INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Reconfigurable Flight Control using Model Predictive Control

José Duarte Pereira Gonçalves

Dissertação para obtenção do Grau de Mestre em

Engenharia Aeroespacial

Júri

Presidente:	Prof. João Manuel Lage de Miranda Lemos
Orientador:	Prof. Bertinho Manuel D' Andrade da Costa
Co-Orientador:	Prof. João Manuel Lage de Miranda Lemos
Vogal:	Prof. José Raul Carreira Azinheira

Março 2009

Dedico este trabalho ao meu avô, que sempre se preocupou com a minha carreira académica, e certamente continuarà a olhar por mim.

Acknowledgements

I would like to thank my supervisor, Professor Bertinho Manuel D' Andrade da Costa, for his supervision and support during the development of this project. I am thankful to my co-supervisor, Professor João Manuel Lage de Miranda Lemos, for his suggestions and guidelines for the development of this thesis.

I could never express my full gratitude to my family, especially my parents, for the unconditional support through all my life, for always having faith in me and for the constant wise words of advice. Last but not least, I would like to thank all my friends, directly or indirectly involved with the present thesis.

Abstract

Aircraft accidents due to control surface failures have prompted research on controllers with reconfiguration capabilities. This thesis describes a design approach incorporating Model Predictive Control (MPC) with a linear internal model to achieve a level of reconfiguration in a generic Uninhabited Aerial Vehicle (UAV).

MPC has the advantage of allowing constraints on the inputs, outputs, and states of the system possessing as well tuning flexibility. This thesis describes the aircraft control surfaces fault implementation and the development of two distinct controllers to overcome these failures. Each controller is subject to a range of failures for a certain time period, during a generic manoeuvre. Finally controllers resulting action is analysed.

Keywords: Aircraft, Model Predictive Control, Reconfigurable Control, Fault Tolerant Control, Fault Detection Identification.

Resumo

Acidentes aéreos devido a falhas na superfícies de controlo, motivaram a investigação na área de controladores com um certo grau de reconfiguração. A presente tese descreve uma estratégia de controlo que incorpora Controlo Preditivo com Modelo (MPC), recorrendo a um modelo interno linear, afim de obter reconfiguração na operação de um UAV.

O MPC tem como grande vantagem o facto de permitir a introdução de limites tanto nas entradas, nas saídas e nas variáveis de estado, como também possui uma afinação flexível. Nesta tese descreve-se a implementação de falhas em superfícies de voo bem como o desenvolvimento de dois controladores capazes de superar essas mesmas falhas. Os controladores são sujeitos a testes, nos quais é lhes exigido que executem uma manobra durante um tempo fixo. Finalmente os dados resultantes da acção dos mesmos são analisados .

Palavras-Chave: Aeronaves, Controlo Preditivo, Controlo Reconfigurável, Controlo Tolerante a Falhas, Detecção e Identificação de Falhas.

Contents

List of Abbreviations

Abbreviation	Description	Definition
UAV	Uninhabited Aerial Vehicle	page 5
MPC	Model Predictive Control	page 49
FDI	Fault Detection and Identification	page 73
FDD	Fault Detection and Diagnosis	page 30
ADG	Air Driven Generator	page 2
NTSB	National Transportation Safety Board	page 3
ATC	Air Traffic Controller	page 3
RSEP	Rudder System Enhancement Program	page 4
FTCS	Fault Tolerant Control Systems	page 25
AFTCS	Active Fault Tolerant Control Systems	page 25
PFTCS	Passive Fault Tolerant Control Systems	page 25
MMST	Multiple Models Switching and Tunning	page 28
IMM	Interacting Multiple Models	page 28
PCA	Propulsion Controlled Aircraft	page 28
MRAC	Model Reference Adaptive Control	page 28
ANN	Artificial Neural Network	page 28
SMC	Sliding Mode Control	page 29
CA	Control Allocation	page 29
EA	Eigenstructure Assignment	page 29
SISO	Single-input single-output	page 28
MBPC	Model Based Predictive Control	page 32
GPC	Generalised Predictive Control	page 32
DMC	Dynamic Matrix Control	page 32
SOLO	Sequential Open-Loop Optimizing Control	page 32

Abbreviation	Description	Definition
GUI	Graphical User Interface	page 97

List of Symbols

Symbol	Description
A	*Aspect ratio.*
a	Finite-wing lift curve slope
b	Wing span.
c	Wing cord.
C_M	Wing *pitch* moment coefficient.
$\mathcal{D}$	Drag.
e	Oswald efficiency factor.
g	Gravity acceleration.
F_B	*Aircraft Body*-fixed reference frame
$X_B,\ Y_B,\ Z_B$	Axis about *Aircraft* body frame.
F_E	*Earth* coordinated frame
$X_E,\ Y_E,\ Z_E$	Earth-fixed reference frame axis.
h_F	Height between X_B axis and the vertical stabiliser's pressure centre.
$I_{xx},\ I_{yy},\ I_{zz}$	moment of inertia about X_B, Y_B and Z_B axis
$I_{xy},\ I_{yz},\ I_{xz}$	Product if inertia about O_{xyz} frame.
i_k	Inertia moment functions.
K_e	Engine growth ratio.
l	Lift length ratio.
$\mathcal{L}$	Lift.
m	Aircraft's mass.
$P,\ Q,\ R$	*Roll, Pitch and Yaw* rate.
$U,\ V,\ Z$	Velocity in X_B, Y_B and Z_B.
$\mathcal{V}$	Aircraft airspeed.

Symbol	Description
V_0	Velocity through the Propeller.
$X,\ Y,\ Z$	Force in X_B, Y_B and Z_B axis.
$L,\ M,\ N$	*Roll, Pitch and Yaw* moment.
α	Angle of attack.
α_{L0}	Zero lift angle.
β	*Sideslip* angle.
$\phi,\ \theta,\ \psi$	*Roll, Pitch and Yaw* Euler angle.
Δ	Zero lift angle to control surface defection ratio.
η	Control surface deflection angle.
q	Dynamic pressure.
S	Wing Surface Area.
S_d	Propeller disc area.
T	Propeller Thrust.
T_H	Throttle input.
ρ	Air density.
ε_T	Horizontal Stabiliser incidence angle.
t_s	5% Settling time.
t_s^*	5% Settling time for the fault actuator.
σ_i	Fault activation step signal.
t_j	Actuator failure time.
η_{fault}	Deflection angle on total power loss.
δt	Control sample rate.
$z^*(k)$	Optimal control.
J	Cost functional.
P	Prediction horizon.
M	Control horizon.
w^y	Weight on the states , or outputs.
w^u	Weight on the control input.
$w^{\Delta u}$	Weight on the control input rate.
u_{min}, u_{max}	Constraints on input.
$\Delta u_{min}, \Delta u_{max}$	Constraints on input rate.
y_{min}, y_{max}	Constraints on output.

List of Figures

List of Tables

Chapter 1

Introduction

Aircraft and airline industries are considered to be some of the safest industries. High standards were developed to prevent the huge human and financial losses in case of aircraft accidents. Advances in technologies enabled the development of modern autopilots that contributed to a reduction of the pilots workload and increased airline industry safety. Most autopilot controllers are tested in a wide variety of flight regimes which include robustness to a certain level and several types of failures. However they do not usually include failures involving flight surfaces, because they have no reconfigurable capabilities.

The occurrence of several tragic aircraft accidents and the development of Uninhabited Aerial Vehicles (UAV), contributed to the launch of research in reconfigurable control by defining mechanisms in autopilots capable of addressing the problems like control surfaces failures and power losses. This chapter presents the framework and motivates the work of this thesis.

1.1 Motivation

A significant case happened in 1989 [1], a United Airlines DC-10, Flight 232 enroute from Denver to Minneapolis, suffered a catastrophic failure which cut through hydraulic lines from all three independent systems as shown in figure 1.1, leaving the plane uncontrolled at 11 Km, as reported in [11]. About 1 hour and 7 minutes after takeoff the crew heard a loud bang, followed by vibration and a shuddering of the airframe, the n^{o}2 engine had suffered an uncontained failure. It then appeared that the hydraulic pressure was zero. The DC-10

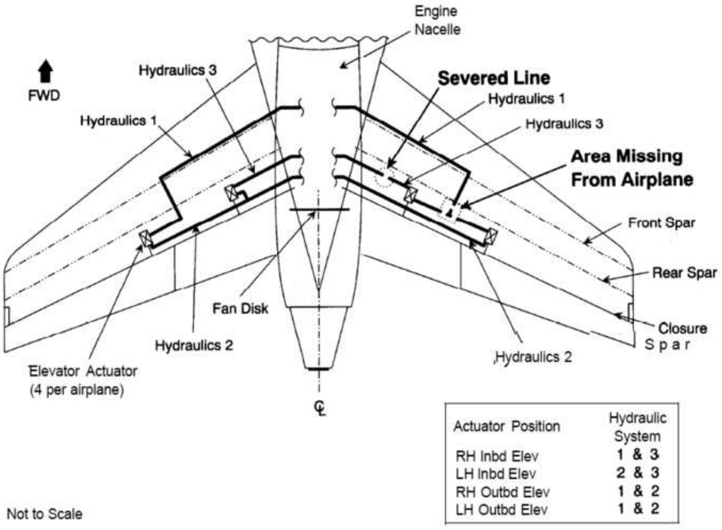


Figure 1.1: Horizontal Stabiliser Hydraulic System Damage [1]

didn't respond anymore to flight control inputs and the descending right turn was arrested by using n^{o}1 engine power reduction. The air driven generator (ADG), which powers the n^{o}1 auxiliary hydraulic pump, was deployed but hydraulic power couldn't be restored. An off-duty training check airman was travelling on Flight 232 and offered his assistance. He was asked to manipulate the throttles to control pitch and roll, which was difficult because the plane had a continuous tendency to turn to the right illustrated in figure 1.2. The n^{o}1 and 3 thrust levers couldn't be used symmetrically, so he used two hands to manipulate the throttles. Sioux City Airport was sighted at 16 Km out, but the aircraft was aligned with the closed runway 22 (2 Km long) instead of the longer (2.7 Km) runway 31. Given the position and the difficulty in making left turns, the approach to runway 22 was continued. The aircraft approached with a high sink rate (30 Km/h for the last 20 seconds) at an airspeed of 400 Km/h. At about 30 m above the ground the nose began to pitch downward and the right wing dropped. The plane touched down on the threshold slightly left of the centreline, skidded to the right and rolled inverted. The DC-10 caught fire and cartwheeled.

In the previous case, consequences could have been worse if a less able crew were to be in command, that what happen with the Boeing 737. In nearly 30 years of commercial service, the 737 has flown more than 60 million flight hours, safety authorities say there is no evidence linking an errant rudder to any of the 64 planes destroyed in accidents, including 24 crashed

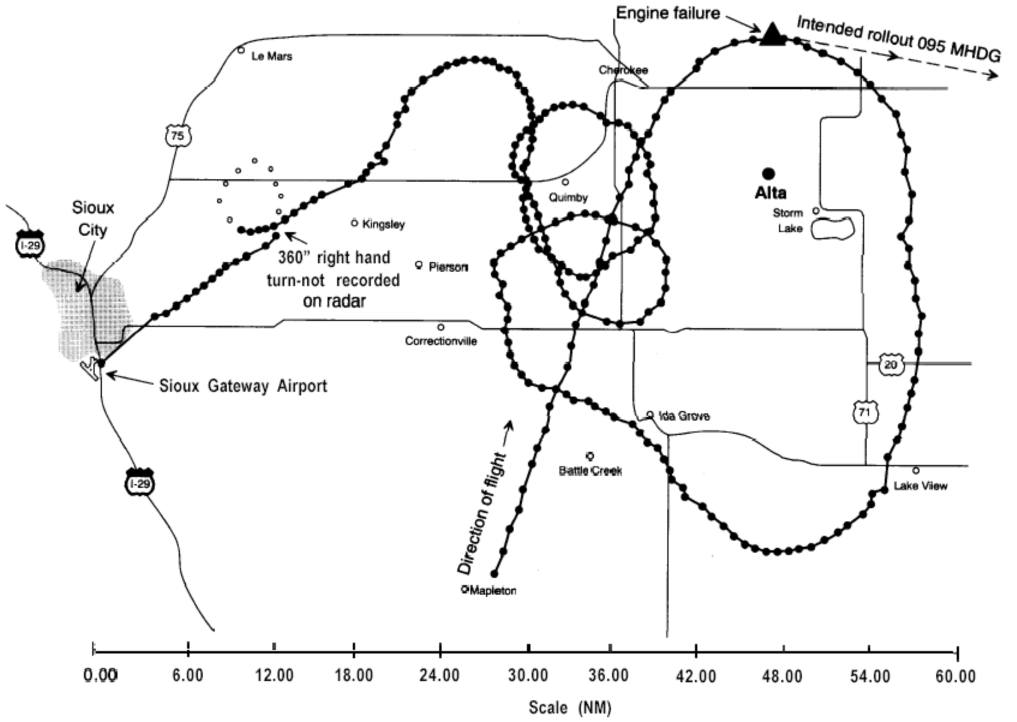


Figure 1.2: Flight 232 Ground Radar Track [11]

in the last decade. However the National Transportation Safety Board (NTSB) has identified the most probable cause of two major accidents on Model 737 series airplanes as a jammed secondary slide in the main rudder power control unit (PCU) servo valve in combination with overtravel of the primary slide (see figure 1.3). One of them was Flight 427 [11], that was approaching Pittsburgh runway 28R when ATC reported traffic in the area, which was confirmed in sight by the first officer. At that moment the aircraft was levelling of at 2 Km (speed 350 Km/h) and rolling out of a 15^o left turn (roll rate 2deg/sec) with flaps at 1, the gear still retracted and autopilot and auto throttle systems engaged. The aircraft then suddenly entered the wake vortex of a Delta Airlines Boeing 727 that preceded it by approximately 69 seconds (6.8 Km). Over the next 3 seconds the aircraft rolled left to approximately 18^o of bank. The autopilot attempted to initiate a roll back to the right as the aircraft went in and out of a wake vortex core, resulting in two loud ”thumps”. The first officer then manually overrode the autopilot without disengaging it by putting in a large right-wheel command at a rate of 150 deg/sec. The airplane started rolling back to the right at a velocity that peaked 36 deg/sec, but the aircraft never reached a wings level attitude. At 610 m above the ground the aircraft's

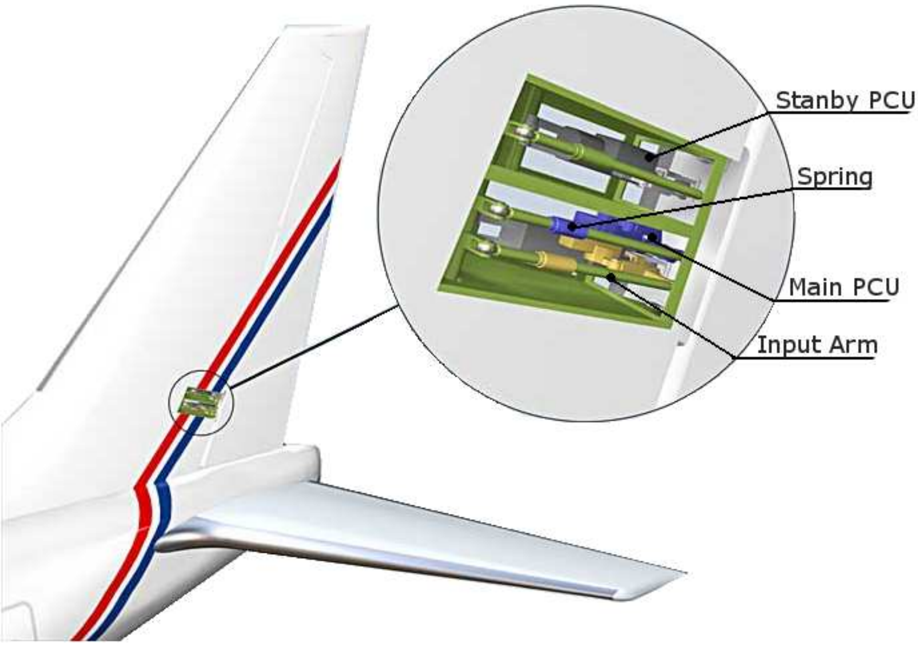


Figure 1.3: Boeing 737 Rudder System Scheme [6]

attitude passed 40^o nose low and 15^o left bank. The left roll hesitated briefly, but continued and the nose again dropped. The plane descended fast and impacted the ground nose first at 480 Km/h in an 80^o nose down, 60^o left bank attitude and with significant sideslip. Figure 1.3 shows Rudder System Enhancement Program (RSEP) introduced in 2003 which should be implemented on all series of 737s by 12 Nov 2008. It replaces the infamous dual concentric servo valve with separate input rods, control valves and actuators. The standby PCU is controlled by a separate input rod and control valve powered by the standby hydraulic system. All three input rods have individual jam override mechanisms that allow inputs to be transferred to the remaining free rods if a jam occurs.

Flight accidents such as these have motivated researchers for decades to try and develop a reconfigurable flight control system that can handle failures such as these. The conventional approach to ensure undiminished performance in case of an actuator failure has been to introduce more and more functionally redundant hardware elements. This approach apart from leading to inefficiencies in the overall system performance due to weight penalty, power consumption and space requirement, has also increased the maintenance burden because more components may fail.

The main motivation for reconfigurable control is the increase in survivability and safety, reduction of maintenance and lowering redundancy. For flight control applications, a reconfigurable controller should allow acceptable control of the aircraft under a variety of damage or fault conditions. An aircraft failure can be divided in three scenarios:

- Possibility to continue flight. This is achievable only if the control strategy allows the aircraft to have performance similarly to that of an undamaged aircraft and control to be restored to a good extent.
- Modification of the original flight path. If damage to aircraft is severe and the performance of the healthy aircraft cannot be fully recovered an alternative route may be possible.
- Abandon flight trajectory and land safely. Under such conditions the reconfigurable strategy should at least provide safe landing capabilities to the pilot.

Despite these benefits, there are currently few aircraft which use a reconfigurable controller because generally the aircraft industry is very conservative and is reluctant to use complex control techniques. Aircraft receptive to use this kind of control strategy are UAV's, where the controller will have full authority over the plane, this is unlikely to be the case in civil aircraft where one or several pilots will be available.

1.2 Objectives

This thesis addresses the problem of simulating an aircraft surface failure and developing a reconfigurable control system by applying Model Predictive Control (MPC) with a linear internal model. Nevertheless to achieve this purpose, it is important to define all the steps of this research:

- Study the aircraft model in order to define, an operation point as also the actuators failures of interest [Ailerons; Elevators; Rudder; Flaps].
- Develop a *Simulink* block capable to introduce the pretended failures in the aircraft models.

- Perform a survey on Reconfigurable Control methods, and choose the most suitable to the present thesis.

- Define a battery of tests, to ensure the controller performance. These tests should prove the controller ability to track reference commands before and after a failure.

- Design controllers. In the end the system must be able to determine a contingency strategy to be applied when the fault scenario is detected.

- Test controllers with a Guidance System, in order to evaluate the aircraft's capability of following trajectories in a failure situation. Study the controlled performance effects if a Fault Detection and Identification system was to be introduced.

- Create a simulation Graphical User Interface (GUI), to facilitate a failure simulation and also data analysis.

- Introduce the ability to visualise the aircraft trajectory and attitude in a flight simulator software (*FlightGear*).

1.3 Thesis Preview

This thesis is organised as described below following the introductory chapter:
Chapter 2 presents the details of the aircraft model used in this research. The model platform, coordinate frames and actuator simulation.
Chapter 3 presents the failure scenario designs and methods. It describes both the type of failures introduced and their actual implementation.
Chapter 4 presents the Reconfigurable Controllers overview, and also takes a closer look to MPC controllers used throughout this research. It describes the controller implementation and Matlab MPC block.
Chapter 5 presents MPC parameter selection, and the results from each test comparing MPC with a *Classic* implementation controller. Chapter 6 summarises the results of this research and offers recommendations for further research.

Chapter 2

Aircraft Model

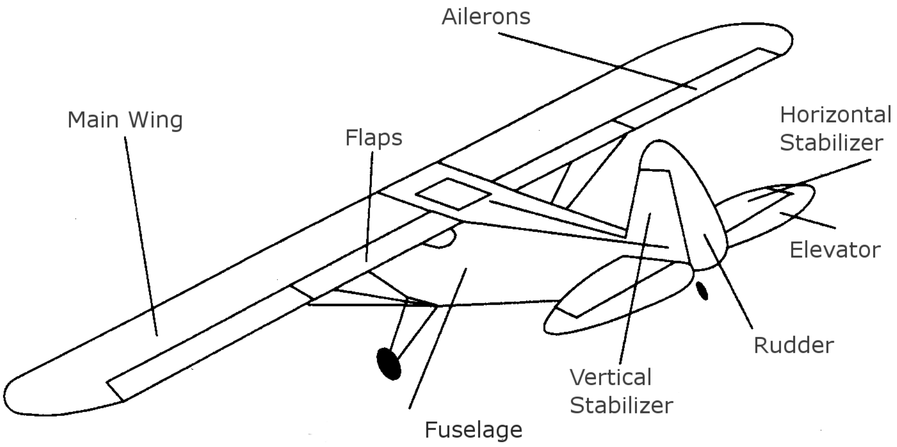


Figure 2.1: *Piper* PA18 *Super Cub* general structure

This thesis is based is the 1/4 scale *Piper* PA 18 *Super Cub* aircraft equipped with 50 cc engine, a model developed in [30]. The basic technical characteristics of this scale airplane are:

- Speed: min.70 Km/h, max.120Km/h
- Engine: Quadra 50 cc
- Power: 3.4 KW

Dimensions and weight:

- Wing span: 2.7 m
- Overall lenght: 1.72 m
- Weight: 10.5 Kg

This aircraft has a total of seven aerodynamic surfaces for control, six of them arranged as three pairs. The control surfaces that induce the largest moments in the principal axis are:

- Elevators (Horizontal tails) - Engaged in equally deflection for longitudinal maneuvers.
- Ailerons - Engaged symmetrically for both heading changes and roll maneuvers.
- Rudder - Used to induce Yaw moments.

The remaining pair are Flaps used for lift gains both on take-off and landing. This aircraft model is decomposed into its major subsystems : the actuator model, the propulsion systems, the aircraft's aerodynamics, gravitical model, and the six degree of freedom full nonlinear differential equations. The model's major assumptions are as follows:

- Aircraft body is rigid
- Earth is an inertial reference frame
- Aircraft mass is constant

2.1 Reference Frames

The aircraft motion is expressed by defining the following frames as in [30]:

- Earth-fixed frame F_E : This reference frame, also called the topodetic frame, is a right-handed orthogonal system which is considered to be fixed in space. Its origin will be chosen to coincide with the aircraft's center of gravity at the start of a flight test manoeuvre. The Z_E-axis points downwards, parallel to the local direction of gravity. The X_E-axis is directed to the North, the Y_E-axis to the East.
- Body-fixed reference frame F_B : This is a right-handed orthogonal reference system which has its origin, O_B at the centre of the gravity of the aircraft. The $X_BO_BZ_B$ plane coincides with the aircraft's plane of symmetry. The X_B-axis is directed towards the nose

of the aircraft, the Y_B-axis points to the right wing , and the Z_B-axis points towards the bottom of the aircraft.

Initially the two reference frames are considered to be in the same place. For simulation purposes there is a difference in the Z-axis. The major attitude tracking variables are then defined from the difference between the body frame and the earth-fixed reference. The ordering of the rotations is done as follows:

- ϕ Rotation about $O_E X_E$ or *roll* angle
- θ Rotation about $O_E Y_E$ or *pitch* angle.
- ψ Rotation about $O_E Z_E$ or *yaw* angle.

This can be seen in the following figures 2.2. Throughout simulations the aircraft trajectory has to be analysed in earth-based coordinates, to obtain that a transformation from the body frame to the earth-fixed frame is used. Given that $R_{BE} = R_{EB}^{-1}$ is the coordinate transform from F_B to F_E. R_{BE} is defined as

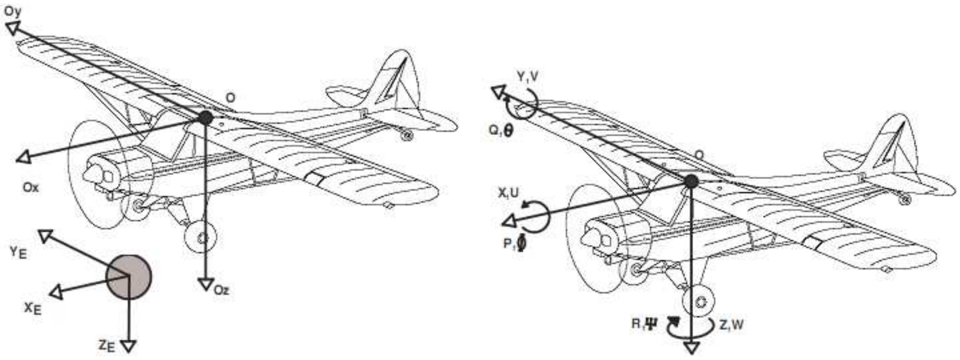


Figure 2.2: Earth and Body frames and Positive Reference for Angular and Linear Velocities

$$R_{BE} = \begin{bmatrix} \cos(\psi)\cos(\theta) & \sin(\psi)\cos(\theta) & -\sin(\theta) \\ \cos(\psi)\sin(\theta)\sin(\phi) - \sin(\psi)\cos(\phi) & \sin(\psi)\sin(\theta)\sin(\phi) + \cos(\psi)\cos(\phi) & \cos(\theta)\sin(\phi) \\ \cos(\psi)\sin(\theta)\sin(\phi) + \sin(\psi)\sin(\phi) & \sin(\psi)\sin(\theta)\cos(\phi) - \cos(\psi)\sin(\phi) & \cos(\theta)\cos(\phi) \end{bmatrix} \tag{2.1}$$

Normally the aircraft flight trajectory doesn't match the aircraft's longitudinal axis X_B, so must use the following angles as shown in figure 2.3 defined as:

- α - angle of attack.

- β - sideslip angle.

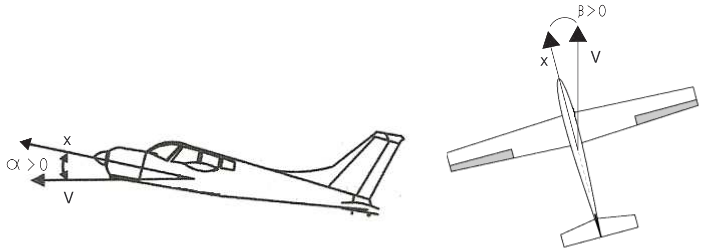


Figure 2.3: *Attack* and *sideslip* angles.

2.2 General Equations of Motion

The equations that determine the relation between the state variables given the major three subsystems for the 6-DOF model are: Force equations that provide us the linear velocities:

$$\begin{aligned}\dot{U} &= RV - QW + \frac{1}{m}X \\ \dot{V} &= PW - RU + \frac{1}{m}Y \\ \dot{W} &= QU - PV + \frac{1}{m}Z\end{aligned} \tag{2.2}$$

where X,Y and Z are the force components in Ox, Oy and Oz, the velocities U,V and W are defined in the same axis. From the moment equations we obtain the angular velocities:

$$\begin{aligned}\dot{P} &= i_1 PQ + i_2 QR + i_3 L + i_4 N \\ \dot{Q} &= i_5 PR + i_6 (R^2 - P^2) + i_7 M \\ \dot{R} &= i_8 PQ + i_9 QR + i_{10} L + i_{11} N\end{aligned} \tag{2.3}$$

where N, M and L are the torque components acting on the center of gravity of the aircraft. The constants i_k are functions of the moment of inertia of the aircraft defined in appendix A. Now it's possible to obtain the euler angles:

$$\begin{aligned}\dot{\phi} &= P + R\tan(\theta)\cos(\phi) + Q\tan(\theta)\sin(\phi) \\ \dot{\theta} &= Q\cos(\phi) - R\sin(\phi) \\ \dot{\psi} &= R\frac{\cos(\phi)}{\cos(\theta)} + Q\frac{\sin(\phi)}{\cos(\theta)}\end{aligned} \tag{2.4}$$

Three other variables are used to best describe the aircraft motion, the true velocity $\mathcal{V}$, sideslip angle β and angle of attack α given by:

$$\begin{aligned}\mathcal{V} &= \sqrt{U^2+V^2+W^2}\\ \alpha &= \frac{W}{U}\\ \beta &= \frac{V}{U}\end{aligned} \tag{2.5}$$

The handling of the equations and the simulations are made easier decoupling the longitudinal motion from lateral-directional motion, nevertheless it requires certain simplifications that reduce the accuracy of the model[31, 8]. The model used for simulation purposes in thesis isn't based on decoupled equations and uses the complete set of nonlinear equations, so the results obtained should represent the reality with a significant accuracy. This model is available as a *Simulink* file, produced by *Rato L.M. e Neves da Silva* in their master thesis [30].

2.3 Model Input and Outputs

2.3.1 Actuators (Inputs)

These forces and moments described in the previous section are functions of the control surfaces, thrust and drag. The *Simulink* model inputs are deflection angles of the elevators, ailerons, rudder, flaps and throttle. The following figure shows the control surfaces positive orientation.

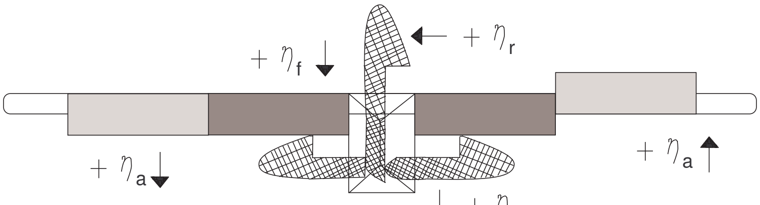


Figure 2.4: Control surfaces positive deflections

A table featuring a complete list of the model's inputs can be seen below. The model of the aircraft's engine is of considerable complexity and throttle is not relevant to the objectives of this work, so this model won't be studied. The dynamic of the hydraulic or as the present aircraft electric actuator used to deflect the control surfaces present nonlinear behaviours

Inputs	Description	Max	Min	Model units
η_e	*Elevator deflection*	+17.5°	-17.5°	*rad*
η_a	*Aileron deflection*	+20°	-15°	*rad*
η_r	*Rudder deflection*	+30°	-30°	*rad*
η_f	*Flaps deflection*	45°	0°	*rad*
T_H	*Throttle*	100	0	(%)

Table 2.1: Aircraft Model Inputs

and dead zones. The mechanical charge that the actuator has to hold up is nonlinear due to friction and also aerodynamic forces. Friction is one of the greatest obstacles in high precision positioning systems, it can cause steady state and tracking errors, while it may result in limit cycles, for these reasons the actuator is a servomechanism. Nevertheless for controller designing purposes, it is common to approximate the actuators as first order systems ($t_s = 3 \times$ *time constant*), in the present case we have the following systems:

- Elevator: $\frac{\eta_e(s)}{\eta_e^*(s)} = \frac{60}{s+60}$
- Aileron: $\frac{\eta_a(s)}{\eta_a^*(s)} = \frac{60}{s+60}$
- Rudder: $\frac{\eta_r(s)}{\eta_r^*(s)} = \frac{60}{s+60}$
- Flap: $\frac{\eta_f(s)}{\eta_f^*(s)} = \frac{60}{s+60}$

In *Simulink* environment the actuator block is defined as standard state-space format:

$$\dot{\underline{\eta}} = A_{act}\underline{\eta} + B_{act}\underline{\eta}^* \tag{2.6}$$

$$\underline{\eta} = \begin{bmatrix} \eta_a & \eta_e & \eta_r & \eta_f \end{bmatrix}' \tag{2.7}$$

$$\underline{\eta}^* = \begin{bmatrix} \eta_a^* & \eta_e^* & \eta_r^* & \eta_f^* \end{bmatrix}' \tag{2.8}$$

2.3.2 Sensors (Outputs)

In order to determine its position, orientation and velocity, the aircraft possesses a set of inertial measurement devices. Gyroscopes are used to measure the angular velocity of the system in the inertial reference frame. Linear accelerometers measure how the vehicle is moving in space(there is a linear accelerometer for each axis). With the information provided by the sensors, a computer can integrate and calculate the angular and linear velocities.

The most common way to obtain the aircraft velocity is using a Pitot tube, that is based on the *Bernoulli* law, this tube has both lateral an frontal openings, to be able to measure static and total air pressure (static pressure + dynamic pressure) [33]. Another important sensor is the altimeter, which measures the atmospheric pressure from static port outside the aircraft. When the angle of attack of an aircraft is to be measured, a sensor must be physically used to take the readings while in flight. The sensor is usually located ahead of the aircraft, on the fuselage nose, or on a wing tip. These sensing devices may be the pivoted vane, the differential-pressure tube, and the null seeking pressure tube. There are a number of devices which can be used to measure sideslip angle, some use optical methods, some use inertial methods, some GPS and some both GPS and inertial. The *Simulink* model calculate all these variables as described previously, a table featuring a complete list of the simulation outputs can be seen below.

Output	Description	Model units
U	*Velocity in the x body axis*	*m/s*
V	*Velocity in the y body axis*	*m/s*
W	*Velocity in the z body axis*	*m/s*
P	*Roll rate*	*rad/s*
Q	*Pitch rate*	*rad/s*
R	*Yaw rate*	*rad/s*
ϕ	*Euler roll angle*	*rad*
θ	*Euler pitch angle*	*rad*
ψ	*Euler yaw angle*	*rad*
α	*Attack angle*	*rad*
β	*Sideslip angle*	*rad*

Table 2.2: Aircraft Model Outputs

2.4 Model Linearization

Using the previous equations of motion a nonlinear model is obtained for our aircraft. If we assume that the motion of the aeroplane consists of small deviations from a reference condition of steady flight, then a linearized model around a trim condition can be obtained. To compute a linear model for small perturbations around the trim or steady-state condition, the first step and also the most difficult is to replace all nonlinearities in the general dynamic equations with their first order Taylor series approximations [35, 28]. Another approach is to use an identification algorithm using data collected either from a nonlinear model or from a physical airframe. The method used in this present thesis was to compute numerically the effect of small changes in the state variables and inputs on the state derivatives from the linerization point. As the nonlinear model is available in *Simulink*, the linearization model is derived using the *MATLAB* linearization function *LINMOD*. The Steady flight was defined at the following equilibrium point:

- Aileron deflection η_a: 0.0 rad
- Elevator deflection η_e: -0.0285 rad
- Rudder deflection η_r: 0.0 rad
- Flaps deflection η_f: 0.0 rad
- Throttle: 6.24%
- Velocity: 21.156 m/s
- *Pitch* θ: 5.6×10^{-5} rad
- x_0=[U = 21.156 V = 0 W = 1.25×10^{-3} P = 0 Q = 9.38×10^{-11} R = 0 $\phi = 0$ $\theta = 5.6 \times 10^{-5}$ ψ =0 $\alpha = 0$ $\beta = 0$]

The result of the linearization is that we can write the model in a standard state-space format given by the following equation:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned} \tag{2.9}$$

Now it is possible to calculate the system poles through the A matrix eigenvalues, resulting in these poles (see figure 2.5):

$$\begin{aligned} \text{Poles} = [0,\ &-27.41,\ -0.54 \pm \jmath 1.72,\ -0.01,\ -5.21 \pm \jmath 6.11 \\ &-11.26,\ -0.04 \pm \jmath 0.52,\ -60,\ -60,\ -60,\ -60]\ \text{rad/s} \end{aligned} \tag{2.10}$$

The last four poles in -60 represent the actuator dynamics, this value doesn't match the first order models presented in [30], this faster dynamic was chosen for controlling purposes in

[7]. The -11.26 pole stands for the propulsive system dynamics, the rest correspond to the longitudinal and lateral motion modes. In the longitudinal motion modes we have $-5.21 \pm j6.11$ and $-0.04 \pm j0.52$ poles, the first correspond to *short-period* mode that has a bigger bandwidth as it a faster oscillation, the remaning one match with *long-period* also known as *phugoid*. Representing the lateral motion we have *Dutch-roll* mode is also a fast oscillatory mode so its poles are place at $-0.54 \pm j1.72$.

Finally the non oscillatory modes are *roll* and *spiral* with corresponding poles at -27.41 and -0.01. The first pole is 0 that match the Euler yaw angle ψ.

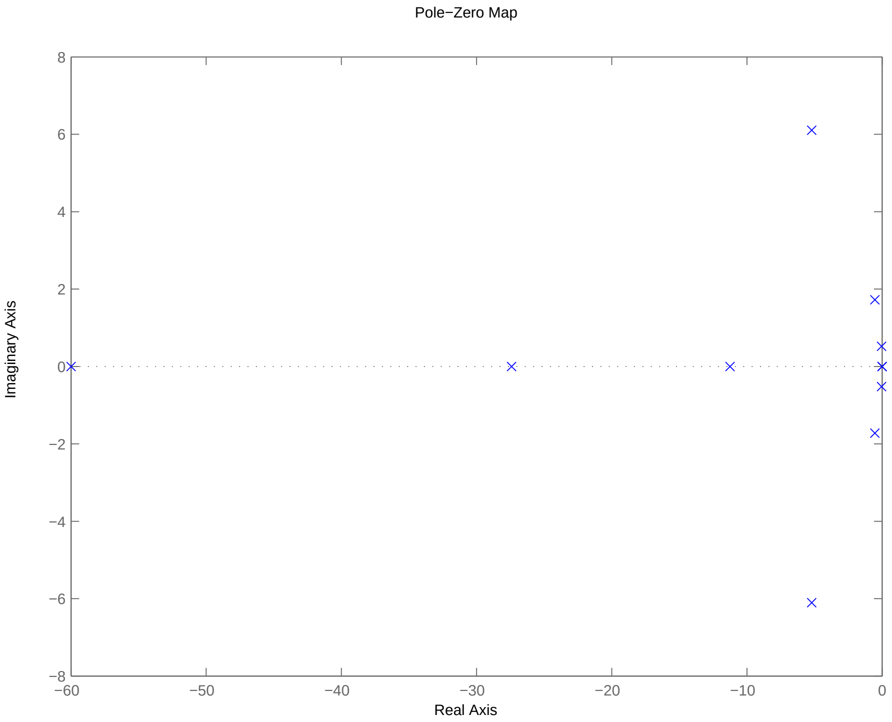


Figure 2.5: Aircraft Model Poles

2.4.1 Flying Qualities

The term *'flying qualities'* is defined as those qualities or characteristics of an aircraft which govern the ease and precision with which a pilot is able to perform the tasks required in support of an aircraft role. Performance of these tasks implies control of the aircraft states: airspeed, altitude, Euler angles, etc.

Where possible, the requirements have been stated in terms of three values of the stability or

control parameter being specified. Each is a minimum condition to meet one of three Levels of acceptability related to the ability to complete the operational missions for which the airplane is designed.
Firstly the aircraft should be classified by its weight, so according to the ($MIL151$ in [28]) requirements the 1/4 scale *Piper* PA18 *Super Cub* is *Class I*.

Longitudinal Flying Qualities

Longitudinal modes are generally expressed by their damping ratio.

- *Phugoid Mode* - $\zeta_{Ph} = 0.0668 > 0.04$ its classified as *Level I*.
- *Short Period* - $0.3 < \zeta_{SP} = 0.6489 < 1.3$ its classified as *Level I* for all flight categories.
- To ensure separate modes $w_{Ph}/w_{SP} < 0.1$, for this aircraft $w_{Ph}/w_{SP} = 0.0653$.

Lateral Flying Qualities

Lateral qualities are also analysed in terms of damping ratio and also through time constants.

- *Spiral Mode* - From the time to double amplitude $t_{double} = -\frac{ln(2)}{p} = 80.64 > 12s$ so it is classified as *Level I*.
- *Dutch Roll* - $\zeta_{Dr} = 0.2985 > 0.19$, $w_{Dr} = 1.8033 > 1\ rad/s$ and $\zeta_{Dr}.w_{Dr} = 0.5383 > 0.35$ its classified as *Level I* for all flight categories.
- *Roll Mode* - From the time constant $T = -\frac{1}{p} = 0.0365 < 1s$ its classified as *Level I* for all flight categories.

Now its possible to conclude this aircraft has good flight qualities for all mission flight phases.

Chapter 3

Fault Scenario

A fault is understood as any kind of malfunction in the actual dynamic system that leads to an unacceptable anomaly in the overall system performance. In an aircraft such malfunctions may occur either in external hardware like sensors (instruments faults), actuator (actuator fault) or structural damage. In this work, the interest has been to detect and resolve faults in control surfaces effectiveness and actuator.

3.1 Failures Types

There are three types of faults encountered in a system that can originate failures, given the three parts in which a system can be split in [15] :
Sensor faults:

- Sensor loss: which can be viewed as serious measurements variations.

Actuators faults:

- Partial hydraulic/power loss: Maximum rate decrease on several control surfaces.
- Full hydraulics/power loss: One or more control surfaces become stuck at last position for hydraulic driven aircraft, or float on light aircraft.
- Control loss (internal fault): One or more control surfaces become stuck at last position.

This kind of faults can also be accompanied by structural damage, in that case we have:

- Loss of part/all of control surface: Effectiveness of control surface is reduced, but rate is not; minor change in the aerodynamics.

- Loss of engine: Large change in possible operating region; significant change in the aerodynamics.

Finally *Structural Faults*:

- Damage to aircraft surface: Possible change in operating region, and significant change in aerodynamics.

Some failures can induce changes in the aircraft dynamics and aerodynamic behaviour, requiring for that deeper modifications on the *Simulink* model, so the only type of failures studied in this thesis were partial and full power loss.

3.2 Fault Simulation

The 1/4 scale *Piper* PA18 *Super Cub* is equipped with standard RC modeling servos. The term '*servo*', is short for servomechanism or servomotor, defined as a relatively small, low-powered device to control a much larger force or momentum [30]. In the present situation, servos were small electrically-driven motors used to control the aircraft's aerodynamic control surfaces, which in turn created much larger aerodynamic forces and moments [27]. These systems are made from an electric motor engaged mechanically to a variable resistor also known as *Rheostat* or Potentiometer. Pulse-width modulation (PWM) signals sent to the servo are translated into

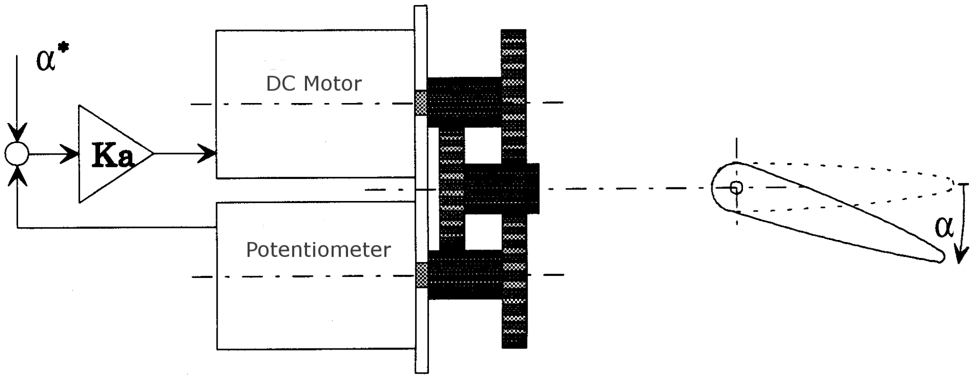


Figure 3.1: Actuator Servomechanism

position commands by electronics inside the servo. When the servo is commanded to rotate, the DC motor is powered until the potentiometer reaches the value corresponding to the commanded angle α^*.

In order to have a fast response but also taking into account noise effects a gain K_a is used in the feedback control chain. This makes the amplifier to saturate everytime the angle error $(\alpha^* - \alpha)$ overcome $\frac{U_{sat}}{K_a}$, as K_a is a high value a small error causes saturation.

3.2.1 Partial Power Loss

The model used in this thesis uses all control surfaces with the same characteristics as mentioned before. The actuators are represent by the following first order model $G(s) = \frac{60}{s+60}$, having all a *settling* time $(t_s = 0.05s)$.

A partial power loss causes a change in the actuator dynamics, to be more specific on his maximum rate. So this can be simulated, increasing the settling time to a maximum of 4 seconds. There is no interest in higher values because they have similar effects as a full power loss. In order to introduce this fault into the actuator state of spaces: $\{\dot{\eta}(t) = A_{act}\eta(t) + B_{act}\eta^*(t)\}$ the following formulation is used:

$$A_{act} = -B_{act} = \begin{cases} d_{i,j} = a_i(t) & \text{if} \quad i = j, \\ d_{i,j} = 0 & \text{if} \quad i \neq j \quad \forall i,j \in \{0, ..., 4\} \end{cases} \tag{3.1}$$

In the presence of actuator partial loss of power, $a(t)$ can be expressed as

$$a_i(t) = -60 + \sigma_i(t)\left(60 + \frac{3}{t_s^*}\right) \tag{3.2}$$

$$\sigma(t) = [\sigma_1(t), \sigma_2(t), \sigma_3(t), \sigma_4(t)] \tag{3.3}$$

$$\sigma_i(t) = \begin{cases} 1 & \text{if the } i\text{th actuator fails at } t_j \\ 0 & \text{otherwise} \end{cases} \tag{3.4}$$

Where the t_s^* is the settling time for the fault actuator, for simulation purposes σ_i is a step signal, reaching 1 at t_j actuator failure time.

In figure 3.2 it is possible to notice the change on the actuator dynamics. The healthy actuator responds very rapidly as expected with a *settling* time of 0.05 seconds, in other hand, the damage one has a failure that induces a $t_s = 1s$. Due to this fact the step response of the fault actuator reaches 5% of the steady value 10^o after 1 second.

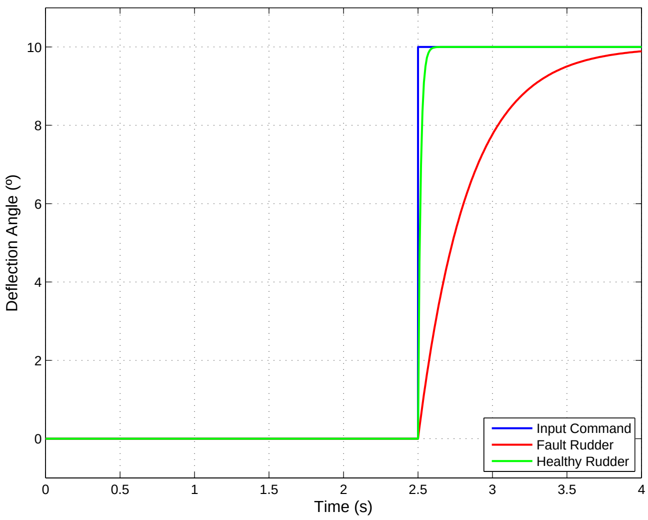


Figure 3.2: Rudder Partial Power Loss Demonstration

3.2.2 Full Power Loss

For instance, while engaged in a steady climb the entire elevator both right and left fails and becomes stuck at a negative deflection (positive Pitch moment), there is no other control effector to be used to recover from this kind of failure.

For this reason each elevator and ailerons pairs were mathematically divided into two. This was accomplished in *Simulink* by splitting the actuator signal and then recombining it again before sending it to the aerodynamic model. In figure 3.3, the command signal for elevator and

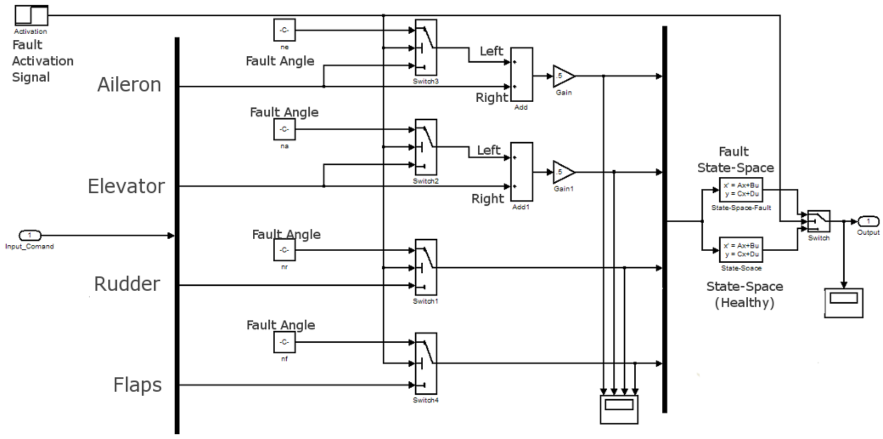


Figure 3.3: *Simulink* actuator fault system

ailerons are split in two different signals. For all intentional purposes, these two signals became two separate control surfaces that can fail independently. The figure shows also that they are recombined and multiplied by a gain of 0.5 before being sent to the aerodynamic model. For simulation purposes only, the value used to define a specific fault deflection is related with one of the pair control surfaces in both aileron and elevator, since the model only receives one value for these surfaces this strategy neglects all coupling effects that would arise if these failure were to happen in a actual aircraft. In the case of a full power loss, the control surface is considered to be stuck at fixed angle, the actuation command signals are then overridden by the desired deflection and $\eta(t)$ can be expressed as:

$$\eta_i(t) = \eta_i^*(t) + \frac{1}{2}\sigma_i(t)\left(\eta_{fault} - \eta_i^*(t)\right) \quad i = 1, 2 \tag{3.5}$$

$$\eta_i(t) = \eta_i^*(t) + \sigma_i(t)\left(\eta_{fault} - \eta_i^*(t)\right) \quad i = 3, 4 \tag{3.6}$$

where σ_i signal is already defined in equation 3.3, $\eta^*(t)$ is the command signal, and η_{fault} is the position or deflection angle which the control surfaces remains when a total power loss is simulated. Finally $i = 1, 2$ stands for aileron and elevator command signals and $i = 3, 4$ for rudder and flaps.

There are three types of failures explored in this situation. The first is the *hard over* failure in which one of the surfaces proceeds to its maximal positive limit. The second type of failure is the *frozen or stuck* surface, as would be expected the surface simply stays locked in its position at the point of failure. The final failure is the *hard under* failure, in this scenario the surface proceeds to its maximal negative deflection. The moment this is triggered its determined by the operator, an example of this can be seen in figure 3.4. In this figure a sinusoidal signal with amplitude 1 is the commanded to the control surface. At 2.5 seconds a Hard Over failure occurs driving the position to its maximum limit of 30° as it is the Rudder. This limit is achieved with a *settling time* of 0.05 seconds. No loose situation was simulated, what would correspond to a free floating surface.

3.3 Contingency Strategy

It is common sense that actuator redundancy is needed for actuator failure compensation. There are two different kinds of actuator redundancy. The first situation is that actuators have similar physical characteristics, for example, they are segments of a multiple-segment rudder or

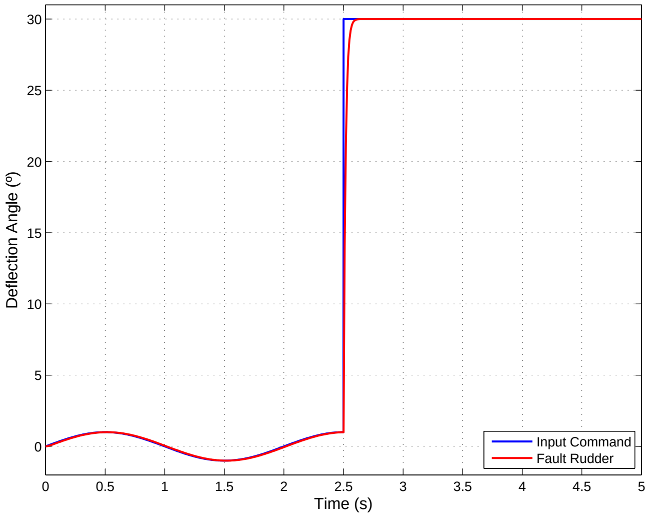


Figure 3.4: Rudder Hard Over Demonstration

elevator of an aircraft. The second type is when actuator have different physical characteristics and control effects, for example, differential engine thrust can be used to compensate for a failed rudder. Because of this, it is useful to study the effects of actuators perturbations in the aircraft general dynamic. The following table 3.1 was obtained from the nominal equilibrium point and then deflecting each surface by 1° or increasing the engine power by 10% during 5 seconds. From the data it is possible to obtain the actuator effects in the aircraft motion, and reach the

Eq. Point		Perturbations at:					
			$\delta_{\eta a}$	$\delta_{\eta e}$	$\delta_{\eta r}$	$\delta_{\eta f}$	δ_{TH}
U_0	21.1558	U	21.5816	24.3260	21.2903	20.3179	22.2998
V_0	0	V	0.9439	0	0.4026	0	0
W_0	0.0012	W	-0.0535	-0.3602	-0.0070	0.0553	-0.0678
P_0	0	P	0.0645	0	-0.0152	0	0
Q_0	0	Q	0.0267	0.0248	0.0048	-0.0065	0.0398
R_0	0	R	0.1217	0	-0.0368	0	0
ϕ_0	0	ϕ	0.3106	0	-0.0713	0	0
θ_0	0.0001	θ	-0.0191	-0.0847	0.0208	0.0535	0.2025
ψ_0	0	ψ	0.2782	0	-0.1146	0	0

Table 3.1: Dynamic Behaviour after Actuator Perturbation

most adequate redundancy scheme for each type of failure:

- *Aileron failure* - from all actuators, the ailerons failures are the ones that can induce the most significant changes in all model outputs. The larger variations as expected are in yaw angle ψ and roll angle ϕ, another affected variable is the longitudinal velocity U. To overcome this failure, there are two possibilities use the other aileron if available or the rudder since this control surface can induce similar moments.

- *Elevator failure* - causes great change in longitudinal velocity U and also in pitch angle θ, the contingency strategy may pass through the use of the healthy elevator or if both are inoperable, it is possible to use *Flaps*, but this measure will only solve positive deflections of the elevator. For the opposite case there's no other actuator capable of induce a negative pitch moment.
- *Rudder failure* - this surface as mentioned before has a most similar effect as the Ailerons. Although in bigger aircraft is common the use of differential thrust to overcome this fault, in this aircraft rudder is single-segment, so in the event of a failure the contingency strategy has only the ailerons to compensate the forces induced by the damaged rudder.
- *Flaps failure* - Perturbations in this actuator are the less significant, affecting only the pitch angle θ, this is because flaps main purpose is to increase lift, being used only for landing and take-off. To solve this failure a small actuation in the elevators is sufficient.

Chapter 4

Reconfigurable Control

The initial motivation for the research on reconfigurable Fault Tolerant Control Systems (FTCS) was avionics and flight control systems with the purpose to improve the reliability and safety on aircraft. This was activated because of American Airlines DC-10 crash [2]. By definition, Fault Tolerant Control Systems (FTCS) are control systems that posses the ability to accommodate system component failures *automatically*.

They are capable of maintaining overall system stability and acceptable performance in the event of such failures. FTCS were also known as self-repairing, reconfigurable, restructurable or self-designing control systems [37, 34].

4.1 Controllers Overview

In a simple way, FTCS can be divided into two main areas, passive (PFTCS) and active (AFTCS) [16]. In the first case, controllers are designed to be robust to a class of known faults. The basic strategy is to make the closed-loop system robust against uncertainties and some restrictive faults, because of that there's no need for FDI schemes nor controller reconfiguration, penalising the fault-tolerant capability.

In the other hand AFTCS respond to the system component failures actively by reconfiguring control actions achieving stability and acceptable performance of the entire system.

AFTCS schemes rely on a real-time FDI system to provide the most recent information about the plant. A critical aspect in a AFTCS is the limited amount of time available for the FDI and for the system reconfiguration [29]. The general structure of a typical AFTCS is shown in figure 4.1, and consists of four major components:

- An on-line and real-time FDI (dashed because its assumed to be implement)
- A reconfigurator mechanism
- A reconfigurable controller
- A command governor(Guidance System)

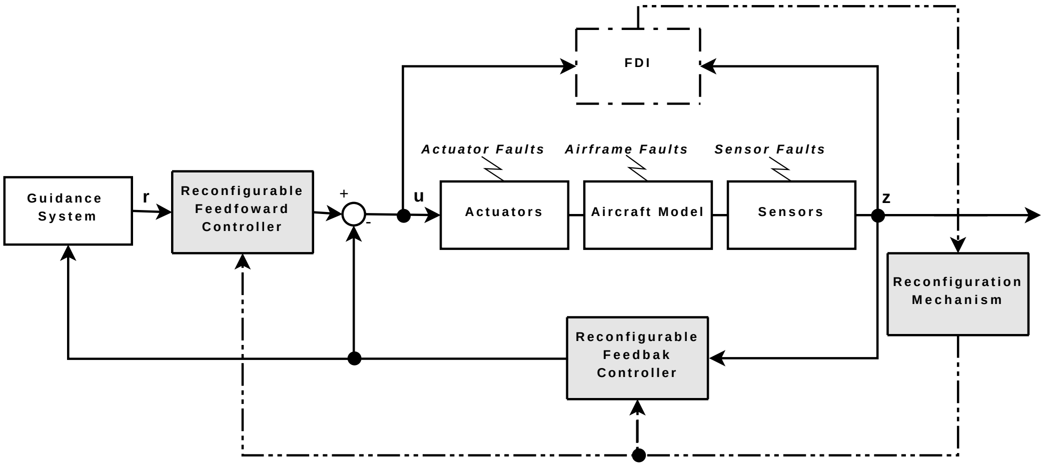


Figure 4.1: Overall FTCS structure

On-line FDI systems for all kinds of faults need to be reliable and decide in time the activation of the reconfiguration mechanism. Using this information provided by the FDI module, the reconfigurable controller should be redesigned automatically to maintain the stability and to recover the performance of the system to its standard level if possible, also important is the capability for the controller to achieve command reference tracking. In order to avoid actuator saturation or performance degradation, a command governor may need to be used to adjust command input or reference trajectory automatically, if possible a different approach should be used to provide advisory information to human operator in the event of fault [32, 4]. In this thesis we care about the Flight Reconfigurable Control, in this case some challenges arise [15]:

- It is a multivariable problem, with strong cross couplings between modes usually appearing after failures. An aircraft loses its symmetry after surface damage (i.e. damage to the body, wings or movable surfaces) and conventional simplified separated longitudinal and lateral direction control approaches may not be applicable.

- It is a non-linear problem, which means that the trim values and the linearization state changes after failures, requiring methods to find the new trim values, this suggests the use of a non-linear such as adaptive control algorithms.

- The system may be highly unstable, leaving very little time for reconfiguration. This demands a very efficient FDI scheme.

- Actuator authority is limited. After an aircraft has sustained damage to a surface, its ability to produce the required control forces degrades and the controller's demands on the surface and therefore on the actuator deflections and deflection rates increase.

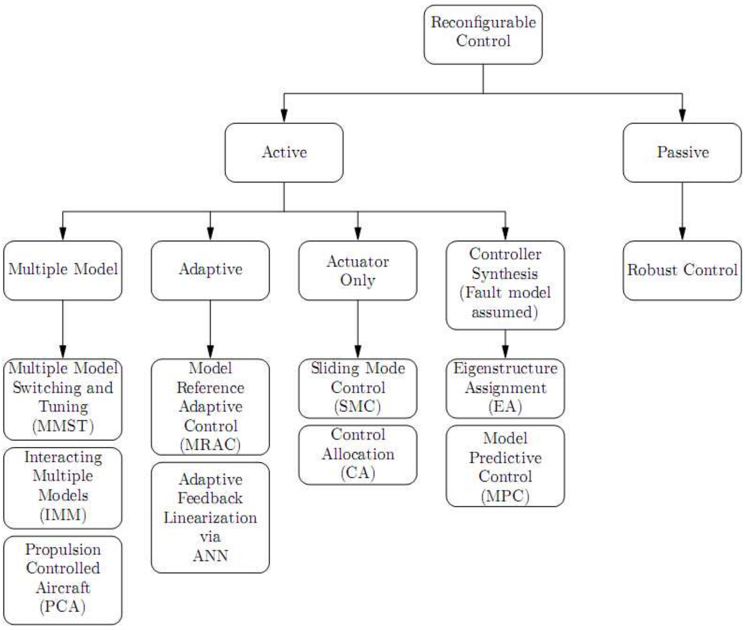

Reconfigurable Control
Active
Passive
Multiple Model
Adaptive
Actuator Only
Controller Synthesis (Fault model assumed)
Robust Control
Multiple Model Switching and Tuning (MMST)
Interacting Multiple Models (IMM)
Propulsion Controlled Aircraft (PCA)
Model Reference Adaptive Control (MRAC)
Adaptive Feedback Linearization via ANN
Sliding Mode Control (SMC)
Control Allocation (CA)
Eigenstructure Assignment (EA)
Model Predictive Control (MPC)

Figure 4.2: Reconfigurable Control Overview[15]

Reconfigurable flight control is for the most part still an academic notion. Although there have been very few controllers implemented on physical systems and none on commercial aircraft, over the last 20 years several research programs have been formed to investigate their potential and as result there are a variety of active methods. The figure 4.2 sums up the most important approaches in the present times [3, 19, 17].

4.1.1 Active Approaches

AFTCS approaches can be arranged into the following four categories:

Multiple Models

- *Multiple Models Switching and Tuning* (MMST) - When a failure occurs, MMST switches to the pre-computed control law corresponding to the failure situation. The difficulty with this approach how to choose the controller pair to switch to at each time instant.

- *Interacting Multiple Models* (IMM) - In IMM rather than using the model which is closest to the current failure scenario as in MMST, IMM computes a fault model as a convex combination of all pre-computed faults models and then uses this new model to perform control.

- *Propulsion Controlled Aircraft* (PCA) - is a specific instance of multi-model approach where the fault model is identical to the nominal one, however in which all control surfaces are free floating.

Adaptive

- *Model Reference Adaptive Control* (MRAC) - The objective of this method is to force the plant output to track a reference model. Primarily a model structure must be assumed, secondly adaptive control requires that the system parameters change slowly enough for the estimation algorithm to track them. However faults may cause abrupt and drastic changes in the states moving the system instantaneously to a new region of the state space. So adaptive control may well be an important part of a reconfigurable algorithm, but can't handle the general problem by itself.

- *Adaptive Feedback Linearization* via ANN - This approach splits the dynamics of the plant into three single-input-single-output (SISO) subsystems for roll, pitch, and yaw, each one with a model reference adaptive controller. The combination with artificial neural networks has been tested successfully, being able to cancel inversion errors.

Actuator Only

- *Sliding Mode Control* (SMC) - SMC is only applicable for failures which cause a loss of effectiveness of the control surface, excluding the floating or jammed surface failure scenarios these results from the assumption that the input function is square and invertible, forcing the existence of one and only surface for every controller variable. The proposed controller is set up in a two-loop, an outer-loop responsible for roll, pitch and yaw that provides angular rate commands to the inner-loop that subsequently uses the actuators. A key aspect is the online adaptation of the boundary layer that can handle partial loss of actuator surfaces, while avoiding limits and integrator windup by reducing the tracking performance.

- *Control Allocation* (CA) - This is a technique that produces the desired set of forces and moments on an aircraft from a set of actuators. The output of the control law can be a set of desired forces and moments and the job of the allocator is to select appropriate actuator positions which will achieve the desired results.

Controller Synthesis

- *Eigenstructure Assignment* (EA) - The concept behind this controller is the use of state feedback to place the eigenvalues of a linear system then uses the remaining degrees of freedom to align the eigenvectors as accurately as possible.

- *Model Predictive Control* (MPC) - Model predictive control has been proposed as a method for reconfiguration due to its ability to handle constraints and changing model dynamics systematically. As already demonstrated MPC can handle jammed actuators without the need to explicitly model the failure. Failures can also be handled in a natural fashion by changing the internal model used to make prediction in either an adaptive fashion, a multi-model switching scheme or by assuming a FDI scheme that provides a fault model.

Table 4.1 presents a comparison of the methods considered in this survey. Filled circles mean that the method has the property while empty circles implies that the approach could be modified to incorporate the property.

Method	Failures		Robust	Adaptive	Fault Model		Constraints	Model Type	
	Actuator	Structural			FDI	Assumed		Linear	Nonlin.
MMST		●		●	●			●	
IMM		●		●	●		○	●	
PCA	●		○			●		●	●
CA	●					●	○	●	
Feed.Lin.	●	●		●	●				●
SMC	○	●	●				●		●
EA		●				●		●	
MRAC		●		●	●			●	○
MPC	●	●	○	○	●	●	●	●	●

Table 4.1: Comparison of Reconfigurable Control Methods [15]

Although MPC still has fundamental issues that need to be cleared, it is the only one that has the potential of solving the general reconfigurable control problem.

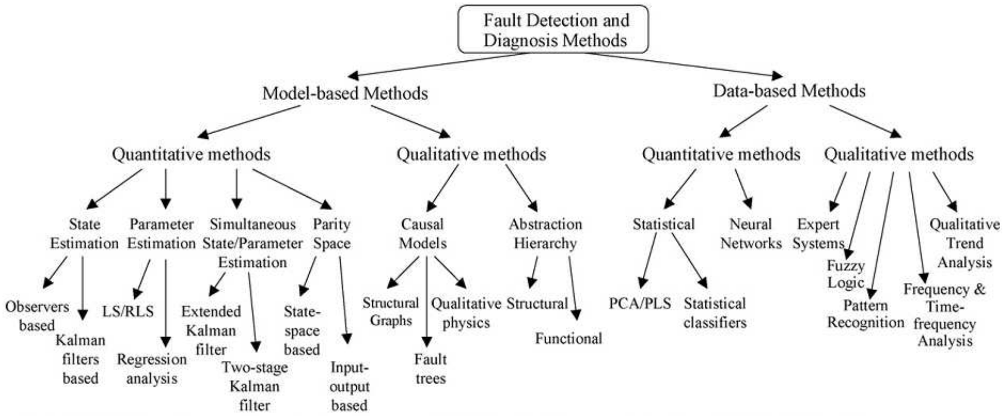
4.2 Fault Detection and Identification

Fault detection and diagnosis (FDD) or identification (FDI) is an active research area in fault tolerant control systems, and has a huge volume of studies published [15, 18].
In this approach, a fault detection and isolation (FDI) component is used to perform the following tasks:

- Indicate the occurrence of a fault (fault detection).
- Determine the exact location of the failure (fault isolation).
- Determine the type and value of the failure (fault identification).

The FDI component can be used together with a reconfigurable controller to form a fault tolerant control system, and the structure and parameters of the controller are to be changed based on the fault diagnosis result to accommodate failures. The performance of a FDI scheme is measured by how fast does it responds to a fault, its sensitivity to faults, its propensity to issue false alarms and the number of faults which go undetected. A system which is designed to respond quickly to certain abrupt changes must necessarily be sensitive to certain high frequency effects, this in turn will increase the sensitivity of the system to noise, via the occurrence of false alarms signaled by the failure detection system.

Many methods have been developed for the detection and diagnosis of certain types of system faults, and they fall into two major categories: model-based and model-free (figure 4.3 gives a general overview in each category). These two schemes can further be classified as quantitative and qualitative approaches. Essentially, a quantitative model-based FDD scheme uses a mathematical model (often known as analytical redundancy) to carry out in real-time. The four most commonly used techniques are based on state estimation, parameter estimation, parity space and combination of the first three. From a modelling perspective, there are data-based methods that do not assume any form of physics information and rely on process history data to build the appropriate model. Among various data-based modelling methods, neural networks (NN) have been an active area of research due to their learning ability. These methods are attractive because the physical or mathematical models for complex systems are usually hard to obtain.



Note: LS/RLS: Least Squares/Recursive Least Squares; PCA: Principal Component Analysis; PLS: Partial Least Squares.

Figure 4.3: Overview on fault detection methods[36]

In this thesis no FDI system was implemented, nevertheless the control strategy described in section 4.4 depends on the existence of a fault detection and identification system. The main role of this system, would be to determine which surface suffers the failure, the deflection angle and then select the most adequate controller, updating at the same time its input and output bounds (constraints).

4.3 Model Predictive Control

Predictive control has been known by several names over the years: Model Based Predictive Control (MBPC), Generalised Predictive Control (GPC), Dynamic Matrix Control (DMC) and Sequential Open-Loop Optimizing Control (SOLO), among others [5, 14, 22, 13, 21, 23, 16, 12, 24]. Since all simulations were to be made in a *Simulink* environment, the *Matlab's Model Predictive Control Toolbox* was chosen, that gathers software capable of design, analyse and implement the desired control system, and provides a convenient graphical user interface that supports customisation.

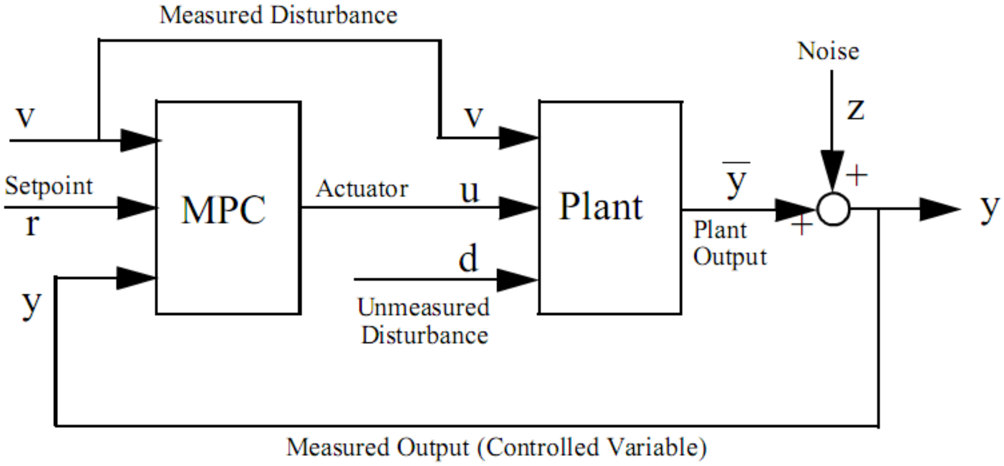


Figure 4.4: Matlab MPC Block Overview

The MPC toolbox also provides a *Simulink* block that can be easily integrated at the control scheme and the aircraft's *Simulink* model. The usual MPC Toolbox application involves a plant having multiple inputs and multiple outputs, but for exemplification purposes the figure 4.4 illustrates a block diagram of a single input single output (SISO) or (MIMO) MPC toolbox application. The unmeasured disturbance is always present, being an input not affected by the controller or the plant, also represents all the unknowns, unpredictable events that upset plant operation.

Some applications beside having *unmeasured disturbances* d, also have *measured disturbance* v, that is another independent input affecting $\bar{y}$. This methodology is capable to perform feedforward control, compensating the v's impact on $\bar{y}$ immediately rather than waiting until the effect appears in the y measurement. In other words, an MPC Toolbox design always

Symbol	Description
d	*Unmeasured disturbance.* Unknown for its effect on the plant output. The controller provides feedback compensation for such disturbances
r	*Setpoint(or reference).* The target value for the output
u	*Manipulated variable(or actuator).* The signal adjusted by the controller in order to achieve its objectives
v	*Measured disturbance(optional).* The controller provides feedfoward compensation for such disturbances as they occur to minimize their impact on the output.
y	*Measured output.* Used to estimate the true value, $\bar{y}$
$\bar{y}$	*Output(or controlled variable).* The true value, uncorrupted by measurement noise, to be held at the setpoint.
z	*Measurement noise.* Stand for all sampling errors, electrical noise, drifting calibration and other effects that impair measurement precision and accuracy.

Table 4.2: Description of MPC Toolbox Signals

provides feedback compensation for unmeasured disturbances and feedforward compensation for any measured disturbance (table 4.2 describes the signals presented in figure 4.2).
MPC Toolbox design generates a $discrete - time$ controller, that takes action at regularly spaced, discrete time instants. These *sampling instants* Δt are the times at which the controller acts.

Figure 4.5 ilustrates the state of a hypothetical $SISO$ MPC system that has been operating for many sampling instants. Where k is a variable representing the current instant, y_k the latest measured output and $y_{k-1}, y_{k-2}, ...$ as previous measurements represented by the filled circles. The controller's previous control values, $u_{k-4}, ..., u_{k-1}$ are also shown as filled circles, normally a $zero - order\ hold$ receives each move from the controller and holds it until the next sampling instant, causing the step-wise variations.
To calculate the next move u_k, the controller operates in two distinct phases:

- *Estimation* - In order to make an intelligent move, the controller needs to know the plant current state. This includes the true value of the controlled variable $\bar{y}_k$, and any internal variables that influence the future trend $\bar{y}_{k+1}$, $..., \bar{y}_{k+P}$. To accomplish this, the

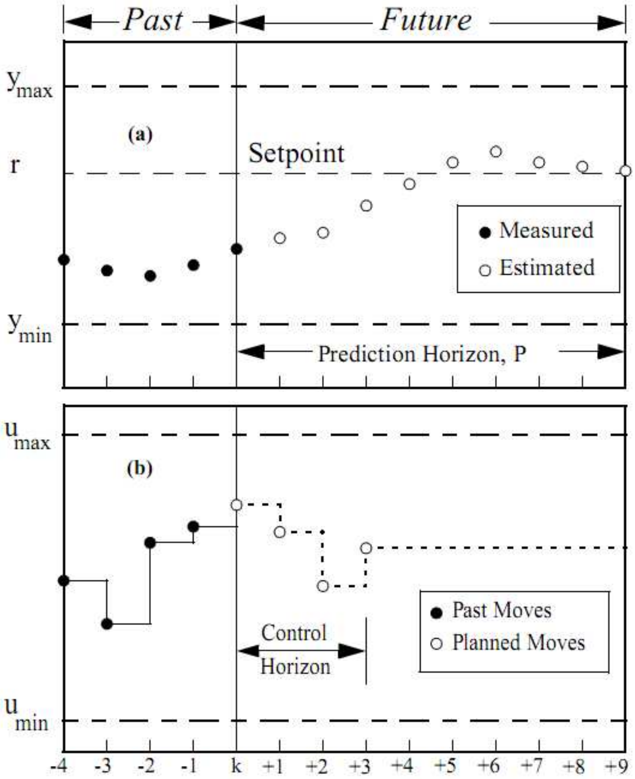


Figure 4.5: MPC Receding Horizon Graphic

controller uses all past and current measurements and the models $u \to \bar{y}$, $d \to \bar{y}$, $v \to \bar{y}$ and $z \to \bar{y}$.

- *Optimization* - Values of setpoints, measured disturbances and constraints are specified over a finite horizon of future sampling instants, $k+1, k+2, \ ..., k+P$, where P (a finite integer ≥ 1) is the prediction horizon. The controller computes M moves $u_k, u_{k+1}, \ ..., u_{k+M-1}$, where M $(\geq 1, \leq P)$ is the control horizon.

After performing all calculations, the controller applies u_k to the plant. The plant operates with this constant input until the next sampling instant Δt time units later. The controller obtains new measurements, and totally revises its parameters, this cycle repeats indefinitely. As predictions made during the optimization stage are imperfect, periodic measurement feedback allows the controller to correct for this error and for unexpected disturbances. Most designers choose P and M such that controller performance is insensitive to small adjustments in these horizons.

4.3.1 Prediction Model

The MPC controller uses a Plant Model that is a linear time-invariant(LTI) system described by the equations:

$$\begin{aligned} x(k+1) &= Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \\ y_m(k) &= C_m x(k) + D_{vm} v(k) + D_{dm} d(k) \\ y_u(k) &= C_u x(k) + D_{vu} v(k) + D_{du} d(k) + D_{uu} u(k) \end{aligned} \tag{4.1}$$

where $x(k)$ is the n_x- dimensional state vector of the plant, $u(k)$ is the n_u-dimensional vector of manipulated variables (MV;i.e. Command inputs), $v(k)$ is the n_v-dimensional vector of measured disturbances (MD), $d(k)$ is the n_d-dimensional vector of unmeasured disturbances (UD) entering the plant, $y_m(k)$ is the vector of measured outputs (MO), and $y_u(k)$ is the vector of unmeasured outputs (UO). The overall output vector $y(k)$ collects $y_m(k)$ and $y_u(k)$. Finally $d(k)$ collects both state disturbances ($B_d \neq 0$) and output disturbances ($D_d \neq 0$). The unmeasured disturbances $d(k)$ is modeled as the output of the LTI system:

$$\begin{aligned} x_d(k+1) &= \bar{A} x_d(k) + \bar{B} n_d \\ d(k) &= \bar{C} x_d(k) + \bar{D} n_d(k) \end{aligned} \tag{4.2}$$

The system described by the above equations is driven by the random Gaussian noise $n_d(k)$, having zero mean and unit covariance matrix. In the present work the matrices A, B, C, D of the model representing the process to control are obtained by linearizing a nonlinear dynamical system, such as:

$$\begin{aligned} x' &= f(x, u, v, d) \\ y &= h(x, u, v, d) \end{aligned} \tag{4.3}$$

at some nominal value $x = x_0$, $u = u_0$, $v = v_0$, $d = d_0$. In these equations x' denotes either the time derivative (continuous time model) or the successor $x(k+1)$ (discrete time model). To obtain x_0, u_0, v_0, d_0 it was used $TRIM$ function on the simulink model describing the nonlinear dynamical equations, and A, B, C, D by using $LINMOD$. The linearized model has the form:

$$\begin{aligned} x' \cong{} & f(x_0, u_0, v_0, d_0) + \nabla_x f(x_0, u_0, v_0, d_0)(x - x_0) + \nabla_u f(x_0, u_0, v_0, d_0)(u - u_0) \\ & + \nabla_v f(x_0, u_0, v_0, d_0)(v - v_0) + \nabla_d f(x_0, u_0, v_0, d_0)(d - d_0) \\ y \cong{} & h(x_0, u_0, v_0, d_0) + \nabla_x h(x_0, u_0, v_0, d_0)(x - x_0) + \nabla_u h(x_0, u_0, v_0, d_0)(u - u_0) \\ & + \nabla_v h(x_0, u_0, v_0, d_0)(v - v_0) + \nabla_d h(x_0, u_0, v_0, d_0)(d - d_0) \end{aligned} \tag{4.4}$$

The matrices A, B, C, D of the model are obtained from the Jacobian matrices appearing in the equations above. The linearized dynamics are affected by the constant terms $F = f(x_0, u_0, v_0, d_0)$ and $H = h(x_0, u_0, v_0, d_0)$. The MPC algorithm adds a measured disturbance $v = 1$, so that F and H can be embedded into B_v and D_v respectively, as additionals columns.

4.3.2 Optimization Problem

Having the estimates of $x(k)$, $x_d(k)$ available at time k, the MPC control action at time k is obtained by solving the optimization problem formulated as follows:

$$\min_{\Delta u(k|k),\ ...,\Delta u(m-1+k|k),\varepsilon} \left\{ \sum_{i=0}^{p-1} \left(\sum_{j=1}^{n_y} |w^y_{i+1,j}(y_j(k+i+1|k) - r_j(k+i+1))|^2 \right.\right.$$
$$\left.\left. + \sum_{j=1}^{n_u} |w^{\Delta u}_{i,j} \Delta u_j(k+i|k)|^2 + \sum_{j=1}^{n_u} |w^u_{i,j}(u_j(k+i|k) - u_{jtarget}(k+i))|^2 \right) + \rho_\varepsilon \varepsilon^2 \right\} \tag{4.5}$$

where '$(\ )_j$' denotes the $j - th$ component of a vector, '$(k+i|k)$' denotes the value predicted for time $k+i$ based on the information available as time k, $r(k)$ is the current sample of the output reference, subject to:

$$\begin{gathered}
u_{j,min}(i) - \varepsilon V^u_{jmin}(i) \le u_j(k+i|k) \le u_{j,max}(i) + \varepsilon V^u_{jmax}(i) \\
\Delta u_{j,min}(i) - \varepsilon V^{\Delta u}_{jmin}(i) \le \Delta u_j(k+i|k) \le \Delta u_{j,max}(i) + \varepsilon V^{\Delta u}_{jmax}(i) \\
y_{j,min}(i) - \varepsilon V^y_{jmin}(i) \le y_j(k+i+1|k) \le y_{j,max}(i) + \varepsilon V^y_{jmax}(i) \\
\Delta u(k+h|k) = 0, h = m,\ ..., p-1 \\
i = 0,\ ..., p-1 \\
\varepsilon \ge 0
\end{gathered} \tag{4.6}$$

the sequence of input increments being $\Delta u(k|k), \cdots, \Delta u(m-1+k|k)$, and ε is the slack variable, finally $u(k) = u(k-1) + \Delta u(k|k)^*$, where $\Delta u(k|k)^*$ is the first element of the optimal sequence. When the reference r is not known in advance, the current reference $r(k)$ is used over the whole prediction horizon. $w^{\Delta u}_{i,j}$, $w^u_{i,j}$, $w^y_{i,j}$, are nonnegative weights for the corresponding variable. The smaller w, the less important is the behaviour of the corresponding variable to the overall performance index. In other hand $u_{j,min}$, $u_{j,max}$, $\Delta u_{j,min}$, $\Delta u_{j,max}$, $y_{j,min}$, $y_{j,max}$ are the lower and upper bounds on the corresponding variables. The weight ρ_ε on the slack variable penalizes the violation of the constraints, the larger ρ_ε with respect to input and output weights, the more the constraint violation is penalized. The Equal Concern for the Relaxation(ERC) vectors V^u_{min}, V^u_{max}, $V^{\Delta u}_{min}$, $V^{\Delta u}_{max}$, V^y_{min}, V^y_{max} have nonnegative entries which represent the concern for relaxing the corresponding constraint, so the larger V, the softer is the constraint. If $V = 0$ this means that the constraint is *hard* so it can't be violated. As hard output constraints may cause infeasibility of the optimization problem, because of unpredictied disturbances, model mismatch, or just because of the numerical round off. By default:

$$\rho_\varepsilon = 10^5 \max \left\{ w_{i,j}^{\Delta u}, w_{i,j}^{u}, w_{i,j}^{y} \right\}$$

Vector $u_{target}(k+i)$ is a setpoint for the input vector. One typically uses u_{target} if the number of inputs is greater than the number of outputs, as a lower-priority setpoint. The algorithm implemented in this toolbox uses different procedures depending on the presence of constraints. For instance, if all bounds are infinite, then the slack variable ε is removed, and the problem in equations 4.5 and 4.6 is solved analytically, otherwise a Quadratic Programming(QP) is used. Since output constraints are always soft, the QP problem is never unfeasible, nevertheless if for numerical reasons the QP problem becomes unfeasible, the second sample from the previous optimal sequence is applied, $u(k) = u(k-1) + \Delta^* u(k|k-1)$. To solve the optimization problem the following matrices are used, assuming that $d(k) = n_d(k)$ is a white Gaussian noise, for simplicity, denote by:

$$x \leftarrow \begin{bmatrix} x \\ x_d \end{bmatrix}, A \leftarrow \begin{bmatrix} A & B_d\bar{C} \\ 0 & \bar{A} \end{bmatrix}, B_u \leftarrow \begin{bmatrix} B_u \\ 0 \end{bmatrix}, B_d \leftarrow \begin{bmatrix} B_d\bar{D} \\ \bar{B} \end{bmatrix}, C \leftarrow \begin{bmatrix} C & D_d\bar{C} \end{bmatrix} \tag{4.7}$$

Then, the prediction model is given by:

$$\begin{gathered} x(k+1) = Ax(k) + B_u u(k) + B_v v(k) + B_d n_d(k) \\ y(k) = Cx(k) + D_v v(k) + D_d n_d(k) \end{gathered} \tag{4.8}$$

Considering for simplicity the prediction of the future trajectories of the model performed at time $k = 0$. $n_d(i) = 0$ is set for all predictions instants i, obtaining:

$$y(i|0) = C\left[A^i x(0) + \sum_{h=0}^{i-1}\left(A^{i-1}B_u\left(u(-1) + \sum_{j=0}^{h}\Delta u(j)\right) + B_v v(h)\right)\right] + D_v v(i) \tag{4.9}$$

which gives:

$$\begin{bmatrix} y(1) \\ \cdots \\ y(p) \end{bmatrix} = S_x x(0) + S_{u1}u(-1) + S_u \begin{bmatrix} \Delta u(0) \\ \cdots \\ \Delta u(p-1) \end{bmatrix} + H_v \begin{bmatrix} v(0) \\ \cdots \\ v(p) \end{bmatrix} \tag{4.10}$$

where:

$$S_x = \begin{bmatrix} CA \\ CA^2 \\ \cdots \\ CA^p \end{bmatrix} \in \mathbb{R}^{pn_y \times n_x}, S_{u1} = \begin{bmatrix} CB_u \\ CB_u + CAB_u \\ \cdots \\ \sum_{h=0}^{p-1} CA^h B_u \end{bmatrix} \in \mathbb{R}^{pn_y \times n_u}$$

$$S_u = \begin{bmatrix} CB_u & 0 & \cdots & 0 \\ CB_u + CAB_u & CB_u & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots \\ \sum_{h=0}^{p-1} CA^h B_u & \sum_{h=0}^{p-2} CA^h B_u & \cdots & CB_u \end{bmatrix} \in \mathbb{R}^{pn_y \times pn_u}$$

$$H_v = \begin{bmatrix} CB_v & D_v & 0 & \cdots & 0 \\ CAB_v & CB_v & D_v & \cdots & 0 \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ CA^{p-1}B_v & CA^{p-2}B_v & CA^{p-3}B_v & \cdots & D_v \end{bmatrix} \in \mathbb{R}^{pn_y \times (p+1)n_v}$$

Let $z = [z_0; \cdots ; z_{m-1}]$ be the free optimization variables of the optimization problem (in case of systems with a single manipulated variables, $z_0; \cdots ; z_{m-1}$ are scalars). So the cost function to be optimized is:

$$\begin{aligned} J(z, \varepsilon) = & \left(\begin{bmatrix} u(0) \\ \cdots \\ u(p-1) \end{bmatrix} - \begin{bmatrix} u_{target}(0) \\ \cdots \\ u_{target}(p-1) \end{bmatrix} \right)^T W_u^2 \left(\begin{bmatrix} u(0) \\ \cdots \\ u(p-1) \end{bmatrix} - \begin{bmatrix} u_{target}(0) \\ \cdots \\ u_{target}(p-1) \end{bmatrix} \right) \\ & + \begin{bmatrix} \Delta u(0) \\ \cdots \\ \Delta u(p-1) \end{bmatrix}^T W_{\Delta u}^2 \begin{bmatrix} \Delta u(0) \\ \cdots \\ \Delta u(p-1) \end{bmatrix} \\ & + \left(\begin{bmatrix} y(1) \\ \cdots \\ y(1) \end{bmatrix} - \begin{bmatrix} r(1) \\ \cdots \\ r(p) \end{bmatrix} \right)^T W_y^2 \left(\begin{bmatrix} y(1) \\ \cdots \\ y(1) \end{bmatrix} - \begin{bmatrix} r(1) \\ \cdots \\ r(p) \end{bmatrix} \right) + \rho_\varepsilon \varepsilon^2 \end{aligned} \tag{4.11}$$

where:

$$W_u = diag(w^u_{0,1}, w^u_{0,2}, \cdots, w^u_{0,n_u}, \cdots, w^u_{p-1,1}, w^u_{p-1,2}, \cdots, w^u_{p-1,n_u})$$
$$W_{\Delta u} = diag(w^{\Delta u}_{0,1}, w^{\Delta u}_{0,2}, \cdots, w^{\Delta u}_{0,n_u}, \cdots, w^{\Delta u}_{p-1,1}, w^{\Delta u}_{p-1,2}, \cdots, w^{\Delta u}_{p-1,n_u})$$
$$W_y = diag(w^y_{1,1}, w^y_{1,2}, \cdots, w^y_{1,n_y}, \cdots, w^y_{p,1}, w^y_{p,2}, \cdots, w^y_{p,n_y})$$

Finally, after substituting $u(k)$, $\Delta u(k)$, $y(k), J(z)$ the cost function can be rewritten as:

$$\begin{aligned} J(z,\varepsilon) = \rho_\varepsilon \varepsilon^2 + z^T K_{\Delta u} z + 2 \left(\begin{bmatrix} r(1) \\ \cdots \\ r(p) \end{bmatrix}^T K_r + \begin{bmatrix} v(0) \\ \cdots \\ v(p) \end{bmatrix} K_v + u(-1)^T K_u \right. \\ \left. + \begin{bmatrix} u_{target}(0) \\ \cdots \\ u_{target}(p-1) \end{bmatrix}^T K_{ut} + x(0)^T K_x \right) z + constant \end{aligned} \tag{4.12}$$

Let us now consider the limits on inputs, input increments, and outputs along with the constraint $\varepsilon \geq 0$:

$$\begin{bmatrix} y_{min}(1) - \varepsilon V^y_{min}(1) \\ \cdots \\ y_{min}(p) - \varepsilon V^y_{min}(p) \\ u_{min}(0) - \varepsilon V^u_{min}(0) \\ \cdots \\ u_{min}(p-1) - \varepsilon V^u_{min}(p-1) \\ \Delta u_{min}(0) - \varepsilon V^{\Delta u}_{min}(0) \\ \cdots \\ \Delta u_{min}(p-1) - \varepsilon V^{\Delta u}_{min}(p-1) \end{bmatrix} \leq \begin{bmatrix} y(1) \\ \cdots \\ y(p) \\ u(0) \\ \cdots \\ u(p-1) \\ \Delta u(0) \\ \cdots \\ \Delta u(p-1) \end{bmatrix} \leq \begin{bmatrix} y_{max}(1) - \varepsilon V^y_{max}(1) \\ \cdots \\ y_{max}(p) - \varepsilon V^y_{max}(p) \\ u_{max}(0) - \varepsilon V^u_{max}(0) \\ \cdots \\ u_{max}(p-1) - \varepsilon V^u_{max}(p-1) \\ \Delta u_{max}(0) - \varepsilon V^{\Delta u}_{max}(0) \\ \cdots \\ \Delta u_{max}(p-1) - \varepsilon V^{\Delta u}_{max}(p-1) \end{bmatrix}$$

Similarly to what was done for the cost function, let $u(k), \Delta u(k), y(k)$ can be substituted by:

$$M_z z + M_\varepsilon \varepsilon \leq M_{\lim} + M_v \begin{bmatrix} v(0) \\ \cdots \\ v(p) \end{bmatrix} + M_u u(-1) + M_x x(0) \tag{4.13}$$

where matrices $M_z, M_\varepsilon, M_{lim}, M_v, M_u, M_x$ are obtained from the upper and lower bounds and ERC values. After the matrices are build as described before the optimization problem is solved at each time step k depending if the problem is constrained or not.

Unconstrained MPC

In this case upper and lower bounds are removed and the optimal solution is computed analytically:

$$z^* = -K_{\Delta u}^{-1}\left(\begin{bmatrix} r(1) \\ \cdots \\ r(p) \end{bmatrix}^T K_r + \begin{bmatrix} v(0) \\ \cdots \\ v(p) \end{bmatrix} K_v + u(-1)^T K_u + \begin{bmatrix} u_{target}(0) \\ \cdots \\ u_{target}(p-1) \end{bmatrix}^T K_{ut} + x(0)^T K_x\right)^T \tag{4.14}$$

and the MPC controller sets $\Delta u(k) = z_0^*, u(k) = u(k-1) + \Delta u(k)$.

Constrained MPC

The optimal solutions z^*, ε^* are computed by solving the quadratic program described in equation 4.12 and equation 4.13 using a Quadratic problem solver. The algorithm solves the convex program:

$$\begin{aligned} \min \quad & \tfrac{1}{2}x^T Hx + f^T x \\ \text{subject to} \quad & Ax \leq b, x \geq x_{min} \end{aligned} \tag{4.15}$$

using Dantzig-Wolfe's active set method [10]. The Hessian matrix H should be positive definite, by default $x_{min} = 10^{-5}$. The Dantzig-Wolfe's algorithm uses the direction of the largest gradient, and the optimum is usually found after about $n + q$ iterations, where $n = dim(x)$ is the number of optimization variables, and $q = dim(b)$ is the number of constraints. More than $3(n + q)$ iterations are rarely required.

4.4 Control Strategy

The goal of this strategy is to allow the aircraft to maintain basic functionality after a fault occurred. In this work, the basic functionality is considered to be maintained if the aircraft is able to reach a straight level flight and can perform a turn or climb manoeuvre at desired set point. As stated previously, no additional hardware or redundancy is available in this aircraft, so the strategy will be limited to the development of the MPC controllers.

The objective is to test the MPC controllers in a failure situation, nevertheless these controllers must be able to control the aircraft in the nominal condition, as well in the failure

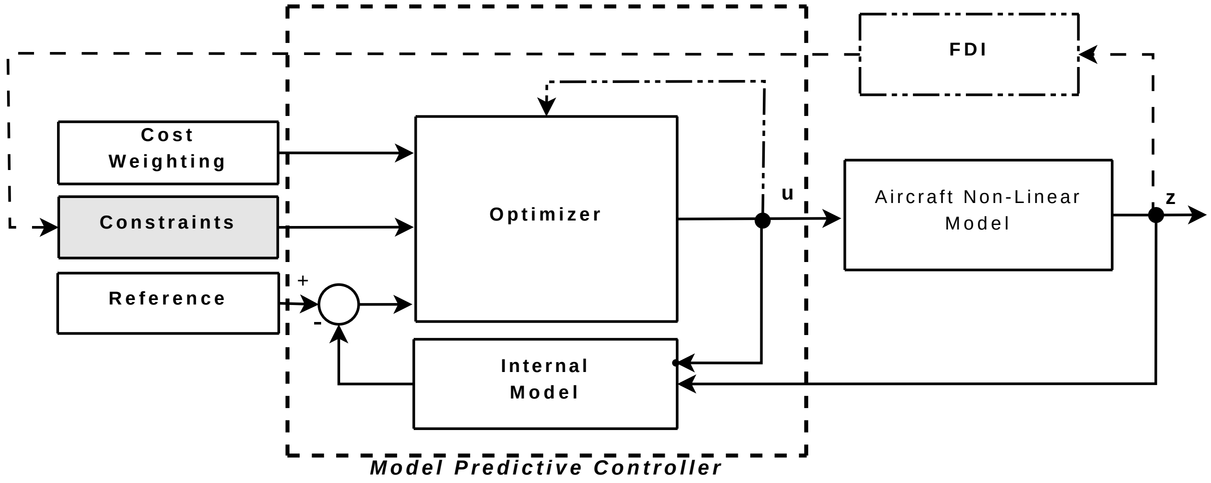


Figure 4.6: Model Predictive Control Scheme

situation. For comparative reasons a *Classic* controller is used in every test. This block consists on several control chains (see appendixB) designed in [7] for a non-failure scenario.

As mentioned in chapter 3, the faults to be tested are *Partial and Total power loss*, on all four control surfaces, since some failures affect mostly *pitch* angle θ in the case of *elevator* and *flaps*, or *yaw* angle ψ in the *rudder* and *aileron* failure, two different MPC controllers had to be designed:

- *Pitch Hold* - Control mainly the aircraft velocity and *pitch* angle θ during *elevator* and *flaps* failures tests.
- *Bank Hold* - Focus on the control of *yaw* angle ψ, and velocity during *rudder* and *aileron* failures tests.

These two controllers only differ in the Cost Weighting and Constraints parameters, the Internal Linear model remains the same, figure 4.6 illustrates these different controller aspects.

As shown in figure 4.6 an important feature is the online constraint update, a *Simulink* block was created to change the upper an lower limits of control surface (see figure 4.7). The *Constraints* block is inherently dependant on a FDI system, that can also rapidly give a reliable information on the surface's jammed deflection as an *external manipulated variable* as shown in figure 4.7. In figure 4.6 this FDI block is dashed because it wasn't implement in the present thesis.

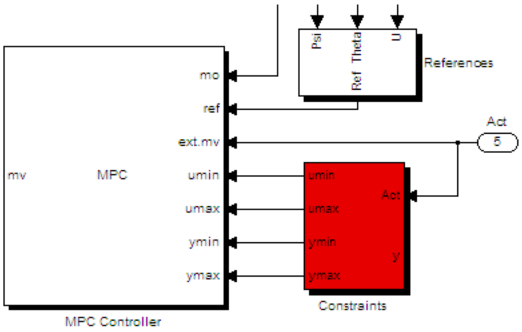


Figure 4.7: Online Constraints Update Block

For instance, during a *Pitch Hold Controller* test, a Flaps *Hard Over* failure occur, the MPC controller has to control elevators in order to suppress this failure. At the same time $u_{j,min}$, $u_{j,max}$ are both updated to the deflection value which the surface is jammed, in this particular case it would be $u_{4,min} = 45^o$, $u_{4,max} = 45^o$, since Flaps are the fourth model input and its *Hard Over* failure correspond to a maximal deflection.

Although aileron and elevator failures occur in only one of the surfaces(in this case always the left one), the model receives only a singular input value and also a unique constraint value, if only one aileron is frozen in a position, the input bound have to stay unchanged because the healthy aileron(right one) can proceed to its maximum deflection 20^o as also to -20^o, notice that in the case of a *Partial Power Loss* these limits remain the same. This *Constraints block* has also the ability to change the output variables limits, but in this case the results were always poorer, because constraints on output variables, can lead very rapidly to unfeasible solutions.

In the next chapter each MPC controllers will be studied more profoundly.

Chapter 5

Simulation Results

The goal of this chapter is to demonstrate that it is possible to recover enough of the nominal performance so that a pilot could continue to fly the aircraft after one of the failures described in chapter 3. With this purpose two MPC controllers have been developed to track a reference trajectory provided by the nominal reference block. For the purpose of demonstration, a

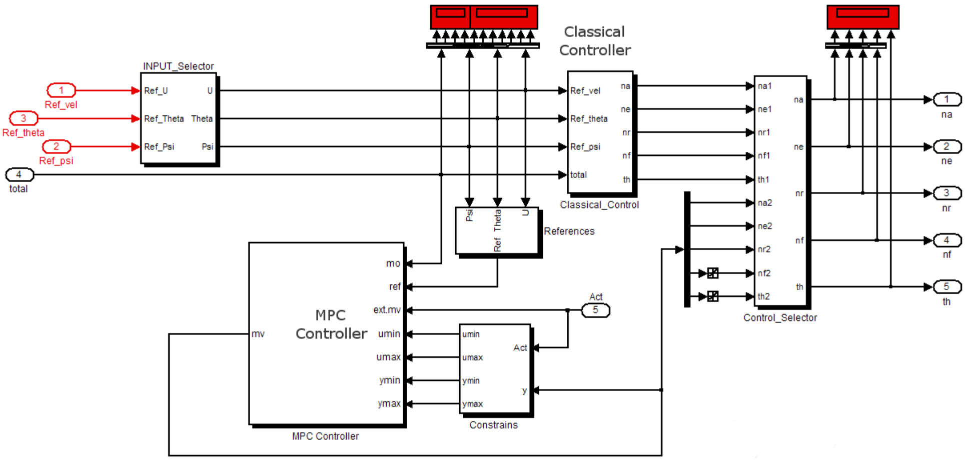


Figure 5.1: Control Structure

Classic Controller developed in [7] is used to compare performances between this approach and the MPC one. In order to stress the response of the system, failures were chosen to occur during a transient, just after a change of reference. The aircraft has to fly the rest of the trajectory with the failed surface. The data obtained during the simulation is then plotted against the *Classical Controller* response. In the following figure 5.1 shows the *Simulink* control system with a *Constraints Update* block, *Controller Selector*, *Classical* Controller and MPC controller.

5.1 MPC Controllers Settings

Two different MPC controllers were chosen to deal with the different failures. One more fitted to deal with Pitch affecting faults and another one prepared to deal with Yaw/Bank perturbations. Tuning these controllers was a difficult and time consuming endeavour, as it is not clear how aircraft performance requirements should be translated into a MPC cost function. Tuning the cost function parameters is the most significant aspect in the development time. The MPC internal linear model was obtained from the nominal point described in section 2 as:

- Throttle: 6.24%
- Aileron deflection η_a: 0.0 rad
- Elevator deflection η_e: -0.0285 rad
- Rudder deflection η_r: 0.0 rad
- Flaps deflection η_f: 0.0 rad
- Velocity: 21.156 m/s
- *Pitch* θ: 5.6×10^{-5} rad

The mismatch between the internal MPC model and the *true* aircraft becomes difficult when dealing with the nonlinear model. The horizons chosen are largely the result of a trail and error approach. Controllers were tuned for this particular failures scenarios and it is unknown if would work for other failures or flight conditions. Starting from the MPC block specifications for aircraft control [25], and then changing them using an empirical approach, the following parameters were tuned:

- Sample time (Interval) = 0.02 seconds
- Prediction Horizon P = 10 (Interval times)
- Control Horizon M= 2 (Interval times)

5.1.1 Pitch Hold Controller

The first controller is a simple pitch hold. In this case, the pitch angle can be commanded to any position while maintaining steady flight.

Since the controller is responsible for fault compensation of Elevator and Flaps, it was requested a climb maneuver from the nominal pitch angle θ_0 , up to a determined value and back to the initial point as shown in figure 5.2. For that a steady value of velocity and yaw angle ψ was defined for the aircraft to follow. The main purpose of this controller is to successfully command the flaps and the healthy elevator if an elevator starts to malfunction, a similar action is requested in the case of flaps failure. The faults considered are the ones stated in chapter 3. The controller's cost function parameters are summarised in the following tables:

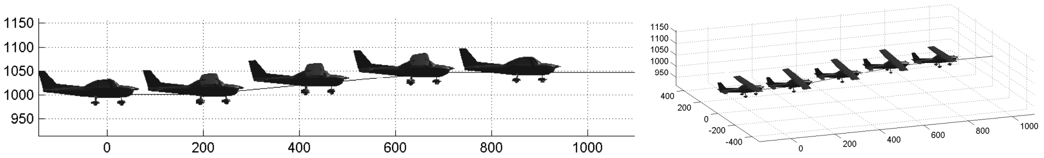


Figure 5.2: Aircraft Trajectory in Pitch Hold Test [*aircraft not to scale*]

Input	Amplitude (rad)		Rate (rad/s)		Rate Weight	Weight
	u_{min}	u_{max}	Δu_{min}	Δu_{max}	$w^{\Delta u}$	w^u
η_a	-0.349	0.349	-1	1	0.1	10
η_e	-0.305	0.305	-1	1	0.1	0
η_r	-0.523	0.523	-0.1	0.1	0.1	10
η_f	0	0.7854	-0.005	0.005	0.5	0
T_H	0	1	-0.1	0.1	0.1	0

Table 5.1: *Pitch Hold* Controller Input Tuning Parameters

The inputs weightings correspond to the w^u variable described in chapter 4. The Amplitude Input constraints reflect the physical limitations of the actuator, the *Matlab* Controller is also able to cope with rate bounds in the actuators. These specifications (parameter selection), revelled to be very important, because it was possible to penalise flaps actuation, throttle and rudder, otherwise the airplane would start to become unstable. With the Rate Weight option, the throttle and elevators commands were encouraged because they were less penalised, in contrast to the flaps. Promoting the solution of pitch problems through the use of the healthy elevator and not using only flaps. Due to an experienced instability, between the use of rudder and aileron a weight of 10 was introduced in these variables.

Output	Amplitude		Weight
	y_{min}	y_{max}	w^y
U - Velocity in X_E	$-\infty$	∞	1.2
V - Velocity in Y_E	$-\infty$	∞	0
W - Velocity in Z_E	$-\infty$	∞	0
P - Roll rate	$-\infty$	∞	0
Q - Pitch rate	$-\infty$	∞	0
R - Yaw rate	$-\infty$	∞	0
ϕ - Roll angle	$-\infty$	∞	0
θ - Pitch angle	$-\infty$	∞	10
ψ - Yaw angle	$-\infty$	∞	2
α - Attack angle	$-\infty$	∞	0
β - Sideslip angle	$-\infty$	∞	0

Table 5.2: *Pitch Hold* Controller Output Tuning Parameters

The Outputs weightings correspond to the w^y as described in the previous chapter. Since the references chosen to simulate these faults were respectively, Velocity in X_E (Ground Velocity) U, Pitch angle θ and Yaw angle ψ, cost function was tuned to penalised the error in tracking these variables.

5.1.2 Bank Hold Controller

This Second controller is a Bank/Yaw hold. The purpose of the present controller is to allow the yaw angle ψ to be commanded to any position while maintaining steady flight.

This controller is responsible for Ailerons and Rudder faults, because of that it was requested a yaw angle ψ maneuver from the a nominal angle, to a setpoint value and back to the initial condition, see figure 5.3. For that we defined a steady value of velocity and pitch angle θ for the aircraft to follow.

In a very simple way, the main goal of this controller is to successfully control the rudder and

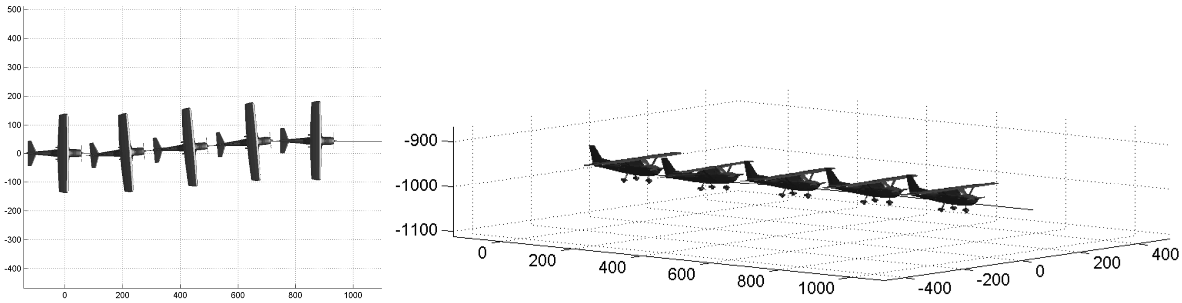


Figure 5.3: Aircraft Trajectory in Bank Hold Test [*aircraft not to scale*]

the *healthy* aileron of the aircraft on the event of an aileron failure, or in the case of a rudder fault, to control effectively the ailerons. The specifications for this mode are very similar to the previous ones, although changes were introduced in order to achieve a better tracking of yaw angle. These values are summarised in the following tables:

Input	Amplitude (rad)		Rate (rad/s)		Rate Weight	Weight
	u_{min}	u_{max}	Δu_{min}	Δu_{max}	$w^{\Delta u}$	w^u
η_a	-0.349	0.349	-1	1	0.2	0
η_e	-0.305	0.305	-1	1	0.1	0
η_r	-0.523	0.523	-0.1	0.1	0.2	0
η_f	0	0.7854	-0.005	0.005	0.5	0
T_H	0	1	-0.1	0.1	0.1	0

Table 5.3: *Bank Hold* Controller Input Tuning Parameters

For this controller there was no need to introduce weights on input variables, the use of the *Rate Weight* parameter was enough to penalise the use of unwanted actuators, like the Flaps. The values for Output variable weight defined above, are in accordance with this controller mission, that is to track yaw angle in the event of a failure that affects mainly this parameter. A lower weight value was given to Ground Velocity U, because faults in this domain induce a *Sideslip angle*β that affects largely this velocity, making it difficult to maintain the reference velocity. The parameters Sample instant Δt, Prediction Horizon P and Control Horizon M are the same as the ones used in the *Pitch Hold Controller*.

Output	Amplitude		Weight
	y_{min}	y_{max}	w^y
U - Velocity in X_E	$-\infty$	∞	0.5
V - Velocity in Y_E	$-\infty$	∞	0
W - Velocity in Z_E	$-\infty$	∞	0
P - Roll rate	$-\infty$	∞	0
Q - Pitch rate	$-\infty$	∞	0
R - Yaw rate	$-\infty$	∞	0
ϕ *- Roll angle*	$-\infty$	∞	0
θ *- Pitch angle*	$-\infty$	∞	2
ψ *- Yaw angle*	$-\infty$	∞	10
α *- Attack angle*	$-\infty$	∞	0
β *- Sideslip angle*	$-\infty$	∞	0

Table 5.4: *Bank Hold* Controller Output Tuning Parameters

5.2 Fault Simulation Results

As mentioned in chapter 3 two types of failures will be tested, *Partial and Total Actuator Power Loss*. The last case includes three situations *Hard Over*, *Hard Under* and Frozen or Stuck.

The *Piper* PA18 *Super Cub* is assumed to be at a steady level flight and a turning or climbing maneuver is performed, in order to reach the desired Pitch or Yaw angle set point of 5 *degrees*. After a few seconds (15 seconds of simulation), a fault is introduced in the aircraft model, now the controllers have to be able to compensate the fault by adjusting the actual aircraft attitude. Then the Pitch or Yaw angle has to return to the nominal position, performing a total of 40 seconds.

Meanwhile in order to allow the aircraft to have a performance similar to the undamaged aircraft or to restore flight conditions a good extent, two additional constant references are request to track *Velocity in* X_E, U and Pitch or Yaw depending on the fault simulated. In order to best understand both MPC and *Classic* controllers action, actuator data figures will be presented.

5.2.1 Aileron Failures

In order to perform this particular test we will compare MPC *Bank Hold* Controller and a *Classic* controller, they must be able to command the aircraft from straight forward trajectory

($\psi = 0°$) to a steady state ($\psi = 5°$), five seconds later the left aileron fails and the controller must be able to maintain the steady value, and 10 seconds later bring the airplane to the initial condition. At the same time maintain Pitch angle at ($\theta = 0.3^o$) and *Velocity in* X_E at ($U = 22m/s$) as shown in figure 5.3. For an easiest data analyses and due to their greater importance only *Velocity in* X_E U, *Yaw angle* ψ, Ailerons/Rudder Deflections, and Throttle will be shown.

Aileron Total Power Loss Hard Over

In this failure scenario, the left aileron surface proceed hard over to its maximum position 20 degrees at 15 seconds. From figure 5.4 is possible to see that both controllers are able to maintain velocity reference value, in Yaw angle graphic the response changes, because after the fault, the *Classic* controller loses completely control. In other hand MPC is able get good results, however its possible to notice a small steady state error.

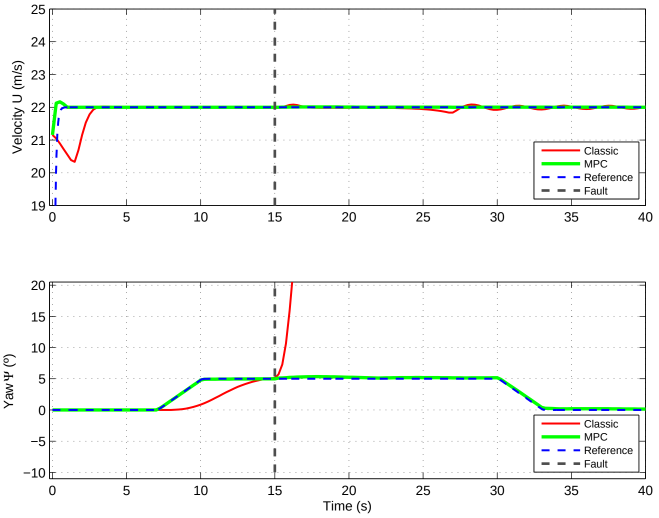


Figure 5.4: Aileron at 20° Reference Variables

Trough figure 5.5 is possible to understand why the *Classic* case performs so poorly, this is because it uses the rudder negative deflection, to compensate the fault, worsening the left turning moment caused by the hard over aileron. As the *Classic* controller, MPC drives the *Right* aileron to the same position, as the left aileron. Due to the symmetric signal convention see chapter 2, the $-20°$ value shown for this surface corresponds to same position as the left aileron. Although in the case of the MPC controller, a positive deflection is commanded to

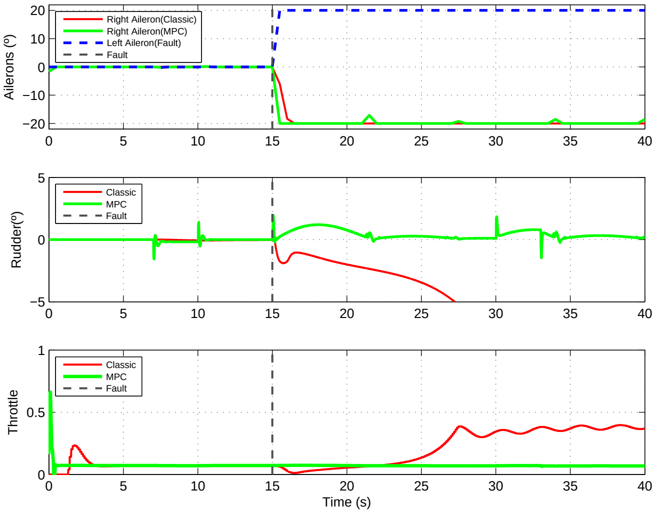


Figure 5.5: Aileron at 20° Actuation

the rudder as the failure occurs. The rest of the manoeuvres are achieved with this actuator.

Aileron Total Power Loss Hard Under

The following test scenario has the same Yaw angle profile with fault at 15 seconds, but now a hard under excursion of the of left aileron is injected. References are also the same.

The scenario is similar to the previous one, good velocity control, but the aircraft is lost in the

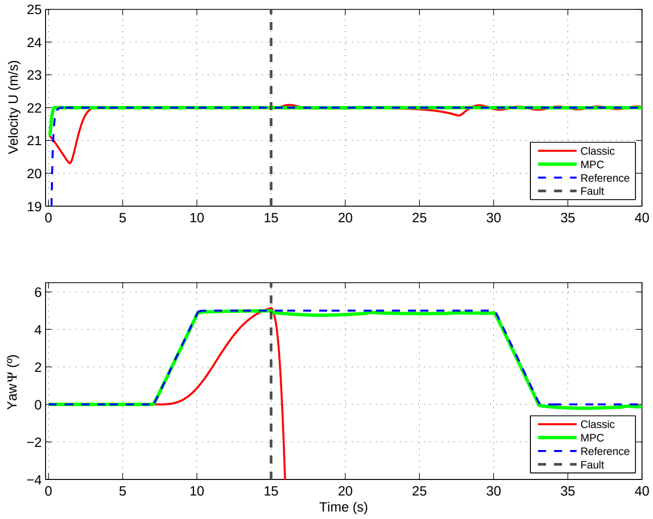


Figure 5.6: Aileron at −20° Reference Variables

Classical case and totally recovered by the MPC, beside a minor error, see figure 5.6. Taking

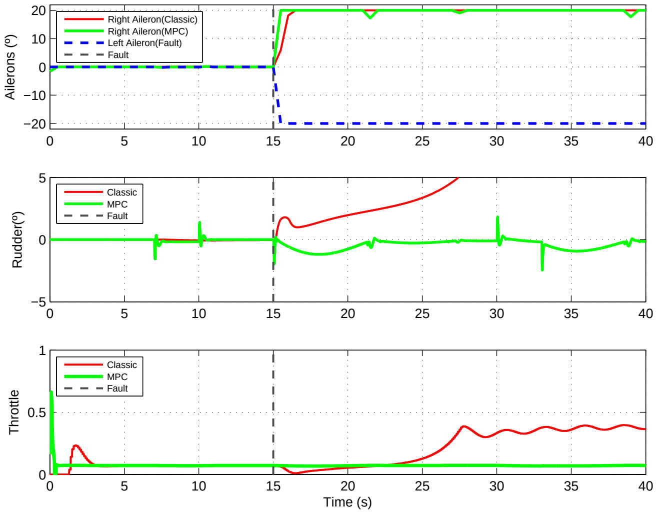


Figure 5.7: Aileron at $-20°$ Actuation

a closer look to actuation data (figure 5.7), it is possible to see that the rudder reacts again in the wrong way, a different approach has the MPC controller using correctly the rudder giving the aircraft a smooth reaction to the failure. Another interesting fact is that MPC perform the whole test without changing throttle, so without harming velocity.

Aileron Total Power Loss Frozen

This test is very much the same as the previous ones, the main difference is that the actuator isn't driven to any particular deflection, the actuator is fixed in position at 15 seconds. This consist on a less aggressive failure, the biggest challenge here is to finish the maneuver with the right aileron. As expected both controllers follow the two reference signals, although MPC performs better (figure 5.8). Since the failed aileron wasn't in a limit position, this time the other aileron could be used to perform Yaw angle tracking (figure 5.9). This time MPC uses both rudder and aileron to finish the test, in other hand *Classic* only uses only the right actuator, reason by that he couldn't control the aircraft's attitude previously.

Aileron Partial Power Loss

With this test a break in the power supply to both right and left ailerons is simulated, making a change in the actuator dynamic. Since actuators are simulated by a first order model with settling time ($t_s = 0.05s$), the failure is activated by changing this value to ($t_s = 3.5s$), what

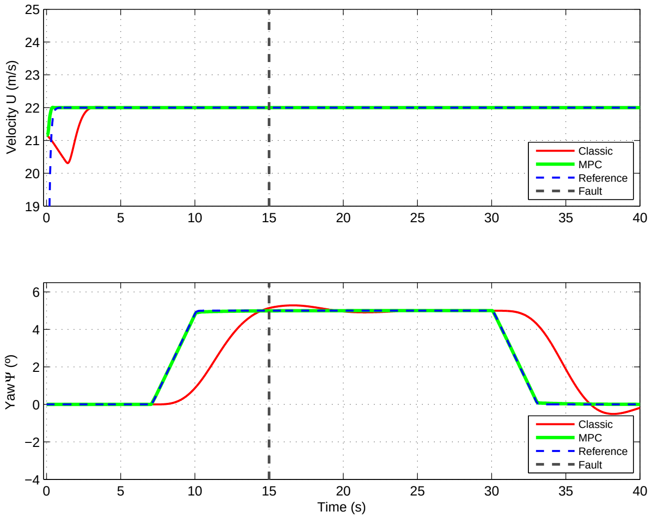


Figure 5.8: Aileron *Frozen* Reference Variables

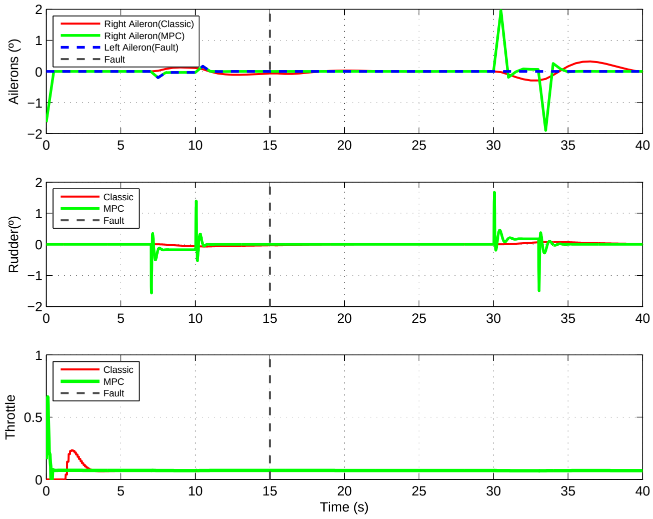


Figure 5.9: Aileron *Frozen* Actuation

makes the ailerons very slow to operate. All the other test parameters where maintained. As shows figure 5.10, once more the MPC controller is capable to deal with this severe fault and maintain satisfactory, the follow up of Yaw angle in contrast to the other controller that initially performs well, but the in the end Yaw angle control becomes unstable. MPC controller's reconfiguration strategy was to use manly the rudder to perform all the manoeuvres after the failure, because of that the aircraft's attitude stays at the desired set point (figure 5.11).

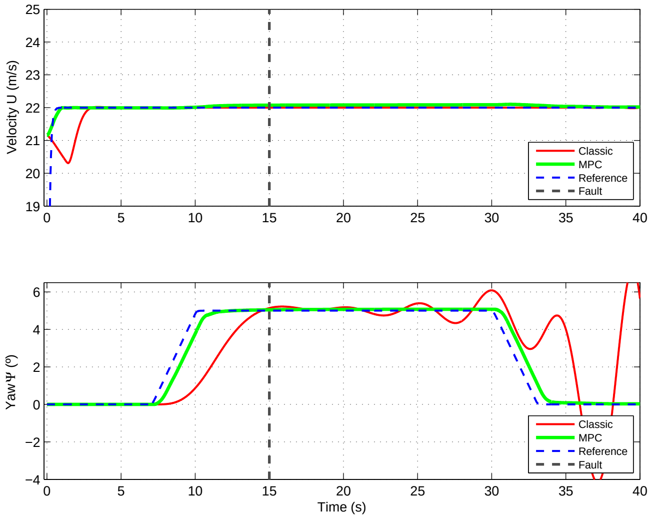


Figure 5.10: Aileron *Partial Power Loss* Reference Variables

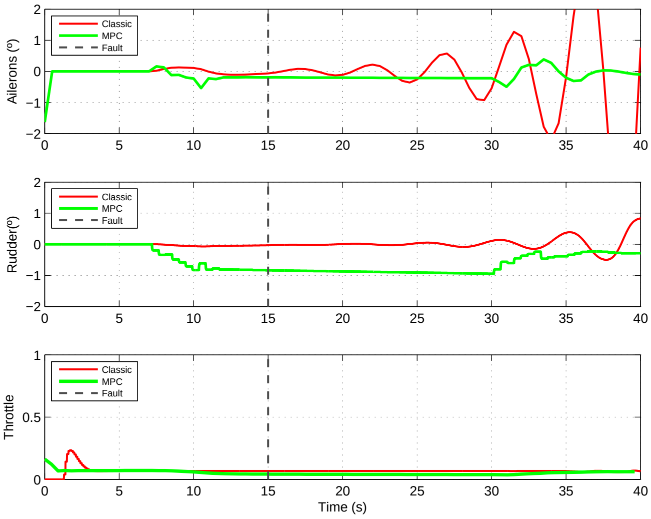


Figure 5.11: Aileron *Partial Power Loss* Actuation

5.2.2 Elevator Failure

Usually the elevators operate in a coupled mode, when a failure occurs for example the elevators are both negatively deflected, in this particular case the aircraft has no other actuator capable to counteract this situation. So to simulate Full Power Loss failures only one elevator is affected, a realistic event on a bigger aircraft which has multiple-segmented actuator surfaces. In this particular test the MPC *Pitch Hold* controller and a *Classic* controller are compared, they must be able to command the aircraft from a level flight trajectory ($\theta = 0.3°$) to a steady state

($\theta = 5°$), five seconds later the left elevator fails, then controller must be able to maintain the steady value for 10 seconds and later bring the airplane to the initial condition (see figure 5.2). At the same time maintain Yaw angle at ($\psi = 0°$) and *Velocity in* X_E at ($U = 22m/s$). Only the most relevant data is shown is the following figures, so they are *Velocity in* X_E U, *Pitch angle* θ, Elevators/Flaps Deflections, and Throttle.

Elevator Total Power Loss Hard Over

This test will make one of the elevators to reach is physical maximum deflection 17.5°, this faults like others occurs at 15 seconds, as in other tests.

Analysing the figure 5.12, only MPC has a good and effective control of the proposed outputs.

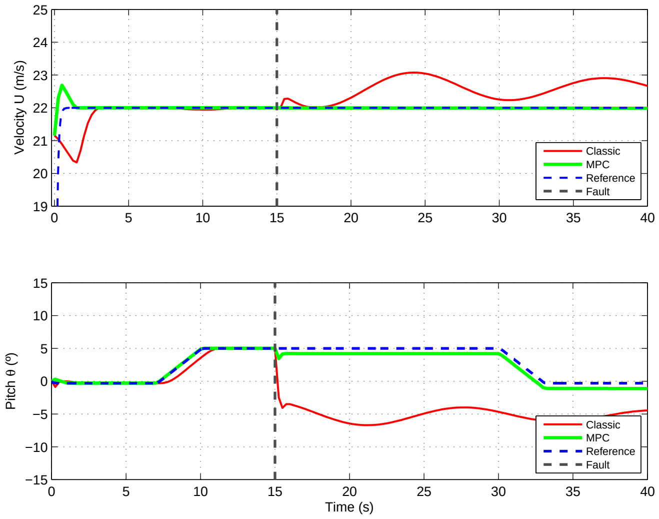


Figure 5.12: Elevator at 17.5° Reference Variables

However a steady state error is present after the hard over deflection. Because both elevators have the same signal convention, the right elevator is commanded to the opposite position to compensate the right one, what makes sense, being both commands MPC and *Classic*, almost the same (see figure 5.13). Major differences appear in the next two actuators, since the MPC reconfiguration strategy is capable to use Flaps to compensate the negative Pitching Moment induced by the defection of the elevator. The steady state error results from the continuous use of Flaps to control pitch angle what is in fact the only solution.

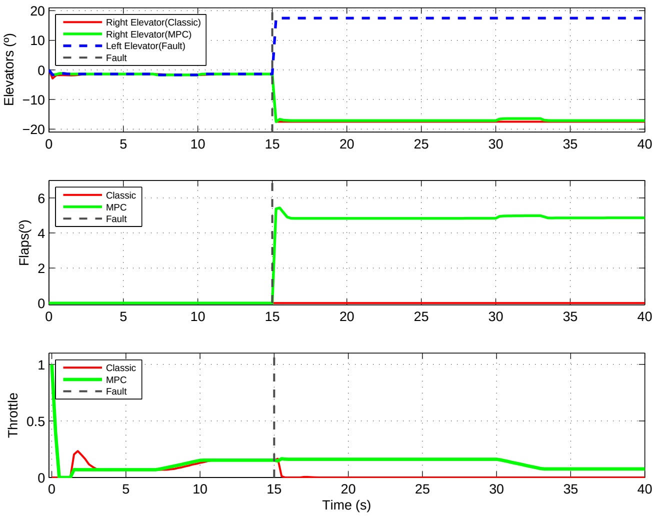


Figure 5.13: Elevator at 17.5° Actuation

Elevator Total Power Loss Hard Under

The opposite test is performed, leading the left elevator to his negative limit -17.5°.

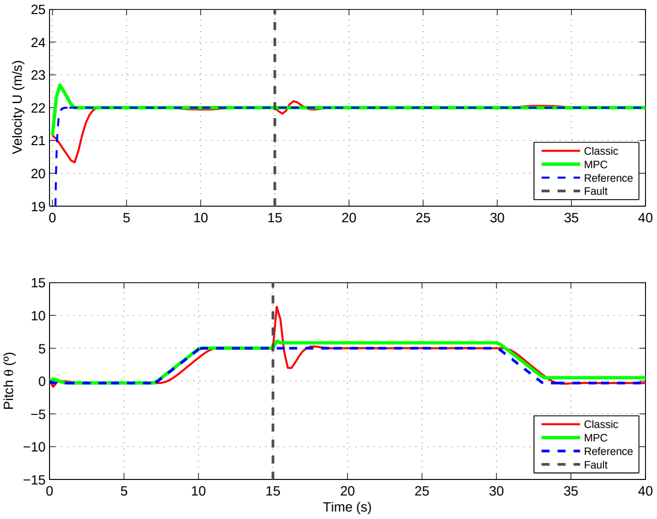


Figure 5.14: Elevator at -17.5° Reference Variables

In the present case it is possible to conclude that the two controllers compensate adequately the elevator actuator damage. Nevertheless MPC controller presents again a steady state error, but the immediate response to the failure is worst using a *Classic* controller (see figure 5.14). As figure 5.15 shows controller behave similarly, the MPC increases slightly Throttle and uses the healthy elevator to reduce the positive pitch moment.

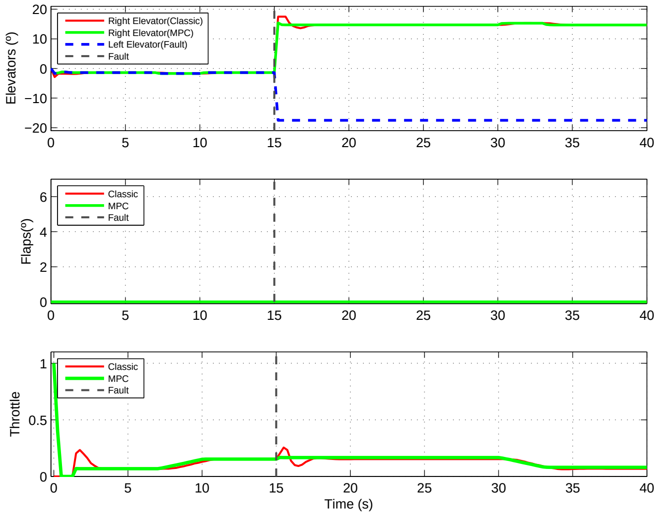


Figure 5.15: Elevator at -17.5° Actuation

Elevator Total Power Loss Frozen

In this case the left elevator is kept stationary since 15 seconds of simulation. As expected both controllers track Velocity and Pitch angle reference, bringing back the aircraft to the initial state, by simply reducing engine power (see figure 5.16). Both controllers apply the right elevator and throttle to perform this simulation as shown in figure 5.17.

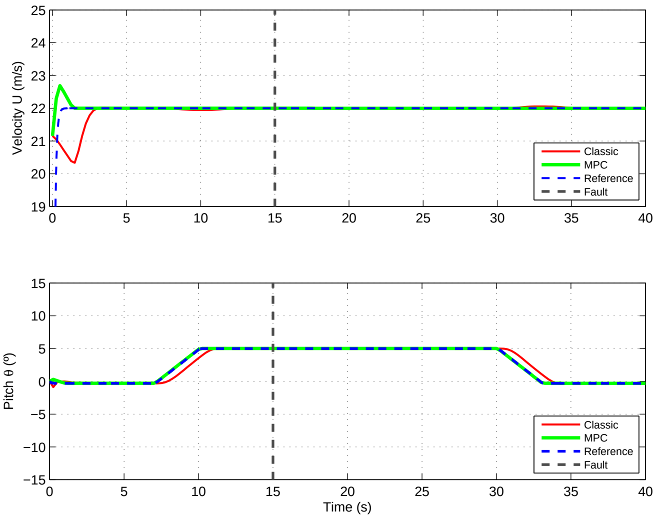


Figure 5.16: Elevator *Frozen* Reference Variables

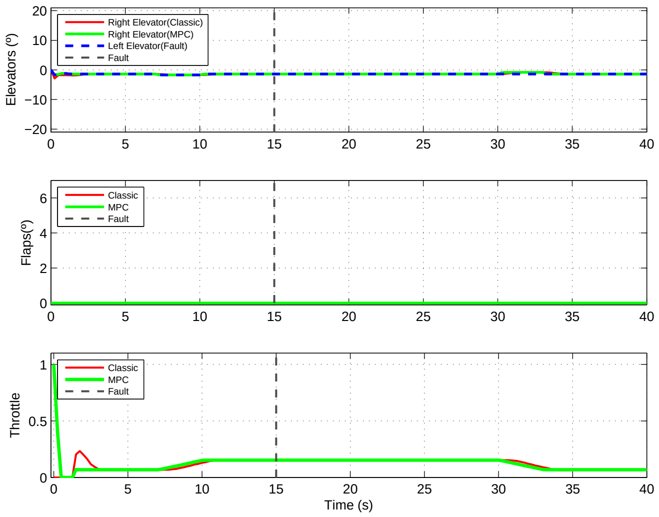


Figure 5.17: Elevator *Frozen* Actuation

Elevator Partial Power Loss

As in the aileron experience, now both elevators are affected by this failure. So by the $15th$ second the elevator response time is increased up to $(t_s = 3.5s)$.

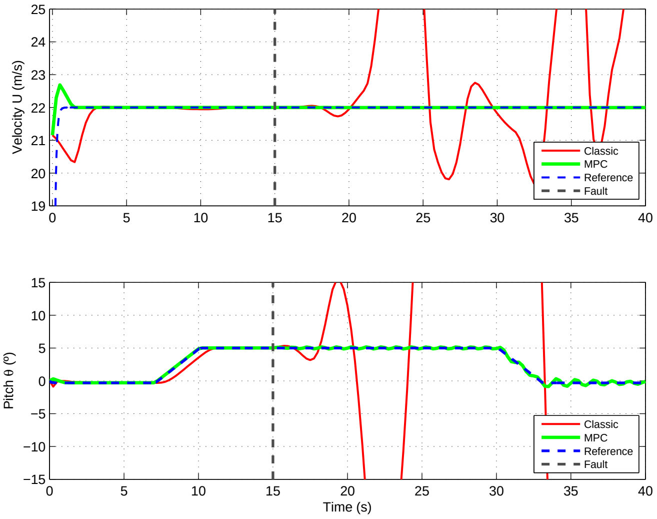


Figure 5.18: Elevator *Partial Power Loss* Reference Variables

As figure 5.18 demonstrates MPC is fully capable to deal with this kind of failures, in counterpoint the *Classic* controller makes the aircraft to enter in great instability. Analysing actuator graphics (figure 5.19) it is very simple to conclude that the time lag in response induces a saturation on the *Classic* controller. MPC suffers a slight instability, in order to

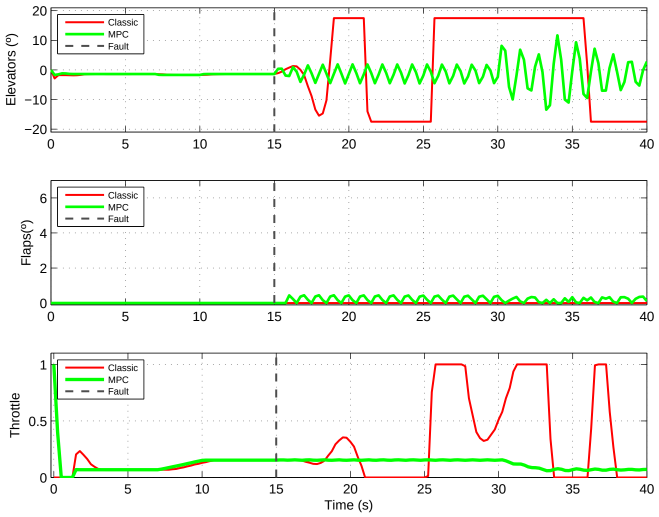


Figure 5.19: Elevator *Partial Power Loss* Actuation

track down Pitch angle, yet maintains control satisfactory.

5.2.3 Rudder Failure

The goal of the rudder failure contingency controller is to successfully command the ailerons of the aircraft when a fault in the rudder has occurred. The contingency controller in the case of a failure on the rudder is totally equivalent to the *Bank Hold* controller presented in the previous subsection. Its objective is to provide a zero steady state error during a manoeuvre that start at ($\psi = 0°$) to a steady state ($\psi = 5°$), five seconds later the rudder fails and the controller must be able to maintain the steady value and 10 seconds later bring the airplane to the initial condition, maintaining Pitch angle at ($\theta = 0.3°$) and *Velocity in* X_E at ($U = 22m/s$) as shown in figure 5.3. Since results where so catastrophic using the maximum physical limits of this actuator, lower values [$4^o; -4^o$] were selected to perform Hard Over and Hard Under simulation. The data presented is the same as to aileron failure.

Rudder Total Power Loss Hard Over

Rudder in this aircraft is single surface, so when the fault occur the present actuator is driven to its maximum position 4°. Results obtained for this test, show a velocity steady state low error, but Yaw angle look slightly oscillatory in the MPC controlled case (see figure 5.20). The actuator have a similar values, still MPC increases throttle and the *Classic* controller not.

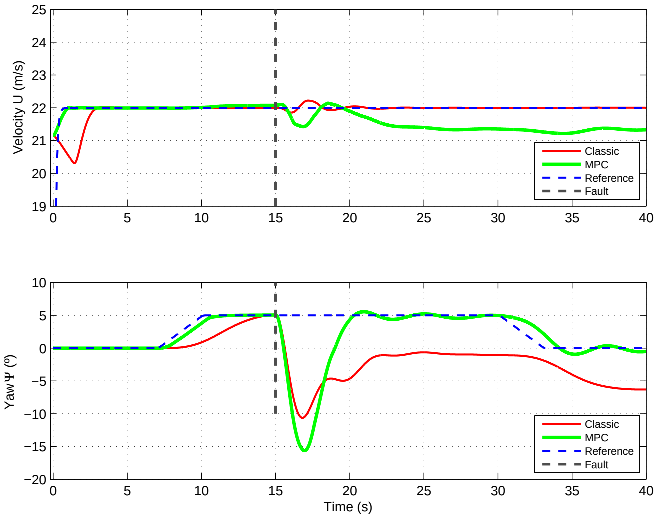


Figure 5.20: Rudder at 4° Reference Variables

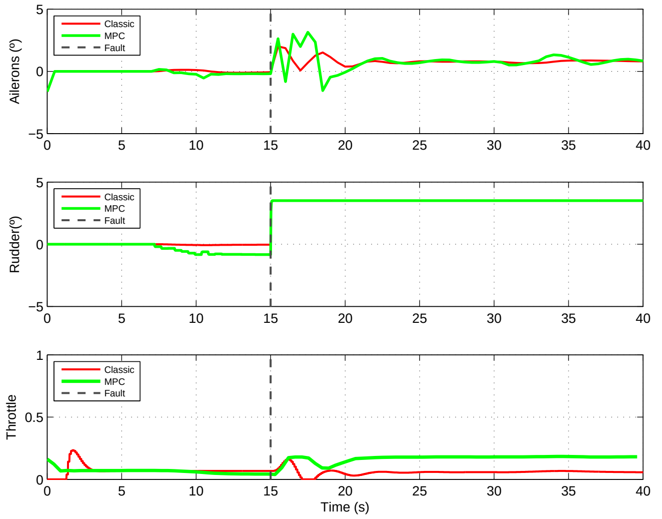


Figure 5.21: Rudder at 4° Actuation

The aileron actuation is harder in MPC giving the idea that the aircraft is unstable with this controller as shows figure 5.21.

Rudder Total Power Loss Hard Under

Now we perform the opposite case (negative deflection), using the same fault time.

Analysing figure 5.22, no controller can control effectively velocity and Yaw angle. Though MPC recorvers from the yaw moment injected, gains a large steady state error ($\approx 1m/s$) in the

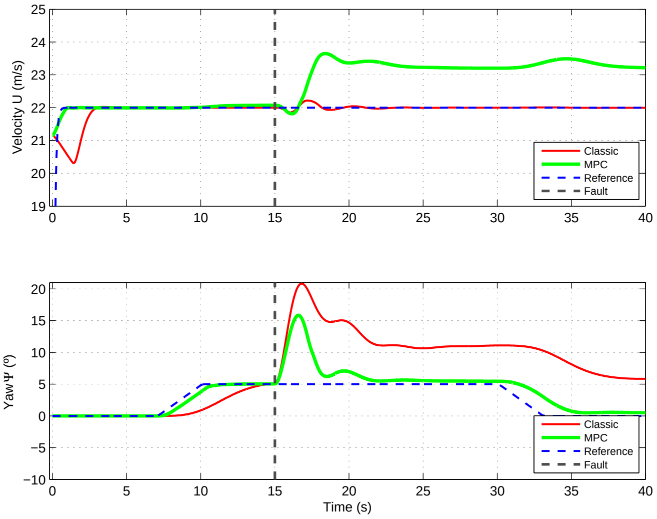


Figure 5.22: Rudder at -4° Reference Variables

velocity control. Analysing the actuation figure 5.23, its possible to distinguish that Throttle

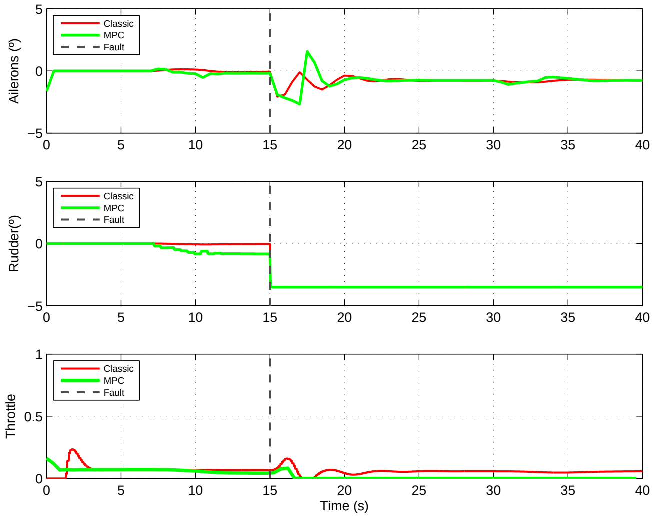


Figure 5.23: Rudder at -4° Actuation

commands are different. MPC almost turns off the engine reason why there's a bad velocity control, however this controller has a more aggressive approach when using ailerons.

Rudder Total Power Loss Frozen

The simulation is the same as in 5.2.1, but this time is the rudder that gets stuck in fixed deflection angle. MPC finally seams to overcome this failure, achieving velocity complete

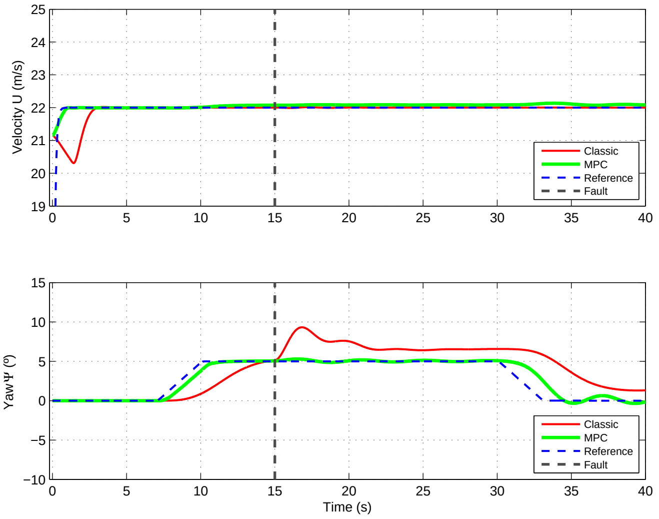


Figure 5.24: Rudder *Frozen* Reference Variables

control, and yaw angle tracking, even so isn't clear if there exists a certain degree of instability (see figure 5.24). From these values shown in figure 5.25, MPC reduces Throttle comparing

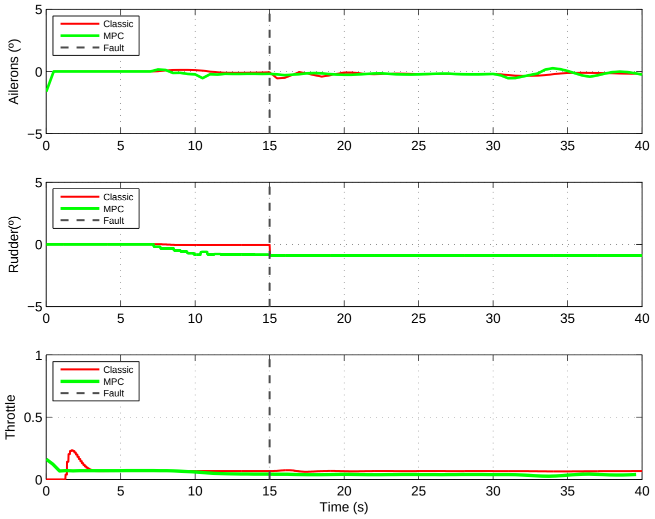


Figure 5.25: Rudder *Frozen* Actuation

with the *Classic* case, and uses aileron more aggressively as in previous simulations.

Rudder Partial Power Loss

This experiment will change rudder dynamics after the 15 seconds, becoming then a first order system with ($t_s = 3.5s$). During this simulation both controllers have good control

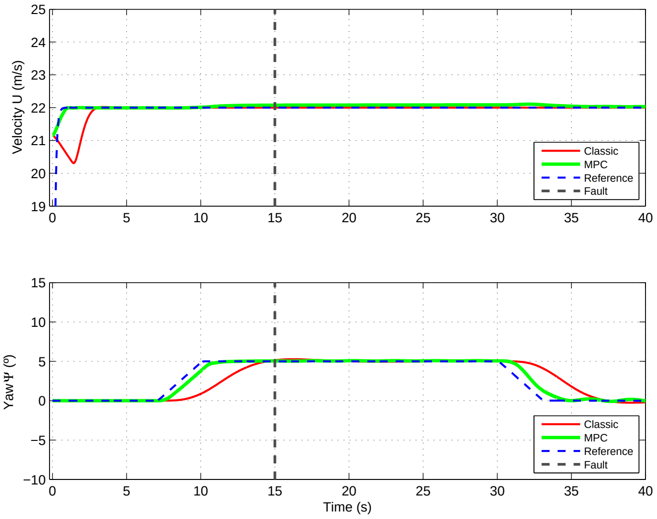


Figure 5.26: Rudder *Partial Power Loss* Reference Variables

of the aircraft(see figure 5.26). Nevertheless is possible to notice a slower transient response at 30 seconds when the Yaw angle is requested to return to the initial position. The MPC

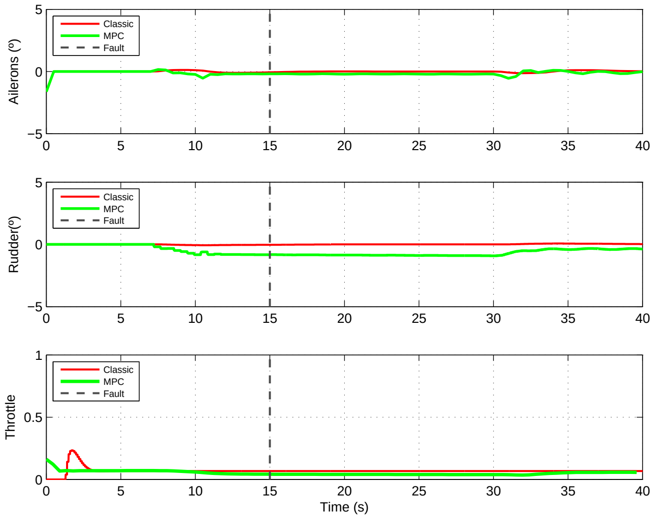


Figure 5.27: Rudder *Partial Power Loss* Actuation

reconfiguration strategy, coordinates the slower rudder and the ailerons, *Classic* controller only uses the ailerons to finish the test demands in terms of Yaw tracking, as visible in figure 5.27.

5.2.4 Flaps Failure

Flaps are surfaces more suited for take-off and landing operation regimes. So to perform a failure simulation with some realistic sense, only the Hard Over situation was chosen to simulate. Since all other described failures have no effect in the aircraft behaviour in the present case.

For instance the *Hard Under* and *Frozen* failure would leave Flaps in their normal position $\eta_f = 0^o$, situation that makes no sense to simulate. The *Partial Power Loss* would be interesting to test in landing procedure but this kind of elaborate experiments are beyond this thesis goals.

The following scenario will test *Pitch Hold* against a *Classic* controller in the same profile test used before for Elevator failures tests (see figure 5.2).

Flaps Total Power Loss Hard Over

At 15 seconds a Hard Over Flaps is introduced in the model leading this pair of surfaces to their upper limit 45°.

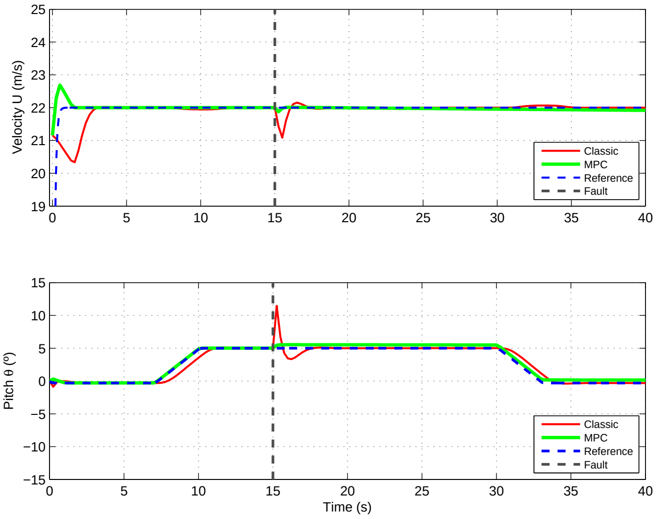


Figure 5.28: Flaps at 45° Reference Variables

Results shows (figure 5.28) that both controllers manage to control velocity and Pitch angle, still MPC is much smother at failure time but possesses a small steady state error in the pitch angle. Actuation values are similar, differing only in the Throttle actuation. The MPC controller uses less engine power, this explains the poorer velocity control (see figure 5.29).

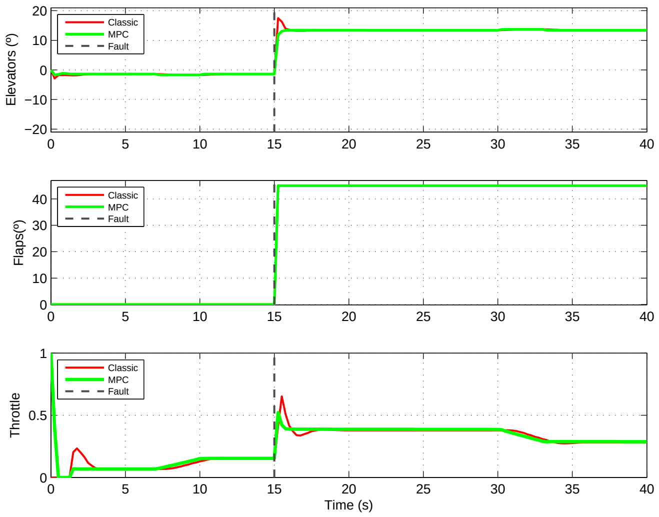


Figure 5.29: Flaps at 45° Actuation

5.2.5 Constraints Update Time Evaluation

An important aspect of MPC controllers is the need for a fault detection and identification system. In this thesis this wasn't really implemented, nevertheless the update constraints block presented in chapter 4 depends on the identification of the *Total Power Loss* failure, in order to change the fault actuator input constraints. In figure 5.30 a simple experiment was conducted to evaluate the effect of a delay in the constraint update system, after a *Flaps Hard Over* failure. This experiment shows the effect of delaying the constraint update form $0s$ to $1s$, in the last case the aircraft reaches a dangerous pitch angle $\theta \approx 90^o$, nevertheless after the update the MPC controller manages to regain the aircraft full control.

From this graphic it is easy to conclude that in a real implementation the FDI system must be chosen very carefully, because much of the controllers performance relies on a rapid and effective fault identification system.

5.3 Trajectory Tracking

Eventhough *Bank Hold* and *Pitch Hold* controllers were developed to work for the previous tests, due to the good results obtained in Aileron, Elevator and Flaps failures, it was decided to test the performance of these controllers, using them with a Guidance System.

In [7] a Guidance System was developed for this particular aircraft, this block gives the con-

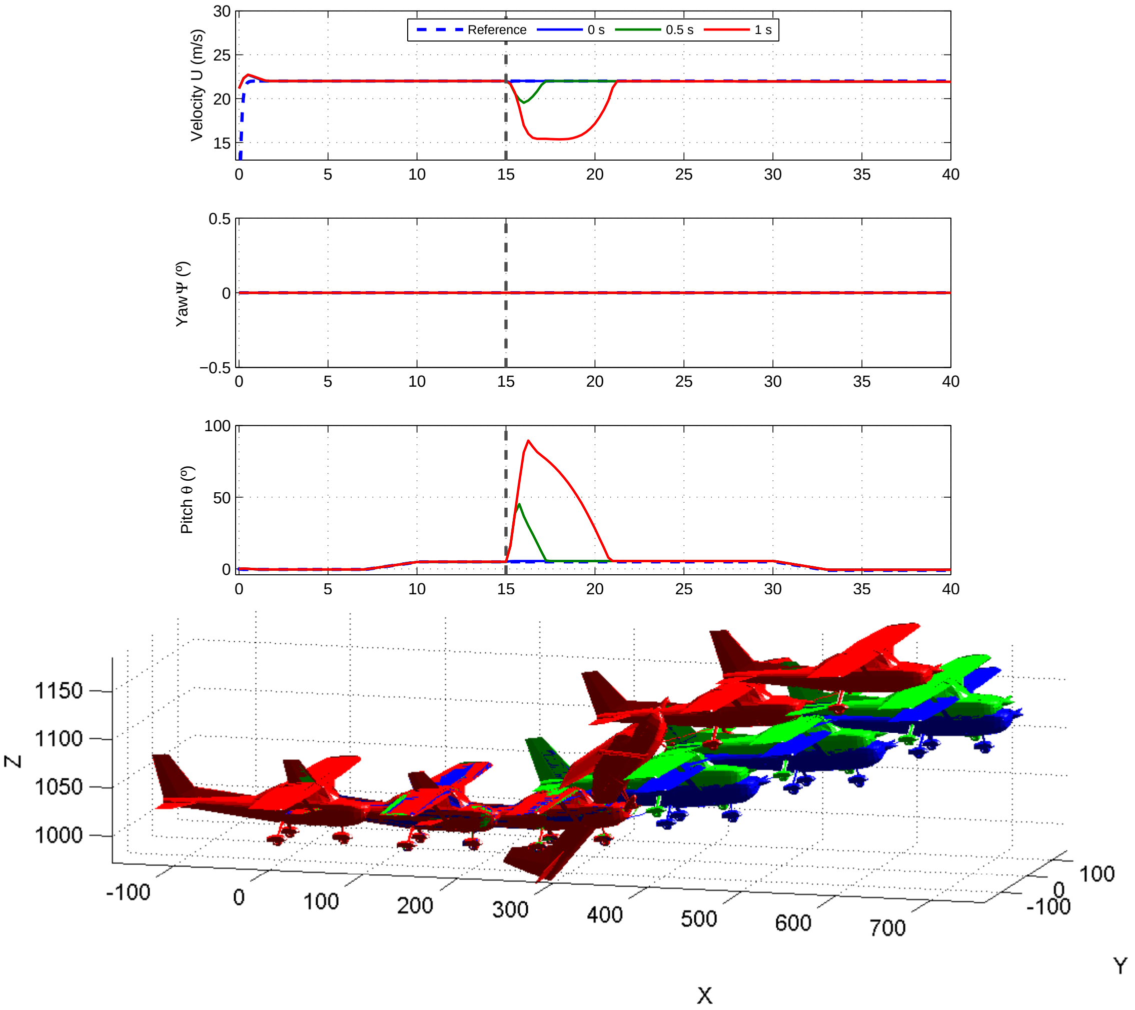


Figure 5.30: Constraints Update Time Evaluation [*aircraft not to scale*]

troller references like Velocity, Pitch angle and Yaw angle. Several trajectories are available, like spiral, sinusoidal , a simple climb manoeuvres.

Since Rudder compensation strategy was very poor, no good results were obtained in these simulations, because of that, experiments with this surface were excluded.

Aileron Total Power Loss Hard Over

The *Bank Hold* controller was used to guide the aircraft in a climbing spiral during 2 *min*, almost at the middle of the experiment an Aileron Hard Over is introduced, the left aileron is driven to its maximum limit 20° .

Considering that the MPC controller wasn't tuned to perform this kind of maneuver, figure

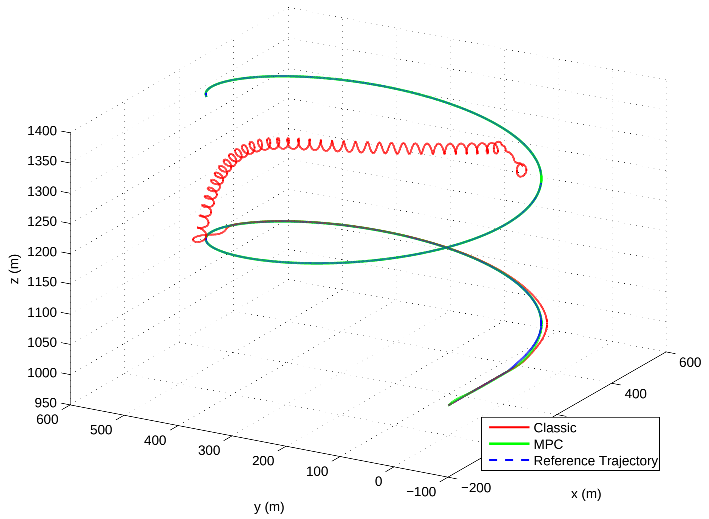


Figure 5.31: Spiral Manoeuvre with Aileron at 20°

5.31 shows an excellent job in controlling the aircraft before the failure, and after the fault the aircraft behave almost as if the failure didn't existed.

Aileron Partial Power Loss

This simulation is equal to the one performed previously in subsection 5.2.1 (Ailerons Failure). To this particular test a horizontal sinusoidal trajectory was chosen. During this simulation the aircraft must maintain always the same altitude. The MPC controller have reasonable

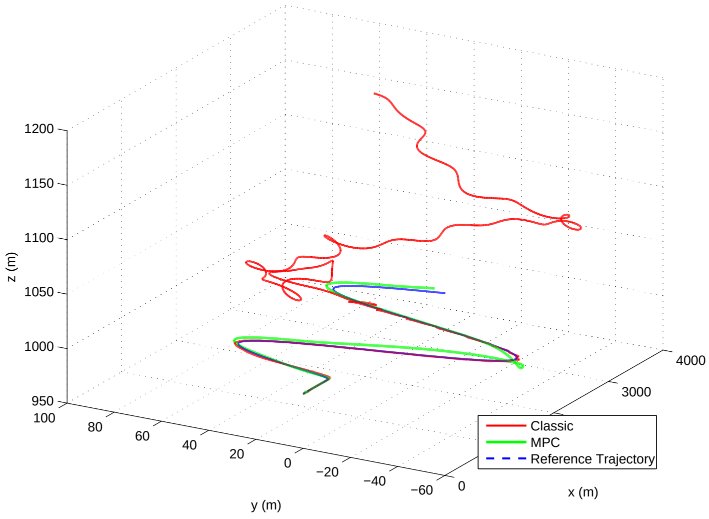


Figure 5.32: Sinusoidal Horizontal Manoeuvre Aileron *Partial Power Loss*

altitude tracking when changing Yaw angle direction, this results from the controller's tuning that favours Yaw control comparing to Pitch and velocity tracking (see figure 5.32).

Elevator Total Power Loss Hard Over

For this failure the *Pith Hold* controller will be tested in the same spiral climbing trajectory. Now the left elevator is driven to 17.5°. Again MPC controller is able control the aircraft during the entire simulation, showing that there's a real gain in using this controller to manage this kind of failures (see figure 5.33).

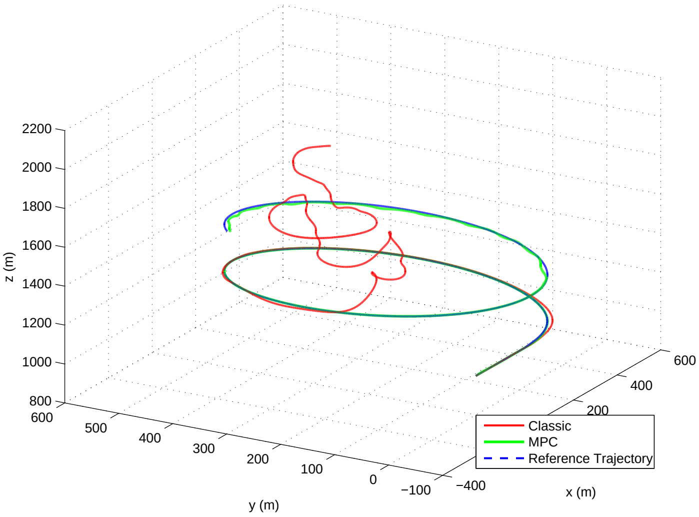


Figure 5.33: Spiral Manoeuvre with Elevator at 17.5°

Elevator Partial Power Loss

In order to evaluate the elevator slower response a Sinusoidal Vertical trajectory was chosen. This time the aircraft has to maintain its direction ($\psi = 0°$), and climb and descend alternately. Fault is introduced in the middle of the test (1 *min*). The results are consistent with the ones obtained in the previous *Elevator Partial Power Loss* test, MPC reacts smoothly and the aircraft is controllable all the time as shown in figure 5.34.

Flaps Total Power Loss Hard Over

The objective of this fault is recreate a take-off maneuver, in which Flaps fails to a Hard Over position. *Pitch Hold* controller must be capable of maintaining straight forward climb and

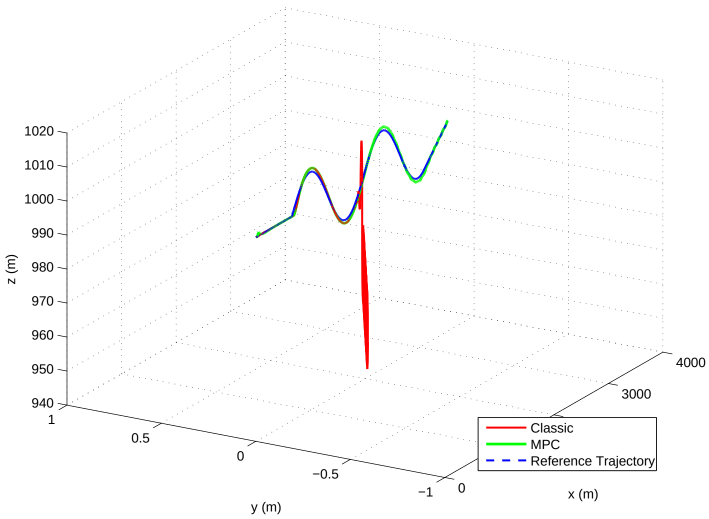


Figure 5.34: Sinusoidal Vertical Manoeuvre Elevator *Partial Power Loss*

descend. Figure 5.35, shows that both controllers have a similar performance, not being able

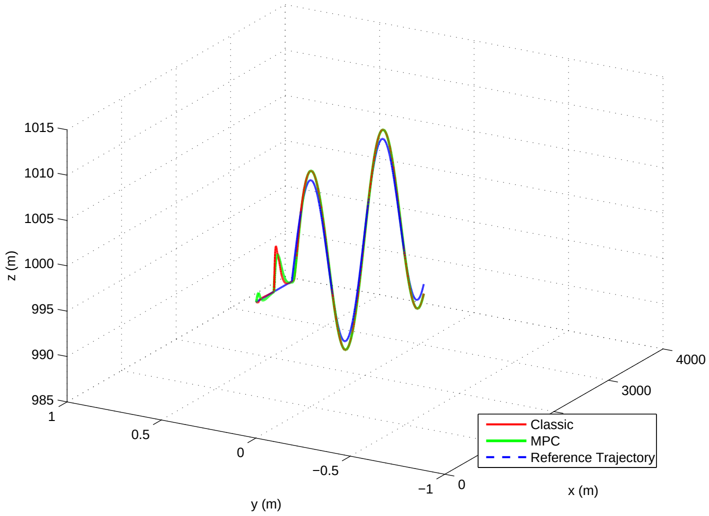
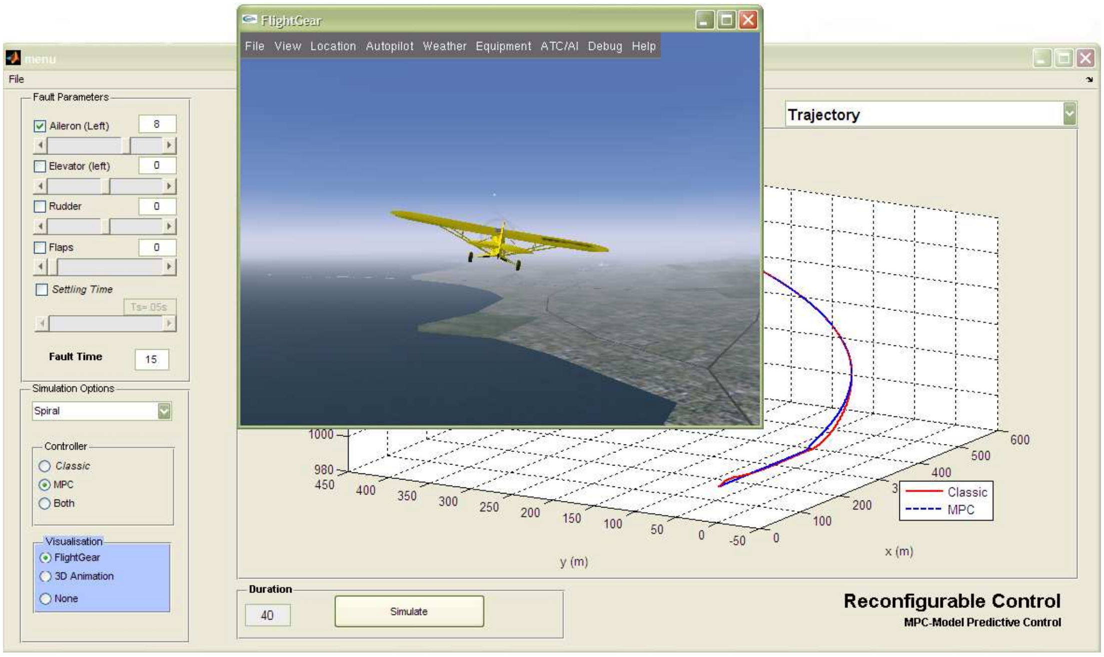


Figure 5.35: Sinusoidal Vertical Manoeuvre Flaps at 45°

react adequately to the initial pitching moment, nevertheless the aircraft tracks satisfactorily the requested trajectory afterwards.

5.3.1 Trajectory Visualisation

In order to give a visual perspective of the overhaul aircraft behaviour when suffers the failure and the controllers reaction to overcome this same failure, a Graphical User Interface (GUI) was developed, further details are described in AppendixD. This application permits the user to chose and configure all the simulation parameters, such as the time fault, the trajectory to follow, the surface and respective deflection. After the simulation is performed in $MatLab$, it is possible to visualise a 3D flight simulation using $FlighGear$. This application is an open source, multi-platform, flight simulation program written in C++ and licensed under GNU General Public License, which means that the source is freely available and modifiable. Figure 5.36 shows both Fault simulation GUI and FlightGear windows.



menu

File

Fault Parameters

☑ Aileron (Left) 8

□ Elevator (left) 0

□ Rudder 0

□ Flaps 0

□ *Settling Time* Ts=.05s

Fault Time 15

Simulation Options

Spiral

Controller

○ *Classic*

● MPC

○ Both

Visualisation

● FlightGear

○ 3D Animation

○ None

Duration 40

Trajectory

Figure 5.36: Fault Simulation Graphical Interface

Chapter 6

Conclusions and Further Remarks

6.1 Conclusions

This thesis provides an approach to the design of a reconfigurable controller using Model Predictive Control. It is shown that in case of substantial control surfaces failures MPC has the ability to reconfigure its control law and maintain acceptable performance.

A unique advantage of the MPC controller presented in this work is the fact that it uses an internal model of the plant to predict the outputs over the prediction horizon P. In case of a failure during flight, the MPC's internal inputs and vehicle states constraints can be updated, MPC controllers can explicitly take the input constraints of each fault and use it to find the optimal strategy, as defined by its cost function for a given failure. For example, constraints can be used to limit the performance of the vehicle in case of a failure in order to safeguard against unforeseen dangers.

In the first place the aircraft model was studied profoundly, and two types of failures were found to be suitable for electrical servos failures. One that represents changes in the actuator dynamics *Partial Power Loss* and another *Total Power Loss* which represents all situations where a control surface becomes stuck in a determined position.

After a Reconfigurable Controllers survey, Model Predictive Control was the chosen method, because MPC controllers can be designed so they have an intrinsic ability to handle jammed actuators without the need to explicitly model the failure.

To test the controllers capabilities two types of simulation were developed. The aircraft should start in a steady level flight and a turning or climbing maneuver was performed, in order to reach the desired Pitch or Yaw angle set point of 5 *degrees*. After a few seconds (15 seconds of simulation), a fault was introduced in the aircraft model, then controllers had to be able to compensate the fault by adjusting the actual aircraft attitude. Then the Pitch or Yaw angle has to return to the nominal position, performing a total of 40 seconds.

Having tested several controllers and failures, two controllers were chosen. One capable of dealing with Elevator and Flaps failures called *Pitch Hold* and another more suited for Aileron and Rudder failures known as *Bank Hold*.

The *Pitch Hold* controller mode was found to recover performance suitably after a very short duration. The aircraft proceeded without aggressive commands in order to regain the nominal condition as fast as possible. In case of a Flaps failure, the recovery strategy was similar as the one used by a *Classic* controller, giving a clear idea, that when Flaps get stuck there's no real need to develop controllers to this specific failure.

The *Bank Hold* controller mode proved to be extremely sensitive reacting very quickly, introducing in some cases a minimal instability. The failure contingency controller proved it was able to bring the aircraft back to wings level. On the other hand in Rudder failure control, the results where never satisfactory, giving no guarantees of being capable of overcoming these faults, so we must conclude that with the attitude requirements (Pitch angle,Yaw angle) and Velocity, it is impossible to deal with this failure. In the case of Rudder failures the MPC controller has difficulty tracking the reference signals. In other cases a larger transient is seen with a steady state error throughout the rest of the maneuvers.

In terms of trajectory tracking surprisingly the developed controllers behave very well, showing good control of the aircraft. An important feature of this control strategy, is the online constraints update in a real implementation this system would have a certain delay. So a small experiment was conducted to evaluate a 1 second delay effect in the performance of the controller, from which it was possible to realise that in a real implementation the FDI system must be chosen very carefully, because much of the controllers performance relies on a rapid and effective fault identification system.

Controllers where able to maintain altitude and the desired attitude, working with the Guidance System, showing MPC controller's ability to reconfigure itself on both longitudinal and directional channels.

	Aileron				Elevator				Rudder				Flaps
	20°	-20°	$\approx 0^o$	$\nearrow t_s$	17.5°	-17.5°	$\approx 0^o$	$\nearrow t_s$	30°	-30°	$\approx 0^o$	$\nearrow t_s$	45°
MPC	●	●	●	●	●	●	●	●	-	-	○	●	●
Classic	-	-	○	-	-	●	●	-	-	-	-	○	●

Table 6.1: *Results Failure Simulation* [*Good* •,*Poor* ○,*Fails* −]

In table 6.1, results from the failures simulation tests are summarised, tests were the controllers proven to control effectively the aircraft before and after the failure have a filled circle, in other hand an empty circle stands for a poorer control. It easy to conclude that we can obtain a real gain in the use of MPC controllers to manage an Elevator or Aileron failures, yet for Rudder faults the aircraft was lost in most of the cases. Finally for Flaps Hard Over, MPC showed that there's no need for contingency strategy concerning these kind of failures.

MPC shows many advantages over other approaches in reconfigurable control. The polynomial methodology is more straightforward than model predictive control methods and can obviously work for failures foreseen, the amount of time creating the controllers is extended greatly as each is tuned for its particular failure.The MPC's advantage to this is that only two controllers are designed and the algorithm itself solves for the new strategy as the failures are encountered. There is great promise in MPC, as faster computers make highly computational control algorithms easier to implement. The design of next generation controllers for aerospace vehicles will then shift to creating higher fidelity models to place inside the MPC engine. This will drastically reduce the amount of time spent designing a control system and reconfiguration capabilities would be possible as long as the model is able to incorporate failure information [20, 26].

6.2 Further Remarks

The process of updating the internal model has many applicable consequences and leads itself to many other types of reconfiguration.

- One of the most relevant aspects of Reconfigurable control is the presence of an efficient FDI system, for this thesis it was assumed to be present. Fault and isolation techniques do not generate fault models instantly, and are often very uncertain with the level of uncertainty decreasing over time. So the introduction of this kind of systems would give a reliable information about the true controller's capabilities to overcome failures.

- A key aspect that most certainly would increase dramatically the controller's capabilities was a non-linear internal model. Aircraft are inherently nonlinear systems and the MPC relies mostly on the capability of his internal model to predict future behaviours of the aircraft, so it is easy to guess the advantages in the use of a nonlinear internal model [9].
- Controllers used in this thesis performed satisfactorily, for this small UAV, however we should get better results using a bigger aircraft. Specially if the aircraft has multiple-segment control surfaces, because in the event of a failure in one of the multiple actuators, the remaining ones are more fitted to compensate the failed one. Other aspects, could be interesting to be studied like multiple engine aircraft, where in the event of a Rudder failure differential thrust could be used to overcome this problem [2].

Bibliography

[1] American airlines inc., dc-10-10, n110aa chicago-0'hare international airport. Technical report, National Transportation Safety Board, 1979.

[2] Emergency landing using thrust control and shift of weight. Technical report, NASA Tech Briefs, Vol. 26, No. 5, May.

[3] Toal Daniel Edin Omerdic Ahmad Hammad, Trevor M. Young. Compensation of jammed control surface of large transport aircraft by control reconfiguration. In *15th Mediterranean Conference on Control Automation*, 2007.

[4] Shuhao Chen. Adaptive control of linear systems with actuator failures. Master's thesis, University of Virginia, 2004.

[5] Kale M.M. Chipperfield A.J. Stabilised mpc formulations for robust reconfigurable flight control. *Elsevier Science*, 2004.

[6] Boeing Company. 737 family. `http://www.boeing.com/commercial/737family/index.html`, November 2008.

[7] Fiúza José e Silva Leonardo. Piloto automático de uma aeronave. Master's thesis, Instituto Superior Técnico, 2005.

[8] Duff Reid Lloyd Etkin Bernard. *Dynamics of Flight - Stability and Control.* Wiley, 3rd edition, 1996.

[9] Frank Findeisen Rolf. An introduction to nonlinear model predictive control. In *21st Benelux Meeting on Systems and Control*, 2002.

[10] R. Fletcher. *Pratical Methods of Optimization.* John Wiley and Sons, Chichester, UK, 1987.

[11] Flight Safety Foundation. The aviation safety network. `http://aviation-safety.net`, November 2008.

[12] Elling W. Jacobsen. Model predictive control- control theory and practice, February 2005.

[13] Maciejowski J.M. Fault-tolerant aspects of mpc. *IEE*, 1999.

[14] Maciejowski J.M. Modelling and predictive control: Enabling technologies for reconfiguration. *Elsevier Science*, 1999.

[15] Colin N. Jones. Reconfigurable flight control first year report. Technical report, Pembroke College, 2005.

[16] M.M. Kale and A.J. Chipperfield. Reconfigurable flight control strategies using model predictive control. *Proceedings of 2002 IEEE*, 2002.

[17] Maciejowski Kerrigan E.C. Fault-tolerant control of a ship propulsion system using model predictive control. 1998.

[18] Rajeeva Kumar. Failure detection, isolation and estimation for flight control surfaces and actuator. Technical report, Flight Mechanics Control Division, 1997.

[19] Liu Chun-Te Lin Chun-Liang. Failure detection and adaptive compensation for fault tolerant flight control systems. *IEEE*, 2007.

[20] Christopher M. Shearer. Constrained model predictive control of a nonlinear aerospace. Master's thesis, Technology Air University, 1997.

[21] Huzmezan M. Maciejowski J.M. Reconfigurable flight control of a high incidence research model using predictive control. In *UKACC International Conference CONTROL'98*, 1998.

[22] Kerrigan E.C. Maciejowski J.M., Goulart P.J. *Constrained Control Using Model Predictive Control.* 2004.

[23] Lemos J.M. Maciejowski J.M. *Predictive methods in fault-tolerant control.* Prentice Hall, 2002.

[24] Mathworks. *Model Predictive Control Toolbox for use with Matlab*, 2004.

[25] Mathworks. *Simulink Simulation and Model-Based Design*, 2004.

[26] Donald Mclean. *Automatic Flight Control Systems.* Prentice Hall International, U.K., 1990.

[27] Evangelos Papadopoulos and C. Chasparis Georgios. Analysis and model-based control of servomechanism with friction. In *International Conference on Intelligent Robots and Systems (IROS)*, 2002.

[28] Azinheira José R. *Controlo de Voo.* Instituto Superior Técnico, 2006.

[29] Graham R. Drozeski. *A Fault-Tolerant Control Architecture for Unmanned Aerial Vehicles.* PhD thesis, University of Virginia, 2005.

[30] R. Rato L.M. e Neves da Silva. Piper pa 18 super cub 1/4 escala - modelo não linear. Master's thesis, INESC, 1993.

[31] Nelson Robert. *Flight Stability and Automatic Control.* McGraw Hill, 2nd edition, 1998.

[32] S. Eick Robert. A reconfiguration scheme for flight control adaptation to fixed-position actuation failures. Master's thesis, University of Florida, 2003.

[33] J. Roskam. *Airplane Flight Dynamics and Automatic Flight.* Aviation and Engineering Corporation, Ottawa, 1979.

[34] Michael Elgersma Sonja Glavaški. Reconfigurable control for active management of aircraft system failures. In *American Control Conference*, 2001.

[35] Brian Stevens and Lewis. *Aircraft Control and Simulation.* Wiley, 2nd edition, 2003.

[36] Zhang Youmin and Jiang Jin. Bibliographical review on reconfigurable fault-tolerant control systems. In *Annual Reviews in Control*, 2008.

[37] Jiang Jin Zhang Youmin. Bibliographical review on reconfigurable fault-tolerant control systems. In *IFAC Symposium on Fault Detection*, 2003.

Appendices

Appendix A

Aircraft Model

This appendix presents all *Simulink* blocks used in simulations, all equations and parameters needed to operate the system. The Guidance system is based in [7] and Basic Aircraft Model is based on [30].

A.1 Global System in Simulink

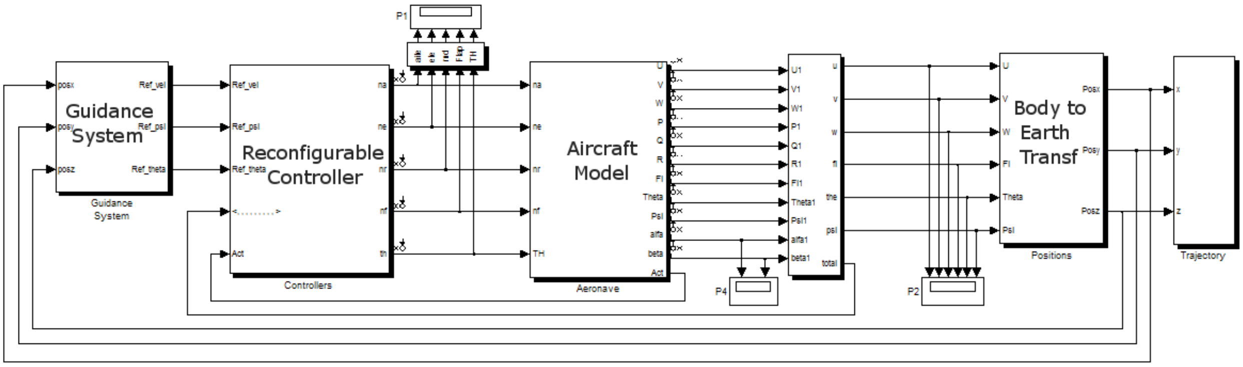


Figure A.1: Global System in *Simulink*

A.2 Aircraft Model in Simulink

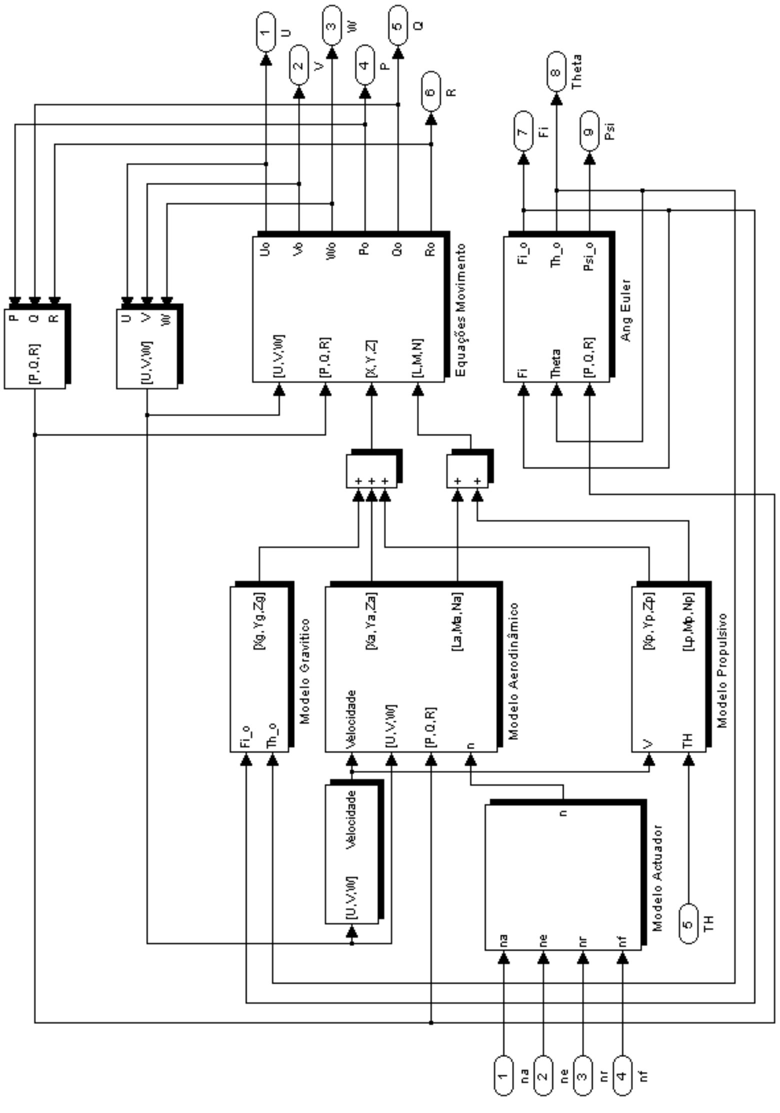


Figure A.2: Aircarft Model Block Diagram *Simulink*

This model is formed by the main blocks, Gravity Block, Aerodynamic Block and Propulsive. They are responsible for calculating forces (X, Y e Z) and and for the external moments (L, M e N). Linear velocities (U, V e W) and rotational ones (P, Q e R) are obtained integrating the previous forces and moments. Euler angle θ, ϕ e ψ calculations are performed by a specific block using rotational velocities.

A.3 Model Mathematical Formulations

$$V_x^E = c(\psi)c(\theta)U + [c(\psi)s(\theta)s(\phi) - s(\psi)c(\phi)]V + [c(\psi)s(\theta)c(\phi) + s(\psi)s(\phi)]W \quad (A.1)$$

$$V_y^E = s(\psi)c(\theta)U + [s(\psi)s(\theta)s(\phi) + c(\psi)c(\phi)]V + [s(\psi)s(\theta)c(\phi) - c(\psi)s(\phi)]W \quad (A.2)$$

$$V_z^E = -s(\theta)U + c(\theta)s(\phi)V + c(\theta)c(\phi)W \quad (A.3)$$

$$\dot{U} = RV - QW + \frac{1}{m}X \quad (A.4)$$

$$\dot{V} = PW - RU + \frac{1}{m}Y \quad (A.5)$$

$$\dot{W} = QU - PV + \frac{1}{m}Z \quad (A.6)$$

$$\dot{P} = i_1PQ + i_2QR + i_3L + i_4N \quad (A.7)$$

$$\dot{Q} = i_5PR + i_6(R^2 - P^2) + i_7M \quad (A.8)$$

$$\dot{R} = i_8PQ + i_9QR + i_{10}L + i_{11}N \quad (A.9)$$

$$\dot{\phi} = P + R\tan(\theta)\cos(\phi) + Q\tan(\theta)\sin(\phi) \quad (A.10)$$

$$\dot{\theta} = Q\cos(\phi) - R\sin(\phi) \quad (A.11)$$

$$\dot{\psi} = R\frac{\cos(\phi)}{\cos(\theta)} + Q\frac{\sin(\phi)}{\cos(\theta)} \quad (A.12)$$

$$X = X_g + X_a + X_p \quad (A.13)$$

$$Y = Y_g + Y_a + Y_p \quad (A.14)$$

$$Z = Z_g + Z_a + Z_p \quad (A.15)$$

$$L = L_a + L_p \quad (A.16)$$

$$M = M_a + M_p \quad (A.17)$$

$$N = N_a + N_p \quad (A.18)$$

$$X_g = -mg\sin(\theta) \quad (A.19)$$

$$Y_g = mg\cos(\theta)\sin(\phi) \quad (A.20)$$

$$Z_g = mg\cos(\theta)\cos(\phi) \quad (A.21)$$

$$X_a = \mathcal{L}_W \sin(\alpha) + \mathcal{L}_F \sin(\beta) - (\mathcal{D}_W + \mathcal{D}_B)\cos(\alpha)\cos(\beta) + \mathcal{L}_T \sin(\alpha + \alpha_W) \quad (A.22)$$

$$Y_a = -(\mathcal{D}_W + \mathcal{D}_B)\cos(\alpha)\sin(\beta) - \mathcal{L}_F \cos(\beta) \quad (A.23)$$

$$Z_a = -\mathcal{L}_W \cos(\alpha) - \mathcal{L}_T \cos(\alpha + \alpha_W) - (\mathcal{D}_W + \mathcal{D}_B)\sin(\alpha)\cos(\beta) \quad (A.24)$$

$$\alpha = \frac{W}{U} \quad (A.25)$$

$$\beta = \frac{V}{U} \quad (A.26)$$

$$C_{L_W} = a_W(\alpha - \alpha_{L0} + \Delta_f \eta_f) \quad (A.27)$$

$$\alpha_W = -K_1 C_{L_W} + K_2 \frac{\dot{W} l_T}{\mathcal{V}^2} \quad (A.28)$$

$$\mathcal{L}_W = q S_W C_{L_W} \quad (A.29)$$

$$\mathcal{L}_T = q S_T a_T(\alpha + \alpha_W + \varepsilon_T + \Delta_e \eta_e + \frac{Q l_T}{\mathcal{V}}) \quad (A.30)$$

$$\mathcal{L}_F = q S_F a_F(\beta + \Delta_r \eta_r - \frac{R l_F}{\mathcal{V}} + \frac{P \gamma_P}{\mathcal{V}}) \quad (A.31)$$

$$\mathcal{D}_W = q S_W (C_{\mathcal{D}_0} + \frac{C_{L_W}^2}{\pi A_W e}) \quad (A.32)$$

$$\mathcal{D}_B = q S_{ref} C_{\mathcal{D}_B} \quad (A.33)$$

$$q = \frac{1}{2}\rho \mathcal{V}^2 \quad (A.34)$$

$$\mathcal{V} = \sqrt{U^2 + V^2 + W^2} \quad (A.35)$$

$$L_a = L_W + L_F \quad (A.36)$$

$$M_a = M_W + M_T + M_B \quad (A.37)$$

$$N_a = N_W + N_F + N_B \quad (A.38)$$

$$L_W = qS_W b\Delta_{l_a}\eta_a + qS_W b\gamma_9\beta + [\gamma_4 + \gamma_5(\alpha - \alpha_{L0})]\rho\mathcal{V}R + \gamma_1\rho\mathcal{V}P \tag{A.39}$$

$$L_F = -h_F\mathcal{L}_F \tag{A.40}$$

$$M_W = qS_W cC_{M_{ac}} - l_W\mathcal{L}_W \tag{A.41}$$

$$M_T = -l_T\mathcal{L}_T \tag{A.42}$$

$$M_B = qK_{M_B}\alpha \tag{A.43}$$

$$\begin{aligned} N_W = {} & qS_W b\Delta_{n_a}\eta_a + qS_W b[\gamma_{10} + \gamma_{11}(\alpha - \alpha_{L0})]\beta + \\ & [\gamma_6 + \gamma_7(\alpha - \alpha_{L0}) + \gamma_8(\alpha - \alpha_{L0})^2]\rho\mathcal{V}R + [\gamma_2 + \gamma_3(\alpha - \alpha_{L0})]\rho\mathcal{V}P \end{aligned} \tag{A.44}$$

$$N_F = l_F\mathcal{L}_F \tag{A.45}$$

$$N_B = -qK_{N_B}\beta \tag{A.46}$$

$$\Delta_{n_a} = K_3 C_{L_W} \tag{A.47}$$

$$X_p = T \tag{A.48}$$

$$Y_p = T\sin(\varepsilon_y) \tag{A.49}$$

$$Z_p = T\sin(\varepsilon_z) \tag{A.50}$$

$$L_p = Q \tag{A.51}$$

$$M_p = -l_p T\sin(\varepsilon_z) \tag{A.52}$$

$$N_p = l_p T\sin(\varepsilon_y) \tag{A.53}$$

$$\dot{T} = \frac{1}{K_e}P_{max}\eta_P T_H - \frac{TV_0}{K_e} \tag{A.54}$$

$$Q = \frac{P_{max}P_P}{2\pi}\frac{T_H}{V_0} \tag{A.55}$$

$$V_0 = \frac{\mathcal{V}}{2} + \sqrt{\frac{T}{2\rho S_d} + \frac{\mathcal{V}^2}{4}} \tag{A.56}$$

$$\dot{\underline{\eta}} = A_{act}\underline{\eta} + B_{act}\underline{\eta}^* \tag{A.57}$$

$$\underline{\eta} = \left[\begin{array}{cccc} \eta_a & \eta_e & \eta_r & \eta_f \end{array}\right]' \tag{A.58}$$

$$\underline{\eta}^* = \left[\begin{array}{cccc} \eta_a^* & \eta_e^* & \eta_r^* & \eta_f^* \end{array}\right]' \tag{A.59}$$

A.4 Parameters

$\rho = 1.23\ Kgm^{-3}$

$g = 9.8065\ ms^{-2}$

$m = 10.5\ Kg$

$I_{xx} = 1.9\ Kgm^2$

$I_{yy} = 2.5\ Kgm^2$

$I_{zz} = 3.5\ Kgm^2$

$I_{xz} = 0.052\ Kgm^2$

$i_1 = \frac{(I_{zz}+I_{xx}-I_{yy})I_{xz}}{I_{xx}I_{zz}-I_{xz}^2}$

$i_2 = \frac{I_{yy}I_{zz}-I_{zz}^2-I_{xz}^2}{I_{xx}I_{zz}-I_{xz}^2}$

$i_3 = \frac{I_{zz}}{I_{xx}I_{zz}-I_{xz}^2}$

$i_4 = \frac{I_{xz}}{I_{xx}I_{zz}-I_{xz}^2}$

$i_5 = \frac{I_{zz}-I_{xx}}{I_{yy}}$

$i_6 = \frac{I_{xz}}{I_{yy}}$

$i_7 = \frac{1}{I_{yy}}$

$i_8 = \frac{I_{xx}^2-I_{yy}I_{xx}+I_{xz}^2}{I_{xx}I_{zz}-I_{xz}^2}$

$i_9 = \frac{(I_{yy}-I_{zz}-I_{xx})I_{xz}}{I_{xx}I_{zz}-I_{xz}^2}$

$i_{10} = \frac{I_{xz}}{I_{xx}I_{zz}-I_{xz}^2}$

$i_{11} = \frac{I_{xx}}{I_{xx}I_{zz}-I_{xz}^2}$

$S_W = 1.04\ m^2$

$b = 2.7\ m$

$c = 0.4\ m$

$l_W = 0.0\ m$

$A_W = 7.0$

$a_W = 4.7\ rad^{-1}$

$\alpha_{L0} = -0.082\ rad$

$C_{M_{ac}} = -0.065$

$\Delta_f = 0.27$

$\Delta_{l_a} = 0.54\ rad^{-1}$

$K_3 = 0.03$

$C_{\mathcal{D}_0} = 0.007$

$e = 0.90$

$S_T = 0.19\ m^2$

$a_T = 3.8\ rad^{-1}$

$\varepsilon_T = 0.017\ rad$

$\Delta_e = 0.75$

$l_T = 1.0\ m$

$S_F = 0.09\ m^2$

$a_F = 1.8\ rad^{-1}$

$\Delta_r = -0.78$

$l_F = 1.0\ m$

$h_F = 0.17\ m$

$K_1 = 0.086\ rad$

$K_2 = 0.4\ rad$

$S_{ref} C_{\mathcal{D}_B} = 0.014\ m^2$

$K_{M_B} = 0.058 m^3\ rad^{-1}$

$K_{N_B} = 0.17 m^3\ rad^{-1}$

$\gamma_1 = -2.0\ m^4 rad^{-1}$

$\gamma_2 = 0.17\ m^4 rad^{-1}$

$\gamma_3 = -3.0\ m^4 rad^{-2}$

$\gamma_4 = 0.16\ m^4 rad^{-1}$

$\gamma_5 = 1.6\ m^4 rad^{-2}$

$\gamma_6 = -0.018\ m^4 rad^{-1}$

$\gamma_7 = -0.14\ m^4 rad^{-2}$

$\gamma_8 = -0.086\ m^4 rad^{-3}$

$\gamma_9 = -0.028\ rad^{-1}$

$\gamma_{10} = -0.0022\ rad^{-1}$

$\gamma_{11} = -0.053\ rad^{-2}$

$\gamma_P = 0.43\ m$

$\varepsilon_y = 2^o = 0.035\ rad$

$\varepsilon_z = 1^o = 0.017\ rad$

$l_p = 0.4\ m$

$P_{max} = 3400\ W$

$\eta_P = 0.8$

$P_P = 0.5$

$S_d = 0.20\ m^2$

$K_e = 2.0\ m^2$

$$A_{act} = -B_{act} = \begin{bmatrix} -60 & 0 & 0 & 0 \\ 0 & -60 & 0 & 0 \\ 0 & 0 & -60 & 0 \\ 0 & 0 & 0 & -60 \end{bmatrix}$$

.

Appendix B

Classic Controller

The $'Classic'$ control system used in the present thesis was developed in [7], where the classical methodology to design and to develop controllers were used. In [7], a *Simulink* block was developed to implement the classical control system. The same *Simulink* block was used in this thesis to compare fault simulations results. The main control chains were defined as:

- *Pitch*, *Yaw* and *Roll* angle.
- Velocity, *U*.
- Altitude, *Pos z*.

Using these variables an autopilot or a guidance system can control the aircraft's trajectory.

B.1 Longitudinal Control

Longitudinal control is responsible for the aircraft's velocity and altitude. By controlling the velocity with get a direct relation between Altitude and Pitch angle. Because of that we only need two control loops.

- Elevator, $\eta_e \rightarrow$ altitude, *Pos z*.
- Engine Power, $T_H \rightarrow$ velocity, U.

This velocity control could result in a additional fuel spendding, so this isn't an alternative and also a realistic approach.

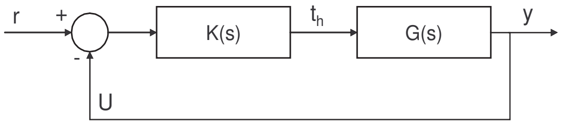


Figure B.1: Velocity Control Loop

Good velocity control is obtained, with a lower response time and no steady state error. The altitude control loop is more complex than the previous one, consists on two cascaded loops, with a inner loop that track *pitch* angle.

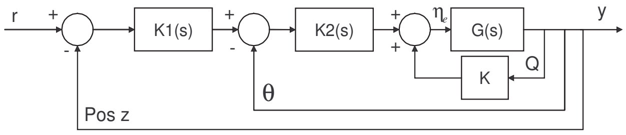


Figure B.2: Altitude Control Loop

B.2 Lateral Control

Lateral aircraft control is important in order to execute turns. This maneuver can be achieved in two different ways, by using the Rudder that changes the *yaw* angle or changing the *roll* angle with the ailerons. In fact we get two control chains.

- Rudder, η_r → *yaw* angle, ψ.

  ou

- Ailerons, η_a → *yaw* angle, ψ.

In order to avoid *sideslip* angle (β) the Rudder has to be actuated in coordinated turns maneuvers.

A controller that performs turns using ailerons was developed:

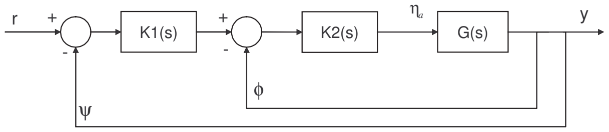


Figure B.3: Turn Control System

To perform a coordinated turn the *sideslip* angle must be $\beta = 0$, so another control loop is used:

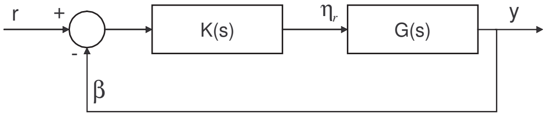


Figure B.4: Turn Control System

B.3 Linear Controllers

The aircraft's dynamics changes with velocity, so it is important to design specific controllers to each fight condition. The project will use polynomial controllers, so system transfer functions have to be known. After identifying each system corresponding to the different control chains the controllers obtained must discretezed. In the next figure B.5 a dicretized model, where the D/A block simbolizes the digital-analog converter and the A/D the opposite case

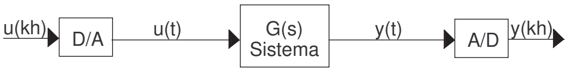


Figure B.5: Discretized System

In digital control the key aspect is the choice of samplimg time. Everytime a system is discretezed, the poles p, are transformed into e^{pT_s} by T_s we understand to be sample time.

The controller's sample time should be chosen taking into account the following aspects:

- Desired Bandwidth.
- Zeros placement.
- Possibility of losing control (Robustness).
- Calculation Time.
- Maximum admissible time for the system to be on open loop .

B.4 Polynomial Control Principles

The following figure shows a polynomial controller with two freedom degrees:

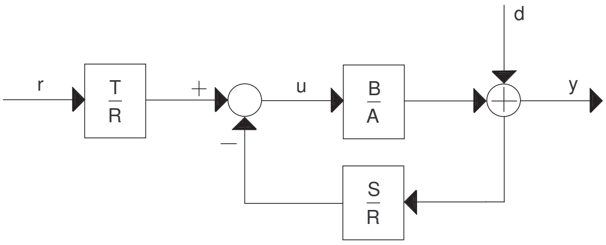


Figure B.6: Two Freedom Degree System

As the process is modulated by the transfer function $H(z) = B(z)/A(z)$, there's the need of determine the (R, S and T polynom), in order that the controller work like $H(z) = B_m(z)/A_m(z)$ satisfying in closed loop $\partial A_m - \partial B_m \geq \partial A - \partial B$.

Controller is described by:

$$R(q)u(k) = T(q)r(k) + S(q)y(k) \tag{B.1}$$

where R advance operator.

B.5 Controller Specifications

The general objectives of this controller are:

- Perturbation Rejection.
- Command signal following.
- System Stabilisation.
- System Robustness.

Appendix C

Guidance System

The guidance system used in the present thesis was developed in [7], as well as the *Simulink* block. This system is basically an outer control loop that receives 3D positions and then gives references signals to an internal controller in order to execute a trajectory following.

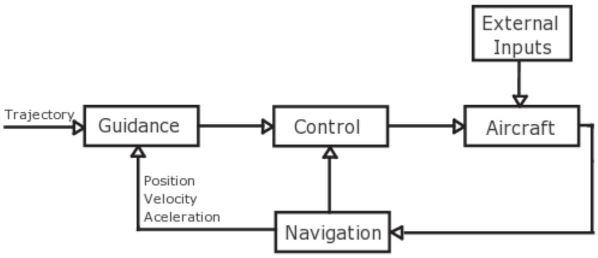


Figure C.1: Guidance System Overview

C.1 Trajectories Build

Several trajectories were used in order to test the develop MPC controllers. Creating these trajectories can represent a good amount of work, so a different approach is used to build the trajectory data. The trajectories used in this thesis were created by applying reference signals to the aircraft's model and his internal controllers.

The following reference trajectories were used:

- Climb Maneuver.

- Vertical and Horizontal Sinusoidal Trajectories.

- Spiral.

In order to avoid obstacles, in the present of atmospheric instability or the aircraft failures like in this thesis, these references have to be adapted and updated in real time.

C.2 Control System

Two different approaches can be used in this kind of systems one (*Tracking*) and the other (*Path Following*). Tracking is considered to be trajectory following with an exact time line, in contrast to Path Following that have the same trajectory reference data but without time constraints. To create this system an approach with a PID controller was chosen. Since the internal controllers were designed to receive as inputs, longitudinal velocity $U(t)$, *yaw* angle $\psi(t)$ and *pitch* angle $\theta(t)$, and with the model's outputs ($x(t),\ y(t)\ and\ z(t)$), any reference trajectory can be followed. From a reference trajectory ($x_d(t),\ y_d(t)\ and\ z_d(t)$), we obtain the following velocities ($v_{xd}(t)$, $v_{yd}(t)$ and $v_{zd}(t)$). In the next figure C.2 the guidance system block are shown:

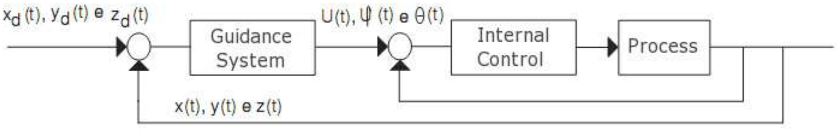


Figure C.2: Guidance System Block Diagram

The reference velocities can always be generated *offline*. So this system consists in the transformation of reference positions and velocities in velocity and attitude angles commands. This transformation is accomplish using some simplifications showed by the following figure C.3.

The following mathematical formulations are used to obtain velocity and attitude angles.

$$\begin{cases} U(t) = \sqrt{v_{xd}^2(t) + v_{yd}^2(t) + v_{zd}^2(t)} \\ \psi(t) = arctan\left(\frac{v_{yd}(t)}{v_{xd}(t)}\right) \\ \theta(t) = arctan\left(\frac{v_{zd}(t)}{v_d(t)}\right), \quad v_d(t) = \sqrt{v_{xd}^2(t) + v_{yd}^2(t)} \end{cases} \tag{C.1}$$

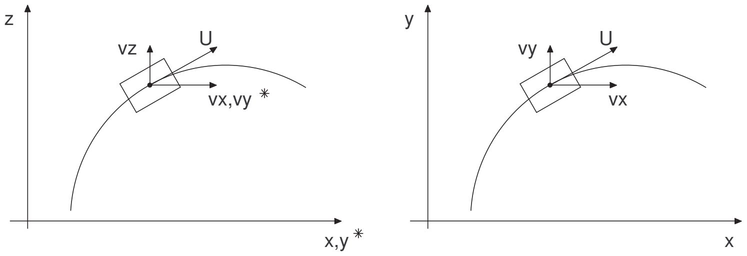


Figure C.3: Transformation of Velocity vector. * represents a projection in XOY

In closed loop the objective is to avoid position error. The final controller is obtained by an empirical way, as the correction of velocity is performed by a PID block, dimensioned according to the problems needs. So velocity and angles are given by:

$$\begin{cases} U(t) = \sqrt{v_x^2(t) + v_y^2(t) + v_z^2(t)} \\ \psi(t) = arctan\left(\frac{v_y(t)}{v_x(t)}\right) \\ \theta(t) = arctan\left(\frac{v_z(t)}{v(t)}\right), \quad v(t) = \sqrt{v_x^2(t) + v_y^2(t)} \end{cases} \tag{C.2}$$

considering:

$$\begin{cases} v_x(t) = v_{xd}(t) + G_1[x_d(t) - x(t)] \\ v_y(t) = v_{yd}(t) + G_2[y_d(t) - y(t)] \\ v_z(t) = v_{zd}(t) + G_3[z_d(t) - z(t)] \end{cases} \tag{C.3}$$

where G_i correspond to PID gains.

In order to have a smooth references for the internal controllers and low response times an unique set of PID gains were chosen. Since the errors in X_E-axis and Y_E-axis considering the*Earth Frame* are almost the same is common sense that the blocks responsible have the same gains, however the altitude controller has a different value. So the used gains are:

$$G1 \quad = G2 \Rightarrow k_p = 0.5,\ k_d = 0.8\ and\ k_i = 0.055 \tag{C.4}$$

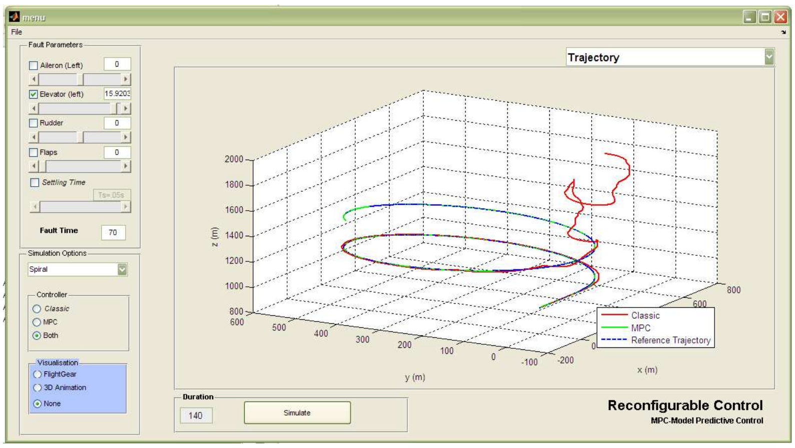
$$G3 \quad \Rightarrow k_p = 0.85,\ k_d = 0.5\ and\ k_i = 0.2 \tag{C.5}$$

These gains were obtained by an empirical way, considering the previous aspects and the main objective that is to decrease the aircraft's distance to the reference trajectory. Finally, the guidance system to work properly has the airframe position in each time, this data can be easily obtained using a *Global Positioning System*(GPS).

Appendix D

Fault Simulation GUI

A graphical user interface (GUI) was develop to assist this research. One of the time consuming aspects was the preparation of each test, so this menu was created using *MatLab GUIDE* software. With this tool the user is capable of easily configure the desired experiment and afterwards visualise all the important data. The following figure shows the general view of this tool. This application can be easily obtained by running *menu.mat* a Matlab script. All the important aspects to perform a simulation test can be easily tuned. The first thing is to choose what control surface will suffer the failure, secondly the user has to specify the desired deflection if he want a *Total Power Loss*. To simulate a *Partial Power Loss* the *settling time box* has to be selected and the pretended time selected also, using the sliding



menu

Fault Parameters

□ Aileron (Left) 0

☑ Elevator (left) 15.9203

□ Rudder 0

□ Flaps 0

□ *Settling Time* Ts=.05s

Fault Time 70

Simulation Options

Spiral

Controller

○ *Classic*

○ MPC

● Both

Visualisation

○ FlightGear

○ 3D Animation

● None

Trajectory

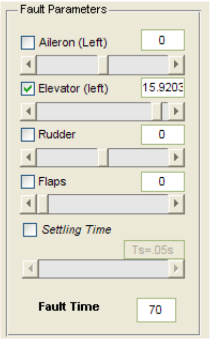
Duration 140

Reconfigurable Control

MPC-Model Predictive Control

Figure D.1: Fault Simulation GUI

bar as shown in figure D.2. There's also the option to chose the time that this failure will happen. Afterwards the user must chose the desired trajectory that he wants the aircraft to follow represented in figure D.3. The the controller must be chosen between the MPC or a *Classic* one. If a comparative test is desired the user has to select the *Both* option (see figure D.4). This application offers the possibility of interacting with other programs, so the user can use the *visualisation* menu to chose from a *Flight Gear* simulation or 3*D Animation* in Matlab environment suitable for comparative test (see figure D.5).



Fault Parameters

☐ Aileron (Left) 0

☑ Elevator (left) 15.9203

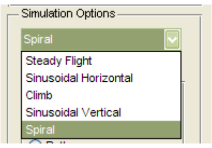
☐ Rudder 0

☐ Flaps 0

☐ *Settling Time* Ts=.05s

Fault Time 70

Figure D.2: Fault Simulation Options

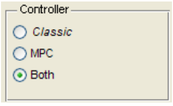


Simulation Options

Spiral

- Steady Flight
- Sinusoidal Horizontal
- Climb
- Sinusoidal Vertical
- Spiral

Figure D.3: Simulation Trajectory Options



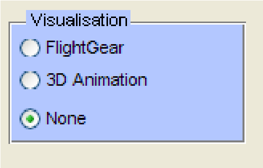
Controller

○ *Classic*

○ MPC

● Both

Figure D.4: Simulation Controllers Options



Visualisation

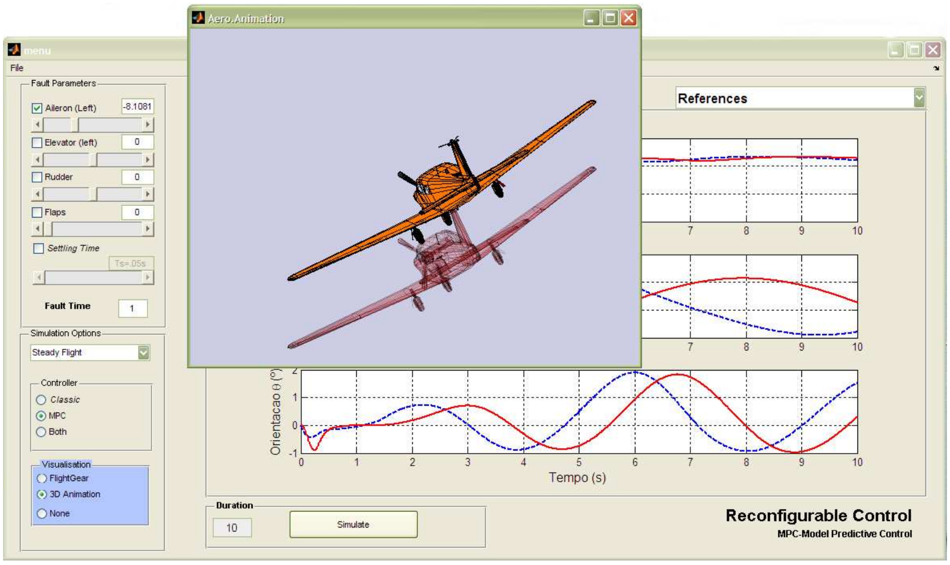
○ FlightGear

○ 3D Animation

● None

Figure D.5: 3D Visualisation Options

3*D Animation* option offers the possibility of displaying two airpcraft at the same time, this capability is suitable for comparative simulations between MPC and *Classic* controllers (see figure D.6).



Fault Parameters

☑ Aileron (Left) -8.1081

□ Elevator (left) 0

□ Rudder 0

□ Flaps 0

□ *Settling Time* Ts=.05s

Fault Time 1

Simulation Options: Steady Flight

Controller: ○ *Classic* ● MPC ○ Both

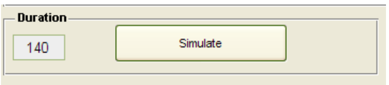
Visualisation: ○ FlightGear ● 3D Animation ○ None

References

Duration 10

Figure D.6: 3D Comparative Animation

Now the time length for his simulation can be selected using the *Duration Box* as shown in figure D.7, finally by pressing the *Simulate* bottom and test will run.



Duration 140

Figure D.7: Simualtion Time Options

Finally the user can chose have the data he wants to visualise, by selecting the pop-up menu the desired graphics (see figure D.8).

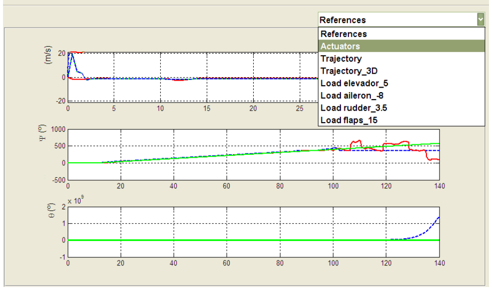


Figure D.8: Simulation Graphics Options