_{des}	[rad/s]	Desired pitch rate
r	[-]	Reference
s_E	[-]	Health factor of the elevator
S_C	[m^2]	Reference surface of the canard
s_C	[-]	Health factor of the canard
S_w	[m^2]	Reference wing surface
SM	[-]	Static margin
T	[N]	Thrust
t	[s]	Time
t_f	[s]	Time of the failure
u	[m/s]	Longitudinal velocity in the body-fixed frame
u	[-]	Input vector
V	[m/s]	Airspeed
V	[-]	Input weights matrix
v	[-]	Pseudocontrol input
v_{ad}	[-]	Adaptive part of the pseudocontrol input
V_a	[-]	Lyapunov candidate function
V_{ini}	[m/s]	Initial velocity
W	[-]	Output weights matrix
w	[m/s]	Transversal velocity in the body-fixed frame
x	[-]	States vector
$\bar{x}$	[-]	Input to the neural network
X_a	[N]	Longitudinal aerodynamic force in the body-fixed frame
x_{ac_C}	[m]	Position of the canard's aerodynamic center
x_{ac_W}	[m]	Position of the wing's aerodynamic center

Symbol	IS Units	Description
x_{cog}	[m]	Position of the center of gravity
x_{NP}	[m]	Position of the neutral point
x_{sf}	[-]	Sensor output
y	[-]	Output
Z_a	[N]	Transversal aerodynamic force in the body-fixed frame

1
Introduction

1.1 Background

Since the invention of the first functional aircraft by the Wright brothers, control techniques and systems have played a critical role in the development of modern aviation as we know it today. This very first airplane, the Flyer I (Figure 1.1), had a manual three-axis control mechanism based on wing-warping, a canard and a rudder that allowed the skilled pilot to steer the aircraft and maintain its equilibrium in spite of the fact that the airplane was unstable because of its configuration.

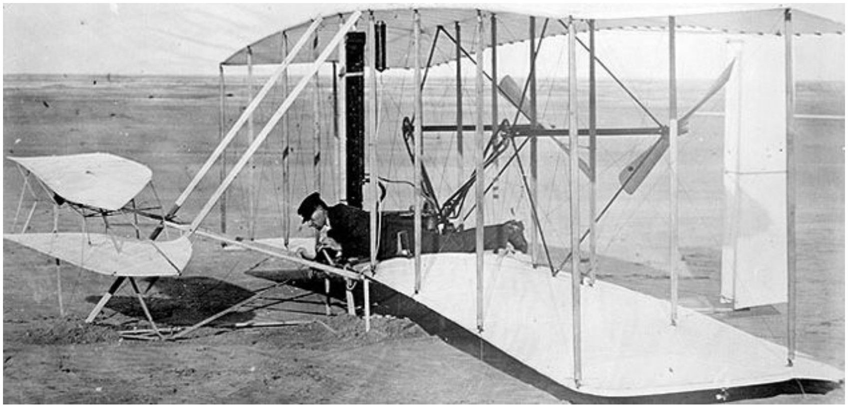


Figure 1.1: The Flyer I of the Wright brothers, the first controlled, powered and heavier-than-air aircraft.

In the following decades, the advent of electronics and computers brought about the development of automatic control systems capable of maintaining the aircraft stable in its working regime under the presence of perturbations such as turbulence or the redistribution of weights, or even control the dynamics of a dynamically unstable aircraft, taking pressure off the pilots who could then focus on the navigation procedures.

Nevertheless, despite all these efforts to increase the aviation safety, any unexpected failure of a system or subsystem such as an actuator or a sensor could result in the loss of control of the aircraft and in the worst case, a catastrophic accident. In order to prevent this, new fault tolerance control systems are being developed. Robust control systems and adaptive control systems that allow the aircraft operate even in the presence of a failure can reduce the risk of a fatal accident significantly. Moreover, the implementation of machine learning and artificial intelligence in the control systems may allow the different subsystems to work even without an accurate model of the aircraft dynamics.

The development of this kind of modern control systems is complex due to different reasons. One of them is the limited availability of accurate information about the aircraft's states since sometimes the number of available sensors is small in order to reduce implementation costs. Another limitation for the design is the availability of a good model of the aircraft dynamics. The actual behavior of the aircraft is defined by highly non-linear phenomena that arise due to the presence of non-stationary aerodynamic effects, the fluid-structure coupling producing aeroelastic effects and the dynamics of the different subsystems that make up the aircraft as a whole. Normally, these effects are not considered in the models because it would increase the complexity beyond practical purposes. Instead, the dynamics are represented by simplified models, being transfer functions linearized around an equilibrium point in most of the cases.

The massive development of computational tools over the last decades has brought unprecedented analysis capabilities to engineers. These tools are widely used in the early stages of any aircraft or spacecraft, where the level of accuracy and detail is not specially restrictive. However, the same tools may stop being useful in more advanced stages of the design due to the necessity of high-fidelity models that cannot be provided by these tools given the difficulties found when considering the nonlinear effects mentioned above. In order to refine the initial models, the corresponding data can be gathered from experimental techniques such as flight tests or wind-tunnel tests performed with a full-scale model of the vehicle. However, these techniques normally imply high fabrication and operational costs and therefore are falling into disuse.

As a result of this trend, together with the technological progress and miniaturization of the electronic components, an alternative technique is playing an important role in the aircraft development process: testing sub-scale models. According to NASA's researcher J.Chambers [1], sub-scale models can be defined in this context as physical, downsized reproductions of components or vehicles used to examine characteristics of larger full-scale counterparts. These sub-scale models are suitable for flight or wind-tunnel testing thanks to the associated cost reduction. It has also become available for small companies and academic institutions, strengthening low-cost research projects and enhancing the education in the engineering field, as reported by Jouannet, Berry and Krus [2].

Testing sub-scale prototypes constitutes a valuable tool in aircraft design since it can complement the sparse knowledge available in the early design stages with critical data that could be very difficult to obtain by other methods. Furthermore, it has proved itself extremely valuable in research of critical stability and control characteristics for complex flight conditions that are not easily studied with conventional techniques, such as dangerous maneuvers outside the flight envelope and flight at high angles of attack.

An example of the extreme complexity that the design of a control system for a high maneuverable aircraft implies is the Rockwell-MBB X-31, shown in Figure 1.2. This model is an experimental jet fighter designed to test fighter thrust vectoring technology within the *Enhanced Fighter Maneuverability* program. The main objective of the project was to provide additional control authority in pitch and yaw in order to achieve significantly more maneuverability than most conventional fighters. An advanced control system provided controlled flight at high angles of attack where conventional aircraft would stall or lose control. During flight testing, the X-31 aircraft established several milestones, successfully achieving controlled flight at a 70° angle of attack and executing a rapid minimum-radius 180° turn using a post-stall maneuver, flying well outside the range of angle of attack normal for conventional aircraft. Two X-31s were built, carrying out over 500 flight tests between 1990 and 1995. However, on January 19, 1995, one of aircraft crashed during a high altitude test due to the formation of ice inside the pitot tube, sending incorrect airspeed data to the flight control computers [3]. This kind of accidents demonstrate the importance of redundancy in the data acquisition sensors and subsystems and the impact that fault tolerant control systems would have in enhancing the flight safety.

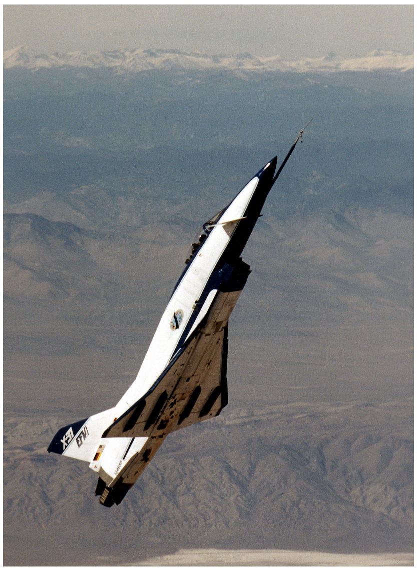


Figure 1.2: X-31 performing a high angle of attack maneuver. Source: NASA.

Sub-scale models present themselves as an adequate platform to test these advanced control systems given that the absence of a pilot and the non-excessive fabrication costs reduce the critical human and economic impact that their full-scale counterparts would have in case of crash. However, although at first sight the possibilities of sub-scale testing may seem extremely attractive, it is important to notice that both testing and results are constrained by certain factors. Among all the scaling methods available, the prevalent one is the dynamic scaling. In this case, aeroelastic effects are usually neglected due to their complexity although they could play an important role in the sub-scale model dynamics if the stiffness of the components and actuators is not high enough. Key scaling factors include geometric similarity, aerodynamic similarity (Reynolds number and compressibility effects), inertia scaling and Froude number. Since normally it is not possible to satisfy all the similitude requirements simultaneously, it is critical to be aware of the limitations of the sub-scale test and the results must be interpreted carefully. The principal problem in dynamic sub-scale testing is the aerodynamic similarity between the model and the full-scale aircraft. Even though compressibility effects can be included in the sub-scale analysis, there is always a significant discrepancy in the Reynolds number that can play an important role when analyzing viscosity-dependent phenomena such as flow separation at high angles of attack and maximum lift conditions.

Building and flight testing sub-scale demonstrators is an important part of the aircraft design education at Linköping University. The research team has access to four advanced sub-scale aircraft although none of them has been equipped with an automatic flight control system. The two relevant models for this work are the Dassault Rafale fighter and the Generic Future Fighter (GFF), shown in Figure 1.3.

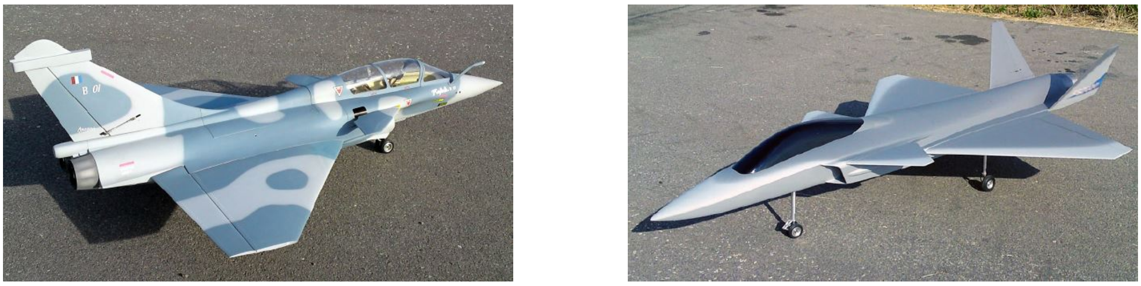


Figure 1.3: Two of the sub-scale research aircraft and concept demonstrators available at Linköping University. Left: Dassault Rafale fighter test-bed. Right: Generic Future Fighter. Courtesy of Linköpings University.

The interest in these models lies in the possibility of modifying their on-board flight control systems in order to analyze the feasibility of new control architecture implementation and study the dynamic response of the aircraft when exposed to different flight conditions. This is one of the main objectives of this work and the details will be explained with more insight in the following sections.

1.2 Objectives

The main objective of the present work is the analysis of the performance of different modern control systems when the aircraft flies under a series of specific conditions, such as relaxed longitudinal stability, actuator and/or sensor failures and external disturbances. The ultimate goal is to implement one of the controllers studied into the actual subscale model available in the department and perform flight tests in order to compare the actual behavior of the aircraft with the results obtained from the simulations.

In order to analyze the aircraft's behavior when different control systems are used, a flight simulator that allows to arbitrarily switch between several controllers and that includes the dynamics and the possible failures of actuators and sensors is developed using *Matlab* and *Simulink* [4]. This simulator is initially focused on the longitudinal dynamics of the aircraft since the variables of interest for this work, the angle of attack α and the pitch rate q, belong to the short-period mode. It will be assumed in most of the cases that the longitudinal dynamics are decoupled from the lateral-directional dynamics and that the latter are stable and not excited. However, this will not be completely true in the cases where the failure of a control surface induces an asymmetric lift distribution, producing undesired roll and/or yaw moments and lateral forces.

Finally, if time allows it, the control laws that prove themselves more reliable and show better performance will be implemented in the actual controller of the sub-scale models shown before, as long as the hardware available and the software modifications needed allow it.

Therefore, the goals pursued by this work are the following:

- Create a flight simulation tool based on *Matlab* and *Simulink*. This tool must allow to choose between different control systems, consider nonlinear behavior of the aircraft dynamics and contemplate the possibility of actuator and sensor failures.
- Design a series of controllers for the longitudinal dynamics that are effective when high angle of attack maneuvers are performed and that are able to recover stability after stall. Certain handling requisites should be satisfied.
- Study the reliability of these controllers under the event of a system malfunction, exploring the limitations of the control system response depending on the gravity of the failure.
- Study the maximum level of relaxed stability that these control systems can handle successfully.
- Compare the results obtained from the nonlinear simulator with those obtained from a traditional analysis using linear transfer functions.
- Demonstrate that the control systems developed for the simulator can be implemented in the low-cost Commercial off-the-shelf (COTS) controller with the available sensors.

1.3 Methodology

In order to achieve the objectives explained in the previous section and avoid time delays or lack of insight in the analysis, the scope of each of the individual studies must be clearly established. To do so, a brief description of the different control systems and actuator and sensor failures that are studied in this work is presented.

1.3.1 Control systems

The overall objective of this work is the study of the longitudinal dynamics of the sub-scale models shown in Figure 1.3 and the design of reliable control systems. These aircraft present an advanced delta-canard configuration, quite convenient for the implementation of fault tolerant control techniques given the redundancy in the actuators that this configuration implies. Pitch control is achieved by combining symmetric elevon deflection and canard deflection. However, the differential deflection of the control surfaces is possible and, in fact, plays an important role in the fault tolerant control systems since the failure of one of the actuators could produce an asymmetric distribution of lift and, in consequence, an asymmetric deflection of the control surfaces could be required to maintain stability.

This configuration, characterized by the lack of an horizontal stabilizer in the rear of the fuselage and the presence of the canard, makes the aircraft's longitudinal dynamics be statically and dynamically unstable since the neutral point is ahead of the center of gravity. The neutral point is defined as the point of the aircraft where the resultant of all the aerodynamic forces is applied. Therefore, when the position of the center of gravity and the neutral point is the same the stability is said to be neutral, while stability is naturally achieved when the center of gravity is ahead of the neutral point.

Since the distribution of weights in the aircraft cannot be arbitrarily modified during a flight under normal circumstances, the most common way to achieve stability for an aircraft with relaxed longitudinal stability is the implementation of stability augmentation systems. Normally this systems are based on closed-loop controllers in which the control action is dependent on feedback from the process in the form of measurements of the states of the aircraft. By comparing the output of the aircraft with the reference commanded by the pilot or the navigation system, the controller is able to command the corresponding actuator actions so that the deviation between the output and the reference converges to zero, achieving stability.

However, the task of designing a closed-loop control system for an aircraft that presents an acceptable performance under all the possible conditions of the flight envelope has always been tedious. Traditionally, control systems have been designed from a simplified model of the actual system by linearizing the dynamics around a trim point that is dependent on the flight conditions. This simplified models, represented by transfer functions, establish a relationship between one of the states of the aircraft and the action of one of the actuators available. For this reason, this systems are called single-input-single-output (SISO). This approach introduces the need of having a bank of models and controllers stored in the memory of the on-board computer and switch between them depending on the specific flight conditions. Furthermore, it can be difficult to design the controller with the structure needed to tolerate the presence of model uncertainties, disturbances and/or failures and the coupling between the dynamics represented by independent SISO transfer functions.

The aircraft studied in this work count with more than one actuator to control the longitudinal dynamics. The differential deflection of both the canard and the elevons makes a total of four control surfaces available. Therefore, in order to avoid the problems that arise from the use of the traditional SISO control systems, this work will explore the design of modern control systems. One of the main advantages of using modern control theory instead of the traditional approaches is that, in the case of complex systems with multiple-inputs and multiple-outputs (MIMO), these schemes allow to assign the feedback gains for all the inputs simultaneously while in classical control theory the individual gains must be separately selected using trial-and-error strategies.

Following this basis, the objective is to design modern control systems that allow to control the short period mode and avoid stall or minimize its effects. Since the full-scale models of the aircraft available in the laboratory have a wide flight envelope that covers a wide range of conditions (high angle of attack maneuvers and a wide range of dynamic pressures), the coefficients that define the dynamics of the aircraft will also change with these conditions. Therefore, a dynamic model that is stable and that fulfills the handling requirements for one flight condition may even become unstable in a different point of the flight envelope. This scenario calls for the use of adaptive control systems that are able to identify the changes in the aircraft dynamics and modify the corresponding parameters of the controller. This changes can be induced by the operation in different points of the flight envelope as well as the failure of one or several of the actuators or sensors, making this type of controllers also suitable for fault tolerant approaches. However, although the implementation of adaptive algorithms may be suitable for subscale models such as the NASA's AirSTAR test aircraft [6], their use in full-scale aircraft is still controversial nowadays given the risk that the instability or lack of robustness of these systems implies for flight safety. For this reason, only experimental and military aircraft such as the X-15A-3, that crashed, and the X-45A J-UCAS [7], include this kind of advanced control systems.

A brief description of the control systems studied in this work is presented. The mathematical background of every one of them will be presented in the following sections.

- Model Reference Adaptive Control (MRAC): this control scheme is extremely useful since it is able to estimate the coefficients that define the dynamics of the system and update the parameters of the controller depending on the flight conditions, what allows to maintain controllability even in the event of certain failures and when highly nonlinear effects play a decisive role in the dynamical behavior of the aircraft, like under the performance of high angle of attack maneuvers. Furthermore, this controller allows to track the dynamics of a reference model that presents the desired dynamic properties. Therefore, it is even possible to make one aircraft behave like an entire different aircraft. As an example, the space shuttle pilots trained on a Gulfstream-II aircraft with this system implemented that simulated the feel of the space shuttle [5]. However, in spite of all the advantages mentioned, stability cannot always be guaranteed since the estimation of the dynamic coefficients could diverge under certain circumstances, such as the presence of unmodeled dynamics acting on the system. For this reason, the implementation of this kind of controllers in real systems is a controversial topic nowadays.

- Dynamic inversion: this type of controller, which has gained in popularity in recent years for aircraft control design [10], is based on the technique of feedback linearization. This technique takes into account the nonlinearities of the aircraft and thus does not require to schedule the controller used depending on the flight conditions. Therefore, it is suitable for a wide range of operating conditions, including high angle of attack and hypervelocity design. However, this scheme requires the model of the aircraft to be known and full state feedback is also necessary, which can be difficult to achieve in sub-scale applications due to the limitation in the quality of the sensors available.

- A combination of the MRAC and the dynamic inversion systems could avoid the characteristic problems that every one of them bring individually.

1.3.2 Actuator and sensor failures

The inclusion in the flight simulator of dynamic models to simulate the behavior of the actuators and the sensors of the aircraft is one of the main purposes of this work since it allows to study the effects of a malfunctioning system in the dynamics of the aircraft and, in consequence, improve the design of the controller in order to minimize the negative effects of this failures and avoid a possible catastrophe. The main failures studied in this work are presented below.

The nature of actuator failures can be very diverse, from the malfunction of an hydraulic actuator due to a poor design to the oscillation of a control surface's position around the trim point due to aeroelastic effects. The modeling of most of these failures is highly complex, like the oscillation around the trim point that would require a statistical study of the frequency and amplitude of the actuator's motion, and is therefore far beyond the scope of this work. However, some of the most usual failures do not require such specific analysis, being easier to model and equally relevant for the study of their effects on the dynamics of the aircraft. The actuator failures studied in this work are:

- Jammed control surface: when this failure occurs, the corresponding control surface is locked at a certain deflection and can no longer be moved. The deflection at post-failure condition can be the one existing at the moment of the failure or the surface can move at a certain position within the range of motion and remain there. This failure does not alter the aerodynamic characteristics of the control surface and can be modeled by simply disconnecting the corresponding input to the aircraft dynamic system and keeping it constant.

- Loss of effectiveness of a control surface: this failure can be caused by the partial or total destruction and/or deformation of the control surface. Another cause could be the loss of command power at high angles of attack due to aerodynamic effects. When this failure occurs, the aerodynamic efficiency of the control surface is deteriorated. Therefore, the asymmetric distributions of lift and drag could produce rolling and/or yawing moments, inducing the coupling of the longitudinal and the lateral-directional dynamics of the aircraft. This failure can be modeled by scaling the aerodynamic derivatives of the corresponding control surface with an efficiency parameter.

Regarding the sensor failures, they represent one of the most critical scenarios because of its difficulty to be detected and identified, being the cause of an elevated number of aircraft accidents. For example, the presence of ice in the pitot tube could affect the airspeed measurements, resulting in an incorrect actuator deflection commanded by the control system that bases its control laws in the feedback obtained from the sensors. This was the cause of the X-31 catastrophic accident mentioned before. The main phenomena and failures that affect the sensors measurements considered in this work are the following:

- Noise: defined as random variation of the output due to environmental and internal sources. It can be modeled using a normal probability distribution defined by its mean value and its standard deviation.
- Delay: defined as the time that the sensor takes to react to changes in the input. There are several types of delays; constant pure time delay, dynamic delay (such as a time constant characterizing a first order system type of response), initial dead time and the combinations of them all. It is caused by the natural dynamics that describe the functioning of the sensor.
- Dead band: it is the range of input for which the sensor does not produce any output.
- Bias: defined as a constant offset of the sensor output. However, it can also be variable in time, changing randomly, or triggered by more or less abrupt variations of external parameters, including the sensor input. The bias studied in this work is constant in time.
- Drift: it is an addition to the sensor output linear with time. The slope of the drift function can vary in a way and for reasons similar to the bias.
- Position and rate saturation: the output and/or its rate of variation is limited.

The mathematical background of both the actuator and sensor failures modeling will be presented in the following sections.

1.4 Limitations

The objectives of this work are quite varied and involve different technical fields that would require dedicated investigation. Therefore, it is expected to find the following problems and limitations to obtain the expected results:

- Due to the differences in their operating regime, the sub-scale models will not experience certain phenomena that their full-scale counterparts will, such as the appearance of shock waves or the high altitude flight. For this reason, the dynamic and aerodynamic similarity is not always achieved and the dependency of the aerodynamic coefficients with the Reynolds number and the Mach number will be different. This issue makes the results obtained not applicable to the full-scale models and they should be analyzed carefully.

- The accuracy of the mathematical models available for the dynamics of the aircraft is limited given the difficulties found when estimating them. In fact, nonlinear phenomena such as aeroelastic effects can play an important role in the behavior of the controller given the relative low stiffness of the actuators that could make it hard to reach the desired deflection of the control surfaces.
- Initially, only the study of the longitudinal dynamics is performed. This could not be enough in the case of an actuator failure that causes the coupling between the longitudinal and the lateral-directional dynamics.
- The accurate modeling of the actuators and sensors dynamics can present some difficulties. The estimation of the delays can be the hardest issue since it is highly influenced by the electronic systems used.
- The implementation of the control systems studied in the actual sub-scale models may present some problems due to the number of sensors available and their performance. The modification of the COTS code could also be limited and might not be suitable for the controllers proposed.

1.5 Thesis outline

Apart from this introductory chapter, the document consists of six main parts more. The second one comprises the development of the mathematical models for the aircraft dynamics and the actuator and sensor failures, as well as an overview of the flight simulator distribution and an explanation of how each of the subsystems is implemented and connected to the rest. The third part is focused on the mathematical background of the different control systems proposed and contains a more detailed insight into the problems and advantages that arise from their implementation. The fourth chapter presents all the results obtained from the simulations, their analysis and a brief comparison between the controllers performance. The fifth part includes the discussion about the methodology followed in this work and the results obtained as well as a final implementation proposal. Finally, the sixth and final chapters show the main conclusions obtained from this work and the suggestion of possible future work respectively.

2
Model of the Flight Simulator

In this section, the different models (dynamical, aerodynamic) used to build the flight simulator are shown and explained. The flight simulator used in this thesis is based on the *MathWorks* library *Airlib*, implemented in *Simulink*, although it has been modified to fulfill the objectives of this work. This simulator includes a nonlinear model of the aircraft's dynamics with a linear model of the aerodynamic forces. It also includes the possibility of implementing a flight controller and the presence of failures in the actuators or/and the sensors, which is the main goal of the thesis.

2.1 Nonlinear model of the dynamics

In order to understand the aircraft's behavior it is necessary to study the aircraft's equations of motion, shown in Eq. (2.1). Only the longitudinal equations are shown since only the longitudinal dynamics are studied in this work.

$$\begin{aligned}
\dot{u} &= -q\ w - g \sin\theta + \frac{T + X_a}{m} \\
\dot{w} &= q\ u + g\ \cos\theta + \frac{Z_a}{m} \\
\dot{q} &= \frac{M}{I_{yy}} \\
\dot{\theta} &= q
\end{aligned} \tag{2.1}$$

where u and w are the velocities of the center of gravity of the aircraft in the x and z directions of the body-fixed frame respectively, q is the pitch rate, θ the pitch angle, g the gravitational acceleration, m is the mass of the aircraft, I_{yy} the moment of inertia in the y direction, T the thrust, X_a and Z_a the aerodynamic forces in the body-fixed frame and M the pitching moment. The different reference systems are shown in Figure 2.1.

The aerodynamic forces and moments shown in the previous equations are in the body-fixed frame. However, the following analysis is performed in the wind frame since the data about the aerodynamic derivatives is available with respect to that reference system. The relation between the forces in both reference systems is the following:

$$\begin{aligned}
X_a &= L\ \sin\alpha - D\ \cos\alpha \\
Z_a &= -L\ \cos\alpha - D\ \sin\alpha
\end{aligned} \tag{2.2}$$

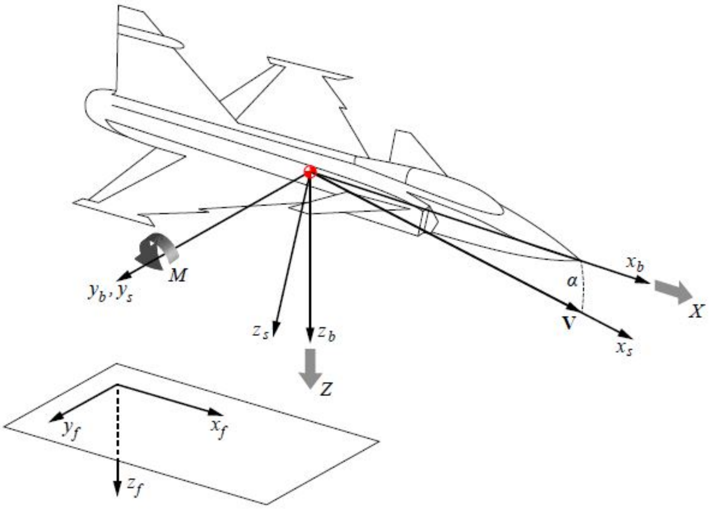


Figure 2.1: Diagram showing the different reference systems and the longitudinal forces and moments. The body-fixed frame is represented with the subscript b, the stability frame with the subscript s and the Earth-fixed frame with the subscript f.

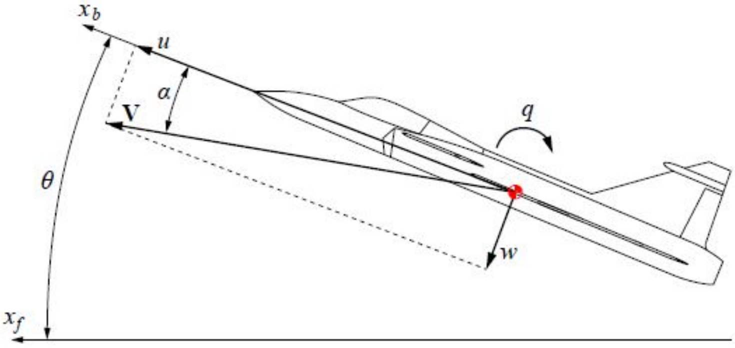


Figure 2.2: Diagram showing the relation between the variables in the body-fixed frame and the wind frame.

where α is the angle of attack and L and D the lift and drag forces respectively. Notice that M does not need to be converted to the wind reference system since the y axis of both frames are parallel. Now, substituting the expressions of X_a and Z_a into Eq. (2.1) and considering the following relations:

$$u = V \ \cos\alpha; \qquad\qquad w = V \ \sin\alpha \tag{2.3}$$

where V is the true airspeed, an expression for the equations of motion in terms of the forces and the variables in the wind reference system is available.

$$\begin{cases} \dot{V} &= \dfrac{T\ \cos\alpha - D}{m} + g\,(\cos\theta\sin\alpha - \sin\theta\cos\alpha) \\[2ex] \dot{\alpha} &= q - \dfrac{T\ \sin\alpha + L}{m\ V} + \dfrac{g}{V}\,(\cos\theta\cos\alpha + \sin\theta\sin\alpha) \\[2ex] \dot{q} &= \dfrac{M}{I_{yy}} \\[2ex] \dot{\theta} &= q \end{cases} \tag{2.4}$$

The reason why these equations are used instead of the ones using the body-fixed frame variables is that the main objective of this work is the design of control systems capable of operating correctly at high angles of attack. These equations will be linearized in Section 2.3 in order to obtain a simple linear model from which the dynamics of the aircraft can be analyzed and the preliminary design of the control systems can be performed.

2.2 Aerodynamic model

As it is widely known, the linear model of the aerodynamic forces and moment is the following:

$$L = q_d\ S_w\ CL; \qquad D = q_d\ S_w\ CD; \qquad M = q_d\ S_w\ c_w\ CM \tag{2.5}$$

where q_d is the dynamic pressure, S_w is the wing reference surface, c_w is the mean aerodynamic chord and CL, CD and CM are the lift, drag, and pitching moment coefficients respectively. These coefficients are a nonlinear function of the flight conditions. However, for a specific point in the flight envelope, the following expressions can be used:

$$\begin{aligned} CL &= CL_0 + CL_\alpha\ \alpha + CL_q\ \frac{c_w}{2V}\ q + CL_{\dot{\alpha}}\ \frac{c_w}{2V}\ \dot{\alpha} + CL_{\delta_E}\ \delta_E + CL_{\delta_C}\ \delta_C \\ CD &= CD_0 + \frac{1}{\pi\ e\ AR}\ CL^2 \\ CM &= CM_0 + CM_\alpha\ \alpha + CM_q\ \frac{c_w}{2V}\ q + CM_{\dot{\alpha}}\ \frac{c_w}{2V}\ \dot{\alpha} + CM_{\delta_E}\ \delta_E + CM_{\delta_C}\ \delta_C \end{aligned} \tag{2.6}$$

where e is the Oswald factor, AR is the aspect ratio of the wing and δ_E and δ_C are the elevator and canard deflections respectively. The deflection of the control surfaces is defined positive when they generate a positive lift, as shown in Figure 2.3.

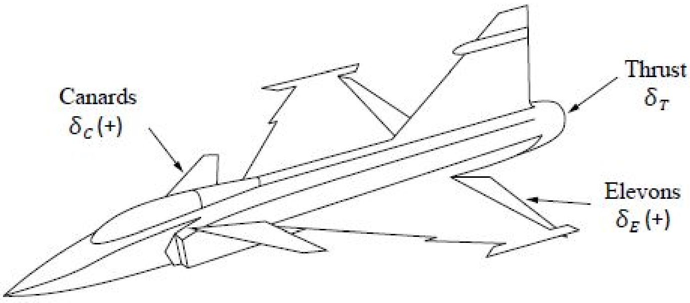


Figure 2.3: Diagram showing the sign convention adopted in this work for the control surface deflections.

Now that the aerodynamic model used in the simulator is known, the effects of the actuator failures in the aircraft's dynamics can be introduced.

2.2.1 Model of the actuator failures

The control of an aircraft is mainly performed with the control surfaces and the throttle. Therefore, a failure of any nature in any of the actuators would result in a loss of maneuverability, control power or even the loss of stability. For this reason, the study of the control systems' robustness in the event of a failure is crucial in the design of fault tolerant control systems. Since performing flight tests with failures in the actuators could jeopardize the integrity of the subscale models available, the robustness analysis is performed by simulating the failures in the flight simulator.

The actuator failures have a direct effect on the aircraft's dynamics. The first of the failures studied in this work is the partial or total destruction of one of the control surfaces caused by a possible crash with an object in flight or due to deformation. When this occurs, the aerodynamic efficiency of the control surface is deteriorated from the instant of the failure on. Therefore, the aerodynamic forces and moments produced by the damaged control surface are different for the same pilot input. In order to simulate this phenomena, a health factor is defined for each of the control surfaces. For the configuration of the aircraft studied in this work, s_E and s_C are defined as the health factors of the elevator and the canard respectively. Moreover, it is necessary to define one efficiency parameter for every control surface available. In the case of the elevator and the canard, the efficiency parameters are selected to be the lift derivatives with respect to the elevator CL_{δ_E} and canard CL_{δ_C} deflections respectively. Therefore, the health factors model the magnitude of the failure through the ratio between the efficiency parameters after and before the failure:

$$s_E = \frac{(CL_{\delta_E})_{\text{After failure}}}{(CL_{\delta_E})_{\text{Before failure}}}; \qquad s_C = \frac{(CL_{\delta_C})_{\text{After failure}}}{(CL_{\delta_C})_{\text{Before failure}}} \tag{2.7}$$

It is easy to notice that for the case when there is not any failure, the health factors will be equal to one, while they will be equal to zero in the case of total destruction or absence of the control surface.

The partial destruction of one of the control surfaces not only affects the lift derivatives with respect to the corresponding control deflection, but also affects the rest of the aerodynamic coefficients. Therefore, in order to study the effects of this type of failure, it is necessary to analyze the independent contribution of the aircraft's body, wing and control surfaces in the total aerodynamic derivatives. The objective of this analysis is to get expressions of the aerodynamic derivatives in terms of the efficiency parameters CL_{δ_E} and CL_{δ_C} so that when a failure occurs, all the aerodynamic derivatives are properly modified. As an example, the derivation of the total CL_α expression as a function of the efficiency parameters is shown.

The total CL_α can be split into the contributions of the body, the wing and the canard as follows:

$$CL_\alpha = CL_{\alpha_B} + \left(1 - \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{down}}\right) CL_{\alpha_W} + \eta_C \frac{S_C}{S_w}\left(1 + \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{up}}\right) CL_{\alpha_C} \tag{2.8}$$

where CL_{α_B}, CL_{α_W} and CL_{α_C} are the contributions of the body, the wing and the canard respectively, $\partial\varepsilon/\partial\alpha|_{\text{down}}$ is the change in the effective angle of attack of the wing due to the downwash caused by the canard, $\partial\varepsilon/\partial\alpha|_{\text{up}}$ is the change in the effective angle of attack of the canard due to the upwash caused by the main wing, η_C is the dynamic pressure efficiency in the canard and S_C is the canard surface.

On the other hand, the lift derivatives with respect to the control surface deflections can be expressed as a function of the lift derivatives with respect to the angle of attack of each of the wing and the canard:

$$\begin{aligned} CL_{\delta_E} &= \tau_{\delta_E}\, CL_{\alpha_W}; & \rightarrow \qquad CL_{\alpha_W} &= \frac{1}{\tau_{\delta_E}} CL_{\delta_E} \\ CL_{\delta_C} &= \tau_{\delta_C}\, \eta_C\, \frac{S_C}{S_w}\, CL_{\alpha_C}; & \rightarrow \qquad CL_{\alpha_C} &= \frac{1}{\tau_{\delta_C}} \frac{CL_{\delta_C}}{\eta_C\, \dfrac{S_C}{S_w}} \end{aligned} \tag{2.9}$$

where τ_{δ_E} and τ_{δ_C} are scaling factors that depend on the ratio between the areas and the chords of the part of the control surfaces that can be deflected and the fixed parts of the surfaces. Now, substituting these expressions into Eq. (2.8), an expression of the total CL_α as a function of the so called efficiency parameters is obtained:

$$CL_\alpha = CL_{\alpha B} + \frac{1}{\tau_{\delta_E}}\left(1 - \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{down}}\right) CL_{\delta_E} + \frac{1}{\tau_{\delta_C}}\left(1 + \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{up}}\right) CL_{\delta_C} \tag{2.10}$$

The rest of the aerodynamic derivatives are obtained following an analogous procedure and are shown below:

$$
\begin{aligned}
CL_0 &= CL_{0_B} + CL_0 \frac{1}{\tau_{\delta_E}} \left(1 - \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{down}}\right) \frac{CL_{\delta_E}}{CL_\alpha} + CL_0 \frac{1}{\tau_{\delta_C}} \left(1 + \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{up}}\right) \frac{CL_{\delta_C}}{CL_\alpha} \\
CL_q &= CL_{q_{WB}} - 2\frac{1}{\tau_{\delta_C}} \frac{x_{cog} - x_{ac_C}}{c_w} CL_{\delta_C} \\
CL_{\dot{\alpha}} &= CL_{\dot{\alpha}_B} + \frac{2}{\tau_{\delta_E}} \frac{x_{ac_W} - x_{ac_C}}{c_w} \left|\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{down}} CL_{\delta_E} \\
CM_0 &= CM_{0_B} - CM_0 \frac{1}{\tau_{\delta_E}} \left(1 - \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{down}}\right) \frac{x_{cog} - x_{ac_W}}{c_w} \frac{CL_{\delta_E}}{CM_\alpha} - ... \\
&\quad - CM_0 \frac{1}{\tau_{\delta_C}} \left(1 + \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{up}}\right) \frac{x_{cog} - x_{ac_C}}{c_w} \frac{CL_{\delta_C}}{CM_\alpha} \\
CM_\alpha &= CM_{\alpha_B} + \frac{1}{\tau_{\delta_E}} \left(1 - \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{down}}\right) \frac{x_{cog} - x_{ac_W}}{c_w} CL_{\delta_E} + ... \\
&\quad + \frac{1}{\tau_{\delta_C}} \left(1 + \left.\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{up}}\right) \frac{x_{cog} - x_{ac_C}}{c_w} CL_{\delta_C} \\
CM_q &= CM_{q_{WB}} - 2\frac{1}{\tau_{\delta_C}} \left(\frac{x_{cog} - x_{ac_C}}{c_w}\right)^2 CL_{\delta_C} \\
CM_{\dot{\alpha}} &= CM_{\dot{\alpha}_B} - \frac{2}{\tau_{\delta_E}} \left(\frac{x_{ac_W} - x_{ac_C}}{c_w}\right) \left|\frac{\partial \varepsilon}{\partial \alpha}\right|_{\text{down}} \left(\frac{x_{ac_W} - x_{cog}}{c_w}\right) CL_{\delta_E}
\end{aligned}
\tag{2.11}
$$

where x_{cog}, x_{ac_W} and x_{ac_C} are the positions of the center of gravity and the aerodynamic centers of the wing and the canard respectively. From these expressions, the effects of a partial destruction of a control surface in the overall aerodynamic coefficients can be observed. Note that if the failure occurs only in the right or the left control surface, the resulting asymmetric distribution of the aerodynamic forces will generate forces and moments that affect the lateral-directional dynamics. However, this behavior will not be studied in this work since the objective is the analysis of the longitudinal dynamics.

The other actuator failure studied in this work is the jam of the control surface at a certain position. This type of failure could be caused by the failure of the servo actuators and it has serious implications in the aircraft's dynamics given that this control surface stops being available for the control system and, moreover, it will produce forces and moments that affect the lateral-directional dynamics if the failure does not occur in the right and the left surface at the same time. To model this failure, the control surface deflection is kept constant at the desired position after the instant of the failure. It can be done as a step input or following the actuator dynamics. The behavior of the actuator in the event of a failure is shown in Figure 2.4.

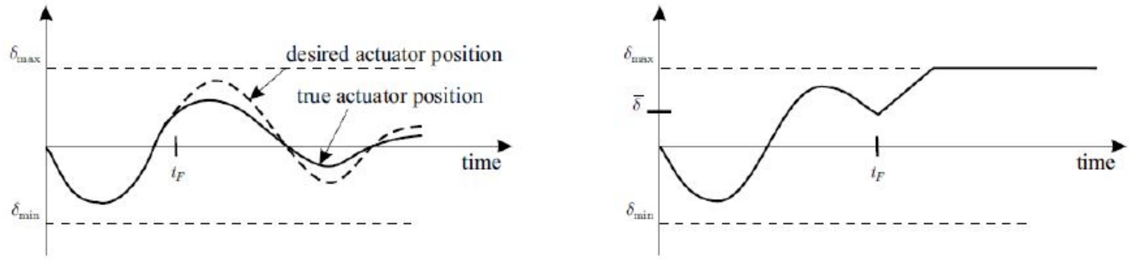


Figure 2.4: Diagrams showing the behavior of the control surface deflection in the event of an actuator failure. The left diagram corresponds to the loss of effectiveness of the control surface and the right one to the jam of the actuator at a certain position.

2.3 Linearization of the aircraft dynamics

Now that the equations that define the dynamics of the aircraft and the aerodynamic model have been presented, it is possible to obtain a linear model of the dynamics by linearizing Eq. (2.4) around an arbitrary equilibrium point. In order to do this, the states of the aircraft can be redefined as the sum of the state in its trim condition and a perturbation around this point:

$$\begin{aligned} V &= V_0 + \Delta V \quad ; \qquad & \alpha &= \alpha_0 + \Delta\alpha \\ q &= q_0 + \Delta q \quad ; \qquad & \theta &= \theta_0 + \Delta\theta \end{aligned} \tag{2.12}$$

In most of the cases, the equilibrium point is chosen so that the time derivatives of the states are null. Therefore, $q_0 = 0$ in the rest of this work. Now, applying first order Taylor series expansion to Eq. (2.4) and expressing the result in matrix form, the following linear system of equations is obtained:

$$\begin{bmatrix} \Delta\dot{V} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_V & X_\alpha & X_q & X_\theta \\ Z_V & Z_\alpha & Z_q & Z_\theta \\ M_V & M_\alpha & M_q & M_\theta \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta V \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} X_{\delta_E} & X_{\delta_C} \\ Z_{\delta_E} & Z_{\delta_C} \\ M_{\delta_E} & M_{\delta_C} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_E \\ \Delta\delta_C \end{bmatrix} \tag{2.13}$$

where the parameters inside of the matrix of the system are:

$$
\begin{aligned}
X_V &= -\frac{2\ q_{d_0}\ S_w\ CD_{\text{trim}}}{m\ V_0} \\
X_\alpha &= -\frac{T_0\ \sin\alpha_0 + 2\ q_{d_0}\ S_w\ K_D\ CL_{\text{trim}}\ CL_\alpha}{m} + g\left(\cos\theta_0\cos\alpha_0 + \sin\theta_0\sin\alpha_0\right) \\
X_q &= -\frac{2\ q_{d_0}\ S_w\ c_w\ K_D\ CL_{\text{trim}}\ CL_q}{2\ m\ V_0} \\
X_\theta &= -g\left(\sin\theta_0\sin\alpha_0 + \cos\theta_0\cos\alpha_0\right) \\
Z_V &= \frac{T_0\ \sin\alpha_0 - q_{d_0}\ S_w\ CL_{\text{trim}} - m\ g\ \left(\cos\theta_0\cos\alpha_0 + \sin\theta_0\sin\alpha_0\right)}{m\ V_0^2 + q_{d_0}\ S_w\ CL_{\dot\alpha}\ \dfrac{c_w}{2}} \\
Z_\alpha &= \frac{-T_0\ \cos\alpha_0 - q_{d_0}\ S_w\ CL_\alpha + m\ g\ \left(-\cos\theta_0\sin\alpha_0 + \sin\theta_0\cos\alpha_0\right)}{m\ V_0 + q_{d_0}\ S_w\ CL_{\dot\alpha}\ \dfrac{c_w}{2\ V_0}} \\
Z_q &= \frac{m\ V_0 - q_{d_0}\ S_w\ CL_q\ \dfrac{c_w}{2\ V_0}}{m\ V_0 + q_{d_0}\ S_w\ CL_{\dot\alpha}\ \dfrac{c_w}{2\ V_0}} \\
Z_\theta &= \frac{m\ g\left(-\sin\theta_0\cos\alpha_0 + \cos\theta_0\sin\alpha_0\right)}{m\ V_0 + q_{d_0}\ S_w\ CL_{\dot\alpha}\ \dfrac{c_w}{2\ V_0}} \\
M_V &= \frac{2\ q_{d_0}\ S_w\ c_w\ CM_{\text{trim}}}{I_{yy}\ V_0} + \frac{q_{d_0}\ S_w\ c_w^2\ CM_{\dot\alpha}}{2\ I_{yy}\ V_0}\ Z_V \\
M_\alpha &= \frac{q_{d_0}\ S_w\ c_w\ CM_\alpha}{I_{yy}} + \frac{q_{d_0}\ S_w\ c_w^2\ CM_{\dot\alpha}}{2\ I_{yy}\ V_0}\ Z_\alpha \\
M_q &= \frac{q_{d_0}\ S_w\ c_w^2\ CM_q}{2\ I_{yy}\ V_0} + \frac{q_{d_0}\ S_w\ c_w^2\ CM_{\dot\alpha}}{2\ I_{yy}\ V_0}\ Z_q \\
M_\theta &= \frac{q_{d_0}\ S_w\ c_w^2\ CM_{\dot\alpha}}{2\ I_{yy}\ V_0}\ Z_\theta
\end{aligned}
\tag{2.14}
$$

and the ones in the control matrix:

$$
\begin{aligned}
X_{\delta_E} &= -\frac{2\ q_{d_0}\ S_w\ K_D\ CL_{\text{trim}}\ CL_{\delta_E}}{m} \\
X_{\delta_C} &= -\frac{2\ q_{d_0}\ S_w\ K_D\ CL_{\text{trim}}\ CL_{\delta_C}}{m} \\
Z_{\delta_E} &= -\frac{q_{d_0}\ S_w\ CL_{\delta_E}}{m\ V_0 + q_{d_0}\ S_w\ CL_{\dot{\alpha}}\ \dfrac{c_w}{2\ V_0}} \\
Z_{\delta_C} &= -\frac{q_{d_0}\ S_w\ CL_{\delta_C}}{m\ V_0 + q_{d_0}\ S_w\ CL_{\dot{\alpha}}\ \dfrac{c_w}{2\ V_0}} \\
M_{\delta_E} &= \frac{q_{d_0}\ S_w\ c_w\ CM_{\delta_E}}{I_{yy}} + \frac{q_{d_0}\ S_w\ c_w^2\ CM_{\dot{\alpha}}}{2\ I_{yy}\ V_0}\ Z_{\delta_E} \\
M_{\delta_C} &= \frac{q_{d_0}\ S_w\ c_w\ CM_{\delta_C}}{I_{yy}} + \frac{q_{d_0}\ S_w\ c_w^2\ CM_{\dot{\alpha}}}{2\ I_{yy}\ V_0}\ Z_{\delta_C}
\end{aligned}
\tag{2.15}
$$

where the subscript 0 corresponds to the equilibrium conditions and $K_D = 1/(\pi\ e\ AR)$ is the induced drag coefficient. It can be observed that the previous parameters depend on the trim conditions, the aerodynamic coefficients and the geometry, mass and inertia of the aircraft. The system shown in Eq. (2.13) is the state space representation of the aircraft's dynamics, with the form:

$$\dot{\boldsymbol{x}} = \boldsymbol{A}\boldsymbol{x} + \boldsymbol{B}\boldsymbol{u} \tag{2.16}$$

where $\boldsymbol{x} = [\Delta V\ \Delta\alpha\ \Delta q\ \Delta\theta]^{\mathrm{T}}$ is the states vector, $\boldsymbol{u} = [\Delta\delta_E\ \Delta\delta_C]^{\mathrm{T}}$ is the input vector and $\boldsymbol{A}$ and $\boldsymbol{B}$ are the system and control matrices respectively. This type of representation is useful to study the dynamics of the aircraft when it operates close to the equilibrium point and for the design of linear control systems. In this work, this model will be used in the preliminary design of the nonlinear controllers to analyze the viability of their implementation in the actual system.

2.4 Model of the sensor failures

As it was mentioned in the first chapter, a sensor failure can have critical consequences during flight given its difficulty to be identified and the need of accurate measurements of the states for the flight controller to work properly. For this reason, the study of the control system's robustness in the event of a sensor failure is performed in this work. To do so, it is necessary to model these failures and include them in the flight simulator. Three different sensor failures are studied in this work: measurement noise, a step bias and a drifting bias.

The measurement noise σ_n is modeled as white Gaussian noise and it is added to the actual value of the state. This type of noise is characterized by its standard deviation and follows a normal distribution with null average value in the time domain. The step bias σ_b is a constant offset of the sensor output and the drifting bias σ_d is a linear addition to the sensor output, as shown in Figure 2.5. The slope of the drift function K_d can vary randomly or triggered by abrupt variations of external parameters, including the sensor input. It can be modeled as a linear function of time t with an initial moment t_f as follows:

$$\sigma_d(t) = K_d(t - t_f) \tag{2.17}$$

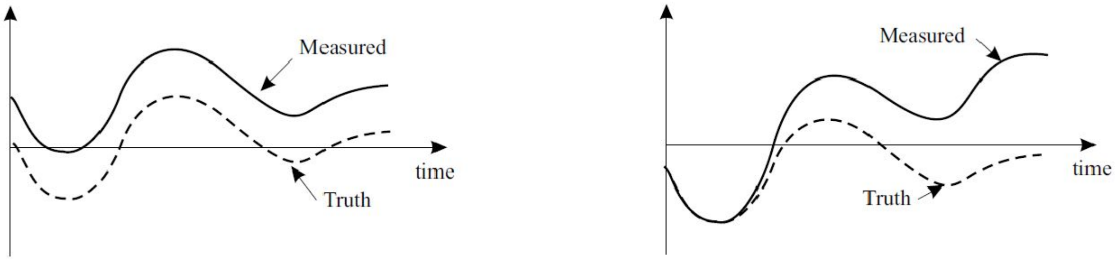


Figure 2.5: Diagrams showing the behavior of the sensor outputs in the event of a step bias (left) and drift bias (right).

Therefore, the resultant output of the sensors x_{sf} will be the sum of the contribution of every failure and the actual state x:

$$x_{sf} = x + \sigma_n + \sigma_b + \sigma_d \tag{2.18}$$

2.5 Model of the actuator dynamics

The dynamics of the actuators play a significant role in the overall performance of the control systems and the aircraft behavior. For this reason, the implementation of an accurate model of the actuator dynamics in the flight simulator is crucial to get reliable results. However, the lack of a complete dynamics model of the real actuators used in the GFF subscale aircraft forces the use of a simpler model that only considers the deflection saturation to a maximum or minimum value and the angular rate saturation that limits the maximum angular velocity of the control surface.

2.6 Simulator layout

In this section, the most important levels of the flight simulator built in *Simulink* are shown. As it was mentioned before, the simulator is based on the *MathWorks* library *Airlib*, although several changes have been made in order to include the actuator and sensor failures, the custom control systems and the possibility of splitting the contribution of each part of the aircraft in the overall aerodynamic coefficients. The first level of the simulator is shown in Figure 2.6.

The Reference block includes the states reference commanded by the pilot or the autopilot, depending on the case. The Control Systems block is where the controllers studied in this work are implemented. This block takes the reference commanded by the pilot and the feedback from the sensors and generates the required control surface deflection. The Actuators block includes the characteristic dynamics of each actuator and the model of the control surface jam. The Sensors block includes the model of the sensor failures. Finally, inside of the General Aircraft Model block, the forces and moments that affect the aircraft's dynamics are modeled and the equations of motion are solved. The content of this block is shown in Figure 2.7.

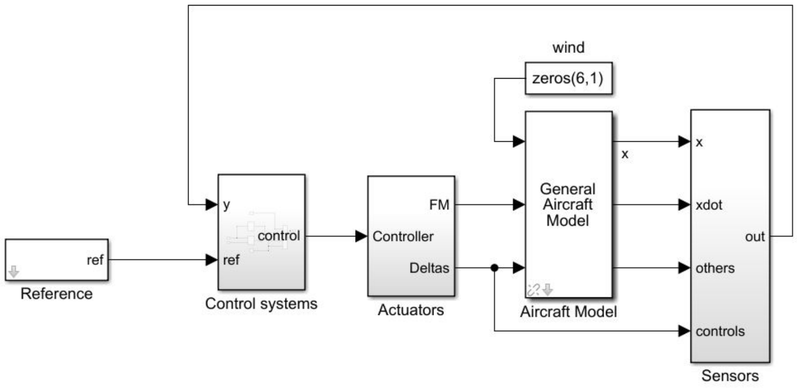


Figure 2.6: First level of the flight simulator implemented in *Simulink*.

The Airdata group block includes a model of the atmosphere and the flight conditions as a function of velocity and altitude such as temperature, Mach number, density, pressure and gravitational acceleration. The content of the Aerodynamics group is shown in Figure 2.8. This block includes the model of the aerodynamic coefficients as described in Section 2.2 and the conversion to the body-fixed frame. It can also be observed that the possibility of including a model of the lateral-directional aerodynamic derivatives is available, although it is not used for this work.

Finally, the content of the Aircraft Equations of Motion block is shown in Figure 2.9. This block gathers all the forces and moments applied to the aircraft and integrates the 12 ordinary differential equations that define the aircraft's dynamics.

In the following chapters, the theoretical basis of the different control systems studied in this work will be shown and the content of the Control Systems block will be revealed.

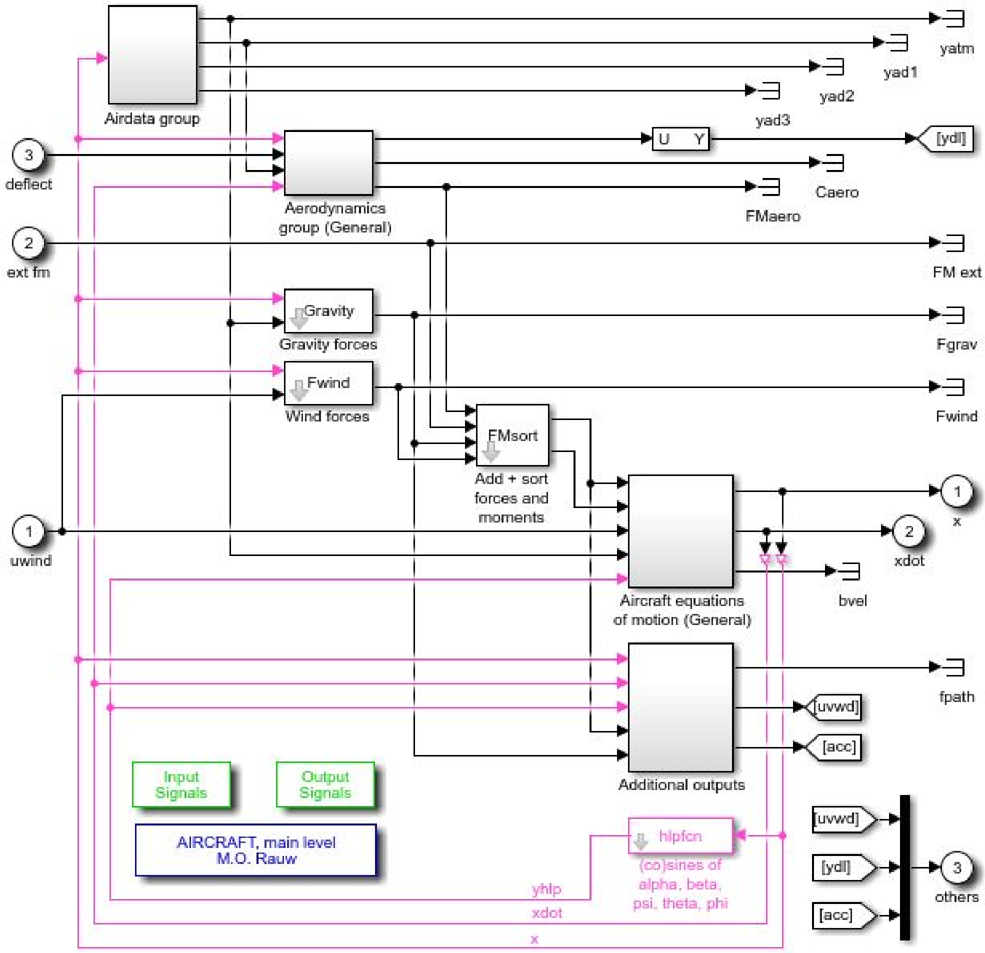


Figure 2.7: General Aircraft Model block.

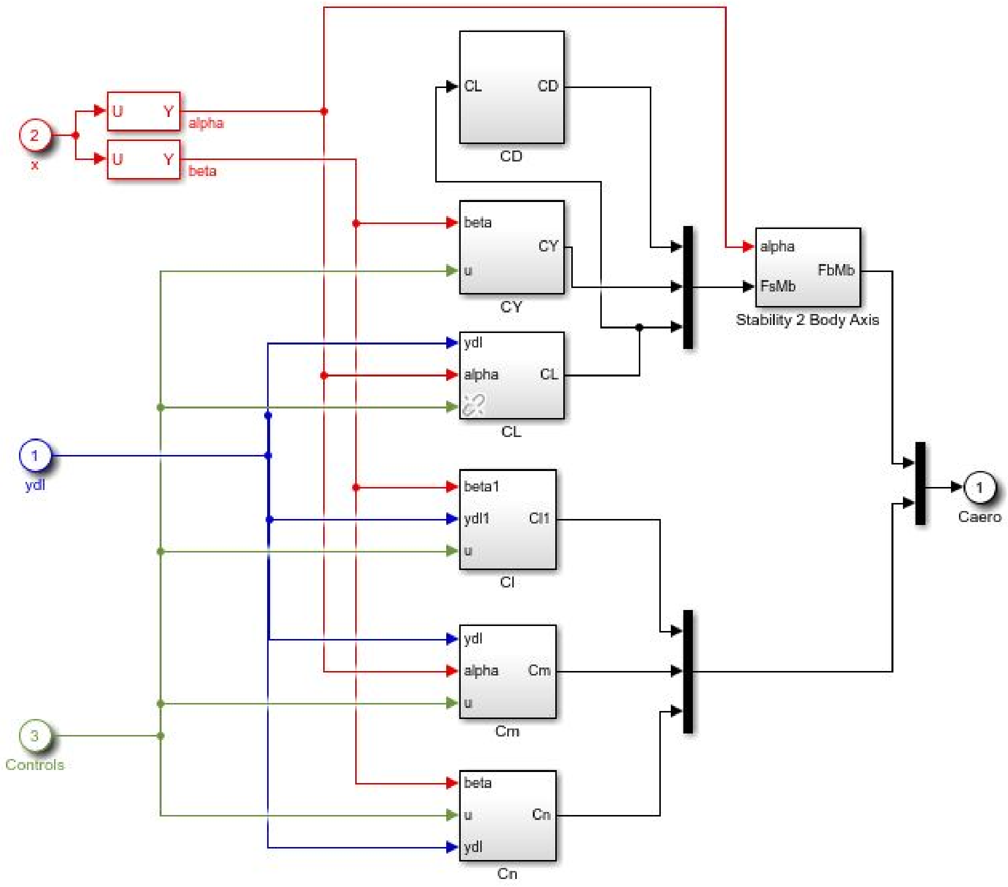


Figure 2.8: Aerodynamics group block.

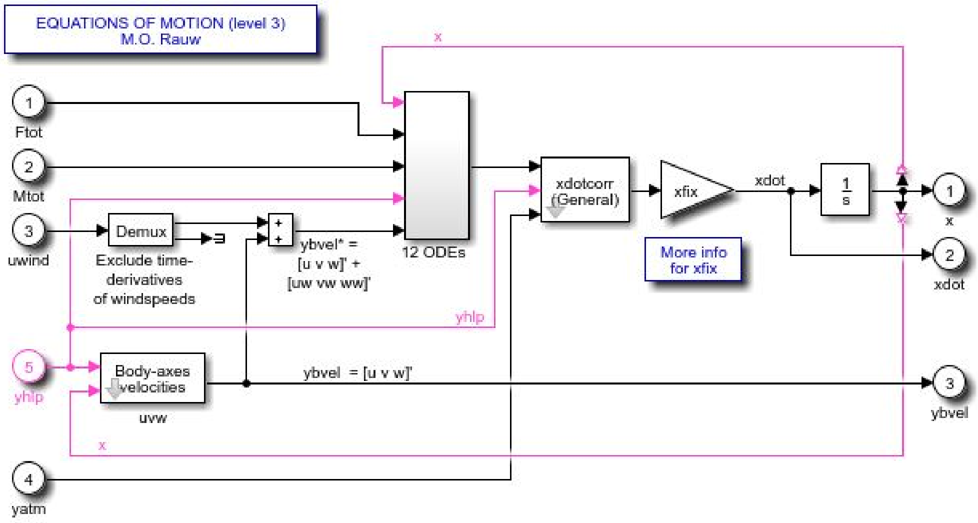


Figure 2.9: Aircraft Equations of Motion block.

3
Control Systems

In this chapter, the mathematical foundation of the different control systems studied in this work is presented. As mentioned before, the main objective of this thesis is to explore several control laws that allow the *Generic Future Fighter* to follow the commanded pitch rate under different flight conditions, including high angle of attack maneuvers and possible failures in the actuators or the sensors. Therefore, the classical control techniques consisting in linear models with gain scheduling are not of interest since the modeling of the aircraft dynamics for these conditions may be quite complex and inaccurate.

In order to make the system robust enough when aerodynamic nonlinear effects play a significant role in the aircraft's dynamics, the proposed control systems are based on the so called dynamic inversion or feedback linearization. Moreover, this control law is combined with the Model Reference Adaptive Control scheme. In this algorithm, a linear transfer function that relates the desired output variable to the control input is adopted as a reference model and the controller forces the aircraft to track the desired dynamics. This reference model must satisfy the U.S. *Military Specification for the Flying Qualities of Piloted Airplanes MIL-F-8785C* [9] regarding natural frequency and damping of the longitudinal dynamics short period. In addition, an adaptive algorithm estimates the characteristic parameters of the aircraft's dynamics online so that it is not necessary to have a highly reliable model of the dynamics. As will be shown in later chapters, these kind of adaptive algorithms also allow the aircraft to operate correctly in the event of a failure. Finally, two different adaptive algorithms are studied: the so called simple adaptation and the neural networks.

In the following sections, the reference model adopted for this work as well as the mathematical background of the dynamic inversion controller and the adaptive algorithms will be shown.

3.1 Reference model

As mentioned before, the reference model in chosen so that the *Military Specification for the Flying Qualities of Piloted Airplanes MIL-F-8785C* are satisfied. These specifications define airplane classes, flight phases and flying quality levels so that different modes can be specified for the various combinations. The definition of the flight phases is the following:

- Category A: Nonterminal flight phases generally requiring rapid maneuvering.
- Category B: Nonterminal flight phases normally accomplished using gradual maneuvers without precision tracking, although accurate flight-path control may be required.
- Category C: Terminal flight phases normally accomplished using gradual maneuvers and usually requiring accurate flight-path control.

On the other hand, the definition of the flying quality levels is the following:

- Level 1: Flying qualities adequate for the mission flight phase.
- Level 2: Flying qualities adequate to accomplish the mission flight phase, but some increase in pilot workload or degradation in mission effectiveness exists.
- Level 3: Flying qualities such that the airplane can be controlled safely, but pilot workload is excessive, or mission effectiveness is inadequate, or both.

Following these definitions, the short-period requirements are specified in terms of the natural frequency and damping of the short-period mode. The damping ratio ξ_{sp} requirements are shown in Table 3.1.

Level	Cat. A & C Flight Phases		Cat. B Flight Phases	
	Minimum	Maximum	Minimum	Maximum
1	0.35	1.30	0.30	2.00
2	0.25	2.00	0.20	2.00
3	0.15	no limit	0.15	no limit

Table 3.1: Short-period damping ratio limits.

The requirements on equivalent undamped natural frequency $\omega_{n_{sp}}$ are shown in Table 3.2 and are specified indirectly in terms of the quantity $\omega_{n_{sp}}^2/\left(n/\alpha\right)$. The denominator (n/α) of this term is the aircraft load factor response to angle of attack in g's per radian.

Level	Cat. A Phases		Cat. B Phases		Cat. C Phases	
	Min.	Max.	Min.	Max.	Min.	Max.
1	0.28 $\omega_n \geq 1$	3.60	0.085	3.60	0.16 $\omega_n \geq 0.7$	3.60
2	0.16 $\omega_n \geq 0.6$	10	0.038	10	0.096 $\omega_n \geq 0.4$	10
3	0.16	no limit	0.038	no limit	0.096	no limit

Table 3.2: Limits on $\omega_{n_{sp}}^2/\left(n/\alpha\right)$.

The reference model is chosen so that the flight phase is category A, since the controller is designed to control the short-period mode characterized by high frequency motion and rapid maneuvers and the flying qualities are level 1 since good handling qualities during the whole flight phase are desired. Therefore, considering the requirements established by *MIL-F-8785C* and the desired behavior of the aircraft, $\omega_{n_{sp}}$ and ξ_{sp} are arbitrarily chosen and shown in Table 3.3:

$\omega_{n_{sp}}$ [rad/s]	ξ_{sp} [-]
10	0.8

Table 3.3: Natural frequency and damping ratio chosen for the reference model of the short-period.

Now that the dynamic properties of the reference model have been established and neglecting the effects of the phugoid mode, a second-order linear transfer function can be defined to describe the reference system. The objective of the controllers studied in this work is to control the pitch rate q using the elevons and the canards. However, the canards deflection is proportionally related to the elevons deflection as will be shown in later sections. Therefore, the transfer functions needed for the reference model only depend on the elevons deflections and they are the following:

$$\frac{q(s)}{\delta_E(s)} = \frac{6s+600}{s^2+16s+100}; \qquad \frac{\dot{q}(s)}{\delta_E(s)} = \frac{6s^2+600s}{s^2+16s+100} \tag{3.1}$$

The transfer function that relates the time derivative of q to δ_E is defined because it is necessary to implement the dynamic inversion algorithm. The numerator of the transfer functions is chosen so that the maximum deflection possible of the elevons provides the desired maximum pitch rate.

3.2 Dynamic inversion

All the controllers developed in this work are based on the dynamic inversion technique, also known as feedback linearization. The dynamic inversion controller takes into account the nonlinearities of the aircraft and thus does not require gain scheduling. As such it is suitable for a wide range of operating conditions, including high-angle-of-attack and hypervelocity design. This thesis explores different approaches to the dynamic inversion: for linear models and for nonlinear models, including adaptive algorithms and without them. The advantages and drawbacks of each one of these methods will be presented in later chapters, although it is in nonlinear systems where the dynamic inversion shows its true power. In the following subsections, the mathematical background of the different dynamic inversion techniques studied is shown. The content of this section is based on Section 5.8 of [10] and [11].

3.2.1 Dynamic inversion for linear models

Let the system be described in state-space form by:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{3.2}$$

with state $x(t) \in \mathbb{R}^n$, input $u(t) \in \mathbb{R}^m$ and output $y(t) \in \mathbb{R}^p$. The entire state $x(t)$ is available for feedback purposes, i.e. it must be possible to measure all the states. It is assumed that the system is square, that is, the number of inputs m is equal to the number of outputs p so that vectors $u(t)$ and $y(t)$ have the same dimension.

It is desired to control the output $y(t)$ so that it follows a desired reference $r(t)$. In the case of study, the output is q and the desired reference is the desired pitch rate q_{des} obtained from the reference model. The tracking error e is defined as:

$$e(t) = r(t) - y(t) \tag{3.3}$$

In dynamic inversion, one differentiates the $y(t)$ until $u(t)$ appears in the expression for the derivative. This is known technically as input-output feedback linearization. Taking the first derivative yields:

$$\dot{y} = C\dot{x} = CAx + CBu \tag{3.4}$$

where $u(t)$ appears if matrix CB is not zero. In this case, since the system is square, so is matrix CB. If CB is nonsingular then it is done. If not, the expression is differentiated until the coefficient multiplying $u(t)$ is nonzero. For aircraft, it is generally the case that CB is nonsingular because of the way in which the control actuators enter into the aircraft dynamic equations, with one actuator for each degree of freedom.

Since the objective is to make $y(t)$ track the reference r, it is possible to obtain the control input $u(t)$ necessary to make that happen by substituting $\dot{y}$ for $\dot{r}$ in Eq. 3.4 and solving for $u(t)$:

$$u(t) = (CB)^{-1} (\dot{r} - CAx + v) \tag{3.5}$$

where v is an auxiliary input used in order to improve the control law performance and will be defined soon.

Substituting this expression into Eq. 3.4 yields:

$$\dot{y} = CAx + CB \left[(CB)^{-1} (\dot{r} - CAx + v) \right] = \dot{r} + v \tag{3.6}$$

or:

$$\dot{e} = -v \tag{3.7}$$

This expression is the error dynamics. To complete the design it is necessary to select $v(t)$ so that the system is stable. This task can be done by using a wide variety of techniques including robust control, LQR/LTR and other linear system design techniques. A simple but effective choice for $v(t)$ is $v(t) = Ke$, where K is a positive definite gain matrix. Therefore, the closed-loop error dynamics are given by:

$$\dot{e} = -Ke \tag{3.8}$$

which is a stable system as long as K is positive definite. The gain K should be selected so that the closed-loop system satisfies the *MIL-F-8785C* flying qualities requirements.

Finally, the overall dynamic inversion control input is:

$$u = (CB)^{-1}(\dot{r} - CAx + Ke) \tag{3.9}$$

Note that the term involving the error is an outer proportional feedback tracking loop, the term CAx is an inner control loop using full state-variable feedback and the term involving $\dot{r}$ is a feedforward term known as velocity feedforward that greatly improves the tracking accuracy of the closed-loop system.

In order to implement a dynamic inversion control algorithm, the dynamics of the system must be accurately known and a high-fidelity model of the aircraft dynamics is built into the controller. Moreover, full state feedback is required for the inner loop.

3.2.2 Nonlinear dynamic inversion

The procedure followed to derive the nonlinear dynamic inversion control input is very similar to the one followed for the linear case. Let the plant be described in nonlinear state-variable form by:

$$\begin{aligned} \dot{x} &= f(x) + g(x)u \\ y &= h(x) \end{aligned} \tag{3.10}$$

As in the linear case, the number of inputs is assumed to be equal to the number of outputs so that vectors $u(t)$ and $y(t)$ have the same dimensions. The tracking error e is defined as in Eq. 3.3. Differentiating the output yields:

$$\dot{y} = \frac{\partial h}{\partial x}\dot{x} = \frac{\partial h}{\partial x}f(x) + \frac{\partial h}{\partial x}g(x)u \equiv F(x) + G(x)u \tag{3.11}$$

From this point, the control law is obtained by following a procedure analogous to the linear case:

$$u = G^{-1}(x)\left[\dot{r} - F(x) + Ke\right] \tag{3.12}$$

Note that in this case, the functions $F(x)$ and $G(x)$ that define the dynamics of the aircraft are nonlinear functions. Therefore, the nonlinear model of the dynamics must be known in order to implement this controller and the use of full lookup tables could be necessary. The structure of the nonlinear dynamic inversion controller is shown in Figure 3.1.

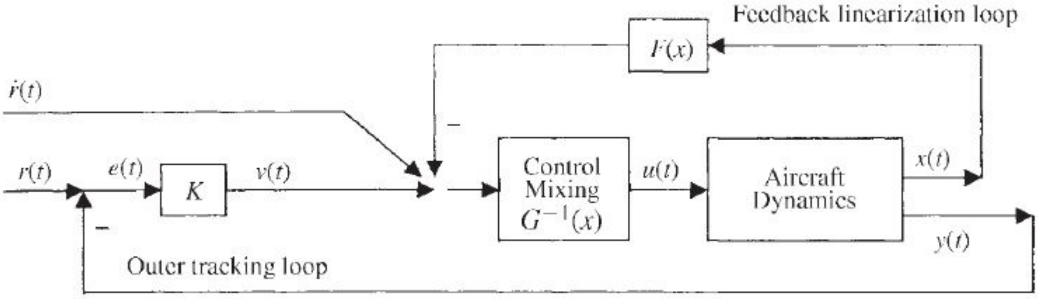


Figure 3.1: Structure of the nonlinear dynamic inversion controller. Source: [10].

3.3 Adaptive dynamic inversion

The dynamic inversion algorithms shown in the previous section consider that the available model of the aircraft dynamics is perfect. However, the obtention of a high-fidelity model for the whole flight envelope is normally quite complex and expensive given the high cost of the flight tests or computer simulations that are necessary to gather the data. In particular, adaptive control is specially interesting for this thesis since the resources available to gather data about the GFF subscale model are limited and the model of the aircraft dynamics is not accurate enough to implement a traditional dynamic inversion algorithm. The adaptive controller, whose structure is shown in Figure 3.2, makes it possible to implement a dynamic inversion algorithm without knowing the parameters that characterize the dynamics since they are estimated online.

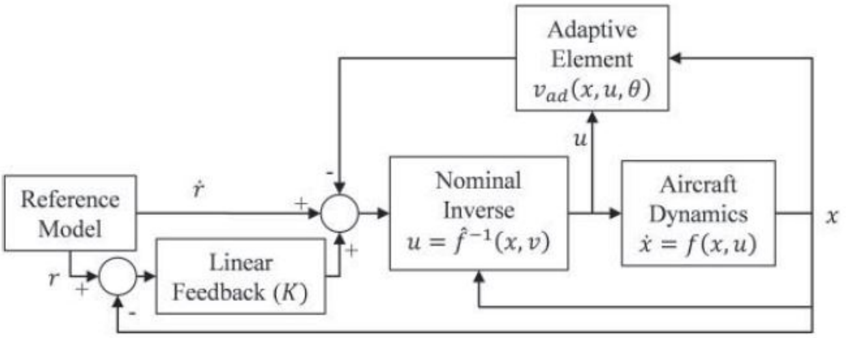

Adaptive Element $v_{ad}(x,u,\theta)$
u
Reference Model
$\dot{r}$
+
-
Nominal Inverse $u=\hat{f}^{-1}(x,v)$
Aircraft Dynamics $\dot{x}=f(x,u)$
x
r
+
-
+
Linear Feedback (K)

Figure 3.2: Structure of the model reference adaptive control system based on dynamic inversion. Source: [10].

In order to implement an adaptive dynamic inversion controller, the form of the equations that describe the dynamics must be modified in order to get an expression in terms of a regressor $\varphi(t,x)$ made up of functions that describe the evolution of the states and a vector of parameters ϑ assumed to be constant or slowly varying with time. The expression is the following:

$$\dot{x} = f(t,x,v) = \varphi(t,x)\vartheta + v + \Delta(t,x,v) \tag{3.13}$$

where v is a vector of pseudocontrols and $\Delta(t,x,v)$ is a disturbance term caused by failures. The pseudocontrol input is established so that the plant follows the reference:

$$v = \dot{r}(t) - \varphi(t,x)\hat{\vartheta} + Ke(t) \tag{3.14}$$

where $\hat{\vartheta}$ is the estimation of the true parameters vector ϑ. Note than when the estimation of the parameters $\hat{\vartheta}$ matches the actual value ϑ, the plant follows the reference $\dot{x} = \dot{r}$.

3.3.1 Example of application

In order to clarify the procedure that needs to be followed to implement this type of controller, the derivation of Eq. 3.13 and the obtention of the regressor $\varphi(t,x)$ and the parameters vector ϑ will be explained. To do so, the linear model of the GFF dynamics is used in order to simplify the mathematics and make the example more illustrative and easier to follow. The procedure followed in the nonlinear case is analogous.

Let the plant be described in state-space form as:

$$\begin{aligned} \dot{x} &= Ax + B\delta_E + \Delta \\ y &= Cx \end{aligned} \tag{3.15}$$

where $x = [V \ \alpha \ q \ \theta]^T$. Note that the control input is the elevon deflection δ_E and Δ is a vector of disturbances caused by failures. The objective is to control the pitch rate q, therefore, the output matrix is $C = [0 \ 0 \ 1 \ 0]$.

Since it is assumed that the model of the dynamics is not perfect, it is necessary to define the best approximation available of the actual matrices A and B as $\hat{A}$ and $\hat{B}$ respectively. Therefore, the dynamic inversion control input shown in Eq. 3.9 will be a function of these estimated matrices as:

$$\delta_E = \left(C\hat{B}\right)^{-1}\left(\dot{q}_{des} - C\hat{A}x + Ke\right) \tag{3.16}$$

where $\dot{q}_{des}$ is the derivative of the desired pitch rate. Substituting this control law in the equation of the dynamics, the expression of the closed-loop is obtained:

$$\begin{aligned} \dot{x} &= Ax + B\left(C\hat{B}\right)^{-1}\left[\dot{q}_{des} - C\hat{A}x + Ke\right] + \Delta = \\ &= \left[A - B\left(C\hat{B}\right)^{-1}C\hat{A}\right]x + B\left(C\hat{B}\right)^{-1}(\dot{q}_{des} + Ke) + \Delta \end{aligned} \tag{3.17}$$

Premultiplying this expression by C, the derivative of the output $\dot{q}$ is obtained:

$$\dot{q} = \left[CA - CB\left(C\hat{B}\right)^{-1}C\hat{A}\right]x + CB\left(C\hat{B}\right)^{-1}(\dot{q}_{des} + Ke) + \Delta_q \tag{3.18}$$

where Δ_q is the disturbance in the pitch rate. Since this expression is linear with the states, the closed-loop equation can also be expressed as:

$$\dot{q} = \vartheta_V V + \vartheta_\alpha \alpha + \vartheta_q q + \vartheta_\theta \theta + \vartheta_{\dot{q}_{des}} \dot{q}_{des} + \vartheta_e K e + \Delta_q \tag{3.19}$$

This expression can be modified as follows:

$$\dot{q} = \vartheta_V V + \vartheta_\alpha \alpha + \vartheta_q q + \vartheta_\theta \theta + \dot{q}_{des} + (\vartheta_{\dot{q}_{des}} - 1)\dot{q}_{des} + Ke + (\vartheta_e - 1)Ke + \Delta_q \tag{3.20}$$

Note that the parameters $\vartheta_{\dot{q}_{des}}$ and ϑ_e come from the matrix product $CB\left(C\hat{B}\right)^{-1}$. Therefore, when the estimation of the model is close to the actual dynamics, the parameters $\vartheta_{\dot{q}_{des}}$ and ϑ_e will be close to 1 and the terms $(\vartheta_{\dot{q}_{des}} - 1)$ and $(\vartheta_e - 1)$ in 3.20 can be neglected. This yields:

$$\dot{q} = \vartheta_V V + \vartheta_\alpha \alpha + \vartheta_q q + \vartheta_\theta \theta + \dot{q}_{des} + Ke + \Delta_q \tag{3.21}$$

Now it is possible to express the dynamics of the aircraft in the form of Eq. 3.13. The regressor $\varphi(t, x)$ and the parameters vector ϑ are defined as:

$$\varphi(t, x) = \begin{bmatrix} V & \alpha & q & \theta \end{bmatrix}; \qquad \vartheta = \begin{bmatrix} \vartheta_V & \vartheta_\alpha & \vartheta_q & \vartheta_\theta \end{bmatrix}^T \tag{3.22}$$

The system can now be expressed as:

$$\dot{q} = \varphi(t, x)\vartheta + \dot{q}_{des} + Ke + \Delta_q \tag{3.23}$$

Finally, the disturbance term can be included in the regressor by adding a constant parameter ϑ_c as:

$$\varphi(t, x) = \begin{bmatrix} V & \alpha & q & \theta & 1 \end{bmatrix}; \qquad \vartheta = \begin{bmatrix} \vartheta_V & \vartheta_\alpha & \vartheta_q & \vartheta_\theta & \vartheta_c \end{bmatrix}^T \tag{3.24}$$

Now the system is ready to implement the adaptive dynamic inversion controller. In the following sections, the different adaptive algorithms used in this thesis are explained.

3.4 Adaptive algorithms

The objective of the adaptation is to estimate the parameters vector ϑ that define the dynamics of the aircraft. In this work, two different adaptation algorithms have been explored: the simple adaptation and the neural networks.

3.4.1 Simple adaptation

This method is the simplest one although it has been proven to be effective, as will be shown in later chapters. The simple adaptation provides an update law for the estimated parameters $\hat{\vartheta}$ of the form:

$$\dot{\hat{\vartheta}} = -\gamma\varphi(t,x)^T e \tag{3.25}$$

where γ is a positive definite learning rate matrix. It can be observed that this update law depends proportionally on the error e. Therefore, when the output matches the reference, the adaptive algorithm assumes that the system has been correctly identified.

One of the most critical aspects of the adaptive algorithms is their stability. In particular, a control system that has not been proved to be stable is unacceptable for aerospace applications given the catastrophic consequences that the misbehavior of the controller could cause. For this reason, a proof of stability for this adaptive algorithm is derived.

Substituting the control law shown in Eq. 3.14 into Eq. 3.13 yields:

$$\dot{x} = \varphi(t,x)\vartheta + \dot{r} - \varphi(t,x)\hat{\vartheta} + Ke \tag{3.26}$$

Operating:

$$\dot{x} - \dot{r} - Ke = -\varphi(t,x)\left[\hat{\vartheta} - \vartheta\right]; \qquad \rightarrow \qquad \dot{e} + Ke = \varphi(t,x)\tilde{\vartheta} \tag{3.27}$$

where $\tilde{\vartheta} = \hat{\vartheta} - \vartheta$. To prove stability, the following positive definite function is used as a Lyapunov candidate function:

$$V_a = \frac{1}{2}e^T e + \frac{1}{2}\tilde{\vartheta}^T\gamma^{-1}\tilde{\vartheta} \tag{3.28}$$

The time derivative of this function is:

$$\begin{aligned}\frac{dV_a}{dt} &= \frac{d}{de}\left(\frac{1}{2}e^t e\right)\frac{de}{dt} + \frac{d}{d\tilde{\vartheta}}\left(\frac{1}{2}\tilde{\vartheta}^T\gamma^{-1}\tilde{\vartheta}\right)\frac{d\tilde{\vartheta}}{dt} = e^T\dot{e} + \tilde{\vartheta}^T\gamma^{-1}\dot{\tilde{\vartheta}} = \\ &= -e^T Ke + e^T\varphi(t,x)\tilde{\vartheta} - \tilde{\vartheta}^T\gamma^{-1}\gamma\varphi(t,x)^T e\end{aligned} \tag{3.29}$$

Since the result is a scalar, it is obvious that $e^T\varphi(t,x)\tilde{\vartheta} = \tilde{\vartheta}^T\varphi(t,x)^T e$. Therefore:

$$\dot{V}_a = -e^T Ke \tag{3.30}$$

Since $V_a = 0$ when $e = 0$ and $\dot{V}_a < 0$ always, $\dot{e}$ is uniformly continuously bounded, which allows to conclude that e converges to 0 asymptotically, making the system asymptotically stable (as a result of Barbalat's lemma). This theorem shows the ability of the adaptive control law to guarantee asymptotic tracking of the desired input signal. However, no guarantees have been made for adaptive control in the presence of unstructured uncertainty in the model of the dynamics.

3.4.2 Neural networks

Neural networks (NN) for identification and control were first proposed by Narendra and Parthasarathy [12]. Results in this initial research were limited to simulation and no proofs of stability in the closed-loop were provided. First results in proving stability of online feedback linearization adaptive Single Hidden Layer (SHL) neural network augmented controllers were obtained by Chen and Khalil [13] for discrete time systems, and by Chen and Liu [14] for continuous time systems.

Regarding the types of NN, a useful distinction for classifying these NN formulations is parametric NN and nonparametric NN. In the parametric NN, model error has a known basis function with one or more unknown parameters. In the nonparametric NN the designer does not explicitly include knowledge about the functional form of the model error. Here, sufficient parameterization is included to allow the NN to perform the estimation to desired accuracy, even for a nonlinear system.

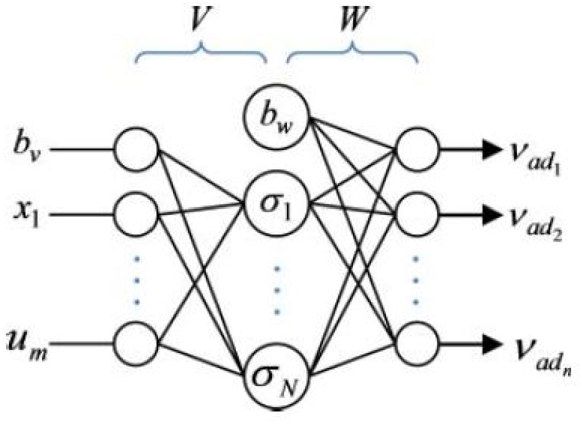


Figure 3.3: Diagram of the nonparametric Single Hidden Layer Neural Network structure. Source: [10].

In this work, the nonparametric single hidden layer NN shown in Figure 3.3 is explored. In this type of NN there is one single layer of neurons between the input layer and the output layer. The SHL uses a squashing function $\sigma(z) \in \mathbb{R}^N$ at the hidden layer neurons, such as:

$$\sigma_j(z) = \frac{1}{1 + e^{-a_j z}} \tag{3.31}$$

for $j = 1, ..., N$, with the value a_j chosen to be different for each j and N is the number of hidden layer neurons. Using this, the adaptive part of the pseudocontrol input obtained with SHL NN can be written as:

$$v_{ad}(x, u) = W^T \sigma \left(V^T \bar{x} \right) \tag{3.32}$$

where $W \in \mathbb{R}^{N \times n}$ are the output weights, $V \in \mathbb{R}^{n+m+1 \times N}$ are the input weights and $\bar{x} \in \mathbb{R}^{n+m+1}$ is the input to the NN, being in the case of study a vector of nonlinear functions of the states, similar to the regressor shown in Eq. 3.24. The complete set of the elements of W and V are the NN adjustment parameters.

An important question that comes up is the choice of the number of hidden layer neurons N. With a small number, it is clear by inspection that there may be insufficient capability to curve fit a reasonable model error function. Adding additional hidden layer neurons will certainly improve this curve-fitting capability. However, there will be diminishing returns if the number increases beyond some level. In practice, it has been found that the incremental benefit is not particularly great beyond $N = 5$ for the flight control problems [10]. Therefore, $N = 5$ is the number of hidden layer neurons used in this work.

The universal approximation theorem for the SHL NN tells us that the fitting error can be bounded within a compact set of states and plant inputs. It can be achieved for any chosen error bound by adding additional middle layer neurons to the NN [15]. There are many examples in the literature of proofs of boundedness of all system signals for these type of controllers. These proofs typically involve a Lyapunov candidate function that decreases outside of a compact set. This ensures convergence to a set containing zero tracking error.

The adaptive laws considered in this work come directly from a proof of boundedness. The functions that describe the training of the NN are the following:

$$\begin{aligned} \dot{W} &= -\left[\left(\sigma - \sigma' V^T \bar{x}\right) e^T + \lambda ||e|| W\right] \Gamma_W \\ \dot{V} &= -\Gamma_V \left[\bar{x} e^T W^T \sigma' + \lambda ||e|| V\right] \end{aligned} \tag{3.33}$$

where Γ_W and Γ_V are appropiately dimensioned diagonal matrices of learning rates. The matrix σ' is the gradient of σ. The e-modification scalar $\lambda > 0$ is necessary for the associated boundedness theorem proof. Note the importance of the tracking error. When $e = 0$, these parameters do not change.

3.5 Inclusion of the canards in the controller

All the controllers shown in this chapter have been designed for a SISO system because the dynamic inversion technique requires the same number of inputs and outputs. Therefore, since the objective is to control the pitch rate q, only one control input is necessary, being the elevons the ones used because their control power is higher than the control power of the canards. However, the use of the canards together with the elevons provides several advantages in the control of the aircraft. In the first place, the saturation of the actuators is a critical problem in the dynamic inversion controller because the algorithm is not aware of the saturation and the control input obtained from the control law may be out of the range of possible deflection of the actuator. This phenomenon can cause the loss of control of the aircraft. The use of the canards together with the elevons reduces the elevon deflection necessary to produce the desired pitch moment, preventing saturation.

Another advantage of the use of the canards is that the control system is more robust in the event of an actuator failure. If only the elevons are used, the jam of the control surface implies the loss of controllability of the aircraft. However, the aircraft would still be controllable if the canards are used.

For these reasons, the canards are included in the controller. In order to not modify the SISO designs previously explained, the canard deflection will be proportional to the elevon deflection by following the next relation:

$$\delta_C = -0.5\delta_E \tag{3.34}$$

The factor -0.5 has been chosen arbitrarily. It is negative because the deflection of the canards must be in the opposite direction to the elevons deflection in order to produce the desired pitching moment. Regarding its value, it has been chosen after considering the aerodynamic effects that the canard deflection induces in the main wing, such as turbulence and downwash. In order to reduce this effects, it is interesting to keep the canard deflection as low as possible while having the desired control power.

4
Simulation Results

Now that the control systems studied in this work have been introduced, the results obtained from the simulation of the longitudinal dynamics of the GFF are presented. The main objective of this chapter is to show the performance of the different controllers for several flight conditions, including actuator and sensor failures, the presence of noise in the measurement of the states and the mismatch between the dynamics model and the actual behavior of the aircraft. As a remainder, the different controllers (all of them based on model reference control) included in the simulator are the following:

- Non-adaptive linear dynamic inversion controller.
- Adaptive linear dynamic inversion controller based on simple adaptation.
- Adaptive linear dynamic inversion controller based on neural networks.
- Non-adaptive nonlinear dynamic inversion controller.
- Adaptive nonlinear dynamic inversion controller based on simple adaptation.
- Adaptive nonlinear dynamic inversion controller based on neural networks.

Moreover, since the final objective of the thesis is to implement one control system into the actual subscale model of the GFF available in the laboratory, it is necessary to decide which of the simulated controllers is the best option in terms of complexity, cost and performance to be eventually implemented. In order to make a decision, the following figures of merit are analyzed:

- Computational cost, measured as computation time.
- Memory needed to store the parameters of the controller.
- Mean square error between the reference and the actual q.
- Robustness in the event of noise, failures and changes in the operation point of the flight envelope.

Regarding the simulations, the initial flight conditions for which the simulations have been performed are shown in Table 4.1. The linear models of the dynamics used in the design of the linear controllers have been obtained by using these initial conditions as trim conditions.

V_{ini} [m/s]	h_{ini} [m]	$\gamma_{s_{ini}}$ [deg]
40	60	0

Table 4.1: Initial flight conditions used for the simulations.

where V_{ini} is the initial velocity, h_{ini} is the initial altitude and $\gamma_{s_{ini}}$ is the initial slope of the trajectory.

Finally, the control input commanded by the pilot is a set of two elevon doublets characterized by a duration of 2 seconds each and an amplitude of 2 degrees as shown in Figure 4.1.

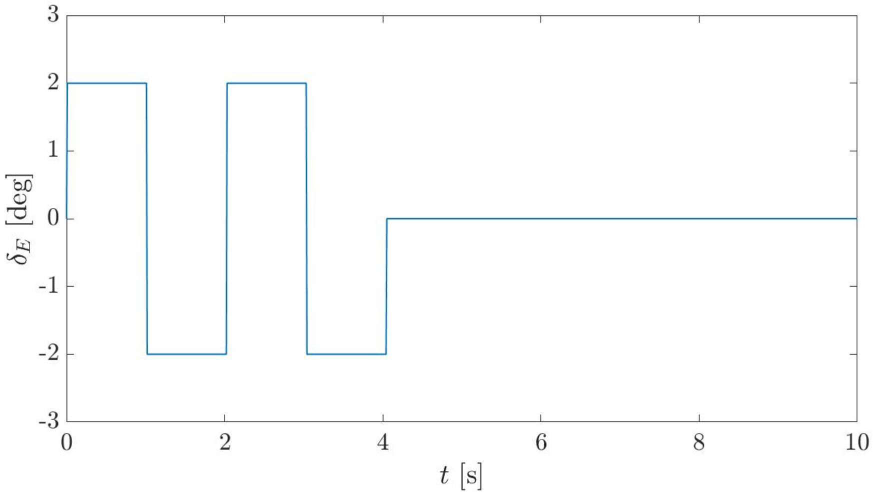


Figure 4.1: Control input commanded by the pilot in the simulations.

The results of the simulations are shown in the following sections and the optimal controller proposed to be implemented in the subscale model is presented at the end of the chapter after having analyzed the different figures of merit.

4.1 Nominal conditions

In this section, the results obtained from the simulations for nominal conditions are shown. The temporal evolution of the pitch rate q is shown in Figure 4.2, the control surface deflections in Figure 4.3 and the quadratic error in Figure 4.4. It can be observed that all the controllers are able to track the reference satisfactorily, although some of them are more accurate than the rest. Analyzing e^2, it can be seen that there is a high peak in the error at the beginning of the simulation for the adaptive controllers. The cause of this result is that the vector of estimated parameters is initialized with zeros. Therefore, the algorithm does not have any information about the system at the beginning and it takes some iterations until it is able to estimate the model. On the other hand, the non-adaptive algorithms track the reference almost perfectly because the models of the dynamics implemented into the algorithm match the actual system. The behavior of these controllers when there exist a mismatch between the model and the actual dynamics will be shown in later sections. Moreover, it can also be observed that e^2 converges to zero when the doublets are finished for all the cases studied.

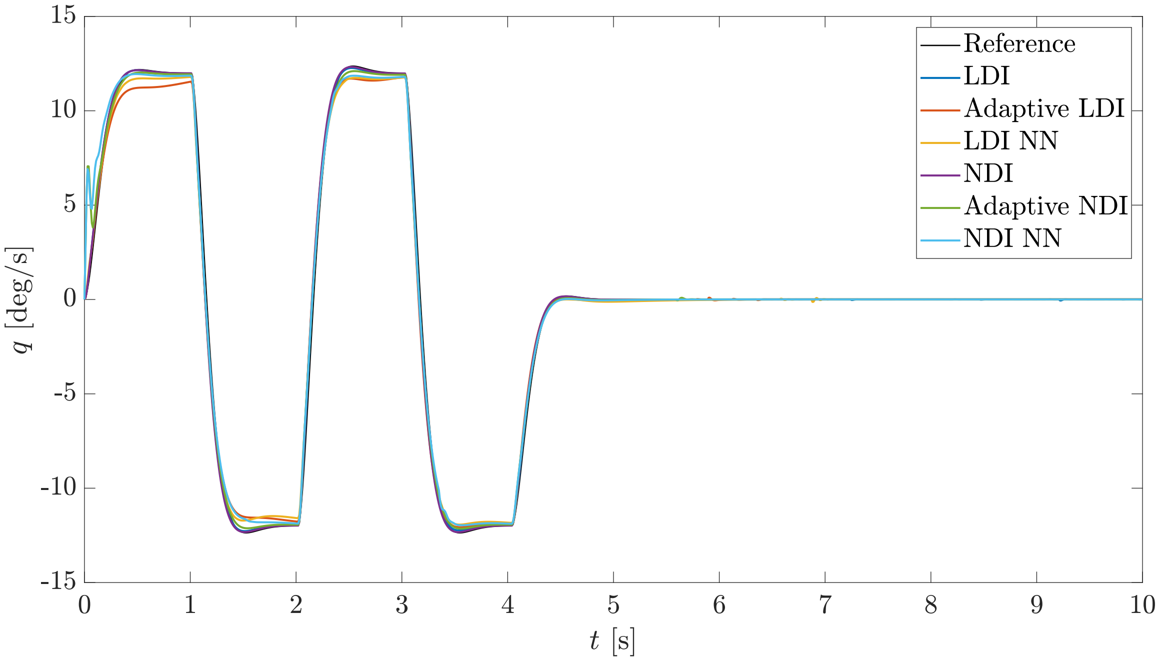


Figure 4.2: Comparison of the pitch rate for the different controllers under nominal conditions.

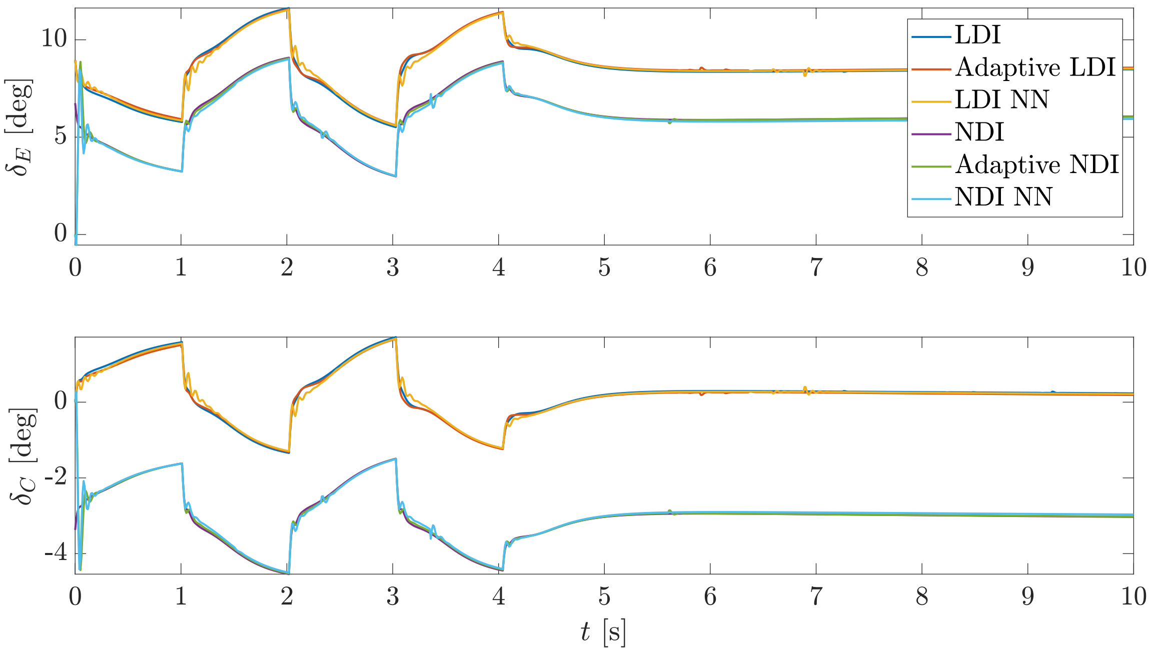


Figure 4.3: Comparison of the control surface deflections for the different controllers under nominal conditions.

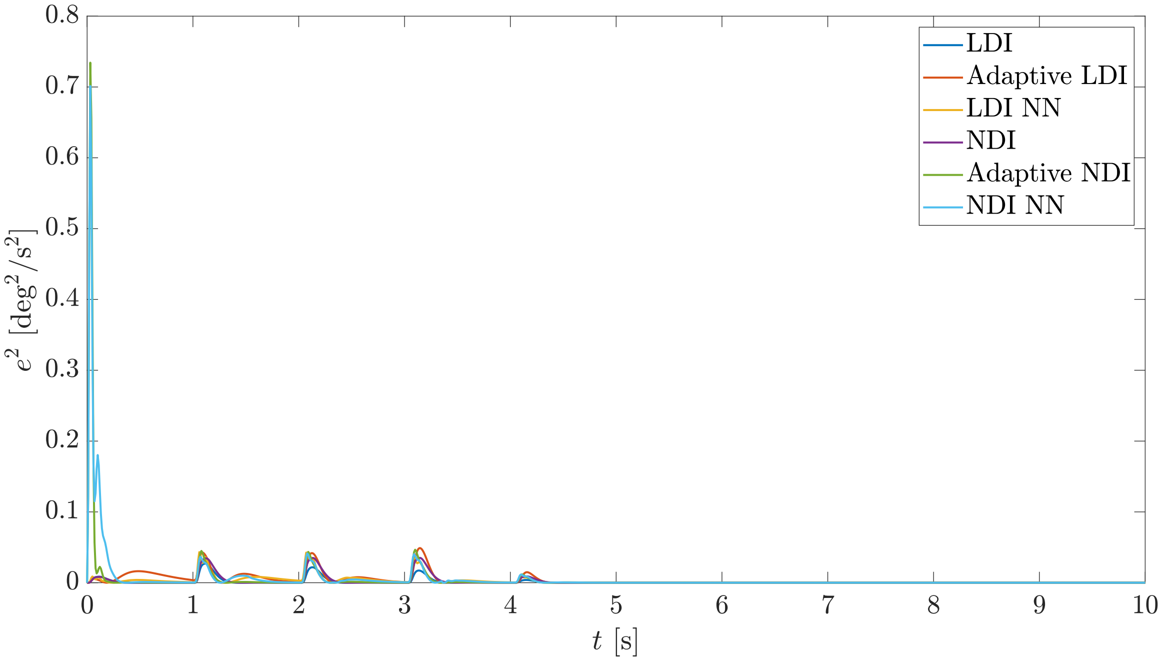


Figure 4.4: Comparison of the quadratic error for the different controllers under nominal conditions.

Regarding the control inputs, it can be observed that there is a difference between the deflections commanded by the linear controllers and the nonlinear controllers. This difference is caused because the way in which the canard deflection is included in both types of controllers is different. In the linear controllers, the canard deflection is proportional to the elevon deflection with respect to the trim position, while in the nonlinear case, it is proportional to the absolute deflection.

4.2 Actuator failures

In this section, different actuator failures will be included in the simulations in order to analyze the robustness of the controllers in the event of one of them. As it has been mentioned before, the actuator failures studied are the partial destruction of a control surface and the jam of the actuator at a certain position. The simulations are started with nominal conditions and the failures are introduced at $t_f = 1.5$ s until the end of the simulation.

4.2.1 Partial destruction of the elevons

The results obtained from the study of the controllers performance in the event of partial destruction of the elevons are shown in this section. In order to show the importance of the percentage of surface destroyed, the analysis has been performed for two cases: a 20% and 50% destruction. The pitch rate, surface deflections and e^2 for the 20% destruction case are shown in Figures 4.5, 4.7 and 4.9, while the results for the 50% case are shown in Figures 4.6, 4.8 and 4.10 respectively.

In the first place, it is important to notice the improvement in the tracking performance that the adaptive controllers provide. It can be observed that, at the time of the failure, all the non-adaptive controllers lose the reference completely and they are not able to track it anymore. However, the adaptive controllers are able to identify the new dynamics model resultant from the failure very rapidly, and they track the reference almost perfectly in roughly half a second in the worst case.

Analyzing the effects of the percentage of surface destroyed, it can be seen in the plots that the higher this percentage, the higher is the tracking error for the non-adaptive controllers. The effect in the adaptive controllers is only important at the moment of the failure since the error peak is higher for higher destruction but all the adaptive controllers recover from the failure regardless of its magnitude. It has been checked by performing a series of simulations that all the adaptive controllers can track the reference in the extreme case of 85% destruction of the elevons. From this percentage on, the control power is too low and the canards are not able to provide the pitching moment needed to track the reference because there exists saturation of the actuators.

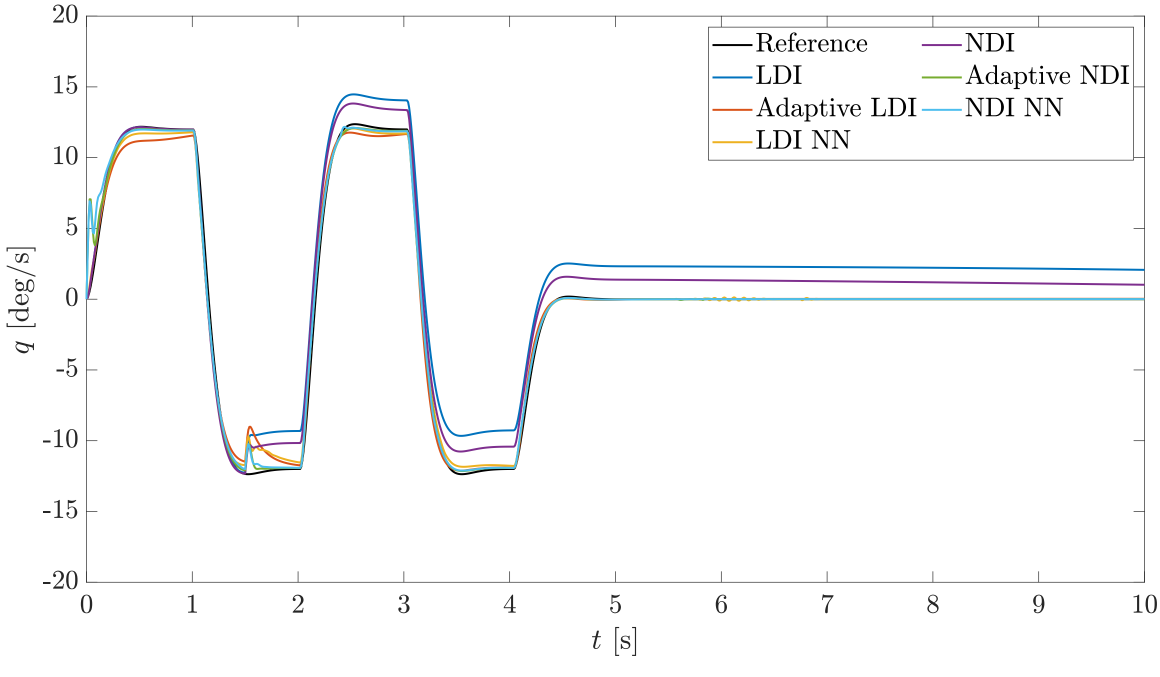


Figure 4.5: Comparison of the pitch rate for the different controllers in the event of a 20% destruction of the elevons from the instant $t_f = 1.5$ s on.

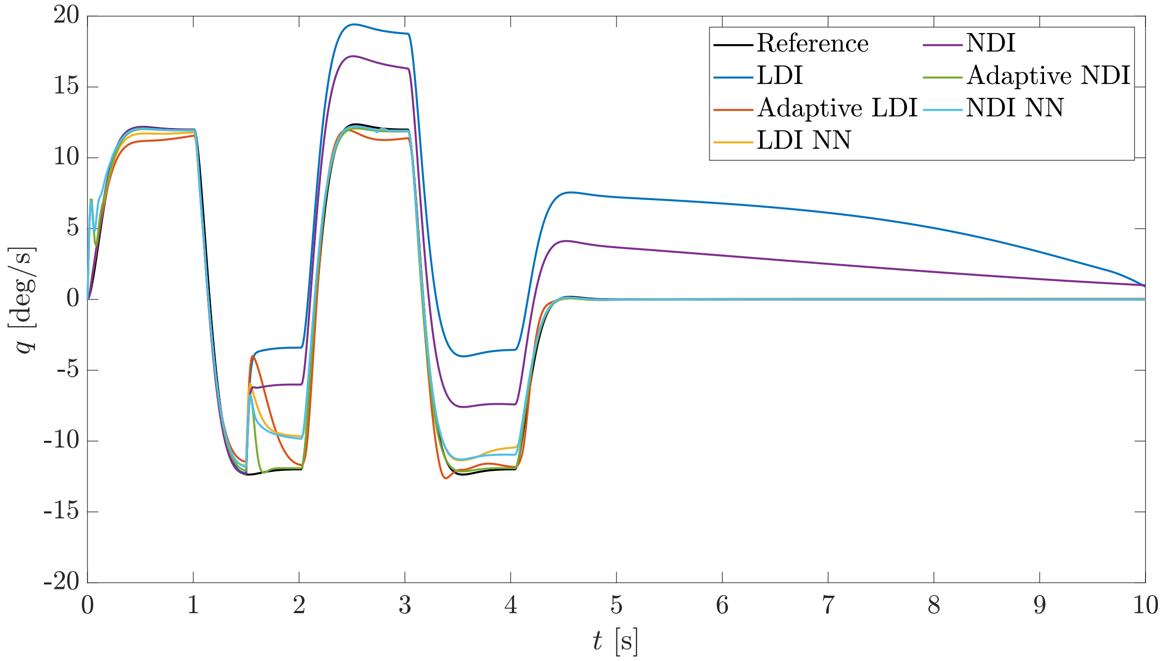


Figure 4.6: Comparison of the pitch rate for the different controllers in the event of a 50% destruction of the elevons from the instant $t_f = 1.5$ s on.

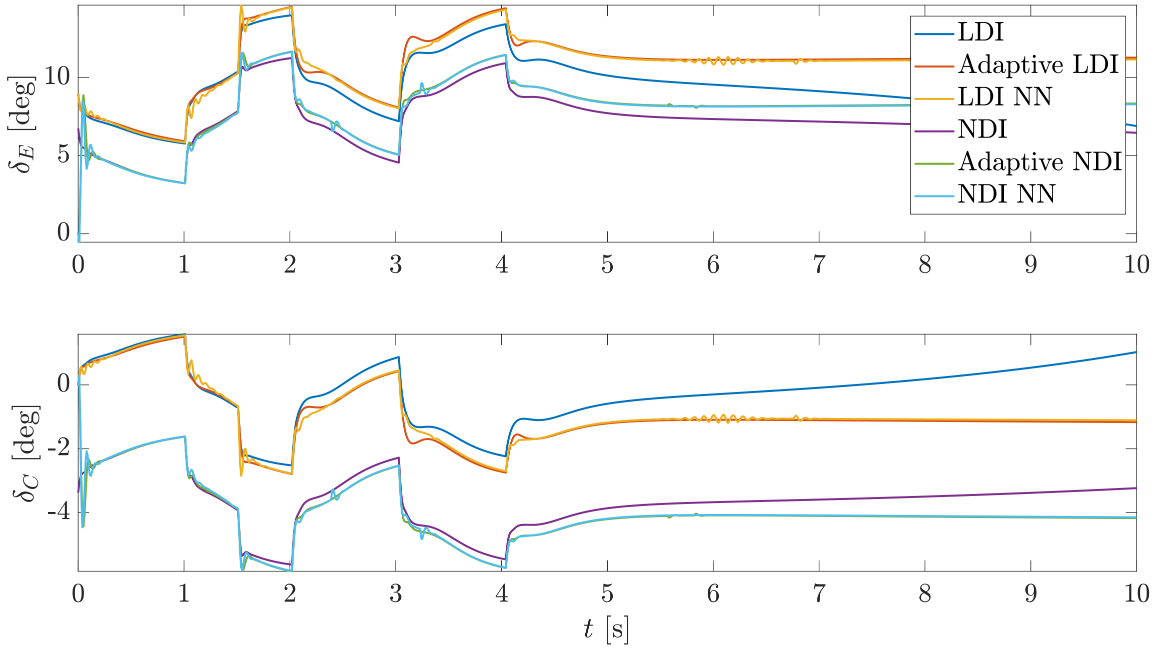


Figure 4.7: Comparison of the control surface deflections for the different controllers in the event of a 20% destruction of the elevons from the instant $t_f = 1.5$ s on.

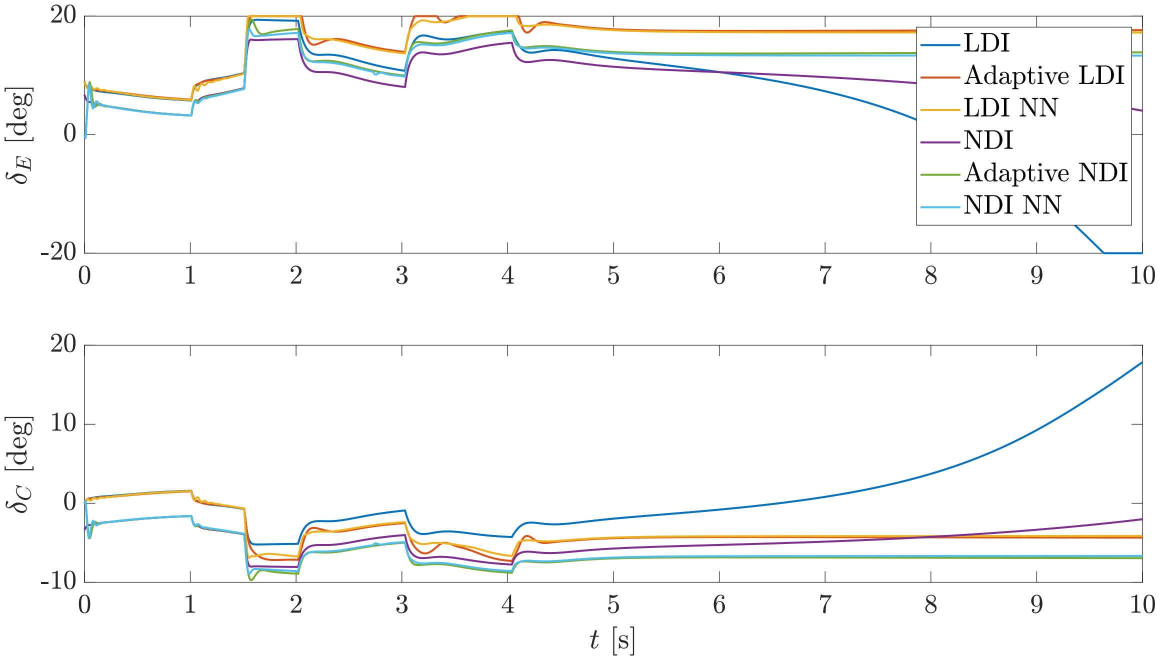


Figure 4.8: Comparison of the control surface deflections for the different controllers in the event of a 50% destruction of the elevons from the instant $t_f = 1.5$ s on.

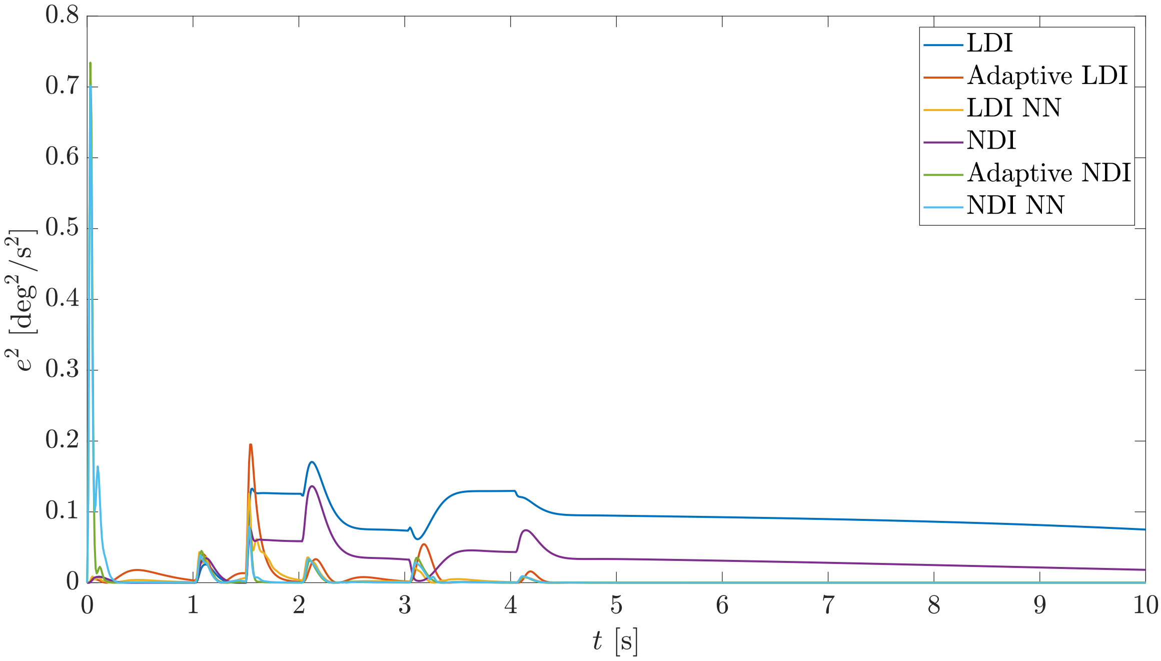


Figure 4.9: Comparison of the quadratic error for the different controllers in the event of a 20% destruction of the elevons from the instant $t_f = 1.5$ s on.

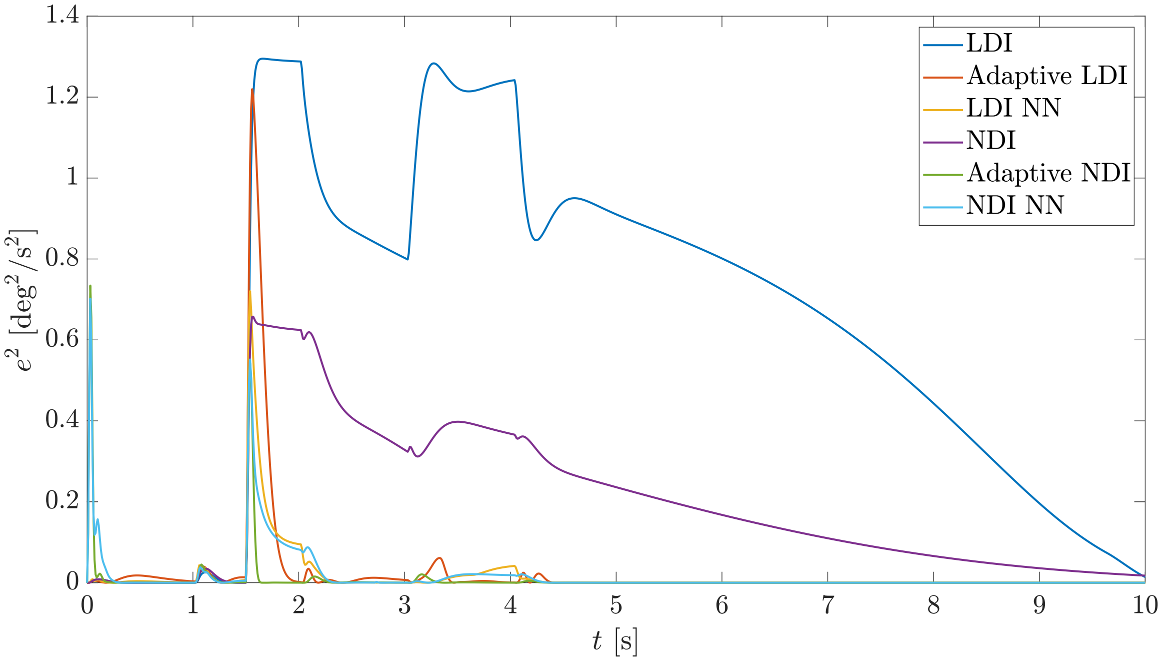


Figure 4.10: Comparison of the quadratic error for the different controllers in the event of a 50% destruction of the elevons from the instant $t_f = 1.5$ s on.

Finally, it can be seen that the controller that presents the best performance in this study is the adaptive nonlinear dynamic inversion controller, since it is the fastest in recovering from the failure and the one that tracks the reference more accurately.

4.2.2 Jam of the elevons

The results obtained from the study of the controllers performance in the event of actuator jam are shown in this section. Two different cases are studied: jam at a position of 5 and 15 degrees. The pitch rate, surface deflections and e^2 for the 5 degrees jam are shown in Figures 4.11, 4.13 and 4.15, while the results for the 15 degrees jam are shown in Figures 4.12, 4.14 and 4.16 respectively.

It can be observed that all the adaptive controllers present a satisfactory performance since they are able to track the reference, while the non-adaptive controllers are not able to follow the desired pitch rate correctly. The quadratic error converges to zero for all the adaptive controllers regardless of the jam position. However, in the non-adaptive case, e^2 even diverges for the nonlinear controller when the jam is at 15 degrees. These results are another proof of robustness and reliability of the adaptive controllers.

Regarding the control surface deflections, it can be seen that all the control power relies on the canards once the elevon is jammed. Therefore, the system would not be controllable in the event of elevon jam with the absence of the canards in the control algorithm.

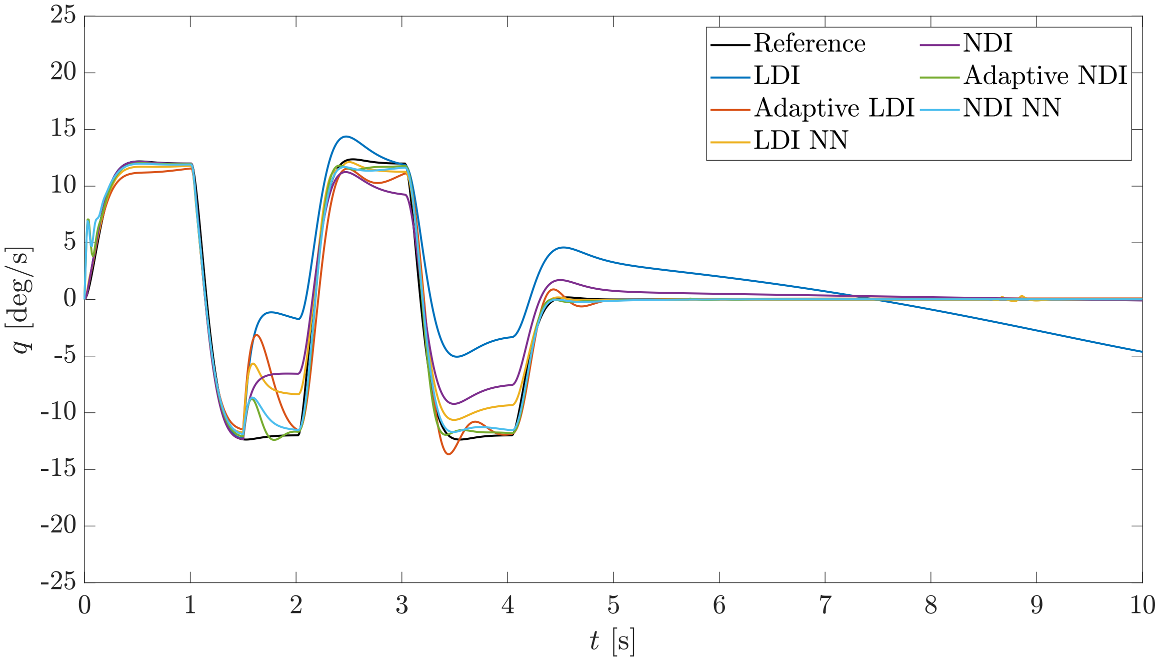


Figure 4.11: Comparison of the pitch rate for the different controllers in the event of elevon jam at 5 degrees from the instant $t_f = 1.5$ s on.

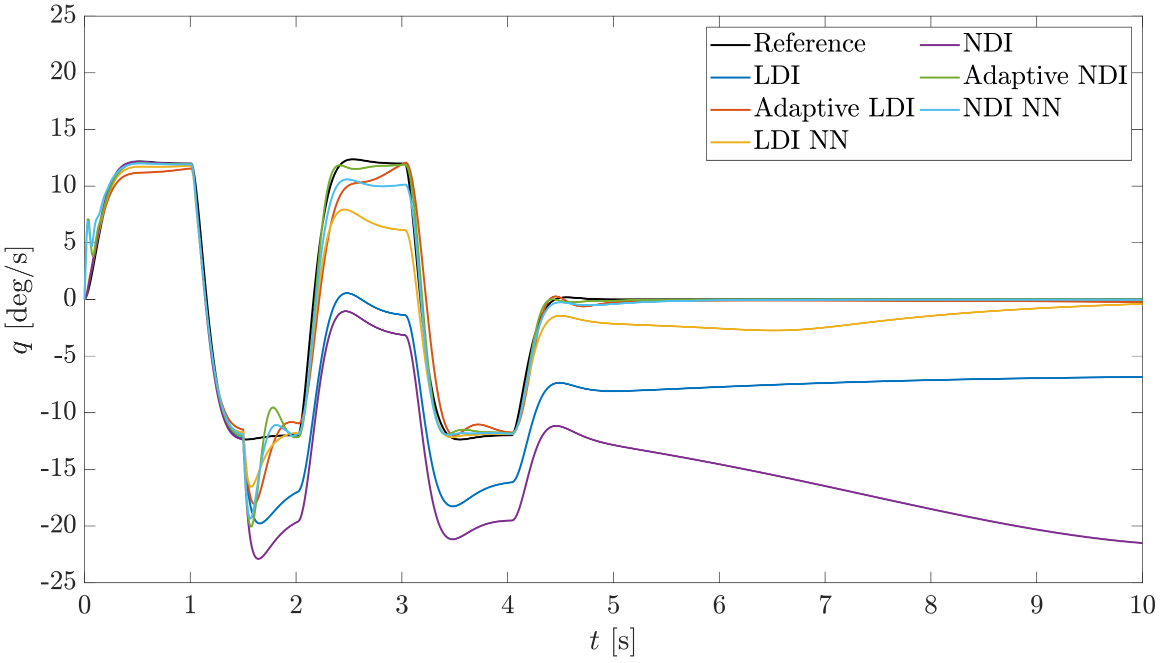


Figure 4.12: Comparison of the pitch rate for the different controllers in the event of elevon jam at 15 degrees from the instant $t_f = 1.5$ s on.

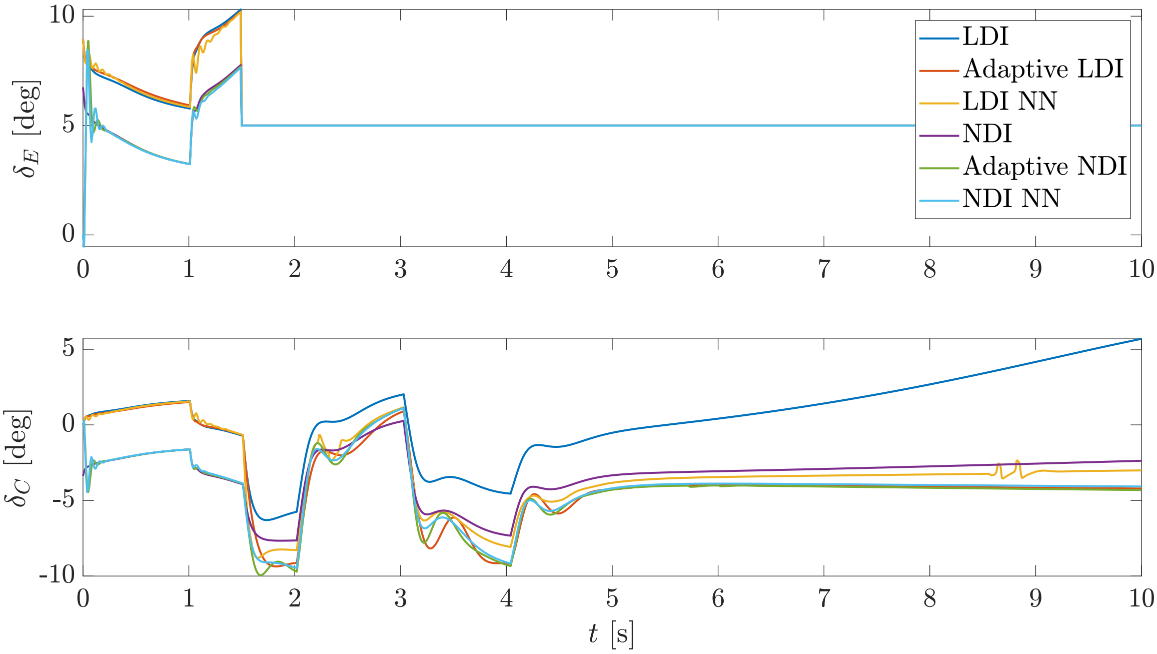


Figure 4.13: Comparison of the control surface deflections for the different controllers in the event of elevon jam at 5 degrees from the instant $t_f = 1.5$ s on.

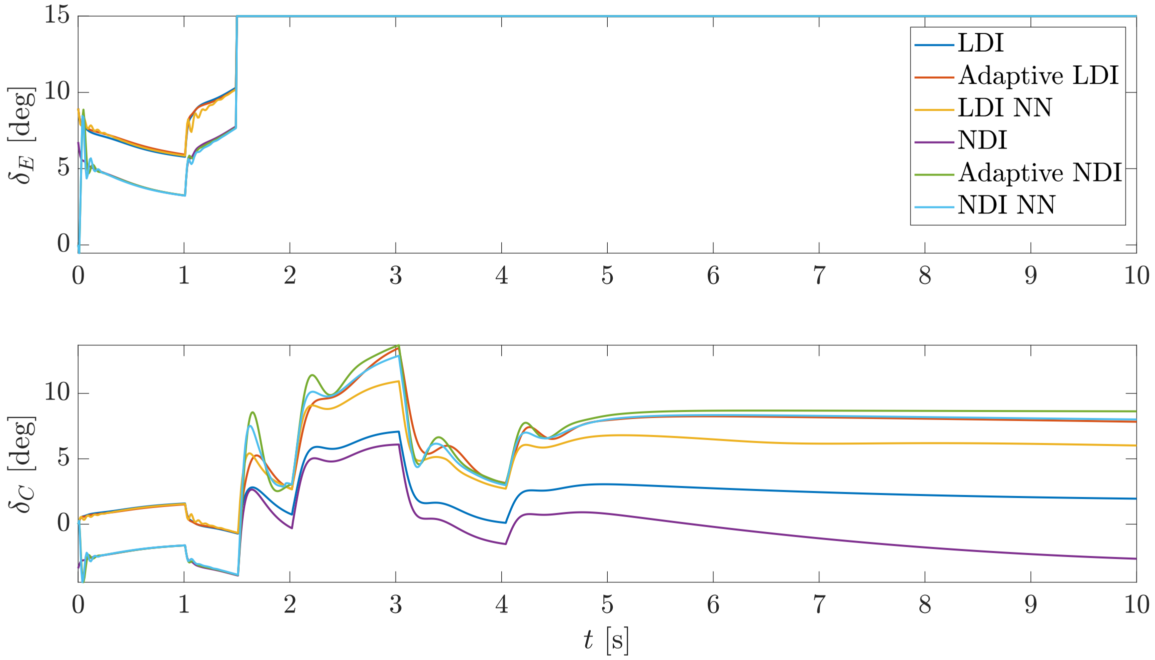


Figure 4.14: Comparison of the control surface deflections for the different controllers in the event of elevon jam at 15 degrees from the instant $t_f = 1.5$ s on.

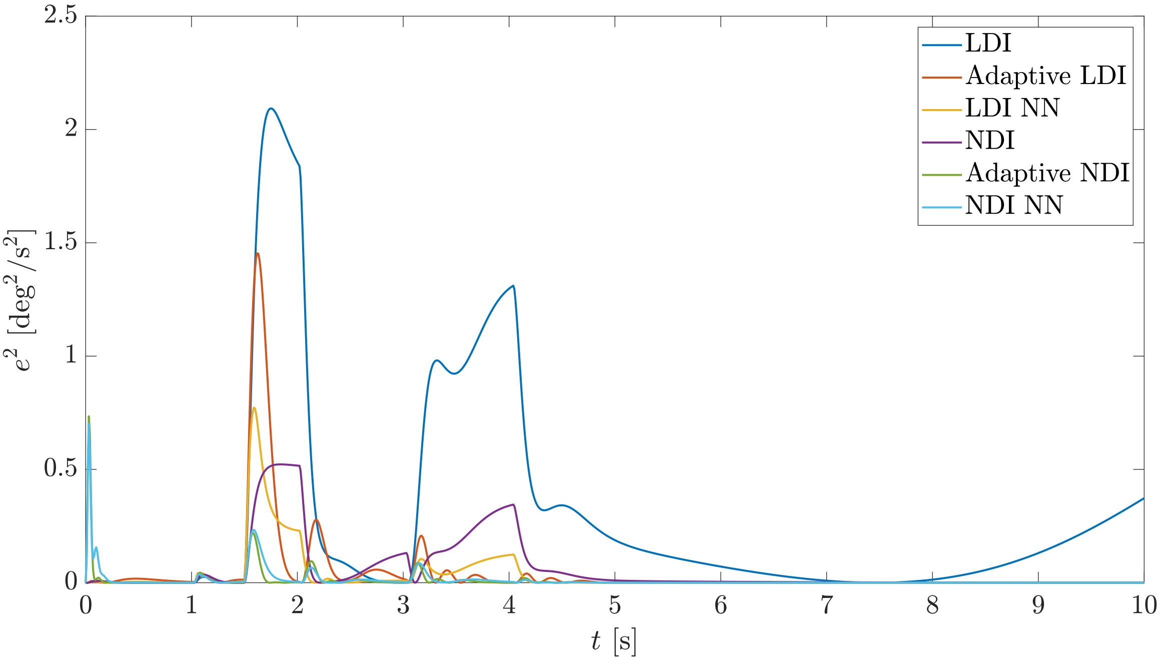


Figure 4.15: Comparison of the quadratic error for the different controllers in the event of elevon jam at 5 degrees from the instant $t_f = 1.5$ s on.

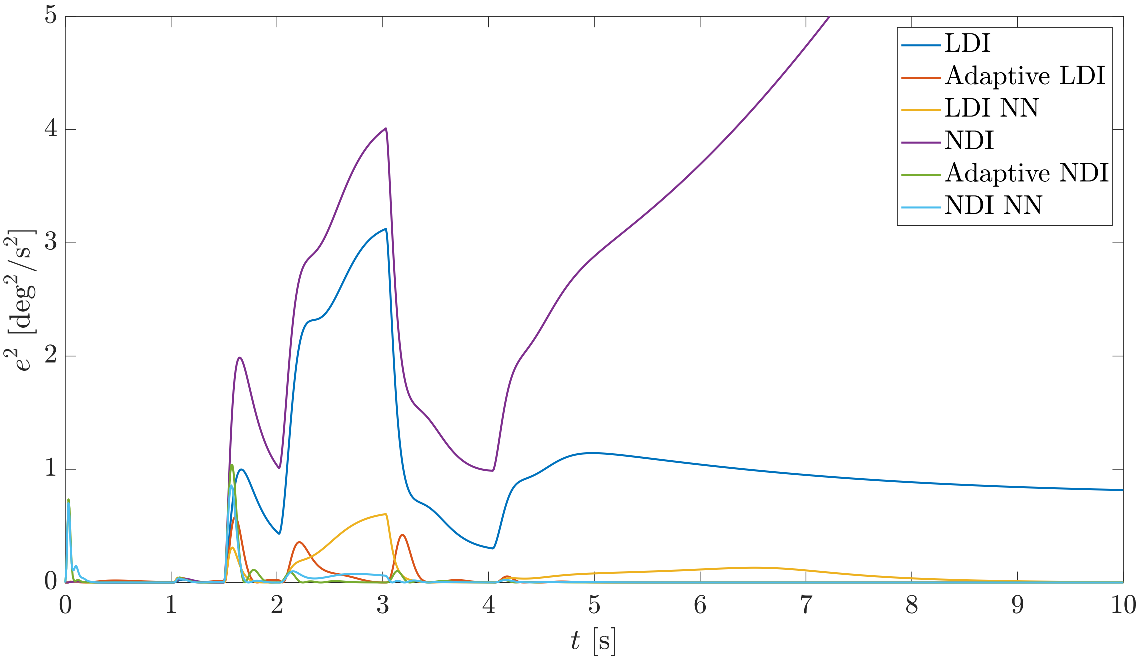


Figure 4.16: Comparison of the quadratic error for the different controllers in the event of elevon jam at 15 degrees from the instant $t_f = 1.5$ s on.

Analyzing the results obtained with the adaptive controllers, it can be stated that the one that presents the best performance is the adaptive nonlinear dynamic inversion controller because it shows the shortest recovery time after the failure and the best tracking performance.

4.3 Sensor failures

In this section, the results obtained from the simulations including sensor failures are shown and analyzed. Two different sensor failures have been studied: white Gaussian noise in the pitch rate measurements and a constant step bias. It will be shown that the presence of noise is the most detrimental phenomenon for the performance of the dynamic inversion controllers.

4.3.1 Measurement noise

In order to analyze the effects of the presence of measurement noise in the performance of the controllers, two different cases have been studied: noise in the pitch rate measurements with standard deviations σ_s of 1 deg/s and 5 deg/s. The pitch rate, surface deflections and e^2 for the $\sigma_s = 1$ deg/s case are shown in Figures 4.17, 4.19 and 4.21, while the results for the $\sigma_s = 5$ deg/s case are shown in Figures 4.18, 4.20 and 4.22 respectively.

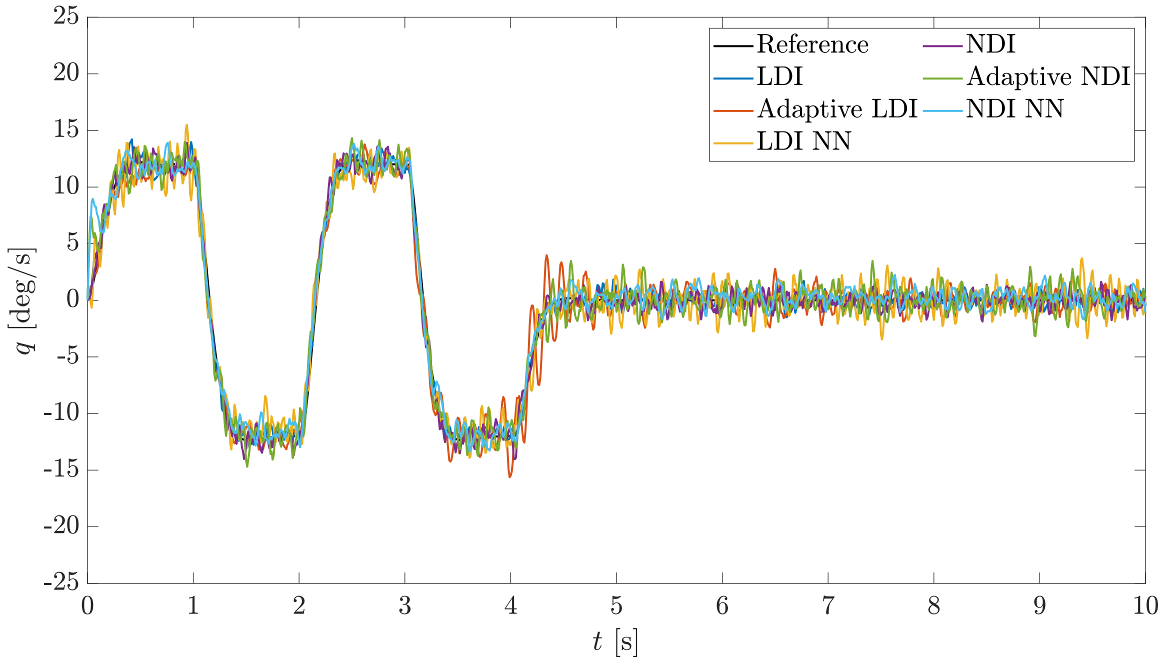


Figure 4.17: Comparison of the pitch rate for the different controllers in the presence of noise in the pitch rate measurements with a standard deviation of 1 deg/s.

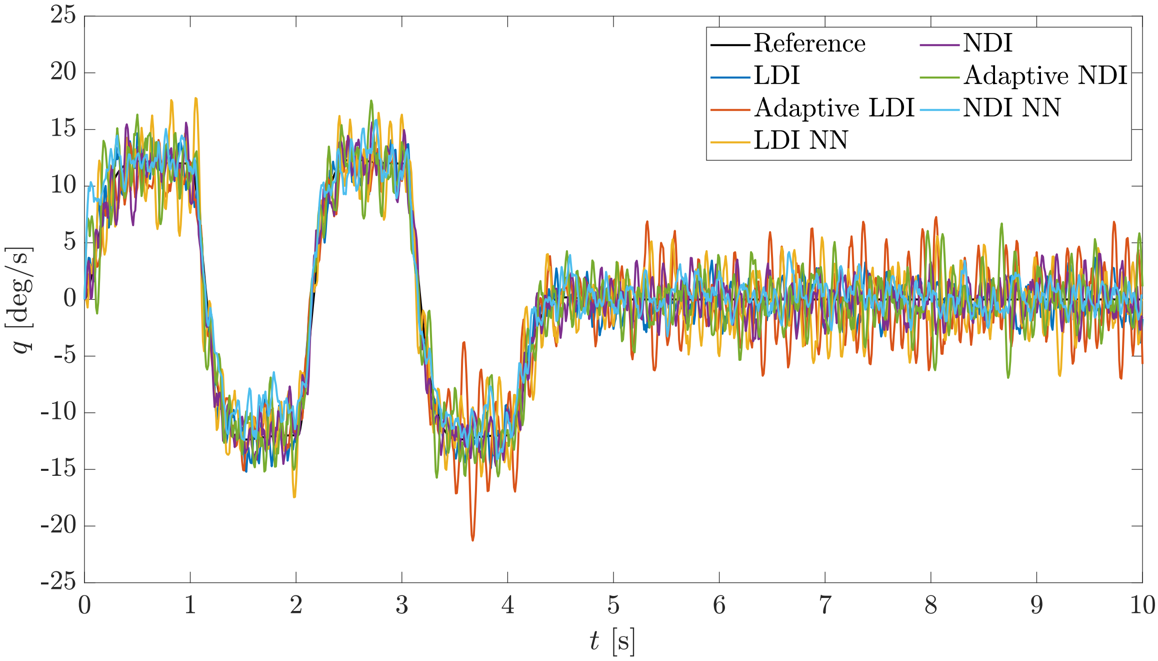


Figure 4.18: Comparison of the pitch rate for the different controllers in the presence of noise in the pitch rate measurements with a standard deviation of 5 deg/s.

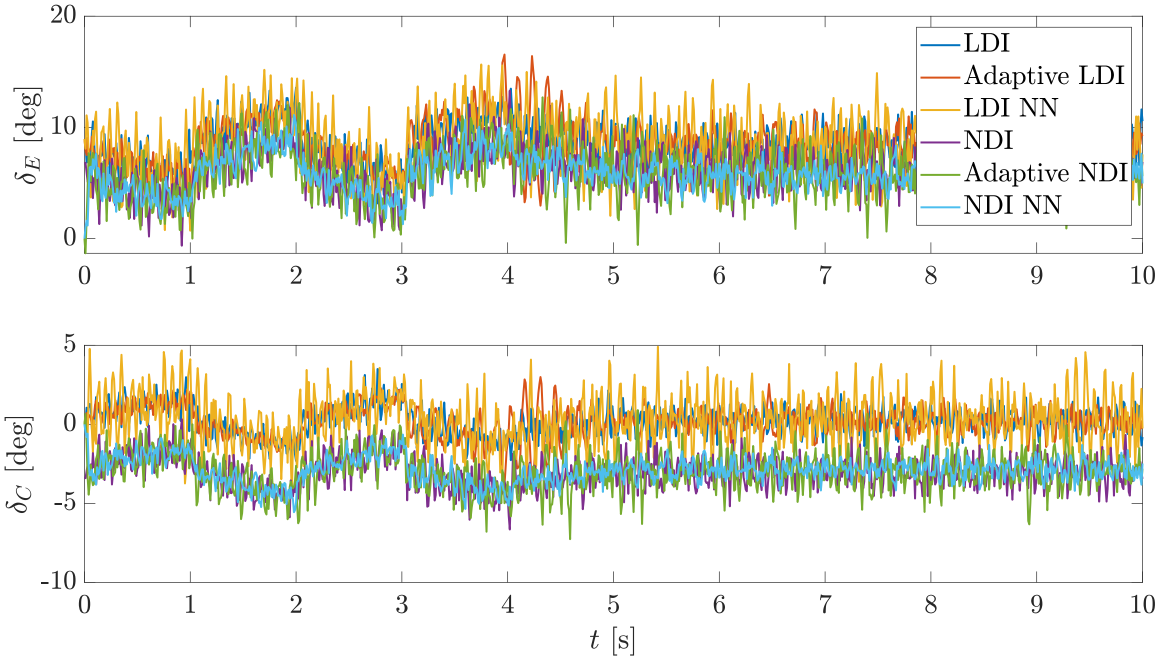


Figure 4.19: Comparison of the control surface deflections for the different controllers in the presence of noise in the pitch rate measurements with a standard deviation of 1 deg/s.

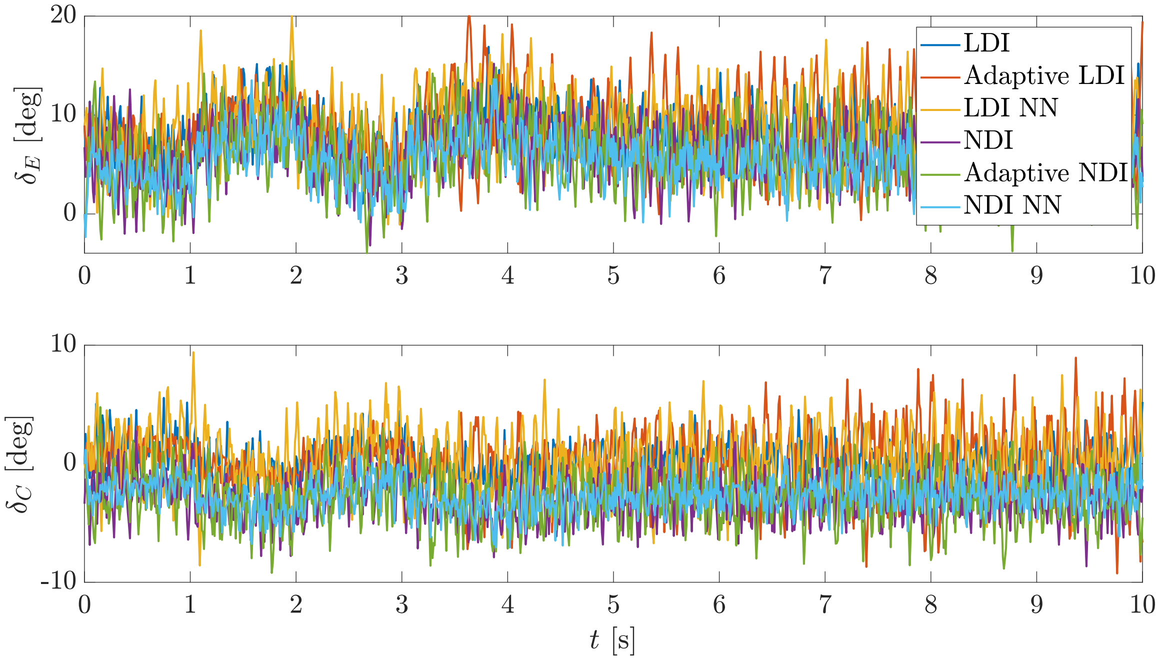


Figure 4.20: Comparison of the control surface deflections for the different controllers in the presence of noise in the pitch rate measurements with a standard deviation of 5 deg/s.

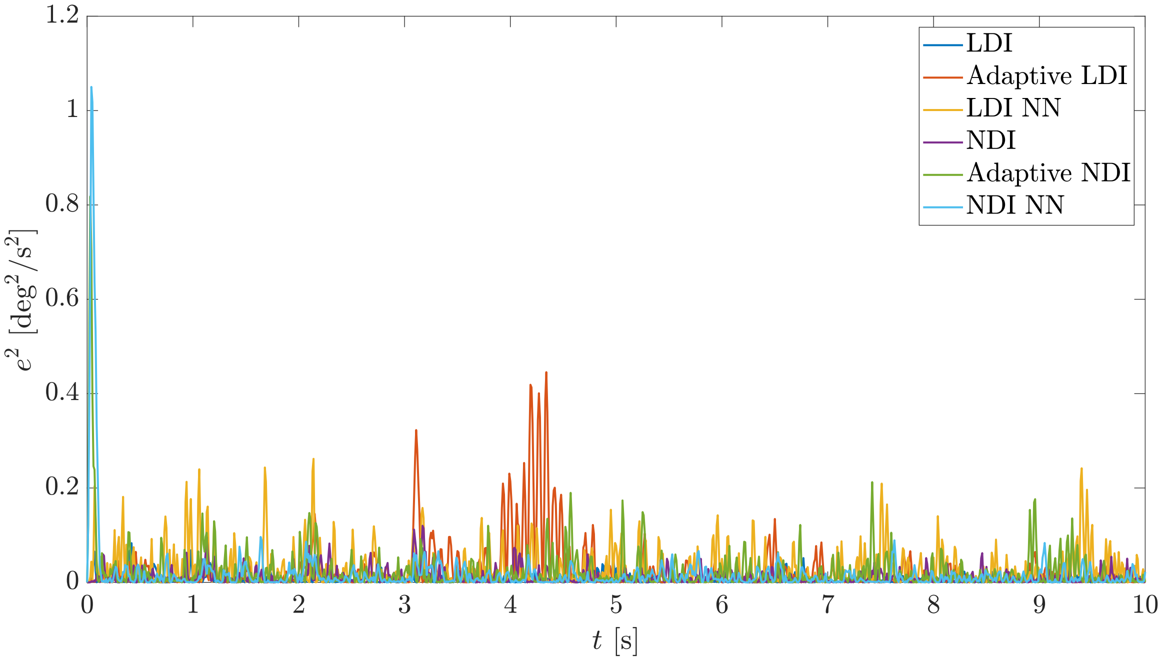


Figure 4.21: Comparison of the quadratic error for the different controllers in the presence of noise in the pitch rate measurements with a standard deviation of 1 deg/s.

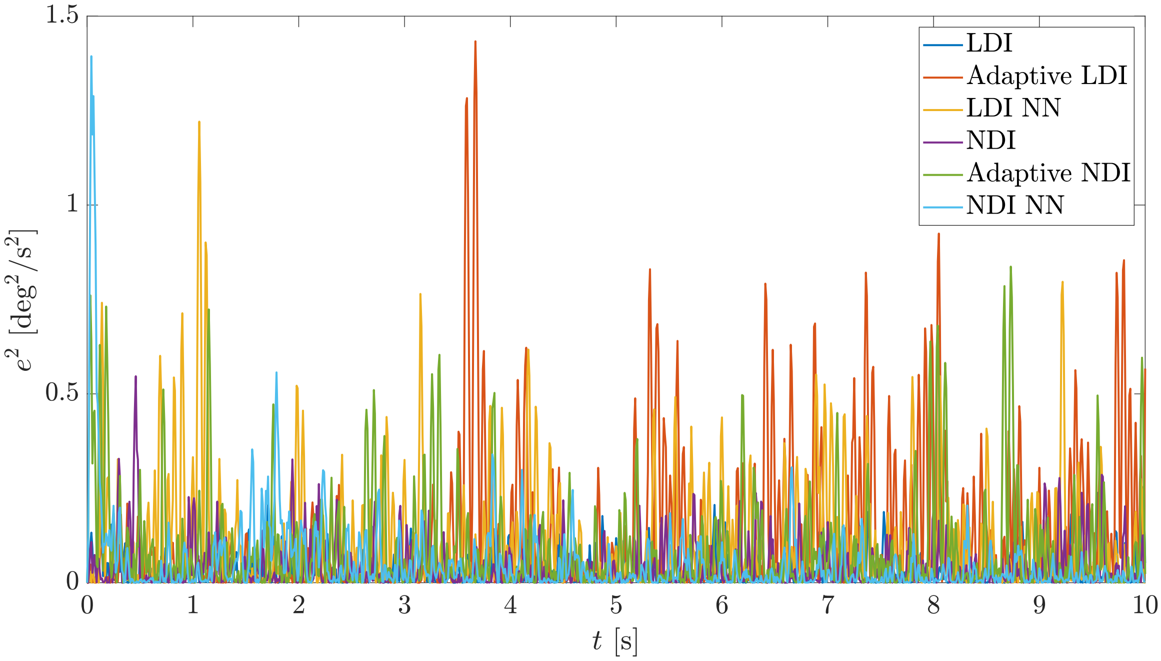


Figure 4.22: Comparison of the quadratic error for the different controllers in the presence of noise in the pitch rate measurements with a standard deviation of 5 deg/s.

As it was advanced, the performance of the dynamic inversion controllers is seriously deteriorated when there exists noise in the measurements. It can be observed that, although the controllers can track the shape of the reference, there are high frequency oscillations in the response of the aircraft. The amplitude of these oscillations is higher for higher σ_s. Moreover, it can be noticed that the adaptive controllers work worse than the non-adaptive ones in the presence of noise. This effect can be caused by the difficulty in estimating the parameters of the system when there is noise in the measurements.

Finally, analyzing the control surface deflections, it can be observed that the actuators must be able to respond very quickly to the commands of the dynamic inversion algorithm in order to maintain the aircraft stability. However, the real actuators present certain dynamics that are not modeled in this work and therefore, it is possible that the satisfactory implementation of these controllers into the subscale model of the GFF encounters some difficulties if the noise of the sensors is too high.

4.3.2 Step bias

The effects of the presence of a constant step bias in the pitch rate measurements are shown and analyzed in this section. Two different cases are studied: a small bias of 2.5 deg/s and a large bias of 5 deg/s. The pitch rate, surface deflections and e^2 for the small bias case are shown in Figures 4.23, 4.25 and 4.27, while the results for the large bias case are shown in Figures 4.24, 4.26 and 4.28 respectively.

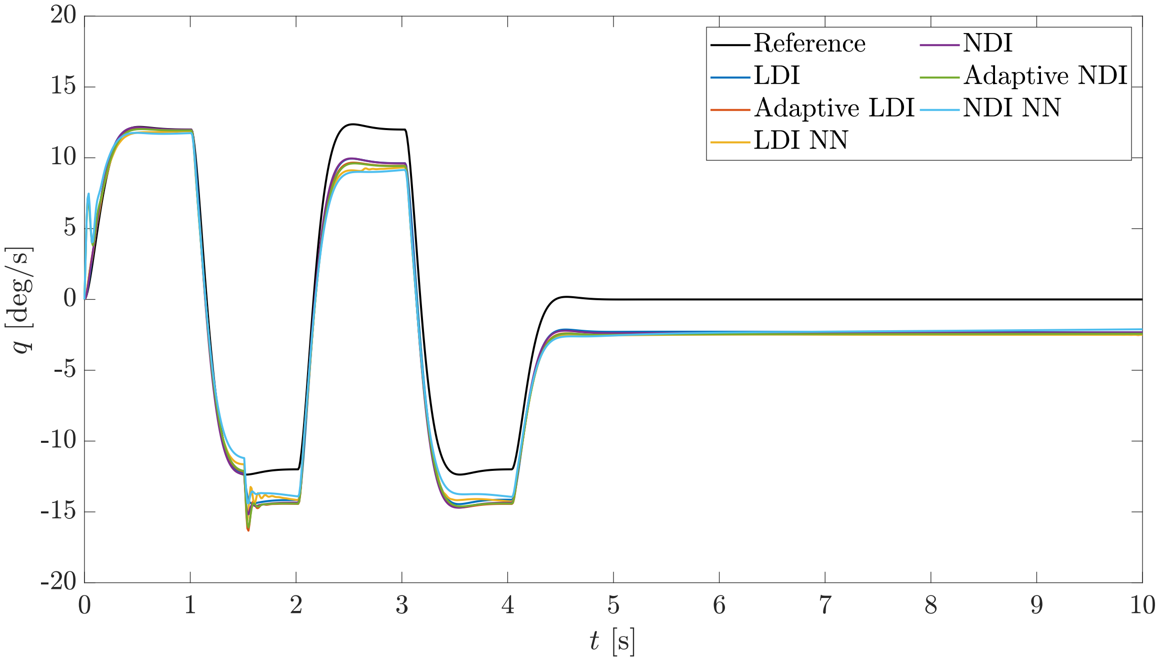


Figure 4.23: Comparison of the pitch rate for the different controllers in the presence of a step bias of 2.5 deg/s in the pitch rate measurements from the instant $t_f = 1.5$ s on.

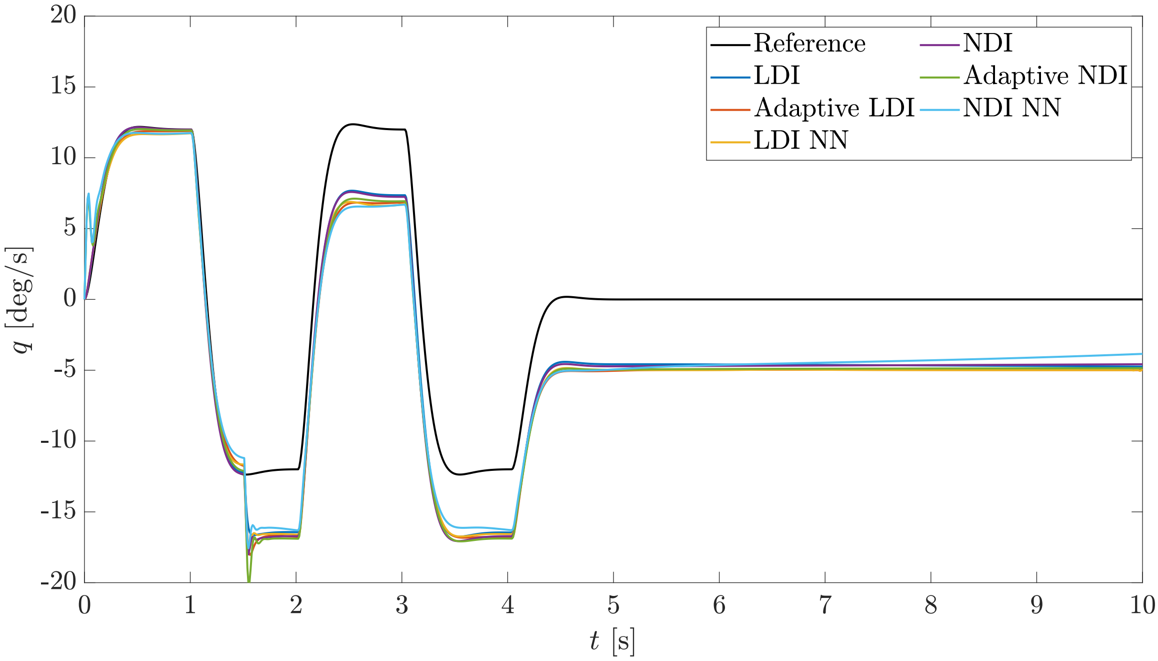


Figure 4.24: Comparison of the pitch rate for the different controllers in the presence of a step bias of 5 deg/s in the pitch rate measurements from the instant $t_f = 1.5$ s on.

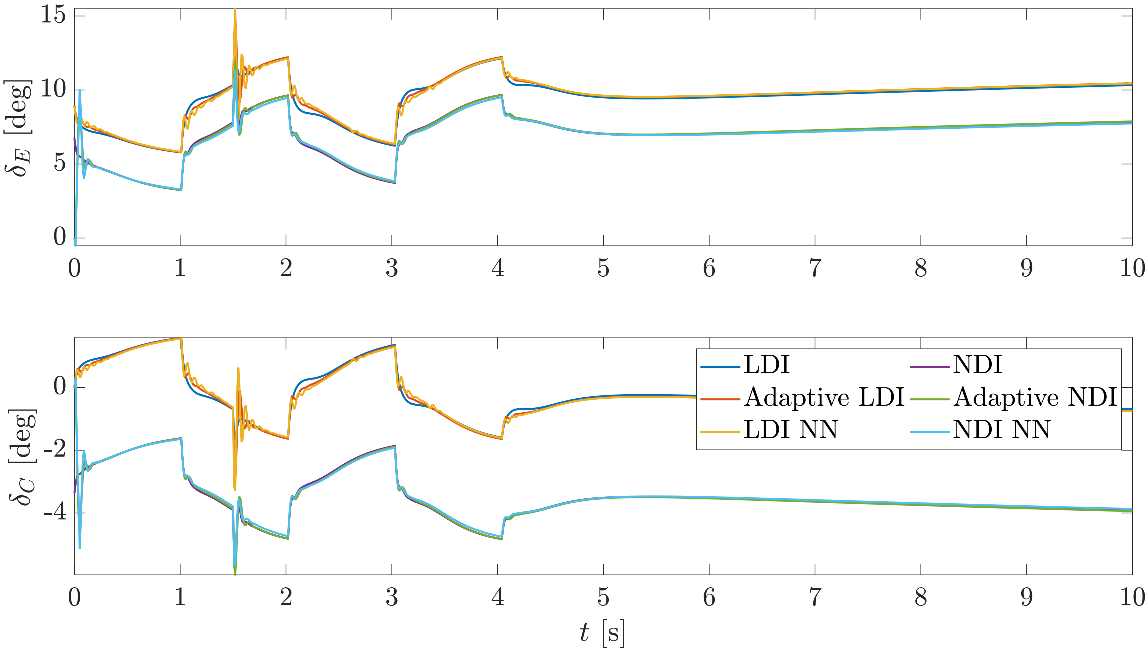


Figure 4.25: Control surface deflections for the different controllers in the presence of a step bias of 2.5 deg/s in the pitch rate measurements from the instant $t_f = 1.5$ s on.

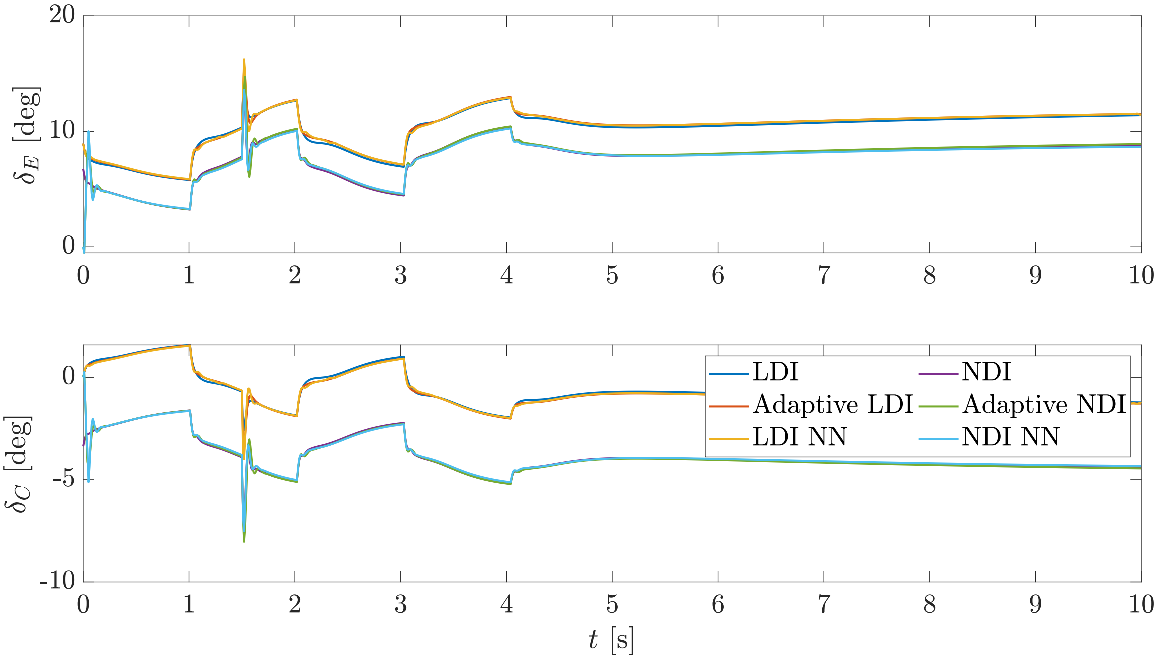


Figure 4.26: Control surface deflections for the different controllers in the presence of a step bias of 5 deg/s in the pitch rate measurements from the instant $t_f = 1.5$ s on.

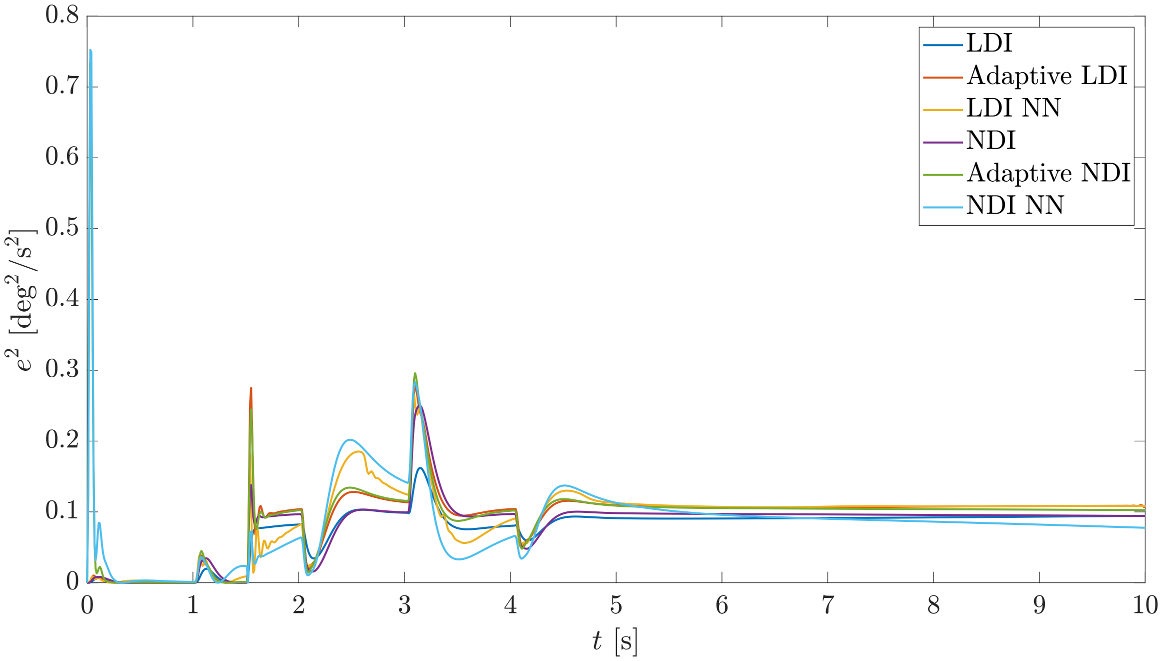


Figure 4.27: Comparison of the quadratic error for the different controllers in the presence of a step bias of 2.5 deg/s in the pitch rate measurements from the instant $t_f = 1.5$ s on.

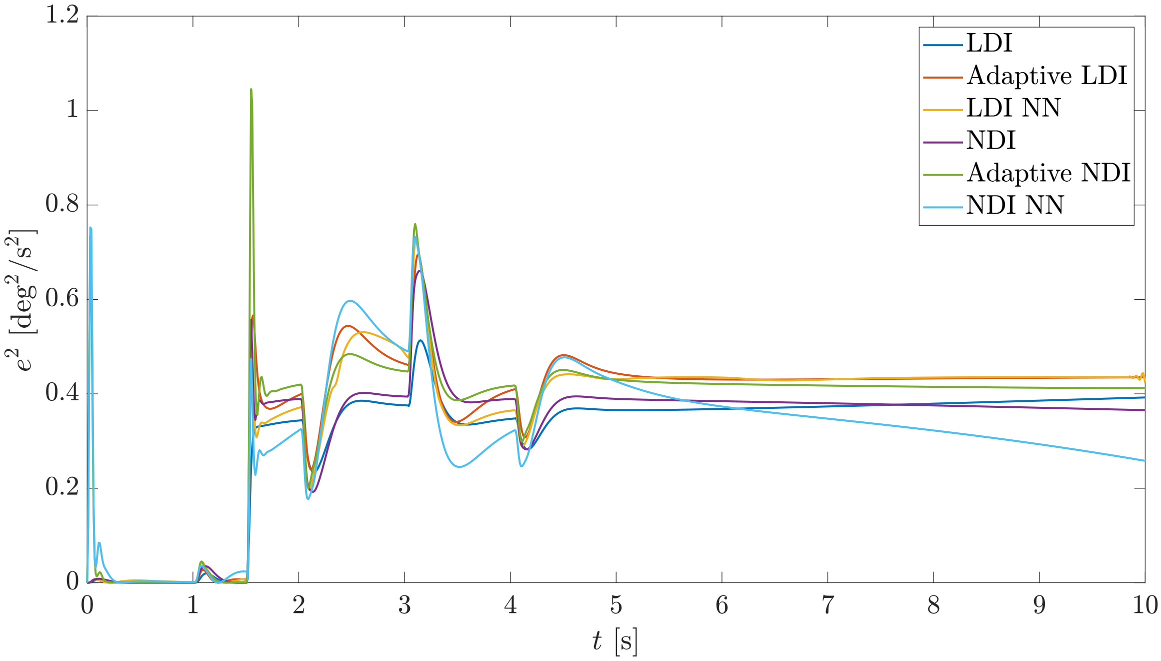


Figure 4.28: Comparison of the quadratic error for the different controllers in the presence of a step bias of 5 deg/s in the pitch rate measurements from the instant $t_f = 1.5$ s on.

Regarding the response of the aircraft, it can be observed that none of the control systems can track the reference after the introduction of the failure. Instead, the pitch rate presents a constant error with respect to the reference. This phenomenon occurs because the control algorithm receives pitch rate measurements that match the reference although in reality there exists a difference equal to the bias. In consequence, the tracking error will be directly proportional to the magnitude of the bias.

It could be straightforward to think that the presence of bias is a serious problem for the control of the aircraft since it cannot be compensated with any of the controllers. However, it is relatively easy to calibrate the sensors in order to eliminate this issue.

4.4 Unstable configuration

Since the type of aircraft studied in this thesis is normally statically unstable, the analysis of the controllers performance for different degrees of instability is critical. To perform this study, a parameter called static margin SM is defined as:

$$SM = \frac{x_{NP} - x_{cog}}{c_w} \tag{4.1}$$

where x_{NP} is the position of the neutral point of the aircraft. From this definition it is obvious that a positive static margin implies a stable configuration while negative values mean that the aircraft is unstable. The static margin of the GFF subscale model for nominal conditions has been estimated to be $SM = 14$ %.

In order to analyze the performance of the controllers when the aircraft presents an unstable configuration, two different cases are studied: $SM = -5$ % and $SM = -30$ %. The pitch rate, surface deflections and e^2 for the $SM = -5$ % case are shown in Figures 4.29, 4.31 and 4.33, while the results for the $SM = -30$ % case are shown in Figures 4.30, 4.32 and 4.34 respectively.

It can be observed that all the controllers work satisfactorily regardless of the level of instability. Since the dynamic inversion algorithm integrates the dynamics of the aircraft to compute the control input that is necessary to get the desired response, it is able to work correctly for almost every configuration as long as the model of the dynamics implemented into the algorithm is accurate enough. In the case of the adaptive controllers, it can be observed that there is a peak in the tracking error at the beginning of the simulation caused by the initial conditions of the estimated parameters. However, the algorithm rapidly estimates the parameters of the system and the controller is able to track the reference with a high accuracy.

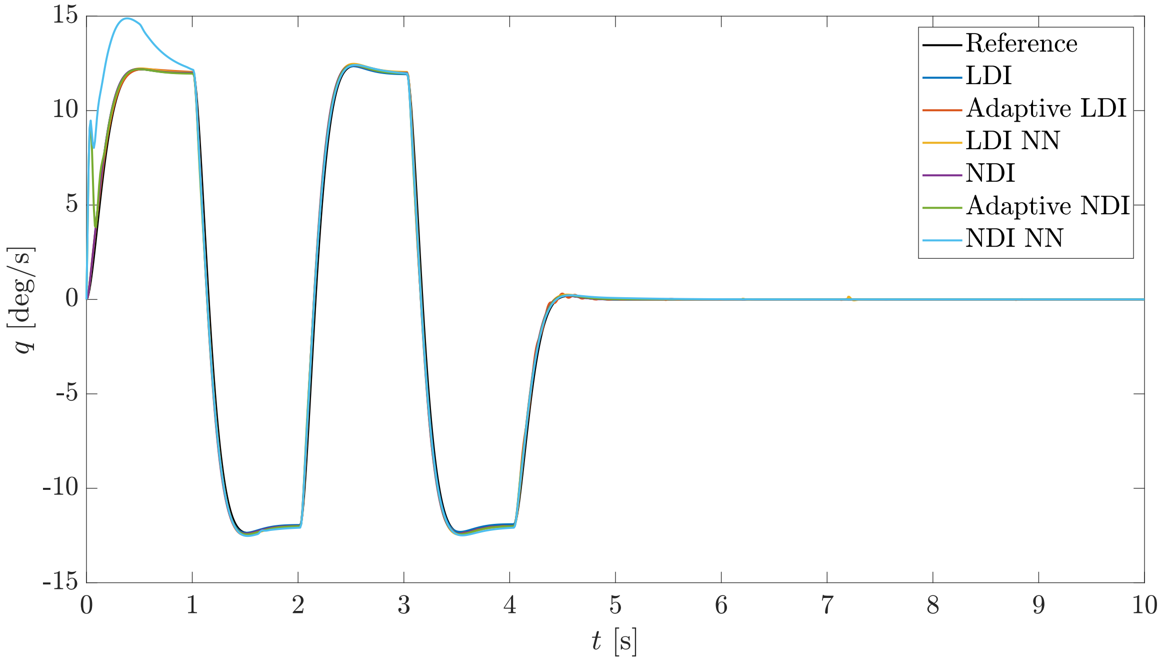


Figure 4.29: Comparison of the pitch rate for the different controllers when the aircraft has an unstable configuration with $SM = -5\%$.

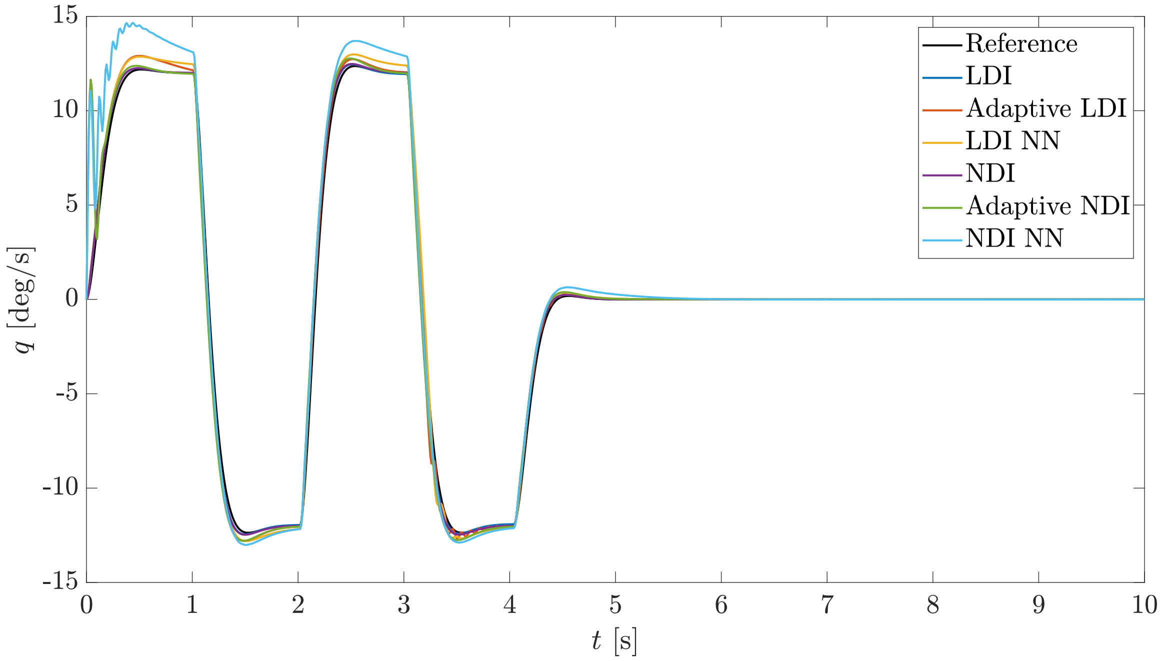


Figure 4.30: Comparison of the pitch rate for the different controllers when the aircraft has an unstable configuration with $SM = -30\%$.

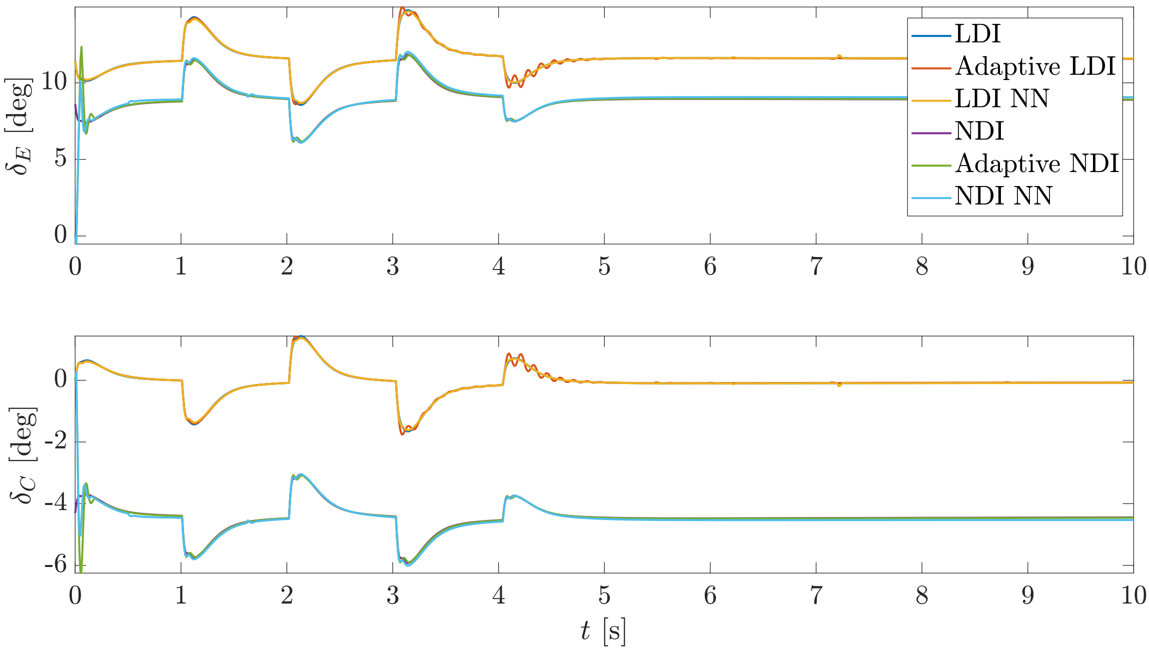


Figure 4.31: Comparison of the control surface deflections for the different controllers when the aircraft has an unstable configuration with $SM = -5\%$.

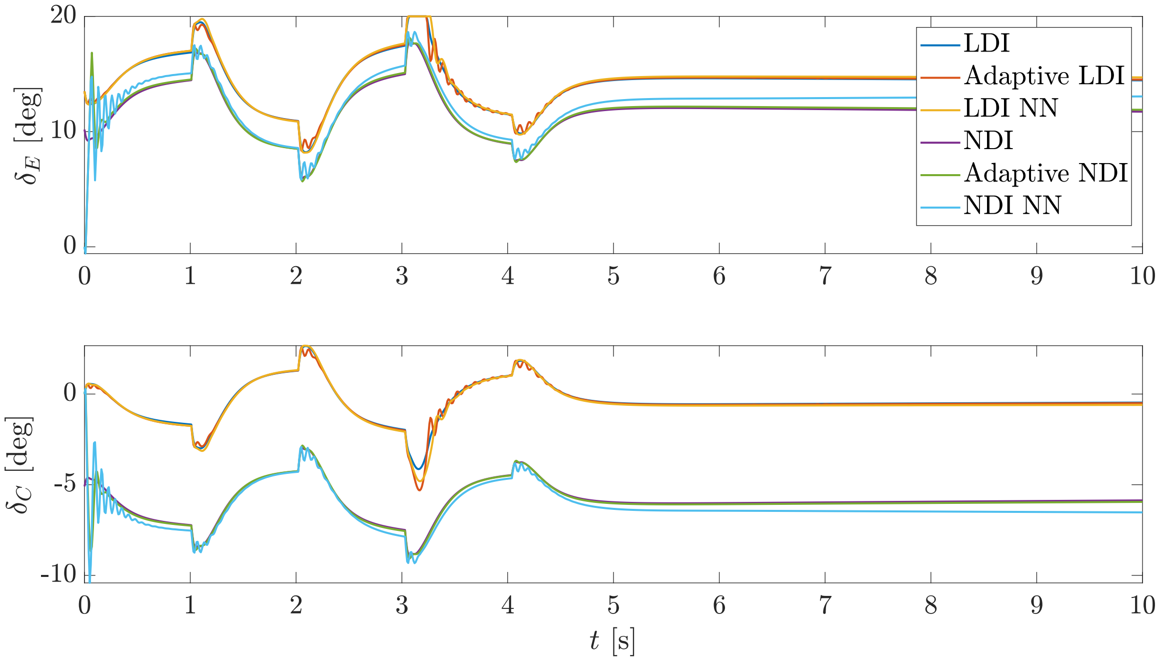


Figure 4.32: Comparison of the control surface deflections for the different controllers when the aircraft has an unstable configuration with $SM = -30\%$.

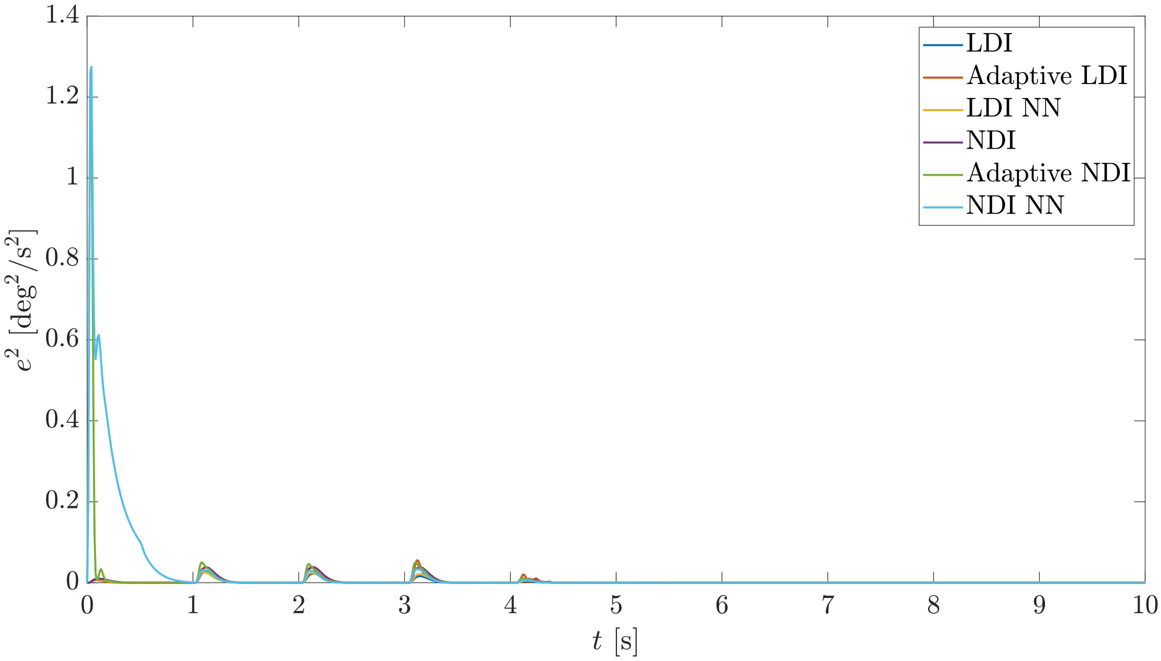


Figure 4.33: Comparison of the quadratic error for the different controllers when the aircraft has an unstable configuration with $SM = -5\%$.

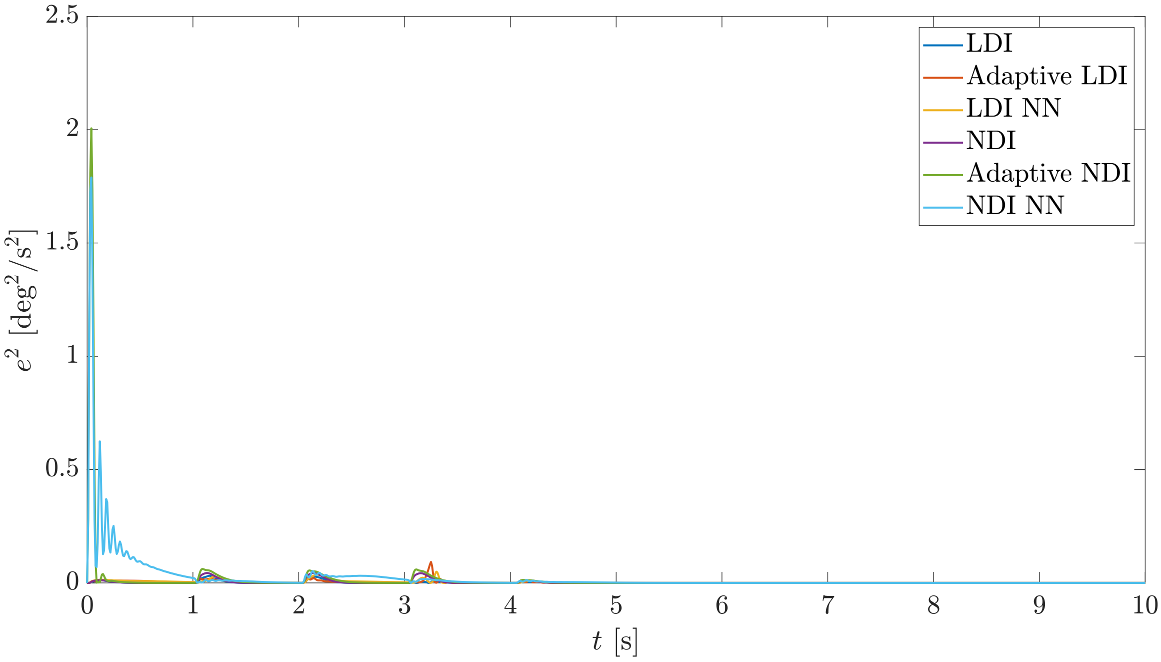


Figure 4.34: Comparison of the quadratic error for the different controllers when the aircraft has an unstable configuration with $SM = -30\%$.

4.5 Modeling error

The study of the controllers performance in the presence of errors or uncertainties in the dynamics models is of great importance. This model errors could exist due to an inaccurate estimation of the parameters that define the aircraft dynamics, such as the mass, inertia and aerodynamic coefficients. Also, in the linear models, this mismatch between the available model and the actual dynamics could be produced because the aircraft is operating far from the trim conditions for which the model was obtained.

In order to perform the analysis of the modeling error effects, the parameters that define the aircraft dynamics have been modified by scaling them by a random factor with a established maximum uncertainty Δ_e. Two different cases are studied: maximum uncertainty of 50 % and 90 %. The pitch rate, surface deflections and e^2 for the 50 % maximum uncertainty case are shown in Figures 4.35, 4.37 and 4.39, while the results for the 90 % maximum uncertainty case are shown in Figures 4.36, 4.38 and 4.40 respectively.

It can be observed that the performance of the non-adaptive controllers is significantly deteriorated in the presence of modeling errors. The presence of a maximum uncertainty of 50 % produces an excessive tracking error during the performance of the maneuver. On the other hand, apart from the higher tracking error, the maximum uncertainty of 90 % introduces a series of oscillations in the response for the non-adaptive linear controller. Regarding the performance of the adaptive controllers, it can be seen that all of them are able to follow the reference with a high accuracy regardless of the uncertainty in the model.

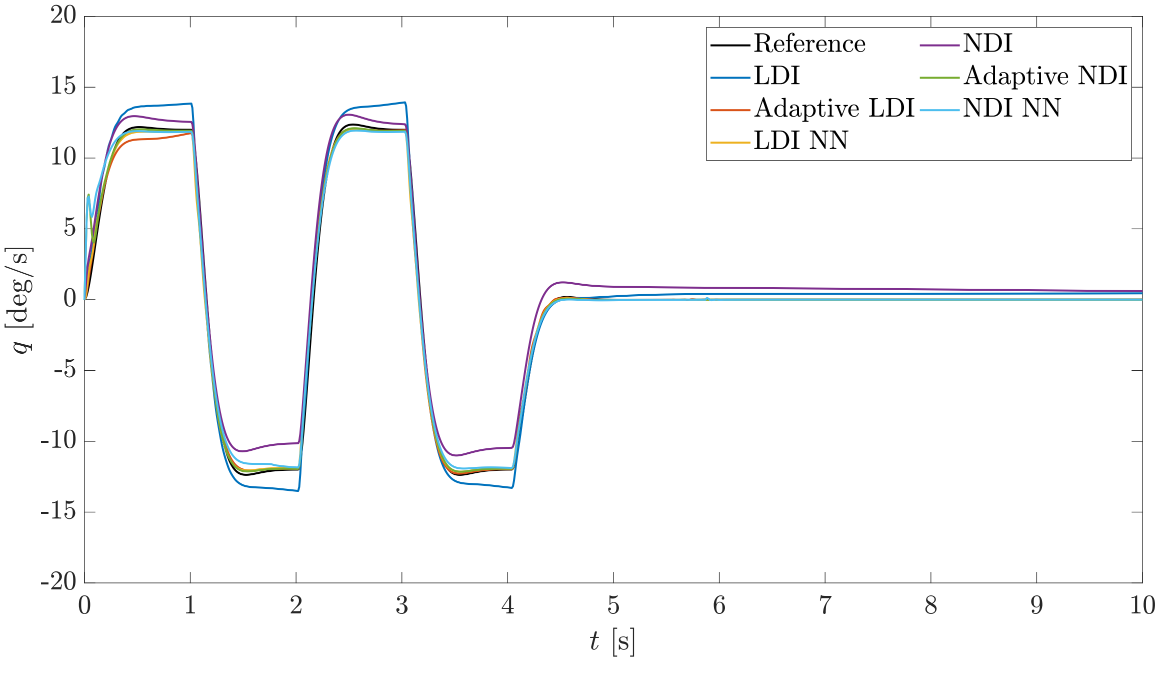


Figure 4.35: Comparison of the pitch rate for the different controllers when there is a maximum relative error of 50% in the coefficients that define the model.

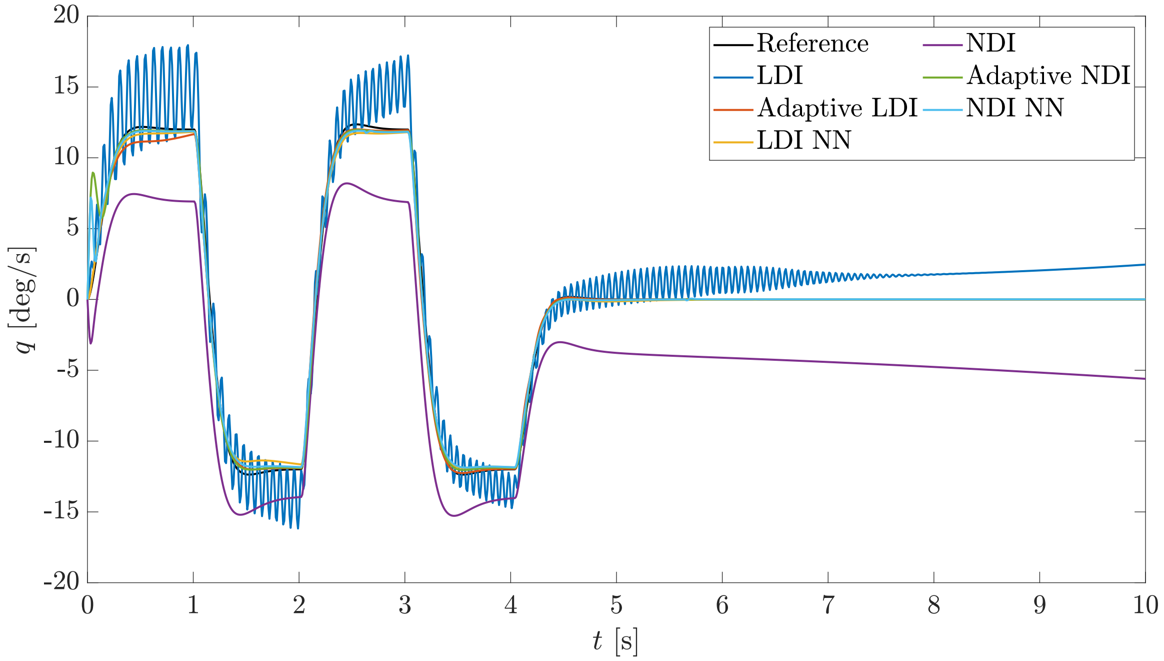


Figure 4.36: Comparison of the pitch rate for the different controllers when there is a maximum relative error of 90% in the coefficients that define the model.

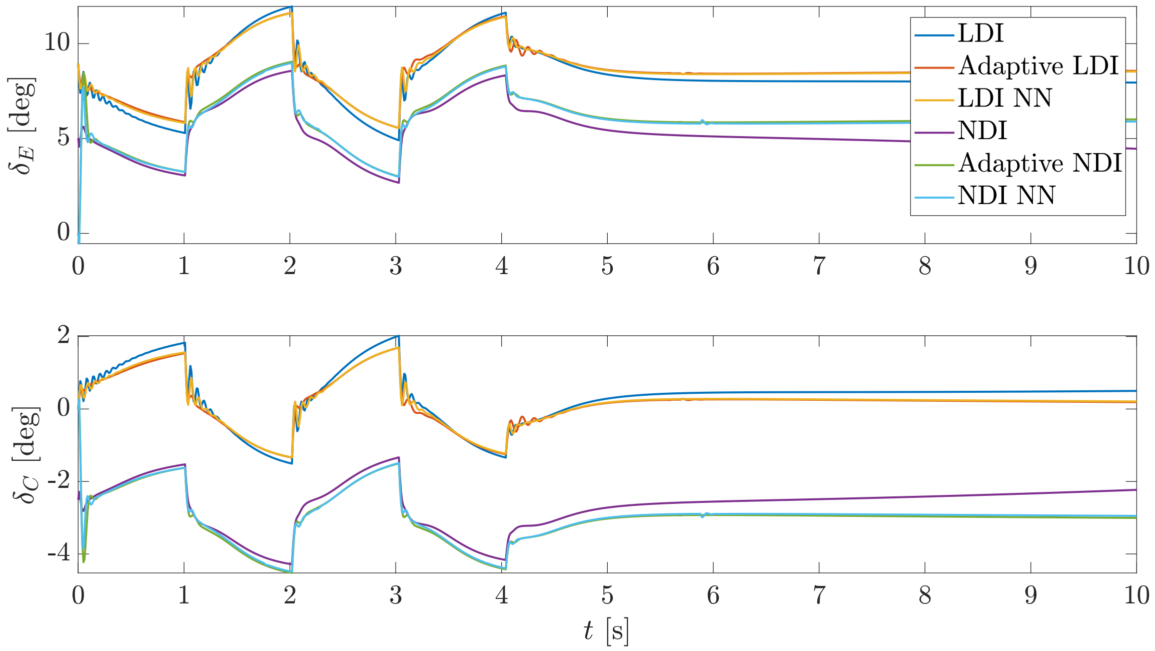


Figure 4.37: Comparison of the control surface deflections for the different controllers when there is a maximum relative error of 50% in the coefficients that define the model.

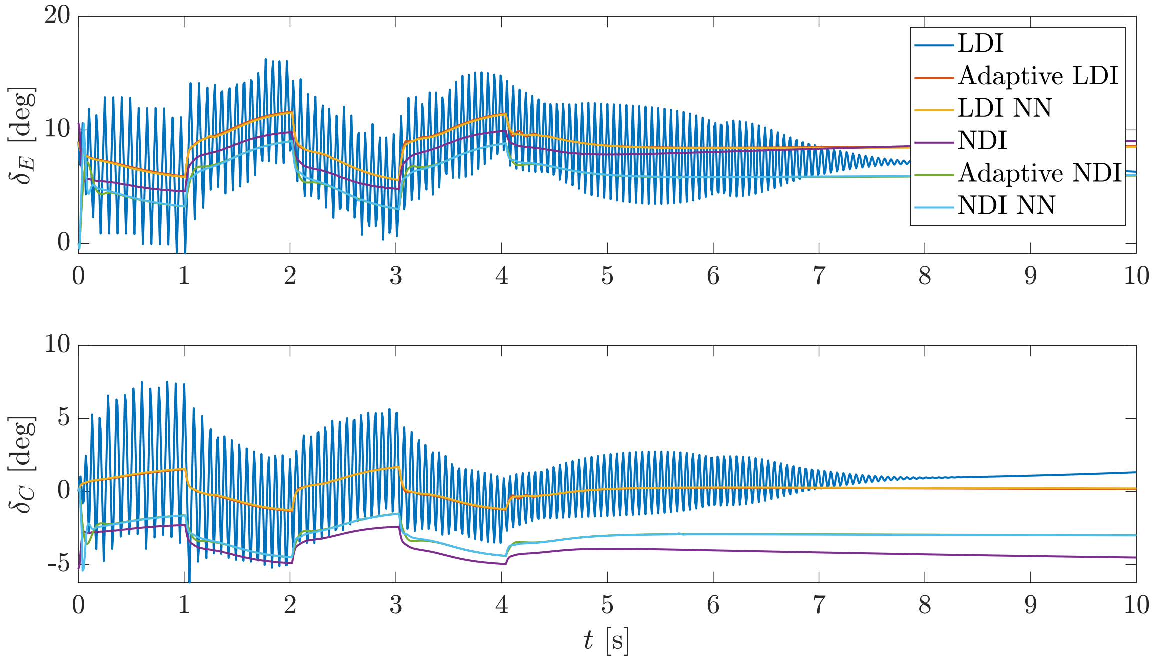


Figure 4.38: Comparison of the control surface deflections for the different controllers when there is a maximum relative error of 90% in the coefficients that define the model.

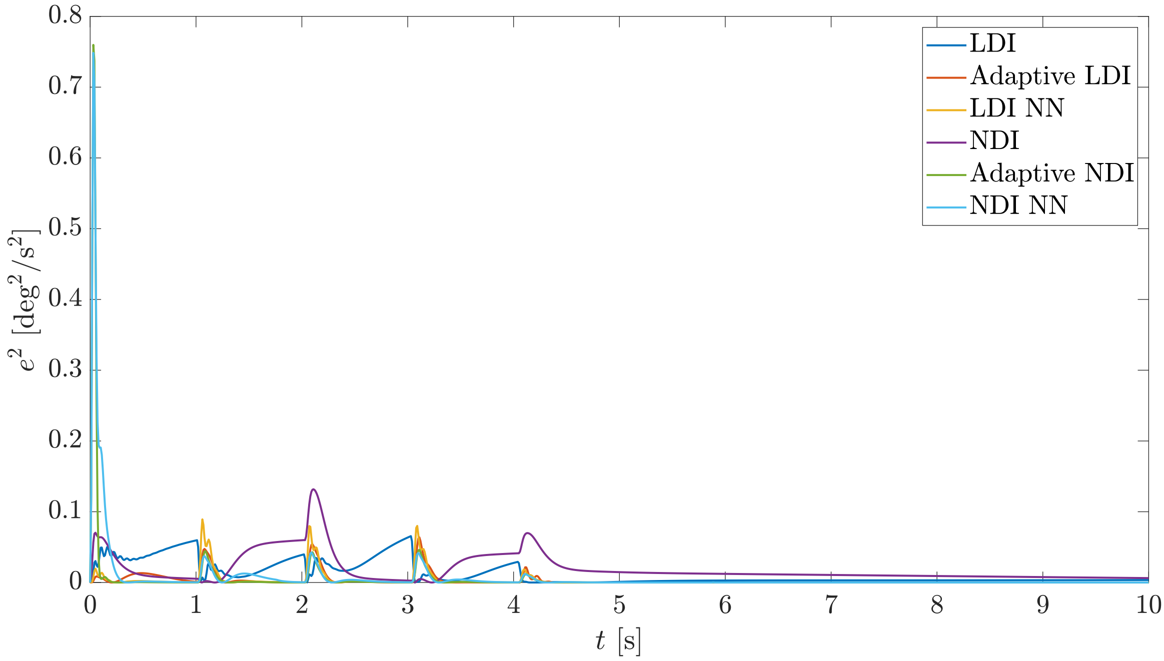


Figure 4.39: Comparison of the quadratic error for the different controllers when there is a maximum relative error of 50% in the coefficients that define the model.

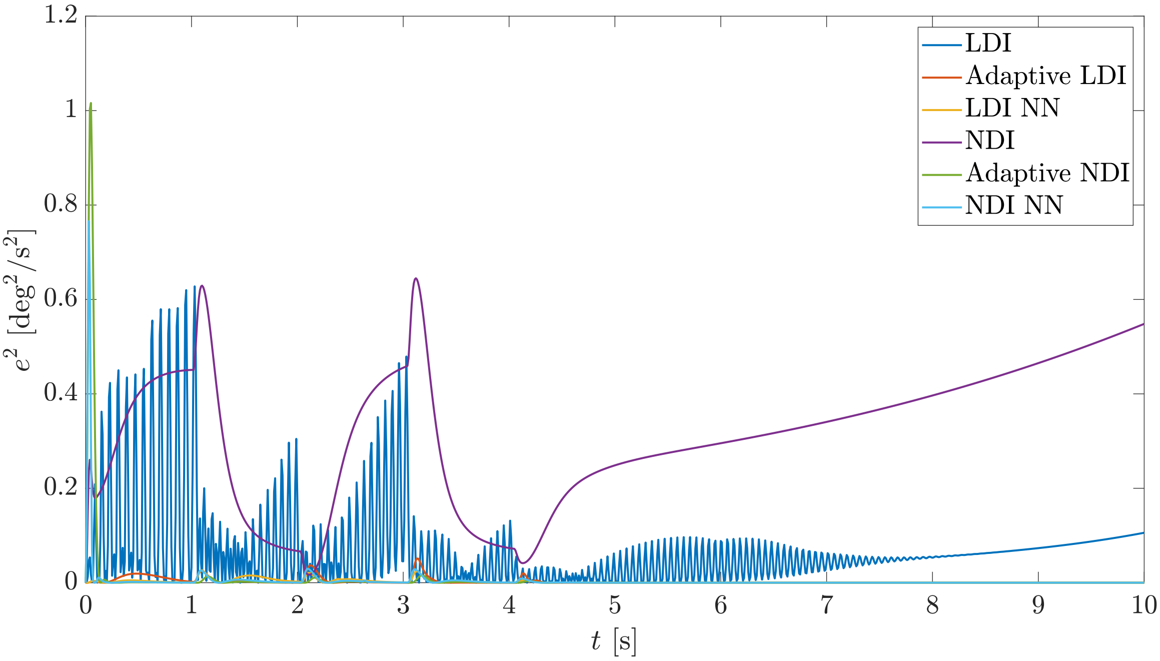


Figure 4.40: Comparison of the quadratic error for the different controllers when there is a maximum relative error of 90% in the coefficients that define the model.

This analysis is specially interesting for this thesis because the final objective is to implement one of these controllers into the subscale model of the GFF. Since the accuracy of the aerodynamic parameters of the subscale model available is limited, the implementation of a non-adaptive controller could present a risk for its good performance. Therefore, the implementation of an adaptive controller seems to be more convenient a priori. However, it is necessary to analyze other significant parameters before making a decision.

4.6 Analysis of the results

The results presented in the previous sections are useful to analyze the performance and fault tolerance capabilities of the different controllers studied in this work. Since only one of them can be implemented in the real aircraft due to time constrains, it is necessary to make a decision. In order to do this, some interesting parameters that measure the performance and the cost of the control algorithms are analyzed.

In the first place, the mean squared error *mse* between the reference and the actual pitch rate for the different cases studied is shown in Table 4.2. It can be observed that the average *mse* is one order of magnitude lower for the adaptive controllers compared to the non-adaptive ones. Also, as it was shown in the previous sections, the adaptive controllers are much more robust in the event of failures or model uncertainties. This last characteristic is specially important because the dynamical model of the subscale model available is not accurate enough to implement a non-adaptive control system.

Case	LDI	Ad. LDI	LDI NN	NDI	Ad. NDI	NDI NN
Nominal	2.2×10^{-5}	6.4×10^{-5}	4.5×10^{-5}	3.6×10^{-5}	7.7×10^{-5}	1.0×10^{-4}
$s_E = 80\%$	1.4×10^{-3}	1.0×10^{-4}	6.1×10^{-5}	5.3×10^{-4}	7.9×10^{-5}	9.4×10^{-5}
$s_E = 50\%$	1.1×10^{-2}	3.9×10^{-4}	2.6×10^{-4}	3.3×10^{-3}	1.2×10^{-4}	2.7×10^{-4}
Jam 5°	5.0×10^{-3}	7.4×10^{-4}	5.3×10^{-4}	1.0×10^{-3}	1.5×10^{-4}	1.9×10^{-4}
Jam 15°	1.6×10^{-2}	5.1×10^{-4}	1.4×10^{-3}	6.0×10^{-2}	2.9×10^{-4}	3.4×10^{-4}
$\sigma_s = 1$	1.1×10^{-4}	4.2×10^{-4}	5.1×10^{-4}	1.8×10^{-4}	4.0×10^{-4}	3.0×10^{-4}
$\sigma_s = 5$	5.8×10^{-4}	2.2×10^{-3}	1.9×10^{-3}	7.6×10^{-4}	1.6×10^{-3}	8.9×10^{-4}
$\sigma_b = 2.5$	1.3×10^{-3}	1.6×10^{-3}	1.6×10^{-3}	1.4×10^{-3}	1.6×10^{-3}	1.5×10^{-3}
$\sigma_b = 5$	5.4×10^{-3}	6.3×10^{-3}	6.3×10^{-3}	5.7×10^{-3}	6.3×10^{-3}	5.4×10^{-3}
$SM = -5\%$	2.4×10^{-5}	3.4×10^{-5}	2.5×10^{-5}	4.1×10^{-5}	1.3×10^{-4}	3.7×10^{-4}
$SM = -30\%$	2.6×10^{-5}	4.4×10^{-5}	5.7×10^{-5}	5.0×10^{-5}	2.3×10^{-4}	3.9×10^{-4}
$\Delta_e = 50\%$	2.1×10^{-4}	4.9×10^{-5}	5.1×10^{-5}	3.4×10^{-4}	8.1×10^{-5}	1.2×10^{-4}
$\Delta_e = 90\%$	1.2×10^{-3}	4.6×10^{-5}	3.7×10^{-5}	5.5×10^{-3}	1.2×10^{-4}	7.1×10^{-5}
Average	3.3×10^{-3}	9.6×10^{-4}	9.8×10^{-4}	6.0×10^{-3}	8.6×10^{-4}	7.7×10^{-4}

Table 4.2: Mean squared error obtained by using the different controllers for all the cases.

Another relevant parameter to analyze is the memory necessary to store the variables that each of the controllers use. Since the memory of the microprocessor installed in the subscale model is limited, it is interesting to implement a controller that does not require too much memory in order to operate correctly. The estimations of memory for each of the controllers, shown in Table 4.3, have been performed by adding the weight of the parameters and variables needed to run the algorithms.

	LDI	Ad. LDI	LDI NN	NDI	Ad. NDI	NDI NN
Memory [Bytes]	104	392	776	64	456	824

Table 4.3: Memory needed to store the variables and parameters that each of the controllers use.

It can be observed that the non-adaptive controllers require much less memory to work since these type of algorithm are quite simple and do not need adaptation laws. Comparing the adaptive controllers, the ones based on neural networks require around twice the memory of those based on simple adaptation. The reason is that the neural network requires more parameters to work.

The last parameter to analyze is the computational cost, measured as computation time. In order to do it, five simulations have been performed with each controller and the average of the simulation time is taken as reference. It is obvious that this value is not the actual computation time once the system is implemented into the microprocessor, however, it is useful to perform a preliminary estimation of the controllers computational cost. The results obtained are shown in Table 4.4.

	LDI	Ad. LDI	LDI NN	NDI	Ad. NDI	NDI NN
Time [s]	11.51	12.76	18.91	11.87	13.82	17.15

Table 4.4: Estimated computational cost of the different controllers.

It can be observed that the controllers based on neural networks have a very high computational cost compared to the rest. Moreover, the non-adaptive algorithms present the lowest computation time since the model of the dynamics is assumed to be known and, unlike in the adaptive ones, it is already implemented inside of the controller. Regarding the controllers based on simple adaptation, it can be notice that the computational cost of both linear and nonlinear versions is very similar.

5 Discussion

In this chapter, the methodology followed in this work and the validity of the results obtained are critically discussed. Finally, one of the controllers is proposed for implementation.

5.1 Discussion of the methodology

Regarding the methodology, the nonlinear model of the aircraft dynamics shown in Eq. (2.4) has been used to create the flight simulator. This model represent the real behavior of the aircraft accurately. However, the model of the aerodynamic forces and moments is linear with the states of the aircraft and the control surface deflections. Moreover, the aerodynamic coefficients used for the simulations are time-invariant and constant regardless of the flight conditions. In reality, the aerodynamic coefficients present a complex dependence with several parameters such as the Mach number and the Reynolds number that characterize the flight conditions, as well as the angle of attack. Also, the presence of highly nonlinear phenomena such as turbulence, stall, shock-waves or aeroelastic effects has important effects in the aerodynamic properties of the aircraft. These complex phenomena have not been included in the simulator and therefore the results obtained may not represent the actual response of the aircraft when the effects of any of these phenomena become relevant.

One way of including these aerodynamic effects in the simulator is the use of lookup tables. This technique would allow to change the aerodynamic coefficients depending on the flight conditions. However, the performance of extensive computational fluid dynamics simulations and flight tests is necessary in order to obtain the data required to build these lookup tables. This procedure is affordable for big companies that require a high-fidelity model of the dynamics in order to fulfill the design specifications and satisfy the airworthiness and safety requirements. Regarding the case of study in this work, the requirements for subscale model applications are not as restrictive as for full-scale aircraft. Therefore, the demand for resources and accuracy to conduct this thesis is much lower than in industry and the simpler model adopted in the simulator is suitable enough.

On another note, it was shown in Chapter 2 that the presence of actuator failures can modify the expected behavior of the aircraft. The partial destruction of the control surfaces produces changes in the aerodynamic coefficients and the jam of the actuator have severe consequences on the aircraft dynamics. However, all the simulations performed in this work consider that these failures occur symmetrically, i.e. the destruction takes place at the same position and at the same time in both elevons or canards and in the case of the jam, it occurs with the same deflection for both surfaces. The reason of modeling the failures this way is that the flight simulator only includes the longitudinal dynamics and an asymmetric failure would induce effects in the lateral-directional dynamics of the aircraft. Therefore, in order to obtain the real response of the aircraft when a random actuator failure occurs, the inclusion of the lateral-directional dynamics in the simulator is mandatory.

Another important matter about the fidelity of the simulations' results is the model of the actuators dynamics. The simulator was created by using a very simple model of the actuator's dynamics that only contemplates their saturation at a maximum or minimum deflection and a rate saturation that limits the maximum angular velocity that the actuators can reach. However, the actual dynamics are defined by more complex phenomena that can be represented by a second order ordinary differential equation in order to include time delays and damping effects. This type of model was not included in the simulator due to the lack of data. This mismatch between the real and the modeled actuator dynamics could play a quite significant role in the control systems performance. The actuator deflections required by the dynamic inversion algorithm can change very fast and chaotically in some cases as shown in Figures 4.19, 4.20 and 4.38. In a real system, this quick response may be damped by the actual dynamics that would act as a filter of the actuator oscillations. However, this effect could cause the loss of control of the aircraft since the resultant control input would not be the one required by the dynamic inversion loop. In order to solve this issue, the model of the actuator dynamics must be included in the dynamic inversion algorithm.

5.2 Discussion of the results

Since the objective of this work is to design one controller for the GFF subscale model, it is necessary to choose one among the different ones studied. In order to do so, the strengths and weaknesses of each controller are highlighted.

In general, the non-adaptive linear and nonlinear controllers present a very similar performance for all the cases studied. Both of them work satisfactorily for nominal conditions and when the model of the dynamics implemented into the controllers is quite accurate. Nevertheless, the results show that these non-adaptive controllers are not suitable when the presence of actuator failures or modeling errors are contemplated for the flight conditions. They also show the lowest storage memory and computational cost needed among all the cases studied. Therefore, if the lack of robustness in the event of failures is affordable and it is assumed that the model of the dynamics available is accurate enough, one of these two controllers would be the best implementation option. Comparing these last two candidates, the linear controller is designed to work for flight conditions that are close to the trim point in the flight envelope. For this reason, the use of gain scheduling would be necessary to run this controller effectively throughout all the flight envelope. On the other hand, the nonlinear controller works satisfactorily regardless of the flight conditions. This makes the last one the simplest and most effective of the non-adaptive controllers.

Regarding the adaptive controllers, it has been shown that their performance is also very similar for both linear or nonlinear candidates and the ones based on simple adaptation or neural networks. All of them are able to work outstandingly in the event of partial destruction or jam of the control surfaces, the presence of modeling errors and throughout all the flight envelope. The only noticeable difference in the performance among these candidates is that the nonlinear controllers need a little more time to learn the parameters of the dynamics to track the reference, although this difference is lower than one second for all the cases studied. However, if the storage memory is compared, it has been shown that the adaptation algorithms based on dynamic inversion need almost twice the memory that the simple adaptation algorithms need. Also, the computational cost is much higher for the neural networks. For these reasons, the best candidates for real implementation among the adaptive controllers are both linear and nonlinear controllers based on simple adaptation.

It is interesting to mention that the classical nonlinear dynamic inversion controller would be the one that presents the best performance according to different sources such as [10] since it is capable of considering the nonlinear phenomena that affect the dynamics and it does not need gain scheduling to work regardless of the flight conditions. However, the inclusion of adaptive algorithms in the dynamic inversion architecture cancels all these advantages given that even the linear controllers are now able to work in the presence of nonlinear phenomena and far from the trim point.

Analyzing the importance of the different failures studied in this work, it can be stated that the sensor failures represent the most dangerous of the possible malfunctions given that the control input commanded by the dynamic inversion algorithm strongly depends on the state feedback provided by the sensors. The measurement noise presents a serious problem since it induces rapid changes in the required control input, what could produce the loss of control given the possible actuators' inability to follow these commands. For this reason, the use of estimators and filters for the measurements is almost mandatory when this type of controllers is used. On the other hand, the presence of a bias between the measurements and the actual states also has significant implications in the controllers' performance because the control system is not able to identify the difference between the desired output and the actual state.

5.3 Implementation proposal

Finally, analyzing the results presented in Chapter 4 and their relative importance when it comes to reaching the performance objectives and evaluate the feasibility of implementation into a real system, the controller that is proposed to be implemented into the GFF subscale model is the adaptive linear dynamic inversion controller. The main reasons are:

- Its performance is satisfactory for all the cases studied, showing good fault tolerance capabilities and robustness in different scenarios.
- It is the adaptive controller that requires the least memory to work. This is a critical issue since the memory of the microprocessor is limited.
- The computational cost is the lowest among the adaptive algorithms.
- Its structure is simple enough to be implemented into the subscale model.

6
Conclusions

Throughout this work, an extensive analysis of the performance of different control systems based on dynamic inversion and model reference adaptive control has been conducted. In order to demonstrate the robustness of these controllers, a series of simulations including different adverse situations has been performed. Finally, the main conclusions obtained from this work are the following:

- The design of flight control systems is currently experiencing a revolution thanks to the development of new computational technologies. The miniaturization of sensors and systems allows to increase the computational power of the on-board computers and therefore, the application of complex algorithms that require large amounts of flight data to work is possible. For this reason, new technologies such as machine learning are starting to be applied to flight control systems.

- Nowadays, the adaptive control systems represent an important research focus thanks to the advantages that they present. These systems are able to work when there exist disturbances or changes in the system characteristics and flight conditions, such as stall, actuator or sensor failures and modeling errors. However, their application in full-scale aircraft is still controversial given the risk that the possible malfunction of these algorithms represent for flight safety.

- The flight simulator created for this work is able to reproduce the longitudinal dynamics of an aircraft. However, although the model of the dynamics is nonlinear, the aerodynamic model is linear and uses constant coefficients. This issue could produce a mismatch between the results of the simulations and the actual behavior of the aircraft.

- In real flight, an actuator failure would probably not be symmetric and effects on the lateral-directional dynamics of the aircraft would appear. Therefore, the results obtained from the longitudinal dynamics simulator in these cases could be false due to the coupling between the longitudinal and lateral-directional dynamics. In order to obtain reliable results, a simulator that includes the three modes is necessary.

- The actuator dynamics play a crucial role in the performance of the controllers. The simple model adopted for this work allows the quick response of the actuators required by the dynamic inversion algorithm. However, the real dynamics could present some delays and overshooting, making the actual deflection of the control surface different from the one commanded by the control law. This issue could cause unstable behavior of the aircraft. To solve this, the actuator dynamics must be considered when designing the controllers.

- The dynamic inversion algorithm is a powerful tool to control the aircraft dynamics. However, it requires a high-fidelity model of the dynamics and a high-quality measurement of the aircraft states. In real systems, this requirements can be difficult to achieve due to possible limitations in the resources available for the modeling process and the quality of the sensors available.

- The combination of the dynamic inversion and the model reference adaptive control techniques solves the weaknesses that each one of them bring individually. The adaptive part of the controller eliminates the need of a high-fidelity model of the dynamics and the dynamic inversion loop considers the nonlinear phenomena, making the controller more reliable and robust.

- The choice of the reference model is one of the most important steps in the design process. This model must be stable and satisfy the handling qualities requirements. However, the aircraft must have the maneuverability necessary to track the reference. A too high demanding reference model could cause the saturation of the aircraft actuators if the control power is not enough.

- From the analysis of the simulation results, it has been shown that the non-adaptive algorithms present a better performance when the model of the dynamics is perfectly known and there is not any disturbance during flight. However, the adaptive schemes are much more reliable and robust in the event of an actuator failure or the presence of modeling errors.

- It has been demonstrated that the presence of sensor failures such as noise or bias is the most detrimental phenomena for the performance of the dynamic inversion controller. Therefore, the correct calibration of the sensors and the use of estimators and filters in the measurements is mandatory for the satisfactory performance of the controller.

- The use of the canards plays an important role in the robustness of the control system. Having this extra control surface decreases the elevon deflection needed to achieve the required pitching moment, preventing from actuator saturation. Moreover, it allows to maintain the aircraft controllable in the event of an actuator jam or high destruction of the control surface, increasing significantly the robustness of the control system.

- The simple adaptation and the neural network algorithms present similar performance for all the cases studied. However, the neural networks require almost twice the storage memory to work and the computational cost is also significantly higher. For this reason, the simple adaptation scheme is more suitable for real implementation in a microprocessor.

- Finally, the linear dynamic inversion based on simple adaptation has been chosen for future implementation thanks to its good performance and robustness, its simplicity, the low storage memory required and its low computational cost.

7 Future Work

During the conduct of this thesis, a series of interesting ideas regarding possible future projects related to the expansion of the work done have emerged:

- Improvement of the aerodynamic model available of the GFF subscale model and characterization of the aerodynamic coefficients throughout the flight envelope in order to implement lookup tables in the flight simulator.
- Implementation of the lateral-directional dynamics in the flight simulator, including the effects that the actuator failures would have in these modes.
- Development of a more complex model of the actuator dynamics and its inclusion in the flight simulator and the control laws.
- Expansion of the control systems in order to control the lateral-directional dynamics of the aircraft.
- Development of an autopilot based on adaptive dynamic inversion that would force the aircraft to track a desired trajectory in a three-dimensional space.
- Exploration of new adaptive algorithms such as Adaptive Robust Control, Sliding Mode Surface and other neural network structures.
- Exploration of the effects on the controllers performance of the use of different controlled variables such as the load factor or the angle of attack.
- Implementation of the chosen controller into the subscale model available in the laboratory and performance of flight tests in order to validate the results obtained from the simulations.

Appendices

A
Simulator Manual

In this appendix, a brief explanation about how to use the flight simulator developed with *MATLAB* and *Simulink* is presented. This simulator recreates the longitudinal dynamics of the specified aircraft and different control systems can be chosen to control these dynamics. Also, a series of abnormal flight conditions can be established.

A.1 Definition of the aircraft parameters

In the first place, it is necessary to specify the parameters that define the inertial and geometrical characteristics of the aircraft as well as the aerodynamic coefficients. To do so, a script containing all this data must be run from the main script. As an example, the script containing the data about the *Generic Future Fighter* is shown. Note that, in case it is desired to create a new script from scratch, it is necessary to define the same variables with the same name since the rest of the functions used by the simulator also depend on these variables. Once all the parameters are defined, they are saved in a *.mat* file in order to be used in the main script.

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%% Geometric parameters, Inertia and Aerodynamic Derivatives of the  %%%
%%% GFF fighter.                                                       %%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
clc; clearvars;

% Mass and Inertia
M   = 17.64;        % Reference Mass [Kg]
Ix  = 0.56;         % Inertia [Kg m2]
Iy  = 5.28;         % Inertia [Kg m2]
Iz  = 5.56;         % Inertia [Kg m2]
Ixz = 0.05;         % Inertia [Kg m2]

% Geometric Parameters
Sw    = 0.921;              % Wing Surface [m2]
bw    = 1.47;               % Wingspan [m]
cw    = 0.627;              % Aerodynamic mean chord [m]
AR    = bw/cw;              % Aspect Ratio [-]
xcog  = 1.6378;             % Position of the center of gravity [m]
xacW  = 1.77;               % Position of the aerodynamic center wing [m]
xacH  = 0;                  % Position of the aerodynamic center tail [m]
xacC  = 1.178;              % Position of the aerodynamic center canard [m]
xC    = xcog - xacC;        % Distance from cog to ac of canard [m]
xcw   = 1.582;              % Start of the MAC
xPN   = xcog+0.086;         % Position of the neutral point
xH    = 0;                  % Distance from cog to ac of tail [m]
xV    = 0.5;                % Distance from cog to ac of rudder [m]
yeL   = 0;                  % Not needed for longitudinal case
zV    = 0;                  % Not needed for longitudinal case
SM    = (xPN-xcog)/cw;      % Static margin [-]

```

```matlab
% Aerodynamic Efficiencies
downw  = 0.3;            % Downwash in the tail
upw    = 0.1;            % Upwash in the canard
downc  = 0.1;            % Downwash in the wing caused by the canard
sidew  = 0.3;            % Sidewash
etah   = 0.9;            % Dynamic pressure efficiency in horizontal tail
etav   = 0.9;            % Dynamic pressure efficiency in vertical tail
taude  = 0.5;            % Elevator angle of attack effectiveness
taudc  = 1;              % Canard angle of attack effectiveness
taudel = 1;              % Elevon angle of attack effectiveness
taudr  = 0.6;            % Rudder effectiveness
osw    = 0.3038;         % Oswald parameter
K      = 1/(pi*AR*osw);  % Drag constant

% Aerodynamic Derivatives (Symmetric airfoil NACA0008)
CL0   = -0.0168;
CLa   = 2.5376;
CLad  = 0.9299*2;
CLq   = -5*2;
CLde  = 0;
CLdc  = 0.1406;
CLdel = 0.5641;

CD0   = 0.0260;
CDa   = 0;
CDde  = 0;
CDdc  = 0;
CDdel = 0;

Cm0   = 0.0534;
Cma   = -0.2;
Cmad  = -0.1596*2;
Cmq   = -1.4692*2;
Cmde  = 0;
Cmdc  = 0.1823;
Cmdel = -0.2816;

% Calculation of the wing-body and tail contributions
CL0H   = CL0/taude*(1-downw)*CLde/CLa;
CL0C   = CL0/taudc*(1+upw)*CLdc/CLa;
CL0W   = CL0/taudel*(1-downc)*CLdel/CLa;
CL0B   = CL0 - CL0W - CL0H - CL0C;
CLaH   = 1/taude*(1-downw)*CLde;
CLaC   = 1/taudc*(1+upw)*CLdc;
CLaW   = 1/taudel*(1-downc)*CLdel;
CLaB   = CLa - CLaW - CLaH -CLaC;
CLqH   = 2*xH/(taude*cw)*CLde;
CLqC   = -2*xC/(taudc*cw)*CLdc;
CLqWB  = CLq - CLqH - CLqC;
CLadH  = 2*(xacH-xacW)*downw*CLde/(taude*cw);
CLadW  = 2*(xacW-xacC)*downc*CLdel/(taudel*cw);
CLadB  = CLad - CLadW - CLadH;

Cm0H   = Cm0/taude*(1-downw)*(xH/cw)*(CLde/Cma);
Cm0W   = -Cm0/taudel*(1-downc)*((xcog-xacW)/cw)*(CLdel/Cma);
```

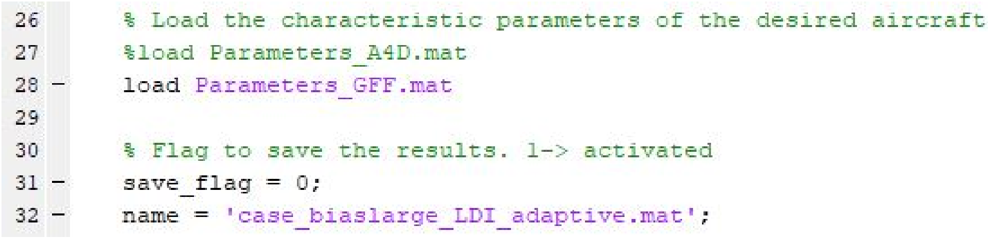
```
Cm0C    = -Cm0/taudc*(1+upw)*(xC/cw)*(CLdc/Cma);
Cm0B    = Cm0 - Cm0H - Cm0W - Cm0C;
CmaH    = -(1-downw)*xH/(taude*cw)*CLde;
CmaW    = 1/taudel*(1-downc)*(xcog-xacW)/cw*CLdel;
CmaC    = 1/taudc*(1+upw)*xC/cw*CLdc;
CmaB    = Cma - CmaH - CmaW - CmaC;
CmqH    = -CLqH*(xH/cw);
CmqC    = CLqC*(xC/cw);
CmqWB   = Cmq - CmqH - CmqC;
CmadH   = -CLadH*xH/cw;
CmadW   = -CLadW*(xacW-xcog)/cw;
CmadB   = Cmad - CmadH - CmadW;

Cn0    = 0;
Cnb    = 0;
Cnp    = 0;
Cnr    = 0;
Cnda   = 0;
Cndr   = 0;
CnbV   = 0;
CnbAC  = 0;

Cl0    = 0;
Clb    = 0;
Clp    = 0;
Clr    = 0;
Clda   = 0;
Cldr   = 0;
ClbV   = 0;
ClbAC  = 0;

CY0   = 0;
CYb   = 0;
CYp   = 0;
CYr   = 0;
CYda  = 0;
CYdr  = 0;

save Parameters_GFF.mat
```

Parameters_GFF.m

A.2 Main script

The main script, with name *Main_ Simulator.m*, is the program from which all the settings of the simulations are configured. It is in this script where the flight conditions are defined, the parameters needed by the controllers are established and the simulation time and input commanded by the pilot are chosen.

The first step is to load the parameters of the aircraft that are stored in a separate script, as explained in the previous section. After that, the possibility of saving the results of the simulation with a desired file name is presented (Figure A.1).

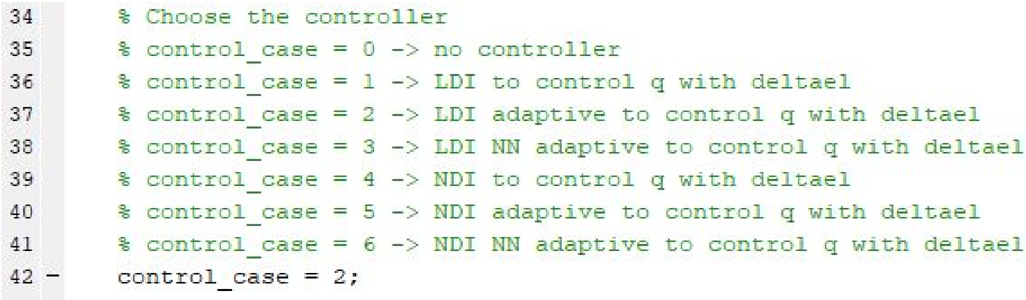


```
% Load the characteristic parameters of the desired aircraft
%load Parameters_A4D.mat
load Parameters_GFF.mat

% Flag to save the results. 1-> activated
save_flag = 0;
name = 'case_biaslarge_LDI_adaptive.mat';
```

Figure A.1: Step 1: Load the aircraft parameters and choose if it is desired to save the simulation results.

The next step is to choose the control system based on dynamic inversion DI and model reference control applied to the aircraft in order to control the pitch rate (Figure A.2). The possibilities depending on the value of the parameter *control_ case* are the following:

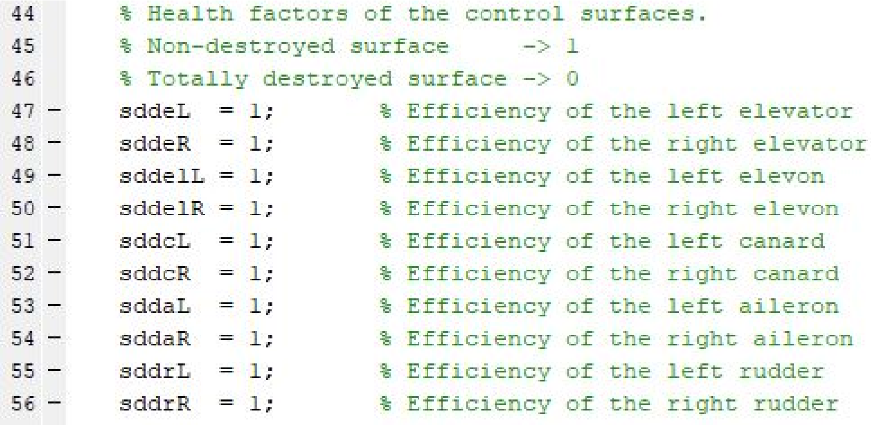
- *control_ case*= 0: no controller applied. The dynamics of the aircraft are directly controlled with the input commanded by the pilot with the control pad stick.
- *control_ case*= 1: linear DI controller.
- *control_ case*= 2: adaptive linear DI controller based on simple adaptation.
- *control_ case*= 3: adaptive linear DI controller based on neural networks.
- *control_ case*= 4: nonlinear DI controller.
- *control_ case*= 5: adaptive nonlinear DI controller based on simple adaptation.
- *control_ case*= 6: adaptive nonlinear DI controller based on neural networks.



```
% Choose the controller
% control_case = 0 -> no controller
% control_case = 1 -> LDI to control q with deltael
% control_case = 2 -> LDI adaptive to control q with deltael
% control_case = 3 -> LDI NN adaptive to control q with deltael
% control_case = 4 -> NDI to control q with deltael
% control_case = 5 -> NDI adaptive to control q with deltael
% control_case = 6 -> NDI NN adaptive to control q with deltael
control_case = 2;
```

Figure A.2: Step 2: Choose the control system.

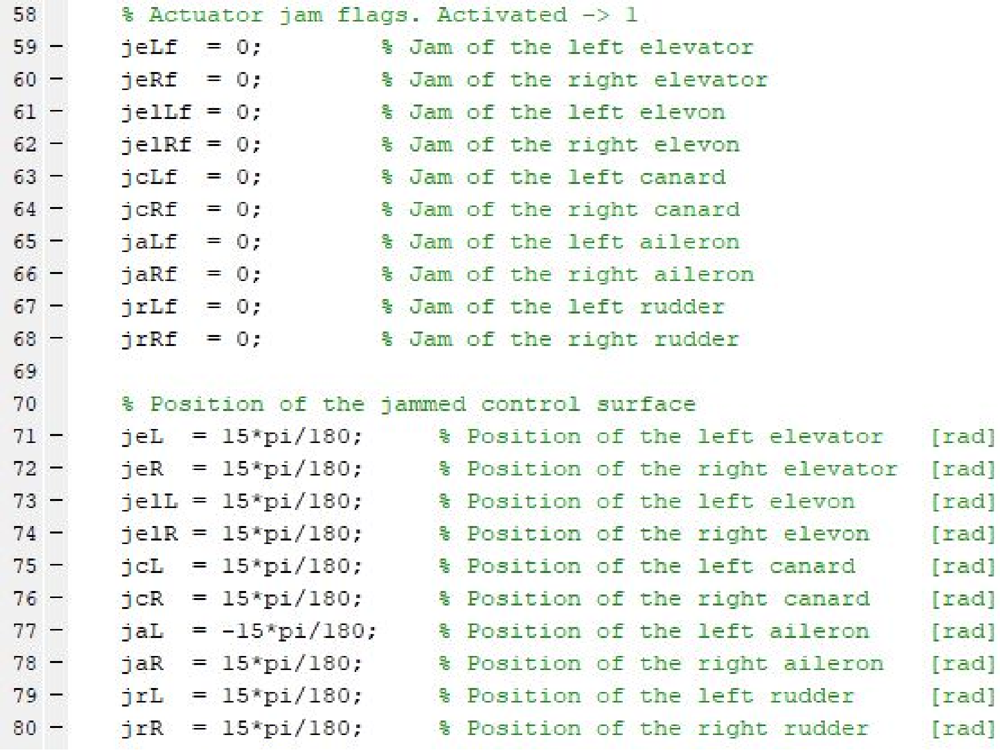
Once the control system has been established, it is necessary to define the flight conditions. In the first place, the presence of actuator failures such as control surface destruction or jam of an actuator must be established. First, the values of the control surfaces health factors must be defined as shown in Figure A.3.



```
% Health factors of the control surfaces.
% Non-destroyed surface     -> 1
% Totally destroyed surface -> 0
sddeL  = 1;       % Efficiency of the left elevator
sddeR  = 1;       % Efficiency of the right elevator
sddelL = 1;       % Efficiency of the left elevon
sddelR = 1;       % Efficiency of the right elevon
sddcL  = 1;       % Efficiency of the left canard
sddcR  = 1;       % Efficiency of the right canard
sddaL  = 1;       % Efficiency of the left aileron
sddaR  = 1;       % Efficiency of the right aileron
sddrL  = 1;       % Efficiency of the left rudder
sddrR  = 1;       % Efficiency of the right rudder
```

Figure A.3: Step 3: Definition of the control surfaces health factors.

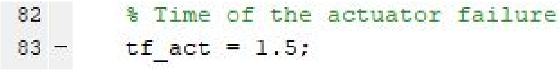
Also, the jam of a control surface is defined by activating the corresponding flag in case of jam and defining the value of the angle at which the actuator has been stuck, as shown in Figure A.4.



```
% Actuator jam flags. Activated -> 1
jeLf  = 0;        % Jam of the left elevator
jeRf  = 0;        % Jam of the right elevator
jelLf = 0;        % Jam of the left elevon
jelRf = 0;        % Jam of the right elevon
jcLf  = 0;        % Jam of the left canard
jcRf  = 0;        % Jam of the right canard
jaLf  = 0;        % Jam of the left aileron
jaRf  = 0;        % Jam of the right aileron
jrLf  = 0;        % Jam of the left rudder
jrRf  = 0;        % Jam of the right rudder

% Position of the jammed control surface
jeL  = 15*pi/180;     % Position of the left elevator   [rad]
jeR  = 15*pi/180;     % Position of the right elevator  [rad]
jelL = 15*pi/180;     % Position of the left elevon     [rad]
jelR = 15*pi/180;     % Position of the right elevon    [rad]
jcL  = 15*pi/180;     % Position of the left canard     [rad]
jcR  = 15*pi/180;     % Position of the right canard    [rad]
jaL  = -15*pi/180;    % Position of the left aileron    [rad]
jaR  = 15*pi/180;     % Position of the right aileron   [rad]
jrL  = 15*pi/180;     % Position of the left rudder     [rad]
jrR  = 15*pi/180;     % Position of the right rudder    [rad]
```

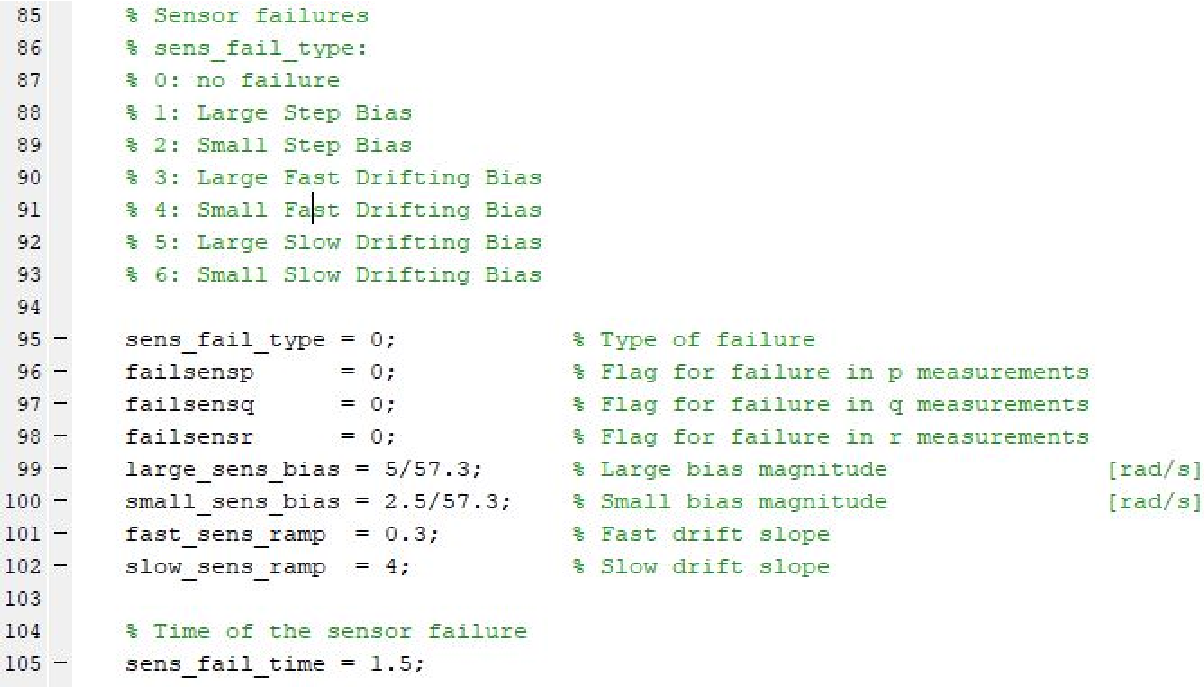
Figure A.4: Step 4: Definition of the control surfaces that are jammed and the value of the jam angle.

To finish the definition of the actuator failures it is necessary to establish the time at which they occur, as shown in Figure A.5.



```
% Time of the actuator failure
tf_act = 1.5;
```

Figure A.5: Step 5: Definition of the actuator failure time.



```
% Sensor failures
% sens_fail_type:
% 0: no failure
% 1: Large Step Bias
% 2: Small Step Bias
% 3: Large Fast Drifting Bias
% 4: Small Fast Drifting Bias
% 5: Large Slow Drifting Bias
% 6: Small Slow Drifting Bias

sens_fail_type = 0;            % Type of failure
failsensp       = 0;           % Flag for failure in p measurements
failsensq       = 0;           % Flag for failure in q measurements
failsensr       = 0;           % Flag for failure in r measurements
large_sens_bias = 5/57.3;      % Large bias magnitude                [rad/s]
small_sens_bias = 2.5/57.3;    % Small bias magnitude                [rad/s]
fast_sens_ramp  = 0.3;         % Fast drift slope
slow_sens_ramp  = 4;           % Slow drift slope

% Time of the sensor failure
sens_fail_time = 1.5;
```

Figure A.6: Step 6: Definition of the sensor failures and the time at which they occur.

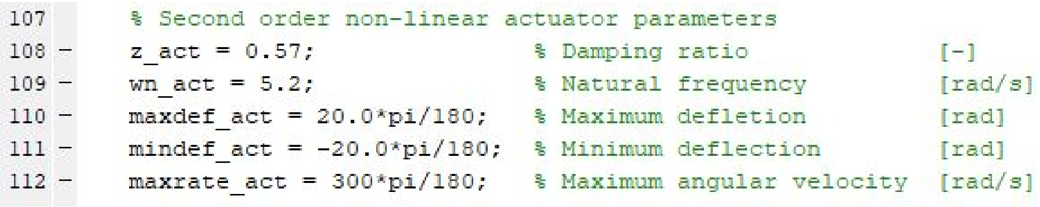
The following step is the definition of the sensor failures. First, the type of failure must be specified by using the variable *sens_fail_ type* and the options are the following:

- *sens_fail_ type* = 0: no failure in the sensors.
- *sens_fail_ type* = 1: presence of a large constant step bias between the measurement and the actual state.
- *sens_fail_ type* = 2: presence of a small constant step bias between the measurement and the actual state.
- *sens_fail_ type* = 3: presence of a bias with maximum magnitude equal to the large step bias case that is drifting with time with a large slope.
- *sens_fail_ type* = 4: presence of a bias with maximum magnitude equal to the small step bias case that is drifting with time with a large slope.
- *sens_fail_ type* = 5: presence of a bias with maximum magnitude equal to the large step bias case that is drifting with time with a small slope.

- *sens_fail_type* = 6: presence of a bias with maximum magnitude equal to the small step bias case that is drifting with time with a small slope.

Once the type of failure has been specified, it is necessary to establish the sensors that are affected by this failure by activating the corresponding flags. It is possible to choose between the measurements of the roll rate, the pitch rate and the yaw rate. Also, the maximum magnitude of the bias and the slope of the drifting bias with time must be specified for the different cases, as shown in Figure A.6. Finally, the time at which these sensor failures occur must be established.

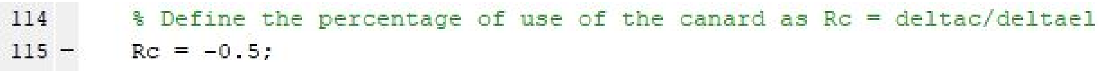
After the configuration of the possible failures, the parameters that define the dynamics of the actuators must be defined as shown in Figure A.7. These parameters are the natural frequency, the damping ratio, the maximum and minimum deflections and the maximum angular rate. Although the natural frequency and the damping ratio are not used in the *Simulink* model, their values are shown here for the future implementation of a more accurate model.



```
% Second order non-linear actuator parameters
z_act = 0.57;                % Damping ratio           [-]
wn_act = 5.2;                % Natural frequency       [rad/s]
maxdef_act = 20.0*pi/180;    % Maximum defletion       [rad]
mindef_act = -20.0*pi/180;   % Minimum deflection      [rad]
maxrate_act = 300*pi/180;    % Maximum angular velocity  [rad/s]
```

Figure A.7: Step 7: Definition of the parameters that define the actuator dynamics.

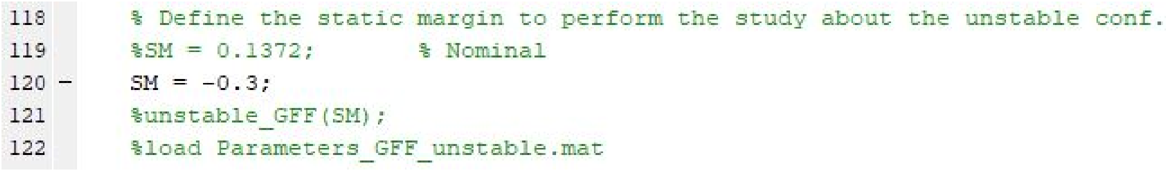
The next step is to define the constant that establish the proportional relation between the deflection of the canards and the elevons, as explained in Section 3.5 and shown in Figure A.8. This constant can be chosen arbitrarily although positive values could make the dynamics of the aircraft to become unstable. A value of 0 means that the canards are not used and their deflection is constant at 0 degrees.



```
% Define the percentage of use of the canard as Rc = deltac/deltael
Rc = -0.5;
```

Figure A.8: Step 8: Definition of the relation between the deflection of the canard and the elevons.

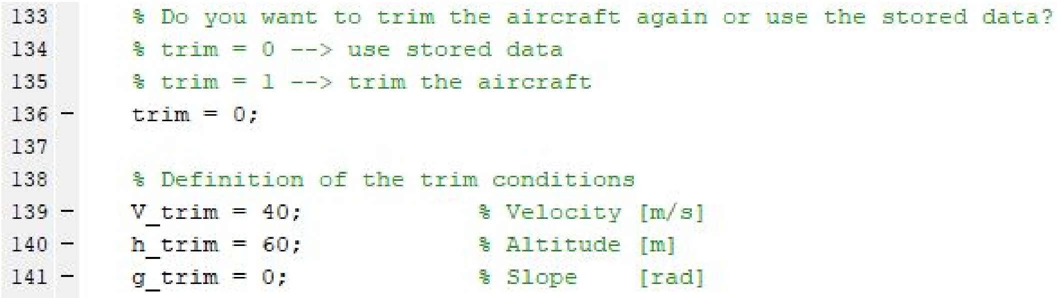
Now, it is necessary to define the static margin of the aircraft in order to perform the study of the controllers' performance when the aircraft is flying with unstable configuration. To do so, lines 121 and 122 of the script must be uncommented. The function *unstable_GFF.m* modifies the aerodynamic coefficients of the aircraft depending on the defined static margin and saves the data in a file called *Parameters_GFF_unstable.mat.* If it is not desired to perform the analysis for unstable configuration, the whole block shown in Figure A.9 can be commented.



```
% Define the static margin to perform the study about the unstable conf.
%SM = 0.1372;        % Nominal
SM = -0.3;
%unstable_GFF(SM);
%load Parameters_GFF_unstable.mat
```

Figure A.9: Step 9: Definition of the static margin and modification of the aircraft parameters for the study of unstable configuration.

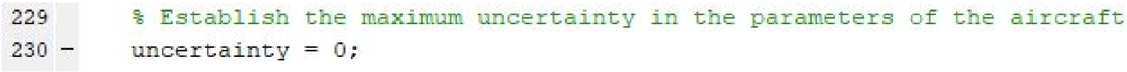
Once the parameters of the aircraft have finally been defined, it is possible to choose between trimming the aircraft for the desired trim conditions or use the trim parameters used in previous simulations and that are saved in a separate file (Figure A.10). The objective of this step is to reduce the computation time of the main script since the trim conditions are calculated by using an iterative algorithm that could require some seconds to run. Therefore, if the trim flag is set to 1, the aircraft will be trimmed again with the defined trim flight conditions. If it is set to zero, the stored variables will be used. The function that trims the aircraft returns the angle of attack, the pitch angle, the thrust and the control surfaces deflections necessary to achieve the specified trim conditions.



```
% Do you want to trim the aircraft again or use the stored data?
% trim = 0 --> use stored data
% trim = 1 --> trim the aircraft
trim = 0;

% Definition of the trim conditions
V_trim = 40;            % Velocity [m/s]
h_trim = 60;            % Altitude [m]
g_trim = 0;             % Slope    [rad]
```

Figure A.10: Step 10: Choose if it is desired to trim the aircraft or use the trim conditions stored in a separate file. Also, define the trim conditions in case it is desired to trim the aircraft.

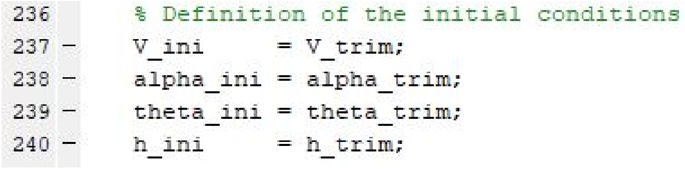
It is also possible to establish some uncertainty in the parameters of the model of the dynamics in order to analyze the performance of the dynamic inversion controllers when the model implemented does not represent the behavior of the aircraft perfectly. To do so, a variable that established the maximum uncertainty possible in the parameters of the dynamics is defined as shown in Figure A.11. All the aerodynamic and control derivatives as well as the parameters that define the nonlinear model of the dynamics will be modified by a random scalar which maximum value is the one defined with the variable *uncertainty*. This way, some parameters will be modified by this maximum value while other will remain unchanged.



```
% Establish the maximum uncertainty in the parameters of the aircraft
uncertainty = 0;
```

Figure A.11: Step 11: Define the maximum uncertainty in the aircraft parameters.

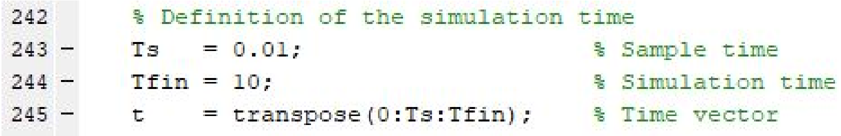
The next step is the definition of the initial conditions used for the simulation. In the case shown in Figure A.12, the initial conditions have been set equal to the trim conditions. However, any initial conditions can be arbitrarily defined.



```
% Definition of the initial conditions
V_ini     = V_trim;
alpha_ini = alpha_trim;
theta_ini = theta_trim;
h_ini     = h_trim;
```

Figure A.12: Step 12: Definition of the simulation's initial conditions.

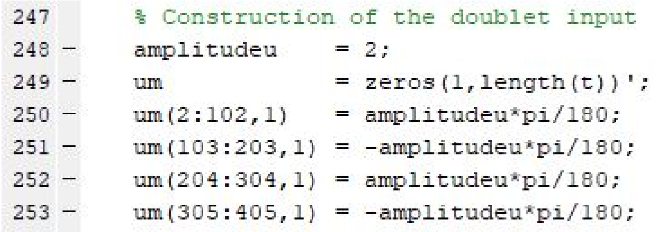
Now, the sample time and the simulation time must be defined as shown in Figure A.13.



```
% Definition of the simulation time
Ts   = 0.01;                    % Sample time
Tfin = 10;                      % Simulation time
t    = transpose(0:Ts:Tfin);    % Time vector
```

Figure A.13: Step 13: Definition of the sample and simulation time.

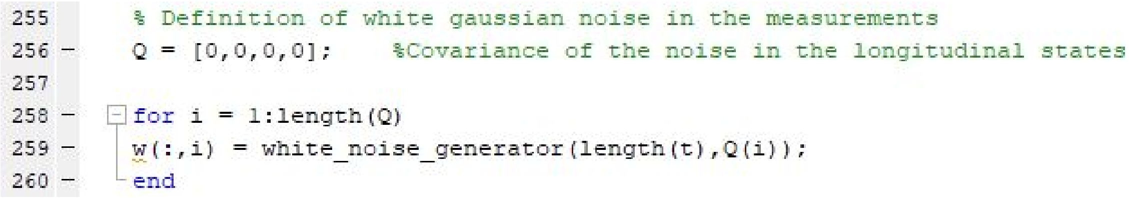
The next step is the definition of the control input commanded by the pilot. In the case shown in Figure A.14, a doublet with amplitude *amplitudeu* has been defined, although this control input can also be arbitrarily defined.



```
% Construction of the doublet input
amplitudeu    = 2;
um            = zeros(1,length(t))';
um(2:102,1)   = amplitudeu*pi/180;
um(103:203,1) = -amplitudeu*pi/180;
um(204:304,1) = amplitudeu*pi/180;
um(305:405,1) = -amplitudeu*pi/180;
```

Figure A.14: Step 14: Definition of control input commanded by the pilot.

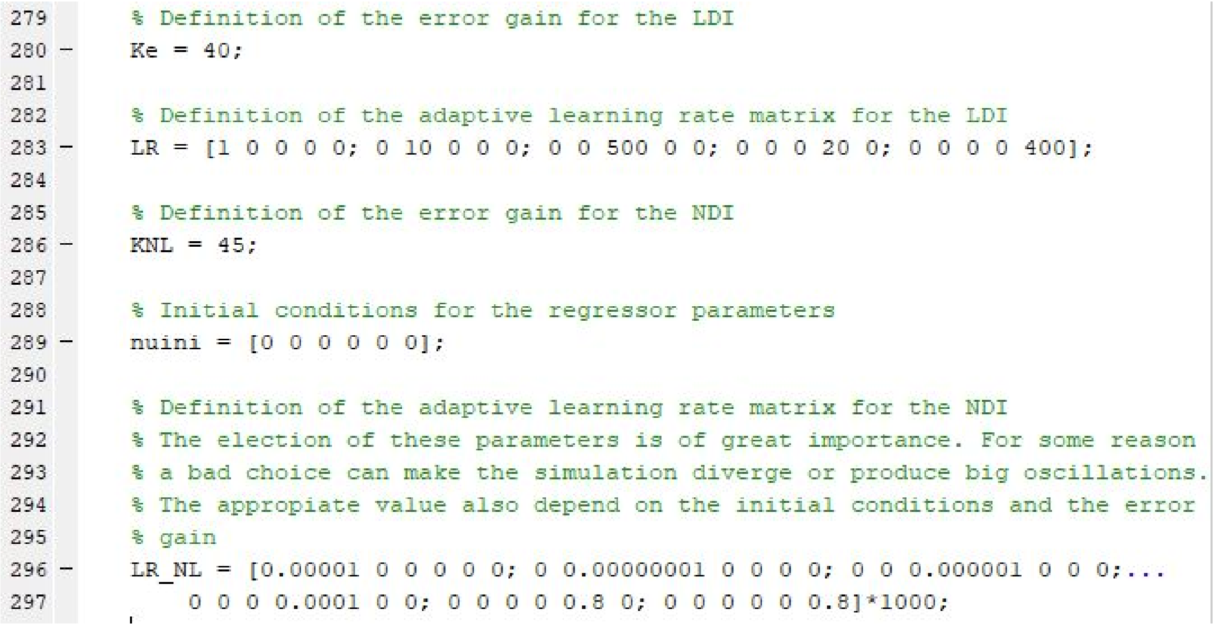
It is also necessary to establish the covariance of the noise affecting the measurements of the states that define the aircraft dynamics, as shown in Figure A.15. In order, each of the positions in the vector Q represent the covariance of the noise in the airspeed, the angle of attack, the pitch rate and the pitch angle. Finally, a vector of Gaussian white noise is created by using the function *white_ noise_ generator.m*. This noise vector will be added to the measurement of the states in the simulation.



```
% Definition of white gaussian noise in the measurements
Q = [0,0,0,0];    %Covariance of the noise in the longitudinal states

for i = 1:length(Q)
w(:,i) = white_noise_generator(length(t),Q(i));
end
```

Figure A.15: Step 15: Definition of the noise covariance for the measurement of the aircraft states and creation of a noise vector.

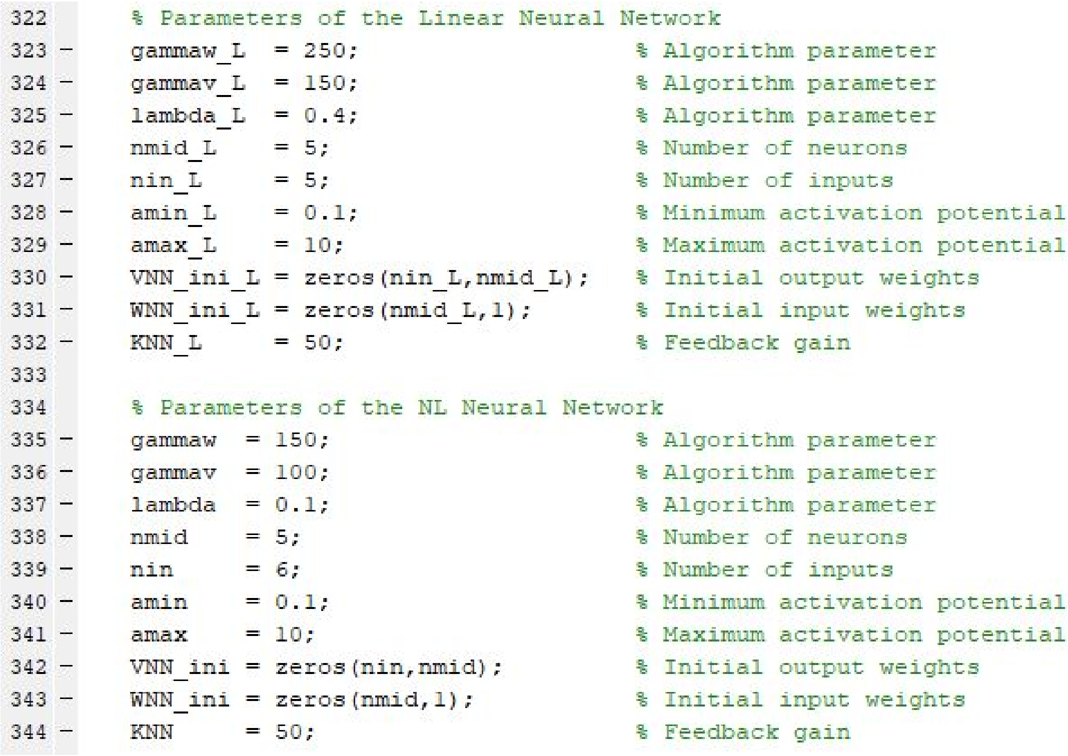
The last step before running the simulations consists on defining the different parameters that the control systems need to work, such as the feedback gains, the learning rates and the initial conditions of the algorithms (Figures A.16 and A.17). All the feedback gains and learning rates must be positive for the controller to be stable. Also, the closed-loop dynamics of the aircraft could be unstable depending on the values of these gains and learning rates for some of the cases studied. Therefore, it could be necessary to tune these gains in order to stabilize the system.



```
% Definition of the error gain for the LDI
Ke = 40;

% Definition of the adaptive learning rate matrix for the LDI
LR = [1 0 0 0 0; 0 10 0 0 0; 0 0 500 0 0; 0 0 0 20 0; 0 0 0 0 400];

% Definition of the error gain for the NDI
KNL = 45;

% Initial conditions for the regressor parameters
nuini = [0 0 0 0 0 0];

% Definition of the adaptive learning rate matrix for the NDI
% The election of these parameters is of great importance. For some reason
% a bad choice can make the simulation diverge or produce big oscillations.
% The appropiate value also depend on the initial conditions and the error
% gain
LR_NL = [0.00001 0 0 0 0 0; 0 0.00000001 0 0 0 0; 0 0 0.000001 0 0 0;...
    0 0 0 0.0001 0 0; 0 0 0 0 0.8 0; 0 0 0 0 0 0.8]*1000;
```

Figure A.16: Step 16.1: Definition of the feedback gains, learning rates and initial estimations for the nonadaptive and simple adaptation versions of the linear and nonlinear controllers.

Finally, when the simulation finishes, the values of the aircraft states and the control surface deflections are saved in case it is desired to post-process the results. Also, the plots of the evolution of the states and the control inputs with time are generated, as well as the two-dimensional trajectory of the aircraft.



```
% Parameters of the Linear Neural Network
gammaw_L  = 250;                   % Algorithm parameter
gammav_L  = 150;                   % Algorithm parameter
lambda_L  = 0.4;                   % Algorithm parameter
nmid_L    = 5;                     % Number of neurons
nin_L     = 5;                     % Number of inputs
amin_L    = 0.1;                   % Minimum activation potential
amax_L    = 10;                    % Maximum activation potential
VNN_ini_L = zeros(nin_L,nmid_L);   % Initial output weights
WNN_ini_L = zeros(nmid_L,1);       % Initial input weights
KNN_L     = 50;                    % Feedback gain

% Parameters of the NL Neural Network
gammaw  = 150;                     % Algorithm parameter
gammav  = 100;                     % Algorithm parameter
lambda  = 0.1;                     % Algorithm parameter
nmid    = 5;                       % Number of neurons
nin     = 6;                       % Number of inputs
amin    = 0.1;                     % Minimum activation potential
amax    = 10;                      % Maximum activation potential
VNN_ini = zeros(nin,nmid);         % Initial output weights
WNN_ini = zeros(nmid,1);           % Initial input weights
KNN     = 50;                      % Feedback gain
```

Figure A.17: Step 16.2: Definition of the parameters needed by the controllers based on neural networks.

Bibliography

[1] Joseph R. Chambers. *Modelling Flight: The Role of Dynamically Scaled Free-Flight Models in Support of NASA's Aerospace Programs.* NASA, Washington DC, USA, first edition, 2010.

[2] Christopher Jouannet, Patrick Berry, Petter Krus. Aircraft design education at Linköpings University. In *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, volume 221, pages 217-224, 2007.

[3] *Ice In or On Static System Cause of X-31 Crash*, NASA, November 1995.

[4] *MATLAB and Simulink 2018b*, The MathWorks, Inc., Natick, Massachusetts, United States.

[5] *Test Drive: Shuttle Training Aircraft Preps Astronauts for Landing*, NASA, March 2005.

[6] I. M. Gregory, C. Cao, E. Xargay, N. Hovakimyan, X. Zou, *L1 Adaptive Control Design for NASA AirSTAR Flight Test Vehicle*, NASA Langley Research Center.

[7] K. A. Wise, E. Lavretsky, *Robust and Adaptive Control of X-45A J-UCAS: A Design Trade Study*, The Boeing Company, IFAC, 2011.

[8] Alejandro S. Rueda, *Design and Testing of a Flight Control System for Unstable Subscale Aircraft*, IEI, Linköpings University, 2015.

[9] *Military Specification for the Flying Qualities of Piloted Airplanes MIL-F-8785C*

[10] Brian L. Stevens, Frank L. Lewis, Eric N. Johnson, *Aircraft Control and Simulation. Dynamics, Control Design and Autonomous Systems*, Wiley, 2016.

[11] James R. Fisher, *Aircraft Control Using Nonlinear Dynamic Inversion in Conjunction with Adaptive Robust Control*, 2004.

[12] K.S. Narendra, K. Parthasarathy, *Identification and Control of Dynamical Systems Using Neural Networks*, IEEE Transactions on Neural Networks, 1990.

[13] F.C. Chen, H.K. Khalil, *Adaptive Control of Nonlinear Systems Using Neural Networks*, International Journal of Control, 1992.

[14] F.C. Chen, C.C. Liu, *Adaptively Controlling Nonlinear Continuous-Time Systems Using Multilayer Neural Networks*, IEEE Trans. Autom. Contr, 1994.

[15] G. Cybenko, *Approximation by Superpositions of a Sigmoidal Function*, Mathematics of Control, Signals and Systems, Springer, 1989.

[16] Dale Enns, Dan Bugajski, Russ Hendrick, Gunter Stein, *Dynamic Inversion: An Evolving Methodology for Flight Control Design*, International Journal of Control, 1994.

[17] *Active Control Technology: Applications and Lessons Learned*, AGARD Conference Proceedings 560, AGARD, 1994.

[18] Anthony J. Calise, Rolf T. Rysdyk, *Nonlinear Adaptive Flight Control Using Neural Networks*, Georgia Institute of Technology, School of Aerospace Engineering.

[19] Christopher J. Miller, *Nonlinear Dynamic Inversion Baseline Control Law: Architecture and Performance Predictions*, NASA Dryden Flight Research Center.

[20] Jan Roskam, *Airplane Flight Dynamics and Automatic Flight Controls, Part 1*, DARcorporation, 1995.

[21] Marcello R. Napolitano, *Aircraft Dynamics: From Modeling to Simulation*, Wiley, 2012.

[22] Dmitry Ignatyev, Alexander Khrabrov, *Experimental Study and Neural Network Modeling of Aerodynamic Characteristics of Canard Aircraft at High Angles of Attack*, MDPI, March 2018.