Control of Unmanned Aerial Vehicles using

Non-linear Dynamic Inversion

Master's Thesis
Division of Automatic Control
Department of Electrical Engineering
Linköping University

Mia Karlsson

LiTH-ISY-EX-3284-2002

Linköping 2002

Control of Unmanned Aerial Vehicles using

Non-linear Dynamic Inversion

Master's Thesis
Division of Automatic Control
Department of Electrical Engineering
Linköping University

Mia Karlsson

LiTH-ISY-EX-3284-2002

Supervisors:	M.Sc. Jonas Lövgren, Saab AB.
	Lic. Ola Härkegård, ISY, Linköping University.
Examiner:	Prof. Torkel Glad, ISY, Linköping University.

Linköping, December 20, 2002

Avdelning, Institution
Division, Department

Institutionen för Systemteknik
581 83 LINKÖPING

Datum
Date

2002-12-20

Språk
Language

Svenska/Swedish
X Engelska/English

Rapporttyp
Report category

Licentiatavhandling
X Examensarbete
C-uppsats
D-uppsats
Övrig rapport

ISBN

ISRN LITH-ISY-EX-3284-2002

Serietitel och serienummer
Title of series, numbering

ISSN

URL för elektronisk version
http://www.ep.liu.se/exjobb/isy/2002/3284/

Titel Title	Design av styrlagar för obemannade farkoster med hjälp av exakt linjärisering Control of Unmanned Aerial Vehicles using Non-linear Dynamic Inversion
Författare Author	Mia Karlsson

Sammanfattning
Abstract

This master's thesis deals with the control design method called Non-linear Dynamic Inversion (NDI) and how it can be applied to Unmanned Aerial Vehicles (UAVs). In this thesis, simulations are conducted using a model for the unmanned aerial vehicle SHARC (Swedish Highly Advanced Research Configuration), which Saab AB is developing.

The idea with NDI is to cancel the non-linear dynamics and then the system can be controlled as a linear system. This design method needs much information about the system, or the output will not be as desired. Since it is impossible to know the exact mathematical model of a system, some kind of robust control theory is needed. In this thesis integral action is used.

A problem with NDI is that the mathematical model of a system is often very complex, which means that the controller also will be complex. Therefore, a controller that uses pure NDI is only discussed, and the simulations are instead based on approximations that use a cascaded NDI. Two such methods are investigated. One that uses much information from aerodata tables, and one that uses the derivatives of some measured outputs. Both methods generate satisfying results. The outputs from the second method are more oscillatory but the method is found to be more robust. If the signals are noisy, indications are that method one will be better.

Nyckelord
Keyword

Non-linear Dynamic Inversion, exact linearization, cascaded structure, integral action, flight control, Unmanned Aerial Vehicles

Abstract

This master's thesis deals with the control design method called Non-linear Dynamic Inversion (NDI) and how it can be applied to Unmanned Aerial Vehicles (UAVs). In this thesis, simulations are conducted using a model for the unmanned aerial vehicle SHARC (Swedish Highly Advanced Research Configuration), which Saab AB is developing.

The idea with NDI is to cancel the non-linear dynamics and then the system can be controlled as a linear system. This design method needs much information about the system, or the output will not be as desired. Since it is impossible to know the exact mathematical model of a system, some kind of robust control theory is needed. In this thesis integral action is used.

A problem with NDI is that the mathematical model of a system is often very complex, which means that the controller also will be complex. Therefore, a controller that uses pure NDI is only discussed, and the simulations are instead based on approximations that use a cascaded NDI. Two such methods are investigated. One that uses much information from aerodata tables, and one that uses the derivatives of some measured outputs. Both methods generate satisfying results. The outputs from the second method are more oscillatory but the method is found to be more robust. If the signals are noisy, indications are that method one will be better.

Acknowledgments

This thesis finalizes my Master of Science in Applied Physics and Electrical Engineering at Linköping university.

I have written this master's thesis at Saab Aerospace, department Gripen Handling Qualities, from August to December 2002. It has been a wonderful time, thanks to the people working there. I would especially like to thank my supervisor Jonas Lövgren and my vice-supervisor Karin Ståhl-Gunnarsson for being helpful and always finding time for my questions.

Another person that I would like to thank is my supervisor at Linköping university, Ola Härkegård, with whom I have had many interesting discussions.

I am also very grateful to all of you who have read the drafts of this thesis.

I would also like to thank my family and friends, for their support and for giving me such a good time during my studies.

Finally I would like to thank Markus, making my life happy, and to whom I am also grateful for making me interested in this very exciting and fascinating world of engineering.

Linköping, December 2002

Mia Karlsson

Notation

Symbol	**Unit**	**Definition**
α	rad	angle of attack
β	rad	sideslip angle
b	m	wing span
$\bar{c}$	m	aerodynamic mean chord
C_l	-	rolling moment coefficient
C_L	-	lifting force coefficient
C_m	-	pitching moment coefficient
C_n	-	yawing moment coefficient
C_N	-	normal force coefficient
C_T	-	tangential force coefficient
C_Y	-	side force coefficient
δ_a	rad	aileron deflection
δ_e	rad	elevator deflection
δ_r	rad	rudder deflection
ϕ	rad	roll angle
F_T	N	engine thrust force
γ	rad	flight path angle
g	m/s^2	acceleration due to gravity
I_x, I_y, I_z, I_{xz}, I_{yz}, I_{xy}	kgm^2	moment of inertia
L	Nm	rolling moment
L_{lift}	N	lift force
m	kg	aircraft mass
M	-	Mach number
M	Nm	pitching moment
n_z	-	load factor
N	N	normal force
N	Nm	yawing moment
p	rad/s	roll rate
θ	rad	pitch angle
q	rad/s	pitch rate
q_a	N/m^2	dynamic pressure
ρ	kg/m^3	air density
r	rad/s	yaw rate
S	m^2	reference area
T	N	tangential force
u	m/s	longitudinal velocity
v	m/s	lateral velocity
V_T	m/s	true airspeed
ω_n	rad/s	natural frequency

w	m/s	normal velocity
(x_b, y_b, z_b)	m	body-axes coordinate system
(x_e, y_e, z_e)	m	earth-axes coordinate system
ψ	rad	yaw angle
Y	N	side force
ζ	-	damping factor

Contents

1 Introduction

This master’s thesis has been carried out at Saab Aerospace, department Gripen Handling Qualities. The task is to investigate the control design method Non-linear Dynamic Inversion (NDI) and apply it to the unmanned aerial vehicle SHARC (Swedish Highly Advanced Research Configuration).

1.1 Background

Non-linear Dynamic Inversion has been applied to flight control design during the two last decades, see [1]. The reason for investigating NDI is that Saab wants an easy control method that is applicable for non-linear systems. When for example Gripen is controlled, LQ-design is used. LQ-design is a method, where different controllers have to be designed for different flight cases, see [2]. The hope is that NDI is an easier method to use and that the same parameters can be used in the entire flight envelope. The intention is that the controller then easily can be adapted to other aircraft.

1.2 Objectives

The purpose is to design control laws for the SHARC in both longitudinal and lateral motion using the design method Non-linear Dynamic Inversion. In order for the SHARC to behave well, even if there are model errors in the design, robust control laws have to be implemented.

1.3 SHARC

Sweden has tested UAV systems since the end of the 1960’s. An example is the French UAV system UGGLAN that is used for tactical reconnaissance within the brigades.

The UAVs can be divided into different groups: High-Altitude Long Endurance (HALE), Medium-Altitude Long Endurance (MALE), Unmanned Reconnaissance Aerial Vehicles (URAV), Unmanned Combat Aerial Vehicles (UCAV), and Tactical Unmanned Aerial Vehicles (TUAV).

In Sweden, the work with UAVs began in 1998 within the framework of the National Aeronautical Research Programme (NFFP). Nine configurations were defined, and one of them was SHARC. At the same time a working team called IPT UAV was formed. It will suggest the direction for Swedish military activity on UAVs during 2001 to 2020.

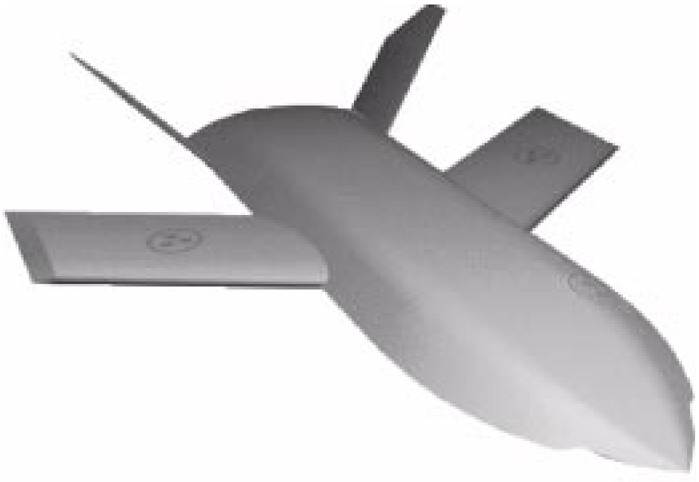

Figure 1.1: The SHARC.

The SHARC is an UCAV that has a low signature feature, a range of 1000 km and it will have high surviveability and a low cost in serial production. It will also be autonomous. It is 10 meters long, has a wingspan of 8 meters, and a take-off-weight of about 5000 kg.

1.4 Limitations

The controllers calculated in this master's thesis will not be tested on a real aircraft, only on a model of the SHARC. On the other hand, model errors will be implemented to simulate the reality. Another limitation is that the model not includes noise. The limits and the dynamics for the control surfaces of SHARC, are not definitive defined. The dynamics of the control surfaces are simulated by servos, that are approximations of the system.

1.5 Outline

In Chapter 2, the general theory of flight mechanics is presented together with SHARC-specific theory, such as dependencies of aerodynamic coefficients, control surfaces, and servos.

Chapter 3 discusses the differences between pure NDI and cascaded NDI.

In Chapter 4 the theory is applied to the SHARC. A controller that uses pure NDI is discussed and two controllers that use cascaded NDI are designed. Chapter 4 also deals with controllers that make the system more robust, and descriptions for the closed loop system are derived.

In Chapter 5 simulations are conducted and parameters are decided in order for the system to behave according the requirements.

The conclusions are finally presented in Chapter 6.

2 Flight Mechanics

In this chapter, flight mechanics needed in order to design controllers for the SHARC is presented.

2.1 General Theory

The motion of an aircraft can be divided into rotational and translational motion. The rotational motion is the rotation around the body axes (x_b, y_b, z_b) and the motion can be expressed in angular velocities (p, q, r) and moments (L, M, N), see Figures 2.1 and 2.2.

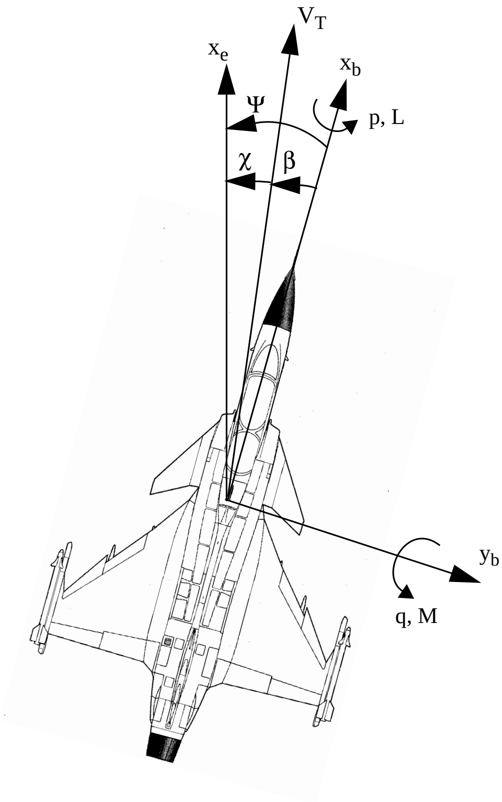

Figure 2.1: Definitions of moments, forces etc.

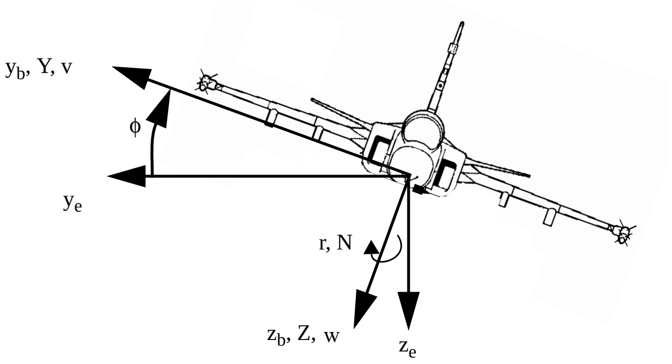

Figure 2.2: Definitions of moments, forces, velocities etc.

Translational motion is movement in the directions of the body axes (x_b, y_b, z_b) and the motion can be described by the velocities (u, v, w) and the forces (X, Y, Z), see Figures 2.1, 2.2, and 2.3. The coordinate system relative earth, (x_e, y_e, z_e) is also seen. In this thesis the forces X and Z are not used, bust instead the tangential force

$$T = -X$$

and the normal force

$$N = -Z$$

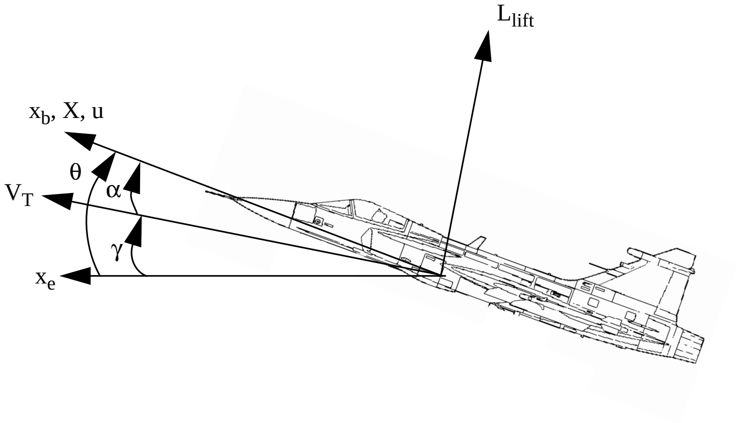

Figure 2.3: Definitions of forces, coordinates, attitudes and velocities.

In Figures 2.1 and 2.3 the true airspeed V_T is seen. The definition of true airspeed is:

$$V_T = |\bar{v} - \bar{v}_w|$$

where $\bar{v}$ is the velocity of centre of gravity relative earth and $\bar{v}_w$ is the velocity of wind relative earth, both expressed in (x_b, y_b, z_b).

Another definition needed is the Mach number defined as:

$$M = \frac{V_T}{a}$$

where a is the speed of the sound.

There are some angles that describe the attitude of the aircraft, for example the euler angles (ϕ, θ, ψ), see Figures 2.1, 2.2, and 2.3, the angle of attack α, see Figure 2.3, and the sideslip angle β, see Figure 2.1. The definitions for α and β are:

$$\alpha = arc\tan\left(\frac{w}{u}\right)$$

$$\beta = arc\sin\left(\frac{v}{V_T}\right)$$

In Figure 2.3, the lift force L_{lift} is seen and is defined as:

$$L_{lift} = q_a S C_L = q_a S(-C_T\sin(\alpha) + C_N\cos(\alpha)) \tag{2.1}$$

The lift force L_{lift} is written like this to get a non-dimensional expression, where C_T and C_N are aerodynamic coefficients corresponding to the tangential and normal forces, and q_a is the dynamic pressure defined as:

$$q_a = \frac{1}{2}\rho V_T^2$$

Using the lift force L_{lift}, the load factor n_z can be defined as:

$$n_z = \frac{L_{lift}}{mg} \tag{2.2}$$

This is an approximation that is good enough for small α. Normally n_z is defined as

$$n_z = \frac{N}{mg}$$

The motion of an aircraft can also be separated into longitudinal and lateral motion. Longitudinal motion considers the forces *N* and *T* and the moment *M*. The equations used in this thesis to describe longitudinal motion are $\dot{\alpha}$ and $\dot{q}$, see (2.3) and (2.10). Lateral motion considers the force *Y* and the moments *L* and *N*. The equations used to describe the lateral motion in this thesis are $\dot{\beta}$, $\dot{r}$, and $\dot{p}$, see (2.4), (2.11), and (2.9).

In order to be able to control the SHARC in both longitudinal and lateral mode, equations for $\dot{\alpha}$, $\dot{q}$, $\dot{\beta}$, $\dot{r}$, and $\dot{p}$ are needed. An expression for $\dot{\alpha}$ is found in [3]:

$$\dot{\alpha} = q - (p\cos\alpha + r\sin\alpha)\tan\beta + \frac{1}{mV_T\cos\beta}(-L_{lift} - F_T\sin\alpha + mg(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta)) \tag{2.3}$$

where F_T is the engine thrust force.

An expression for $\dot{\beta}$ is found in [3]:

$$\dot{\beta} = p\sin(\alpha) - r\cos(\alpha) + \frac{1}{mV_T}(Y - F_T\cos(\alpha)\sin(\beta) + mg_3) \tag{2.4}$$

where

$$g_3 = g(\cos(\beta)\cos(\theta)\sin(\phi) + \sin(\beta)\cos(\alpha)\sin(\theta) - \sin(\alpha)\sin(\beta)\cos(\theta)\cos(\phi)) \quad (2.5)$$

and Y is the side force where

$$Y = q_a S C_Y$$

The equations for $\dot{q}$, $\dot{p}$,and $\dot{r}$ are flat-earth approximations that do not include rotation of the earth and its curvature. The equations are found in [4]:

$$L = I_x\dot{p} - I_{zx}(\dot{r} + pq) - (I_y - I_z)qr \quad (2.6)$$

$$M = I_y\dot{q} - I_{zx}(r^2 - p^2) - (I_z - I_x)rp \quad (2.7)$$

$$N = I_z\dot{r} - I_{zx}(\dot{p} - qr) - (I_x - I_y)pq \quad (2.8)$$

Solving (2.6), (2.7), and (2.8) for $\dot{q}$, $\dot{p}$, and $\dot{r}$ yields:

$$\dot{p} = \frac{L + I_{zx}\left(\frac{N - I_{zx}qr + (I_x - I_y)pq}{I_z} + pq\right) + (I_y - I_z)qr}{\left(1 - \frac{I_{zx}^2}{I_x I_z}\right)I_x} \quad (2.9)$$

$$\dot{q} = \frac{M + I_{zx}(r^2 - p^2) + (I_z - I_x)rp}{I_y} \quad (2.10)$$

$$\dot{r} = \frac{N + I_{zx}\left(\frac{L + I_{zx}pq + (I_y - I_z)qr}{I_x} - qr\right) + (I_x - I_y)pq}{\left(1 - \frac{I_{zx}^2}{I_x I_z}\right)I_z} \quad (2.11)$$

The moments can also be expressed in non-dimensional aerodynamic coefficients:

$$L = q_a S b C_l \quad (2.12)$$

$$M = q_a S \bar{c} C_m \quad (2.13)$$

$$N = q_a S b C_n \quad (2.14)$$

2.2 Flight Mechanics Specific for SHARC

In this section, aerodynamic coefficients, control surfaces and servos specific for the SHARC is presented.

2.2.1 Forces and Aerodynamic Coefficients

The aerodynamic coefficients can be found in tables calculated using results from wind tunnel tests. For the SHARC, the dependencies of the aerodynamic coefficients are presented below.

$$C_m = C_m(M, \alpha, \beta) + C_m(\dot{\alpha}) + C_m(q) + C_m(\delta_e) \tag{2.15}$$

where

$$C_m(\delta_e) \approx C_{m0} + \frac{dC_m}{d\delta_e}\delta_e \tag{2.16}$$

The expression in (2.16) is necessary later in this thesis, in order to be able to solve for the control surface δ_e. It is expressed by a Taylor expansion, where only the first derivative is included to simplify the calculations. C_{m0} is calculated as:

$$C_{m0} = C_m(\delta_e) - \frac{dC_m}{d\delta_e}\delta_e$$

The derivative of the aerodynamic coefficient is approximated by:

$$\frac{dC_m}{d\delta_e} \approx \frac{C_m(\delta_e + \Delta) - C_m(\delta_e - \Delta)}{2\Delta} \tag{2.17}$$

where δ_e is taken from the previous sample. The value of Δ in (2.17) is set to some convenient number to prevent the derivatives from becoming discontinuous. A good value for Δ is 3, that is chosen after some tests have been performed. It would be preferred if the aerodata tables already had contained information about the derivatives.

Other needed aerodynamic coefficients are:

$$C_l = C_l(M, \alpha, \beta) + C_l(p) + C_l(r) + C_l(\delta) \tag{2.18}$$

$$C_n = C_n(M, \alpha, \beta) + C_n(r) + C_n(\delta) \tag{2.19}$$

$$C_N = C_N(M, \alpha, \beta) + C_N(\dot{\alpha}) + C_N(\delta_e) \tag{2.20}$$

$$C_T = C_T(M, \alpha, \beta) + C_T(\delta_a) + C_T(\delta_r) + C_T(\delta_e) \tag{2.21}$$

Some of these expressions are approximated using Taylor expansion in the same way as for C_m. The new expressions are:

$$C_l = C_l(M, \alpha, \beta) + C_l(p) + C_l(r) + \frac{d}{d\delta_a}C_l(\delta_a)\delta_a + \frac{d}{d\delta_r}C_l(\delta_r)\delta_r + C_{l0} \tag{2.22}$$

$$C_n = C_n(M, \alpha, \beta) + C_n(r) + \frac{d}{d\delta_a}C_n(\delta_a)\delta_a + \frac{d}{d\delta_r}C_n(\delta_r)\delta_r + C_{n0} \tag{2.23}$$

$$C_N = C_N(M, \alpha, \beta) + C_N(\dot{\alpha}) + C_{N0} + \frac{dC_N}{d\delta_e}\delta_e \tag{2.24}$$

Using the aerodata tables, one has to be certain that the inputs to the tables are part of the tables. Otherwise the outputs from the tables are set to zero. A solution to the problem is to limit those signals, such as the commanded control surface deflections.

2.3 Control Surfaces

The control surfaces that make it possible to control the SHARC are seen in Figure 2.4. They are the elevator δ_e, the aileron δ_a, and the rudder δ_r.

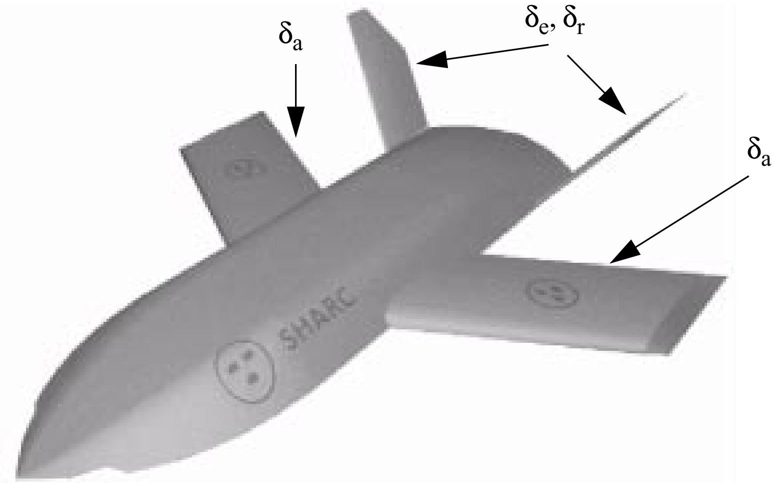

Figure 2.4: The control surfaces for the SHARC.

The definitions of the control surfaces are

$$\delta_e = \frac{\delta_e^{left} + \delta_e^{right}}{2}$$

$$\delta_a = \frac{\delta_a^{left} - \delta_a^{right}}{2}$$

$$\delta_r = \frac{\delta_r^{left} - \delta_r^{right}}{2}$$

One has to decide which variables of the SHARC that should be controlled. Usually, for ordinary aircraft, α is controlled at low speed, and n_z at high speed in longitudinal mode. This is because pilots feel most comfortable if the variables are controlled in that way. Even for UAVs, it is natural to control these variables. The velocity when the angle of attack is no longer controlled, but instead the load factor n_z, is called Corner speed. At the altitude 1000 meters, this occurs at about Mach 0.44. To decide Corner speed, the commanded α is set to its maximum, 15 degrees, and the Mach number is increased until n_z reaches its maximum value, which for the SHARC is six.

In lateral mode the angular velocity p and the sideslip angle β are to be controlled. The aim is to keep β to zero, except at landing if the wind is coming from the side. In this thesis, the SHARC will roll around the body axis x_b. Other alternatives would be to roll around the velocity vector V_T, or around a vector that points in a direction where missiles are fired.

One must also decide which control surfaces that are supposed to control pitch, yaw and roll. One way is to control the moments and decide the angles of the control surfaces by solving a linearly constrained quadratic programming problem, see [5] and [6]. One advantage with this approach is that if one control surface is damaged, then the other surfaces can be optimized to control the aircraft. The calculations are part of a so called control selector, see [7], that is needed if the control surfaces are redundant. This is however not the case in this thesis. The three angular accelerations that are of interest, are all controlled by three surfaces. The relations are found in tables that origin from wind tunnel tests.

$$\begin{aligned}\dot{p} &= f_p(\delta_a, \delta_e, \delta_r)\\ \dot{q} &= f_q(\delta_a, \delta_e, \delta_r)\\ \dot{r} &= f_r(\delta_a, \delta_e, \delta_r)\end{aligned}$$

There are three equations and three control surfaces, which means that all surfaces can be solved for. To simplify, let δ_e control pitch and δ_r and δ_a together control yaw and roll. Even if δ_e affects roll and yaw, as well as δ_r and δ_a affect pitch, the contribution is pretty small and can be approximated by zero.

2.3.1 Servos

In the model of the SHARC, servos for the control surfaces had to be added, see Figure 2.5. The same model of a servo was added for all control surfaces. The model is simple but is a good approximation of general servo dynamics.

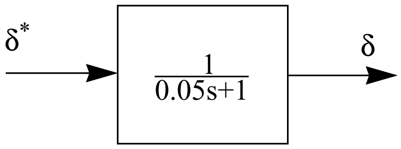

Figure 2.5: The servo used for all control surfaces.

3 Non-linear Dynamic Inversion

In this thesis, two different ways to control the SHARC are investigated, pure Non-linear Dynamic Inversion and cascaded Non-linear Dynamic Inversion. The word “pure” is used to make a distinction between NDI and the approximated variant, which uses a cascaded NDI.

3.1 Pure Non-linear Dynamic Inversion

Non-linear Dynamic Inversion (NDI), also known as exact linearization, see [2], is a structured way to cancel the dynamics and then control the system as a linear system. The states are linearized by letting:

$$\begin{aligned} z_1 &= y \\ z_2 &= \dot{y} \\ &\dots \\ z_n &= y^{(n-1)} \end{aligned}$$

$$\begin{aligned} \dot{z}_1 &= z_2 \\ \dot{z}_2 &= z_3 \\ &\dots \\ \dot{z}_n &= \psi(z, u) \end{aligned}$$

Then let

$$\dot{z}_n = l_0 r - Lz = \psi(z, u)$$

where matrix L is choosen to get proper poles, and solve for u. If the n^{th} state derivative includes a control signal, the system can be exactly linearized. If the control signal occurs before the last state derivative, the entire closed loop system is not linearized and problems will occur if some states are unstable.

3.2 Cascaded Non-linear Dynamic Inversion

The other method used is a cascaded NDI, also known as “NDI using time scale separation”. The system is divided into an inner and an outer loop. The dynamics in the inner loop has to be faster than the dynamics in the outer loop. For example, the output from the inner loop in longitudinal mode is the angular velocity *q,* which is faster than the angle of attack α, which is output from the outer loop, because $\dot{\alpha} \approx q$. The control surfaces have direct effect on $\dot{p}$, $\dot{q}$, and $\dot{r}$, since the surfaces produce the corresponding moments, so δ_e has more effect on $\dot{q}$ than on $\dot{\alpha}$.

A cascaded NDI also cancels the dynamics of a system, but the cancellation is not exact. It relies on the approximation that the inner loop is so fast, that the time it takes to perform something in the inner loop is approximated by zero. The behavior of the closed loop system is decided by the demands of the derivatives in the inner and outer loops. An approximated description of the closed loop system can also be made and consequently the approximated poles can be calculated. The cascaded control structure for the pitch mode is demonstrated in Figure 3.1, where η and ξ are some arbitrary functions.

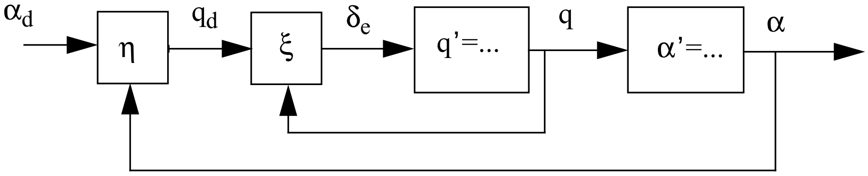

Figure 3.1: The cascaded control structure for the pitch mode.

3.3 An Example, Pure NDI versus Cascaded NDI

To see how these two methods differ, a simple example will be presented by studying the following system:

$$\dot{x}_1 = -f(x_1) + x_2 \tag{3.1}$$

$$\dot{x}_2 = u \tag{3.2}$$

$$y = x_1 \tag{3.3}$$

where *f* is some arbitrary function. Start with pure NDI and define the new states:

$$z_1 = x_1$$

$$z_2 = \dot{x}_1 = -f(x_1) + x_2$$

The derivatives are:

$$\dot{z}_1 = z_2 = -f(x_1) + x_2$$

$$\dot{z}_2 = -\dot{f}(x_1)\dot{x}_1 + \dot{x}_2 = -\dot{f}(x_1)(-f(x_1) + x_2) + u = v \tag{3.4}$$

and

$$y = \iint \dot{z}_2 dt = \iint v dt$$

The signal *v* is designed for the system to get proper poles:

$$v = l_0 r - l_1 z_1 - l_2 z_2 \tag{3.5}$$

The system can then be expressed as:

$$\begin{bmatrix} \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -l_1 & -l_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} 0 \\ l_0 \end{bmatrix} r$$

Set (3.4) and (3.5) equal and solve for *u* expressed in the original states. The control signal *u* will generate the output *y,* with performances defined by the signal *v*.

$$u = l_0 r - a_1 x_1 - a_2(-f(x_1) + x_2) + f'(x_1)(-f(x_1) + x_2) \tag{3.6}$$

Now the second method with the cascaded NDI, will be examined. With the first method the desired closed loop system was designed with the help of matrix *L*. With the second method the desired closed system is designed by the derivatives in (3.7) and (3.8) where x_{1d} and x_{2d} are the desired signals.

$$\dot{x}_1 = k_{x_1}(x_{1d} - x_1) \tag{3.7}$$

$$\dot{x}_2 = k_{x_2}(x_{2d} - x_2) \tag{3.8}$$

Use (3.1) and (3.7), and solve for the x_2 that generates the desired $\dot{x}_1$:

$$x_2 = k_{x_1}(x_{1d} - x_1) + f(x_1) \tag{3.9}$$

Now an approximation is made. The inner loop is supposed to be much faster than the outer loop, which means that x_2 can be approximated with x_{2d}. Equation (3.9) can therefore be written as:

$$x_{2d} = k_{x_1}(x_{1d} - x_1) + f(x_1) \tag{3.10}$$

Use (3.2) and (3.8), and solve for the *u* that generates the wanted $\dot{x}_2$:

$$u = k_{x_2}(x_{2d} - x_2) \tag{3.11}$$

Insert (3.10) into (3.11) and solve for *u*:

$$u = k_{x_2}(k_{x_1}(x_{1d} - x_1) + f(x_1) - x_2) \tag{3.12}$$

The control signals for both methods now can be compared. Equation (3.6) contains $\dot{f}(x_1)$, which is not included in (3.12). The control signal calculated using pure NDI is more complex, but the advantage is that the system is exactly linearized in contrast to the cascaded NDI where approximations are made.

4 Non-linear Dynamic Inversion Applied to SHARC

In this chapter some different control methods using Non-linear Dynamic Inversion (NDI), are applied to the SHARC. First, pure NDI is evaluated. Then two controllers that use a cascaded NDI are calculated. The cascaded NDI method uses approximations that will help simplify the expressions. For example, the inner loops are assumed to be much faster than the outer loops, and the dynamics for the control surfaces are not included, because the surfaces are assumed to move instantly.

4.1 Pure Non-linear Dynamic Inversion

As mentioned before, NDI is an exact linearization and the equations are complex, which soon will be noticed.

Start with the pitch mode, where the angle of attack α is to be controlled. The servo for the control surface angle δ_e^*, see Figure 2.5, can be expressed by:

$$\dot{\delta}_e = \frac{\delta_e^* - \delta_e}{0.05} \tag{4.1}$$

Create the following states:

$$\begin{aligned} z_1 &= \alpha \\ z_2 &= \dot{\alpha} \\ z_3 &= \ddot{\alpha} \end{aligned}$$

where $\dot{\alpha}$ is found in (2.3). The derivatives of the states are:

$$\dot{z}_1 = \dot{\alpha} = z_2$$

$$\begin{aligned}
\dot{z}_2 = z_3 = \ddot{\alpha} = \dot{q} - \Bigg(& (\dot{p}\cos(\alpha) - p\sin(\alpha)\dot{\alpha} + \dot{r}\sin(\alpha) + r\cos(\alpha)\dot{\alpha})\tan(\beta) + \\
& (p\cos(\alpha) + r\sin(\alpha))\frac{\dot{\beta}}{\cos(\beta)^2}\Bigg) - \frac{m(\dot{v}_T\cos(\beta) - v_T\sin(\beta)\dot{\beta})}{m^2 v_T^2\cos(\beta)^2}(-L - F_T\sin(\alpha) + \\
& mg(\cos(\alpha)\cos(\theta)\cos(\phi) + \sin(\alpha)\sin(\theta))) + \\
& \frac{1}{mv_T\cos(\beta)}(-\dot{L} - \dot{F}_T\sin(\alpha) - F_T\cos(\alpha)\dot{\alpha} + mg(-\sin(\alpha)\dot{\alpha}\cos(\theta)\cos(\phi) + \\
& \cos(\alpha)(-\sin(\theta)\dot{\theta}\cos(\phi) - \cos(\theta)\sin(\phi)\dot{\phi}) + \cos(\alpha)\dot{\alpha}\sin(\theta) + \sin(\alpha)\cos(\theta)\dot{\theta}))
\end{aligned} \tag{4.2}$$

$$\dot{z}_3 = \dddot{\alpha} = \dots$$

In the equation for $\dot{z}_2$ there are some derivatives needed, for example $\dot{q}$ that includes the moment M, see (2.10). The moment M is expressed by the elevator δ_e, but the parameter of interest is δ_e^*, see Figure 2.5. To be able to express δ_e^*, the state derivative $\dot{z}_3$ is also needed together with (4.1). The expression for $\dot{z}_3$ is huge and there are also many derivatives that must be realized, for example the derivatives of the aerodynamic coefficients and $\dot{F}_T$. To avoid this huge expression, it may be better to eliminate the derivatives of those variables that vary slowly, for example V_T and F_T. Next step is to make a closed loop system, let

$$\dot{z}_3 = v \tag{4.3}$$

Design v by a state feedback, where L is chosen to get proper poles:

$$v = l_0\alpha_d - L\bar{z} \tag{4.4}$$

Set (4.3) and (4.4) equal and solve for δ_e^*.

Continue with the lateral mode. Because of the servos for the control surfaces δ_a and δ_r, three states are needed. The equations for the servos are:

$$\dot{\delta}_a = \frac{\delta_a^* - \delta_a}{0.05} \tag{4.5}$$

$$\dot{\delta}_r = \frac{\delta_r^* - \delta_r}{0.05} \tag{4.6}$$

The decision was to control the sideslip angle β, therefore the states are:

$$z_4 = \beta$$
$$z_5 = \dot{\beta}$$
$$z_6 = \ddot{\beta}$$

where $\dot{\beta}$ is found in (2.4). The derivatives of the states are:

$$\dot{z}_4 = \dot{\beta} = z_5$$

$$\dot{z}_5 = \ddot{\beta} = z_6 = \dot{p}\sin(\alpha) + p\cos(\alpha)\dot{\alpha} - \dot{r}\cos(\alpha) + r\sin(\alpha)\dot{\alpha} + \tag{4.7}$$

$$\frac{1}{mV_T}(q_a S\dot{C}_Y + \dot{q}_a S C_Y - \dot{F}_T\cos(\alpha)\sin(\beta) - F_T(-\sin(\alpha)\dot{\alpha}\cos(\beta) +$$

$$\cos(\alpha)\cos(\beta)\dot{\beta}) + mg(-\sin(\beta)\dot{\beta}\cos(\theta)\sin(\phi) + \cos(\beta)(-\sin(\theta)\dot{\theta}\sin(\phi) +$$

$$\cos(\theta)\cos(\phi)\dot{\phi}) + \cos(\beta)\dot{\beta}\cos(\alpha)\sin(\theta) + \sin(\beta)(-\sin(\alpha)\dot{\alpha}\sin(\theta) +$$

$$\cos(\alpha)\cos(\theta)\dot{\theta}) - (\cos(\alpha)\dot{\alpha}\sin(\beta) + \sin(\alpha)\cos(\beta)\dot{\beta})\cos(\theta)\cos(\phi) +$$

$$\sin(\alpha)\sin(\beta)(-\sin(\theta)\dot{\theta}\cos\phi - \cos(\theta)\sin(\phi)\dot{\phi})))-$$

$$\frac{\dot{v}_T}{mv_T^2}(q_a S C_Y - F_T\cos(\alpha)\sin(\beta) + mg(\cos(\beta)\cos(\theta)\sin(\phi) +$$

$$\sin(\beta)\cos(\alpha)\sin(\theta) - \sin(\alpha)\sin(\beta)\cos(\theta)\cos(\phi)))$$

$$\dot{z}_6 = \ldots$$

The state derivative $\dot{z}_6$ is necessary because δ_a^* and δ_r^* are needed in the expression instead of δ_a and δ_r. Equations (4.5) and (4.6) are used for that purpose. The expression for $\dot{z}_6$ is not derived in this thesis. The expression is huge and contains many derivatives, which must be approximated by filters. Now the closed loop will be shaped. Design v by letting:

$$v = l_0\beta_d - L\bar{z}$$

Let

$$\dot{z}_6 = v$$

and solve for δ_a^*. This expression contains δ_r^*, so more equations are needed. Another variable to be controlled is p, hence create the new states:

$$z_7 = p$$
$$z_8 = \dot{p}$$

The derivatives of the states are:

$$\dot{z}_7 = \dot{p}$$
$$\dot{z}_8 = \ldots$$

The equation for $\dot{p}$ is found in (2.9). The moments L and N in that equation contain the surfaces deflections δ_a and δ_r. To be able to express in terms of δ_a^* and δ_r^*, $\dot{z}_8$ is also needed. Once again this is a huge expression that not will be calculated in this thesis. It also contains derivatives that have to be approximated in some way. Design the signal v as:

$$v = l_0 p_d - L\bar{z}$$

Let

$$\dot{z}_8 = v$$

and solve for δ_a^*. Now there are two equations where δ_a^* is solved for. Let these equations be equal and solve for δ_r^*.

Now the controller for the system is complete, and if no approximations are made, the system behaves exactly as desired. Unfortunately, the expressions are too complex, and contains too many derivatives that are difficult to create. One simplification is to say that the dynamics of the servos are so fast, that the expressions for the servos are not needed. In that case z_3, z_6, and z_8 will not be used and the expressions for the controller will be smaller.

4.2 Cascaded Non-linear Dynamic Inversion, Method One

Now, a method that uses a cascaded NDI is presented, see [8] and [9]. The control signals in both longitudinal and lateral mode will be calculated.

4.2.1 Longitudinal Mode

At low speed the controlled variable is the angle of attack α and at high speed the controlled variable is exchanged for the load factor n_z.The equations used when α is controlled are (2.3) and (2.9) and the equations used for the pitch moment are (2.13), (2.15), and (2.16). The desired $\dot{\alpha}$ and $\dot{q}$ can be expressed as a constant times the difference between the desired state and the measured state, which gives a first order system:

$$\dot{\alpha} = k_\alpha(\alpha_d - \alpha) \tag{4.8}$$

$$\dot{q} = k_q(q_d - q) \tag{4.9}$$

These equations determine how fast the system will respond on desired commands.

Solving for q in (2.3) and exchanging $\dot{\alpha}$ for the desired one in (4.8), gives that with this q, the desired $\dot{\alpha}$ is obtained and also the desired α. Since the inner loop is faster than the outer loop, as described in Chapter 3.3, the approximation $q \approx q_d$ can be made, which yields:

$$q_d = k_\alpha(\alpha_d - \alpha) + (p\cos\alpha + r\sin\alpha)\tan\beta - \Big(\frac{1}{mv_T\cos\beta} - F_T\sin\alpha + \quad (4.10)$$

$$mg(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta) - q_a S(-C_T\sin(\alpha) + C_N\cos(\alpha))\Big)$$

Continue with the inner loop. Use (2.10) and exchange $\dot{q}$ for the desired one in (4.9):

$$k_q(q_d - q) = \frac{q_a S\bar{c}\left(C_m(M, \alpha, \beta) + C_m(\dot{\alpha}) + C_m(q) + C_{m0} + \frac{dC_m}{d\delta_e}\delta_e\right)}{I_y} + \quad (4.11)$$

$$\frac{I_{zx}(r^2 - p^2) + (I_z - I_x)rp}{I_y}$$

If δ_e in (4.11) is solved for, that δ_e will generate the desired $\dot{q}$. Insert (4.10) into (4.11) and solve for δ_e:

$$\delta_e = \left(\frac{d}{d\delta_e}C_m(\delta_e)\right)^{-1} \cdot \Big(\Big(k_q I_y\Big(k_\alpha(\alpha_d - \alpha) + (p\cos\alpha + r\sin\alpha)\tan(\beta) - \quad (4.12)$$

$$\frac{1}{mV_T\cos\beta}(-F_T\sin\alpha + mg(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta) - q_a S \cdot$$

$$(-C_T\sin(\alpha) + C_N\cos(\alpha))) - q\Big) - I_{zx}(r^2 - p^2) - (I_z - I_x)rp\Big) \cdot$$

$$\frac{1}{q_a S\bar{c}} - C_m(M, \alpha, \beta) - C_m(\dot{\alpha}) - C_m(q) - C_{m0}\Big)$$

Equation (4.12) is used during simulation. If the inner loop is fast enough, $\dot{\alpha}$ will almost conduct like what is expressed in (4.8), and so also the angle of attack α.

4.2.2 Lateral Mode

Lateral mode consists of both yaw and roll mode. In lateral mode, the same variables are controlled in both low and high speed. The variables are the roll rate p and the side-slip angle β and they are controlled by the surfaces δ_a and δ_r. At landing, β is sometimes commanded to a certain value, otherwise the desired β is normally zero.

In roll mode, the angular velocity p is controlled. Use (2.9) together with (2.12) and (2.14) and exchange $\dot{p}$ for $k_p(p_d$-$p)$. Solving for δ_a means that with this δ_a, the desired $\dot{p}$ and consequently the desired p, are obtained.

$$\begin{aligned}\delta_a = {} & \frac{1}{q_a S b\left(\frac{dC_l}{d\delta_a}(\delta_a) + \frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}(\delta_a)\right)}\Bigg(k_p(p_d - p)\left(1 - \frac{I_{zx}^2}{I_z I_x}\right)I_x \\ & - q_a S b\bigg(C_l(M, \alpha, \beta) + C_l(p) + C_l(r) + \frac{d}{d\delta_r}C_l(\delta_r)\delta_r + C_{l0}\bigg) - (I_y - I_z)qr \\ & - I_{zx}\Bigg(\frac{q_a S b\bigg(C_n(M, \alpha, \beta) + C_n(r) + \frac{d}{d\delta_r}C_n(\delta_r)\delta_r + C_{n0}\bigg) - I_{zx}qr + (I_x - I_y)pq}{I_z} \\ & + pq\Bigg)\Bigg)\end{aligned} \tag{4.13}$$

In yaw mode, both an inner loop and an outer loop are needed. In the outer loop the sideslip angle β is controlled, and in the inner loop the faster yaw rate r is controlled. Use the expression for $\dot{\beta}$ in (2.4). Exchanging $\dot{\beta}$ for $k_\beta(\beta_d$-$\beta)$ and r for r_d and then solving for r_d, means that with this r_d, the desired $\dot{\beta}$ is obtained and thus the desired β.

$$r_d = \frac{-k_\beta(\beta_d - \beta) + p\sin(\alpha) + \frac{1}{mv_T}(Y - F_T\cos(\alpha)\sin(\beta) + mg_3)}{\cos(\alpha)} \tag{4.14}$$

Continue with the inner loop. Use the expression for $\dot{r}$ in (2.11) together with (2.12) and (2.14), and exchange $\dot{r}$ for $k_r(r_d$-$r)$. If δ_a is solved for, that δ_a would generate the desired r. Insert (4.14) into that equation and solve for δ_a.

$$\delta_a = \frac{1}{q_a Sb\left(\frac{d}{d\delta_\alpha}C_n\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_\alpha}\right)}\left(\left(1 - \frac{I^2{}_{zx}}{I_x I_z}I_z\right)k_r\right. \qquad (4.15)$$

$$\cdot\left(\frac{p\sin(\alpha) - k_\beta(\beta_d - \beta) + \frac{1}{mV_T}(q_a SC_Y - F_T\cos(\alpha)\sin(\beta) + mg_3)}{\cos(\alpha)} - r\right)$$

$$- q_a Sb\left(C_n(M, \alpha, \beta) + C_n(r) + \frac{d}{d\delta_r}C_n\delta_r + C_{n0}\right)$$

$$- I_{zx}\left(\frac{1}{I_x}\left(q_a Sb\left(C_l(M, \alpha, \beta) + C_l(p) + C_l(r) + \frac{d}{d\delta_r}C_l\delta_r + C_{l0}\right)\right.\right.$$

$$\left.\left.\left. + I_{zx}pq + (I_y - I_z)qr\right) - qr\right) - (I_x - I_y)pq\right)$$

Let (4.13) be equal to (4.15) and solve for δ_r:

$$
\begin{aligned}
\delta_r = {} & \frac{1}{q_aSb}\left(\frac{-\frac{dC_l}{d\delta_r}-\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_r}}{\frac{dC_l}{d\delta_a}+\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}}+\frac{\frac{dC_n}{d\delta_r}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_r}}{\frac{d}{d\delta_a}C_n+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}}\right)^{-1}\cdot\Bigg(\frac{1}{\left(\frac{d}{d\delta_\alpha}C_n+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_\alpha}\right)}\cdot\Bigg(\Bigg(1 \qquad (4.16)\\
& -\frac{I^2_{zx}}{I_xI_z}\Bigg)I_zk_r\Bigg(\frac{p\sin(\alpha)-k_\beta(\beta_{ref}-\beta)+\frac{1}{mV_T}(q_aSC_Y-F_T\cos(\alpha)\sin(\beta)+mg_3)}{\cos(\alpha)}\\
& -r\Bigg)-q_aSb(C_n(M,\alpha,\beta)+C_n(r)+C_{n0})\\
& -I_{zx}\left(\frac{(q_aSb(C_l(M,\alpha,\beta)+C_l(p)+C_l(r)+C_{l0})+I_{zx}pq+(I_y-I_z)qr)}{I_z}-qr\right)\\
& -(I_x-I_y)pq\Bigg)-\frac{1}{\left(\frac{dC_l}{d\delta_a}+\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}\right)}\Bigg(k_p(p_d-p)\left(1-\frac{I^2_{zx}}{I_zI_x}\right)I_x-q_aSb(C_l(M,\alpha,\beta)\\
& +C_l(p)+C_l(r)+C_{l0})-(I_y-I_z)qr\\
& -I_{zx}\left(\frac{(q_aSb(C_n(M,\alpha,\beta)+C_n(r)+C_{n0})-I_{zx}qr+(I_x-I_y)pq)}{I_z}+pq\right)\Bigg)\Bigg)
\end{aligned}
$$

Equations (4.13) and (4.16) are implemented in the simulation environment. If the inner loop for the yaw mode is fast enough, the control signal δ_r generates a $\dot{\beta}$ that almost will be as desired. The response of β also follow of the expression for the desired $\dot{\beta}$. In roll mode there is no inner loop, so $\dot{p}$ would be as desired if problems such as model errors and noise not existed.

4.3 Cascaded Non-linear Dynamic Inversion, Method Two

The expressions for the control signals earlier in Chapter 4.2 are big, therefore another method that brings less complicated equations would be preferable. In this section, a cascaded NDI that uses fewer aerodynamic coefficients is tested. Instead the derivatives of some measured outputs are used together with measured control surface deflections. The problem is, that it is difficult to use derivatives of measured signals because the signals may be noisy. The derivatives are implemented as high pass filters with the break-frequency 20 rad/s. That is, for low frequencies it is a normal derivation, but for frequencies above 20 rad/s, the gain of the transfer function is constant. This filter is used to avoid high frequency signals from being amplified too much. The filter is shown in Figure 4.1 where α is input and its derivative is output.

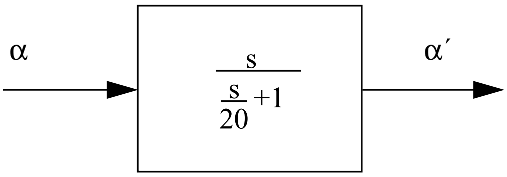

Figure 4.1: The filter for the derivation.

The control signals are calculated by substracting the desired *d* and sensored *s* equations for $\dot{p}$, $\dot{q}$, $\dot{r}$, $\dot{\alpha}$, and $\dot{\beta}$. These less complicated equations are then used in the same way as in the first method. That is in a cascaded control structure, where the inner loops are assumed to be much faster than the outer loops. This can be read about in [10], where also results from flight tests are presented.

The equations for the moments *L*, *M,* and *N* used for this controller, are the following:

$$L = q_a Sb\left(C_l(M, \alpha, \beta) + C_l(p) + C_l(r) + \frac{d}{d\delta_a}(C_l)\delta_a + \frac{d}{d\delta_r}(C_l)\delta_r + C_{l0}\right) \tag{4.17}$$

$$= q_a Sb\left(\Gamma + \frac{d}{d\delta_a}(C_l)\delta_a + \frac{d}{d\delta_r}(C_l)\delta_r\right)$$

$$M = q_a S\bar{c}\left(C_m(M, \alpha, \beta) + C_m(\dot{\alpha}) + C_m(q) + \frac{d}{d\delta_e}(C_m)\delta_e + C_{m0}\right) \tag{4.18}$$

$$= q_a S\bar{c}\left(\Lambda + \frac{d}{d\delta_e}(C_m)\delta_e\right)$$

$$N = q_a S\bar{b}\left(C_n(M, \alpha, \beta) + C_n(r) + \frac{d}{d\delta_a}(C_n)\delta_a + \frac{d}{d\delta_r}(C_n)\delta_r + C_{n0}\right) \tag{4.19}$$
$$= q_a S\bar{b}\left(\Omega + \frac{d}{d\delta_a}C_n(\delta_a)\delta_a + \frac{d}{d\delta_r}C_n(\delta_r)\delta_r\right)$$

4.3.1 Longitudinal Mode

Start with the pitch mode and use (2.3). In the first equation below, there are desired signals *d* and in the second equation there are sensored signals *s*.

$$\dot{\alpha}_d = q_d - (p\cos\alpha + r\sin\alpha)\tan\beta + \frac{1}{mv_T\cos\beta}(-L - F_T\sin\alpha + \tag{4.20}$$
$$mg(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta))$$

$$\dot{\alpha}_s = q_s - (p\cos\alpha + r\sin\alpha)\tan\beta + \frac{1}{mv_T\cos\beta}(-L - F_T\sin\alpha + \tag{4.21}$$
$$mg(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta))$$

where

$$\dot{\alpha}_d = k_\alpha(\alpha_d - \alpha) \tag{4.22}$$

Subtract (4.21) from (4.20) and use (4.22):

$$k_\alpha(\alpha_d - \alpha) - \dot{\alpha}_s = q_d - q_s \tag{4.23}$$

Solve for q$_d$, that is the angular velocity needed to obtain the desired $\dot{\alpha}$.

$$q_d = k_\alpha(\alpha_d - \alpha) - \dot{\alpha}_s + q_s \tag{4.24}$$

The angular velocity q_d is obtained from the inner loop where the elevator δ_e is used to control $\dot{q}$, which also means that q is controlled. Use (2.10) and (4.18):

$$\dot{q}_d = \frac{q_a S\bar{c}\left(\Lambda + \frac{d}{d\delta_e}(C_m)\delta_{ed}\right) + I_{zx}(r^2 - p^2) + (I_z - I_x)rp}{I_y} \tag{4.25}$$

$$\dot{q}_s = \frac{q_a S\bar{c}\left(\Lambda + \frac{d}{d\delta_e}(C_m)\delta_{es}\right) + I_{zx}(r^2 - p^2) + (I_z - I_x)rp}{I_y} \tag{4.26}$$

where

$$\dot{q}_d = k_q(q_d - q) \tag{4.27}$$

Subtract (4.26) from (4.25) and use (4.27).

$$I_y(k_q(q_d - q) - \dot{q}_s) = q_a S c \frac{dC_m}{d\delta_e}(\delta_{ed} - \delta_{es}) \tag{4.28}$$

Insert (4.24) into (4.28) and solve for δ_{ed}. That control signal will generate a $\dot{\alpha}$ that is pretty much the same as the desired one, if the inner loop is fast enough.

$$\delta_{ed} = ((k_\alpha(\alpha_d - \alpha) - \dot{\alpha}_s)k_q - \dot{q}_s)\frac{I_y}{q_a S c \frac{dC_m}{d\delta_e}} + \delta_{es} \tag{4.29}$$

4.3.2 Lateral Mode

The aileron δ_a and the rudder δ_r are used to control $\dot{p}$. Use (2.9), (4.17), and (4.19):

$$\begin{aligned}\dot{p}_d = {} & \left(q_a S b\left(\Gamma + \frac{dC_l}{d\delta_r}\delta_{rd} + \frac{dC_l}{d\delta_a}\delta_{ad}\right) + I_{zx}\left(\left(q_a S b\left(\Omega + \frac{dC_n}{d\delta_r}\delta_{rd} + \frac{dC_n}{d\delta_a}\delta_{ad}\right)\right.\right.\right. \\ & \left.\left.\left. - I_{zx} q r + (I_x - I_y) p q\right)\frac{1}{I_z} + pq\right) + (I_y - I_z) q r\right)\frac{1}{\left(1 - \frac{I_{zx}^2}{I_x I_z}\right) I_x}\end{aligned} \tag{4.30}$$

$$\begin{aligned}\dot{p}_s = {} & \left(q_a S b\left(\Gamma + \frac{dC_l}{d\delta_r}\delta_{rs} + \frac{dC_l}{d\delta_a}\delta_{as}\right) + I_{zx}\left(\left(q_a S b\left(\Omega + \frac{dC_n}{d\delta_r}\delta_{rs} + \frac{dC_n}{d\delta_a}\delta_{as}\right)\right.\right.\right. \\ & \left.\left.\left. - I_{zx} q r + (I_x - I_y) p q\right)\frac{1}{I_z} + pq\right) + (I_y - I_z) q r\right)\frac{1}{\left(1 - \frac{I_{zx}^2}{I_x I_z}\right) I_x}\end{aligned} \tag{4.31}$$

$$\dot{p}_d = k_p(p_d - p) \tag{4.32}$$

Subtract (4.31) from (4.32), use (4.32), and solve for δ_{ad}:

$$\delta_{ad} = \frac{1}{\frac{dC_l}{d\delta_a} + \frac{I_{zx}}{I_z}\frac{d}{d\delta_a}C_n}\left(\frac{k_p(p_d - p) - \dot{p}_s}{q_a S b}\left(1 - \frac{I_{zx}^2}{I_x I_z}\right)I_x + \delta_{as}\left(\frac{dC_l}{d\delta_a} + \frac{I_{zx}}{I_z}\frac{d}{d\delta_a}C_n\right) - (\delta_{rd} - \delta_{rs})\left(\frac{dC_l}{d\delta_r} + \frac{I_{zx}}{I_z}\frac{d}{d\delta_r}C_n\right)\right) \tag{4.33}$$

Continue with the yaw mode. Start with the outer loop where β is to be controlled. Use (2.4) where g_3 is found in (2.5):

$$\dot{\beta}_d = p\sin(\alpha) - r_d\cos(\alpha) + \frac{1}{mv_T}(Y - F_T\cos(\alpha)\sin(\beta) + mg_3) \tag{4.34}$$

$$\dot{\beta}_s = p\sin(\alpha) - r_s\cos(\alpha) + \frac{1}{mv_T}(Y - F_T\cos(\alpha)\sin(\beta) + mg_3) \tag{4.35}$$

$$\dot{\beta}_d = k_\beta(\beta_d - \beta) \tag{4.36}$$

Subtract (4.35) from (4.34), use (4.36), and solve for r_d:

$$r_d = \frac{\dot{\beta}_s - k_\beta(\beta_d - \beta)}{\cos(\alpha)} + r_s \tag{4.37}$$

With the angular velocity r_d in (4.37), the desired $\dot{\beta}$, and also the desired β are obtained. The yaw rate is controlled by the surfaces δ_a and δ_r in the inner loop. Use (2.11), (4.17), and (4.19):

$$\dot{r}_d = \frac{1}{\left(1 - \frac{I_{zx}^2}{I_x I_z}\right)I_z}\left(q_a S\bar{b}\left(\Omega + \frac{d}{d\delta_a}(C_n)\delta_{ad} + \frac{d}{d\delta_r}(C_n)\delta_{rd}\right) + (I_x - I_y)pq + I_{zx}\left(\frac{q_a S b\left(\Gamma + \frac{d}{d\delta_a}((C_l)\delta_{ad}) + \frac{d}{d\delta_r}(C_l)\delta_{rd}\right) + I_{zx}pq + (I_y - I_z)qr}{I_x} - qr\right)\right) \tag{4.38}$$

$$\dot{r}_s = \frac{1}{\left(1-\frac{I_{zx}^2}{I_xI_z}\right)I_z}\Bigg(q_aS\bar{b}\left(\Omega+\frac{d}{d\delta_a}(C_n)\delta_{as}+\frac{d}{d\delta_r}(C_n)\delta_{rs}\right)+(I_x-I_y)pq \tag{4.39}$$

$$+I_{zx}\left(\frac{q_aSb\left(\Gamma+\frac{d}{d\delta_a}((C_l)\delta_{as})+\frac{d}{d\delta_r}(C_l)\delta_{rs}\right)+I_{zx}pq+(I_y-I_z)qr}{I_x}-qr\right)\Bigg)$$

$$\dot{r}_d = k_r(r_d-r) \tag{4.40}$$

Subtract (4.39) from (4.38) and use (4.38):

$$(k_r(r_d-r)-\dot{r}_s)\left(1-\frac{I_{zx}^2}{I_xI_z}\right)I_z = q_aSb\left(\frac{d}{d\delta_a}C_n(\delta_{ad}-\delta_{as})+\frac{d}{d\delta_r}C_n(\delta_{rd}-\delta_{rs})\right. \tag{4.41}$$

$$\left.+\frac{I_{zx}}{I_x}\left(\frac{d}{d\delta_a}C_l(\delta_{ad}-\delta_{as})+\frac{d}{d\delta_r}C_l(\delta_{rd}-\delta_{rs})\right)\right)$$

Insert (4.37) into (4.41) and solve for δ_{ad}:

$$\delta_{ad} = \frac{1}{\left(\frac{dC_n}{d\delta_a}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}\right)}\Bigg(\Bigg(\bigg(\frac{k_\beta(\beta_d-\beta)-\dot{\beta}_s}{\cos(\alpha)}(-k_r)-\dot{r}_s\bigg)\frac{\left(1-\frac{I_{zx}^2}{I_xI_z}\right)I_z}{q_aSb}+ \tag{4.42}$$

$$\left(\frac{dC_n}{d\delta_a}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}\right)\delta_{as}-(\delta_{rd}-\delta_{rs})\left(\frac{dC_n}{d\delta_r}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_r}\right)\Bigg)$$

There are now two equations where δ_{ad} is solved for, (4.33) and (4.42). Set those equations equal and solve for δ_{rd}:

$$\delta_{rd} = \frac{1}{\left(\frac{-\frac{dC_l}{d\delta_r} - \frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_r}}{\frac{d}{d\delta_a}C_l + \frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}} + \frac{\frac{dC_n}{d\delta_r} + \frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_r}}{\frac{d}{d\delta_a}C_n + \frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}}\right)}\left(\frac{1}{\left(\frac{dC_n}{d\delta_a} + \frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}\right)}\left(\left(\frac{k_\beta(\beta_d - \beta) - \dot{\beta}_s}{\cos(\alpha)} \cdot \right.\right.\right. \tag{4.43}$$

$$\left.(-k_r) - \dot{r}_s\right) \cdot \frac{\left(1 - \frac{I_{zx}^2}{I_x I_z}\right)I_z}{q_a S b} + \left(\frac{dC_n}{d\delta_a} + \frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}\right)\delta_{as} + \left.\left(\frac{dC_n}{d\delta_r} + \frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_r}\right)\delta_{rs}\right)$$

$$- \frac{1}{\left(\frac{dC_l}{d\delta_a} + \frac{I_{zx}}{I_x}\frac{dC_n}{d\delta_a}\right)}\left((k_p(p_d - p) - \dot{p}_s)\frac{\left(1 - \frac{I_{zx}^2}{I_x I_z}\right)I_x}{q_a S b} + \left(\frac{dC_l}{d\delta_a} + \frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}\right)\delta_{as} + \right.$$

$$\left.\left.\left(\frac{dC_l}{d\delta_r} + \frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_r}\right)\delta_{rs}\right)\right)$$

Equations (4.29), (4.33), and (4.43) are used during simulation. As can be seen, these equations are less complicated than those generated by the design methods “Pure Non-linear Dynamic Inversion” and “Cascaded Non-linear Dynamic Inversion Method One”.

4.4 Control of the Load Factor

At high speed, the load factor n_z is controlled. The control signals in this section can be used together with the earlier calculated control signals in longitudinal and lateral mode.

Preferably, there would be an expression for $\dot{n}_z$ so the desired dynamics could be demanded in the same way as for the other derivatives, that is

$$\dot{n}_{zd} = k_{n_z}(n_{zd} - n_z)$$

One expression looks like:

$$\dot{n}_z \approx \frac{\dot{L}}{mg} = \frac{q_a S \dot{C}_L}{mg} \tag{4.44}$$

One would prefer to have n_z in an outer loop and the pitch rate q in an inner loop. Therefore equation (4.44) must contain q in some way. An expression for $\dot{C}_L$ is presented below, see also (2.1), (2.20), and (2.21).

$$\dot{C}_L = \frac{dC_L}{d\alpha} \cdot \frac{d\alpha}{dt} + \frac{dC_L}{d\beta} \cdot \frac{d\beta}{dt} + \frac{dC_L}{d\dot{\alpha}} \cdot \frac{d}{dt}\dot{\alpha} + \ldots$$

There are already equations for $\dot{\alpha}$ and $\dot{\beta}$ but there may be problems with the other expressions. Therefore, less complicated alternatives will instead be investigated.

4.4.1 Control of the Load Factor by the Angle of Attack

One way to control n_z is to command a desired n_z, transform the demand into a desired α, and then control α with the same controller as before. The task is now to find an expression that relates n_z to α, as:

$$n_{zd} = k \cdot \alpha_d + m$$

An expression for n_z is presented below, see also (2.1).

$$n_z \approx \frac{L}{mg} = \frac{q_a S C_L}{mg} \approx \frac{q_a S}{mg} \cdot \left(C_{L0} + \frac{dC_L}{d\alpha}\alpha\right) \tag{4.45}$$

Substituting n_{zd} and α_d for n_z and α, and solving for α_d yields:

$$\alpha_d \approx \frac{\frac{mg}{q_a S} n_{zd} - C_{L0}}{\frac{dC_L}{d\alpha}} \tag{4.46}$$

In (4.46) the following equations are needed:

$$\frac{dC_L}{d\alpha} = -\frac{dC_T}{d\alpha}\sin(\alpha) - C_T\cos(\alpha) + \frac{dC_N}{d\alpha}\cos(\alpha) - C_N\sin(\alpha) \tag{4.47}$$

$$C_{L0} = C_L - \frac{dC_L}{d\alpha}\alpha \tag{4.48}$$

where

$$\frac{dC_N}{d\alpha} = \frac{d}{d\alpha}C_N(M, \alpha, \beta) + \frac{d}{d\alpha}C_N(\dot{\alpha}) + \frac{d}{d\alpha}C_N(q) + \frac{d}{d\alpha}C_N(\delta_e)$$

and

$$\frac{dC_T}{d\alpha} = \frac{d}{d\alpha}C_T(M, \alpha, \beta) + \frac{d}{d\alpha}C_T(\delta_a) + \frac{d}{d\alpha}C_T(\delta_r) + \frac{d}{d\alpha}C_T(\delta_e)$$

Equation (4.46) is implemented in the simulation environment.

4.4.2 Control of the Load Factor by the Elevator

In this section a n_z-controller that do not use an α-controller is calculated. Use (2.1), (2.2), and (2.24). If the angle of attack is small, sin(α) can be approximated by zero.

$$\left(C_N(M, \alpha, \beta) + C_N(\dot{\alpha}) + C_N(q) + \frac{d}{d\delta_e}C_N(\delta_e)\delta_e + C_{N0}\right)\cos(\alpha)q_aS = n_zmg$$

Solving for the elevator δ_e and substituting n_{zd} for n_z yields:

$$\delta_e = \frac{\dfrac{n_{zd}mg}{q_aS\cos(\alpha)} - C_N(M, \alpha, \beta) - C_N(\dot{\alpha}) - C_N(q) - C_{N0}}{\dfrac{d}{d\delta_e}C_N(\delta_e)} \tag{4.49}$$

This controller can also be calculated without the approximation where *sin*(α) is set to zero. Equation (4.49) has been implemented in the simulation environment. Unfortunately the derivative in the denominator becomes zero after some simulation time. The problem is that the inputs to the tables, where the aerodynamic coefficients are calculated, get values that are not part of the tables. The result is that some aerodynamic coefficients are set to zero, which leads to division by zero. The underlying problem is that it is difficult to control n_z or α without an inner loop that controls the pitch rate q.

4.5 Robustness

One problem when the SHARC is controlled, is that the model of the system is not perfect and therefore a robust controller is needed. Analyzing the robustness of a system, the different coefficients can be changed to see how the system reacts. In this section integrators will be added to the controllers that use the cascaded NDI approach. Another way to obtain a robust controller, is to use μ-synthesis, see [7], [8], and [11].

Introduce an extra state that describes the difference between the desired signal *r* and measured signal *y*:

$$\dot{x}_{n+1} = r - y = r - cx \tag{4.50}$$

In (4.50) the error is zero in stationarity and this is obtained (if stationarity is reached), even if there are model errors or disturbances. This is the integral action. The name derive from the fact that if (4.50) is integrated, it can be seen that the new state is an integration of the error.

A common control signal that includes the state feedback matrix *L* and the new state x_{n+1}, is:

$$u = l_0 r - Lx - l_{n+1} x_{n+1} \tag{4.51}$$

With the control signal in (4.51) the resulting closed loop system becomes:

$$\begin{bmatrix} \dot{x} \\ \dot{x}_{n+1} \end{bmatrix} = \begin{bmatrix} A - BL & -Bl_{n+1} \\ -C & 0 \end{bmatrix} \begin{bmatrix} x \\ x_{n+1} \end{bmatrix} + \begin{bmatrix} Bl_0 \\ 1 \end{bmatrix} r = \tilde{A}x + \tilde{B}r$$

Decide l_0, l_{n+1} and the matrix *L* by studying the poles of the closed loop system and place the poles so that the requirements regarding damping ratios, frequencies and time constants are fulfilled. The closed loop system without an integrator is presented in Figure 4.2 and the system that includes an integrator in Figure 4.3.

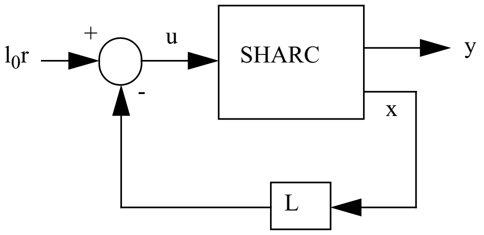

Figure 4.2: The system as a state feedback.

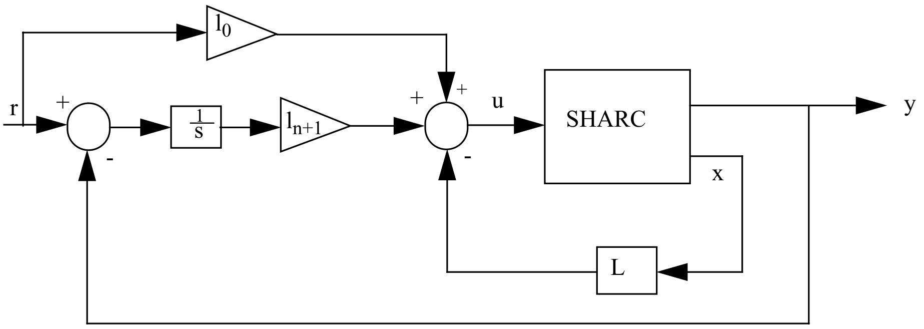

Figure 4.3: An integrator added to the closed loop system.

If the control signal in (4.51) is to be implemented, it could look like this:

$$\delta_e = \ldots + l_\alpha \int (\alpha_d - \alpha) dt$$
$$\delta_a = \ldots + l_p \int (p_d - p) dt$$
$$\delta_r = \ldots + l_\beta \int (\beta_d - \beta) dt$$

The problem is that there are couplings between roll and yaw, so the integration of the error in sideslip should perhaps be added also to δ_a and vice versa.

A better solution is to add the integrators to $\dot{\alpha}_d$, $\dot{\beta}_d$ and $\dot{p}_d$:

$$\dot{\alpha}_d = k_\alpha(\alpha_d - \alpha) + l_\alpha \int (\alpha_d - \alpha) \tag{4.52}$$

$$\dot{\beta}_d = k_\beta(\beta_d - \beta) + l_\beta \int (\beta_d - \beta) \tag{4.53}$$

$$\dot{p}_d = k_p(p_d - p) + l_p \int (p_d - p) \tag{4.54}$$

To improve the integrators, filters that the desired signals can pass, are added. The filters shall resemble the response of the system. They are added in order for the integrated error not to be too big in the beginning of a change in command. In Figure 4.4 an integrator for α is presented. The control signals when integrators are added, are presented in Appendix B.

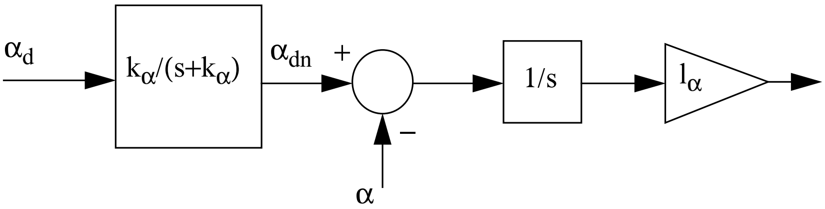

Figure 4.4: An integrator for α.

At high speed, when n_z is controlled instead of α, there would preferably be an integrator even for n_z. Especially since the expression that relates the desired n_z and the desired α consists of approximations. Simulations have been conducted when the integrator for α has been removed and an integrator for n_z has been added directly to δ_e. It is not wise to have both integrators at the same time, because of the approximation in the expression relating n_z and α. That would cause the integrators to work in different directions. Unfortunately, there has not been enough time to evaluate the new integrator.

4.6 Poles of the Closed Loop System

In this section a system description will be made. The poles, damping ratios, frequencies and time constants of the system will also be calculated. These are later used to check if the system fulfils the requirements. First some theory is presented.

4.6.1 General Theory

Linear systems can be represented by a state space description, see [2]:

$$\dot{x} = Ax + Bu \tag{4.55}$$

$$y = Cx \tag{4.56}$$

To decide the poles of the system, solve for s in (4.57).

$$|sI - A| = 0 \tag{4.57}$$

A closed loop system is obtained by a state feedback:

$$u = -Lx + l_0 r \tag{4.58}$$

Insert (4.58) into (4.55):

$$\dot{x} = (A - BL)x + Bl_0 r$$

The poles for the closed loop system is obtained by solving for s in (4.59).

$$|sI - (A - BL)| = 0 \tag{4.59}$$

A system can also be expressed by a transfer function. A second order transfer function looks like:

$$\frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2} \tag{4.60}$$

where ω_n is the frequency and also the absolute distance to the poles. If the distance is long, the system is fast. The damping ratio ζ is defined as:

$$\zeta = \cos(\varphi) \tag{4.61}$$

where φ is the angle to one of the complex conjugated poles. If both poles are on the negative real axis and are unequal, the definition in (4.61) is not true and the damping ratio can be greater than one. This will be the case for the approximated system description for the SHARC.

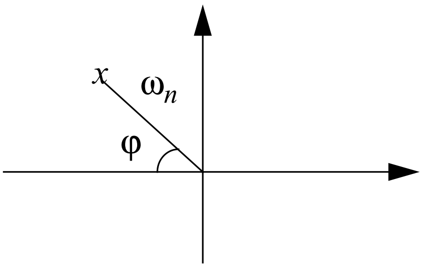

Figure 4.5: The definitions for ω_n and φ.

The rapidity of a system can be described by the time constant, which is the time it takes to reach 63 percent of a desired level.

4.6.2 System without Integrators

First the system description for the closed loop system without integrators are derived. The first step is to find the equations that will be used when the states of the closed loop system are created.

The equations for the desired derivatives in the inner loops can be used directly:

$$\dot{q} = k_q(q_d - q) \tag{4.62}$$

$$\dot{p} = k_p(p_d - p) \tag{4.63}$$

$$\dot{r} = k_r(r_d - r) \tag{4.64}$$

For the outer loops some calculations have to be done. Start with the pitch mode. Use (2.3) and solve for the q that generates the desired $\dot{\alpha}$, that is, set $\dot{\alpha} = k_\alpha(\alpha_d - \alpha)$. Also exchange q for q_d, because the inner loop is supposed to be much faster than the outer loop. Add the term q_d-q_d to (2.3) and insert the new equation into q_d (the first term). The result is:

$$\dot{\alpha} = k_\alpha(\alpha_d - \alpha) + q - q_d \tag{4.65}$$

In lateral mode, (2.4), (2.9), and (2.11) are used together with (2.5), (2.12), and (2.14). Start with (2.4) and solve for the r that generates the wanted $\dot{\beta}$, that is, set $\dot{\beta} = k_\beta(\beta_d - \beta)$. Also exchange r for r_d, because the inner loop is supposed to be much faster than the outer loop. The equation is now:

$$r_d = \frac{-k_\beta(\beta_{ref} - \beta) + p\sin(\alpha) + \frac{1}{mV}(Y - F_T\cos(\alpha)\sin(\beta) + mg_3)}{\cos(\alpha)} \tag{4.66}$$

Add the term $\cos(\alpha)(r_d - r_d)$ to (2.4) and insert (4.66) into -r_d in that term. The result is:

$$\dot{\beta} = k_\beta(\beta_d - \beta) + \cos(\alpha)(r_d - r) \tag{4.67}$$

Equations (4.62), (4.63), (4.64), (4.65), and (4.67) are also obtained for the system when the controller named “Cascaded NDI Method Two”, is used. The following new states can be created:

$$z_1 = \alpha$$
$$z_2 = q - q_d$$
$$z_3 = \beta$$
$$z_4 = r - r_d$$
$$z_5 = p$$

The derivatives of the states are:

$$\begin{aligned}\dot{z}_1 &= \dot{\alpha} = k_\alpha(\alpha_d - z_1) + z_2\\ \dot{z}_2 &\approx \dot{q} = -k_q z_2\\ \dot{z}_3 &= \dot{\beta} = k_\beta(\beta_d - z_3) - \cos(z_1) z_4\\ \dot{z}_4 &\approx \dot{r} = -k_r z_4\\ \dot{z}_5 &= \dot{p} = k_p(p_d - z_5)\end{aligned}$$

The expressions for $\dot{z}_2$ and $\dot{z}_4$ are approximations because the derivatives of q_d and r_d are non-zero. Another approximation is made, set $cos(z_1)$=1 to get a linearized system.

$$\dot{z} = \begin{bmatrix} -k_\alpha & 1 & 0 & 0 & 0 \\ 0 & -k_q & 0 & 0 & 0 \\ 0 & 0 & -k_\beta & -1 & 0 \\ 0 & 0 & 0 & -k_r & 0 \\ 0 & 0 & 0 & 0 & -k_p \end{bmatrix} z + \begin{bmatrix} k_\alpha & 0 & 0 \\ 0 & 0 & 0 \\ 0 & k_\beta & 0 \\ 0 & 0 & 0 \\ 0 & 0 & k_p \end{bmatrix} \begin{bmatrix} \alpha_d \\ \beta_d \\ p_d \end{bmatrix} = Az + Br$$

With the approximations mentioned above, the poles of the system are $-k_\alpha$, $-k_q$, $-k_\beta$, $-k_r$, and $-k_p$. The closed loop system can also be written:

$$\begin{bmatrix} \dot{z}_1 \\ \dot{z}_2 \end{bmatrix} = \begin{bmatrix} -k_\alpha & 1 \\ 0 & -k_q \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} k_\alpha \\ 0 \end{bmatrix} \alpha_d \qquad (4.68)$$

$$\begin{bmatrix} \dot{z}_3 \\ \dot{z}_4 \end{bmatrix} = \begin{bmatrix} -k_\beta & -1 \\ 0 & -k_r \end{bmatrix} \begin{bmatrix} z_3 \\ z_4 \end{bmatrix} + \begin{bmatrix} k_\beta & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \beta_d \\ p_d \end{bmatrix} \qquad (4.69)$$

$$\dot{z}_5 = -k_p z_5 + k_p p_d \qquad (4.70)$$

There are three different modes that have to fulfil the requirements; short period, Dutch roll and roll. Short period is the longitudinal motion where the derivatives of the variables that vary slowly are approximated by zero. Dutch roll is an approximation to the lateral oscillation. It is a flat sideslipping motion in which rolling is suppressed. Finally, in roll mode, there is a requirement on how fast the aircraft rolls, in our case around the body axis x_b.

The poles for the system in (4.68), that represents the pitch mode and also the short period, are $-k_\alpha$ and $-k_q$. The damping ratio and the frequency can be calculated by identifying the poles with the characteristic equation:

$$s^2 + 2\zeta\omega_n s + \omega_n^2$$

this yields:

$$\omega_{n_{sp}} = \sqrt{k_\alpha k_q}$$

$$\zeta_{sp} = \frac{k_\alpha + k_q}{2\sqrt{k_\alpha k_q}}$$

For the system which describes the yaw mode and also the Dutch roll, see (4.69), the poles are -k_β and -k_r. The damping ratio and the frequency are:

$$\omega_{n_d} = \sqrt{k_\beta k_r}$$

$$\zeta_d = \frac{k_\beta + k_r}{2\sqrt{k_\beta k_r}}$$

For the system in (4.70) which describes roll, the pole is -k_p and the time constant is:

$$\tau = \frac{\ln(0.37)}{-k_p}$$

4.6.3 System with Integrators

It will now be a discussion about the system when integrators and filters, that the desired signals pass, are added. If the filters really resemble the system, the signals through the integrators are always zero. If the calculations are ideal, the integrating part can therefore be removed from the system description. The signals are only non-zero when there are model errors, and in that case, the system description is anyhow not correct. The system description in section 4.6.2 will therefore be used even when integrators are added.

5 Simulation

5.1 Control Systems Requirements

There are some parameters in the controller that have to be decided in order for the system to behave in a proper way. The parameters can for example be chosen by studying the step responses or by calculating the parameters, so the requirements regarding damping ratios, frequencies and time constants are fulfilled. In this thesis there will be a mix of the two methods. The damping ratios, frequencies and time constants were calculated in section 4.6.2, using the system description that is an approximation of the real system. Therefore also the plots have to be studied.

There are some definitions needed in order to understand the requirements for the SHARC. The definitions consider different flight phases and flying-quality levels, see [12]. The definitions considering the flying-quality levels are set up for an aircraft with a pilot. They will be used in spite of that the SHARC is unmanned, because no other definitions are currently available.

The flight phases are:

A: Air-to-air combat, ground attack, weapon delivery, reconnaissance, and terrain following.
B: Climb, cruise, loiter, and descent.
C: Takeoff, approach, and landing.

The flying-quality levels are:

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

In this report, level 1 and flight phases A and C are considered.

For short period there are requirements regarding damping ratio and frequency in terms of $\frac{\omega_{n_{sp}}}{n_{z/\alpha}}$. For flight phase A, the requirements are:

$$0.35 \le \zeta_{sp} \le 1.30$$
$$0.28 \le \frac{\omega_{n_{sp}}}{n_{z/\alpha}} \le 3.60$$
$$\omega_{n_{sp}} \ge 1$$

and for flight phase C the requirements are:

$$0.35 \le \zeta_{sp} \le 1.30$$
$$0.16 \le \frac{\omega_{n_{sp}}}{n_{z/\alpha}} \le 3.60$$
$$\omega_{n_{sp}} \ge 0.7$$

For Dutch roll, the requirements regarding damping and frequency for flight phase A are:

$$\zeta_d \ge 0.19$$
$$\zeta_d \omega_{n_d} \ge 0.35$$
$$\omega_{n_d} \ge 1.0$$

For flight phase C, the requirements are:

$$\zeta_d \ge 0.08$$
$$\zeta_d \omega_{n_d} \ge 0.15$$
$$\omega_{n_d} \ge 0.4$$

For roll mode the requirement regarding the time constant for flight phases A and C is:

$$\tau \le 1.4$$

5.2 Deciding the Parameters

It is time to choose the parameters k_α, k_q, k_β, k_r, k_p, l_α, l_β and l_p in order for the closed loop system to fulfil the requirements in section 5.1. The flight cases that will be simulated are presented in Table 5.1 together with the chosen parameters. The reason for testing those flight cases, is that the performances vary for different altitudes and velocities. Therefore, simulations on both low and high altitude, and low and high speed must be conducted. Finally, landing is also an interesting case. In this section, the frequencies, damping ratios and time constants are calculated for the parameters belonging to those flight cases.

Table 5.1: Different flight cases and parameters belonging to them.

Mach number, altitude [m]	k_α	k_q	k_β	k_r	k_p	l_α	l_β	l_p
0.21, 50	3	5.5	2.5	4	2.5	1.5	0.3	0.3
0.21, 50	3	5.5	2.5	4	2.5	-	-	-
0.3, 1000	3	5.5	2.5	4	2.5	1.5	0.3	0.3
0.3, 1000	3	5.5	2.5	4	2.5	-	-	-
0.7, 6000	3	5.5	2.5	4	2.5	1.5	0.3	0.3

As can be seen in Table 5.1, the parameters are the same for all flight cases. If the parameters are changed during flight, scheduling or some similar method must be implemented to avoid abrupt changes. Scheduling is a linear interpolation used between different values of the parameters.

The requirement for the frequency for short period includes the ratio $\Delta n_z/\Delta\alpha$, that varies for different Mach numbers and altitudes, and therefore has to be calculated. The ratio $\Delta n_z/\Delta\alpha$ is calculated for the open system when control surface δ_e is changed with one degree. The values are presented in Table 5.2.

Table 5.2: $\Delta n_z/\Delta\alpha$ for different flight cases.

Mach number, altitude [m]	$\Delta n_z/\Delta\alpha$ [1/rad]:
0.21, 50	4.84
0.3, 1000	7.60
0.7, 6000	32.57

The damping ratios, frequencies and time constants can now be calculated and are presented in Table 5.3.

Table 5.3: Calculated frequencies, damping ratios and time constants.

Mach number, altitude [m]	ζ_{sp}	$\omega_{nsp}/\Delta n_z/\Delta\alpha$	ω_{nsp}	ζ_d	$\zeta_d\omega_{nd}$	ω_{nd}	τ
0.21, 50	1.05	3.41	4.06	1.03	3.25	3.16	0.40
0.3, 1000	1.05	2.17	4.06	1.03	3.25	3.16	0.40
0.7, 6000	1.05	0.51	4.06	1.03	3.25	3.16	0.40

All values in Table 5.3 fulfil the requirements in section 5.1. Do not forget that these values have been calculated from systems descriptions that are approximations of the system. Therefore the values are not exact, and the plots also have to be studied.

5.3 Simulation Environment

The simulations are conducted on a mathematical model of the SHARC and the building of this model is not a part of this thesis. The mathematical model which consists of rigid body equations, is implemented in SystemBuild in MatrixX, see Appendix A. The code with the algorithms are written in Fortran.

When simulating the closed loop system, initial values are set by using trimmed values of the states. They are generated in stationarity for a fix Mach number and altitude. There are many integration algorithms to choose from in MatrixX. During simulation fixed-step Kutta-Merson is used, because this algorithm is fast compared to some others, and it is accurate enough.

5.4 Plots and Comments

In this section the controllers are tested for the different flight cases. There have been more tests conducted than those presented here, but this is a representative selection. For example Mach number 0.3, altitude 6000 m and Mach number 0.7, altitude 1000 m have also been simulated. In Table 5.4 there is information about all flight cases that will be simulated together with the controllers. In Table 5.4 there are some expressions that have to be defined:

NDI1: The controller that uses the Cascaded NDI Method One.
NDI1+i: Same as above, but integrators are added to get a robust controller.
NDI2: The controller that uses the Cascaded NDI Method Two.
NDI2+i: Same as above, but integrators are added.

Table 5.4: Controllers and changed desired signals for different flight cases.

Mach number, altitude [m]	Changed desired signal	Controller	Comment
0.21, 50	α	NDI1 + i	
0.21, 50	α	NDI2	
0.21, 50	α	NDI2 + i	
0.21, 50	β	NDI1 + i	
0.21, 50	p	NDI1 + i	
0.3, 1000	α	NDI1 + i	
0.3, 1000	p	NDI1	
0.3, 1000	p	NDI1 + i	
0.3, 1000	p	NDI2 + i	
0.3, 1000	α, β, p	NDI1 + i	
0.3, 1000	-	NDI1 + i	wind disturbance, pitch
0.3, 1000	-	NDI1 + i	wind disturbance, yaw
0.3, 1000	α	NDI1 + i	model error
0.3, 1000	α	NDI2 + i	model error
0.7, 6000	n_z	NDI1 + i	
0.7, 6000	p	NDI1 + i	

The plots from the simulations that now will be presented, show many signals. First of all the desired signals and the measured counterparts are presented, that is α_d, α, β_d, β, p_d, and p. At high velocities n_{zd} and n_z are of course presented in the plots. The signals in the inner loops are also of interest, that is q and r. It is also interesting to see how the control surfaces behave, therefore δ_e, δ_a, and δ_r are plotted. If the control surfaces reach their physical limits, it will be marked in the plots. To see how many degrees the SHARC has rolled, ϕ is presented, and to see the angle between nose and skyline, θ is plotted. If disturbances are added, these are also plotted.

When the SHARC rolls for a long duration of time, it is not able to keep the attitude θ. The attitude θ has to be controlled in an outer loop and this is not done in this thesis. Therefore the plots for θ sometimes look odd.

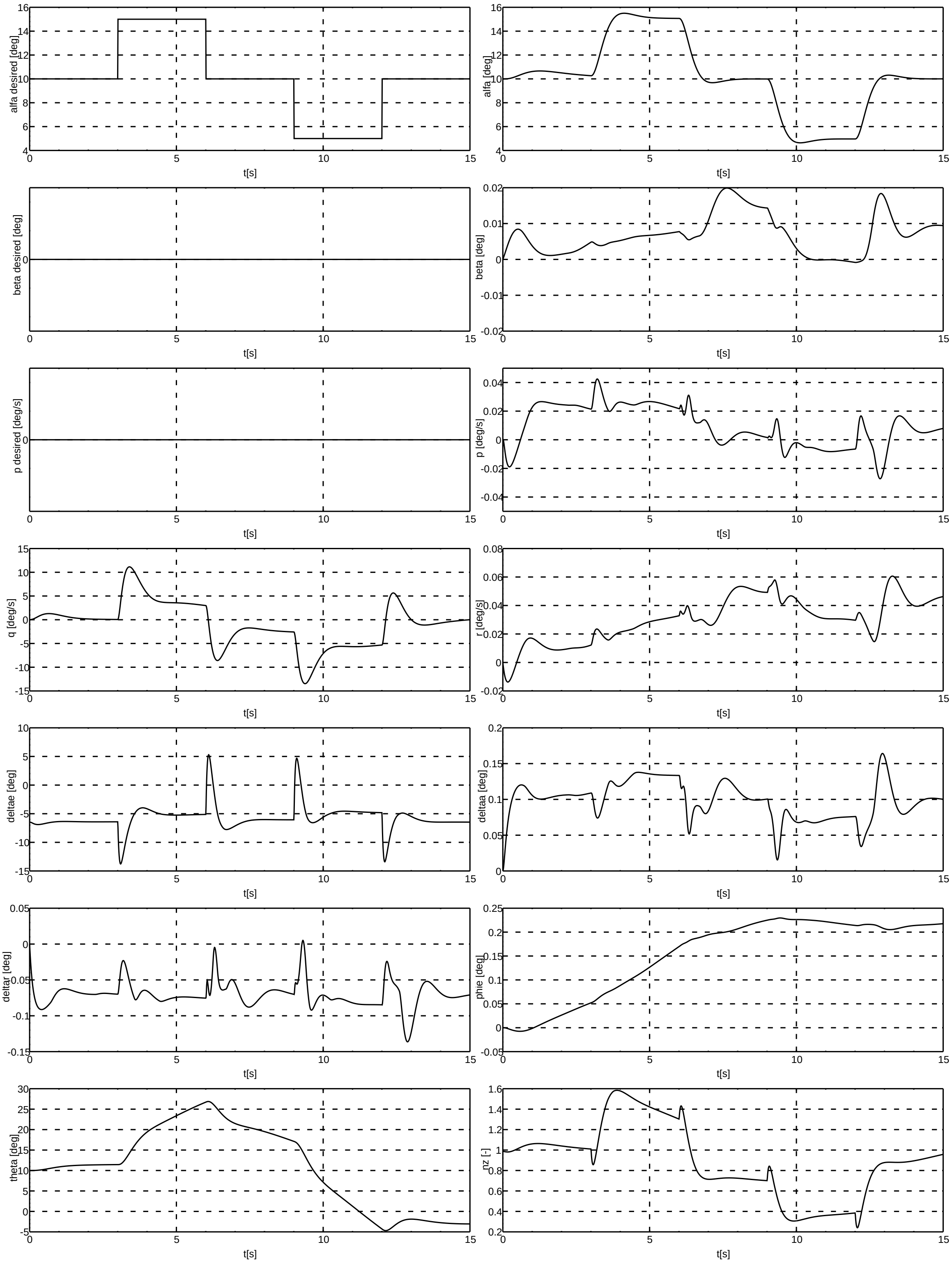

Figure 5.1: Mach number 0.21, altitude 50 m, NDI1+integrator.

In Figure 5.1 the angle of attack α follows the desired signal. The overshoot in the plot for α, depends on the integrator needed to get a more robust system.

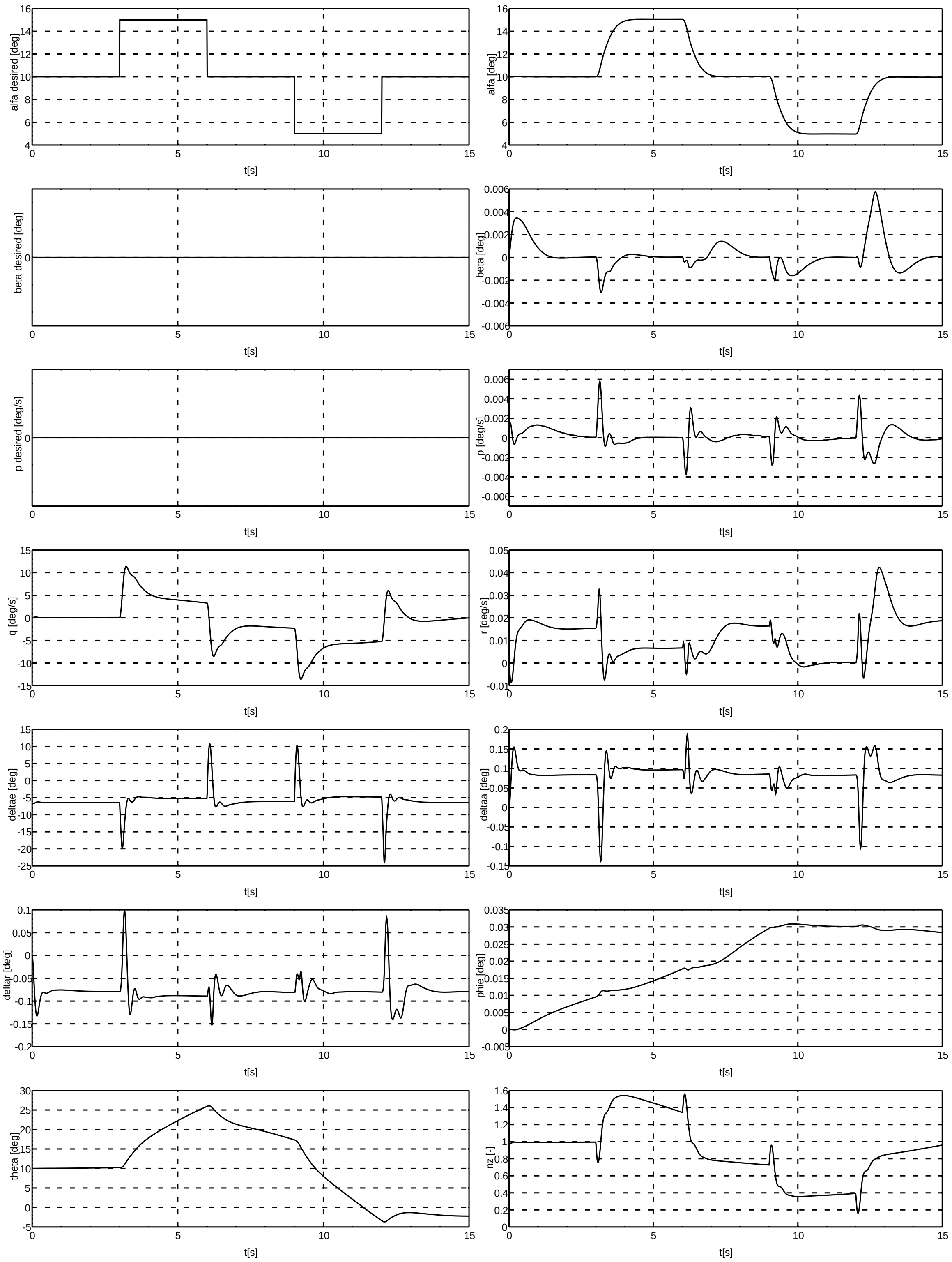

Figure 5.2: Mach number 0.21, altitude 50 m, NDI2.

In Figure 5.2, the control surfaces deflections are changed more abruptly, compared to the surfaces when the controller NDI1 is used.

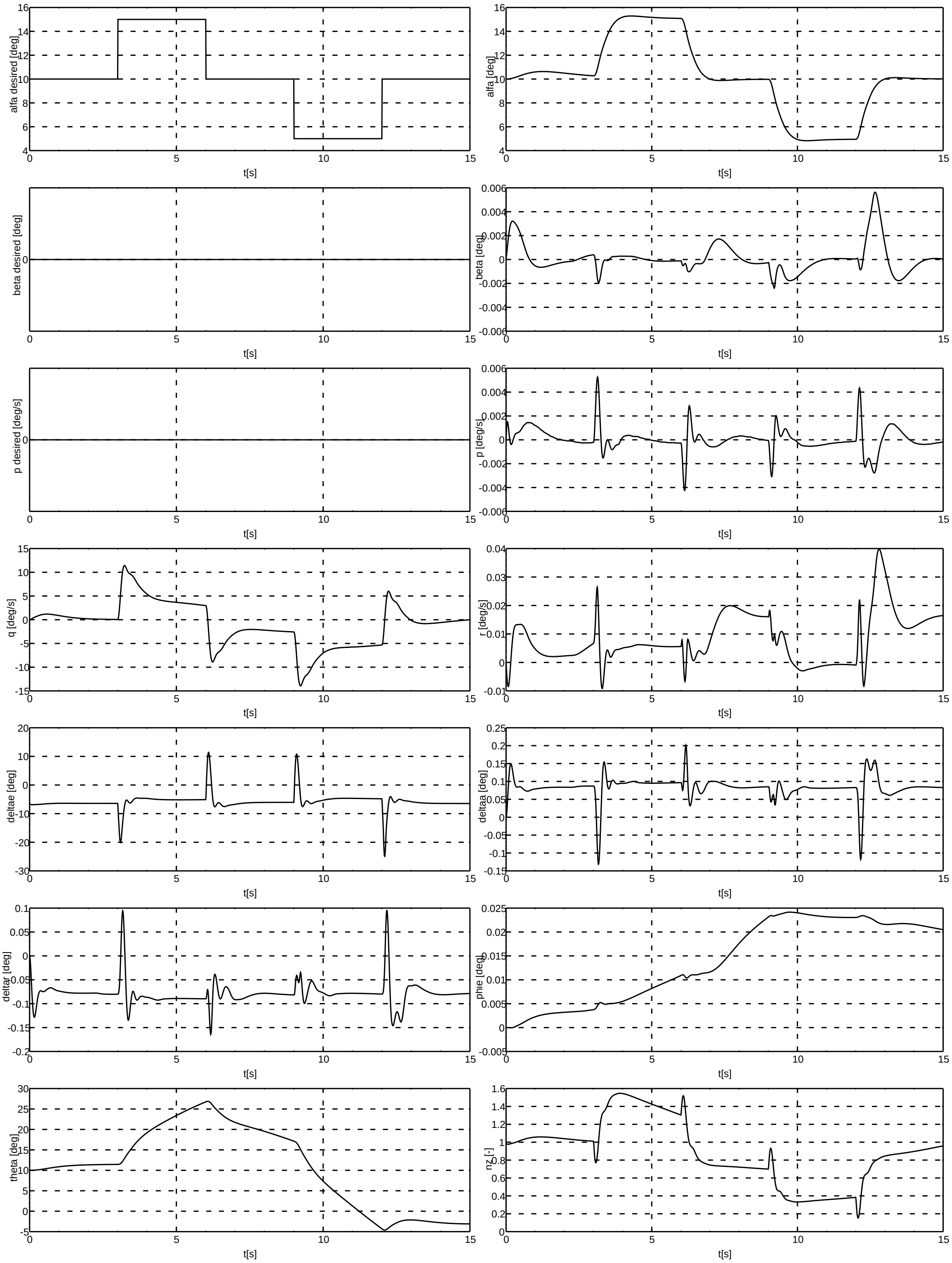

Figure 5.3: Mach number 0.21, altitude 50 m, NDI2+integrator.

There is a small overshoot even when the controller NDI2 + integrator is used, which can be seen in Figure 5.3.

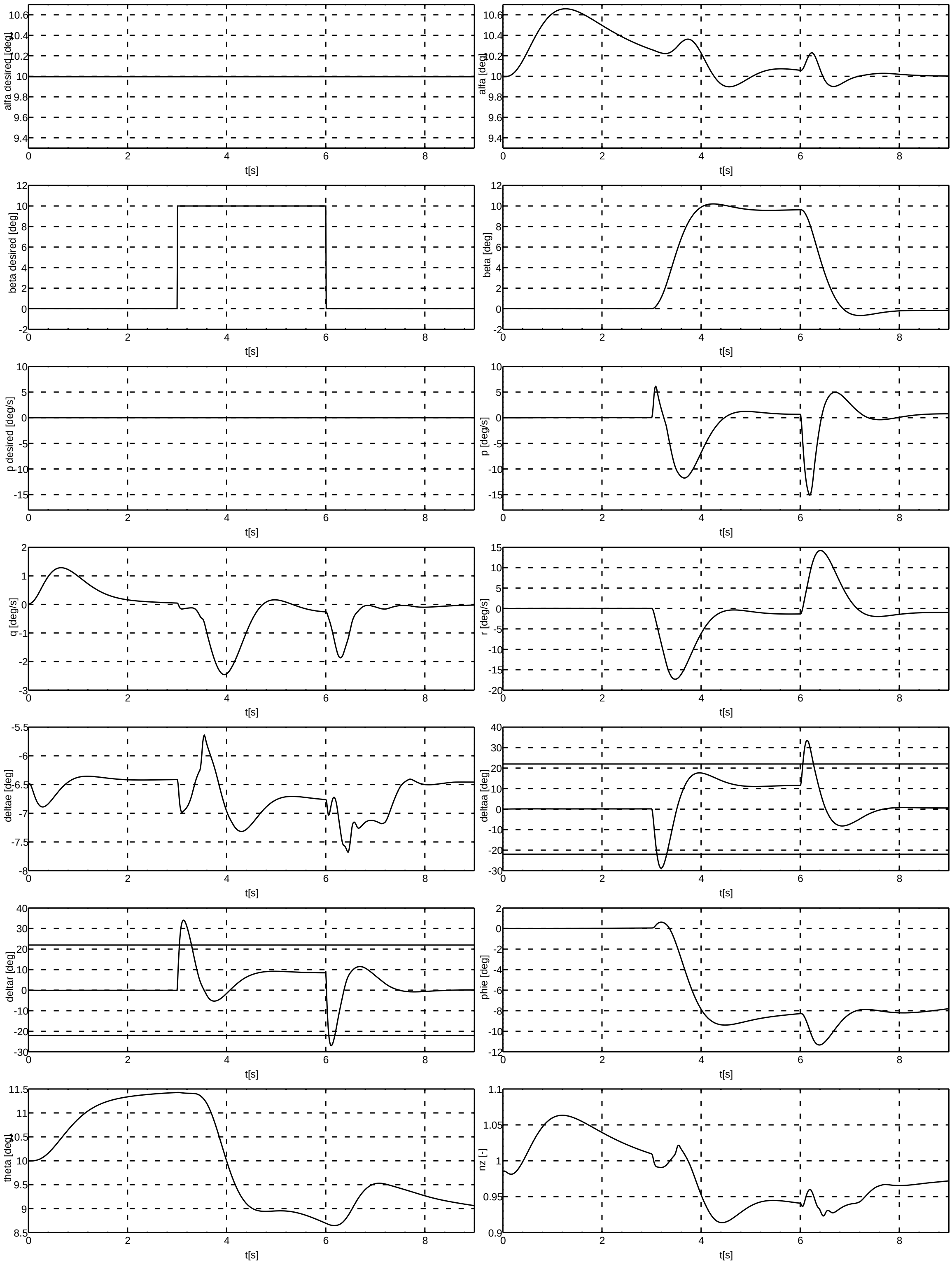

Figure 5.4: Mach number 0.21, altitude 50 m, NDI1+integrator.

In Figure 5.4 a step in β is commanded. Also the roll rate *p* is changed because of the couplings between the roll and the yaw mode. The commanded δ_a and δ_r are both greater than what is possible to obtain.

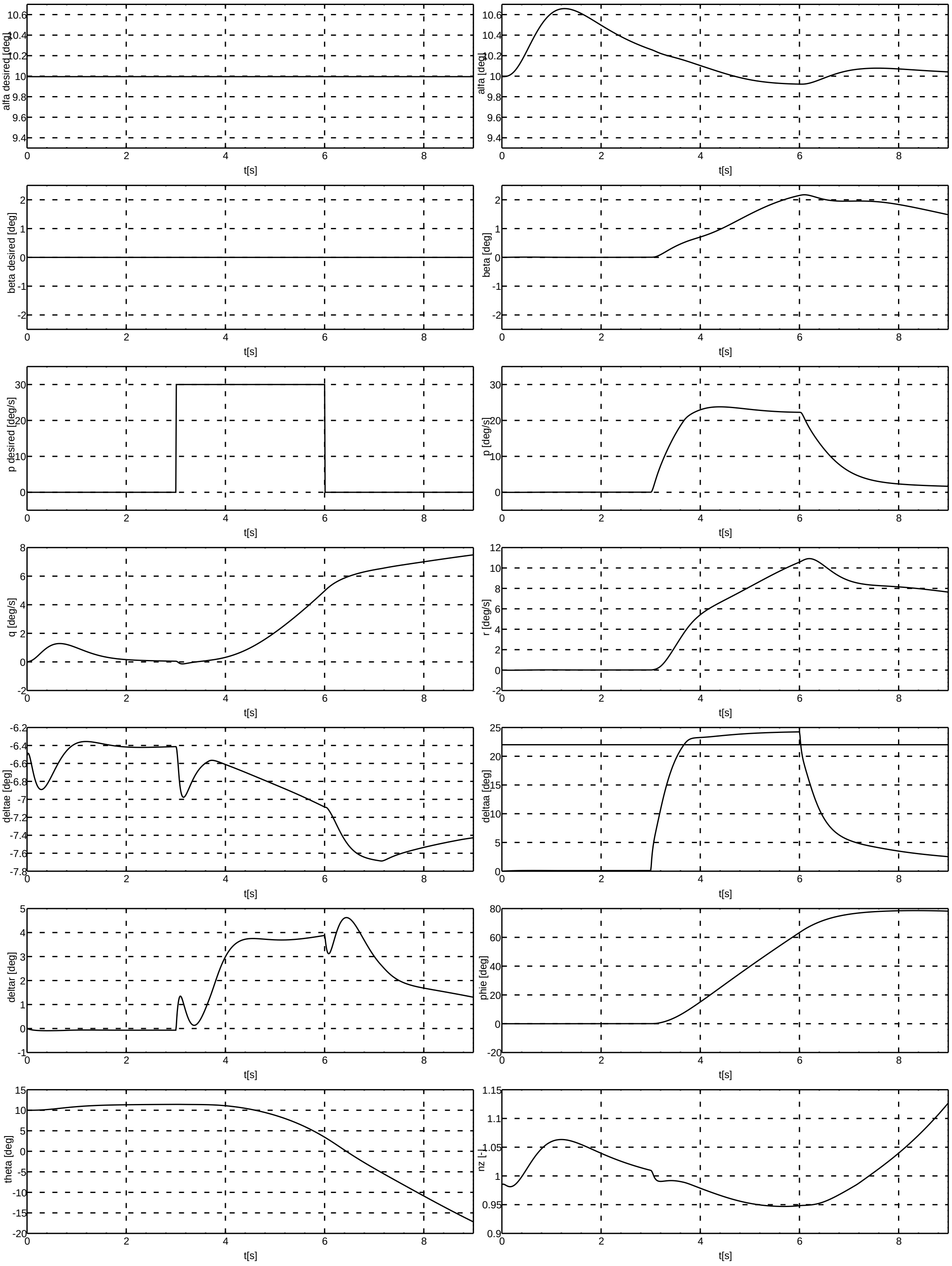

Figure 5.5: Mach number 0.21, altitude 50 m, NDI1+integrator.

In Figure 5.5 a step in the roll rate p is commanded, but the controller is not able to deliver the desired p. This depends on the control surfaces, that have reached their maximum, see the plot for δ_a. At the same time β reaches more than 2 degrees. This is difficult to avoid, because of the couplings between the roll and the yaw mode.

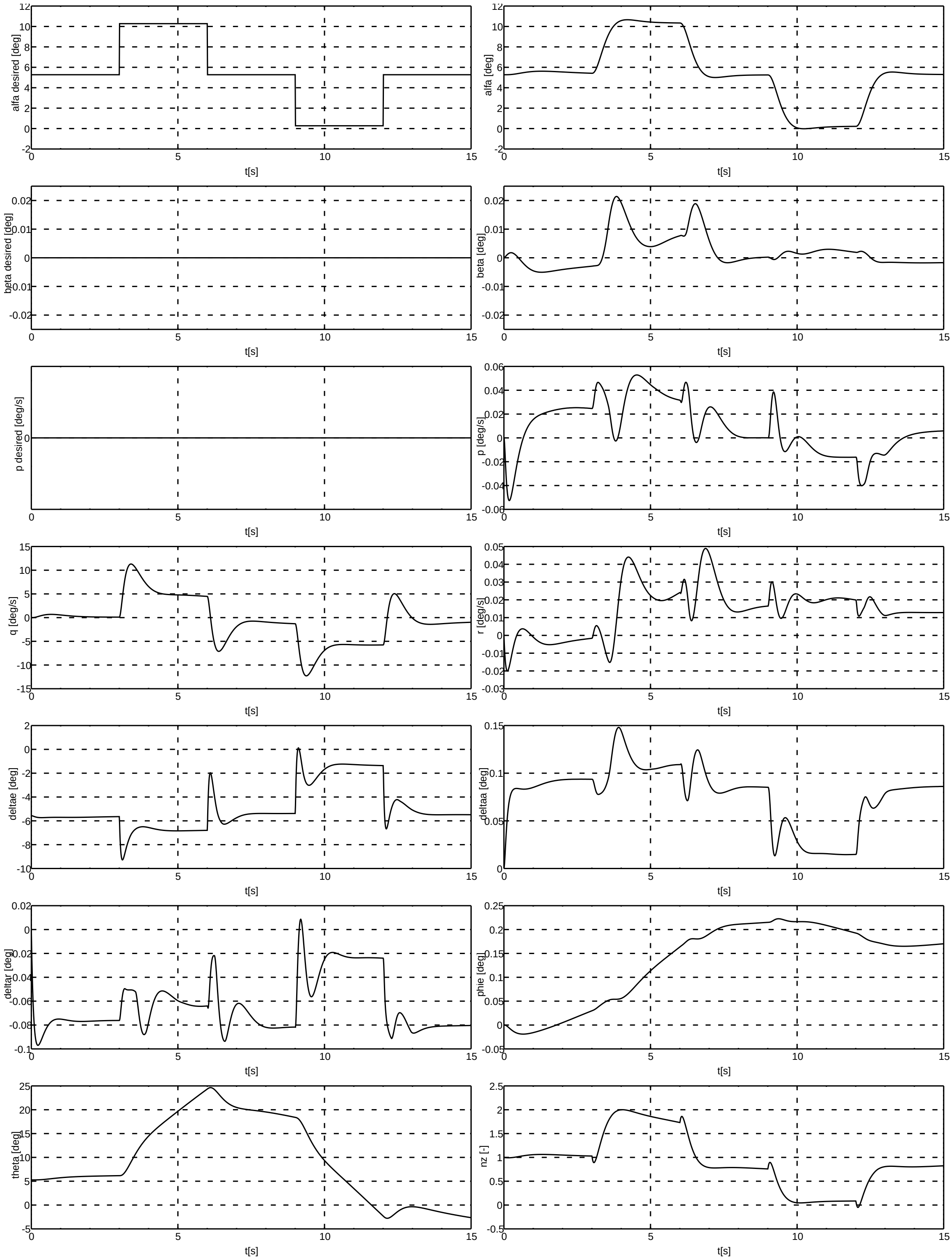

Figure 5.6: Mach number 0.3, altitude 1000 m, NDI1+integrator

In Figure 5.6, the angle of attack α follows the desired signal. The sideslip angle β and the roll rate *p* are close to zero.

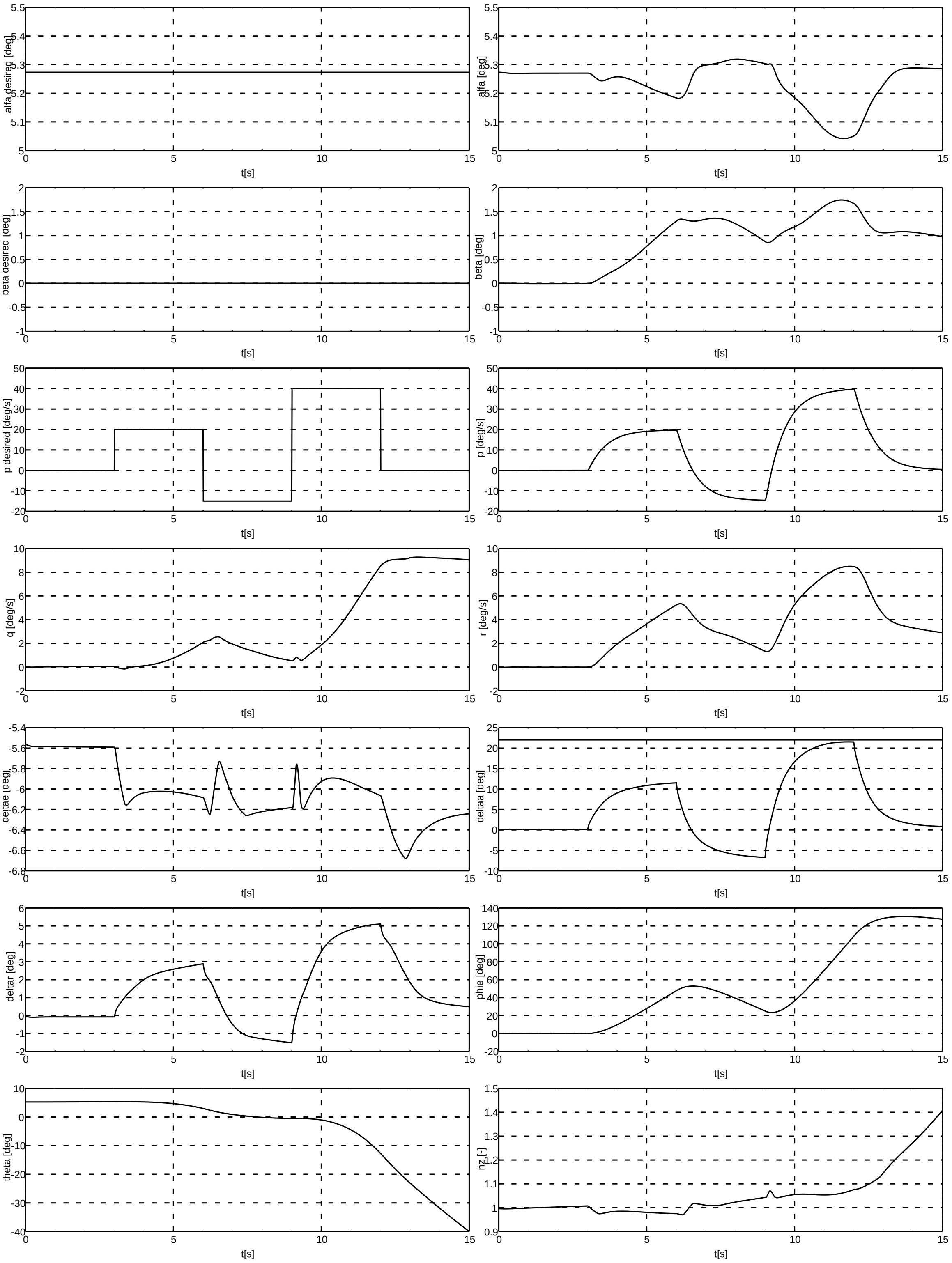

Figure 5.7: Mach number 0.3, altitude 1000 m, NDI1.

The plots in Figure 5.7 show that the roll rate *p* reaches the desired values. The control surface δ_a almost reaches its upper limit. The sideslip angle β and the angle of attack α are also affected because of the couplings.

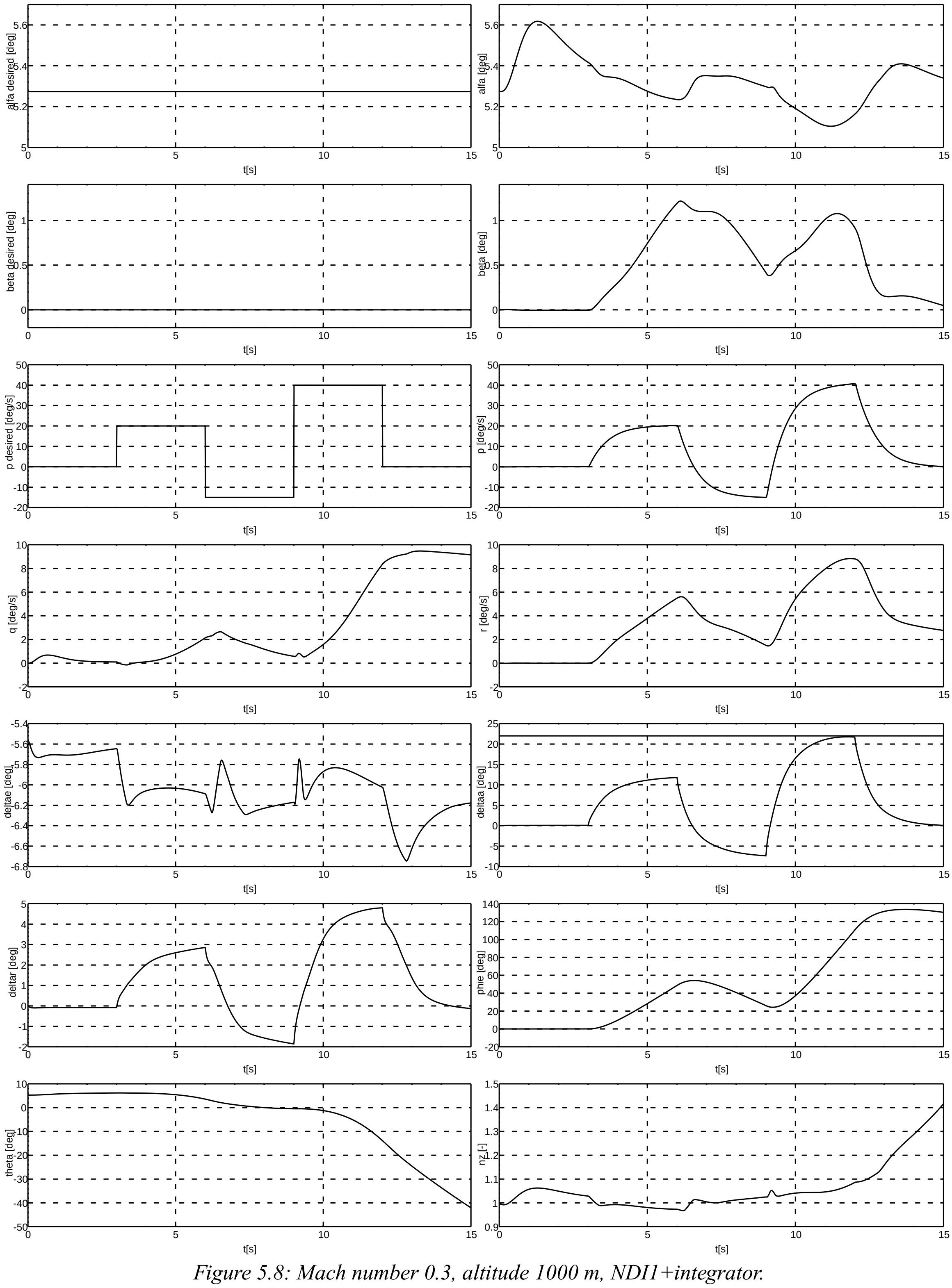

Figure 5.8: Mach number 0.3, altitude 1000 m, NDI1+integrator.

In Figure 5.8, the roll rate *p* follows the desired signal. If Figure 5.8 is compared to Figure 5.7, it can be seen that with an integrator, α and β are closer to the desired values. The control surface δ_a touches its upper limit.

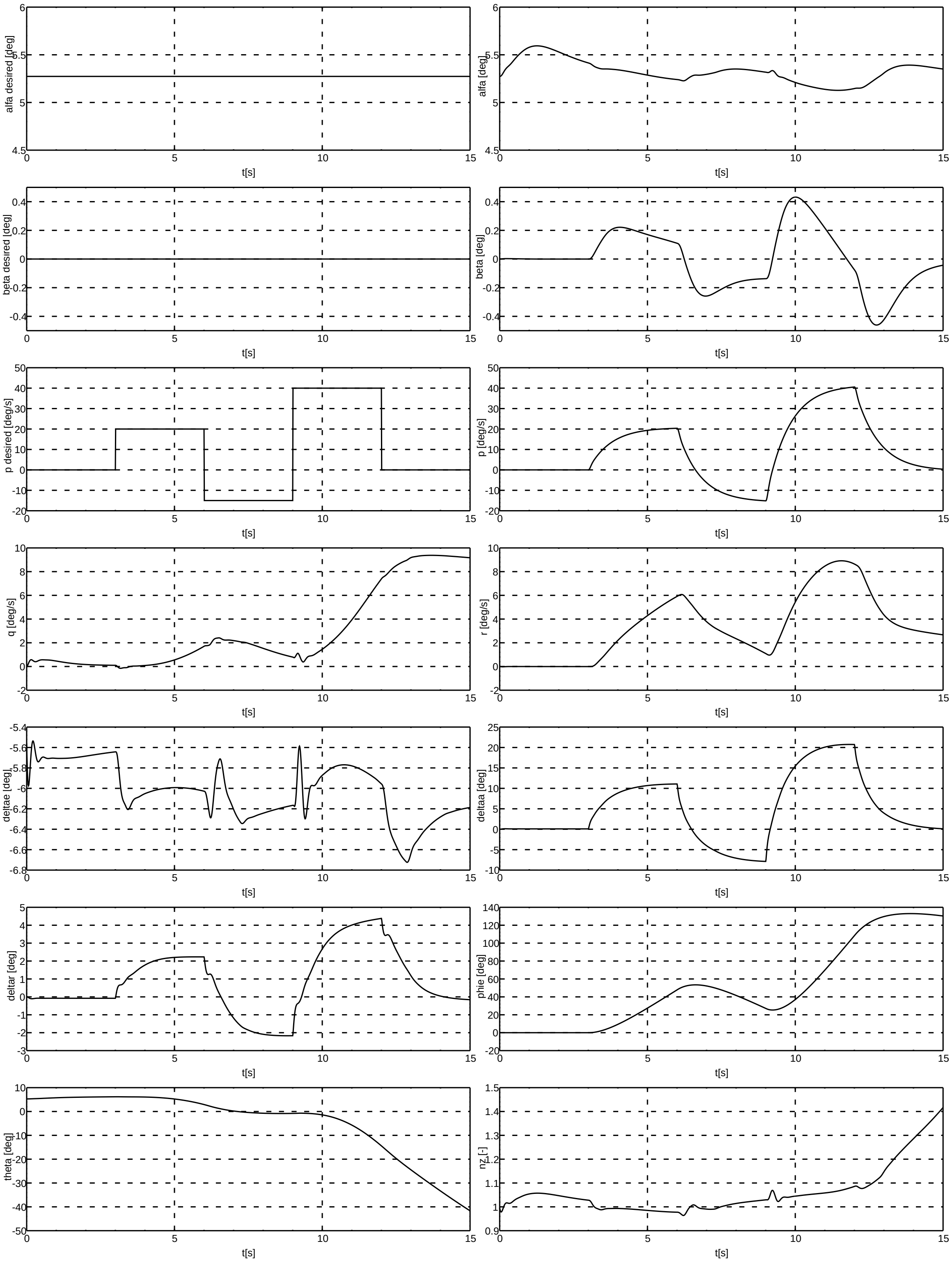

Figure 5.9: Mach number 0.3, altitude 1000 m, NDI2+integrator.

The roll rate *p* has almost the same appearance as when controller NDI1+integrator was used. The control surfaces are a little more oscillatory compared to the control surfaces in Figure 5.8.

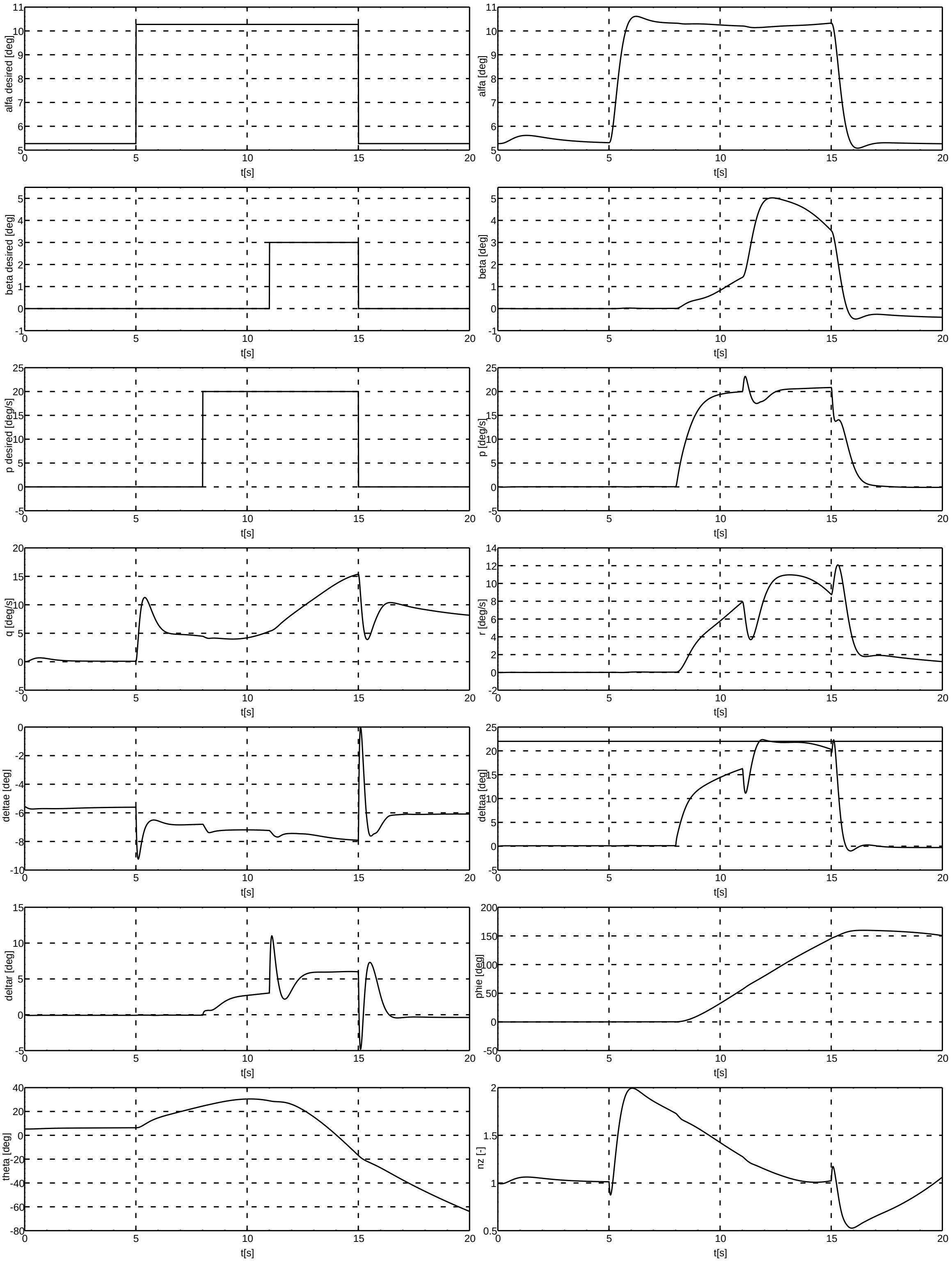

Figure 5.10: Mach number 0.3, altitude 1000 m, NDI1+integrator

In Figure 5.6, α, β, and p are commanded to some values at the same time. The result is acceptable and it is rare that all these signals are commanded at the same time. The surface δ_a touches its upper limit and β is the signal that has most difficulties in keeping the desired value.

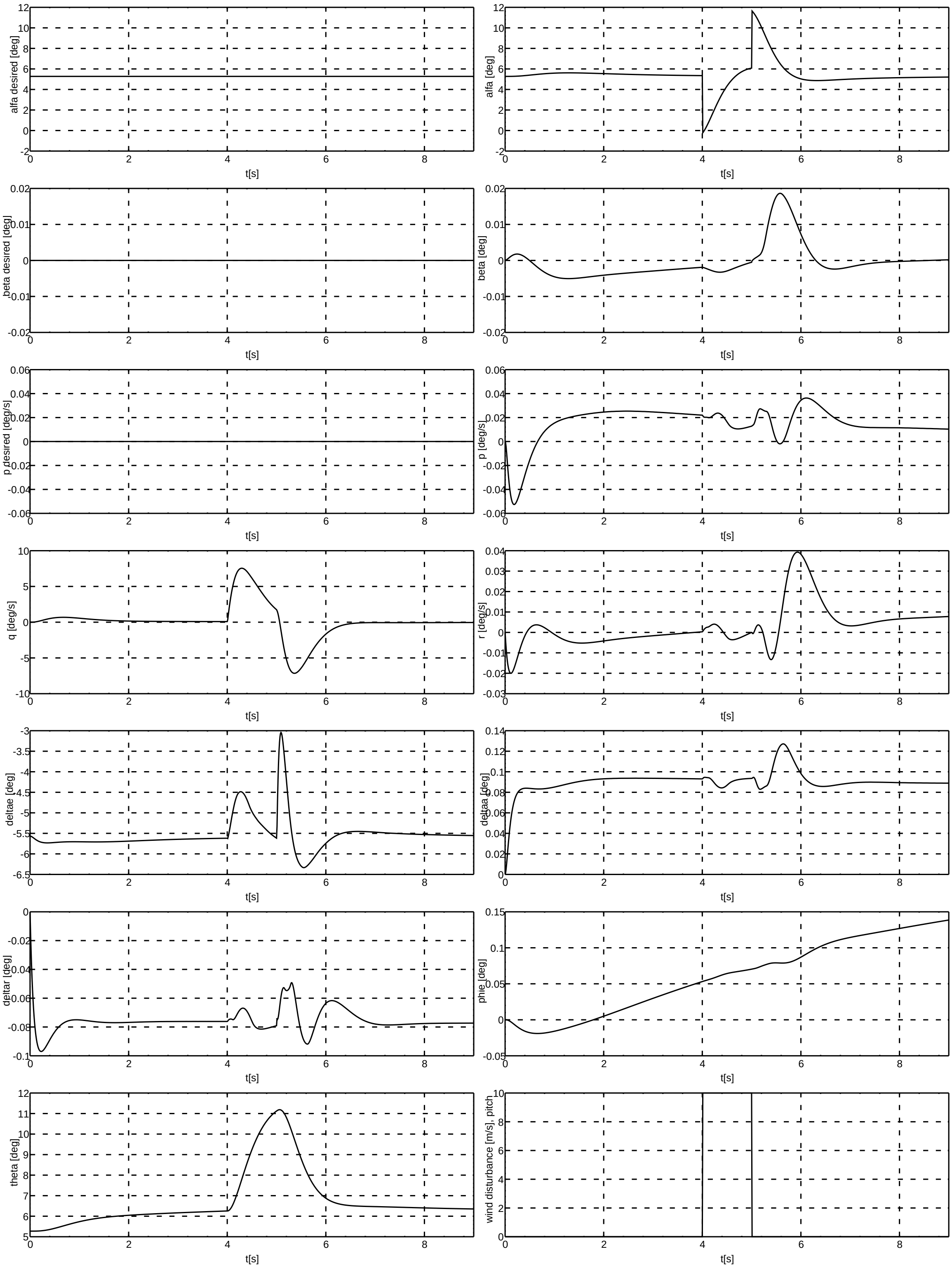

Figure 5.11: Mach number 0.3, altitude 1000 m, NDI1+integrator, wind disturbance in pitch mode.

In Figure 5.11 a wind disturbance in z_b is added, which mainly affects α.

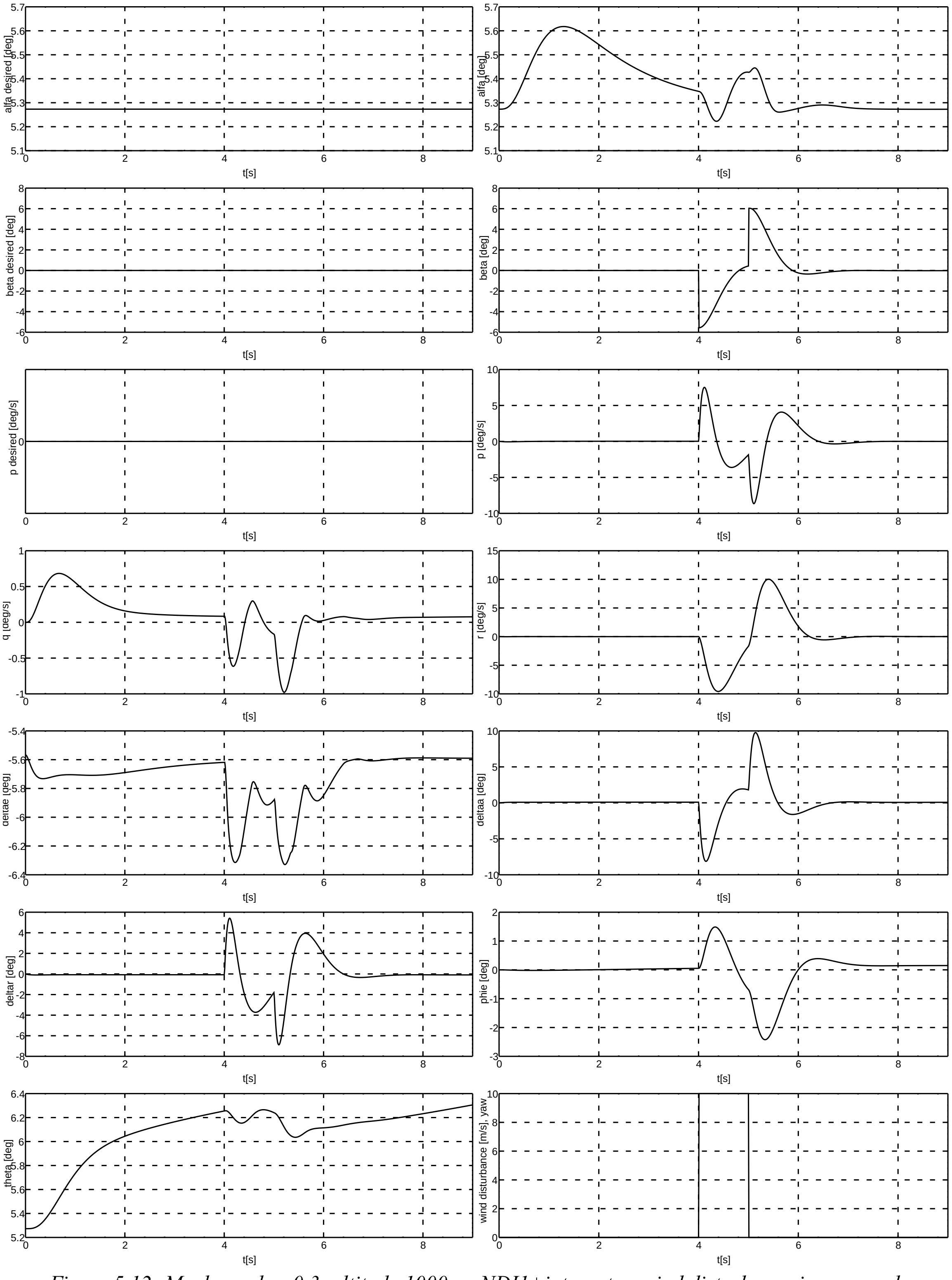

Figure 5.12: Mach number 0.3, altitude 1000 m, NDI1+integrator, wind disturbance in yaw mode.

In Figure 5.12 a wind disturbance in y_b is added. The sideslip angle β and the roll rate p are affected and even α makes a bump.

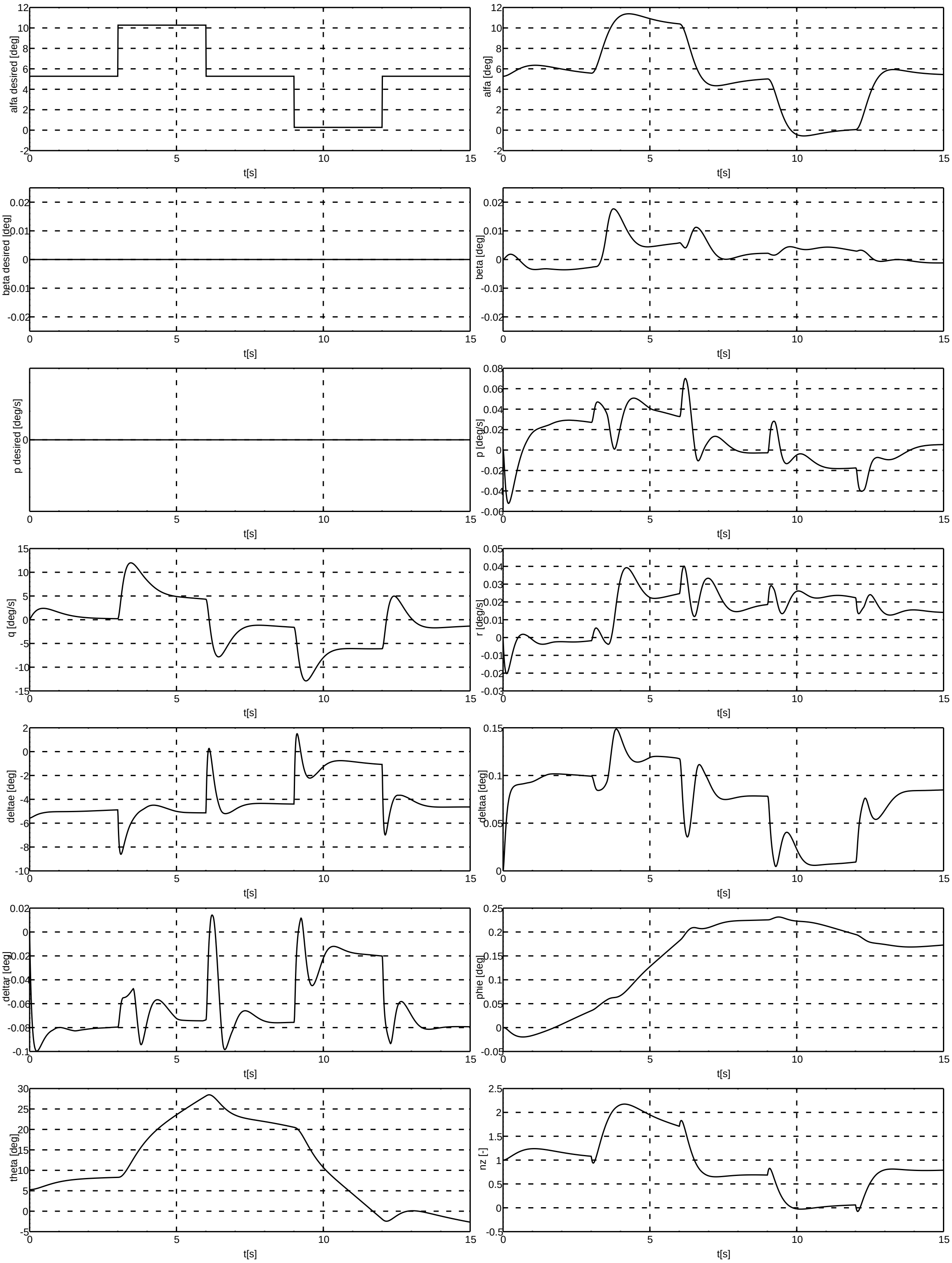

Figure 5.13: Mach number 0.3, altitude 1000 m, NDI1+integrator, model error.

In Figure 5.13 a model error is implemented that has moved the centre of gravity 0.1 m in the direction of x_b. The integrator helps the angle of attack α to reach the desired level in spite of the model error.

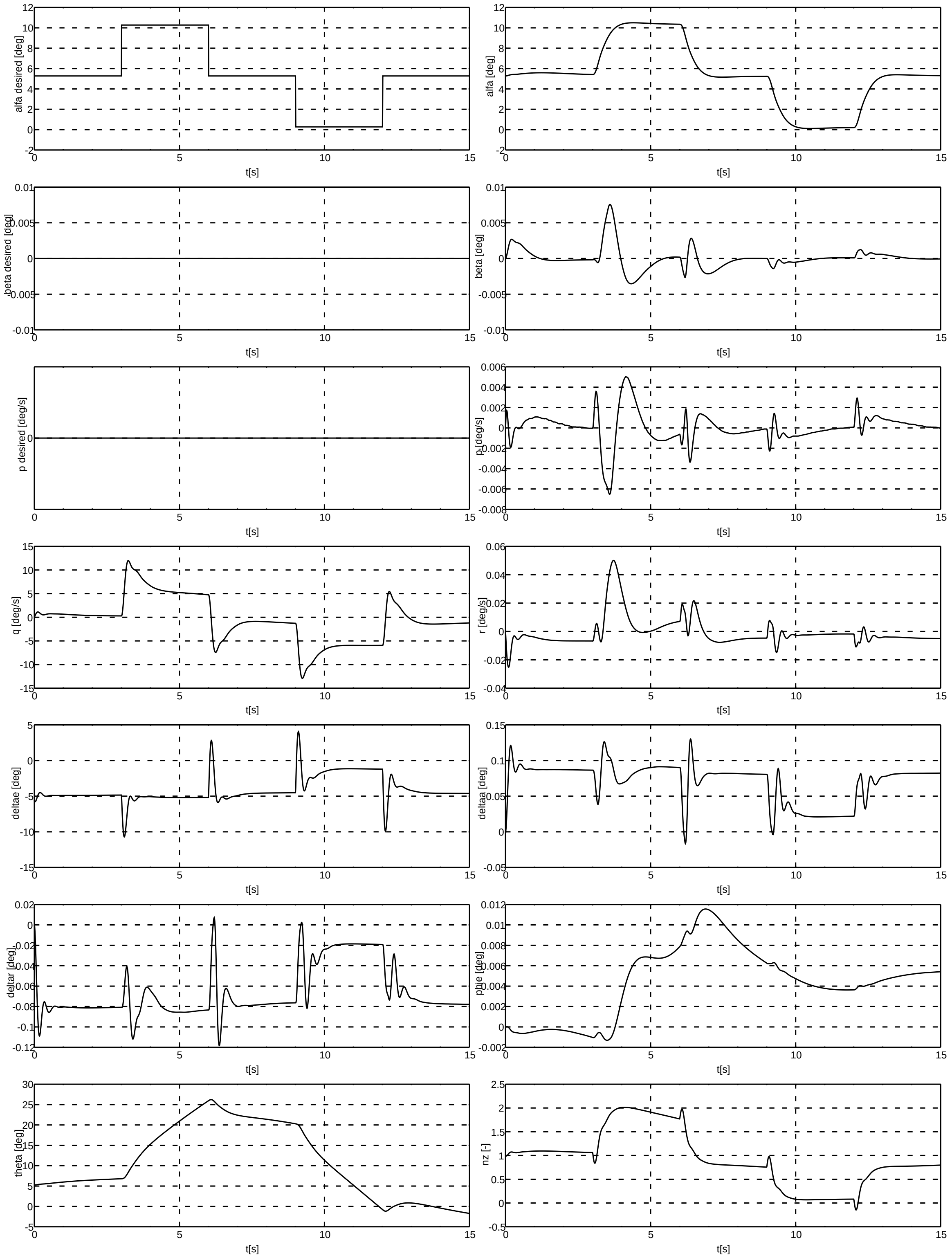

Figure 5.14: Mach number 0.3, altitude 1000 m, NDI2+integrator, model error.

In Figure 5.14 the same model error as in Figure 5.13 is implemented, but now another controller is used. If the plot for α is studied, it can be seen that this controller handles the model error better than the controller NDI1 + integrator. In Figure 5.14, it can also be seen that the signals and control surfaces are more oscillatory compared to when controller NDI1+integrator was used.

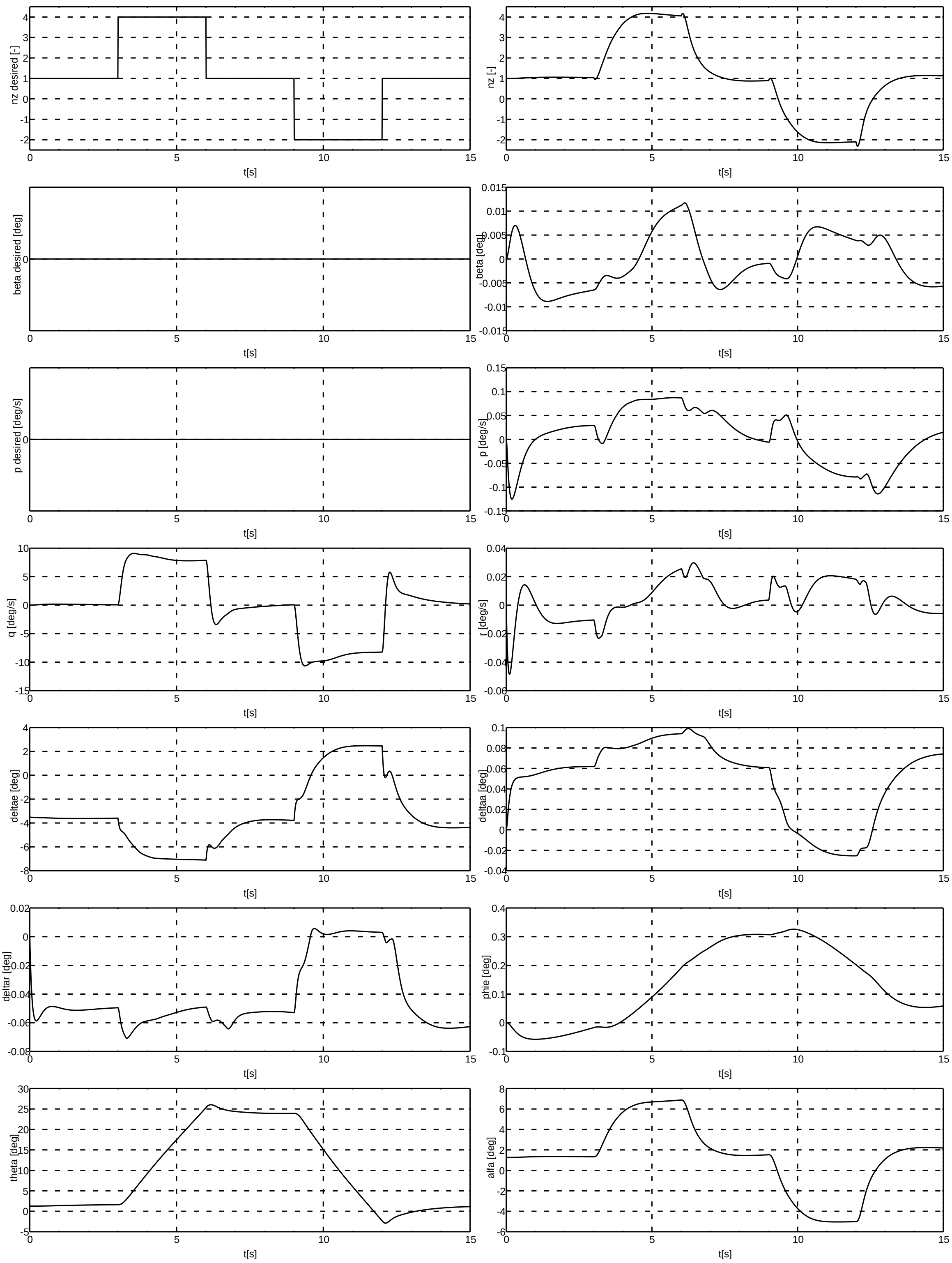

Figure 5.15: Mach number 0.7, altitude 6000 m, NDI1+integrator.

In Figure 5.15 the velocity is high and therefore the load factor n_z is controlled instead of α. The plots look good, even if the load factor n_z is not exactly as desired. This is caused by the fact that no integrator for n_z is implemented, and the expression that relates the desired n_z to the desired α consists of approximations.

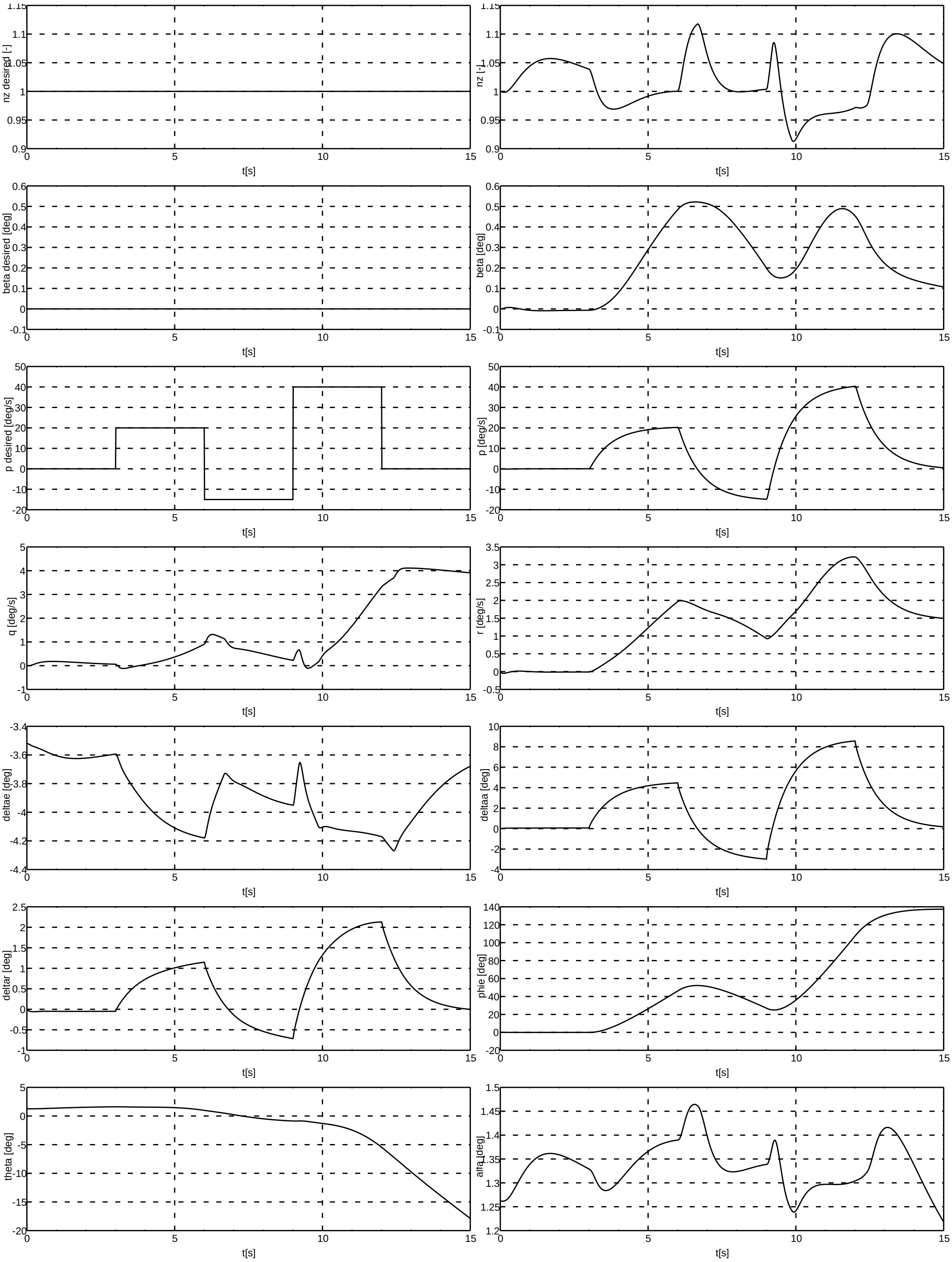

Figure 5.16: Mach number 0.7, altitude 6000 m, NDI1+integrator.

In Figure 5.15 the roll rate p follows the desired signal. The load factor n_z varies, but the output is acceptable. It is difficult to keep a constant load factor at the same time as the aircraft rolls.

The results from the plots in this chapter can now be summarized:

The controllers used during the simulations are most often able to deliver the desired signal. Some problems occur because of couplings between the longitudinal and the lateral mode.

Sometimes the limits for the control surfaces are reached. This happens for example for δ_a when the SHARC rolls at low altitude.

Because of the approximated expression that relates the desired n_z and the desired α, and that no n_z-integrator is available, there is a small divergence between the desired and the measured n_z.

When integrators are added to the both controllers, a small overshoot is noticed, but the integrators are necessary in order to get a robust system.

When the controllers NDI2 and NDI2 + integrator are used, the signals are more oscillatory, and the control surfaces change more abruptly, compared to when the controllers NDI1 and NDI1 + integrator are used.

The advantage with the controller NDI2 + integrator is that it seems to handle model errors better. It depends on the fact that NDI2 + integrator uses less information from the aerodata tables.

6 Conclusions

This master's thesis deals with three methods that all origin from Non-linear Dynamic Inversion (NDI).

The control law expressions calculated when pure NDI theory is used, are huge, especially if the servos are included in the model. If this method is to be evaluated, some approximations must be done that reduce the complexity of the expressions. For example, the derivatives of some variables that vary slowly can be approximated by zero, and the dynamics of the servos could be removed from the equations.

The controller named “Cascaded NDI Method One” that uses much information from aerodata tables, works very well according to the simulation plots. The integrators that are implemented to make the controller more robust, cause a small overshoot, but it is acceptable. The controller is also able to handle model errors. For some Mach numbers and altitudes, the SHARC is not able to roll as fast as desired. That is not a problem for the controller, but for the physical design of the SHARC. If the surfaces reach their limits, some kind of anti-wind-up technique must be implemented, which has not been done in this thesis.

The advantage with the controller named “Cascaded NDI Method Two” that uses the derivatives of some measured outputs, is that the expressions for the control signals are small. This controller also seems to handle model errors better, probably because of the fact that less information from aerodata tables is used. One disadvantage is that the method uses time derivatives of some output signals, and it is not advised to calculate derivatives if a signal is noisy. As can be seen in the simulation plots, the signals are more oscillatory and the control surfaces change more abruptly because of the derivatives, compared to the other controller. In reality, when noise is present, the output will probably not be that good.

In this thesis, only a model error in pitch has been implemented, and the integral action in roll and yaw mode have therefore not been evaluated. The result from the simulations when a model error in pitch was implemented, was good. No integrator for the load factor n_z is implemented and this area could be further researched. Other methods to obtain a robust system, could also be tested, for example the μ-synthesis.

During the simulations, the values of the parameters k_α, k_q, k_β, k_r, k_p, l_α, l_β and l_p have been the same for all flight cases, and the results were good. If it turns out to be necessary to change the parameters for some flight cases, because different performances in the envelope are wanted, scheduling can be used to avoid abrupt changes.

Bibliography

[1] Dale Enns, Dan Bugajski, Russ Hendrick and Gunter Stein. Dynamic inversion: an evolving methodology for flight control design. *International Journal of Control*, 59(1):71-91, January 1994.

[2] Torkel Glad, Lennart Ljung. *Reglerteori - Flervariabla och olinjära metoder*. Studentlitteratur, 1997.

[3] Brian L. Stevens and Frank L. Lewis. *Aircraft control and simulation*. John Wiley & sons, 1992.

[4] Bernard Etkin. *Dynamics of Atmospheric flight*. John Wiley & Sons, 1972.

[5] Ola Härkegård. Dynamic control allocation using constrained quadratic program ming. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, USA, August 2002.

[6] Marc Bodson. Evaluation of Optimization Methods for Control Allocation. *Journal of Guidance, control and dynamics*, 25(4) July-August 2002.

[7] Richard J. Adams and Siva S. Banda. An integrated approach to flight control design using dynamic inversion and μ -synthesis. *Proceedings of the American Control Conference San Francisco*, California, June 1993.

[8] Jacob Reiner, Gary J. Balas and William L. Garrard. Flight Cotrol Using Robust Dynamic Inversion and Time-scale Separation. *Automatica*, 32(11):1493-1504, 1996.

[9] S. Antony Snell, Dale F. Enns and William L. Garrard Jr. Nonlinear inversion flight control for a supermaneuverable aircraft. *Journal of Guidance, control and dynamics*, 15(4) July-August 1992.

[10] Dr Phill Smith and Andrew Berry. *Flight test experience of a non-linear dynamic inversioncontrol law on the VAAC Harrier*, AIAA-2000-3914

[11] Richard J. Adams, James M. Buffington and Siva S. Banda. Design of Nonlinear Control Laws for High-Angle-of Attack Flight. J*ournal of Guidance, Control and Dynamics*, 17(4) July-August 1994.

[12] Robert C. Nelson. *Flight stability and automatic control. McGraw-Hill*, second edition, 1998.

Appendix A

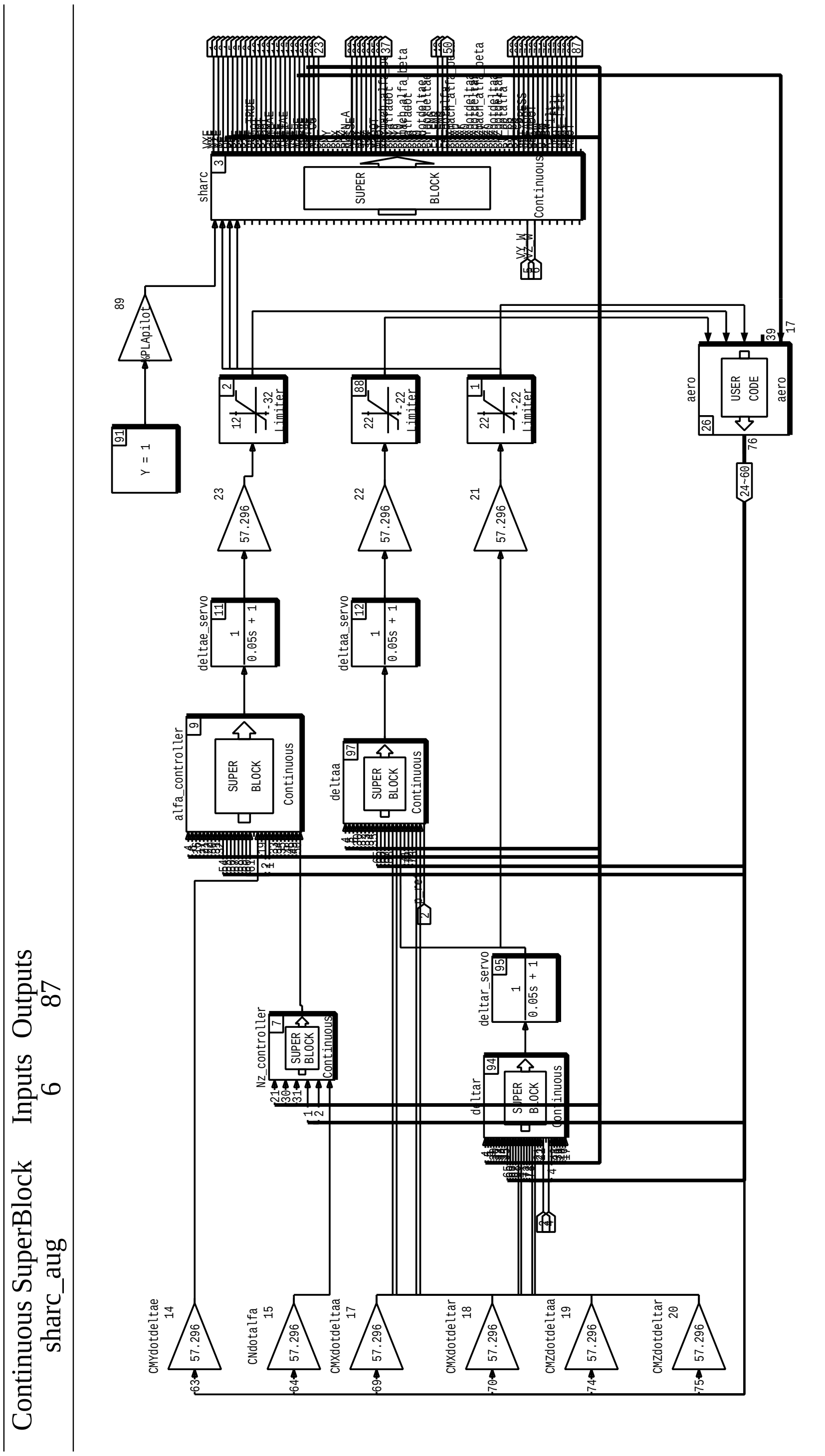

Figure A.1: The Model of the SHARC and the controllers.

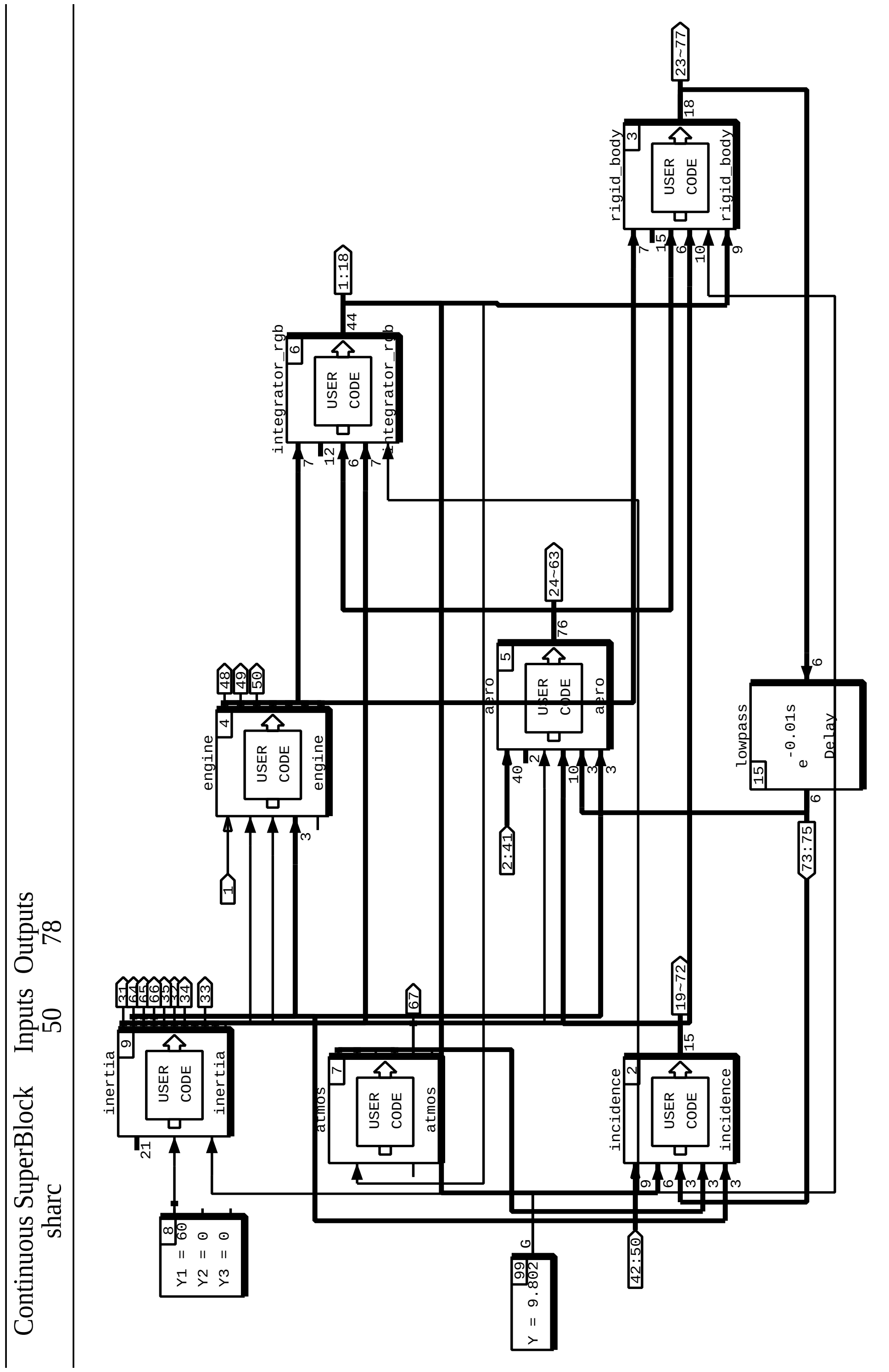

Figure A.2: The model of the SHARC.

Appendix B

The control signals, when the integrators for α, β, and p are added to the controller named “Cascaded NDI Method One”, are presented below. The signal with the subindex “dn” is the filtered desired signal, that can be viewed in Figure 4.4.

$$\delta_e = \Big(\frac{d}{d\delta_e}C_m(\delta_e)\Big)^{-1} \cdot \Big(\Big(k_q I_y\Big(k_\alpha(\alpha_d - \alpha) + l_\alpha\big(\int(\alpha_{dn} - \alpha)dt\big) +$$

$$(p\cos\alpha + r\sin\alpha)\tan(\beta) -$$

$$\frac{1}{mV_T\cos\beta}(-F_T\sin\alpha + mg(\cos\alpha\cos\theta\cos\phi + \sin\alpha\sin\theta) - q_a S \cdot$$

$$(-C_T\sin(\alpha) + C_N\cos(\alpha))) - q\Big) - I_{zx}(r^2 - p^2) - (I_z - I_x)rp\Big) \cdot$$

$$\frac{1}{q_a S\bar{c}} - C_m(M, \alpha, \beta) - C_m(\dot{\alpha}) - C_m(q) - C_{m0}\Big)$$

$$\delta_a = \frac{1}{q_a Sb\left(\frac{dC_l}{d\delta_a}(\delta_a) + \frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}(\delta_a)\right)}\Bigg((k_p(p_{dn} - p) + l_p\big(\int(p_{dn} - p)dt\big))\left(1 - \frac{I_{zx}^2}{I_z I_x}\right)I_x$$

$$-q_a Sb\Big(C_l(M, \alpha, \beta) + C_l(p) + C_l(r) + \frac{d}{d\delta_r}C_l(\delta_r)\delta_r + C_{l0}\Big) - (I_y - I_z)qr$$

$$-I_{zx}\Bigg(\frac{q_a Sb\Big(C_n(M, \alpha, \beta) + C_n(r) + \frac{d}{d\delta_r}C_n(\delta_r)\delta_r + C_{n0}\Big) - I_{zx}qr + (I_x - I_y)pq}{I_z} +$$

$$pq\Bigg)\Bigg)$$

$$\delta_r = \frac{1}{q_a Sb}\left(\frac{-\frac{dC_l}{d\delta_r}-\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_r}}{\frac{dC_l}{d\delta_a}+\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}}+\frac{\frac{dC_n}{d\delta_r}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_r}}{\frac{d}{d\delta_a}C_n+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}}\right)^{-1}\cdot\Bigg(\frac{1}{\left(\frac{d}{d\delta_\alpha}C_n+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_\alpha}\right)}\cdot\Bigg(\Bigg(\Bigg(1$$

$$-\frac{I^2{}_{zx}}{I_x I_z}\Bigg)I_z k_r\Bigg(\Bigg(p\sin(\alpha)-k_\beta(\beta_{ref}-\beta)+l_\beta\left(\int(\beta_{dn}-\beta)dt\right)+$$

$$\frac{1}{mV_T}(q_a SC_Y - F_T\cos(\alpha)\sin(\beta)+mg_3)\Bigg)\Big/\cos(\alpha)-r\Bigg)$$

$$-q_a Sb(C_n(M,\alpha,\beta)+C_n(r)+C_{n0})$$

$$-I_{zx}\left(\frac{(q_a Sb(C_l(M,\alpha,\beta)+C_l(p)+C_l(r)+C_{l0})+I_{zx}pq+(I_y-I_z)qr)}{I_z}-qr\right)$$

$$-(I_x-I_y)pq\Bigg)-\frac{1}{\left(\frac{dC_l}{d\delta_a}+\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}\right)}\Bigg((k_p(p_d-p)+l_p\left(\int(p_{dn}-p)dt\right))\left(1-\frac{I^2_{zx}}{I_z I_x}\right)I_x$$

$$-q_a Sb(C_l(M,\alpha,\beta)+C_l(p)+C_l(r)+C_{l0})-(I_y-I_z)qr$$

$$-I_{zx}\left(\frac{(q_a Sb(C_n(M,\alpha,\beta)+C_n(r)+C_{n0})-I_{zx}qr+(I_x-I_y)pq)}{I_z}+pq\right)\Bigg)\Bigg)$$

Using the controller named “Cascaded NDI Method Two”, the control signals when the integrators are added are:

$$\delta_{ed} = ((k_\alpha(\alpha_d - \alpha) + l_\alpha(\int(\alpha_{dn} - \alpha)dt) - \dot{\alpha}_s)k_q - \dot{q}_s)\frac{I_y}{q_a Sc\frac{dC_m}{d\delta_e}} + \delta_{es}$$

$$\delta_{ad} = \frac{1}{\frac{dC_l}{d\delta_a} + \frac{I_{zx}}{I_z}\frac{d}{d\delta_a}C_n}\left(\frac{k_p(p_{dn} - p) + l_p(\int(p_{dn} - p)dt) - \dot{p}_s}{q_a Sb}\left(1 - \frac{I_{zx}^2}{I_x I_z}\right)I_x\right.$$
$$\left. + \delta_{as}\left(\frac{dC_l}{d\delta_a} + \frac{I_{zx}}{I_z}\frac{d}{d\delta_a}C_n\right) - (\delta_{rd} - \delta_{rs})\left(\frac{dC_l}{d\delta_r} + \frac{I_{zx}}{I_z}\frac{d}{d\delta_r}C_n\right)\right)$$

$$\delta_{rd} = \frac{1}{\left(\frac{-\frac{dC_l}{d\delta_r}-\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_r}}{\frac{d}{d\delta_a}C_l+\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}}+\frac{\frac{dC_n}{d\delta_r}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_r}}{\frac{d}{d\delta_a}C_n+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}}\right)}\left(\left(\frac{1}{\left(\frac{dC_n}{d\delta_a}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}\right)}\left(\right.\right.\right.$$

$$\left(\frac{k_\beta(\beta_d-\beta)+l_\beta(\int(\beta_{dn}-\beta)dt)-\dot{\beta}_s}{\cos(\alpha)}\cdot(-k_r)-\dot{r}_s\right)\cdot\frac{\left(1-\frac{I_{zx}^2}{I_xI_z}\right)I_z}{q_aSb}$$

$$\left.\left.+\left(\frac{dC_n}{d\delta_a}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_a}\right)\delta_{as}+\left(\frac{dC_n}{d\delta_r}+\frac{I_{zx}}{I_x}\frac{dC_l}{d\delta_r}\right)\delta_{rs}\right)-\frac{1}{\left(\frac{dC_l}{d\delta_a}+\frac{I_{zx}}{I_x}\frac{dC_n}{d\delta_a}\right)}\cdot\left((k_p(p_d-p)\right.\right.$$

$$\left.\left.+l_p(\int(p_{dn}-p)dt)-\dot{p}_s)\frac{\left(1-\frac{I_{zx}^2}{I_xI_z}\right)I_x}{q_aSb}+\left(\frac{dC_l}{d\delta_a}+\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_a}\right)\delta_{as}+\left(\frac{dC_l}{d\delta_r}+\frac{I_{zx}}{I_z}\frac{dC_n}{d\delta_r}\right)\delta_{rs}\right)\right)$$