Modelling and Controlling a Suborbital Reusable Launch Vehicle

Master's Thesis in Systems, Control and Mechatronics

PATRIK RODSTEDT

Performed at FOI Swedish Defense Research Agency

CHALMERS UNIVERSITY OF TECHNOLOGY
Department of Signals & Systems
Examiner Chalmers: Torsten Wik
Supervisor FOI: Fredrik Berefelt & John Robinson
Gothenburg, Sweden 2011
Master's Thesis 2011: EX077/2011

Abstract

This thesis is a modelling and control study of a RLV (Reusable Launch Vehicle), which will bring three individuals to space with a rocket engine and then glide back to Earth.

Both a classic linear control design and a family of nonlinear block backstepping controllers were tried.

The linear controller had a proportional and an integral part and was gain scheduled over the envelope for different altitudes and Mach numbers. Tuning the gains in gain scheduling is a time consuming process, which was solved by mapping the flight handling qualities to the space of the controller gains and thereby choosing the gains.

The Block backstepping control laws are designed in one piece for all the controlled variables. This is done over the whole envelope by replacing the open-loop dynamics of the vehicle by a *desired dynamics*. Both the linear and the nonlinear control laws had sufficient control performance.

The worst-case pilot input of the block backstepping control laws were investigated using a global optimiser in a clearance test. It was shown that the backstepping control laws were sensitive to rate saturation of the control surfaces.

Preface

This report was conducted at FOI (Swedish Defense Research Agency) 2011 in Stockholm, Sweden. The methods and design procedures in chapters 2,3,4 were made in collaboration with Johan Knöös from KTH (Royal Institute of Technology), supervised by Fredrik Berefelt and John Robinson at FOI.

Patrik Rodstedt, Stockholm 11/9/11

Contents

Nomenclature

α	Angle of attack
β	Sideslip angle
δ_a	Aileron deflection
δ_b	Bodyflap deflection
δ_e	Elevator deflection
δ_r	Rudder deflection
ϕ	Euler bank angle
ψ	Euler azimuth angle
θ	Euler elevation angle
C_ϕ	Short for $\cos(\phi)$
C_ψ	Short for $\cos(\psi)$
C_θ	Short for $\cos(\theta)$
F_{xb}	Aerodynamic force component in the x_b direction
F_{yb}	Aerodynamic force component in the y_b direction
F_{zb}	Aerodynamic force component in the z_b direction
g	Acceleration of gravity
I_{xxb}	Moment of inertia
I_{yyb}	Moment of inertia
I_{zxb}	Product of inertia

m	Mass of spacecraft
M_{xb}	Aerodynamic moment around x_b axes
M_{yb}	Aerodynamic moment around y_b axes
M_{zb}	Aerodynamic moment around z_b axes
p	Roll rate
q	Pitch rate
r	Yaw rate
S_ϕ	Short for $\sin(\phi)$
S_ψ	Short for $\sin(\psi)$
S_θ	Short for $\sin(\theta)$
u	The x-component of the airspeed expressed in body coordintates
V	Body-axes velocity
v	The y-component of the airspeed expressed in body coordintates
w	The z-component of the airspeed expressed in body coordintates
x_f	Northward Earth-fixed coordinate
y_f	Eastward Earth-fixed coordinate
z_f	downward Earth-fixed coordinate

1

Introduction

1.1 Background

The past years, a number of space tourist projects have shown up. One company that aims to commercialise space traveling is Virgin Galactic. who managed to perform the first commercial manned flight on an altitude of over 13.7 km [1]. Since 2005 it has been possible to book a ticket with Virgin Galactic and have a seat in a future space experience. The ticket prices have a neat sum of 200 000 USD. Virgin Galactic is only operating in America at the moment. This is due to that US Federal Aviation Authority (FAA) has temporary relaxed the regulation requirements for commercial suborbital flights until 2012 and thus made the market possible. In Europe the FAST20XX (Future high-Altitude high-Speed Transport) project is a first step to its own commercial suborbital flight [2].

This thesis is a part of the European space project FAST20XX, on behalf of ESA (European Space Agency). A number of companies and institutions around Europe have signed up as project partners and each partner will deliver small parts to the project. One partner in the project is FOI (Swedish Defense Research Agency), which has the task of designing a manual control system.

The main goal with the project is to, within five to ten years, provide the technology needed for two different types of spacecraft. This means that the project does not aim to deliver a spacecraft at the end of the time period, but to deliver the knowledge to the day when Europe decides to start the development of such a spacecraft.

There are two different types of spacecraft in the project, both satisfies the idea of bringing humans into space. This work focus only on one of them, called Alpha (Airplane Launched Phoenix Aircraft), which is a small carrier with three passengers and a pilot, see Figure 1.1. The Alpha aircraft is planned to be released in the air from a large military airplane of similar size to the Antonov plane and from there use a hybrid rocket motor to proceed out from the atmosphere to give the passengers a view of earth from space before it returns and land on earth in a curve shaped path.

FAST20XX is divided into six technology areas, which are: hybrid propulsion, innovative high performance cooling technique, separation techniques, flow control, guidance and navigation control (GNC), and safety analysis. The GNC part aims to use modern control techniques to give a robust control design. It will have both an auto-pilot and a manual control system. The manual control system will only act as a back-up system when the auto-pilot cannot be used. There are a number of situations when a pilot would have to rely on the manual system, for example in the case when the landing site is moved to a new location or if an accident has occurred. Hence, the manual system is a small but equally important part of the overall design. To be able to commercialise space transportation the safety aspects are of great value. The psychological effect of having a stand-by system if the auto-pilot malfunctions is therefore very important, even if it will never be used in practice.

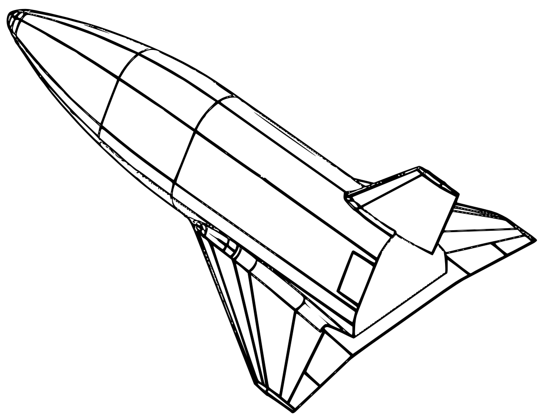

Figure 1.1: This is a schematic image of the Alpha.

1.2 Objective

The following parts were the objective of this work:

- Model and simulate the suborbital carrier Alpha.
- Construct linear and non-linear control laws and specifically try a design built on backstepping.
- Perform a clearance investigation of the control laws.

1.3 Scope

The following assumptions and constraints were applied to this project:

- The sensors are considered to be ideal.

- Control laws during landing procedure are not considered.

1.4 Method

The project was divided into the following sub projects:

- Setup of simulation platform for Alpha in Matlab.
- An Open-loop trim investigation.
- A linear baseline control design.
- A nonlinear control design.
- Clearance test with global optimisation.

To be able to start investigating the Alpha a simulation environment was built in Matlab. The most important step was to interpret and import the aerodata[1] into the simulation platform.

In the second part a study of different trim states and the Alphas open-loop characteristics was performed. Longitudinal trim, vector velocity roll and a coordinated turn were analysed.

The main goal of the project was to try different nonlinear control theories, but the design first started with a baseline linear controller. The baseline controller was tuned with the help of a mapping theorem by Ackermann, which speeded up the gain scheduling process and also gave an overview of the handling qualities.

As nonlinear controllers, two related design methods, NDI (Nonlinear Dynamic Inversion) and backstepping were investigated. Both controllers were designed and proved stable by the use of Lyapunov theory.

In the end a clearance test was set up that searched through different stick combinations to find a worst case scenario and thereby testing the performance of the controllers. The clearance test was made with a optimisation package for Matlab developed by Fredrik Berefelt at FOI [3].

[1]The aerodata holds the information of how the forces and moments act on the spacecraft during flight. It is gathered through wind tunnel experiments.

2

The spacecraft

This chapter aims to be a short summary of the theory used for modelling the Alpha. Since the Alpha behaves as an aircraft at the lower part of its envelope, the theory in this chapter is identical to what is used in the aircraft community.

The chapter starts with a brief explanation of the rigid-body equations, followed by a model of the control surfaces, the oscillatory modes and static stability[1]. It ends with an overview of the flight envelope.

2.1 Rigid-body equations

The first step in describing the dynamics of the Alpha is to derive its force and moment equations in a body-fixed coordinate system. Assume that the Alpha vehicle is a rigid body and Newtons second law gives

$$\boldsymbol{F} + \boldsymbol{W} = \frac{d}{dt}(m\boldsymbol{V}) + \boldsymbol{\omega} \times (m\boldsymbol{V}) \tag{2.1}$$

$$\boldsymbol{M} = \frac{d}{dt}(\boldsymbol{I}\boldsymbol{\omega}) + \boldsymbol{\omega} \times (\boldsymbol{I}\boldsymbol{\omega}), \tag{2.2}$$

where $\boldsymbol{W}$ is the gravitational force vector of the Alpha vehicle, $\boldsymbol{M} = [M_{xb}\ M_{yb}\ M_{zb}]^{\mathrm{T}}$ is the moment about the body-fixed center of gravity, $\boldsymbol{V} = [u\ v\ w]^{\mathrm{T}}$ and $\boldsymbol{\omega} = [p\ q\ r]^{\mathrm{T}}$ are the translational and rotational velocities of the Alpha vehicle, respectively in body-fixed coordinates (see Figure 2.1). Since the vehicle in this study does not have an engine, the force $\boldsymbol{F} = [F_{xb}\ F_{yb}\ F_{zb}]^{\mathrm{T}}$ only involves the aerodynamic forces acting on the body.

[1]Static stability is a situation when the Alpha is flying with no acceleration.

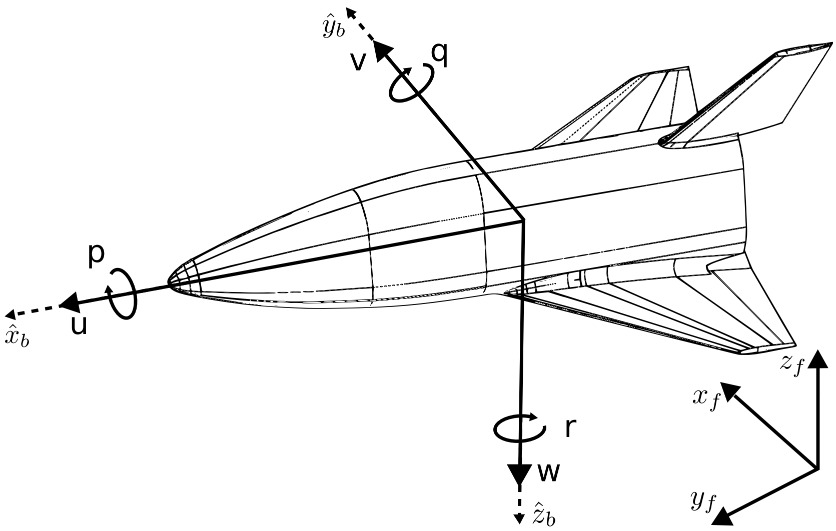

Figure 2.1: Body-fixed velocity vector and angular rates.

In the above equations, m is the Alphas mass and $\boldsymbol{I}$ is its inertia tensor

$$\boldsymbol{I} = \begin{bmatrix} I_{xxb} & 0 & -I_{xzb} \\ 0 & I_{yyb} & 0 \\ -I_{zxb} & 0 & I_{zzb} \end{bmatrix}. \tag{2.3}$$

Expressing the moment and force equations in more detail involves writing the Alphas body-fixed coordinate system to the Earth fixed. This is done by introducing the Cartesian Earth-fixed coordinate system $[x_f \ y_f \ z_f]^{\mathrm{T}}$ and how the body is orientated relative to the Earth by the three Euler angles $[\phi \ \theta \ \psi]^{\mathrm{T}}$ [4, p. 735]. With the help of the Euler angles the gravitational force can be written in the body-fixed coordinates as

$$\begin{bmatrix} W_{xb} \\ W_{yb} \\ W_{zb} \end{bmatrix} = g \begin{bmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{bmatrix}. \tag{2.4}$$

By using the new expression for the gravitational force in body-fixed coordinates (2.4) and carrying out the cross products in (2.1) and (2.2), the following two complete expressions

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} F_{xb} \\ F_{yb} \\ F_{zb} \end{bmatrix} + g \begin{bmatrix} -S_\theta \\ S_\phi C_\theta \\ C_\phi C_\theta \end{bmatrix} + \begin{bmatrix} rv - qw \\ pw - ru \\ qu - pv \end{bmatrix} \tag{2.5}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \begin{bmatrix} I_{xxb} & 0 & -I_{xzb} \\ 0 & I_{yyb} & 0 \\ -I_{zxb} & 0 & I_{zzb} \end{bmatrix}^{-1} \begin{bmatrix} M_{xb} + (I_{yyb} - I_{zzb})qr + I_{xzb}pq \\ M_{yb} + (I_{zzb} - I_{xxb})pr + I_{xzb}(r^2 - p^2) \\ M_{zb} + (I_{xxb} - I_{yyb})pq - I_{xzb}qr \end{bmatrix} \tag{2.6}$$

are achieved, which completely describe the Alpha vehicle in its six degrees of motion, where S_θ and C_θ are short for $\sin\theta$ and $\cos\theta$. The last step of explaining the Alphas motion is to know how its position and orientation in the Earth-fixed coordinates change over time. This is done by using transformation matrices between Earth- and body-fixed coordinates on the translation and rotation velocity vectors (u,v,w) and (p,q,r) of the body. The transformation matrices are built up by cosine and sine elements of the Euler angles (ϕ,θ,ψ). The following two expressions

$$\begin{bmatrix} \dot{x}_f \\ \dot{y}_f \\ \dot{z}_f \end{bmatrix} = \begin{bmatrix} C_\theta C_\psi & S_\phi S_\theta C_\psi - C_\phi S_\psi & C_\phi S_\theta C_\psi + S_\phi S_\psi \\ C_\theta S_\psi & S_\phi S_\theta S_\psi + C_\phi C_\psi & C_\phi S_\theta S_\psi - S_\phi C_\psi \\ -S_\theta & S_\phi C_\theta & C_\phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} V_{wx_f} \\ V_{wzy_f} \\ V_{wz_f} \end{bmatrix} \tag{2.7}$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & S_\phi S_\theta / C_\theta & C_\phi S_\theta / C_\theta \\ 0 & C_\phi & -S_\phi \\ 0 & S_\phi / C_\theta & C_\phi / C_\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix}, \tag{2.8}$$

gives the differential equations of the Earth-fixed coordinates and the Euler angles, where $[V_{wx_f}\ V_{wy_f}\ V_{wz_f}]$ is wind velocity vector. The twelve differential equations (2.5), (2.6), (2.7) and (2.8) can easily be integrated numerically and yields a complete description of the Alphas state in the Earth- and body-fixed coordinates. [4]

In some applications, as aerodynamic modelling, it is more convenient to describe the body velocity with the triple (V,α,β) instead of the usual velocity vector (u,v,w) (see Figures 2.2 and 2.3). The transformation between (u,v,w) and (V,α,β) is given by

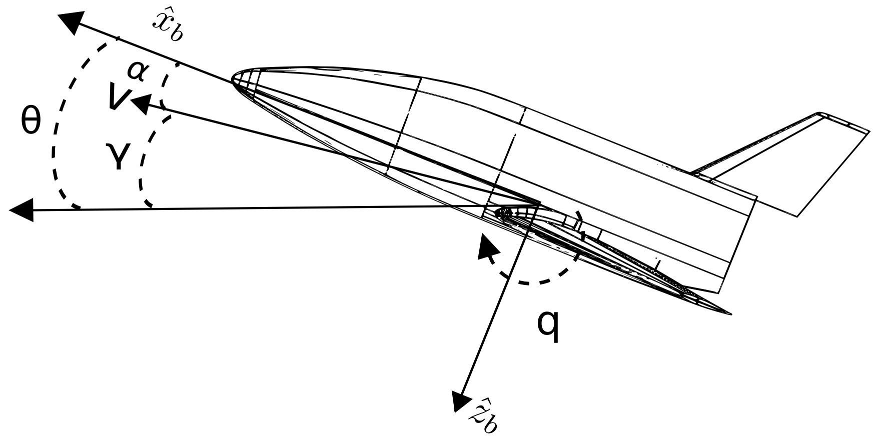

Figure 2.2: The Alpha from the side, where γ is the flight path angle, α is the angle of attack, θ is the Euler elevation angle, V is the body-axis velocity and q is the pitch rate.

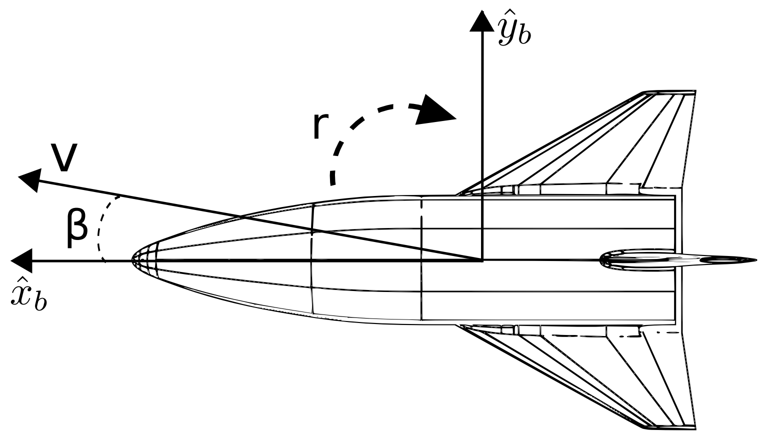

Figure 2.3: The Alpha from the top, where β is the sideslip, V the body-axis velocity and r the yaw rate.

$$
\begin{aligned}
V &= \sqrt{u^2 + v^2 + w^2} \\
\alpha &= \arctan \frac{w}{u} \\
\beta &= \arcsin \frac{v}{V}.
\end{aligned}
\tag{2.9}
$$

It is also very common to use $(\dot{V}, \dot{\alpha}, \dot{\beta})$ in simulations instead of $(\dot{u},\dot{v}, \dot{w})$. The differential equations for $(\dot{V}, \dot{\alpha}, \dot{\beta})$ are derived by differentiating (2.9) to get

$$
\begin{aligned}
\dot{V} &= \frac{1}{m} \left(F_{Tx} \cos\alpha \cos\beta + F_{Tz} \sin\alpha \cos\beta\right) \\
\dot{\alpha} &= \frac{1}{Vm\cos\beta} \left(-F_{Tx} \sin\alpha + F_{Tz} \cos\alpha\right) + q - \left(p\cos\alpha + r\sin\alpha\right)\tan\beta \\
\dot{\beta} &= \frac{1}{Vm} \left(-F_{Tx} \cos\alpha \sin\beta + F_{Ty} \cos\beta - F_{Tz} \sin\alpha \sin\beta\right) + p\sin\alpha - r\cos\alpha,
\end{aligned}
\tag{2.10}
$$

where the force $F_{Ti} = F_{ib} + W_{ib}$ is the total force in the corresponding body-axis.

2.2 Closer look at aerodynamics

The aerodynamic forces and moments on the body depend on a number of variables, such as the body state and the atmosphere. In general, there are no complete formulas to calculate the forces and moments from the Alphas state. Instead they are calculated by first determining the aerodynamic coefficient (C_F,C_M) from a lookup table and then use the formulas

$$F_{ib} = \bar{q}AC_{F_i}(\delta,\alpha,\beta,p,q,r,\dot{\alpha},\dot{\beta}\cdots) \tag{2.11}$$

and

$$M_{jb} = \bar{q}AlC_{M_j}(\delta,\alpha,\beta,p,q,r,\dot{\alpha},\dot{\beta}\cdots). \tag{2.12}$$

Here,

- the index i corresponds to the different body-axis (x_b,y_b,z_b).
- the index j corresponds to the three different moments around the body-axis (x_b,y_b,z_b).
- A denotes the wing-area of the Alpha.
- l denotes the wing-length.
- $\bar{q}$ is called the aerodynamic pressure and equals to $\frac{1}{2}\rho(h)V^2$ and it captures the aerodynamic stress the aircraft experience at the altitude h, the air density ρ and with the velocity V.
- δ denotes the control surfaces. [5]

The lookup table is made from wind tunnel experiments in a lab and are limited to sideslip angles ± 90 deg and angle of attacks between -2 deg and 24 deg. Figures 2.4 and 2.5 shows an example of how the side and normal force coefficients C_{F_x} and C_{F_z} change over the angle of attack α and sideslip angle β for a fixed Mach number of 0.3.

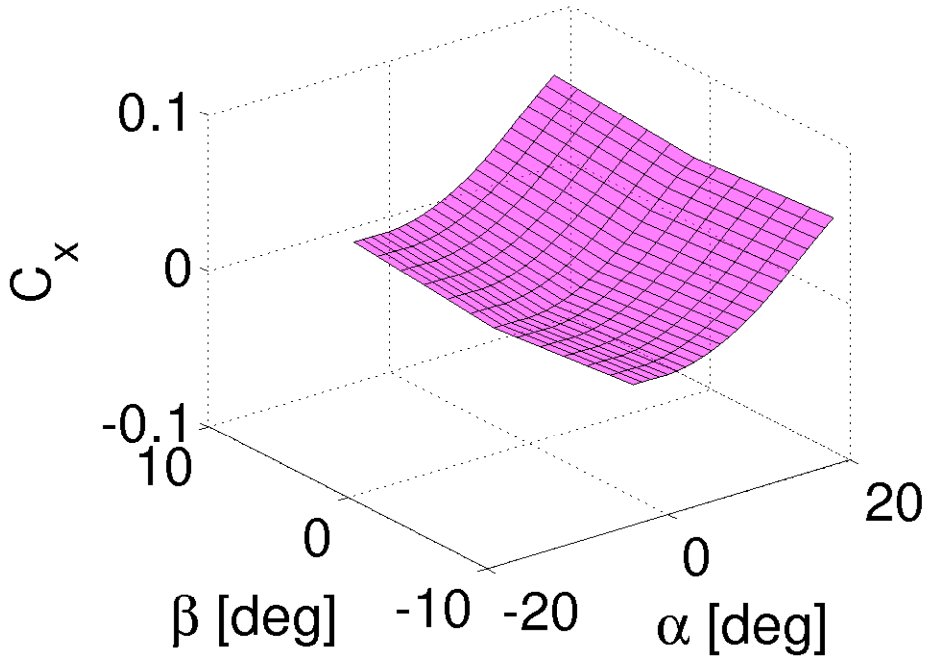

Figure 2.4: The side force coefficient C_{F_x} at Mach number 0.3.

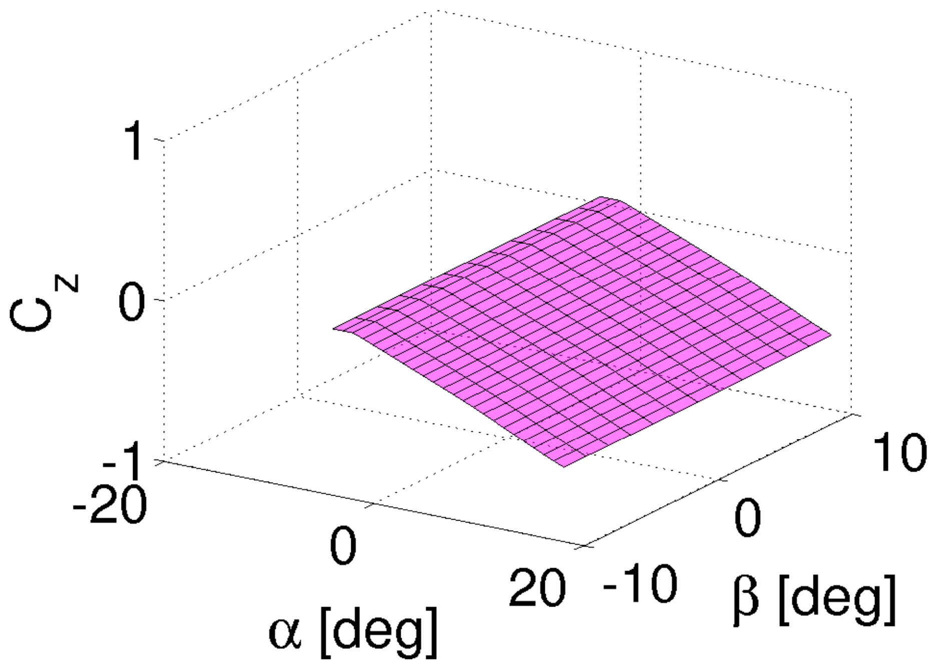

Figure 2.5: The normal force coefficient C_{F_z} at Mach number 0.3.

2.3 Control surfaces

The Alpha has five different control surfaces, all shown in Figure 2.6. The outboard and inboard elevons can be used either as elevators or aileron, by either deflecting the elevon couple symmetrically or differentially,

$$\delta_{symm} = \frac{\delta^{left} + \delta^{right}}{2} \tag{2.13}$$

$$\delta_{diff} = \frac{\delta^{left} - \delta^{right}}{2}. \tag{2.14}$$

A symmetrical deflection corresponds to a pull-up or a pull-down, but a differential deflection makes a roll. In this work the outboard elevons have been used as ailerons and the inboard elevons as elevators. Each control surface has its own deflection limits, which are given in Table 2.1 with a short summary for what each control surface is used.

2.4 Actuator dynamics

The dynamics of the control surfaces are modelled by first order systems

$$\dot{x} = (u - x)\frac{1}{\tau}, \tag{2.15}$$

Table 2.1: A summary of the control surfaces usage and limits.

Control surface	Usage	Deflection limit [deg]
Body Flap	Keep longitudinal trim	-15 to +20
Elevator Inboard	Give pitch control	-20 to +20
Rudder	Give yaw control	-10 to +10
Elevator Outboard	Used as aileron to give roll control	-20 to +20
Speed Break	Velocity control	0 to 40

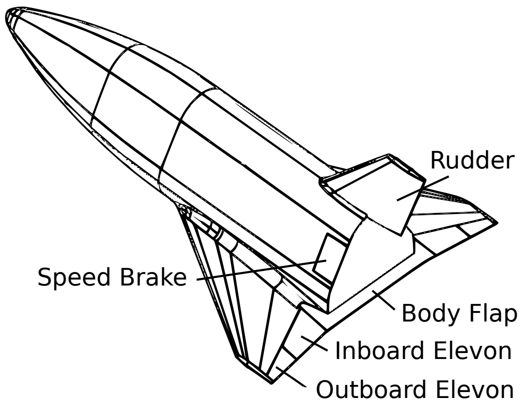

Figure 2.6: The different control surfaces of the Alpha.

where u denotes the commanded actuator deflection. The time constant τ is set to 0.03 s and the angular velocity of the actuators are also limited to saturate at 50 deg/s. These values are taken from a similar project by NASA [6].

2.5 Dynamics

The dynamics of an aircraft can be described by six differential equations (three translational and three rotational degrees of freedom), see (2.5) and (2.6). If the aircraft moves with small deviations from an equilibrium point the flight equations can be approximated by a linearisation. If the flight manoeuvre is slow enough the six differential equations can be separated into two subsystems, one with longitudinal and one with lateral dynamics. Both the longitudinal and the lateral subsystem has a number of oscillatory modes. Each mode has a given name and their damping and eigenfrequency are commonly used as a measure of the handling qualities (see section 2.6). Because the modes characteristics change during the flight envelope, linear flight controllers may need to be gain scheduled.

2.5.1 Longitudinal dynamics

The longitudinal dynamics are made out of a subsystem with the states

$$x = [\alpha \; q \; V \; \theta]^T, \tag{2.16}$$

and the input vector

$$u = \delta_e, \tag{2.17}$$

where δ_e denotes the elevator deflection. This subsystem has two complex eigenvalue pairs, which are called phugoid and short-period mode in the flight community [5]. The phugoid mode is sometimes called long-period mode and has more damping and an eigenfrequency corresponding to a period usually around 30 s. The short-period mode is much lower damped and has a period of just a few seconds.

After linearisation and calculation of the eigenvalues of the subsystem it is always easy to separate the modes apart. The short-period mode will always have the eigenvalue pair with the largest magnitude. It depends on a combination of the pitch rate q and the angle of attack α, whereas the phugoid mode depends mostly on the airspeed V and the Euler pitch angle θ. The dynamics of the short-period mode can be calculated from the approximated subsystem

$$\begin{bmatrix} \dot{\alpha} \\ \dot{q} \end{bmatrix} = \begin{bmatrix} \frac{\partial f_\alpha}{\partial \alpha} & \frac{\partial f_\alpha}{\partial q} \\ \frac{\partial f_q}{\partial \alpha} & \frac{\partial f_q}{\partial q} \end{bmatrix} \begin{bmatrix} \alpha \\ q \end{bmatrix} + \begin{bmatrix} \frac{\partial f_\alpha}{\partial \delta_e} \\ \frac{\partial f_q}{\partial \delta_e} \end{bmatrix} \delta_e, \tag{2.18}$$

where $\frac{\partial f_\alpha}{\partial \alpha}$ is the linearisation around an equilibrium point $\dot{\alpha} = 0$ in (2.10). This subsystem approximation is later used when calculating the handling qualities of the short period mode.

Most of the longitudinal handling qualities are specified in the short-period mode since the pitch responsiveness depends on this mode. The high frequency combined with a too low damping of the mode can give the pilot problems to manoeuvre the aircraft.

2.5.2 Lateral Dynamics

In the same way as the longitudinal dynamics, the lateral subsystem has four states

$$x = [\beta \; \phi \; p \; r]^T, \tag{2.19}$$

but two inputs

$$u = [\delta_a \; \delta_r]^T, \tag{2.20}$$

where δ_a denotes aileron deflection and δ_r denotes rudder deflection. Calculating the eigenvalues of the subsystem shows that the system has one complex pole pair and two real poles. The two real poles are called roll mode and spiral mode and the roll mode is heavily damped while the spiral mode is less damped and can be unstable. The complex pole pair is called the dutch roll mode and it is very low damped with a short period of a few seconds. It can be excited by a rudder impulse and can give problems during landing with gutsy wind.

The short-period, roll, and dutch roll mode are together called the three rotational modes that gives the manoeuvrability of the aircraft. [5]

2.6 Handling qualities

The performance of a flight controller is very hard to evaluate in an objective and quantified way and the main reason is that each pilot has its own opinion of what a good aircraft response is. The opinion is also different depending on the type of aircraft and flight phase (landing, takeoff or steady level flight) [5]. One quantified criteria of aircraft performance is the US Military Specifications for the Flying Quality of Piloted Airplanes [7], which contains one way to analytically express the handling qualities. As the Alpha behaves as an aircraft in its lower part of the flight envelope (below 12 km) the same methods of evaluating classic commercial and military aircraft can be used for the Alpha.

The specifications contains different criteria for each flight class, flight phase and level. The Alpha is considered as a medium weight and manoeuvrable airplane and should be able to do gradual manoeuvres in nonterminal flight, which puts it in airplane class I, flight phase category B, and flying quality level 1 (see Tables 2.2 and 2.3). CAP

Table 2.2: Handling qualities for longitudinal flight phase B and class II.

HQs	**Level 1**	**Good Level 1**
Short period mode ζ_{SP}	$0.35<\zeta_{SP}<1.35$	$0.7<\zeta_{SP}<1.35$
Control anticipation parameter CAP	$0.085<CAP<3.6$	

(Control Anticipation Parameter) [5] is a criteria that indirectly specifies the undamped natural frequency of the short period mode $\omega_{n_{SP}}$ and is defined as,

$$CAP = \frac{\omega_{n_{SP}}}{n/\alpha}, \tag{2.21}$$

where n/α is the aircraft load-factor response in g's per radian.

Table 2.3: Handling qualities for lateral flight phase B and class II.

HQs	**Level 1**	**Level 2**
Dutch roll damping ζ_{DR}	$0.08<\zeta_{DR}$	$0.02<\zeta_{DR}$
Dutch roll frequency $\omega_{n_{DR}}$	$0.4<\omega_{n_{DR}}$	$0.4<\omega_{n_{DR}}$
Dutch roll $\zeta_{DR}\omega_{n_{DR}}$	$0.15<\zeta_{DR}\omega_{n_{DR}}$	$0.05<\zeta_{DR}\omega_{n_{DR}}$
Roll-mode time constant	<1.4 s	<3 s

2.7 The spacecraft and the mission

The manual control system is designed to be used for the lower part of the Alpha's trajectory. The whole trajectory can be seen in Figure 2.7, but only the lower dashed part starting at 12.5 km will be considered during the design. In that part of the envelope the heat dissipation is of less importance and the Alpha will behave more as an aircraft. In Figure 2.7 to the left the Alpha speed is shown and it ranges between 0.23 Mach and 0.46 Mach. This gives an indication of what the altitude and Mach number limits of the

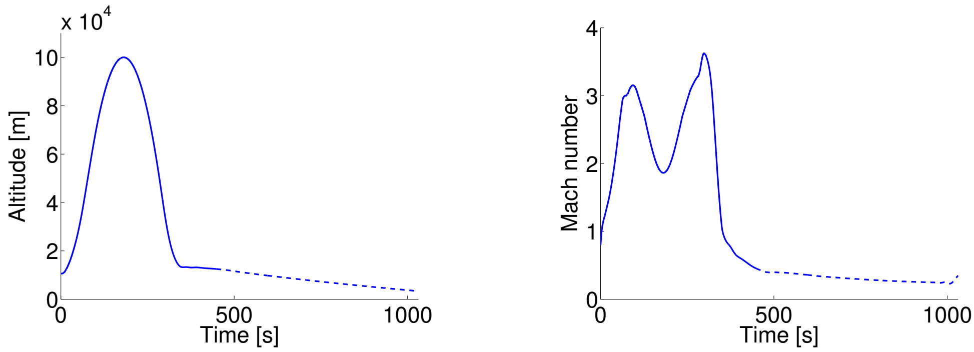

Figure 2.7: The Alpha's trajectory.

mesh used during gain scheduling design should be. To make the Alpha more flexible a Mach number of 0.8 was chosen as the upper limit and 0.2 as the lower limit. It would be preferred choose a lower limit below 0.2, but then the aerodata does not support any smaller Mach number than 0.2. For the altitude the controller is limited to an altitude between 12.5 km and ground. Finally, the gain schedule mesh is built up by 5 different altitudes and 5 different Mach numbers, which means that there are 25 different gain couples from which the linear controller is interpolated.

2.8 Static stability

A trimmed flight is when the Alphas actuators are set so that it is in a static equilibrium and there is no acceleration. In other words when,

$$\dot{p} = \dot{q} = \dot{r} = 0 \text{ and } \dot{u} = \dot{v} = \dot{w} = 0. \tag{2.22}$$

This is called steady-state flight [5]. Of all steady-states there are five standard examples that have their own names. Each one of them have some additional constraints listed in Table 2.4.

The Alphas airframe is designed to be naturally stable and its stays in a trim position by itself during the whole flight. In case of a disturbance it returns back to the trim position on its own.

The control surfaces of an aircraft consists of the ailerons, bodyflaps, elevators and the rudder and they help the aircraft to transit between different trim positions. The

Table 2.4: A summary of the different trim situations.

Trim	Constraints
Steady wings-level flight	$\phi = \dot{\phi} = \dot{\theta} = \dot{\psi} = 0$
Steady turning flight	$\dot{\phi} = \dot{\theta} = 0,\ \dot{\psi} = \text{turn rate}$
Steady pull-up	$\phi = \dot{\phi} = \dot{\psi} = 0,\ \dot{\theta} = \text{pullup rate}$
Steady roll	$\dot{\theta} = \dot{\psi} = 0,\ \dot{\phi} = \text{roll rate}$

aircraft can be designed to be easy or difficult to keep in a trim position and this is given by the aircraft handling qualities.

In Figure 2.8, the Alpha has been steady-state flight trimmed over the flight envelope (a Mach number between 0.2 and 0.8 and the altitude 12.5 km to ground) with a constant flight path angle. To keep the trim during the upper part of the envelope, the bodyflap saturates and the elevator needs to be active too (the grey colored area). At lower altitude the speed and the dynamic pressure is higher and the longitudinal trim is held only with the bodyflap (the white colored area).

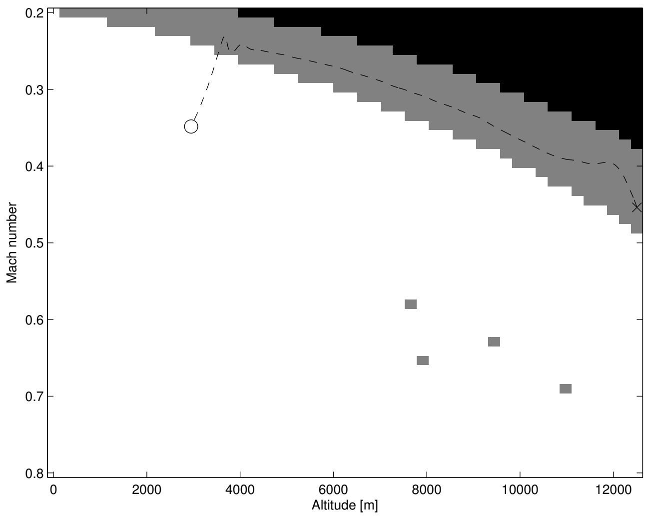

Figure 2.8: This figure shows how the Alpha is trimmable for different Mach numbers and altitudes. The Alphas flight envelope after an altitude of 12.5 km is marked with a dashed line starting at the cross and ending at the circle. The color map tells if the Alpha is trimmable only with the help of the bodyflap(white), with the need of both the bodyflap and its elevator(grey), and if it is not trimmable at all with both the bodyflap and the elevator in use (black).

3

Linear controller

This chapter will present a linear controller built on classic design techniques. The purpose of the linear controller is to compare its design and performance with the later nonlinear controllers.

In flight control there are two requirements of a control system. First, it should keep the vehicle stable during the whole flight and secondly, it should satisfy the flight handling qualities, see section 2.6.

These requirements have to be fulfilled globally and, since the vehicle dynamics depends on the speed and altitude, one single linear controller is not enough for the entire envelope. To be able to use linear control theory a number of controllers have to be designed for various operating points and the control algorithm has to switch between the gains as the vehicle changes speed and altitude, which is known as gain scheduling. In this chapter both a global longitudinal and a lateral controller will be designed.

3.1 SAS and CAS

In the aircraft control community there are three different expressions one have to keep separated, namely the Stability Augmentation System (SAS), the Control Augmentation System (CAS), and the autopilot [5].

The SAS is used to measure and feedback angular rates of different body-axes to produce a desired damping effect, mainly on the three different rotational modes: short-period, roll, and dutch roll mode. In High-performing modern air and spacecraft the damping can be very low or the modes can even be unstable, to increase the maneuverability. A stability augmentation system is then crucial.

The CAS design is used when a specific response to an input is wanted and not just the damping of different modes as in the SAS design. An example of this is when a pilot input affects the normal acceleration along the z-axis. It is also used when a very precise tracking is required, which is very common in modern high performance air and spacecraft.

The autopilot is designed to help the pilot to hold the aircraft steady and to give the pilot a ”time off” during flights.

The control system of the Alpha requires help with both damping and precise tracking, and this motivates the need of a CAS design and not just a SAS. But some parts of the implemented control systems has autopilot characteristics as the choice of controlling the Euler bank angle ϕ.

3.2 Longitudinal controller

There are many different options when choosing control variables and they depend on the overall goal of the CAS. It is common to use pitch-rate or normal acceleration as control variable. A popular control variable is C*, which combines both the pitch-rate and the normal acceleration and it is the control variable used here.

C* control

The C* criteria is a time history criteria and it is built up by a mix of normal acceleration and the pitch-rate. The motivation of constructing a mixed criteria is that depending on the velocity the pilot either use the normal acceleration or the pitch-rate when controlling an aircraft. For low velocity the pitch-rate dominates the feeling of the vehicle, but at high velocity it is the response from normal acceleration that the pilot feels the most. From pilot test it has been concluded that the linear combination

$$C^* = n_z + Kq, \tag{3.1}$$

gives the best results and with $K = 12.4$ [8].

The longitudinal control augmentation system used by the Alpha is implemented with C* as the controlled variable and the controller used is a proportional controller with integral action (see Figure 3.1). There are many options in choosing the type of linear controller. This controller was chosen because of its simplicity and short design time and not for its superior performance. The controller is tuned by the proportional and integral gains K_p and K_I and through the eileron (EI) a pitch moment is produced, which then controls the Alpha.

The control design helps the pilot with tracking as well as fulfilling the handling qualities of the short period mode. An important observation is that the proportional gain is fed back with an inner-loop to lower a large overshoot but still keeping the location of the poles.

3.3 Lateral controller

The lateral controller is slightly more complex than the longitudinal. The main reason is that even when the system is seen as decoupled from the longitudinal dynamics it is a MIMO (Multi Input Multi Output) system. The two inputs aileron (AO) and rudder

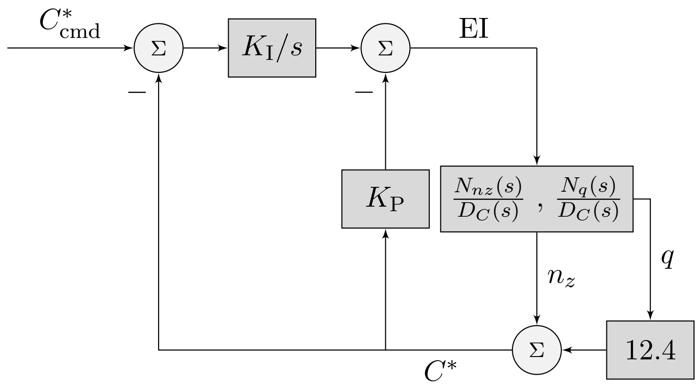

Figure 3.1: Block diagram of the longitudinal CAS with the C* criteria. The transfer functions $\frac{N_{nz}(s)}{D_C(s)}$ and $\frac{N_q(s)}{D_C(s)}$ denotes the open-loop system of the Alpha.

(RU) gives a roll and a yaw moment to control the Euler bank angle ϕ and keep the sideslip angle β zero. In this control law will the sidelip angle not be controlled, instead the rudder will take its position from a look-up table of trim values.

The controller can be seen in Figure 3.2 and it has three gains that need to be tuned. The purpose of the gain K_{pp} in the inner SAS loop is to speed up the roll mode and thus increases the possible roll rate and to push the dutch roll poles until the lateral handling qualities are met. The outer CAS loop with the proportional and integral gains $K_{p\phi}$ and $K_{i\phi}$ has the function of bringing the roll angle ϕ to the commanded ϕ_{cmd} value.

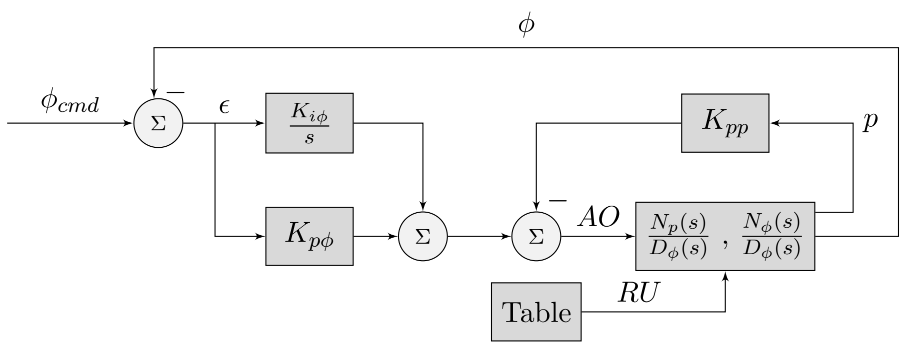

Figure 3.2: Block diagram of the lateral controller. The transfer functions $\frac{N_p(s)}{D_\phi(s)}$ and $\frac{N_\phi(s)}{D_\phi(s)}$ denotes the open-loop system of the Alpha.

3.4 Parameter tuning

In Sections 3.2 and 3.3 the structure of the longitudinal and lateral control systems were shown and motivated. The next step in the design process is to choose the gains so that the handling qualities are met, which means choosing gains so that the rotational modes gets the proper damping and eigenfrequency.

The design is built upon linear control theory and to know the open-loop transfer function the system is linearised around an operating point and from the linearisation calculate the closed-loop system. Below are the closed-loop transfer functions of the lateral and longitudinal control systems,

$$\frac{\Phi(s)}{\Phi_{cmd}(s)} = \frac{N_\phi(s)(K_{p\phi}s + K_i)}{(D_\phi(s) + N_p(s)K_{pp} + K_{p\phi}N_\phi(s))s + K_{i\phi}N_\phi(s)} \tag{3.2}$$

$$\frac{C^*(s)}{C^*_{cmd}(s)} = \frac{K_i(KN_q(s) + N_{n_z}(s))}{(D_C(s) + (KN_q(s) + N_{n_z}(s))K_p)s + (KN_q(s) + N_{n_z}(s))K_i}. \tag{3.3}$$

The $N_x(s)$ and $D_x(s)$ are the nominator and dominator of the linearised open-loop system were the subscript x denotes the output variable (q, n_z or ϕ). They have the forms

$$\begin{aligned} N_{n_z}(s) &= b_1 s + b_2 \\ N_q(s) &= b_1 s + b_2 \\ D_C(s) &= s^3 + \ldots + a_2 s + a_3 \\ N_\phi(s) &= b_1 s^2 + b_2 s + b_3 \\ N_p(s) &= b_1 s^3 + \ldots + b_3 s + b_4 \\ D_\phi(s) &= s^4 + \ldots + a_3 s + a_4. \end{aligned} \tag{3.4}$$

The next step is to make the closed-loop system satisfy the handling qualities by tuning the gains. The handling qualities are many and it is not clear how to do the tuning in a way that all are satisfied. To simplify, the gains were chosen by first mapping the handling qualities to the parameter space of the closed-loop system and then pick the gains graphically. The process of constructing the map and choosing the gains from the maps is explained further below.

Mapping theorem

At first sight the flight handling qualities look very complicated and many, but they can easily be illustrated as lines and curves in the complex plane. In Figure 3.3 the handling qualities of the short period mode are drawn together. The damping criterion becomes a straight line with the slope depending on the damping. The frequency criteria correspond to circles with a radius that equals the natural frequency. Together the circles and lines build up a closed boundary $\partial\Gamma$ in the complex plane. The boundary can be described as

$$\partial\Gamma := \{s \mid s = \sigma + j\omega(\sigma),\ \sigma \in [\sigma^-; \sigma^+]\}.$$

A line with constant damping D and a circle with constant natural frequency ω_n can be written as:

$$s = \sigma + j\sigma\frac{\sqrt{1-D}}{D} \tag{3.5}$$

$$s = \sigma + j\sqrt{\omega_n^2 - \sigma^2} \tag{3.6}$$

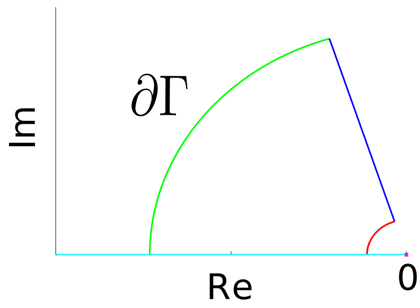

Figure 3.3: Handling qualities of the short period mode illustrated graphically. The inner and outer arch are the two CAP criteria and the slope is the damping.

The short period poles should be inside the boarder $\partial\Gamma$ to fulfill the handling qualities. Similar figures can be made for the lateral handling qualities. Many of the open-loop poles are already inside $\partial\Gamma$, but not all of them and the pilot might desire a uniform response in the whole envelope.

The linear controllers in Section 3.2 have two parameters, a proportional gain K_p and an integral gain K_i. A infinite number of combinations of K_p and K_i keep the closed-loop poles inside the boundary of $\partial\Gamma$ and thus a whole area in the parameter space of K_p and K_i gives an acceptable closed-loop system. One way to find the area is to map the boundary $\partial\Gamma$ into the parameter space $\boldsymbol{q}(K_p,K_i)$, see Figure 3.4.

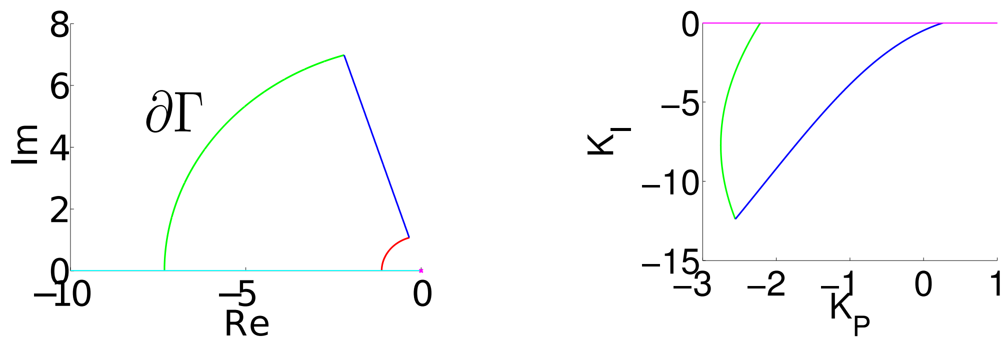

Figure 3.4: Mapping the handling qualities in the complex plane to the left into the parameter space on the right. Values inside of the borders fulfills the handling qualities in both of the figures.

Let the characteristic polynomial $p(s,\boldsymbol{q}) = [1\ s\ \ldots\ s^n]\boldsymbol{a}(\boldsymbol{q}) = [1\ s\ \ldots\ s^n][a_n\ a_{n-1}\ldots 1]^{\mathrm{T}}$ of the closed-loop system be the denominator of either (3.2) or (3.3) and the following set

$$Q_{Im}(\sigma) := \{\boldsymbol{q} \mid p(\sigma + j\omega(\sigma),\boldsymbol{q}) = 0,\ \sigma \in [\sigma^-;\sigma^+]\},$$

be the set of parameters $\boldsymbol{q}(K_p,K_i)$ that gives closed-loop poles on the boundary $\partial\Gamma$ of the handling qualities or, in other words having a root pair at $s = \sigma \pm j\omega(\sigma)$ of (3.5) or

(3.6). Then the following theorem[1] is a way to construct a map between the handling qualities and the parameter space $\boldsymbol{q}$.

Theorem 1 (Boundary Representation Theorem, Ackermann, Kaesbauer)
Consider a polynomial family $p(s,\boldsymbol{q}) = [1\ s\ \dots\ s^n]\boldsymbol{a}(\boldsymbol{q})$ *and the set*

$$Q_{Im}(\sigma) := \{\boldsymbol{q} \mid p(\sigma + j\omega(\sigma),\boldsymbol{q}) = 0,\ \sigma \in [\sigma^-;\sigma^+]\}$$

Now $\boldsymbol{q} \in Q_{Im}(\sigma)$ *if and only if*

$$\begin{bmatrix} d_0(\sigma) & d_1(\sigma) & ... & d_n(\sigma) \\ 0 & d_0(\sigma) & ... & d_{n-1}(\sigma) \end{bmatrix} \boldsymbol{a}(\boldsymbol{q}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

for some $\sigma \in [\sigma^-;\sigma^+]$*, where*

$$\begin{aligned} d_0(\sigma) &= 1 \\ d_1(\sigma) &= 2\sigma \\ &\vdots \\ d_{i+1}(\sigma) &= 2\sigma d_i(\sigma) - [\sigma^2 + \omega^2]d_{i-1}(\sigma), \quad i = 1,2,\dots,n-1 \end{aligned}$$

[9]
When the map between the handling qualities and the parameter space is determined, the controller can be designed by picking a K_p and K_i pair inside of the boundary and then satisfying the handling qualities. If one prefers a more damped controller it is easy to add another damping line in the complex plane and then apply the mapping theorem again. The result will be a more narrow area in the parameter space.

To illustrate, consider an example of a situation when the longitudinal open-loop system of the Alpha does not fulfill the handling qualities (see Figure 3.5). The Alpha has a velocity of 0.4 Mach and an altitude of 10 km and it can bee seen that the short period mode is slightly under-damped. By picking $K_p = -3.4$ and $K_i = -2$ the controller increases the damping a bit and push the pole inside the boundary of the handling qualities. The Alpha changes its dynamics as the altitude and the Mach number changes. Therefore, the same controller cannot be used for all altitudes and Mach numbers. Figure 3.6 illustrates how the boundaries in the mapping theorem moves when the Mach number increases.

3.5 Gain scheduling

Gain scheduling is the process of dividing a control design of a nonlinear system into a a desing of a number of linear time-invariant controllers. Each linear controller will have the same structure but different gains depending on the linearisation of the dynamics at different Mach numbers and altitudes. While the gain scheduling technique is widely

[1] The complete proof can be found at [9, p. 240].

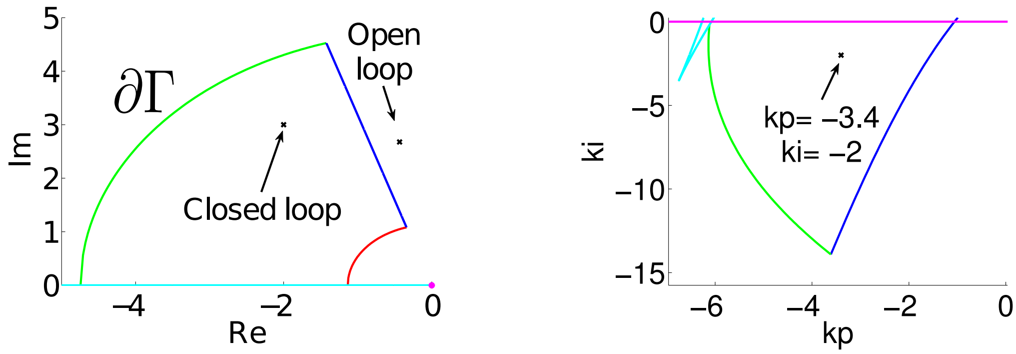

Figure 3.5: These two figures gives an example of when the open-loop pole lay outside of the handling qualities but when closing the loop and picking the gains inside the allowed region in parameter space the handling qualities are met.

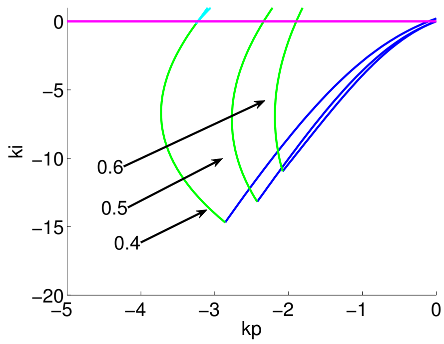

Figure 3.6: Handling qualities for the Mach numbers 0.4, 0.5 and 0.6 mapped into the parameter space of the closed loop system. The areas have high dependence on the Mach number.

used the idea is rather old. It originates from the 1960's and has successfully been used in both aerospace and process control [10]. In the flight community the gain scheduled controllers are often the starting point of control design. In this report two different gain scheduled controllers has been adopted, one longitudinal and one in the lateral direction (it assumes that the channels are decoupled), see section 3.2 and 3.3. Together they serve as the baseline controller of the Alpha.

The most obvious drawback of the gain scheduling process is the amount of work the design process takes. However, each step in the process is rather simple compared to other approaches. One major reason that the gain scheduling approach is still popular is that many safety certification requirements are specified in linear control theory. Below is a short summary of how it has been implemented here.

1. The flight envelope is first divided into a fine grid of equilibrium points by Mach number and altitude (see Section 2.7).

2. The dynamics of the Alpha are approximated by a linearisation in each equilibrium point and this is done numerically.

3. From each linearisation the open-loop transfer function is calculated.

4. Next step is to determine the closed-loop transfer function for the controller connected to the open-loop system.

5. With Ackermann's Boundary representation theorem the handling qualities are mapped into the parameter space of the closed-loop system.

6. Finally the controller gains are picked by choosing a gain-values inside the mapped curve in parameter space, such that the handling qualities are fullfilled.

7. The steps 2 to 6 are then repeated for each point in the envelope.

8. The final result is a mesh of controller gains covering the whole flight envelope. As the Alpha moves through the envelope its controller interpolate between the gains in the mesh.

3.6 Result

This section presents the simulation results of the linear controller. Two different simulations were made and the first is showing steps in the controled variables, angle of attack α and Euler bank angle ϕ. The second simulation shows coordinated turn, a spiral motion towards the ground. Both simulations were carried out at an altitude of 4 km and with a Mach number 0.24.

3.6.1 Simultaneous step

This manoeuvre starts out from a longitudinal steady state trim with zero rotation rates and a flight path angle of -10.6 deg. At time zero a pull-down is commanded with a C^* of -1 followed by step in the Euler bank angle of 20 degrees after five seconds (see Figure 3.7). The manoeuvre tests both the behaviour of the longitudinal and the lateral controller and the decoupling between them. It can be seen that there is not much coupling between the axes. The small dip in the curve during the C^* step comes from that the proportional gain is feedback in an inner-loop to decrease the overshoot. Worth to notice is the specially "smooth" step response the Euler bank angle shows. The sideslip angle is a bit oscillating since it it is not controlled. Some of the oscillation contributes to the small dip in C^*.

In Figure 3.8 the actuator deflections are shown and it can be seen that none of the actuators are close to saturation during the steps.

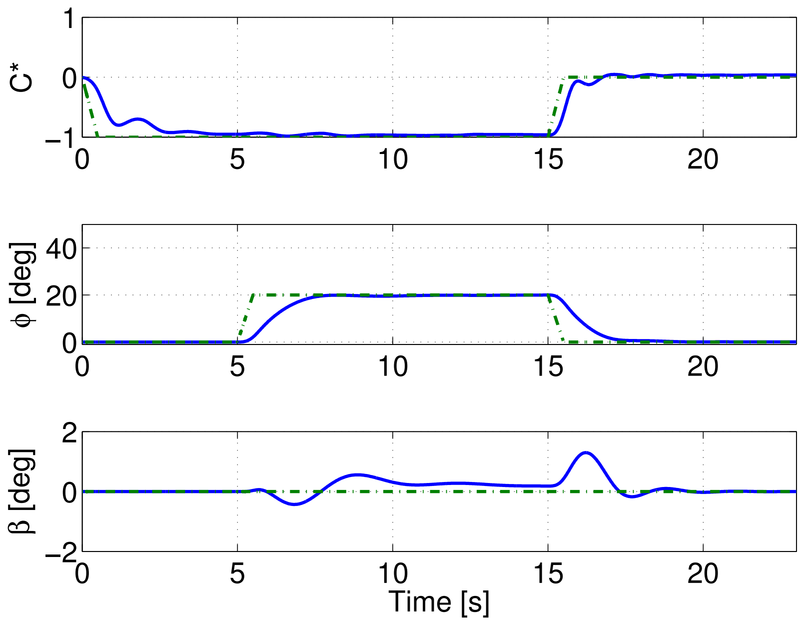

Figure 3.7: Simultaneous performing a pull-down and a roll. The dashed lines shows the reference signal.

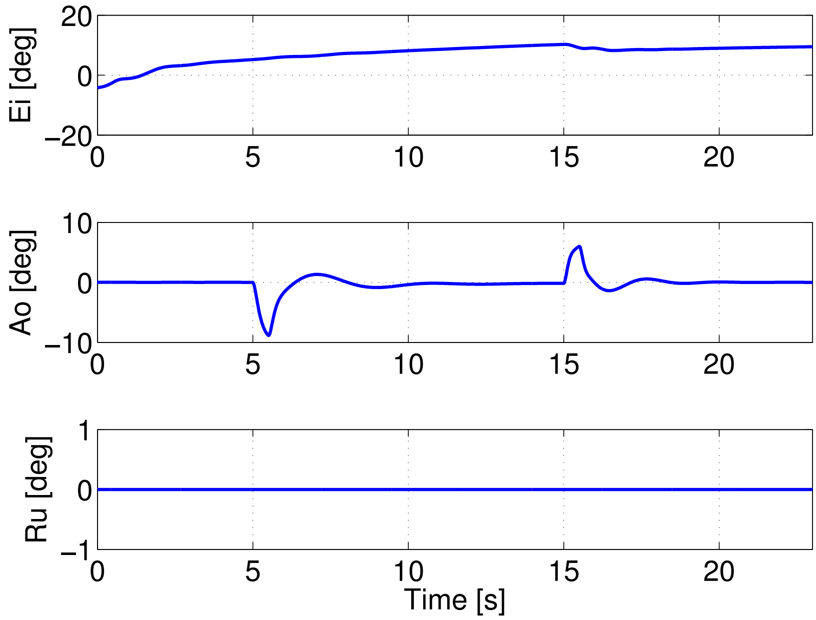

Figure 3.8: This figure show the deflection of the three actuators elevator (Ei), aileron (Ao) and rudder (Ru).

3.6.2 Coordinated turn

At the same starting altitude of 4 km was a coordinated turn initiated, starting from a steady level flight. The desired result of the coordinated turn is a spiral motion toward the ground with a constant flight path angle and a constant turn rate of 4 deg/s, which is the commanded turn rate or yaw rate $\dot{\psi}$. The turn rate may appear small but the maximum turn rate is at 8 deg/s and is only possible to achieve in parts of the envelope.

In Figures 3.9 and 3.10 the coordinated turn is shown in Earth coordinates, the more circular in shape and more sine-shaped the better the coordination turn. The first part of the manoeuvre is not circular because the Alpha starts out from a straight-level flight, but after a while does the Alpha fall in the commanded turn rate and the turn is just slightly elliptic.

The commanded values for C^* and Euler bank angle ϕ are not constant. They change with Mach number and altitude and they are taken from a trim table of coordinated turns together with the rudder value. How the controlled variables change during the envelope can be seen in Figure 3.11. In Figure 3.12 are the Euler angles bank, pitch and yaw shown. The yaw angle starts out at zero and increases slowly the first seconds until it falls in the commanded turn rate. In the end of the simulation the Euler yaw angle has reached 200 degrees in 50 seconds, which (200/50=4) corresponds to the commanded turn rate of 4 deg/s.

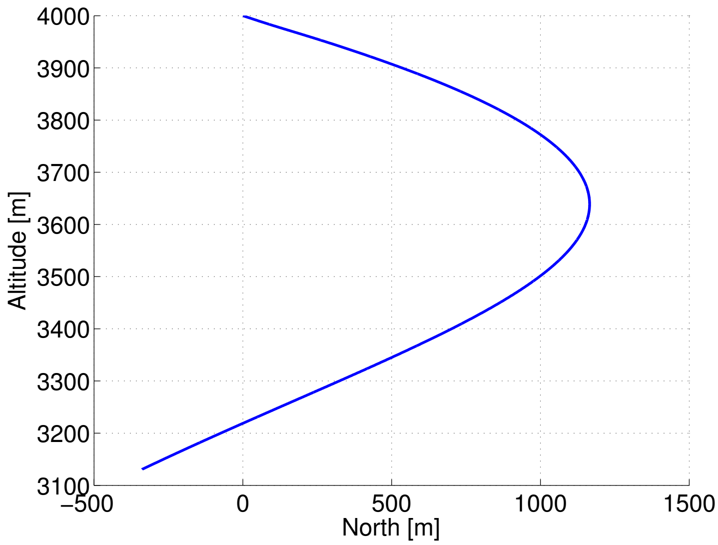

Figure 3.9: Here is the coordinated turn shown in north coordinates and altitude.

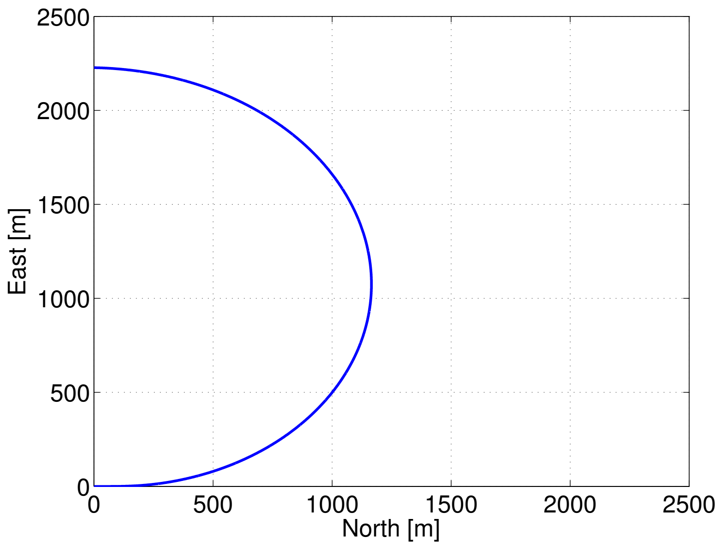

Figure 3.10: Here is the coordinated turn shown in north and east coordinates. The more circular, the better coordinated turn.

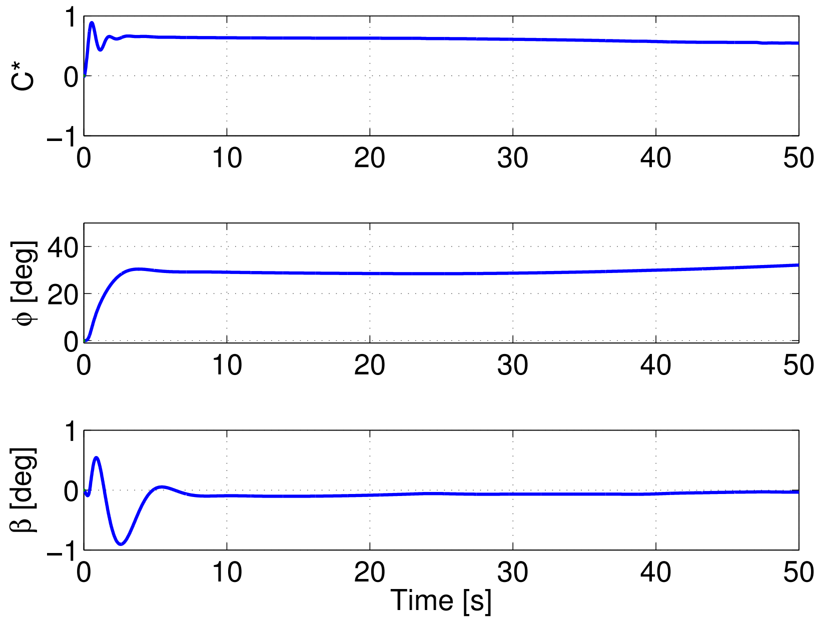

Figure 3.11: This figure shows the controlled variables and how they change as the Alpha moves in the flight envelope. The controlled variables are commanded to make a coordinated turn.

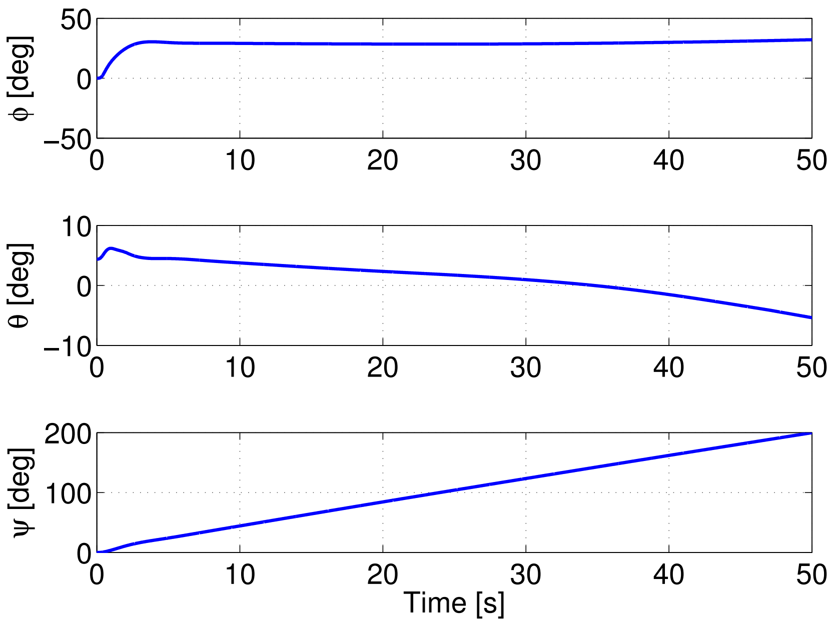

Figure 3.12: The Euler angles during the coordinated turn.

4

Nonlinear Control

This chapter will describe the design of a nonlinear controller using the backstepping technique. The idea with backstepping is to use Lyapunov theory and in a recursive way design the controller and at the same time guarantee stability.

The nonlinear controllers have a number of advantages compared to the linear controllers. One of the major advantages is that a nonlinear controller is designed using the nonlinear dynamics of the vehicle and thus designed in one piece, which can make the controller globally valid. The linear counterpart is designed using linearisation and is thus only valid locally. To deal with this the linear controllers are gain scheduled, a process that is very time consuming.

The backstepping controller synthesised in this chapter is on a vector form and the controlled variables are the angle of attack, sideslip angle, and the Euler bank angle phi. They are controlled using pitch, roll, and yaw rate. The model is extended to include actuator dynamics so that the rates are controlled by the moments, which in turn are manipulated through the moments derivatives. Finally an integrator state is added to handle modelling errors and to synthesis second order dynamics.

A related controller to backstepping is NDI (Nonlinear Dynamic Inversion) and the design process differs only in a few steps. The NDI controller implemented will also have actuator dynamics and the integrator states as the backstepping controller.

In the beginning of the chapter some Lyapunov theory will be presented, followed by how it is implemented in backstepping and NDI design.

4.1 Lyapunov theory

The design of a backstepping controller uses Lyapunov theory, which is part of the design process and guarantees stability. Before the backstepping procedure can be understood one has to understand some basic Lyapunov theory, starting with the definition of stability and presenting the key theorem used in the nonlinear control design.

A nonlinear system can be written as

$$\dot{x} = f(x), \tag{4.1}$$

where $f : D \to \mathbb{R}^n$ is a locally Lipschitz map from an open, connected domain $D \subset \mathbb{R}^n$ and let $x_0 \in D$ be the initial point of the system at time zero. Further, the system has an equilibrium at $x = 0 \ \in D$. Then the following definition will be used for the equilibrium [11].

Definition 1
The equilibrium point is said to be

- *Stable: if for each $\epsilon > 0$ there exists a $\delta(\epsilon) > 0$ such that*

$$\|x_0\| < \delta \ \Rightarrow \ \|x(t)\| < \epsilon, \ \forall \ t \geq 0$$

- *Unstable: if not stable*

- *Asymptotically stable: if stable and attractive, i.e., there exists a $r > 0$, and for each $\epsilon > 0$ a $T(\epsilon) > 0$ exist such that*

$$\|x_0\| < r \ \Rightarrow \ \|x(t)\| < \epsilon, \ \forall \geq T$$

Lyapunov theory is built upon mapping the states x of (4.1) on to a scalar function V that can be thought as the total energy of the system. The time derivative $\dot{V}$ along the solution trajectories to (4.1) then expresses how the energy of the system changes over time. If asymptotically stable, the system is moving to an equilibrium. The following theorem is used in Lyapunov theory to prove stability of a system on a domain D [11].

Theorem 2
Let $x = 0$ be an equilibrium point of the system (4.1) *and $D \subset \mathbb{R}^n$ be an open and connected domain containing $x = 0$ and let $V : D \to \mathbb{R}$ be a continuous, differentiable scalar function such that*

$$V(0) = 0, \quad V(x) > 0 \ \forall x \in D \setminus 0 \tag{4.2}$$

and

$$\dot{V}(x) = V_x^{\mathrm{T}}(x) f(x) \leq 0. \tag{4.3}$$

Then, the equilibrium $x = 0$ is stable.
Further, if

$$\dot{V}(x) = V_x(x) f(x) < 0 \ x \in D \setminus 0 \tag{4.4}$$

then the equilibrium is asymptotically stable.

A continuous and differentiable scalar function V, which fulfills (4.2) and (4.3) is called a *Lyapunov function.* The nonlinear controllers designed in this chapter needs a version of the Theorem that covers when the system is time varying. It will not be covered here, but can be found in [11, Chapter 4.5].

4.2 Lyapunov based control design idea

To explain how the Lyapunov based control design works, consider the nonlinear system

$$\dot{x} = f(x,y), \tag{4.5}$$

where x is the state vector and y is the input vector. If the control task is to bring the state x to the origin and D is the largest set that x is defined on, then the problem turns into how to design a control law $y = u(x)$ so that the origin becomes asymptotically stable on D. To show asymptotic stability Lyapunov theory can be used and Theorem 2 says that if we find a positive definite function $V(x)$ and a function $u(x)$ chosen such that

$$\dot{V} = V_x^{\mathrm{T}} f(x,u(x)) < 0, \ \ x \neq 0, \tag{4.6}$$

then the closed-loop system is asymptotically stable. It is important to note that the theorem does not say how to choose the function or the control function, but if (4.6) is satisfied it stability is guaranteed. One method of how to decide the control function is backstepping.

4.3 Backstepping

Lyapunov theory is a tool to prove stability of a system, but it does not give any tool of how to find the controller. Backstepping can be seen as one solution to that and gives a systematic way of finding a family of possible control laws.

The backstepping designed in this chapter has the objective to control the angle of attack, roll angle and sideslip angle and it is extended from the scalar case and written in vector form, called block backstepping [12]. The linear controller in the chapter before differs in the choice of longitudinal control variable. Here the angle of attack is the controlled variable but in the linear controller it was the C^* criteria, a mix between pitch rate and normal acceleration. The main difference is that C^* is a criteria designed for the pilot and is easier to transfer to a stick motion, while the choice of angle of attack is due to the restriction of having a control affine form (see Section 4.3.1). With the use of angle of attack there has to be a mapping between stick motion, desired aircraft motion and angle of attack change. A change in angle of attack of course gives a change in C^* and vise versa. This is important to keep in mind when comparing the linear and the nonlinear controller.

In the following pages the design process of a backstepping controller and an NDI controller are shown.

4.3.1 Control affine form

A general backstepping system has to be on a lower triangular form as the system,

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x},\boldsymbol{y}) \\
\dot{\boldsymbol{y}}_1 &= \boldsymbol{h}_1(\boldsymbol{x},\boldsymbol{y}_1,\boldsymbol{y}_2) \\
&\vdots \\
\dot{\boldsymbol{y}}_i &= \boldsymbol{h}_i(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_{i+1}) \\
&\vdots \\
\dot{\boldsymbol{y}}_m &= \boldsymbol{h}_m(\boldsymbol{x},\boldsymbol{y}_1,\ldots,\boldsymbol{y}_m,\boldsymbol{u}).
\end{aligned} \tag{4.7}$$

where $\boldsymbol{x} \in \mathbb{R}^n$, $\boldsymbol{y}_i \in \mathbb{R}^m$, $\boldsymbol{h}_i \in \mathbb{R}^n$, $\boldsymbol{f} \in \mathbb{R}^m$, the control input $\boldsymbol{u} \in \mathbb{R}^m$. When the system is written on vector form it is called *block* backstepping. Backstepping is made in a recursive manner and a lower subsystem of (4.7) is used to form a virtual control law $\boldsymbol{y} = \boldsymbol{y}^{\mathrm{d}}$ for a subsystem one level higher and $\boldsymbol{u}$ is the control input for the lowest subsystem. By designing virtual control laws in recursive steps and a Lyapunov function for the first system is known, higher order system can be controlled as long as they are on a lower triangular form.

In flight dynamics the restriction of having a system on a lower triangular form means that the actuators should only give moments and their force contribution should be small enough to be neglected. During control design it is found that forces from the actuators of the Alpha indeed were not small and this was handled by adding the actuator forces as an error to $\boldsymbol{f}(\boldsymbol{x},\boldsymbol{y})$. In addition of having a lower triangular system the following assumptions must hold [13]:

1. The lift and side force coefficients depends only on α and β.
2. The altitude and Mach number vary slowly and their time derivatives are neglected.

4.3.2 Controlled variables and objective

In this backstepping design there will be only four subsystems as in (4.7). However, we will start the design with the most basic system

$$\begin{aligned}
\dot{\boldsymbol{x}} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{g}(\boldsymbol{x})\boldsymbol{y} \\
\dot{\boldsymbol{y}} &= \boldsymbol{h}(\boldsymbol{y}) + \boldsymbol{k}\boldsymbol{u},
\end{aligned} \tag{4.8}$$

where the controlled outputs are the angle of attack, sideslip angle and the Euler bank angle

$$\boldsymbol{x} = [\alpha\ \beta\ \phi]^{\mathrm{T}},$$

and the virtual control vector used is the turn rates

$$\boldsymbol{y} = [p\ q\ r]^{\mathrm{T}}.$$

By comparing terms, the dynamics of Alpha (2.6), (2.8), (2.10) can be written on the form (4.8) with

$$\boldsymbol{f}(\boldsymbol{x}) = \begin{bmatrix} -\frac{L(\boldsymbol{x})}{Vm\cos\beta} \\ \frac{Y(\boldsymbol{x})}{Vm} \\ 0 \end{bmatrix} \tag{4.9}$$

,

$$\boldsymbol{g}(\boldsymbol{x}) = \begin{bmatrix} -\cos\alpha\tan\beta & 1 & -\sin\alpha\tan\beta \\ \sin\alpha & 0 & -\cos\alpha \\ 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \end{bmatrix}. \tag{4.10}$$

and

$$\boldsymbol{h}(\boldsymbol{y}) = -\boldsymbol{I}^{-1}(\boldsymbol{y}\times\boldsymbol{I}\boldsymbol{y}),\ \boldsymbol{k} = \boldsymbol{I}^{-1}. \tag{4.11}$$

In the above expression

$$L(\boldsymbol{x}) = -F_{Tx}\sin\alpha + F_{Tz}\cos\alpha \tag{4.12}$$

and

$$Y(\boldsymbol{x}) = -F_{Tx}\cos\alpha\sin\beta + F_{Ty}\cos\beta - F_{Tz}\sin\alpha\sin\beta \tag{4.13}$$

equals the lift and side force and $\boldsymbol{I}$ is the moment of inertia tensor. The moments

$$\boldsymbol{M}_b = [M_{xb}\ M_{yb}\ M_{zb}]^{\mathrm{T}}$$

in (2.6) will act as the control variable and will be denoted $\boldsymbol{u}$ from now on.

4.3.3 Change of coordinates

The objective of the flight control system is to follow the reference

$$\boldsymbol{x}_c = [\alpha_c\ \beta_c\ \phi_c]^{\mathrm{T}} \tag{4.14}$$

and not move the state vector $\boldsymbol{x}$ to its origin (only the sideslip angle should be controlled to zero). Therefore the formulation of (4.8) is modified to a tracking problem with a smooth time varying reference signal with the aim of bringing the error

$$\tilde{\boldsymbol{x}} = \boldsymbol{x} - \boldsymbol{x}_c \tag{4.15}$$

to zero. In the rest of the chapter the tracking formulation

$$\begin{aligned} \dot{\tilde{\boldsymbol{x}}} &= \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},t) + \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)\boldsymbol{y} \\ \dot{\boldsymbol{y}} &= \boldsymbol{h}(\boldsymbol{y}) + \boldsymbol{k}\boldsymbol{u} \end{aligned} \tag{4.16}$$

is used with the new system $\tilde{\boldsymbol{f}}$ equal to

$$\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},t) = \boldsymbol{f}(\tilde{\boldsymbol{x}} + \boldsymbol{x}_c) - \dot{\boldsymbol{x}}_c \tag{4.17}$$

and $\tilde{\boldsymbol{g}}$ to

$$\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t) = \boldsymbol{g}(\tilde{\boldsymbol{x}} + \boldsymbol{x}_c). \tag{4.18}$$

4.3.4 Actuator dynamics and integral state

The model in (4.16) is fairly easy to extend to include actuator dynamics and an integral state without making backstepping design much more complex. To add a first order actuator state a third row

$$\dot{\boldsymbol{u}} = \boldsymbol{v} \tag{4.19}$$

is added to (4.16) and from now on is it the moment derivative $\boldsymbol{v} \in \mathbb{R}^3$ that is the controller input. In the code implementation, the moment derivatives have to be mapped to the corresponding surface deflections. This is done by linearising the moments in real time and solving

$$\boldsymbol{v} = \frac{\partial \boldsymbol{M_b}}{\partial t} = \frac{\partial \boldsymbol{M_b}}{\partial \boldsymbol{\delta}}|_{\delta_0} \frac{\partial \boldsymbol{\delta}}{\partial t} \tag{4.20}$$

together with the actuator dynamics (2.15) to get the surface deflections $\boldsymbol{\delta}$.

By adding the state

$$\dot{\boldsymbol{\xi}} = -\boldsymbol{A}_\xi \boldsymbol{\xi} + \tilde{\boldsymbol{x}} \tag{4.21}$$

to (4.16) the controller will include a "leaky integral" action, where $\boldsymbol{A}_\xi \in \mathbb{R}^{3\times 3}$ is a "small" diagonal positive semidefinite matrix. Integral action is used both to handle modelling errors and to give the system second order dynamics. The extended model with integrator state and actuator dynamics has the following form

$$\begin{aligned}
\dot{\boldsymbol{\xi}} &= -\boldsymbol{A}_\xi \boldsymbol{\xi} + \tilde{\boldsymbol{x}} \\
\dot{\tilde{\boldsymbol{x}}} &= \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},t) + \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)\boldsymbol{y} \\
\dot{\boldsymbol{y}} &= \boldsymbol{h}(\boldsymbol{y}) + \boldsymbol{k}\boldsymbol{u} \\
\dot{\boldsymbol{u}} &= \boldsymbol{v}.
\end{aligned} \tag{4.22}$$

The next step in the backstepping design is to recursively form the virtual control laws and in the end the input control law $\boldsymbol{v}$.

4.3.5 Error variables

We introduce the two virtual control laws $\boldsymbol{y}_\mathrm{d}$ and $\boldsymbol{u}_\mathrm{d}$. Each one of them will be designed to stabilise the subsystem one "integration level above" so that $\boldsymbol{y}_\mathrm{d}$ will stabilise the $\tilde{\boldsymbol{x}}$ dynamics and $\boldsymbol{u}_\mathrm{d}$ will stabilise the $\boldsymbol{y}$ dynamics in (4.22). Introducing the error variables

$$\begin{aligned}
\boldsymbol{z}_1 &= \boldsymbol{y} - \boldsymbol{y}_\mathrm{d}(\tilde{\boldsymbol{x}},t) \\
\boldsymbol{z}_2 &= \boldsymbol{u} - \boldsymbol{u}_\mathrm{d}(\tilde{\boldsymbol{x}},\boldsymbol{z}_1,t)
\end{aligned} \tag{4.23}$$

and forcing them to zero makes $\boldsymbol{y}$ and $\boldsymbol{u}$ go towards their desired virtual control law values $\boldsymbol{y}_\mathrm{d}$, $\boldsymbol{u}_\mathrm{d}$ and stabilising the system.

If the system is rewritten in the new error variables it becomes

$$\begin{aligned}
\dot{\boldsymbol{\xi}} &= -\boldsymbol{A}_\xi \boldsymbol{\xi} + \tilde{\boldsymbol{x}} \\
\dot{\tilde{\boldsymbol{x}}} &= \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},t) + \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)(\boldsymbol{z}_1 + \boldsymbol{y}_\mathrm{d}(\tilde{\boldsymbol{x}},t)) \\
\dot{\boldsymbol{z}}_1 &= \boldsymbol{h}(\boldsymbol{z}_1,t) + \boldsymbol{k}(\boldsymbol{z}_2 + \boldsymbol{u}_\mathrm{d}(\tilde{\boldsymbol{x}},\boldsymbol{z}_1,t)) - \dot{\boldsymbol{y}}_\mathrm{d}(\tilde{\boldsymbol{x}},y,t) \\
\dot{\boldsymbol{z}}_2 &= \boldsymbol{v} - \dot{\boldsymbol{u}}_\mathrm{d}(\tilde{\boldsymbol{x}},\boldsymbol{z}_1,\boldsymbol{z}_2,t),
\end{aligned} \tag{4.24}$$

where the notations $\dot{\boldsymbol{y}}_\mathrm{d}$ and $\dot{\boldsymbol{u}}_\mathrm{d}$ refers to total time derivative along the solution trajectory. To calculate the $\dot{\boldsymbol{y}}_d$ and $\dot{\boldsymbol{u}}_d$ the time derivative of the lift and side forces have to be known, which depend on the state vector. To be able to differentiate them, the forces were approximated by two polynomials.

The question remains of how to chose the virtual control laws to stabilise the system. In the next section Lyapunov theory is applied to show stability and at the same time chose the two virtual control laws.

4.3.6 Block backstepping control law

Stability of the system (4.24) is shown by introducing the following positive definite Lyapunov function

$$V(\boldsymbol{\xi},\tilde{\boldsymbol{x}},\boldsymbol{z}_1,\boldsymbol{z}_2) = \frac{1}{2}\|\boldsymbol{\xi}\|^2_{Q_\xi} + \frac{1}{2}\|\tilde{\boldsymbol{x}}\|^2_{Q_x} + \frac{1}{2}\|\boldsymbol{z}_1\|^2_{Q_{z_1}} + \frac{1}{2}\|\boldsymbol{z}_2\|^2_{Q_{z_2}}, \tag{4.25}$$

where the matrices $\boldsymbol{Q}_\xi \in \mathbb{R}^3$, $\boldsymbol{Q}_x \in \mathbb{R}^3$, $\boldsymbol{Q}_{z_1} \in \mathbb{R}^3$ and $\boldsymbol{Q}_{z_2} \in \mathbb{R}^3$ are all diagonal positive definite and used as scaling between the variables. To achieve a stable system the virtual control laws $\boldsymbol{y}_\mathrm{d}$, $\boldsymbol{u}_\mathrm{d}$ and the control law $\boldsymbol{v}$ are chosen such that

$$\begin{aligned}\dot{V}(\boldsymbol{\xi},\tilde{\boldsymbol{x}},\boldsymbol{z}_1,\boldsymbol{z}_2) &= \boldsymbol{\xi}^\mathrm{T}\boldsymbol{Q}_\xi\dot{\boldsymbol{\xi}} + \tilde{\boldsymbol{x}}^\mathrm{T}\boldsymbol{Q}_x\dot{\tilde{\boldsymbol{x}}} + \boldsymbol{z}_1^\mathrm{T}\boldsymbol{Q}_{z_1}\dot{\boldsymbol{z}}_1 + \boldsymbol{z}_2^\mathrm{T}\boldsymbol{Q}_{z_2}\dot{\boldsymbol{z}}_2\\ &= \boldsymbol{\xi}^\mathrm{T}\boldsymbol{Q}_\xi(-\boldsymbol{A}_\xi\xi + \tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{x}}^\mathrm{T}\boldsymbol{Q}_x(\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},t) + \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)(\boldsymbol{z_1} + \tilde{\boldsymbol{y}}_\mathrm{d}))\\ &+ \boldsymbol{z}_1^\mathrm{T}\boldsymbol{Q}_{z_1}(\tilde{\boldsymbol{h}}(\boldsymbol{z}_1,t) + \boldsymbol{k}(\boldsymbol{z}_2 + \boldsymbol{u}_d) - \dot{\boldsymbol{y}}) + \boldsymbol{z}_2^\mathrm{T}\boldsymbol{Q}_{z_2}(\boldsymbol{v} - \dot{\boldsymbol{u}}_\mathrm{d})\end{aligned} \tag{4.26}$$

becomes negative definite and Theorem 2 ensures stability. It follows that by using the two virtual control laws

$$\begin{aligned}\boldsymbol{y}_\mathrm{d} &= \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)^{-1}(-\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},t) - \boldsymbol{A}_x\tilde{\boldsymbol{x}} - \boldsymbol{Q}_x^{-1}\boldsymbol{Q}_\xi\boldsymbol{\xi})\\ \boldsymbol{u}_\mathrm{d} &= \boldsymbol{k}^{-1}(-\boldsymbol{h}(\boldsymbol{z}_1,t) - \boldsymbol{A}_{z_1}\boldsymbol{z}_1 + \dot{\boldsymbol{y}}_\mathrm{d} - \boldsymbol{Q}_{z_1}^{-1}\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)^\mathrm{T}\boldsymbol{Q}_x\tilde{\boldsymbol{x}})\end{aligned} \tag{4.27}$$

and the input control law

$$\boldsymbol{v} = \dot{\boldsymbol{u}}_\mathrm{d} - \boldsymbol{A}_{z_2}\boldsymbol{z}_2 - \boldsymbol{Q}_{z_2}^{-1}\boldsymbol{k}^\mathrm{T}\boldsymbol{Q}_{z_1}\boldsymbol{z}_1 \tag{4.28}$$

(4.26) becomes

$$\dot{V}(\boldsymbol{\xi},\tilde{\boldsymbol{x}},\boldsymbol{z}_1,\boldsymbol{z}_2) = -\boldsymbol{\xi}^\mathrm{T}\boldsymbol{Q}_\xi\boldsymbol{A}_\xi\boldsymbol{\xi} - \tilde{\boldsymbol{x}}^\mathrm{T}\boldsymbol{Q}_x\boldsymbol{A}_x\tilde{\boldsymbol{x}} - \boldsymbol{z}_1^\mathrm{T}\boldsymbol{Q}_{z_1}\boldsymbol{A}_{z_1}\boldsymbol{z}_1 - \boldsymbol{z}_2^\mathrm{T}\boldsymbol{Q}_{z_2}\boldsymbol{A}_{z_2}\boldsymbol{z}_2. \tag{4.29}$$

The matrices $\boldsymbol{A}_\xi \in \mathbb{R}^{3\times3}$, $\boldsymbol{A}_x \in \mathbb{R}^{3\times3}$, $\boldsymbol{A}_{z_1} \in \mathbb{R}^{3\times3}$ and $\boldsymbol{A}_{z_2} \in \mathbb{R}^{3\times3}$ are all diagonal and positive semidefinite and (4.29) becomes negative semidefinite. It is possible to have off diagonal elements but keeping them diagonal decouples the system.

If the control laws (4.27) and (4.28) are inserted into (4.24), the closed-loop system takes the following form

$$\begin{aligned}\dot{\boldsymbol{\xi}} &= -\boldsymbol{A}_\xi\boldsymbol{\xi} + \tilde{\boldsymbol{x}}\\ \dot{\tilde{\boldsymbol{x}}} &= -\boldsymbol{A}_x\tilde{\boldsymbol{x}} + \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)\boldsymbol{z}_1 - \boldsymbol{Q}_x^{-1}\boldsymbol{Q}_z\boldsymbol{\xi}\\ \dot{\boldsymbol{z}}_1 &= -\boldsymbol{A}_{z_1}\boldsymbol{z}_1 + \boldsymbol{k}\boldsymbol{z}_2 - \boldsymbol{Q}_{z_1}^{-1}\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)^\mathrm{T}\boldsymbol{Q}_x\tilde{\boldsymbol{x}}\\ \dot{\boldsymbol{z}}_2 &= -\boldsymbol{A}_{z_2}\boldsymbol{z}_2 - \boldsymbol{Q}_{z_2}^{-1}\boldsymbol{k}^\mathrm{T}\boldsymbol{Q}_{z_1}\boldsymbol{z}_1,\end{aligned} \tag{4.30}$$

where it can be seen that the matrices $\boldsymbol{A}_\xi$, $\boldsymbol{A}_x$, ($\boldsymbol{A}_{z_1}$ and $\boldsymbol{A}_{z_2}$) can be used to tunethe *desired dynamics* of the system.

4.3.7 NDI

Nonlinear Dynamic Inversion is very related to the backstepping controller shown above [14]. The difference between the NDI controller and the backstepping controller is that NDI uses that the system is time-scale separated to simplify the control laws (4.27) and (4.28) by dropping terms. By time-scale separation means that the system has a "slow" part and a "fast" part [15]. The dynamics of $\boldsymbol{x}$ are considered as slow and $\boldsymbol{y}$ dynamics as fast in (4.8). If the $\boldsymbol{y}$ dynamics are fast enough they can successfully be used as virtual control laws for the $\boldsymbol{x}$ system.

The NDI controller will have the following simplified control laws

$$\begin{aligned} \boldsymbol{y}_\mathrm{d} &= \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}},t)^{-1}(-\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},t) - \boldsymbol{A}_x\tilde{\boldsymbol{x}} - \boldsymbol{Q}_x^{-1}\boldsymbol{Q}_\xi\boldsymbol{\xi}) \\ \boldsymbol{u}_\mathrm{d} &= \boldsymbol{k}^{-1}(-\boldsymbol{h}(\boldsymbol{z}_1,t) - \boldsymbol{A}_{z_1}\boldsymbol{z}_1) \\ \boldsymbol{v} &= -\boldsymbol{A}_{z_2}\boldsymbol{z}_2. \end{aligned} \tag{4.31}$$

4.4 Result

This section presents the result of the nonlinear simulations of the two nonlinear control laws backstepping and NDI.

First, a simulation similar to the simultaneous step in the linear controller is shown. Next, two different simulations illustrates the performance of steps with different amplitude. Also a "staircase" structured simulation is illustrated to show the result of commanding an angle of attack in Earth coordinates.

Finally, the backstepping controller is compared to the NDI controller.

4.4.1 Backstepping simulations

The following simulations were performed with the backstepping controller. The first manoeuvre starts out from a longitudinal trim with zero rotation rates at 4 km, a flight path angle of -10.6 deg and a Mach number of 0.24. After five seconds a step is introduced in both the angle of attack and the Euler bank angle. The sideslip angle is commanded to stay at zero during the whole manoeuvre. After 15 seconds the Alpha is commanded to pull-up and at the same time bank back to the initial position. In Figure 4.1 it can be seen that the resulting responses are "smooth" and satisfying with no coupling between the axis and the sideslip angle returns shortly after the steps. In Figure 4.2 are the corresponding surface deflections shown.

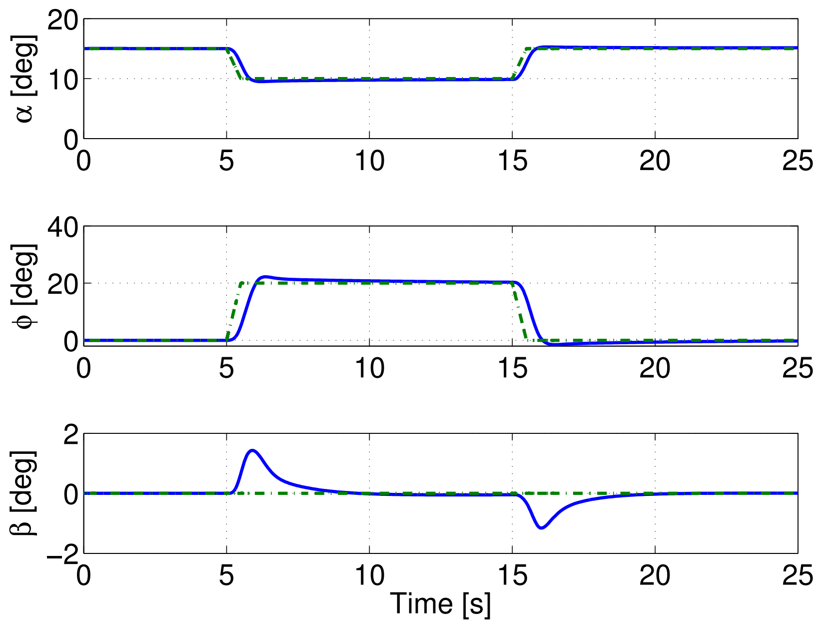

Figure 4.1: A pull-down simultaneous as a roll. The reference signal is in dashed lines.

In the next simulation, steps of 10 deg, 15 deg and 20 deg in angle of attack was performed. The results are shown in Figure 4.3. They were performed at an altitude of

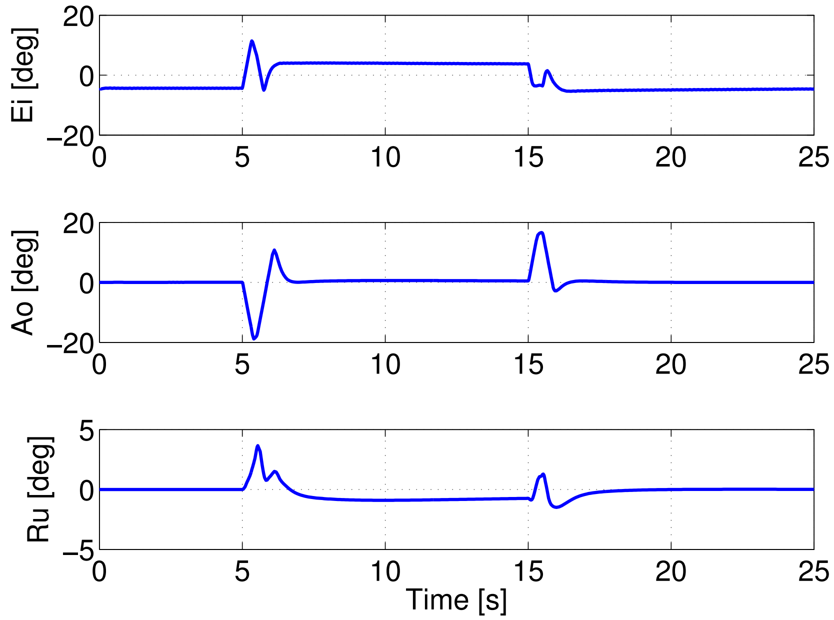

Figure 4.2: This figure shows the performance of the surface deflections.

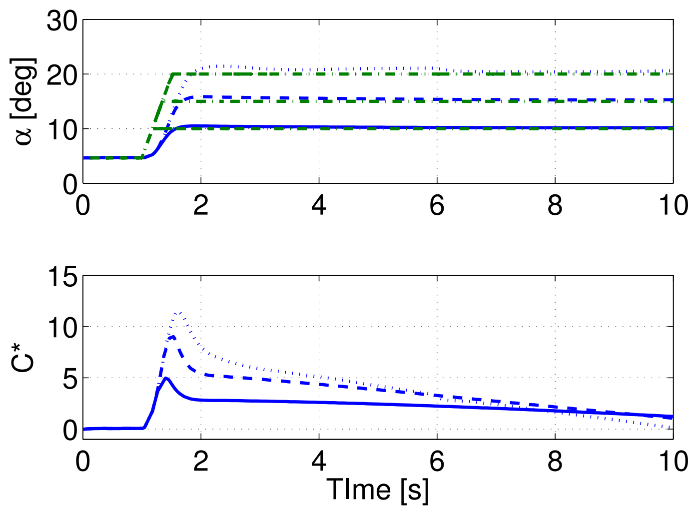

Figure 4.3: This figure shows three steps with the amplitudes 10 deg, 15 deg and 20 deg in angle of attack. In the upper figure is the angle of attack shown and in the lower figure is the corresponding C^*. The reference signal is shown in dot dashed lines.

1 km and a Mach number of 0.35.

First does the simulation show how the controller perform at a lower altitude with different step amplitudes but secondly how the corresponding move would look like in the C^* parameter used in the linear controller. It can be clearly seen that the high angle of attack, 20 deg, is troublesome for the controller and there is some oscillation.

More interesting is how the commanded steps look like in the C^* parameter. A step in angle of attack gives a short but sharp "bump" of two seconds in C^* and then slowly decreasing. The slowly decreasing C^* is due to that the Mach number is decreasing and the lift force goes down. The figure also shows that an extra commanded angle of attack of 5 deg gives a C^* of around 3 deg. This illustrates the need of a mapping between the desired pilot motion and the commanded angle of attack.

The next figure shows steps of 20 deg, 40 deg and 60 deg in the Euler bank angle (see Figure 4.4). They were all performed at an altitude of 1 km and a Mach number of 0.35.

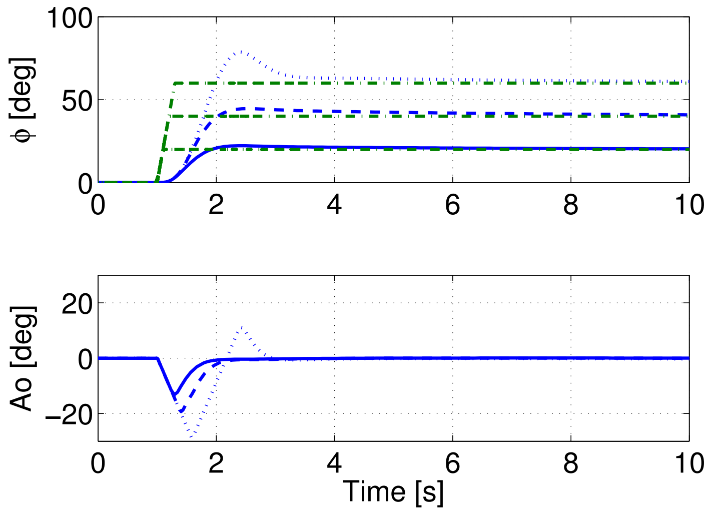

Figure 4.4: This figure shows three steps in the Euler bank angle of 20 deg, 40 deg and respectively 60 deg.

In the lower figure are the aileron deflection shown and it can be seen that the curves are triangular shaped because of the rate limit. When a step of 60 deg is commanded the controller is demanding a deflection beyond -20 deg and it is saturated and therefore the larger overshoot.

The next simulation is started at 4 km and at a Mach number of 0.24. The simulation is as long as 50 seconds and the Alpha will be flying down to an altitude below 2 km.

The simulation starts out in a longitudinal trim and after a second a pull-down of 5 deg is commanded and ten seconds later one more pull-down is commanded. The pattern is repeated until an angle of attack of 0 deg is reached, when instead pull-ups are commanded until the Alpha reaches the initial angle of attack of 15 deg. The resulting flight envelope has a "staircase" structure (see Figure 4.5).

The manoeuvre shows the performance of pull-ups and pull-downs at different altitudes and initial conditions and each step is alike. It can also be seen how the elevator is working with different dynamic pressure due to the altitude and Mach number change. For each step a smaller deflection is needed. In Figure 4.6 the flight in Earth coordinates

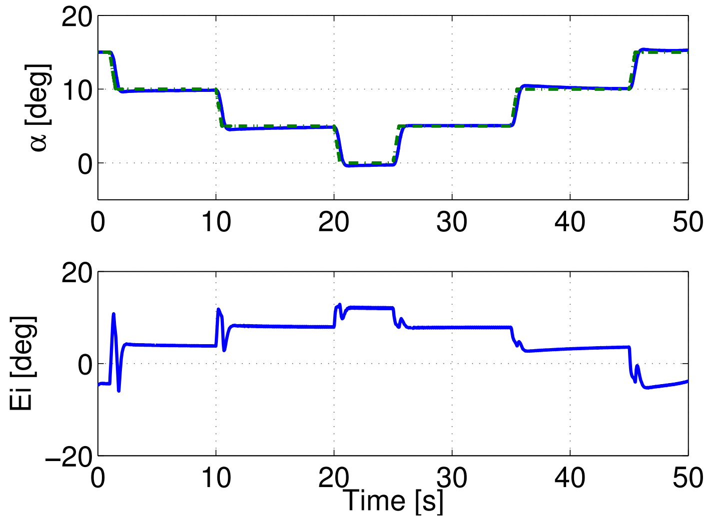

Figure 4.5: This figure shows pull-down and pull-ups by 5 deg in angle of attack in a ”staircase” structure. The simulation started at an altitude of 4 km. The reference signal is in dashed lines.

is shown. It illustrates that with an altitude below 2 km (and a higher Mach number) a

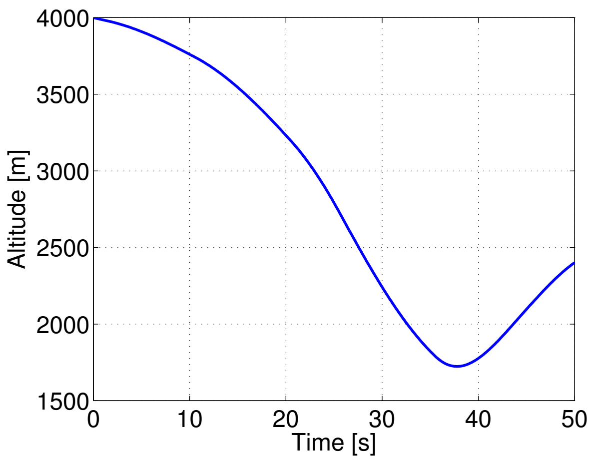

Figure 4.6: In this figure the Alphas movement is shown in Earth coordinates when performing the staircase steps in 4.5.

commanded angle of attack of 15 km is a powerful pull-up, even though it was a straight level flight initially. This shows again the difference between the C^* parameter and the control of angle of attack.

4.4.2 Backstepping and NDI

In Figure 4.7 and 4.8 the backstepping controller is compared to the NDI controller. The simulations took place at an altitude of 1 km and a Mach number of 0.35. In the first simulation (Figure 4.7) a step in angle of attack of 5 deg was ordered by both controllers. In the next simulation (Figure 4.8) a step in the Euler bank angle of 20 deg was made.

Both controllers are designed to have the similar characteristics as far as possible and it can be seen in Figure 4.7 that both controllers have always the same longitudinal response and actuator usage. The response to a step in Euler bank angle shows that the NDI controller is slightly slower, but at the same time it has slightly more rate saturated actuator. More comparisons between the controllers are made in the clearance test in the next chapter.

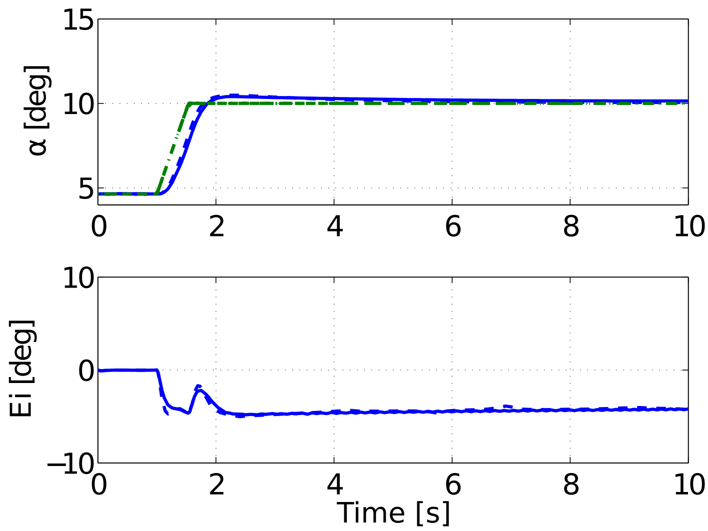

Figure 4.7: This figure shows a comparison between the backstepping controller (solid) and the NDI controller (dashed). A step of 5 deg in angle of attack is performed at 1 km and Mach number 0.35. The reference signal is in dot-dashed lines.

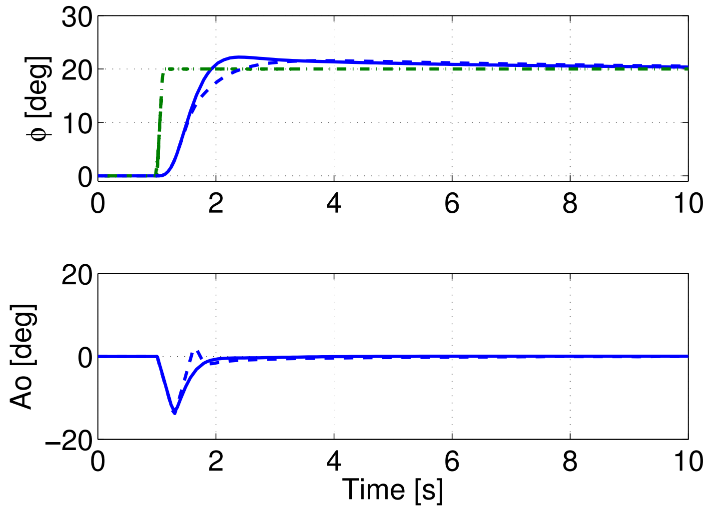

Figure 4.8: This figure shows a comparison between the backstepping controller (solid) and the NDI controller (dashed). A step of 20 deg in Euler bank angle is performed at 1 km and Mach number 0.35. The reference signal is in dot-dashed lines.

5

Clearance of flight control laws

The Alpha has to be functional at different altitudes, Mach numbers and manoeuvres. Therefore a design is put into further development a computer based clearance test is both a cheap and fast way to test the performance of the control laws.

When commanding high angle of attack and rotation rates the dynamics of the Alpha will be more nonlinear and cross-couplings between different axis are stronger. Demanding flight commands can both saturate the control surfaces and their rates, which give more nonlinearities that are not included in the control law design.

It is also important to know how sensitive the Alpha is to a change in the position of the mass center or other parameters.

In this chapter it is investigated how the control laws withstand a clearance test of worst pilot input. The test was performed with a global optimization method called differential evolution. The chapter starts with a brief explanation of global optimisation and differential evolution, followed by the clearance test.

5.1 Global optimisation methods

The clearance test will be performed with a global optimisation method called differential evolution, which is a statistical method. The most widely used statistical method in the industry is Monte Carlo simulations, which are more computationally expensive in contrast to global optimisation methods [16].

Other approaches are analytical linear robustness tests as gain and phase margins or nonlinear continuation or bifurcation analysis against single parameter variations.

In the following sections global optimisation and differential evolution will be briefly introduced.

5.1.1 Problem formulation

This section will only serve as a brief introduction to global optimisation and more can be read in the original paper on differential evolution [17].

A general optimisation problem is to find the "best" or optimal solution to a problem that can be stated as

$$\max_{x} \; f(\boldsymbol{x}) \tag{5.1}$$

subject to

$$\boldsymbol{x} \in \mathbb{R}^n \tag{5.2}$$

and the constraints

$$x_{min} \leq x \leq x_{max}, \tag{5.3}$$

where $f(\boldsymbol{x})$ is called the cost or object function and $\boldsymbol{x}$ is called optimisation parameters.

5.1.2 Differential evolution algorithm

Differential evolution is a global search method that is easy to use with few control parameters and it is also easy to run in parallel on different cores [17]. The algorithm starts by choosing a first population of N parameter vectors by random. During the optimisation the population size stays fixed for each iteration but the elements mutate by combining the parameters from two individuals and add them to a third population member. If the new mutation gives a lower objective function value than the original it will take its place in the population. It can be seen as the population evolve from the iteration before in a similar way as in a living population evolve for each generation by survival of the "fittest". There are different variants of the differential evolution algorithm but the one used here works in the following way [3];

Initialise a population
Evaluate fitness for each member
while not ready
    for each member $\boldsymbol{x}_j$ in population of size N
        Mutation;
        Crossover;
        Evaluate;
        Select;
    end
end

In the *mutation* step three different individuals ($\boldsymbol{x}_{j_1}$ $\boldsymbol{x}_{j_2}$ $\boldsymbol{x}_{j_3}$) are selected at random to build up a mutation vector $\boldsymbol{v}_j$ through

$$\boldsymbol{v}_j = \boldsymbol{x}_{j_1} + F(\boldsymbol{x}_{j_2} - \boldsymbol{x}_{j_3}). \tag{5.4}$$

None of the three individuals can be the current member $\boldsymbol{x}_j$ in the "for loop" and here F is a scale factor that can be used as a tuning parameter, but its default value was used.

Next, a new member $\boldsymbol{u}_j$ is formed by component-wise selecting at random parts from the mutant vector $\boldsymbol{v}_j$ or the current member $\boldsymbol{x}_j$ with some fixed probability and this step is called *crossover*. Then, the objective function value $f(\boldsymbol{u}_j)$ is *evaluated* and compared to the objective value of the original member $\boldsymbol{x}_j$. The one with the lowest objective function value is *selected* and kept for the next iteration.

5.2 Clearance test

Before finishing the design of a control law it is important to do a clearance test. The more that can be foreseen in a computer simulations before a "real" flight test the better, since changes after the simulation stage can be both expensive and time consuming.

The Alpha will fly over a wide range of altitudes and Mach numbers. When the Mach number changes rapidly many aerodynamic parameters will change rapidly and it is therefore important to know how the Alpha will respond to that. The behaviour when commanding high angle of attack and roll combinations are also interesting.

The following clearance test will focus on finding the "worst-case" pilot input as in [18] and [3]. By worst-case pilot input means a pilot stick combination that gives the largest overshoot, or even instability. The overshoot for a step is defined as,

$$\text{overshoot} = \frac{\text{actual} - \text{commanded}}{\text{commanded}}. \tag{5.5}$$

The test will also include the effect of moving the center of mass as well as lowering the actuators deflection rate. A saturated deflection rate can be one reason for PIO (Pilot Induced Oscillations).

5.2.1 Pilot input parameterisations

To find the worst-case pilot input three different parameterisations of stick combinations were used. They are presented below and each one is a sequence of steps with different amplitudes and step length.

Parameterisation 1 In the first parameterisation the optimiser is commanding a sequence of pull-up and pull-down steps in angle of attack α and Euler bank angle ϕ, with the fixed amplitudes,

$$\{0 \ \Delta\alpha_{max}\}$$

and

$$\{\Delta\phi_{min} \ \Delta\phi_{max}\}.$$

By Δ means an α or ϕ value Δ degrees from the straight level trim value of α and ϕ. The search space $\boldsymbol{x}$ as in (5.2) is defined as the relative time instances

$$[\Delta t_{1_\alpha} \ \Delta t_{2_\alpha} \ \Delta t_{3_\alpha} \ \Delta t_{4_\phi} \ \ldots \ \Delta t_{7_\phi}] \ \in [0.5 \ 4] \text{ sec},$$

to perform the steps. Totally, the angles will change at seven different time instances t_i (three different commanded angle of attack and four different Euler bank angle), where

$t_{i_\alpha} = \sum_1^{i_\alpha} \Delta t_{i_\alpha}$ and $t_{i_\phi} = \sum_4^{i_\phi} \Delta t_{i_\phi}$. Figure 5.1 illustrates the step sequence in only the angle of attack, but it looks equivalent for the Euler bank angle. After the step

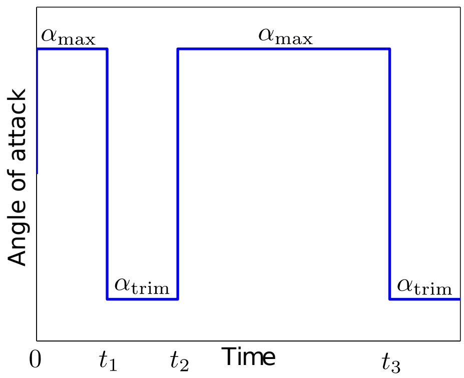

Figure 5.1: Illustration of the step sequence in Parameterisation 1 in angle of attack. The sequence in Euler bank angle looks equivalent.

sequence the trim values in angle of attack and Euler bank angle (α_{trim} and ϕ_{trim}) are commanded.

The aim of the optimiser is to find combinations of Δt_1 to Δt_7 that gives the worst-case, which means the largest overshoots and sideslip. The optimisation problem is stated as to maximise a combination of the overshoot in angle of attack, Euler bank angle and sideslip angle. The cost function is defined as

$$f(\boldsymbol{x}) = \beta_{max}^2 + \alpha_{overshooot}^2 + \phi_{overshoot}^2. \tag{5.6}$$

Parameterisation 2 This parameterisation is similar to the first, but the steps in angle of attack are also allowed to take values below the trimmed value,

$$\{\Delta\alpha_{min}\ \Delta\alpha_{max}\}.$$

The steps in Euler bank angle are done with the same fixed amplitude as in Parameterisation 1,

$$\{\Delta\phi_{min}\ \Delta\phi_{max}\}.$$

As in Parameterisation 1 the optimiser is finding combinations of

$$[\Delta t_{1_\alpha}\ \ldots\ \Delta t_{4_\alpha}\ \Delta t_{5_\phi}\ \ldots\ \Delta t_{8_\phi}]\ \in [0.5\ 4]\ \text{sec},$$

that gives the worst-case pilot input. The step sequence is illustrated in Figure 5.2.

The same cost function (5.6) as in Parameterisation 1 were used.

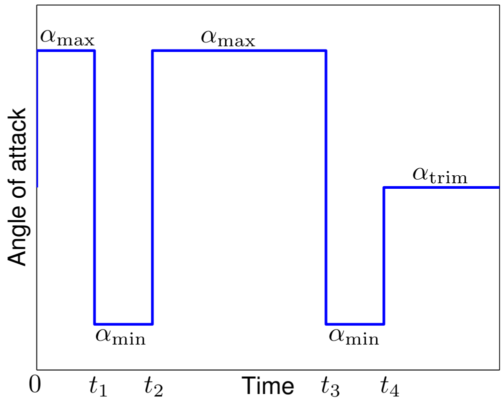

Figure 5.2: Illustration of the step sequence in Parameterisation 2 in angle of attack. The sequence in Euler bank angle looks equivalent.

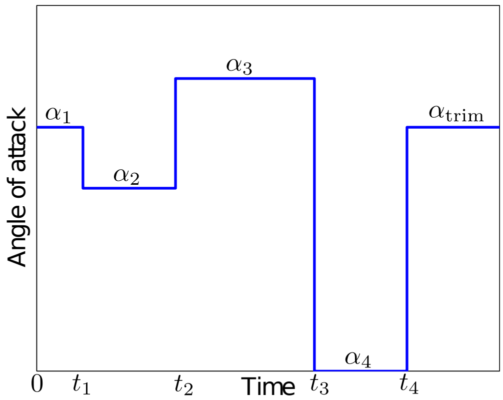

Figure 5.3: Illustration of the step sequence in Parameterisation 3 in angle of attack. The sequence in Euler bank angle looks equivalent.

Parametrisation 3 The third parameterisation will allow the optimiser to vary both the time instances of the steps as well as their amplitudes, see Figure 5.3. As in the first two parameterisations both angle of attack and Euler bank angles are commanded and the following will be the optimisation parameters,

$$[\Delta t_1 \ \ldots \ \Delta t_8] \ \in [0.5 \ 4],$$

$$[\Delta \alpha_1 \ \ldots \ \Delta \alpha_4] \ \in [\Delta \alpha_{min} \ \Delta \alpha_{max}]$$

and

$$[\Delta \phi_1 \ \ldots \ \Delta \phi_4] \ \in [\Delta \phi_{min} \ \Delta \phi_{max}].$$

Here, the Δt_1 to Δt_4 are the time instances of when a step in angle of attack is applied with amplitude $\Delta \alpha_1$ to $\Delta \alpha_4$ from the trim value and Δt_5 to Δt_8 tells when a step with the amplitude $\Delta\phi_1$ to $\Delta\phi_4$ from the trim value in Euler bank angle. There 16 parameters in total, which makes the convergence slower compared to the first two parameterisations, but it is also much more general in allowed pilot input. In the end of the step sequence, the trim values in angle of attack and Euler bank angle (α_{trim} and ϕ_{trim}) are commanded. As in the first two parameterisations the optimiser is looking for how to maximise the angle of attack and the Euler bank angle. The cost function is defined as

$$f(\boldsymbol{x}) = \alpha_{max}^2 + \frac{1}{4}\phi_{max}^2. \tag{5.7}$$

The clearance test was applied on the two nonlinear controllers, backstepping and NDI. All three parameterisations were tried with both controllers at the two altitudes 2 km and 4 km. The first two parameterisations were used to build up rough maps of the controller limits with far less numbers of iterations (fixed at 300) than Parameterisation 3, which was run until convergence. The maps were done by running several optimisations, but with increased maximum and minimum amplitudes ($\Delta\alpha_{min}$, $\Delta\alpha_{max}$, $\Delta\phi_{min}$ and $\Delta\phi_{max}$) for each optimisation. This is a way to illustrate the problem more than giving a proof of concept of how to make a clearance test with global optimisation.

In one test the center of mass was moved 5% aft (around 25 cm) to see the effect it had on the backstepping controller and this was only done at the lower altitude of 2 km. Also, the actuators deflection rate were decreased to as low as 30 degrees per second to search for PIO at the altitude 2 km(see Section 5.2.2). In the end, the two controllers were tried with Parameterisation 3 at 4 km again with an actuator deflection rate of 50 and 100 degrees per second. In Table 5.1 all simulations are summarised.

Table 5.1: A summary of the different clearance tests. Δxcg denotes a movement of the mass center in percent.

Test	Parametrisation	Alt [km]	Rate lim [deg/s]	Δxcg %	$\Delta\alpha$ [deg]	$\Delta\phi$ [deg]
1	1	2	50	0	[0 12]	[−50 50]
2	1	2	30	0	[0 12]	[−50 50]
3	1	2	50	5	[0 12]	[−50 50]
4	2	4	50	0	[−8 8]	[−50 50]
5	2	4	100	0	[−8 8]	[−50 50]
6	3	2	50	0	[0 8]	[−38 38]
7	3	4	50	0	[−4 4]	[−38 38]

5.2.2 Pilot induced oscillations

Pilot Induced Oscillation (PIO) is an unwanted oscillation of the aircraft that comes from an extreme combination of the pilot input and the aircraft. There are many reasons for PIO but some of them are, rate saturated actuators, high gain controllers, system delays and phase lags. One of the first and most famous cases of PIO happened on the first flight of the X-15 [19]. The reason of the PIO was rate saturated actuators.

When the actuator rate saturates it causes a "time delay" and a "gain reduction" as in Figure 5.4. The time delay initiates an unwanted oscillation of the aircraft, which can be seen in Figure 5.5. In the clearance test many PIO showed up, both because of less

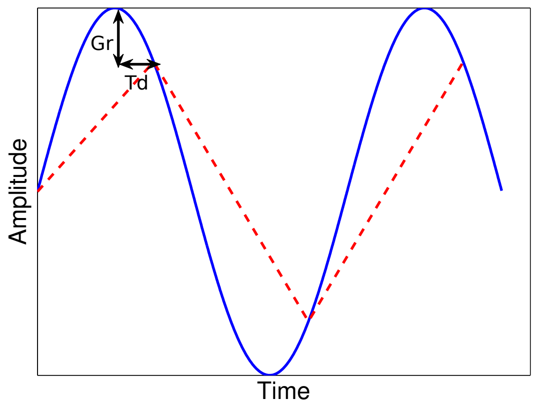

Figure 5.4: This figure illustrates the upcoming of the "time delay" Td and "gain reduction" Gr when the actuators saturates in a PIO. The solid line is the controller output and the dashed line is the actual actuator deflection.

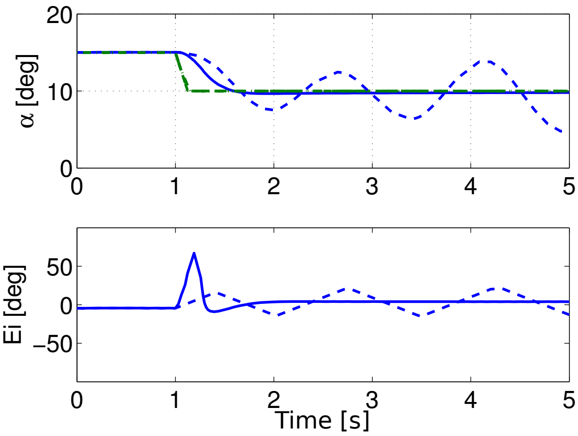

Figure 5.5: This figure shows a PIO coming from saturated actuators. The top figure shows a pull-down in the angle of attack and the lower shows the elevator deflection. The blue dashed lines correspond to a rate limit of 50 deg/s and the blue solid lines when there is no rate limit exists. The green dashed line is the commanded signal. The PIO is so strong that it is unstable.

dense atmosphere at high altitudes and of experimenting with different rate saturations.

5.3 Results

In this section the result from the clearance test are presented. The clearance test was applied on the two nonlinear controllers using the three parameterisations described in 5.2.1. A summary of the different test can be found in Table 5.1.

Test 1 This test was carried out at an altitude of 2 km, a Mach number of 0.35 and a flight path angle of -23.6 deg, when the pilot needs much maneuverability for tracking the landing site. The test was done with Parameterisation 1. In Figure 5.6 the result for the backstepping controller is shown. Each colored square in the figures corresponds

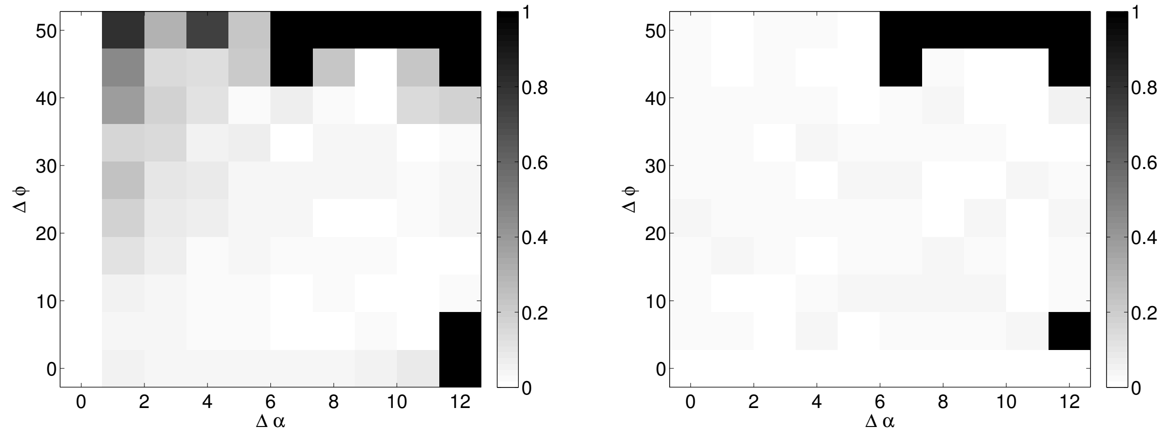

Figure 5.6: Clearance Test 1 for the backstepping controller. To the left is the angle of attack overshoot and to the right the Euler bank angle overshoot.

to the worst-case overshoot after 300 iterations in one optimisation with a fixed $\Delta\alpha$ (x-axis) and $\Delta\phi$ (y-axis). The grey scale illustrates the overshoot. The axis $\Delta\phi$ and $\Delta\alpha$ are the maximum fixed step size the optimiser can command from the trim point in parameterisation 1 (see Figure 5.1). For example, when $\Delta\phi = 50$ deg the optimiser is commanding steps from -50 deg to 50 deg. The figures are truncated at an overshoot larger than 1 (overshoots larger than 100%) and the black areas at the edges shows points when the Alpha becomes unstable and even departure so far that it reaches outside the limits of the aerodata of the angle of attack and Mach number.

In the left of Figure 5.6 the overshoot of the angle of attack is shown and it can be seen that for small commanded angle of attack and large commanded bank angles the overshoots are large. This shows that the decoupling is not ideal and large commanded bank angles affects the angle of attack. The right figure shows the overshoot in Euler bank angle. The backstepping controller is stable under most of the possible commanded angles. Worth to notice are the dark spots in the lower right corners of the figures. In those spots the optimiser holds a pull-up as long as possible until the Alpha stalls.

In Figure 5.7 the same test is carried out on the NDI controller. The dark areas are significantly larger than for the backstepping controller. Some of it could be due to the

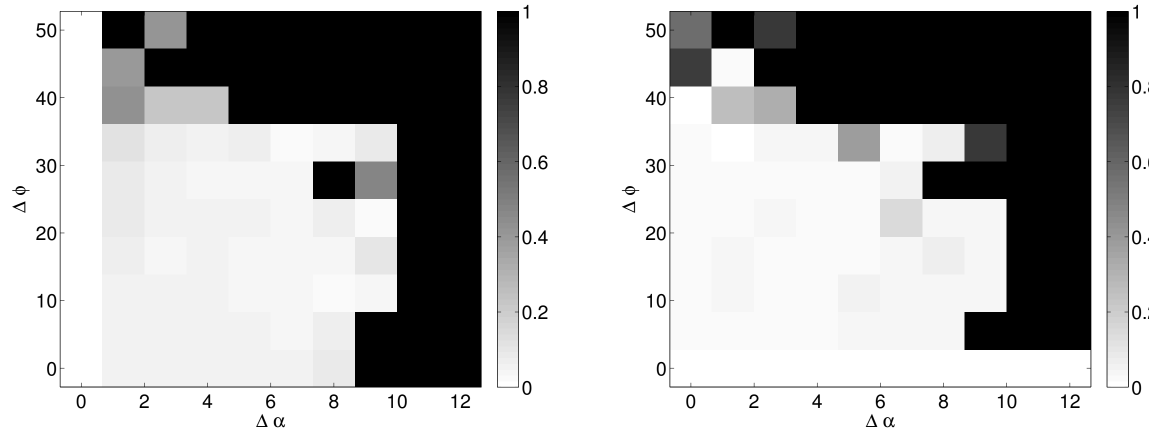

Figure 5.7: Clearance Test 1 for the NDI controller. To the left is the angle of attack overshoot and to the right the Euler bank angle overshoot.

hand made tuning of the control parameters.

Test 2 This test was carried out in the same way as Test 1, but the actuators rate limits were decreased to 30 deg/s. The result is shown in Figures 5.8 and 5.9. The lower rate limit affects both controllers and it can be summarised that both controllers are tuned too fast for the lower rate limit and specially the NDI controller. The result of

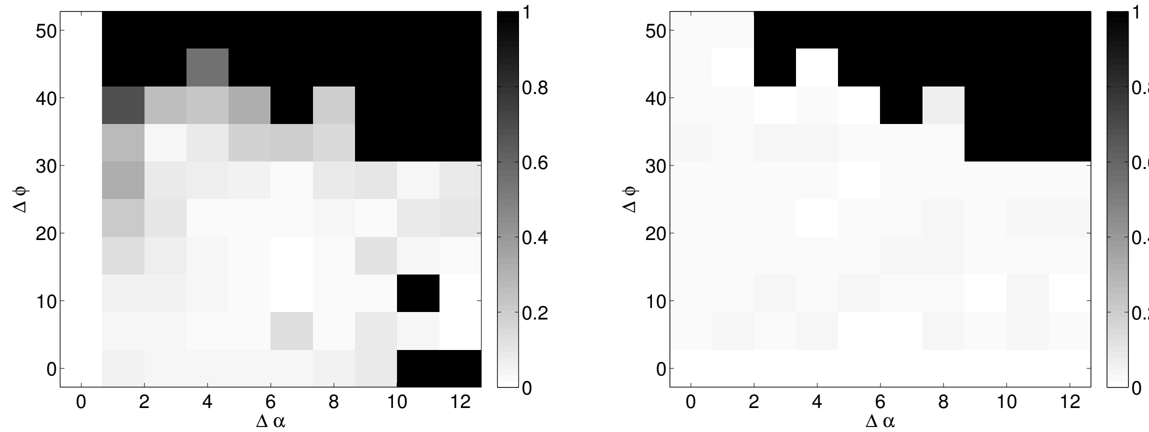

Figure 5.8: Clearance Test 2 for the backstepping controller with an actuator rate limit of 30 deg/s. To the left is the angle of attack overshoot and to the right the Euler bank angle overshoot.

the lower rate limit is that the controller commands an actuator position that cannot be reached fast enough, with the effect of a gain reduction and a time delay as explained in the section about PIO (see section 5.2.2).

In Figure 5.10 the worst-case is shown for the NDI controller with fixed steps $\Delta\phi =$

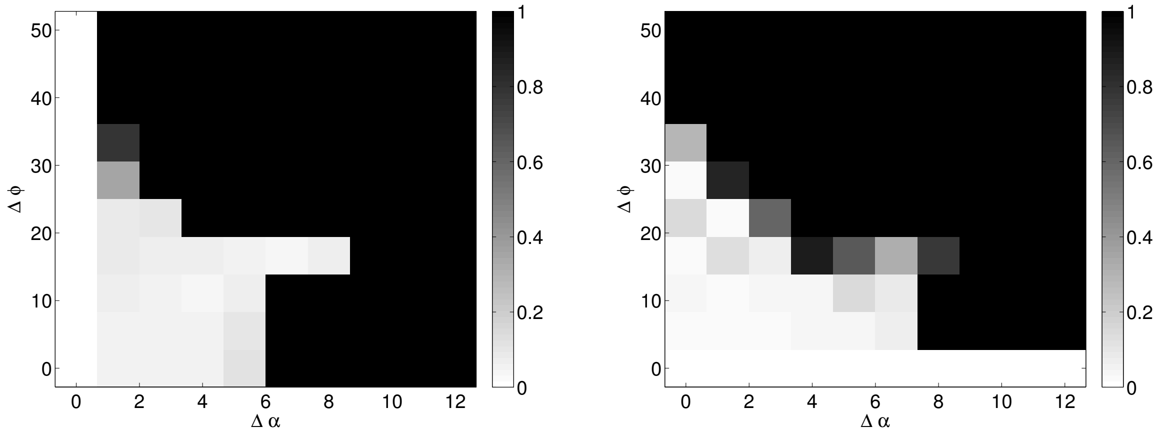

Figure 5.9: Clearance Test 2 for the NDI controller with an actuator rate limit of 30 deg/s. To the left is the angle of attack overshoot and to the right the Euler bank angle overshoot.

22 deg and $\Delta\alpha = 3$ deg in Parameterisation 2. The saturated actuators can be seen in Figure 5.11. Note how the controller manage to bring the Alpha back from the

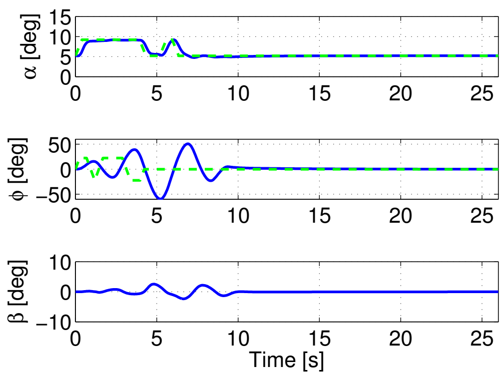

Figure 5.10: This figure shows the controlled variables in the worst-case of the NDI controller when $\Delta\alpha = 3$ deg and $\Delta\phi = 22$ deg in Parameterisation 1.

oscillation after 10 seconds.

Test 3 In this test the center of mass was moved 5% aft, which corresponds to around 25 cm), but otherwise the same setup as in Test 1.

In Figure 5.12 the result for the backstepping is controller presented . The area of stable amplitudes has shrunk and the overshoots in angle of attack are on the average larger.

Test 4 This simulation shows the control outcome at the higher altitude of 4 km, a Mach number of 0.24 and a flight path angle of -10.6 deg (see Figure 5.13). The test

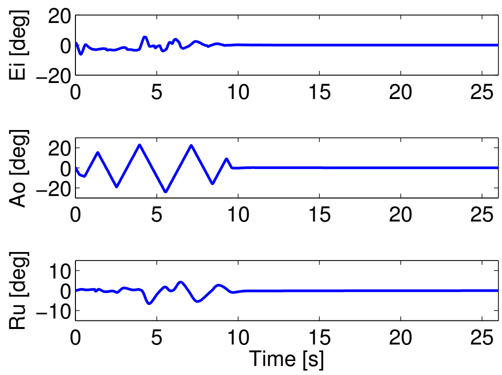

Figure 5.11: This figure shows the actuators in worst-case of the NDI controller when $\Delta\alpha = 3$ deg and $\Delta\phi = 22$ deg in Parameterisation 1.

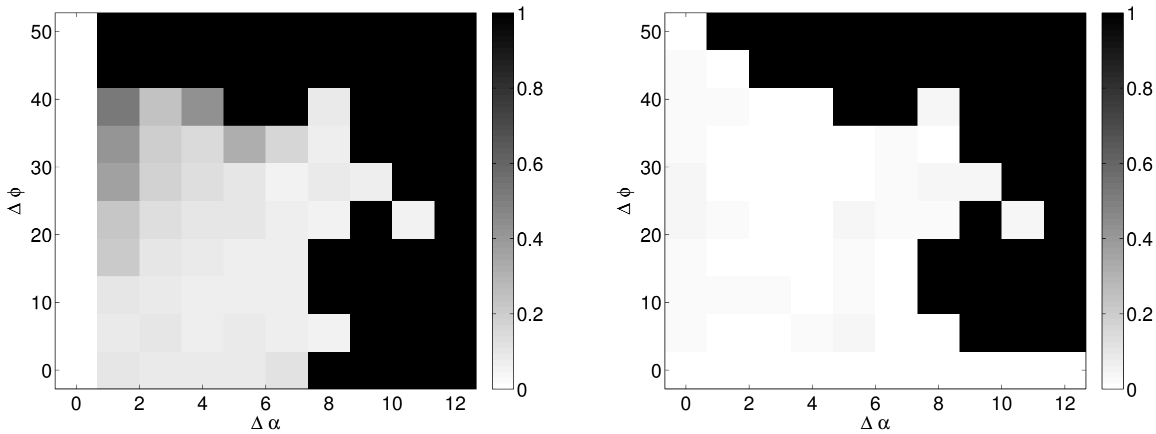

Figure 5.12: Clearance Test 3 for the backstepping controller with the center of mass moved 5% aft. To the left is the angle of attack overshoot and to the right the Euler bank angle overshoot.

was carried out with Parameterisation 2.

The overshoots are still small as on the lower altitude in Test 1, but the higher altitude affects the Alpha in the longitudinal axis. This could be due to the fact that the Alpha needs a larger angle of attack of 15 deg to keep the trim at the higher altitude. This compared to only 5 deg at the altitude 2 km and each increase in $\Delta\alpha$ step size is tougher for to control at the higher angle of attack.

Test 5 This simulation shows the improved performance when the rate limit is doubled to 100 deg/s at an altitude of 4 km, see Figure 5.14. As in Test 4, the test was carried out with Parameterisation 2. The white areas are larger compared to when the rate limit was set to 50 deg/s and also the overshoots are smaller. This confirms that the actuator

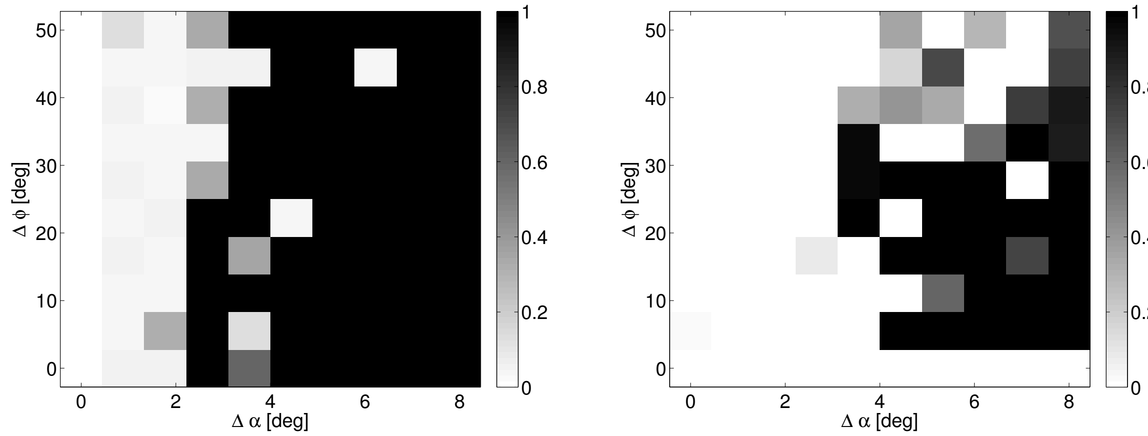

Figure 5.13: Clearance Test 4 for the backstepping controllers at an altitude of 4000. To the left is the angle of attack overshoot and to the right the Euler bank angle overshoot.

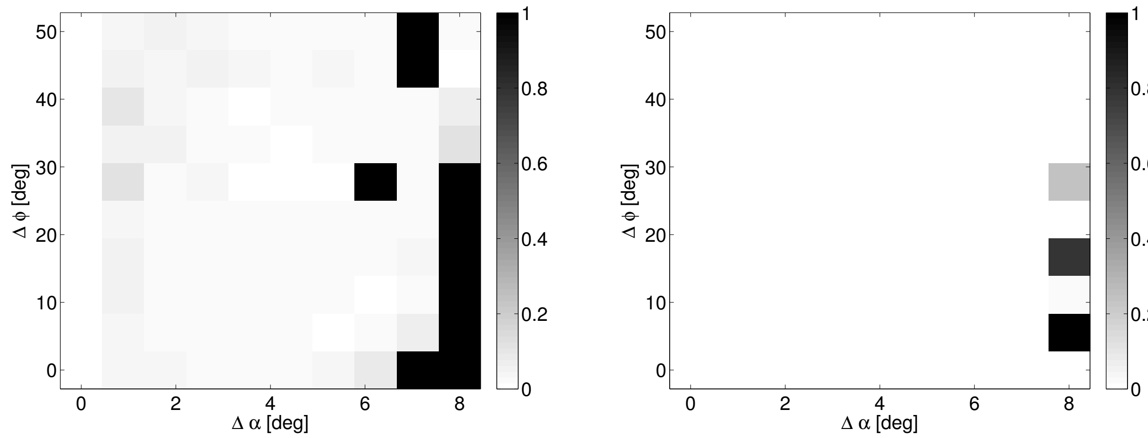

Figure 5.14: Clearance Test 5 for the backstepping controllers at an altitude of 4000 and a actuator rate limit of 100 deg/s. To the left is the angle of attack overshoot and to the right the Euler bank angle overshoot.

rate limit is important to consider in the control design process.

Test 6 and 7 In Test 6 and 7 the optimiser was allowed to try the more general manoeuvres in Parameterisation 3 at the altitudes 2 km and 4 km. In Table 5.2 there is a summary of the test results. The column max α and max ϕ in Table 5.2 means the maximum α and ϕ measured on the whole step sequence of the worst-case. Test 6.1 to 6.7 shows that the more general parameterisation does not result in combinations that gives a departure within the "cleared" (the non black areas with $\Delta\alpha \in [0\ 8]$ and $\Delta\phi \in [-38\ 38]$) areas in figures 5.6 and 5.7. This indicates that easiest way to parameterise a clearance test is to let the optimiser command values only at the minimum

Table 5.2: This table shows a summary of the Tests 6 and 7. The backstepping is denoted B and the NDI controller N. Max α and max ϕ are the maximum deviation from the trim values that the optimiser could find.

Test	Controller	Max α [deg]	Max ϕ [deg]
6.1	B	10.8	41.6
6.2	B	10.5	41.3
6.3	B	8.5	41.6
6.4	B	8.65	42
6.5	N	10.5	40.4
6.6	N	13.6	36.5
6.7	N	8.5	42.4
7.1	B	129.7	254
7.2	B	194	812
7.3	B	4.5	41.2
7.4	N	194.7	183
7.5	N	194.8	199.3

and maximum allowed values. In Test 7.3 it can be seen that the optimiser sometimes get caught in a local maximum.

In Figure 5.15 the worst case in Test 6.1 is shown. The optimiser starts out by commanding a maximum angle of attack and Euler bank angle simultaneously, to next roll back to a minimum roll angle. The rudder is too slow to keep the sideslip and that combined with the high angle of attack of 20 deg gives the large error. The high angle of attack makes the Mach number fall rapidly, which also strains the controller since it is assumed that the Mach number is slowly varying. The controller however, keeps the Alpha stable during the whole flight.

In Figure 5.16 the convergence plot of the simulation is shown. It can be seen that the more general Parameterisation 3 needs around 5000 iterations to converge.

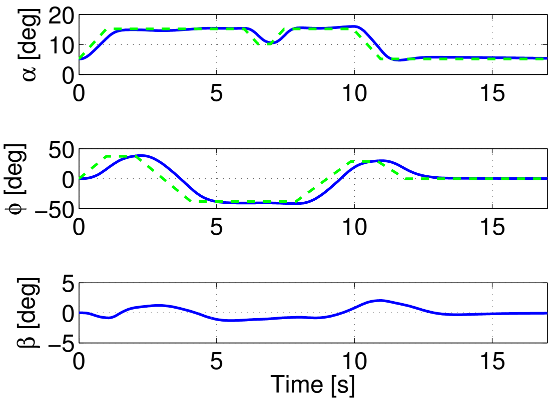

Figure 5.15: The controlled variables found in Test 6.1 in Table 5.2.

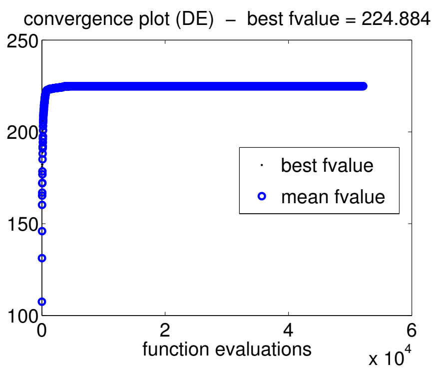

Figure 5.16: The convergence plot of Test 6.1 in Table 5.2. On the y-axis is the cost function and on the x-axis number of function evaluations.

6

Conclusion

In this report we have shown the design procedure of a control law built on a classic design with gain scheduling, as well as two nonlinear designs built on block backstepping. Both the linear and the nonlinear control methods proved to give satisfying results.

The linear control design has two different controllers, one for the lateral and one for the longitudinal system, which increases the design time compared to the nonlinear controllers. The process of gain scheduling the controller is also time consuming, but was fastened by applying a mapping theorem. On the other hand, the block backstepping controller was designed in one piece for the whole envelope, which decreases the development time compared to the linear controller. Backstepping relies on a deeper knowledge of mathematics, which makes the control law much more complex and difficult to troubleshoot than the linear controller.

For a stable airframe as the Alpha, both the linear and the nonlinear control laws gives satisfying results and the controller performance does not motivate the usage of a nonlinear controller. The advantage of using a nonlinear controller is first found when tuning the gains. Retuning the backstepping controller is done fast but redoing the gain scheduling for the linear controller takes a longer time.

The computer based clearance test is a cheap method to investigate the robustness of the control design. To set up a clearance test with a global optimiser is not difficult and the hardest part is to setup a good cost function that will have proper scaling and constraints. By using a global optimiser the computer power is used much more efficiently than in older methods.

The clearance test showed that the nonlinear control designs were very sensitive to rate saturation of the actuators, which is therefore important to consider in the control design.

Future Research

- Apply a method to handle rate saturations.

- Construct a clearance test that includes changes of aerodynamic parameters.
- Continue with the control law design to include landing.
- Construct a map between a pilot-input variable as C^* and the controlled angle of attack in backstepping.

Bibliography

[1] E. S. Agency, Fast20xx (future high-altitude high-speed transport 20xx), retrieved 2011-01-31 http://www.esa.int/SPECIALS/Space_Engineering/SEMV0FXRA0G_0.html (2009).

[2] V. Galactic, Virgin galactic overview, retrieved 2011-01-31 http://www.virgingalactic.com/overview/ (2011).

[3] F. Berefelt, Cofopt - a worst case maneuver search package, version 2.2, FOI (2010).

[4] W. F. Phillips, Mechanics of Flight, John Wiley & Sons, Inc., Hoboken, New Jersey, 2010.

[5] B. L. Stevens, F. L. Lewis, Aircraft control and simulation, John Wiley & Sons, Inc., Hoboken, New Jersey, 2003.

[6] J. V. Daigoro Ito, Jennifer Georgie, D. T. Ward, Reentry vehicle flight controls design guidelines: Dynamic inversion, flight Simulation Laboratory, Texas Engineering Experiment Station, Texas A and M University (2002).

[7] Military Specification, Flying qualities of piloted airplanes (1980).

[8] L. G. M. Harold N. Tobie, Elden M. Elliott, A new longitudinal handling qualities criterion, presented at the 18th Ann. Nat. Aerospace Electron. Conf., May 16-18, 1966.

[9] J. Ackermann, Robust Control, Systems with Uncertain Physical Parameters, Springer-Verlag, London, 1993.

[10] W. Leithead, Survey of gain-scheduling analysis & design, International Journal of Control 73 (1999) 1001–1025.

[11] H. K. Khalil, Nonlinear Systems, third edition Edition, Prentice Hall, Inc., 2002.

[12] M. L. Steinberg, A. B. Page, Nonlinear adaptive flight control with genetic algorithm design optimization, International Journal of Robust and Nonlinear Control 9 (14) (1999) 1097–1115.

[13] O. Härkegård, Flight control design using backstepping, Licentiate thesis no. 875, Department of Electrical Engineering, Linköping University, SE-581 83 Linköping, Sweden (Mar. 2001).

[14] J. Thumberg, J. W. Robinson, Block backstepping, NDI and related cascade designs for efficient development of nonlinear flight control laws, Proc. AIAA Guidance, Navigation and Control Conference 08, Honolulu, HI, Aug. 18-21, 2008, AIAA paper 2008-6960.

[15] P. P. K. Corey J. Schumacher, N. H. McClamroch, Stabiliy analysis of dynamic inversion controller using time-scale separation, aIAA-1998-4322, AIAA Guidance, Navigation, and Control Conference and Exhibit, Boston, MA, Aug. 10-12, 1998, Collection of Technical Papers. Pt. 2 (A98-37001 10-63).

[16] S. B. A. M. P.P Menon, I. Postlethwaite, D. Bates, Multiobjective worst-case analysis of a re-entry vechicle control law, 17th IFAC World Congress (IFAC08), Seoul, South Korea, July 6-11, 2008.

[17] R. Storn, K. Price, Differential evolution- a simple and efficient adaptive scheme for global optimization over continuous spaces, Technical Report TR-95-012, ICSI, March 1995, ftp.icsi.berkeley.edu.

[18] D. B. P.P Menon, I. Postlethwaite, Computation of worst-case pilot inputs for nonlinear flight control system analysis, Journal of Guidane, Control, and Dynamics 29 (1) (2006) 195–198.

[19] D. H. Klyde, D. G. Mitchell, Investigating the role of rate limiting in pilot-induced oscillations, Journal of guidance, control, and dynamics 27 (5) (2004) 804–813.