Controller Design for an Unmanned Reconnaissance Aerial Vehicle

MARCUS HAMMAR

KTH Signals
Sensors and Systems

Master's Degree Project
Stockholm, Sweden 2006

XR-EE-RT2006:001

Abstract

The overall objective with this thesis is to evaluate a method for designing nonlinear controllers in aircrafts with concern in robustness against modelling insecurities and also to minimize the design effort. The later objective is of great importance since there exists, in the industry, an ambition to automatize the design for as far extend as possible. The nonlinear method, State Dependent Riccati Equation (SDRE), used in this thesis is a nonlinear version of classic LQ design and both are evaluated and compared for a few flying conditions. Also another nonlinear control method, Two Timescale Separation (TSS), is tested. LQ and SDRE shows equal performance during both looping and more complicated maneuvers, such as high angle of attack wind-vector roll. Further it is possible to automatize the LQ design as well as it is possible for SDRE. Still SDRE is preferable since it will always be somewhat more accurate than LQ. A comparison with the nonlinear method Two Timescale Separation shows results in favor of LQ/SDRE mostly due to relatively slow dynamics and bad accuracy of TSS. A Monte Carlo simulation is made on LQ and SDRE showing that the model is robust against relatively large modelling error of 40%.

Contents

Acknowledgments

First I would like to thank FOI and especially my supervisor Dr. John W.C. Robinson. I am also grateful to Prof. Bo Wahlberg, at KTH, for being my examiner and several other teachers, for example Dr. Dan Borglund at the department of aeronautics. Further, my parents, deserve gratitude for their never ending support during all my life.

Chapter 1

Introduction

1.1 Background

In modern aircraft design it is vital to quickly be able to develop a controller that will give dynamics which are reasonably close to the desired dynamics in order to gain sufficient understanding of various aspects of the proposed aircraft concept. The main focus is normally on maneuver- and control properties, including handling qualities and actuator effectiveness, properties which are commonly appraised via simulations. To be able to design a controller at an early stage in an aircraft design is especially important in the design of modern fighters and unmanned aerial vehicles (UAVs) since such aircraft can be aerodynamically unstable, at certain speeds, in both pitch and yaw. This means that even basic maneuvering properties cannot be evaluated without the existence of a stabilizing controller.

Aircraft dynamics are most often described (without concern of aero elastic effects) by Newton-Euler's nonlinear rigid body equations, formulated in a vehicle fixed coordinate system. The forces and moments that act on the body, and which can be altered by the control surfaces, are nonlinear functions of the states in the equations of motion. This makes it natural to use nonlinear control theory in the design of controllers. Key aspects of concern are then stability, robustness and performance. In the present work we concentrate on these aspects for the State Dependent Riccati Equation nonlinear design method applied to a realistic model of an UAV which is similar to the SAAB UAV demonstrator Shark, see Figure 1.1.

1.2 Notation and Nomenclature

1.2.1 Mathematics

Vectors and matrices are denoted with bold face letters like $\mathbf{x}$ and $\mathbf{A}$, and the components of a vector or matrix are marked with ordinary letters like

Figure 1.1: The SAAB UAV demonstrator SHARC.

x_2 and A_{13}. Vectors are considered as column vectors in matrix-vector algebraic operations and the transpose of a vector $\mathbf{x} \in \mathbb{R}^n$ is marked as $\mathbf{x}^T$. The identity matrix $\mathbf{I} \in \mathbb{R}^{n \times n}$ is written without explicitly indicating the dimension n. The (one-dimensional) subspace of $\mathbb{R}^n$ spanned by a vector $\mathbf{u} \in \mathbb{R}^n$ is denoted $[\mathbf{u}]$. For the projection of a vector $\mathbf{w} \in \mathbb{R}^n$ onto $[\mathbf{u}]$ we use the notation $P_{[\mathbf{u}]}(\mathbf{w})$. In general we will suppress the time dependence on quantities. If $\mathbf{u}$ is a (time differentiable) vector the time derivative $d\mathbf{u}/dt$ of $\mathbf{u}$ at time t is marked using use the standard dot notation as $\dot{\mathbf{u}}$. For a differentiable function $\mathbf{y} : \mathbb{R}^n \to \mathbb{R}^m$ the Jacobian matrix $\mathbf{T}(\mathbf{x})$ at $\mathbf{x}$, with entry $T_{j,k}(\mathbf{x})$ given by $T_{j,k}(\mathbf{x}) = \partial y_j / \partial x_j(\mathbf{x})$, the notation $\mathbf{T} = \partial \mathbf{y}(\mathbf{x})/\partial \mathbf{x}$ is used. SI units are used throughout this report.

1.2.2 Flight Mechanics

The flight mechanical nomenclature is standard and summarized in Table 1.1. It is explained further in e.g. [3]. In particular, the aerodynamical forces and moments acting on the vehicle are described, in accordance with standard practice, in terms of the aerodynamical (dimensionless) coefficients C_x, C_Y, C_Z (for forces) and C_l, C_n, C_m (for moments), respectively.

1.3 Control Methods

1.3.1 Classic LQ

Linear Quadratic (LQ) design [1] is a classic control design method which is commonly used in the design of control systems for aircraft such as the JAS 39 Gripen [2]. LQ is a linear method which, when applied to nonlinear systems via linearization, guarantees local stabilizing properties. This means that the feedback gains are precomputed for a number of points in the flight envelope, around which the aircraft dynamical equations are linearized, and the desired gain is obtained in-flight by gain-scheduling, i.e. an interpolation

procedure. This precalculation of feedback gains is the main drawback of LQ since it is normally required to calculate the gains in a large number of points in the envelope and at each such point a "tuning" process has to be executed in order to obtain the desired dynamic behaviour. Further, if there are changes made to the aircraft during the design or if the aerodynamical properties are changed by e.g. attaching loads to the aircraft, new feedback gains have to be calculated. The aerodynamical properties can be altered for example by attaching external loads (e.g. fuel tanks or weapons) or fitting sensor equipment. In State Dependent Riccati Equation (SDRE) described below this precalculation of feedback gains can be avoided.

1.3.2 SDRE

The SDRE concept is as old as LQ but it was first relatively recently that for example [4], Cloutier, D'Souza and Mracek addressed SDRE design methodologies and optimality, suboptimality and stability properties of SDRE nonlinear regulation in a more systematic way. Since then a number of papers have been published on the SDRE method covering theory [7, 8], numerics [5] and applications, in particular aerospace applications [6, 9].

To illustrate the idea behind SDRE we start by considering a controlled system of nonlinear ordinary differential equations on the standard affine form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{1.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the control vector, $\mathbf{f}$ is a smooth vector field in $\mathbb{R}^n$ with $\mathbf{f}(0) = 0$ and $\mathbf{g}$ is a continuous matrix function on $\mathbb{R}^n$ with values in $\mathbb{R}^{n\times m}$ The object is to stabilize the equilibrium at 0 present for the uncontrolled system ($\mathbf{u} = 0$). The aircraft equations of motion can be presented in this form (1.1), where $\mathbf{u}$ represents the forces and moments generated by various control surfaces. In SDRE the key idea is to rewrite the vector field $\mathbf{f}$ in (1.1) into a so called State Dependent Coefficient (SDC) form

$$\mathbf{f}(\mathbf{x}) = \mathbf{A}(\mathbf{x})\mathbf{x}. \tag{1.2}$$

The SDC is an exact representation of the uncontrolled nonlinear dynamics which at the same time mimics the structure of the linearized version of the dynamics. The difference between LQ and SDRE is that the system matrix $\mathbf{A}$ is a function of $\mathbf{x}$ in the SDRE case. An analogous "factorization" is done for $\mathbf{g}(\mathbf{x})$ and the entire system (1.1) can then be written on the SDC form

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})\mathbf{u} \tag{1.3}$$

where $\mathbf{B}(\mathbf{x}) = \mathbf{g}(\mathbf{x})$. The factorization for $\mathbf{g}$ is trivial but for $\mathbf{f}$ there exist several solutions. This is explained in more detail below.

The fact that the new representation $\mathbf{A}(\mathbf{x})\mathbf{x}$ of $\mathbf{f}(\mathbf{x})$ in (1.2) is a function of the state means that if one solves the Riccati equation associated with the

corresponding LQ problem online in each time step the resulting feedback gains also will be a function of the state, obtained online. Hence no precalculation of gains and gain-scheduling is needed. Also, since the feedback gain is recalculated in each timestep, the penalty matrices $\mathbf{Q}$ and $\mathbf{R}$ occurring in LQ for the state and control signal, respectively, can be functions of the state and can be tuned for further improvement in the design.

To implement an SDRE controller in practice puts high requirements on the hardware since it has to perform the calculations in real time. This is also the reason SDRE has not been tested in practical applications until recently. In [9], SDRE is used for a missile flight control application with sample rates up to 2 kHz. This is performed using off-the-shelf processors and still managing to solve the Riccati equation in real time. This requirement on the hardware may be a drawback when compared with other control methods. Another disadvantage in SDRE is that it stabilizes the vectorfield around equilibrium points which have to be, as for LQ, precalculated.

1.3.3 TSS

The method of nonlinear dynamic inversion via two-timescale separation [1] (TSS) emerged as a variant of nonlinear dynamic inversion in the 1980's [11] and was by the early 1990's an established technique [12, 13], at least in the research community.

In the basic form of TSS the system in (1.1) is divided into two parts, the (new) $\mathbf{x}$ and the $\mathbf{y}$-system, of equal dimensions, according to

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{y}, \tag{1.4}$$

$$\dot{\mathbf{y}} = \mathbf{h}(\mathbf{y}) + \mathbf{k}\mathbf{u}. \tag{1.5}$$

Here, the $\mathbf{x}$-system represents the velocity states, the "slow" subsystem, and the $\mathbf{y}$-system represents the angular velocity states, the "fast" subsystem. From (1.4), (1.5) we see that the fast state vector $\mathbf{y}$ act as a "virtual control," which controls the slow states via the term $\mathbf{g}(\mathbf{x})\mathbf{y}$, and the dynamics for $\mathbf{y}$ itself are controlled by the original input $\mathbf{u}$.

In our applications below we are, for some control formulations, going to use a slightly different version of TSS where a feedforward term (from the control directly to the fast states) is present. Both of these variants of TSS are explained further in Appendix B. The TSS controllers are used as comparison when evaluating the results obtained using SDRE and LQ below.

[1] A readable introduction to the ideas behind the TSS method can be found in [10].

1.4 Robustness

One of the major problems in the design of flight control systems is modelling uncertainties and parameter variations for the aircraft and its operating environment. In [14] this problem is addressed thoroughly and it is demonstrated using a number of examples how robust flight control systems can be developed for realistic problems. Methods similar to those described in [15] will be used to assess robustness properties in the present work.

1.5 Scope of the study

The purpose of this work is not to design a full scale flight control system for the entire flight envelope of an aircraft. Instead, the aim is to investigate how SDRE compares to other methods regarding maneuvering and robustness for realistic aircraft control problems. Therefore, it is not necessary to design a complete full envelope controller including landing and take-off conditions, nor is it profitable to do longtime scale maneuvering such as climbing and positioning. Rather, complicated maneuvering during short time scales such as high angle of attack steps and combination of angle of attack and rolling maneuvering is tested. Stability and robustness during such maneuvering is considered more important and the long time scale maneuvering is left without consideration.

Flight Mechanics

u, v, w	vehicle velocities in the X, Y and Z-directions
p, q, r	vehicle angular roll, pitch and yaw velocities
ϕ, θ, ψ	Euler bank, attitude and heading angles
x_e, y_e, z_e	vehicle positions in inertial (flat earth) frame
$\mathbf{x}$	state vector
α	angle of attack
β	side slip angle
V	absolute velocity, vehicle speed
$\mathbf{V}$	$[u, v, w]^T$
$\boldsymbol{\omega}$	$[p, q, r]^T$
Mach	Mach number
m	vehicle mass
$\mathbf{I}$	inertia matrix
I_{xx}, I_{yy}, I_{zz}	moments of inertia around X, Y and Z-axis
T	engine thrust
ρ	air density
S	wing reference area
c	wing reference chord
b	wing reference span
δ_a	aileron deflection
δ_e	elevator deflection
δ_r	rudder deflection
δ_t	thrust setting
X, Y, Z	force components in X, Y and Z-directions
L	rolling moment
M	pitching moment
N	yawing moment
C_X	force coefficient in X-direction
C_Y	force coefficient in Y-direction
C_Z	force coefficient in X-direction
C_l	moment coefficient around the X-axis
C_m	moment coefficient around the Y-axis
C_n	moment coefficient around the Z-axis

Table 1.1: Flight mechanical nomenclature.

Chapter 2

Modelling

2.1 Aircraft dynamics

The Newton-Euler equations of motion for a rigid body are given by the equations for the rate of change of the linear and angular momentum, $\mathbf{p}$ and $\boldsymbol{\ell}$, respectively, commonly formulated around the center of mass (CoM) in an inertial coordinate system as

$$\dot{\mathbf{p}} = \mathbf{f}, \tag{2.1}$$
$$\dot{\boldsymbol{\ell}} = \mathbf{m}, \tag{2.2}$$

where $\mathbf{f}$ and $\mathbf{m}$ denote the total external force and total external moment around CoM, respectively. If both sides of (2.1), (2.2) are expressed in terms of a Cartesian body-fixed coordinate system centered in CoM, they can be rewritten on the familiar form [3]

$$\mathbf{F} = \dot{m}\mathbf{V} + m\dot{\mathbf{V}} + \boldsymbol{\omega} \times m\mathbf{V}, \tag{2.3}$$
$$\mathbf{M} = \dot{\mathbf{I}}\boldsymbol{\omega} + \mathbf{I}\dot{\boldsymbol{\omega}} + \boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega}, \tag{2.4}$$

where $\mathbf{V} = [u, v, w]^T$ and $\boldsymbol{\omega} = [p, q, r]^T$ are the *velocity* and *angular velocity,* respectively, $\mathbf{F}$ and $\mathbf{M}$ are the *force* and *moment,* respectively, and m and $\mathbf{I}$ are the *mass* and *inertia matrix,* respectively (all in body coordinates). If we assume that m and $\mathbf{I}$ both are constant then the Eqs. (2.3) and (2.4) reduce to

$$m\dot{\mathbf{V}} = -\boldsymbol{\omega} \times m\mathbf{V} + \mathbf{F}, \tag{2.5}$$
$$\mathbf{I}\dot{\boldsymbol{\omega}} = -\boldsymbol{\omega} \times \mathbf{I}\boldsymbol{\omega} + \mathbf{M}, \tag{2.6}$$

which is the most often used form of the force and moment equations in flight mechanics. In our applications the sum $\mathbf{F}$ of external body forces is divided into gravitational and other external forces (aerodynamical and thrust) as $\mathbf{F} = \mathbf{F}_g + \mathbf{F}_e$. The equations in (2.5) and (2.6) will be the basis for our modelling of the motion of the aircraft.

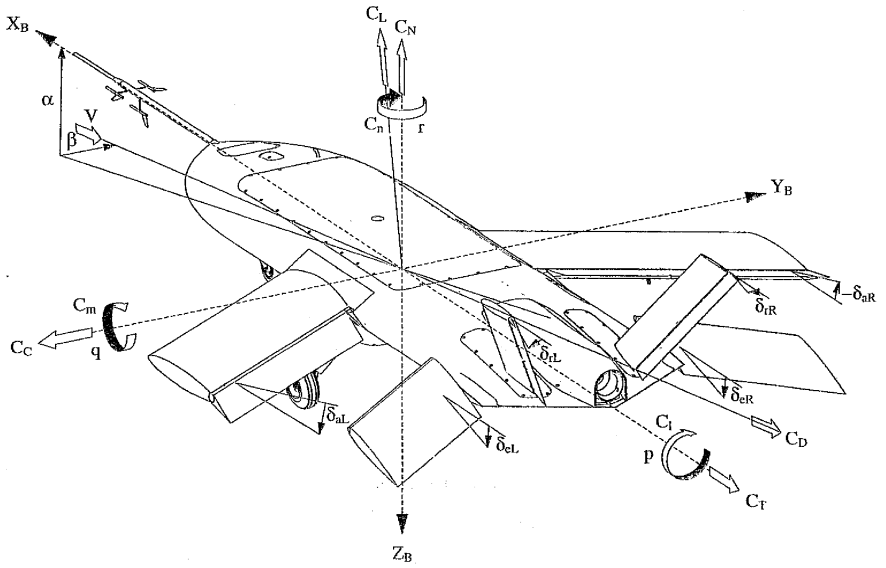

Figure 2.1: Definitions of the coordinate system and the aerodynamic angles.

We shall assume that the aircraft is symmetric in the x, z-plane and the inertia crossproducts I_{xy} and I_{yz} are therefore zero. Further, I_{xz} is small in comparison with I_{xx}, I_{yy} and I_{zz} and is therefore neglected. The matrix of inertia is therefore of the diagonal form

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}.$$

The aerodynamic angles α, *angle of attack*, and β, *sideslip angle*, are defined in terms of the x, y and z-components of the velocity $\mathbf{V}$ as

$$\begin{aligned} u &= V\cos\alpha\cos\beta, \\ v &= V\sin\beta, \\ w &= V\sin\alpha\cos\beta, \end{aligned} \tag{2.7}$$

where $V = \sqrt{u^2+v^2+w^2}$ is the absolute velocity. In Figure 2.1 the standard vehicle Cartesian coordinate system and the definitions of α and β are illustrated. The vector $\mathbf{F}_g$ of gravitational body forces is given by

$$\mathbf{F}_g = [-g\sin\theta, g\cos\theta\sin\phi, g\cos\theta\cos\phi]^T,$$

where g is the gravitational acceleration constant and ϕ, θ are the Euler *bank angle* and *pitch angle,* respectively. If we express the vector $\mathbf{F}_e$ of

other external forces (apart from gravity) as $\mathbf{F}_e = [X, Y, Z]^T$ and the vector of external moments $\mathbf{M}$ as $\mathbf{M} = [L, M, N]^T$ the force and moment equations (2.5) and (2.6) can be written in component form as

$$\begin{aligned}
\dot{u} &= vr - wq - g\sin\theta + X/m, \\
\dot{v} &= wp - ur + g\cos\theta\sin\phi + Y/m, \\
\dot{w} &= uq - vp + g\cos\theta\cos\phi + Z/m, \\
\dot{p} &= qr(I_{yy} - I_{zz})/I_{xx} + L/I_{xx}, \\
\dot{q} &= rp(I_{zz} - I_{xx})/I_{yy} + M/I_{yy}, \\
\dot{r} &= pq(I_{xx} - I_{yy})/I_{zz} + N/I_{zz}.
\end{aligned} \tag{2.8}$$

These six equations represent the standard six-degree of freedom (6DOF) model for aircraft dynamics used in the literature [3].

To complete our dynamical model of the aircraft we need dynamical relations also for the Euler angles and the inertial position. [1] The dynamical equations for the Euler angles ϕ, θ and ψ are given in terms of the body velocities and angular velocities as

$$\begin{aligned}
\dot{\phi} &= p + q\sin\phi\tan\theta + r\cos\phi\tan\theta, \\
\dot{\theta} &= q\cos\phi - r\sin\phi, \\
\dot{\psi} &= q\sin\theta\sec\phi + r\cos\phi\sec\theta,
\end{aligned} \tag{2.9}$$

and the relations for the inertial position coordinates x_e, y_e and z_e are

$$\begin{aligned}
\dot{x}_e &= u\cos\theta\cos\psi + v(\sin\phi\sin\theta\cos\psi - \cos\phi\cos\psi) \\
&\quad + w(\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi), \\
\dot{y}_e &= u\cos\theta\sin\psi + v(\sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi) \\
&\quad + w(\cos\phi\sin\theta\sin\psi + \sin\phi\cos\psi), \\
\dot{z}_e &= -u\sin\theta + v\sin\phi\cos\theta + w\cos\phi\cos\theta.
\end{aligned} \tag{2.10}$$

The derivatives of the Euler angles are found from geometric relations and the last three equations are a coordinate transformation from body coordinates frame to the earth coordinates [3].

The aerodynamical forces are most conveniently expressed in terms of the aerodynamical angles α, β and therefore it is convenient to represent the first three equations in (2.8) in terms of α, β. By solving for α, β in (2.7),

[1] We assume motion over a flat earth on which a coordinate system is fixed, which we moreover assume provides an inertial system.

taking the time derivative and using (2.8) we obtain

$$\begin{aligned}\dot{\alpha} &= \frac{\cos(\alpha)}{V\cos(\beta)}(\frac{Z}{m}) - p\cos(\alpha)\tan(\beta) \\ &\quad +q - r\sin(\alpha)\tan(\beta) - \frac{\sin(\alpha)}{V\cos(\beta)}(\frac{X}{m}), \\ \dot{\beta} &= \frac{\cos(\beta)}{V}(\frac{Y}{m}) + p\sin(\alpha) - r\cos(\alpha) \\ &\quad -\frac{\cos(\alpha)\sin(\beta)}{V}(\frac{X}{m}) - \frac{\sin(\alpha)\sin(\beta)}{V}(\frac{Z}{m}).\end{aligned} \tag{2.11}$$

Since the absolute velocity often varies on a somewhat slower timescale compared to the other variables, it is common to make the simplifying assumption that

$$\dot{V} = 0, \tag{2.12}$$

which also shall be done here. [2] This completes our basic dynamical description of the aircraft motion.

2.1.1 Wind vector roll

The assumption (2.12) is exactly fulfilled for a simple but important maneuver, the *wind vector roll*. A wind vector roll is characterized by the fact that the body angular wind vector $\boldsymbol{\omega}$ satisfies

$$\boldsymbol{\omega} = P_{[\mathbf{V}]}\boldsymbol{\omega} = \frac{\mathbf{V}^T\boldsymbol{\omega}}{V^2}\mathbf{V},$$

i.e. the angular wind vector is perfectly aligned with the wind vector $\mathbf{V}$. If we assume that the aircrafts velocity is at a state such that $\dot{\mathbf{V}} = 0$ (e.g. level wings trimmed flight) we see that we can add to $\boldsymbol{\omega}$ a component $\boldsymbol{\omega}_{\mathbf{v}}$ which is aligned with the wind vector $\mathbf{V}$ and still maintain the condition $\dot{\mathbf{V}} = 0$ since the cross product term $\boldsymbol{\omega}_{\mathbf{v}} \times \mathbf{V}$ then appearing in (2.5) will be zero. [3] For a wind vector roll, only one variable is needed to describe the motion, the *wind vector roll angle* Φ, defined via the following relation for the wind vector roll rate

$$\dot{\Phi} = \frac{\mathbf{V}^T\boldsymbol{\omega}}{V}. \tag{2.13}$$

It is common in real flight control systems to let the lateral stick motion command the wind vector roll rate $\dot{\Phi}$ directly, rather than the roll angle Φ,

[2] The vehicle speed V is normally controlled either manually or via a separate controller, not included in the stability augmenting part of the flight control system considered here.

[3] After a finite amount of time the gravity force term $\mathbf{F}_g$ will change, and so will the total force $\mathbf{F}$, unless compensated for by a change in the aerodynamical forces included in $\mathbf{F}_e$, but the wind vector roll angular velocity itself will not induce any change in V.

and therefore it is necessary to find a dynamical relation also for $\dot{\Phi}$. If we define $\xi = \dot{\Phi}$ we have, using the assumption $\dot{V} = 0$,

$$\dot{\xi} = \frac{d}{dt}\frac{\mathbf{V}^T\boldsymbol{\omega}}{V} = \frac{\dot{\mathbf{V}}^T\boldsymbol{\omega}}{V} + \frac{\mathbf{V}^T\dot{\boldsymbol{\omega}}}{V}, \tag{2.14}$$

which, after expanding the terms using (2.8), can be written

$$\begin{aligned}\dot{\xi} \;=\; & \frac{1}{V}\Big[p(vr - wq - g\sin\theta + X/m) \\ & +q(wp - ur + g\cos\theta\sin\phi + Y/m) \\ & +r(uq - vp + g\cos\theta\cos\phi + Z/m) \\ & +u(qr(I_{yy} - I_{zz})/I_{xx} + L/I_{xx}) \\ & +v(rp(I_{zz} - I_{xx})/I_{yy} + M/I_{yy}) \\ & +w(pq(I_{xx} - I_{yy})/I_{zz} + N/I_{zz})\Big].\end{aligned}$$

If we further substitute the expressions for u, v, w in (2.7) and make the approximation $\cos\beta = 1$, which is valid for small β, we end up with the following (approximate) relation for the time derivative of the wind vector roll rate

$$\begin{aligned}\dot{\xi} \;=\; & \frac{1}{V}\Big[p(-g\sin\theta + X/m) \\ & +q(g\cos\theta\sin\phi + Y/m) \\ & +r(g\cos\theta\cos\phi + Z/m) \\ & +V\cos(\alpha)(qr(I_{yy} - I_{zz})/I_{xx} + L/I_{xx}) \\ & +V\sin(\beta)(rp(I_{zz} - I_{xx})/I_{yy} + M/I_{yy}) \\ & +V\sin(\alpha)(pq(I_{xx} - I_{yy})/I_{zz} + N/I_{zz})\Big].\end{aligned} \tag{2.15}$$

If we take the dynamical relation (2.15) together with (2.11), the three last relations in (2.8), and (2.9), (2.10) we have a complete description of the motion of the aircraft formulated in terms of the variables one most often wishes to directly control. Relying on the assumption (2.12) the equation for V is excluded and the remaining twelve equations are our non-linear equations of motion and are from now on referred to as

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}), \tag{2.16}$$

where $\boldsymbol{x} = [\alpha\ \beta\ p\ q\ r\ \xi]^T$ or $\boldsymbol{x} = [\alpha\ \beta\ p\ q\ r\ \Phi]^T$ and where the term for the control input, $\mathbf{g}(\mathbf{x})\mathbf{u}$, is left out.

is the state vector to be controlled. This (2.16) is also dependent in the parameters z_e, V, $Mach$, ϕ, θ and ψ. These states, $\mathbf{x}$ are not defined such that the equilibrium points are located in the origin necessarily, $\mathbf{f}(\mathbf{0}) = \mathbf{0}$, which is required when using LQ and SDRE. In Section 3.3 a change of variables is made fulfilling this condition.

2.1.2 Actuator dynamics

The control surfaces and their definitions are seen in Figure 2.1. According to this figure there are six control surfaces, two elevators, two ailerons and two rudders denoted as δ_e, δ_a and δ_r respectively. There are also two flaps but such control surfaces are only used for lowspeed flying, i.e. landing and take-off situations which are not addressed in this work. Therefore this control surface is fixed to a zero degree setting. The rest of the control surfaces are coupled such that $\delta_{e_{left}} = \delta_{e_{right}}$, $\delta_{r_{left}} = \delta_{r_{right}}$ and $\delta_{a_{left}} = -\delta_{a_{right}}$ (in the obvious notation). Each control surface has a minimum and maximum possible deflection according to Table 2.1. All control surface actuators are

Control surface	Lower limit	Upper limit
δ_e	-30	10
δ_a	-30	30
δ_r	-30	30

Table 2.1: Control surface deflection interval

modelled by a simple linear first order dynamics with time constant selected to be the same as the one used in the the generic fighter aircraft model ADMIRE [16]. The common transfer function for the dynamics of all actuators is thus

$$G(s) = \frac{1}{0.05s + 1}, \tag{2.17}$$

and this choice of actuator dynamics is motivated by the fact that the ADMIRE model represents a larger aircraft, with slower actuator dynamics. The dynamics for the smaller URAV is at least as fast so this assumption is valid.

2.2 Windtunnel data

A physical model of the aircraft has been made, see Figure 2.2, and tested in a windtunnel. The properties of the model have been evaluated for several different flying conditions and all data are functions of Mach number, sideslip angle and angle of attack and possibly control surface setting and rate of rotation p, q and r. The flying conditions tested are for Mach numbers

$$[0.17\ 0.3\ 0.5\ 0.7\ 0.78\ 0.8\ 0.84\ 0.86\ 0.88\ 0.9\ 0.95],$$

where $\alpha \in [-10 : 32]$ and $\beta \in [-15 : 15]$. The data is stored in such way that the total contribution, for example pitching moment, is as illustrated in Figure 2.3, simply a sum of each component, i.e wind vector and control

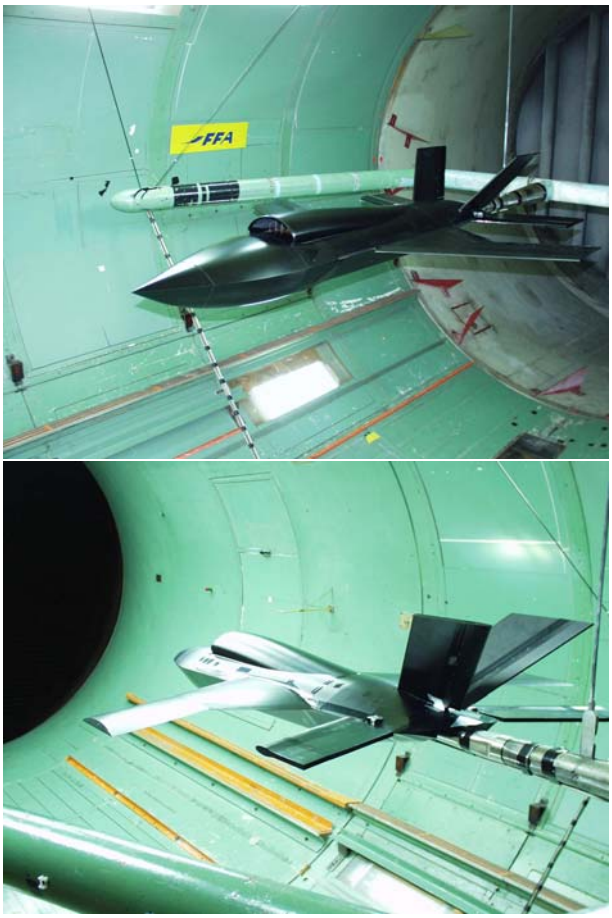

Figure 2.2: Windtunnel model of the URAV mounted in a windtunnel

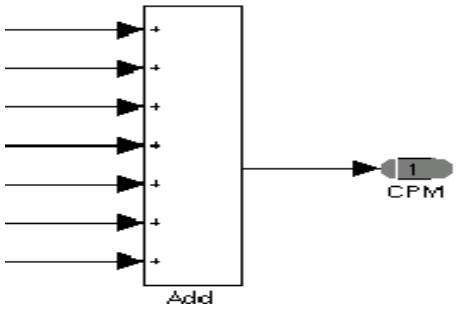

Figure 2.3: Simulink block: Sum of all contributions to the pitching moments.

surfaces,

$$C_m = C_m(\alpha,\beta) + C_m(\alpha,\beta,\delta_{e_l}) + C_m(\alpha,\beta,\delta_{e_r}) + C_m(\alpha,\beta,\delta_{a_l}) \\ + C_m(\alpha,\beta,\delta_{a_r}) + C_m(\alpha,\beta,\delta_r) + C_m(\alpha,\beta,\hat{q}). \quad (2.18)$$

The aerodynamic forces and moments are, according to standard definitions in aerodynamic literature [3], given in terms of the aerodynamic coefficients as

$$X = C_X \frac{1}{2}\rho V^2 S$$
$$Y = C_Y \frac{1}{2}\rho V^2 S$$
$$Z = C_Z \frac{1}{2}\rho V^2 S$$
$$L = C_l \frac{1}{2}\rho V^2 S b$$
$$M = C_m \frac{1}{2}\rho V^2 S c$$
$$N = C_n \frac{1}{2}\rho V^2 S b$$

where ρ is the freestream air density, V is the absolute airspeed which by definition equals the free stream speed. The constant S is the reference wing area defined as the total wingarea including the wingbox, i.e. the area hidden by the cabin, b is the reference span defined as the distance between the wing tips and c is the aerodynamic reference chord defined as

$$c = \frac{1}{S}\int_{-b/2}^{b/2} [c(y)]^2 dy \quad (2.19)$$

where $c(y)$ is the local chord at distance y from midspan. The aerodynamic coefficients are defined as

$$C_X = C_X(\alpha,\beta) + C_X(\alpha,\beta,\delta), \quad (2.20)$$

$$C_Y = C_Y(\alpha,\beta) + C_{Y_{\hat{r}}}(\alpha,\beta)\hat{r} + C_Y(\alpha,\beta,\delta), \tag{2.21}$$
$$C_Z = C_Z(\alpha,\beta) + C_{Z_{\hat{q}}}(\alpha,\beta)\hat{q} + C_Z(\alpha,\beta,\delta), \tag{2.22}$$
$$C_l = C_l(\alpha,\beta) + C_{l_{\hat{p}}}(\alpha,\beta)\hat{p} + C_l(\alpha,\beta,\delta), \tag{2.23}$$
$$C_m = C_m(\alpha,\beta) + C_{m_{\hat{q}}}(\alpha,\beta)\hat{q} + C_m(\alpha,\beta,\delta), \tag{2.24}$$
$$C_n = C_n(\alpha,\beta) + C_{n_{\hat{r}}}(\alpha,\beta)r + C_n(\alpha,\beta,\delta), \tag{2.25}$$

where the first term is due to the wind vector, the second one is due to one of the normalized angular velocities, $\hat{p}, \hat{q}$ or $\hat{r}$ and the last term is due to the control surfaces. The normalized angular velocities are defined as

$$\hat{p} = \frac{pb}{2V},$$
$$\hat{q} = \frac{pc}{2V},$$
$$\hat{r} = \frac{rb}{2V}.$$

All these coefficients are, for separate values, stored in a table from which Simulink uses linear interpolation to find the most suitable values for the current state. This is addressed more in detail below.

2.3 Polynomial approximations of the aerodynamic data

In SDRE control design it is convenient to make polynomial approximations of the aerodynamic data using Least Squares fitting. It is possible to interpolate directly (with smooth interpolation) in raw data but calculation time is lower when using precalculated polynomials. Often linear approximations are good for small angles of attack and side slip but for better accuracy at higher angles non linear polynomials are needed. For the wind vector, different base functions are used for different coefficients. In the longitudinal case, the base functions are according to (2.26) given by

$$\begin{aligned} C_{T,Z,m}(\alpha,\beta) =& (c_{T,Z,m_0} \dots c_{T,Z,m_{\alpha\beta}}) \cdot \\ & (1 \;\; \alpha \;\; \alpha^2 \;\; \alpha^3 \;\; \alpha^4 \;\; \beta \;\; \beta^2 \;\; \beta^3 \;\; |\beta|^3 \;\; \alpha|\beta| \;\; \alpha\beta)^T, \end{aligned} \tag{2.26}$$

and the lateral coefficients use

$$C_{C,l,n}(\alpha,\beta) = (c_{C,l,n_\beta} \dots c_{C,l,n_{\alpha\beta}})(\beta \;\; \beta^2 \;\; \beta^3 \;\; |\beta|^3 \;\; \alpha|\beta| \;\; \alpha\beta)^T. \tag{2.27}$$

The remaining coefficients, due to p, q, r or δ, use the same base functions as for the longitudinal case. These functions are never differentiated so it is acceptable to use base functions with discontinuous derivatives. Since these coefficients are also a function of Mach number each polynomial is evaluated for each separate Mach number. Linear interpolation is assumed valid and used when necessary. The polynomial fit is, as seen in Figure 2.4, good and the residual is in the order of 10^{-3}.

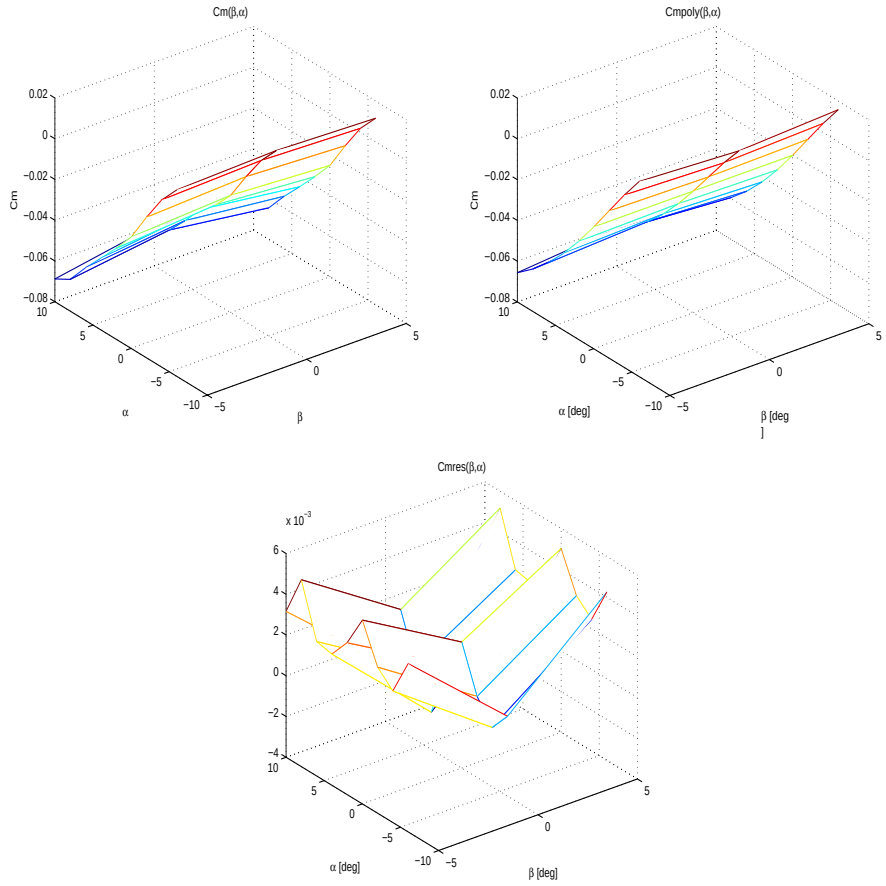

Figure 2.4: Difference between raw aerodynamic data and polynomials for Pitch moment coefficient at Mach = 0.5. Top left: Windtunnel data. Top right: Polynomial. Bottom: Relative norm between windtunnel data and polynomials.

2.4 Simulink

The model is implemented in Simulink using Aerospace blockset for aircraft specific modelling [17]. This blockset is described in detail in section 2.4.1. Figure 2.5 shows the main window for the simulink model. The main block called "URAV-model" is a subsystem containing all the aircraft dynamics and environmental conditions etc. The block "SDRE-controller", which is the control system, is an S-function using the states and a signal to compute the necessary output. The signal builder provides the system with the commanded states as a function of time, see Fig. 2.6. A simplified picture showing the flow of calculations made is found in Figure 2.7.

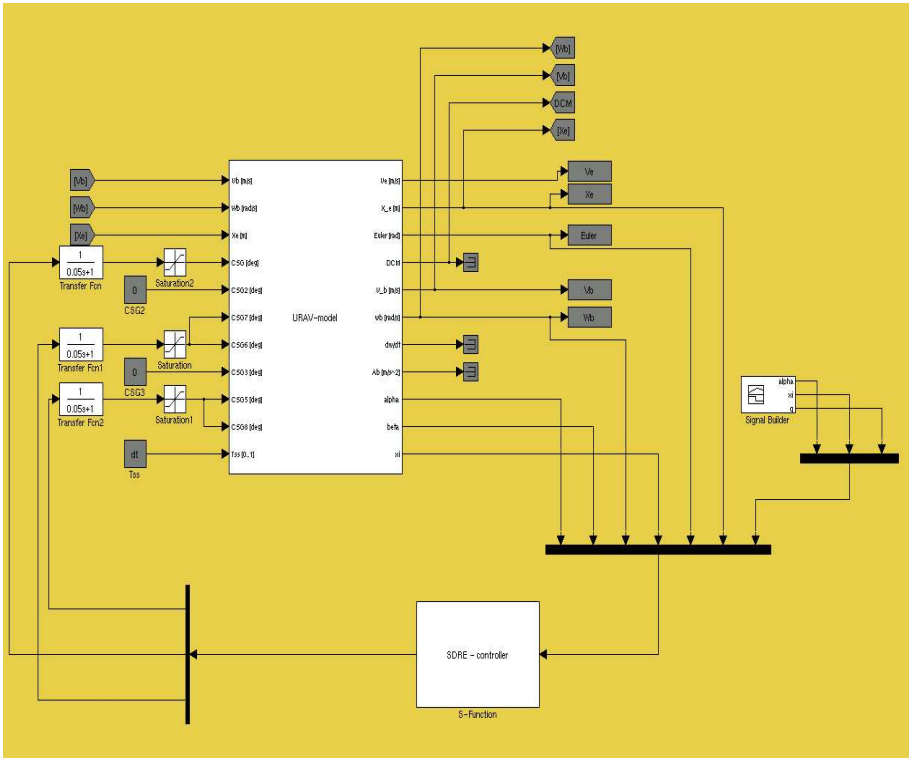

Figure 2.5: Simulink block: URAV modell.

2.4.1 Aerospace Blockset

The Aerospace Blockset [17] is a toolbox for simulink used in modelling and simulations of aircrafts, spacecrafts, rockets and propulsion systems. It includes blocks that implement models for equations of motion, environment conditions, propulsion etc. The following list describes further the blocks used in this thesis;

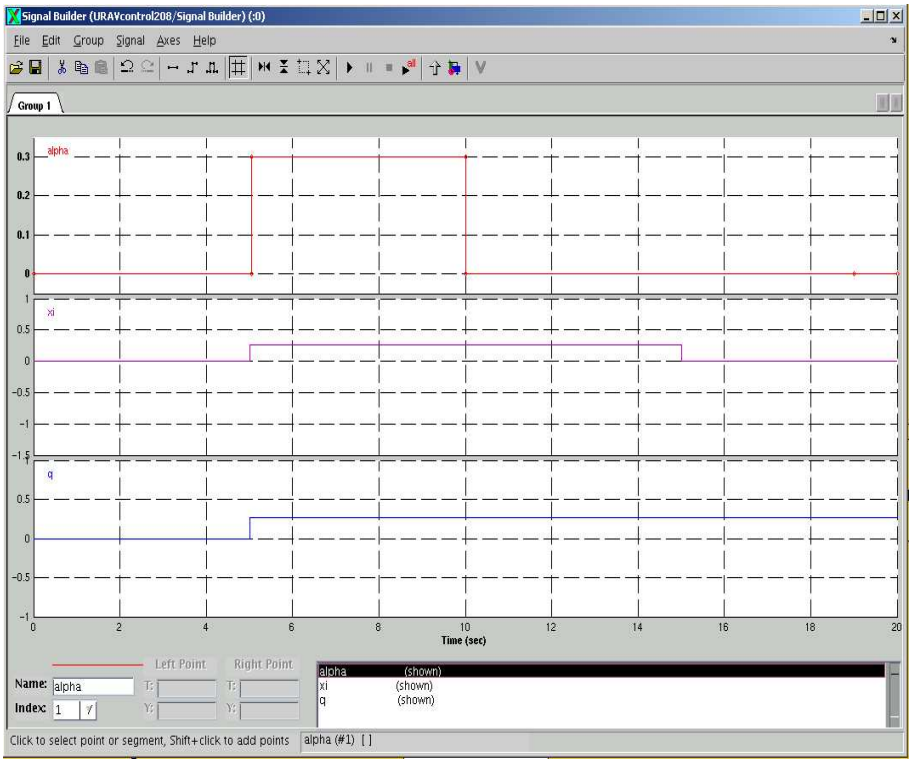

Figure 2.6: Simulink block: Signal builder.

- Equation of motions. A 6 degree-of-freedom model using forces and moments in calculation of the flying dynamics, see Figure 2.8.
- Environmental. Based on standard atmospheric models and gravity this block uses the altitude to calculate the speed of sound, the density, the pressure and the temperature of the air, see Figure 2.9.
- Aerodynamics. Windtunnel data are stored in and interpolated from lookup tables. Three of four inputs are used to calculate different contributions of the forces and moments. See Figure 2.10, Here control surface 1 and 4 corresponds to left and right elevator, control surface 10 and 12 corresponds to left and right elevator and finally control surface 11 corresponds to the rudder.
- Engine. Engine model using Mach number, altitude and throttle setting to calculate the correct thrust. Also fuel consumption is modelled but since mass is considered constant this is not included in this thesis. See Figure 2.11.

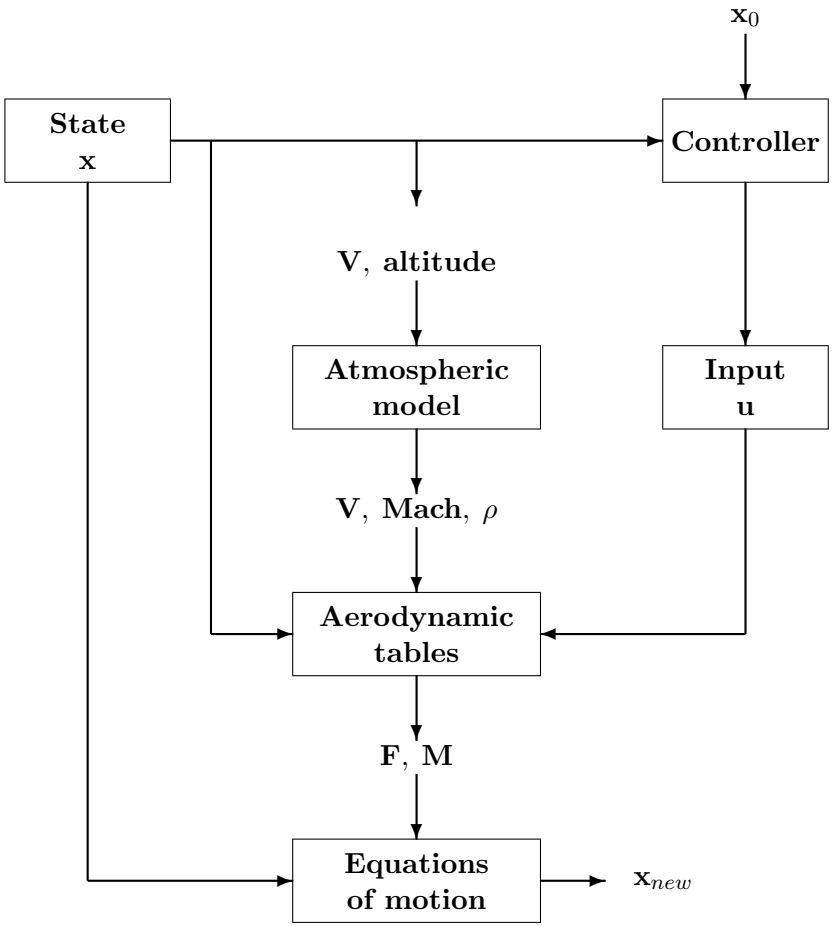

Figure 2.7: The flow of calculations made in simulink.

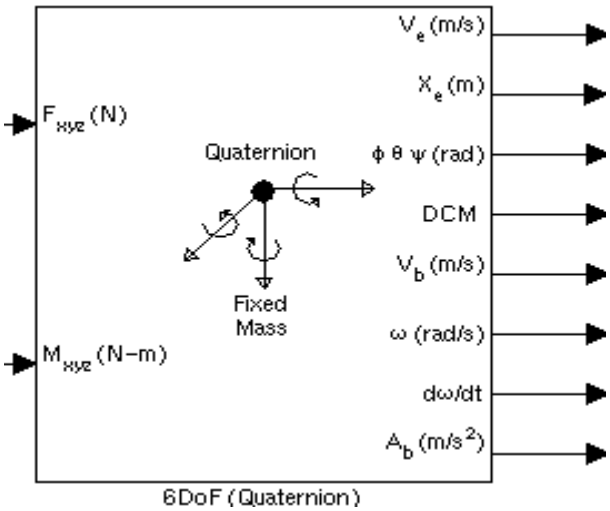

Figure 2.8: Simulink block: 6DoF modell for equations of motion.

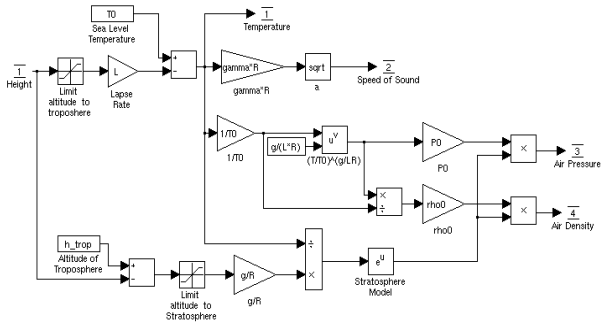

Figure 2.9: Simulink block: Atmospheric modell giving temperature, speed of sound, pressure and density at a given altitude.

2.5 Limitations and assumptions

2.5.1 Windtunnel data

During early flight testing of the model the model demonstrated somewhat strange behaviour. When flying with no sideslip, $\beta = 0$, there should be no side force, roll moment nor yawing moment however the model clearly indicated such behaviour. This originated in an error from the windtunnel testing where probably the *arm* holding the modell was not fixed properly. This may have rendered a sideslip $\beta \neq 0$ yielding not zero forces and moments. The corresponding values were changed (in a somewhat brute force manner) by simply adding suitable corrections in order to make them zero. Still, the backlash in the windtunnel arm might have influenced the accuracy in all the data negatively.

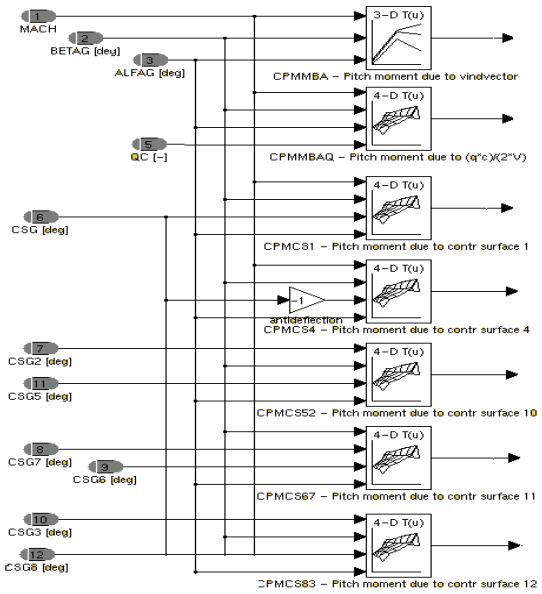

Figure 2.10: Simulink blocks: Lookup tables, one for each component of the pitching moment.

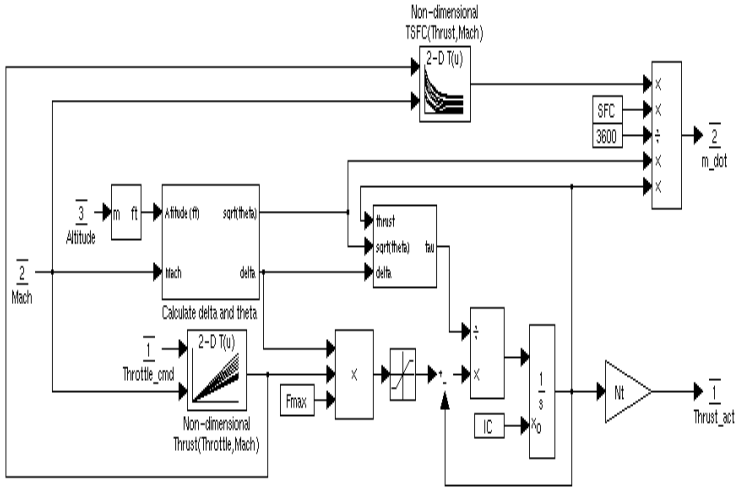

Figure 2.11: Simulink block: Engine model calculating the thrust and fuel consumption.

Chapter 3

Controller Design

The controller design is based on the following model for the overall system

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{u}, \tag{3.1}$$

where $\mathbf{x} \in \mathbb{R}^n$ is the state vector, $\mathbf{u} \in \mathbb{R}^m$ is the control vector, $\mathbf{f}$ is a smooth vector field in $\mathbb{R}^n$ and $\mathbf{g}$ is a continuous matrix function on $\mathbb{R}^n$ with values in $\mathbb{R}^{n\times m}$. In (3.1) the control input $\mathbf{u}$ enters linearly. This assumption can be made to hold also for the aircraft equations of motion (2.16), by proper choice of control input, so the model (3.1) is applicable to (2.16).

One attractive idea for control design is nonlinear dynamic inversion (NDI) where one takes $\mathbf{u} = \mathbf{g}(\mathbf{x})^{-1}[-\mathbf{f}(\mathbf{x}) + \dot{\mathbf{x}}^{des}]$, where $\dot{\mathbf{x}}^{des}$ represents the desired dynamics. This is however only possible if $\mathbf{g}(\mathbf{x})$ is invertible and this is not always the case. In particular, this is never the case for the aircraft equations of motion (2.16), and not even the rigid body dynamics (2.8). The latter fact can be seen as the motivation for NDI via two-timescale separation (TSS) where the inversion is done in two steps. (TSS is further explained in Appendix B.) SDRE, on the other hand, is directly applicable to the system (3.1) as it stands, regardless of the structure of $\mathbf{g}(\mathbf{x})$. The calculations relating to SDRE are central to our study and will be described in in more detail in Section 3.4. However, before addressing the detailed aspects of SDRE method we shall consider the issue of controllability and the choice of control input.

3.1 Controllability and input definition

The input $\boldsymbol{u}$ will consist of moments, $\boldsymbol{u} = [L, M, N]^T$. The corresponding matrix $\mathbf{g}(\mathbf{x})$ as in (3.2) is then easily found be examining (2.11) for the relation for α and β, the three last equations of (2.8) for p,q and r and

(2.15) or (2.13) for ξ or Φ command respectively. This reveals that

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \\ \frac{\cos(\alpha)}{Ixx} & \frac{\sin(\beta)}{Iyy} & \frac{\sin(\alpha)}{Izz} \end{pmatrix}. \tag{3.2}$$

when controlling $\boldsymbol{x} = [\alpha\ \beta\ p\ q\ r\ \xi\ ]^T$ and the equivalent matrix

$$\mathbf{g}(\mathbf{x}) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{1}{I_{xx}} & 0 & 0 \\ 0 & \frac{1}{I_{yy}} & 0 \\ 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 \end{pmatrix}, \tag{3.3}$$

when controlling $\boldsymbol{x} = [\alpha\ \beta\ p\ q\ r\ \Phi]^T$.

The matrices in (3.2) and (3.3) are not only constant or essentially constant, except for small variations in α and β in the last row of the matrix in (3.2), but they also provide a linear relation for the influence of $\mathbf{u}$ in the system dynamics, thus no simplification is necessary here (in comparison with using the control surface setting, δ, as input where the influence is nonlinear). Although the matrices in (3.2), (3.3) consist mostly of zeros they give, in all cases tested (see below), controllability for the SDC version of the system (3.1) according to Def. 3.1 below.

Controllability for a linear time invariant system on state space form (as in (3.15) below) can be determined by applying the following result to the pair of matrices $(\mathbf{A}, \mathbf{B})$ occurring in the state space description.

Definition ([1]) 3.1. The pair $(\mathbf{A}, \mathbf{B})$ is controllable if and only if the controllability matrix $\mathcal{C}$ defined as

$$\mathcal{C} = [\mathbf{B}, \mathbf{AB}, \ldots, \mathbf{A}^{n-1}\mathbf{B}]$$

has full rank.

The criterion in Definition 3.1 can be applied pointwise to the SDC version (1.3) of the system in (3.1) by taking $\mathbf{A} = \mathbf{A}(\mathbf{x})$ and $\mathbf{B} = \mathbf{B}(\mathbf{x})$, and this is done during all maneuvers in the simulations. The corresponding control surface settings, necessary to produce any moment vector $[L, M, N]^T$, are calculated via

$$\begin{pmatrix} L \\ M \\ N \end{pmatrix} = \begin{pmatrix} L(\delta_a, \delta_r) \\ M(\delta_e, \delta_a, \delta_r) \\ N(\delta_a, \delta_r) \end{pmatrix}, \tag{3.4}$$

or, with a common simplification where the moments are assumed to be linearly dependent in the control surfaces,

$$\begin{pmatrix} L \\ M \\ N \end{pmatrix} = \begin{pmatrix} 0 & L_{\delta_a} & L_{\delta_r} \\ M_{\delta_e} & M_{\delta_a} & M_{\delta_r} \\ 0 & N_{\delta_a} & N_{\delta_r} \end{pmatrix} \begin{pmatrix} \delta_e \\ \delta_a \\ \delta_r \end{pmatrix}, \tag{3.5}$$

where L and N are independent of δ_e due to symmetry reasons. The aileron, δ_a, and the rudder, δ_r, are mainly used to produce rolling moment, L and yawing moment, N respectively but they will also have a small effect in remainder moments. Thus, L_{δ_a}, M_{δ_e} and N_{δ_r} will be much larger than the remaining components in the matrix (3.5) and this matrix will therefore have full rank, i.e. its inverse will exist. The nonlinear version (3.4) will be used in this thesis.

The effect on the forces by the control surfaces are small and neglected in comparison with its influence in the moments since $\max(X(\delta), Y(\delta), Z(\delta))$ is less than 10% of $\min(L(\delta), M(\delta), N(\delta))$. This is a common simplification in aircraft control design and it does not effect controllability. Limits in the input is not possible to model using the SDRE design (since the input is linear) however a saturation in the input is added according to

$$\boldsymbol{M} = \max[\min[\boldsymbol{M}_{max}, \boldsymbol{M}], \boldsymbol{M}_{min}] \tag{3.6}$$

where max and min of $\boldsymbol{M}$ are the maximum and minimum moments that are achievable for the current state. This will via (3.4) saturate the control surface setting within the allowed deflections according to Table 2.1.

3.2 Trimming

For a wanted equilibrium flying condition, $\mathbf{x}_0$, trimmed conditions means that the input $\mathbf{u} = \mathbf{u}_0$ is chosen to fulfill

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\mathbf{u}_0. \tag{3.7}$$

These conditions will not make $\mathbf{x}_0$ an equilibrium for the original system but will, as explained in Section 3.3, be used to create an equilibrium at 0 for a substitute system obtained after a change of variables. (In the new variables, 0 will be an equilibrium, as required by the SDRE theory.)

Here two methods to find trimmed conditions for horizontal level flight, constant looping and wind vector roll are described. The first method uses built in optimization functions of MATLAB to find trimmed conditions based on a formulation as

$$\min_{\mathbf{x},\mathbf{u}} \sum W_i \dot{\mathbf{x}}_i^2, \tag{3.8}$$

where W_i are weights that initially normalizes the state derivative and also lets one choose what state derivative is most crucial. The normalizations are

necessary since the different optimization variables have different magnitude and would therefore influence the solution differently. The second method is a more explicit method used when the wanted $\mathbf{x}_0$, or parts of it, is known. This method is used to find trimmed conditions for the constant looping and the wind vector roll and is described further below.

3.2.1 Horizontal level flight

For straight level flight the aircraft has to be in total equilibrium, i.e. one where also the angular velocities are zero. For a symmetric configuration the lateral degrees of freedom ϕ and ψ and v as well as their derivatives are assumed to be zero. Also the lateral control surfaces, δ_a and δ_r, are set to zero. Left are the longitudinal degrees of freedom, pitch angle θ and the velocities u and w. Trimmed conditions, for straight level flight, means that the lift and pitch moment with respect to α and δ_e have to be equal to the aircraft's weight and zero respectively. Furthermore the thrust from the engine must be equal to the drag. Hence, if the influence of the control surface and if the lift contribution from α are considered to be linear (which for horizontal level flight is a fair assumption), then

$$\begin{pmatrix} W \\ 0 \\ D \end{pmatrix} = \begin{pmatrix} L_\alpha & L_{\delta_e} & 0 \\ M_\alpha & M_{\delta_e} & M_{\delta_t} \\ 0 & 0 & T_{\delta_t} \end{pmatrix} \begin{pmatrix} \alpha \\ \delta_e \\ \delta_t \end{pmatrix} \tag{3.9}$$

where all variables are scalar. Here W is the aircraft weight, L is the lift force, M is the pitch moment, D is the drag force and T is thrust. These forces and moments are defined in Section 2.2. If the derivatives, L_α, L_{δ_e}, M_α, M_{δ_e}, of these coefficients are known, then α, δ_e can be calculated explicitly. Note that via ρ and V the needed α and δ_e is dependent on altitude and airspeed. For each altitude and airspeed there exists only one corresponding solution to (3.9).

If these coefficients are unknown, or if the control surface influence cannot be considered linear, the trimming problem has to be solved numerically. For equilibrium the net longitudinal forces and moments have to be zero to make the acceleration in x_e, z_e and θ directions equal to zero. Or, more conveniently, after considering the states in $\mathbf{x}$, $\dot{u}$, $\dot{w}$ and $\dot{q}$ are each $= 0$. The minimization problem is formulated as

$$\min_{\delta_e, \delta_t, \alpha} \sum W_i \dot{\mathbf{x}}_i^2, \tag{3.10}$$

where $\dot{\mathbf{x}}_i$ is $\dot{u}$, $\dot{w}$ and $\dot{q}$ and $W_q = 100$, W_u and $W_w = 1$.

Figures 3.1 to 3.4 show 40 seconds of flight in trimmed conditions at altitude 5000 meters and Mach = 0.5 calculated using (3.10). Notice that the speed, θ and α are essentially constant, that β and q are small and that p, r, ϕ and ψ are zero as expected.

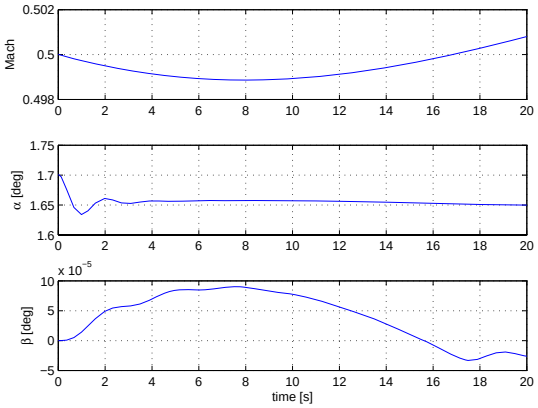

Figure 3.1: Mach, β and α.

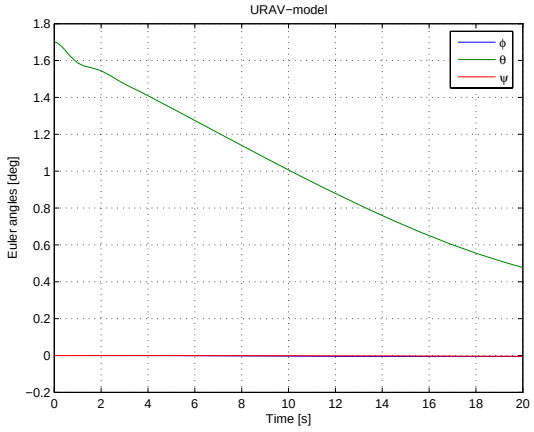

Figure 3.2: Euler angles, ϕ, θ and ψ.

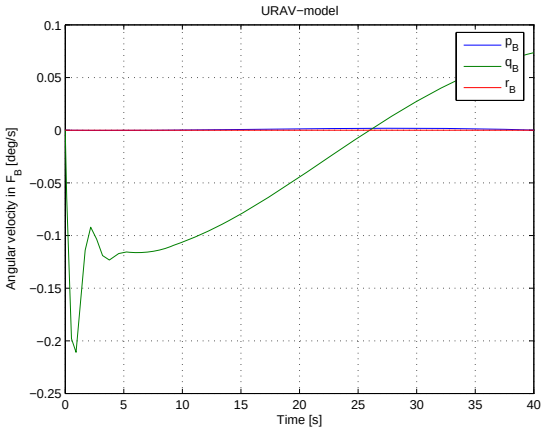

Figure 3.3: Rate of rotation ,p,q,r.

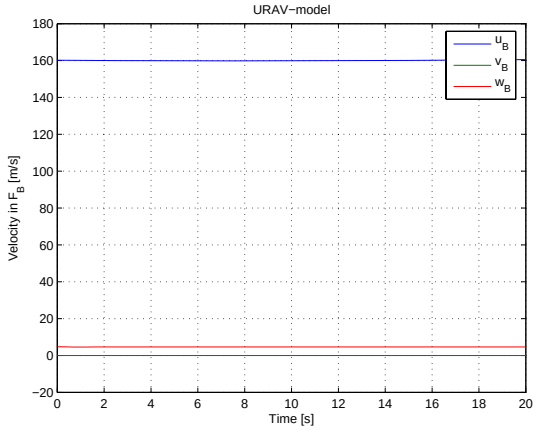

Figure 3.4: Velocities u, v and w.

3.2.2 Looping

For a certain α_0 and altitude and speed there exists a certain q_0 that will put the aircraft in a constant looping, i.e. a circular motion in the earth xz-plane. During a looping the gravity vector will change direction dramatically in the body coordinate system and the throttle setting needed to keep the speed relatively constant will vary considerably during such a maneuver. Since all testing in this thesis will be done keeping the throttle setting constant, performing a perfect looping as described above will not be possible. However, calculating q_0 and the corresponding δ_e (this is still a pure longitudinal maneuver and δ_r and δ_a will be kept at zero degree setting) for this maneuver can be done and loopings, partial or complete, are performed with a high variation in speed. This maneuver evaluates the control systems ability to perform steps in α and its ability to hold it constant during a looping. How to calculate this q_0 and δ_e is described below. Consider the equations for $\dot{\alpha}$, $\dot{\beta}$ and $\dot{\xi}$ in (2.11) and (2.15). Let us denote these equations as

$$\dot{\mathbf{x}} = \mathbf{h}_\theta(\mathbf{x}) + \mathbf{k}_\theta(\mathbf{x}, \omega). \tag{3.11}$$

where the index θ indicates that this equation is parameter dependent in the parameters $\theta = [V\ Mach\ \phi\ \theta\ \psi\ z_e]^T$. Forcing $\dot{\mathbf{x}} = \mathbf{0}$ will, for $\alpha = \alpha_0$ and β and ξ equal to 0, via an iteration process yield the corresponding ω_0. For a looping with β and ξ equal to 0, ω_0 will be equal to $[0\ q_0\ 0]^T$. The moment needed to perform this maneuver with a constant ω_0 is found from (2.6)

$$\dot{\omega}_0 = \mathbf{0} = \mathbf{I}^{-1}(\mathbf{M} - \omega_0 \times \mathbf{I}\omega_0), \tag{3.12}$$

or simply

$$\mathbf{M} = \omega_0 \times \mathbf{I}\omega_0. \tag{3.13}$$

From this, $\mathbf{u}_0$ is found via

$$\mathbf{u}_0 = \mathbf{M} - \mathbf{M}_{state}, \tag{3.14}$$

where $\mathbf{M}$ is the total moment and $\mathbf{M}_{state}$ is the moment contribution from the state, i.e from α, β or q. This, iteration to find ω_0 and matching $\mathbf{M}$ is relatively easy to solve and can be done continuously during the simulation. This means that $\mathbf{x}_0$ and $\mathbf{u}_0$ are updated frequently and even though the speed drops drastically the control system can perform the maneuver with good accuracy. The corresponding control surface deflections are found via (3.4).

3.2.3 Wind vector roll

To perform a wind vector roll at rate ξ_0 first the corresponding p_0, q_0 and r_0 has to be evaluated. These are found by setting $\omega_0 = \xi_0 \frac{\mathbf{V}}{V}$, i.e $[p_0\ q_0\ r_0] = \xi_0[\cos\alpha_0\ 0\ \sin\alpha_0]$, according to (2.7) with $\beta_0 = 0$. Keeping this maneuver

at a constant rate no angular acceleration can exist, $\dot{\omega} = \mathbf{0}$ and the corresponding moments are found via (3.13). Due to the definition of ξ, α and β will be constant and (3.11) will be $\mathbf{0}$. Again the control surface settings are found via (3.4).

3.2.4 Linearization

After trimming of the aircraft it is possible to linearize $\mathbf{f}$ using a first order Taylor expansion evaluated around the equilibrium $\mathbf{x}_0$ previously calculated. The linear system of equations, which is the relation LQ is based on, has the form

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{3.15}$$

where $\mathbf{A} \in \mathbb{R}^{6\times 6}$ is the Jacobian matrix

$$\mathbf{A} = \begin{pmatrix} \frac{\partial \mathbf{f}_1}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_1} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_1} & \cdots & \frac{\partial \mathbf{f}_n}{\partial \mathbf{x}_n} \end{pmatrix}_{\big|\mathbf{x}_0}, \tag{3.16}$$

and $\mathbf{B} \in \mathbb{R}^{6\times 3}$ simply equals $\mathbf{g}(\mathbf{x}_0)$. The input vector $\mathbf{u} \in \mathbb{R}^3$ is is given by $[L\ M\ N]^T$. Each derivative in (3.16) is approximated with a central difference fraction with a second order of accuracy

$$\frac{\partial \mathbf{f}_i}{\partial \mathbf{x}_i} = \frac{\mathbf{f}(\mathbf{x}_{i+h}) - \mathbf{f}(\mathbf{x}_{i-h})}{2h} + \mathcal{O}(h^2) \tag{3.17}$$

where h is a perturbation. In Section 3.4.2 a simpler way of calculating $\mathbf{A}$ is explained and that method is used when designing the LQ controller. From this linear approximation one can for any state calculate the eigenvalues of $\mathbf{A}$ to investigate stability, see Fig. 3.5 for a rootlocus plot of the flying condition in Figure 3.1 to 3.4. The eigenvalue in (0,0) is due to pure integration corresponding to the state of ξ. All the other eigenvalues are located in the left half plane and the system is therefore (for this configuration) locally stable in a neighborhood of this $\mathbf{x}_0$.

3.3 Equilibrium

As described in Section 3.2, both LQ and SDRE stabilize the vector field $\mathbf{f}$ around a certain trimmed state and input, $\mathbf{x}_0$ and $\mathbf{u}_0$ respectively such that

$$\mathbf{0} = \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\mathbf{x}_0)\mathbf{u}_0.$$

Further, the SDRE formulation requires that these states and inputs are located in the origin. To fulfill this the variable substitutions, $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$ and $\tilde{\mathbf{u}} = \mathbf{u} - \mathbf{u}_0$ are made. Accordingly, when $\mathbf{x}$ and $\mathbf{u}$ approaches the

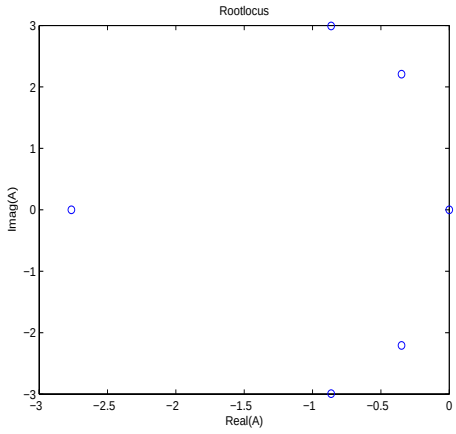

Figure 3.5: Rootlocusplot for **A** corresponding to trimmed condition for straight and level flight at altitude = 5000m and Mach = 0.5

trimmed conditions $\mathbf{x}_0$ and $\mathbf{u}_0$, then $\tilde{\mathbf{x}}$ and $\tilde{\mathbf{u}}$ approaches $\mathbf{0}$. Now, after these substitutions are made, (3.1) looks like

$$\begin{aligned}\dot{\tilde{\mathbf{x}}} &= \mathbf{f}(\tilde{\mathbf{x}}+\mathbf{x}_0)+\mathbf{g}(\tilde{\mathbf{x}}+\mathbf{x}_0)\mathbf{u} \\ &= \underbrace{\mathbf{f}(\tilde{\mathbf{x}}+\mathbf{x}_0)+\mathbf{g}(\tilde{\mathbf{x}}+\mathbf{x}_0)\mathbf{u}_0}_{\tilde{\mathbf{f}}(\tilde{\mathbf{x}})}+\underbrace{\mathbf{g}(\tilde{\mathbf{x}}+\mathbf{x}_0)}_{\tilde{\mathbf{g}}(\tilde{\mathbf{x}})}(\mathbf{u}-\mathbf{u}_0) \\ &= \tilde{\mathbf{f}}(\tilde{\mathbf{x}})+\tilde{\mathbf{g}}(\tilde{\mathbf{x}})\tilde{\mathbf{u}} \end{aligned} \tag{3.18}$$

and therefore

$$\mathbf{0}=\tilde{\mathbf{f}}(\mathbf{0})+\tilde{\mathbf{g}}(\mathbf{0})\mathbf{0}. \tag{3.19}$$

For $\dot{\alpha}$ and $\dot{\beta}$ in (2.11) we now have, assuming $\beta_0 = 0$ and $\tilde{\beta}$ small.

$$\begin{aligned}\dot{\tilde{\alpha}} = &\cos(\tilde{\alpha}+\alpha_0)\frac{Z(\tilde{\mathbf{x}}+\mathbf{x}_0)}{Vm}-\sin(\tilde{\alpha}+\alpha_0)\frac{X(\tilde{\mathbf{x}}+\mathbf{x}_0)}{Vm} \\ &-(\tilde{p}+p_0)\cos(\tilde{\alpha}+\alpha_0)\sin(\tilde{\beta})+(\tilde{q}+q_0)-(\tilde{r}+r_0)\sin(\tilde{\alpha}+\alpha_0)\sin(\tilde{\beta})\end{aligned} \tag{3.20}$$

and

$$\begin{aligned}\dot{\tilde{\beta}} = &\frac{Y(\tilde{\mathbf{x}}+\mathbf{x}_0)}{Vm}-\sin(\tilde{\alpha}+\alpha_0)\sin(\tilde{\beta})\frac{X(\tilde{\mathbf{x}}+\mathbf{x}_0)}{Vm}-\sin(\tilde{\alpha}+\alpha_0)\sin(\tilde{\beta})\frac{Z}{Vm} \\ &+(\tilde{p}+p_0)\sin(\tilde{\alpha}+\alpha_0)-(\tilde{r}+r_0)\cos(\tilde{\alpha}+\alpha_0).\end{aligned} \tag{3.21}$$

For p, q and r we have

$$\begin{aligned}
\dot{\tilde{p}} &= (\tilde{q}\tilde{r} + \tilde{q}r_0 + q_0\tilde{r} + q_0r_0)(I_{yy} - I_{zz})/I_{xx} + L(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{xx} \\
\dot{\tilde{q}} &= (\tilde{p}\tilde{r} + \tilde{p}r_0 + p_0\tilde{r} + p_0r_0)(I_{zz} - I_{xx})/I_{yy} + M(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{yy} \\
\dot{\tilde{r}} &= (\tilde{p}\tilde{q} + \tilde{p}q_0 + p_0\tilde{q} + p_0q_0)(I_{xx} - I_{yy})/I_{zz} + N(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{zz}.
\end{aligned} \tag{3.22}$$

For ξ the substitution yields

$$\begin{aligned}
\dot{\xi} = \frac{1}{V}\Big[&(\tilde{p} + p_0)(-g\sin\theta + X(\tilde{\mathbf{x}} + \mathbf{x}_0)/m) \\
&+(\tilde{q} + q_0)(g\cos\theta\sin\phi + Y(\tilde{\mathbf{x}} + \mathbf{x}_0)/m) \\
&+(\tilde{r} + r_0)(g\cos\theta\cos\phi + Z(\tilde{\mathbf{x}} + \mathbf{x}_0)/m) \\
&+V\cos(\tilde{\alpha} + \alpha_0)((\tilde{q} + q_0)(\tilde{r} + r_0)(I_{yy} - I_{zz})/I_{xx} + L(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{xx}) \\
&+V\sin(\tilde{\beta})((\tilde{r} + r_0)(\tilde{p} + p_0)(I_{zz} - I_{xx})/I_{yy} + M(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{yy}) \\
&+V\sin(\tilde{\alpha} + \alpha_0)((\tilde{p} + p_0)(\tilde{q} + q_0)(I_{xx} - I_{yy})/I_{zz} + N(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{zz})\Big]
\end{aligned} \tag{3.23}$$

and for Φ

$$\dot{\tilde{\Phi}} = \cos(\tilde{\alpha} + \alpha_0)(\tilde{p} + p_0) + \sin(\tilde{\beta})(\tilde{q} + q_0) + \sin(\tilde{\alpha} + \alpha_0)(\tilde{r} + r_0), \tag{3.24}$$

where $\cos(\tilde{\beta}) = 1$. As seen above, in each equation the substitution is not made in the terms for gravity. The reason for this is that the gravity is a function of θ and ϕ which are considered as parameters rather than states.

3.4 State Dependent Riccati Equation

3.4.1 Classical LQ

In classic linear control theory one common method is LQ-design which, in the state feedback case, applies to the class of linear systems as in (3.15), obtained by taking $\mathbf{f}(\mathbf{x}) = \mathbf{A}\mathbf{x}$ and $\mathbf{g}(\mathbf{x}) = \mathbf{B}$ in (3.1), where $\mathbf{A}$ and $\mathbf{B}$ are two constant matrices. The goal is then to minimize

$$J = \int_0^\infty \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{u}^T\mathbf{R}\mathbf{u}\,dt \tag{3.25}$$

subject to the condition

$$\dot{\mathbf{x}} = \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u}, \tag{3.26}$$

where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ and $\mathbf{R} \in \mathbb{R}^{m\times m}$ are, respectively, a positive semidefinite and a positive definite weighting matrix that act as "tuning parameters" for the method. Assuming that $(\mathbf{A}, \mathbf{B})$ is a stabilizable pair (Def. 3.2 below), the quantity J in (3.25) is minimized [1] when $\mathbf{u} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}\mathbf{x}$ where

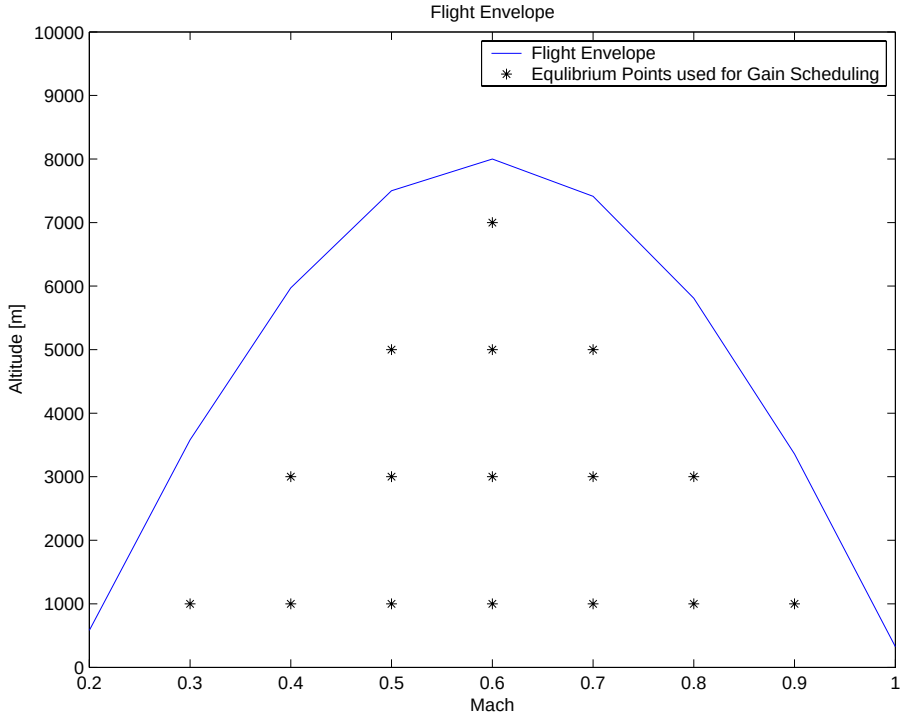

Figure 3.6: Flight envelope with equilibrium points used for Gain Scheduling.

$\mathbf{P} \in \mathbb{R}^{n \times n}$ is the (unique) positive definite solution to the (continuous-time) Riccati Equation

$$\mathbf{A}^T\mathbf{P} + \mathbf{P}\mathbf{A} - \mathbf{P}\mathbf{B}\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P} + \mathbf{Q} = \mathbf{0}. \tag{3.27}$$

This solution gives, for the cost function (3.25) with the chosen penalty matrices $\mathbf{Q}$ and $\mathbf{R}$, the optimal solution as a linear state feedback $\mathbf{u}^* = \mathbf{K}\mathbf{x}$ with gain matrix $\mathbf{K} = -\mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$, for the operation point used in the linearization $\mathbf{A} = \frac{\partial \mathbf{f}}{\partial \mathbf{x}}|_{\mathbf{x}_0}$, where $\mathbf{A} - \mathbf{B}\mathbf{K}$ is strictly stable (all eigenvalues in the left half plane). For each operating point in the envelope, see Fig 3.6 [1] and for each rate of each maneuver (a 100% right lateral stick setting requires a different gain than 50% right lateral stick setting), each aircraft configuration (weapons etc.) and for different aircraft masses (The mass decreases due to fuel consumption.) this gain $\mathbf{K}$ is precalculated. These gains are stored and during flight, gain scheduling is used to interpolate and to find the optimal feedback gain for the current flying condition.

[1] The flight envelope for the URAV considered in this thesis does not necessarily look like this. This figure is merely a schematic example of how it could look like.

3.4.2 SDRE

The idea in State Dependent Riccati Equation method is to rewrite (3.1) into the SDC form

$$\dot{\mathbf{x}} = \mathbf{A}(\mathbf{x})\mathbf{x} + \mathbf{B}(\mathbf{x})\mathbf{u}, \tag{3.28}$$

where the vector field $\mathbf{f}(\mathbf{x})$ is parameterized into the form $\mathbf{A}(\mathbf{x})\mathbf{x}$ for the current $\mathbf{x}$ and $\mathbf{B}(\mathbf{x})$ simply equals $\mathbf{g}(\mathbf{x})$. The SDC-parameterization can be performed in both a numerical and an analytical way. Both approaches are described below but only the latter one is used in practice in this thesis. It is necessary that this factorization $\mathbf{A}(\mathbf{x})$ together with $\mathbf{B}(\mathbf{x})$ is stabilizable, according to the following definition.

Definition ([1]) 3.2. A system $\mathbf{A}$ and $\mathbf{B}$ is stabilizable if there exists a matrix $\mathbf{L}$ such that $\mathbf{A} - \mathbf{BL}$ is stable.

This requirement is weaker than requiring controllability according to Def. 3.1 adapted to the SDRE case.

Definition ([8]) 3.3. The SDC parameterization in (3.28) is controllable if the pair ($\mathbf{A}(\mathbf{x})$,$\mathbf{B}(\mathbf{x})$) is controllable (Def. 3.1) for all $\mathbf{x}$.

If the system is controllable it is also stabilizable and controllability is investigated during all simulations.

An important fact to note is that the matrix $\mathbf{A}(0)$ is simply the Jacobian at 0 of the vector field $\mathbf{f}$ in (3.1), i.e.

$$\mathbf{A}(0) = \frac{\partial \mathbf{f}(\mathbf{x})}{\partial \mathbf{x}}\Big|_{\mathbf{x}=0}$$

see e.g. [8]. This means that LQ and SDRE asymptotically become the same method, as $\mathbf{x}$ becomes small. Now, since $\mathbf{A} = \mathbf{A}(\mathbf{x})$ and $\mathbf{B} = \mathbf{B}(\mathbf{x})$ the resulting algebraic Riccati equation and its solution, $\mathbf{P}$ are functions of $\mathbf{x}$,

$$\mathbf{A}(\mathbf{x})^T\mathbf{P}(\mathbf{x}) + \mathbf{P}(\mathbf{x})\mathbf{A}(\mathbf{x}) - \mathbf{P}(\mathbf{x})\mathbf{B}(\mathbf{x})\mathbf{R}(\mathbf{x})^{-1}\mathbf{B}(\mathbf{x})^T\mathbf{P}(\mathbf{x}) + \mathbf{Q}(\mathbf{x}) = \mathbf{0} \tag{3.29}$$

This state dependent Riccati equation, (3.29) has to be solved by the on-board computer and the online calculated input will also be state dependent,

$$\mathbf{u} = -\mathbf{R}(\mathbf{x})^{-1}\mathbf{B}^T(\mathbf{x})\mathbf{P}(\mathbf{x})\mathbf{x}. \tag{3.30}$$

Analytical SDC-factorization

The relations (3.20)–(3.23) is the vectorfield to parameterize into $\mathbf{A}(\tilde{\mathbf{x}})\tilde{\mathbf{x}}$. Constant terms, such as p_0, q_0 and r_0 might cause problems with singularity when doing the factorization. To cancel these possible singularities the vectorfield in (3.18) is subtracted by the term $\tilde{\mathbf{f}}(\mathbf{0}) = \mathbf{0}$ yielding

$$\tilde{\mathbf{f}}(\tilde{\mathbf{x}}) = \tilde{\mathbf{f}}(\tilde{\mathbf{x}}) - \tilde{\mathbf{f}}(\mathbf{0}) \tag{3.31}$$

which is, as will be shown, $\mathcal{O}(\tilde{\mathbf{x}})$ everywhere. This not only removes constant terms that might cause singularity, but also ensures that $\tilde{\mathbf{f}}(\mathbf{0}) = \mathbf{0}$. Earlier due to numerical uncertainties $\tilde{\mathbf{f}}(\mathbf{0})$ was only $\approx \mathbf{0}$. Further investigation of (3.31) yields

$$\tilde{\mathbf{f}}(\tilde{\mathbf{x}}) - \tilde{\mathbf{f}}(\mathbf{0}) \quad = \quad \mathbf{f}(\tilde{\mathbf{x}} + \mathbf{x}_0) - \mathbf{f}(\mathbf{x}_0) + \mathbf{g}(\tilde{\mathbf{x}} + \mathbf{x}_0)\mathbf{u}_0 - \mathbf{g}(\mathbf{x}_0)\mathbf{u}_0 \tag{3.32}$$

where the last two terms cancel eachother except for a small remainder in row six(except when controlling Φ rather than ξ, then also this row is zero). This (3.32) is $\mathcal{O}(\tilde{\mathbf{x}})$ and is now ready for parameterization.

Example As an example let us consider (3.22), where this singularity occurs explicitly. Here the constant terms $q_0 r_0$ etcetera cannot be divided with any element of $\tilde{\mathbf{x}}$ since it will be singular for $\tilde{\mathbf{x}} \to \mathbf{0}$. Therefor the parameterization will instead be made on

$$\mathbf{I}\dot{\tilde{\omega}} = -(\tilde{\omega} + \omega_0) \times \mathbf{I}(\tilde{\omega} + \omega_0) + \mathbf{M}(\tilde{\mathbf{x}} + \mathbf{x}_0) - (-(\omega_0) \times \mathbf{I}(\omega_0) + \mathbf{M}(\mathbf{x}_0)) \tag{3.33}$$

or

$$\begin{aligned}
\dot{\tilde{p}} &= (\tilde{q}\tilde{r} + \tilde{q}r_0 + q_0\tilde{r})(I_{yy} - I_{zz})/I_{xx} + L(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{xx} - L(\mathbf{x}_0)/I_{xx} \\
\dot{\tilde{q}} &= (\tilde{p}\tilde{r} + \tilde{p}r_0 + p_0\tilde{r})(I_{zz} - I_{xx})/I_{yy} + M(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{yy} - M(\mathbf{x}_0)/I_{yy} \\
\dot{\tilde{r}} &= (\tilde{p}\tilde{q} + \tilde{p}q_0 + p_0\tilde{q})(I_{xx} - I_{yy})/I_{zz} + N(\tilde{\mathbf{x}} + \mathbf{x}_0)/I_{zz} - N(\mathbf{x}_0)/I_{zz}
\end{aligned} \tag{3.34}$$

where these constant terms are cancelled. Now the parameterization can readily be made in $\tilde{p}, \tilde{q}$ and $\tilde{r}$ for the first terms and, as will be shown later, in $\tilde{\alpha}$, $\tilde{\beta}$ or in $\tilde{p}, \tilde{q}, \tilde{r}$ for the aerodynamic moments.

In (3.20) the term $\cos(\tilde{\alpha} + \alpha_0)Z(\tilde{\mathbf{x}} + \mathbf{x}_0)$, for example, is $\mathcal{O}(\tilde{\mathbf{x}}^{\mathbf{0}})$ and its parameterization in $\tilde{\alpha}$ will be singular for $\tilde{\alpha} \to 0$. The substraction by $\cos(\alpha_0)Z(\alpha_0)$ will however, as shown below, make this term $\mathcal{O}(\tilde{\mathbf{x}})$. First let us consider the Taylor expansion of $\cos(x)$

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} + \mathcal{O}(x^6) \tag{3.35}$$

which is an even function starting with a constant = 1. The force $Z(\tilde{\alpha} + \alpha_0)$ is approximated with a polynomial according to (2.26) which includes a constant. Now

$$\cos(\tilde{\alpha}+\alpha_0)Z(\tilde{\alpha}+\alpha_0) = \Big(1-\frac{(\tilde{\alpha}+\alpha_0)^2}{2!}+\frac{(\tilde{\alpha}+\alpha_0)^4}{4!}+\dots\Big)$$
$$\Big(c_0+c_1(\tilde{\alpha}+\alpha_0)+c_2(\tilde{\alpha}+\alpha_0)^2+\dots+g\cos(\theta)\cos(\phi)\Big) \tag{3.36}$$

and

$$\cos(\alpha_0)Z(\alpha_0) = \Big(1-\frac{(\alpha_0)^2}{2!}+\frac{(\alpha_0)^4}{4!}+\dots\Big)$$
$$\Big(c_0+c_1(\alpha_0)+c_2(\alpha_0)^2+\dots+g\cos(\theta)\cos(\phi)\Big) \tag{3.37}$$

where the gravity g is a function of the true state $\mathbf{x}$. Subtracting (3.37) from (3.36) and sorting the terms we get

$$c_0\Big[\Big(1-\frac{(\tilde{\alpha}+\alpha_0)^2}{2!}+\frac{(\tilde{\alpha}+\alpha_0)^4}{4!}+\dots\Big)-\Big(1-\frac{(\alpha_0)^2}{2!}+\frac{(\alpha_0)^4}{4!}+\dots\Big)\Big] =$$
$$c_0\Big[-\frac{\tilde{\alpha}^2+2\tilde{\alpha}\alpha_0}{2!}+\frac{\tilde{\alpha}^4+4\tilde{\alpha}^3\alpha_0+6\tilde{\alpha}^2\alpha_0^2+4\tilde{\alpha}\alpha_0^3}{4!}+\dots\Big], \tag{3.38}$$

$$c_1\Big[\Big((\tilde{\alpha}+\alpha_0)-\frac{(\tilde{\alpha}+\alpha_0)^3}{2!}+\frac{(\tilde{\alpha}+\alpha_0)^5}{4!}+\dots\Big)-\Big(\alpha_0-\frac{(\alpha_0)^3}{2!}+\frac{(\alpha_0)^3}{4!}+\dots\Big)\Big] =$$
$$c_1\Big[\tilde{\alpha}-\frac{\tilde{\alpha}^3+3\tilde{\alpha}^2\alpha_0+3\tilde{\alpha}\alpha_0^2}{2!}+\frac{\tilde{\alpha}^5+5\tilde{\alpha}^4\alpha_0+10\tilde{\alpha}^3\alpha_0^2+10\tilde{\alpha}^2\alpha_0^3+5\tilde{\alpha}\alpha_0^4}{4!}+\dots\Big] \tag{3.39}$$

and

$$g[\cos(\theta)\cos(\phi)\Big(\frac{\tilde{\alpha}^2+2\tilde{\alpha}\alpha_0}{2!}+\dots\Big)], \tag{3.40}$$

which all are $\mathcal{O}(\tilde{\alpha})$. If necessary, for higher accuracy between $\mathbf{f}(\mathbf{x})$ and $\mathbf{A}(\mathbf{x})\mathbf{x}$, more coefficients are added and similar calculations are needed for these. The SDC parameterization is now made according to above and the resulting matrix $A(x)$ is found in Appendix A.

Numerical SDC-factorization

It is also possible to do the factorization numerically to determine the elements of $\mathbf{A}(\mathbf{x})$. However the the vector field $\mathbf{f}(\mathbf{x})$ provides only n equations to determine the n^2 elements of $\tilde{\mathbf{A}}(\tilde{\mathbf{x}})$. Hence, additional relations are needed in order to determine $\tilde{\mathbf{A}}(\tilde{\mathbf{x}})$. The additional information can be obtained by using linearly independent probing vectors $\varsigma_2,\dots,\varsigma_n$. Typically the probing

vectors are obtained from $\tilde{\mathbf{x}}$ by adding small perturbation vectors $\sigma_2, \ldots, \sigma_n$ as

$$\varsigma_2 = \tilde{\mathbf{x}} + \boldsymbol{\sigma}_2, \quad \varsigma_3 = \tilde{\mathbf{x}} + \boldsymbol{\sigma}_3, \quad \ldots, \varsigma_n = \tilde{\mathbf{x}} + \boldsymbol{\sigma}_n. \tag{3.41}$$

The matrix $\tilde{\mathbf{A}}(\tilde{\mathbf{x}})$ is then obtained from

$$[\tilde{\mathbf{f}}(\tilde{\mathbf{x}}), \tilde{\mathbf{f}}(\varsigma_2), \ldots, \tilde{\mathbf{f}}(\varsigma_n)] = \tilde{\mathbf{A}}(\tilde{\mathbf{x}})[\tilde{\mathbf{x}}, \varsigma_2, \ldots, \varsigma_n], \tag{3.42}$$

which is well defined if the vectors $\tilde{\mathbf{x}}, \varsigma_2, \ldots, \varsigma_n$ are linearly independent. If the perturbations $\boldsymbol{\sigma}_2, \ldots, \boldsymbol{\sigma}_n$ are small, the matrix $\tilde{\mathbf{A}}(\tilde{\mathbf{x}})$ is close to the Jacobian of $\tilde{\mathbf{f}}$ at $\tilde{\mathbf{x}} = 0$, i.e.

$$\tilde{\mathbf{A}}(\tilde{\mathbf{x}}) \simeq \frac{\partial \tilde{\mathbf{f}}(\tilde{\mathbf{x}})}{\partial \tilde{\mathbf{x}}}\Big|_{\tilde{\mathbf{x}}=0}. \tag{3.43}$$

These ideas are described in [9].

3.5 The Q(x) and R(x) matrices

To fully understand how to tune the weighting matrices in (3.25) one has to be familiar with the concept feedback gain. This gain is related to the control error $\tilde{\mathbf{x}}$ and the feedback gain is of course increased with an increasing $\mathbf{Q}(\mathbf{x})$ and also with an increasing error $\tilde{\mathbf{x}}$. Therefor the necessary $\mathbf{Q}(\mathbf{x})$ to perform a maneuver properly is decreased with an increase in $\mathbf{x}$. In reality there are other aspects that influences the needed weighting matrices but the statement above is a good start. Further $\mathbf{Q}(\mathbf{x})$ may be used to keep a states deviations small. For example in a step in α. When the step is initialized, $\tilde{\alpha}$ is abruptly increased and the control system tries to track this commanded α_0. During this it is also likely that β deviates from zero somewhat and this maximum deviation may be decreased by a higher weight. Ideally a step in α would be a pure longitudinal motion but there always exists imperfections and small but yet significant asymmetries giving contributions in β. This differ in β is also probable to, when trying to reduce this, induce an oscillating behaviour and which frequency is increased with a higher weight. The contradiction is obvious and there is a trade-off in the choice in $\mathbf{Q}(\mathbf{x})$. The maximum deviation is often relatively small, percentage of one degree possibly, and the weight is chosen to a value where the deviations are quite small and the frequency of the oscillations are reduced. Of course the main performance of, in this case, α is most important and has to be considered in the tuning of $\beta : s$ weight. Further controlling fast states, for example ξ, is different from controlling slow states, for example Φ. To maintain a constant high rollrate ξ will, as stated below, require a weight $\propto 100$ whereas when commanding Φ the needed weight is $\propto 1$. For the matrix $\mathbf{R}(\mathbf{x})$ punishing the input in the optimization the weights are chosen much smaller than the weights for the states since the control surfaces are allowed to fluctuate arbitrarily within the interval in table 2.1 and with the dynamics in (2.17).

3.5.1 SDRE algorithm

The algorithm for SDRE control design is visualized in Fig. 3.5.1 and also explained below.

1. Obtain state vector $\mathbf{x}$ from system
   - $\tilde{\mathbf{x}} = \mathbf{x} - \mathbf{x}_0$
2. Compute the SDC matrices $\mathbf{A}(\tilde{\mathbf{x}})$ and $\mathbf{B}(\tilde{\mathbf{x}})$
3. Together with $\mathbf{Q}$ and $\mathbf{R}$ solve the Riccati equation
4. Compute the state dependent gain $\mathbf{K} = \mathbf{R}^{-1}\mathbf{B}^T\mathbf{P}$
5. Formulate the state dependent feedback $\tilde{\mathbf{u}} = -\mathbf{K}\tilde{\mathbf{x}}$
   - $\mathbf{u} = \tilde{\mathbf{u}} + \mathbf{u}_0$
6. $\mathbf{u}$ to system

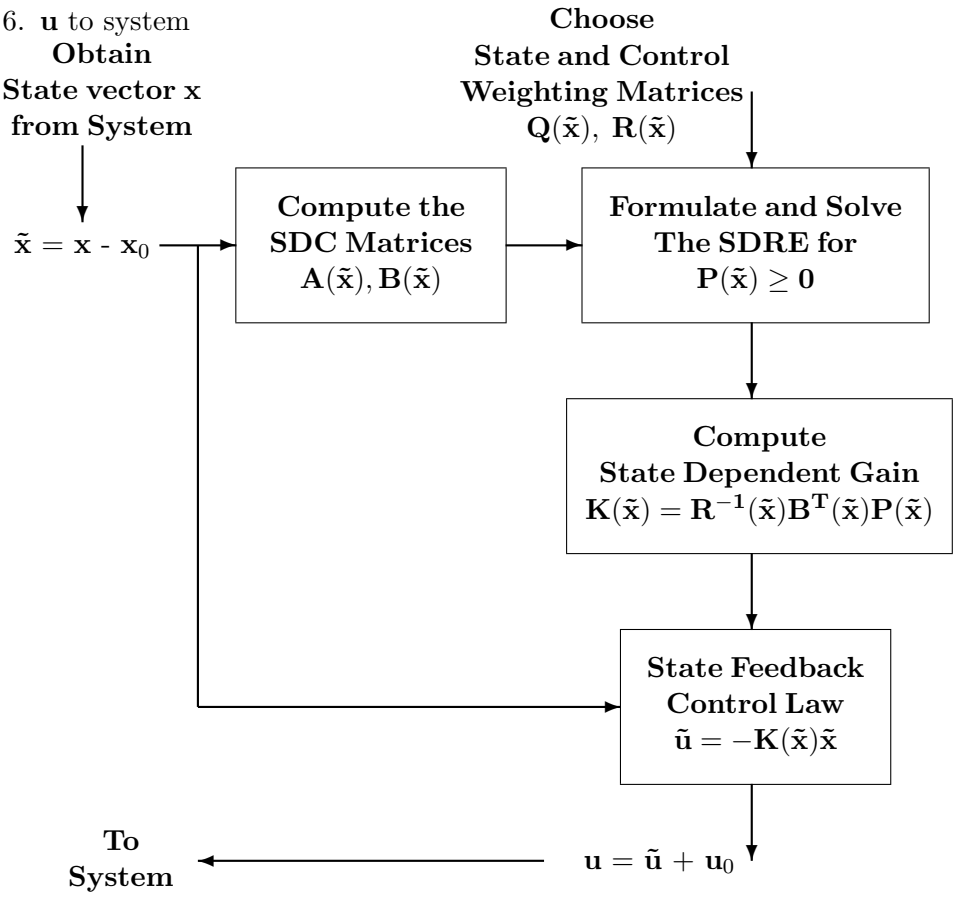

Figure 3.7: Figure showing the SDRE algorithm.

Chapter 4

Results

To test the control system a few maneuvers are made for different flying conditions and performance is investigated. The tests are done using SDRE, LQ and also the nonlinear control design method called TimeScale Separation or TSS. The theories behind this method are found in appendix B. The maneuvers performed, at different flying conditions, are

- A step in α performing a looping as in Figure 4.1.

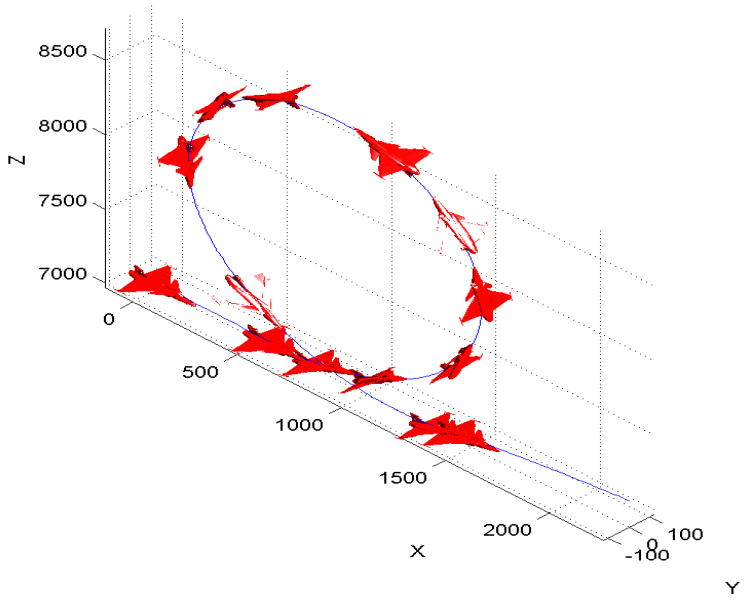

Figure 4.1: 3D plot of a looping.

- A step in α and in ξ performing a wind vector roll as in Figure 4.2.

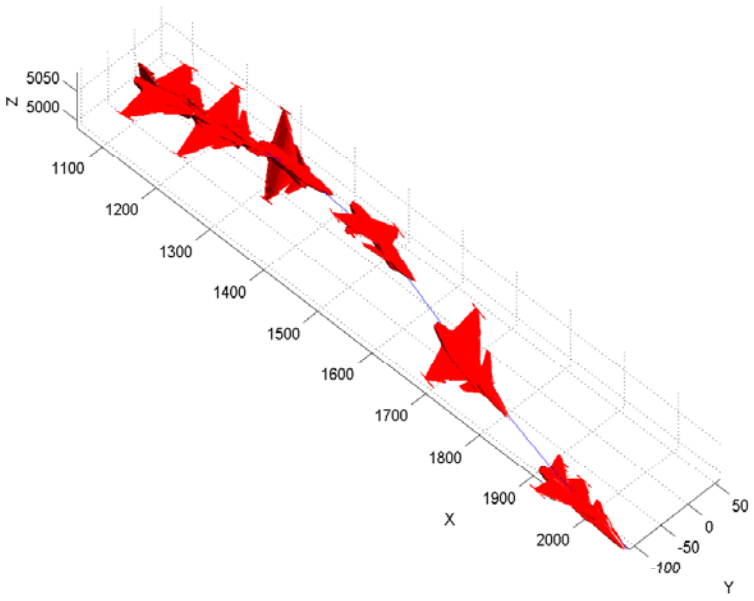

Figure 4.2: 3D plot of a wind vector roll 360 degrees.

- A step in α and a pulse in ξ performing a wind vector roll first in positive, then in negative direction.
- A step in α and in Φ as in Figure 4.3

Each maneuver starts from and returns to level horizontal flight. Rootlocus plots for each maneuver are, for both the open and closed loop system, found in Appendix C

4.1 LQ versus SDRE performance

4.1.1 Looping

The aircraft starts at trimmed level flight at altitude 7000 meters and Mach = 0.8. After two seconds, steps in α and q are made and a looping maneuver begins. The tracking of q is not as important as keeping α constant during the maneuver. Nor is it as important as keeping β as close to zero as possible. For both LQ and SDRE, $Q = diag([3000\ 20000\ 1\ 1\ 1\ 1])$ and $R = diag(1e^{-9}[1\ 1\ 1\ ])$ and Figure 4.4 shows the results for SDRE and LQ respectively. The Mach number drops drastically when the step is made. This is natural since thrust is set to a constant value and when 50% of the looping is performed and the aircraft has reached its highest point the speed increases again. The sideslip angle keeps within satisfying limits < 0.01 degrees. The total similarity in performance may be explained by examining

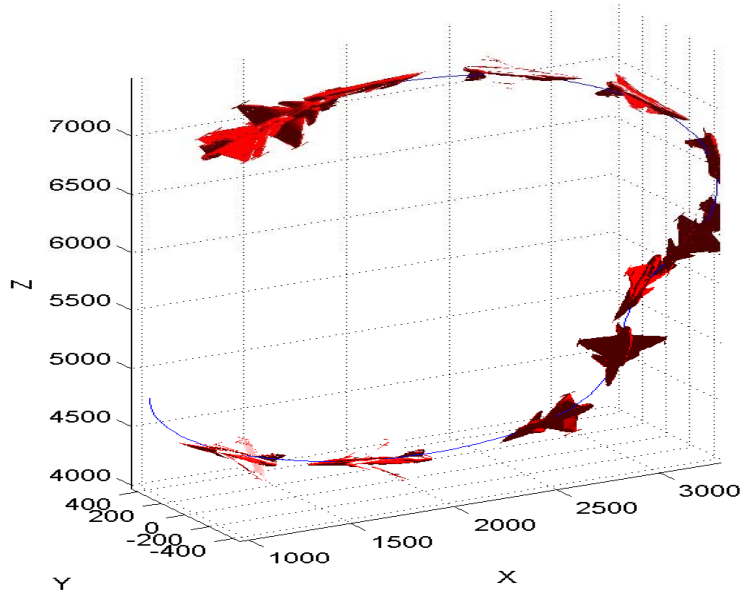

Figure 4.3: 3D plot of a wind vector roll $180° \rightarrow -180° \rightarrow 0°$. (Flying at $\Phi = \pm 180°$ with a high α yields an inverted loop.)

(2.8) and (2.11). During a step in α, only q is of significant size and every other variable is relatively small. Further the forces and moments, X, Z and M are linear (Y, L and N are small) within the angle of attacks of this maneuver.

The corresponding control surface deflections are found in Figure 4.5 where of course it is mainly the elevator, δ_e that has been used. Notice that shortly after 35 seconds a saturation has occurred at 10°. Doing a higher step in $\alpha \approx 20°$, will cause a saturation also at the initial climb at five seconds, as seen in Figure 4.6. This step in α is to high with the consequence that the speed drops too much and the looping cannot be completed. This however shows a saturation at $-20°$ shortly after 5 seconds even though the elevator, δ_e has according to Table 2.1 an allowed interval between $-30°$ and 10°. This indicates, as shown in Figure 4.7, that the pitching moment is a nonlinear function of δ_e with a maximum in the neighborhood of $-20°$. For moderate deflections the moment behaves linearly.

4.1.2 High angle of attack wind vector roll

Performing a wind vector roll at a relatively high α is an interesting maneuver since it there will be a tradeoff between performing well in ξ and in α and β. Two different maneuvers are examined and the first one is a complete 360° wind vector roll. The second one is a pulse in ξ first to a

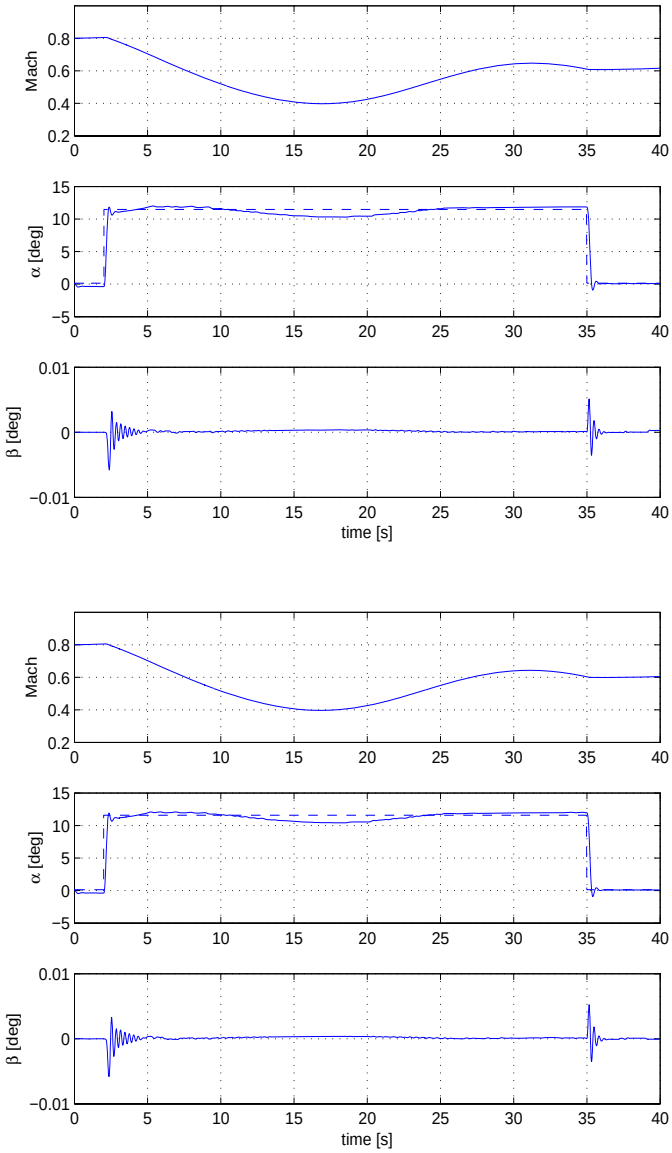

Figure 4.4: Mach, α and β during looping at altitude 7000m. SDRE and LQ.

positive value then abruptly to the same but negative value. Both cases uses $Q = diag([15000\ 20000\ 1\ 1\ 1\ 500])$ and $R = diag(1e^{-9}[1\ 1\ 1\ ])$. First let us consider the complete roll. Here a step in ξ is made shortly after a step in α. This ξ will be kept until a complete roll has been performed. The time for doing this is of course $t = 2\pi/\xi$ that is, approximately 6.28 seconds if ξ

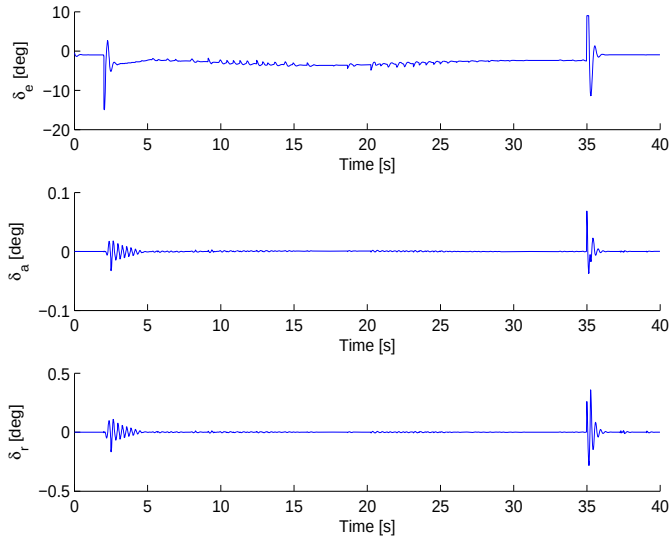

Figure 4.5: Control surface deflection needed to perform the looping using SDRE..

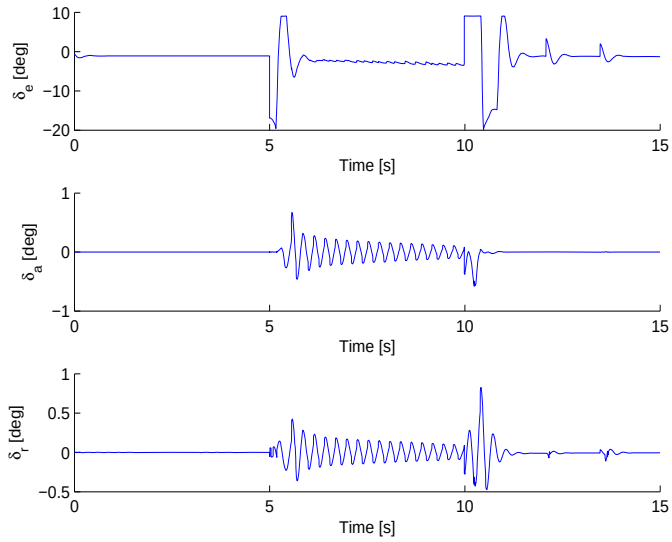

Figure 4.6: Control surface deflection needed to initialize a looping using SDRE for a higher step in α..

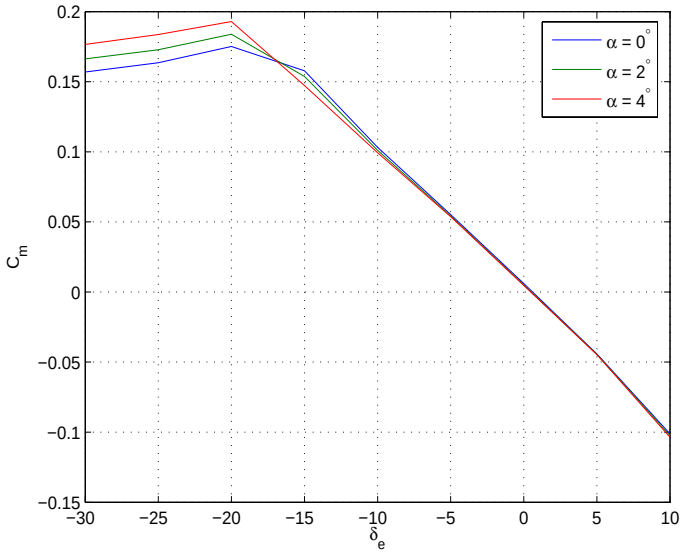

Figure 4.7: Pitching moment as a function of δ_e for $\alpha = 0$, 2 and 4°, all for $\beta = 0$ ∘.

is kept at 1 rad/s. During this maneuver it is important to damp out any deviation in α and β and also to perform the roll properly. Figure 4.8 and 4.9 shows the step in ξ and the behaviour for α and β respectively.

For the second case, first a step in α is made. As soon as the oscillations has settled a step in ξ is made, first to a positive value then abruptly to the same but negative value. Finally a step back to initial conditions is made. See Figure 4.10 and 4.11. Also here $Q = diag([15000\ 20000\ 1\ 1\ 1\ 500])$ and $R = diag(1e^{-9}[1\ 1\ 1\ ])$.

For both maneuvers α and β deviates from original values during the roll in contrary with theory which said that α and β were supposed to be constant. However the deviations are fairly small and the controller manages to damp the deviations.

4.1.3 Wind vector roll angle command

Here the wind vector angle Φ is commanded rather than its derivative ξ and a wind vector roll to $180° \rightarrow -180° \rightarrow 0°$ is commanded. This is, as before, performed at a high angle of attack. During this maneuver $Q = diag([3000\ 20000\ 1\ 1\ 1\ 1])$ and $R = diag(1e^{-9}[1\ 1\ 1\ ])$ as before. Figure 4.12 and 4.13 shows the response in Φ and in Mach, α and β for both SDRE and LQ. Yet again the results are surprisingly similar.

Φ is in comparison with ξ a slow state and relatively easy to track and

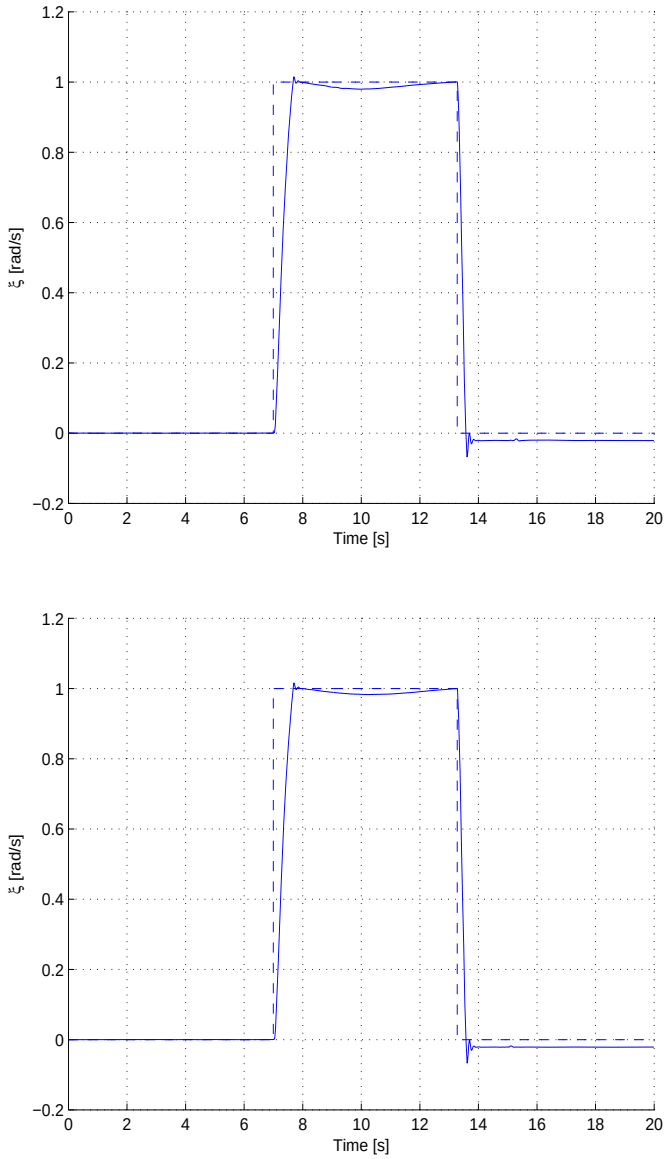

Figure 4.8: Wind vector roll rate ξ for SDRE and LQ at a high angle of attack.

both SDRE and LQ follows the commanded value smoothly even with the small gain in $\mathbf{Q}_\Phi = 1$ for commands in Φ compared to $\mathbf{Q}_\xi = 500$ for commands in ξ. It is possible to increase the gain for Φ, decreasing the risetime somewhat, but it is not recommended since it will have a negative effect in α and β.

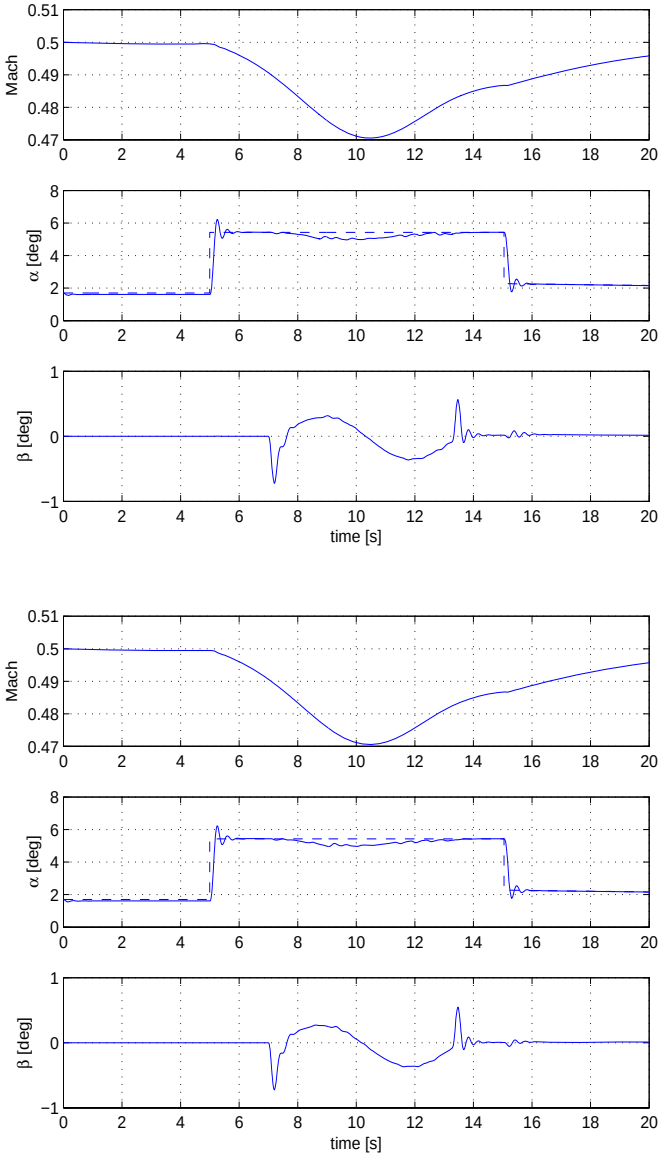

Figure 4.9: Mach, α and β during a complete wind vector roll using SDRE and LQ.

4.2 SDRE & LQ versus TSS

The maneuvers performed in section 4.1.2 and 4.1.3 are also tested using TSS (in its two variants) and below follows the results in comparison to SDRE and LQ. The histories for commanded change α_c in angle of attack,

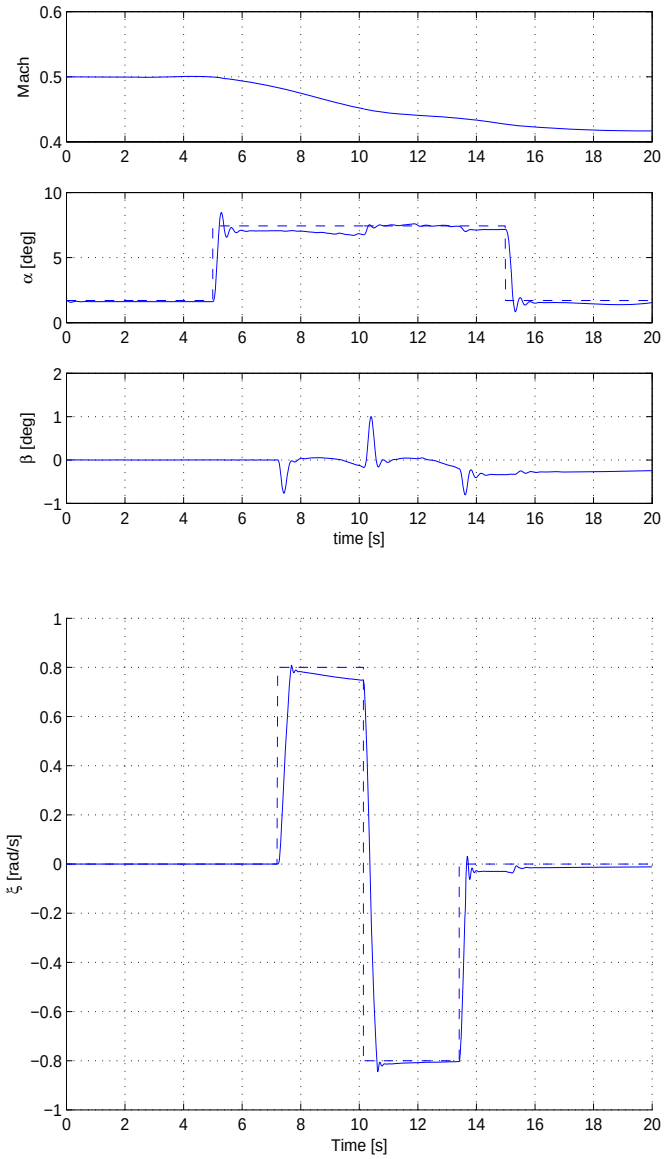

Figure 4.10: Mach, α, β and ξ for high angle of attack wind vector roll using SDRE.

commanded change Φ_c and commanded change ξ_c in wind vector roll rate are shown in Fig. 4.14. Because of numerical issues, the TSS method for step in bank angle Φ was performed with a finite slope of the commanded bank angle Φ_c during the changes in Φ_c and with a slightly smaller "swing". In Table 4.1 the parameters used in the controllers are shown for the two

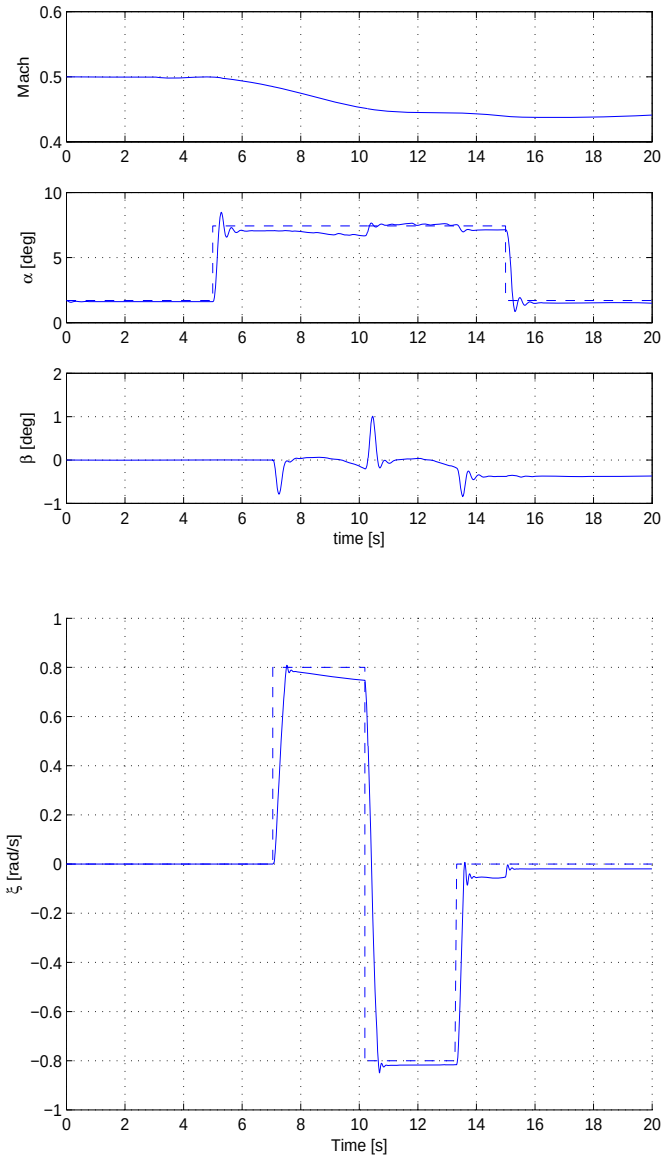

Figure 4.11: Mach, α, β and ξ for high angle of attack wind vector roll using LQ.

different maneuvers considered here. Fig. 4.15 to 4.17 shows the results when commanding ξ and Fig. 4.18 to 4.20 when controlling Φ.

For each maneuver and method the corresponding commanded control surface setting are found in Fig. 4.21

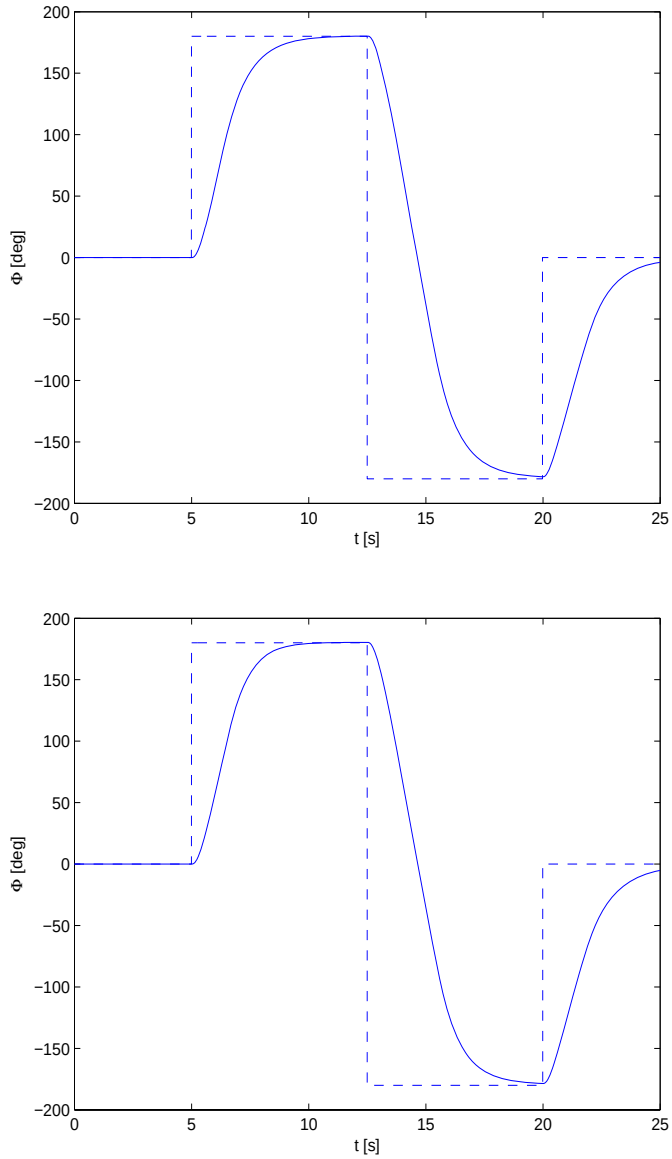

Figure 4.12: Φ for high angle of attack wind vector roll, SDRE and LQ.

4.3 Monte Carlo simulation

To verify if a model is robust, i.e. safe against modelling errors, one can perform a so called Monte Carlo simulation, [24, 25]. This simulation runs the model with a number of stochastic user-limited modelling errors. The results from each simulation is compared with the results from the nominal

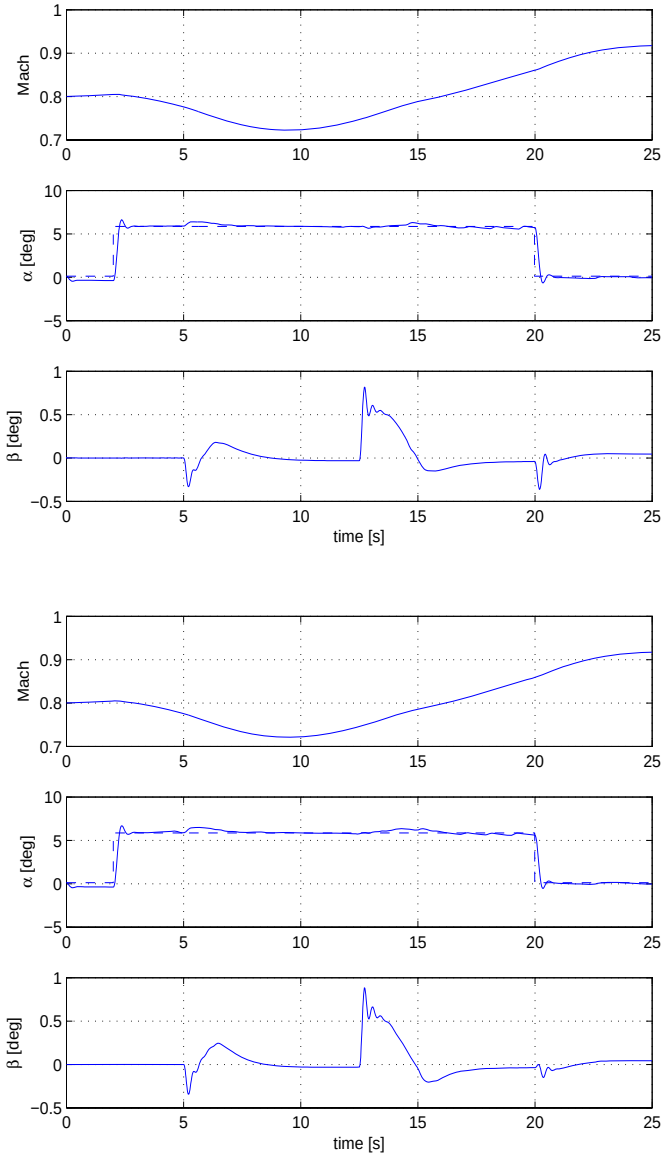

Figure 4.13: Mach, α, β for high angle of attack wind vector roll, SDRE and LQ.

model and a statistical robustness criterion is formulated. The modelling errors used here are within $\pm 40\%$ relative to the nominal value according to

$$C = (1 + \delta)C \tag{4.1}$$

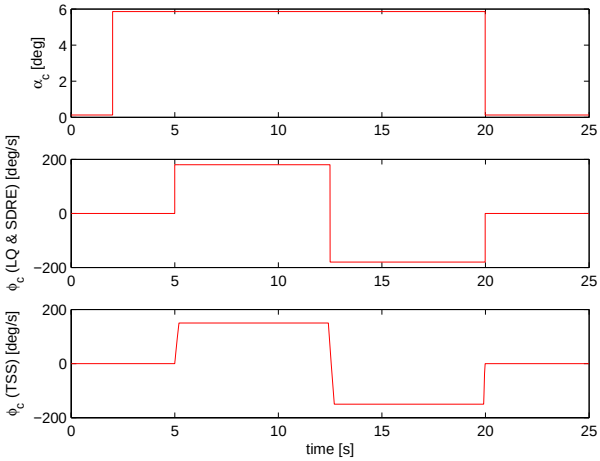

Figure 4.14: Commanded value α_c for the angle of attack α (top) and commanded value ϕ_c for the bank angle ϕ in the bank angle control maneuver simulations (SDRE, LQ in the middle, and TSS bottom). The commanded value ϕ_c used for the TSS method has a swing of $\pm 150°$, rather than $\pm 180°$ for the SDRE and LQ methods, and the flanks of the transition have finite derivatives.

where $\delta \in$ [-0.4 0.4] is the modelling error limit and C is the coefficients to be modified in the simulation. These coefficients, C is the aerodynamic coefficients used in the matrices $\mathbf{A}(\mathbf{x})$ and $\mathbf{B}(\mathbf{x})$, C_X, C_Y, C_Z, C_l, C_m, C_n due to wind vector and C_Y, C_Z, C_l, C_m, C_n due to ω (C_X is independent of ω) and C_l, C_m, C_n due to the control surfaces. There exists other properties, than aerodynamics, which are difficult to model, for example the matrix of inertia. However only the 14 modelling errors in aerodynamics are considered here. The stochastic robustness cost function (4.2) chosen to guide the design is a weighted quadratic sum of n probabilities of design metric violations.

$$J = \sum_{j=1}^{n} w_j P_j^2. \tag{4.2}$$

where w_j are costfunctions, $P_j \in [0, 1]$is the probability that $j : th$ criteria will fail and n is the total number of criteria. Table 4.2 shows a list of maneuvers performed at a set of flying conditions and Table 4.3 shows for each maneuver the corresponding criteria and weights used in (4.2).

The criteria, overshoot, settling time and steady state error and overall performance, of each maneuver and δ is evaluated relative to the nominal model. Steady state error is the offset value between the commanded value and the actual value after all oscillations been damped out. Overshoot and

Parameter	Value	Explanation
T_p, T_q, T_r	$0.1, 0.1, 0.1$	Diag. of $\mathbf{A_y}$ in TSS, TSSff, both man.
$T_\alpha, T_\beta, T_\phi$	$0.3, 0.3, 1.7$	Diag. of $\mathbf{A_y}$ in TSS, bank ang. man.
T_α, T_β, T_ξ	$0.5, 0.5, 0.002$	Diag. of $\mathbf{A_y}$ in TSSff, vel. vec. roll man.
$Q_\alpha, Q_\beta, Q_\phi$	$3000, 20000, 1$	Diag. of $\tilde{\mathbf{Q}}(\tilde{\mathbf{x}})$ SDRE, LQ, bank ang. man.
Q_p, Q_q, Q_r	$1, 1, 1$	Diag. of $\tilde{\mathbf{Q}}(\tilde{\mathbf{x}})$ SDRE, LQ, bank ang. man.
Q_α, Q_β, Q_ξ	$7500, 2500, 250$	Diag. of $\tilde{\mathbf{Q}}(\tilde{\mathbf{x}})$ SDRE, LQ, vel. vec. roll man.
Q_p, Q_q, Q_r	$0.5, 0.5, 0.5$	Diag. of $\tilde{\mathbf{Q}}(\tilde{\mathbf{x}})$ SDRE, LQ, vel. vec. roll man.

Table 4.1: Parameters used in the simulations for the control methods, SDRE, LQ, TSS and TSSff. All matrices in the table are diagonal matrices. The weight matrices $\tilde{\mathbf{Q}}(\tilde{\mathbf{x}})$ ans $\tilde{\mathbf{R}}(\tilde{\mathbf{x}})$ were the same for SDRE and LQ, for any given maneuver, and the weight matrix $\tilde{\mathbf{R}}(\tilde{\mathbf{x}})$ was equal to $10^{-12}\mathbf{I}$ at all times.

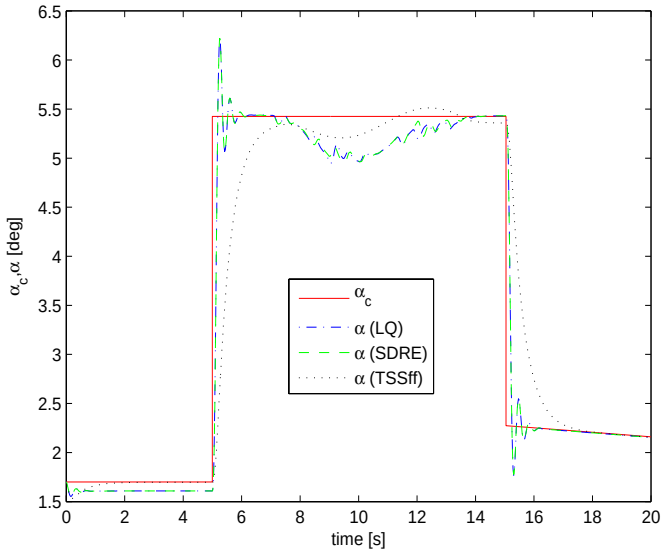

Figure 4.15: α for high angle of attack wind vector roll commanding ξ. SDRE, LQ and TSS.

settling time are defined as in Fig. 4.22 where steady state error equals zero.

To visualize the result a plot containing the response for the nominal model is plotted together with the responses for the modified coefficients is made. See Figure 4.23 for an example when performing a partial looping. (Only partial in an effort to reduce the simulation time in the already time demanding Monte Carlo simulation). The maximum relative overshoot is approximately 10%, the steady offset as well as the settling time is satisfying.

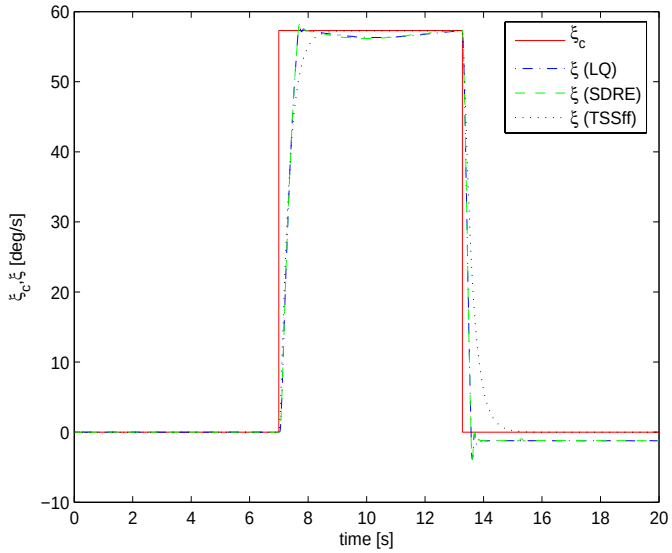

Figure 4.16: ξ for high angle of attack wind vector roll commanding ξ. SDRE, LQ and TSS.

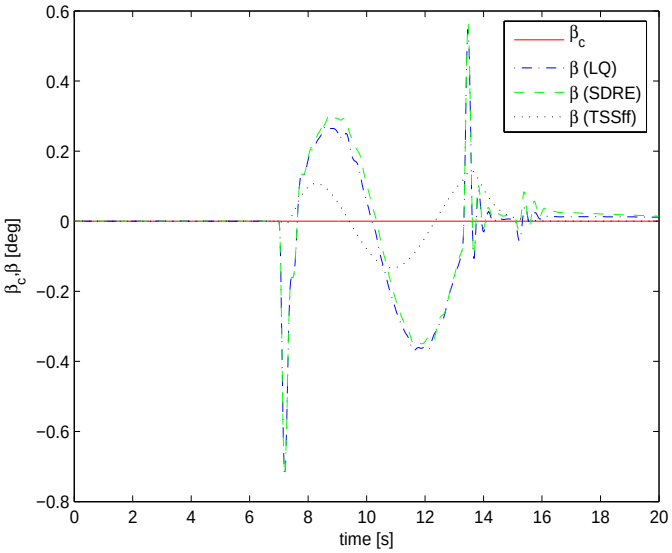

Figure 4.17: β for high angle of attack wind vector roll commanding ξ. SDRE, LQ and TSS.

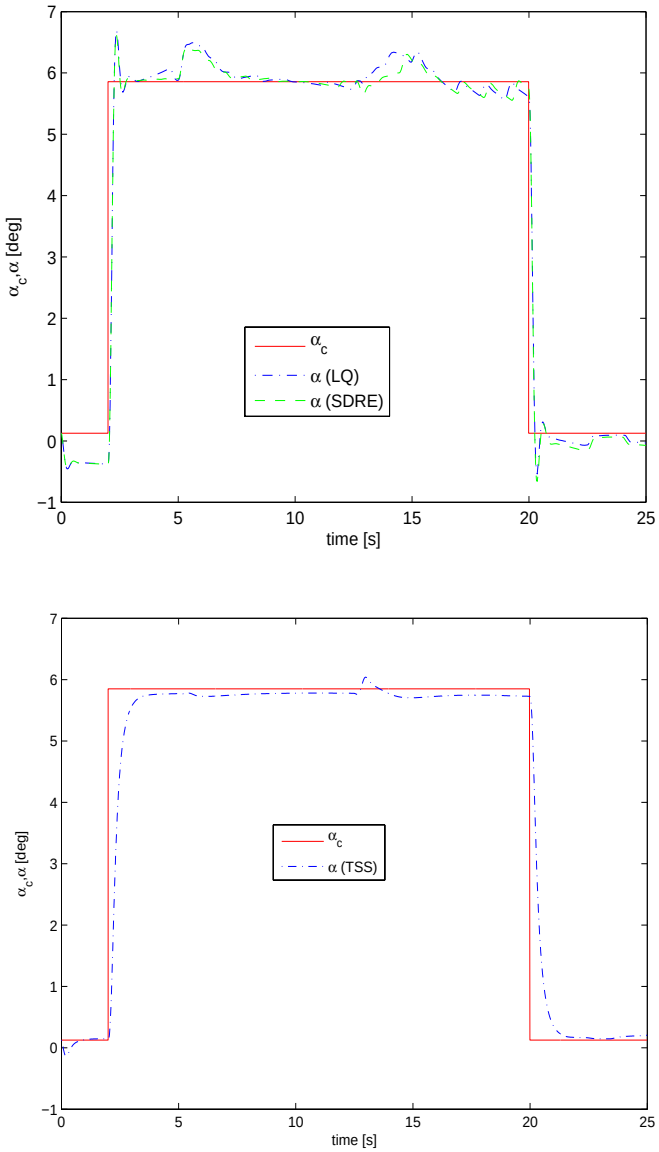

Figure 4.18: α for high angle of attack wind vector roll commanding Φ. Above: SDRE and LQ. Below: TSS.

Using the weights in Table 4.3 and the probabilities from Table 4.4, (4.2) yields a weighted sum of $J_{SDRE} = 0.0533$ and $J_{LQ} = 0.0406$ for SDRE and LQ respectively. The main contributer to these values are the instabilities that occur during maneuver 2. For SDRE the probability is $4/200 = 0.02$

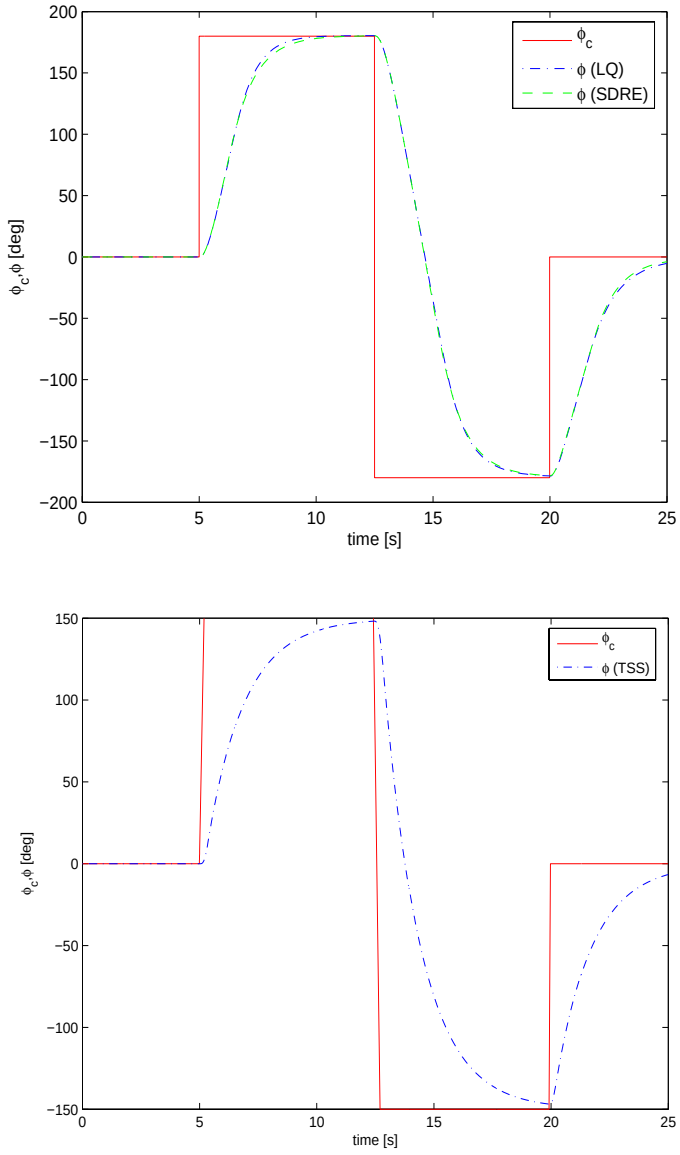

Figure 4.19: Φ for high angle of attack wind vector roll commanding Φ. Above: SDRE and LQ. Below: TSS.

and for LQ $2/168 = 0.0114$. This yields a J which is somewhat smaller for LQ but since the total number of simulations for LQ is less than for SDRE (168 versus 200) this small difference may be due to statistical uncertainties. Probably, as the number of simulations goes to ∞, the probabilities for SDRE and LQ will converge. However both methods yields values of J

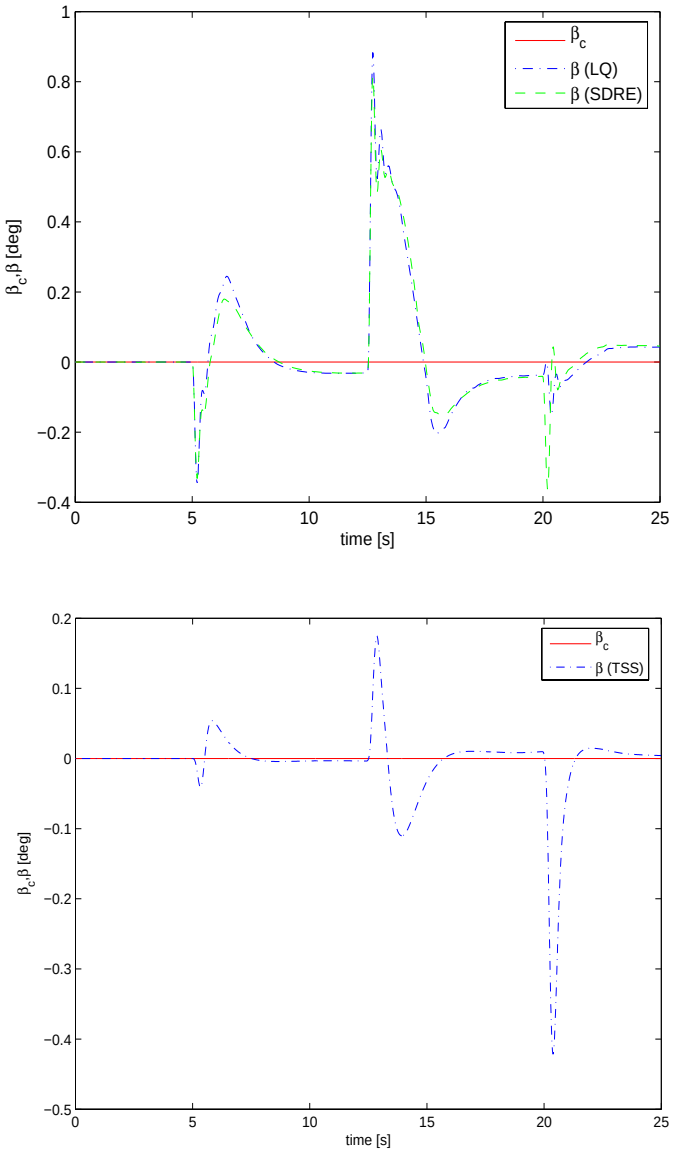

Figure 4.20: β for high angle of attack wind vector roll commanding Φ. Above: SDRE and LQ. Below: TSS.

indicating that the model is robust.

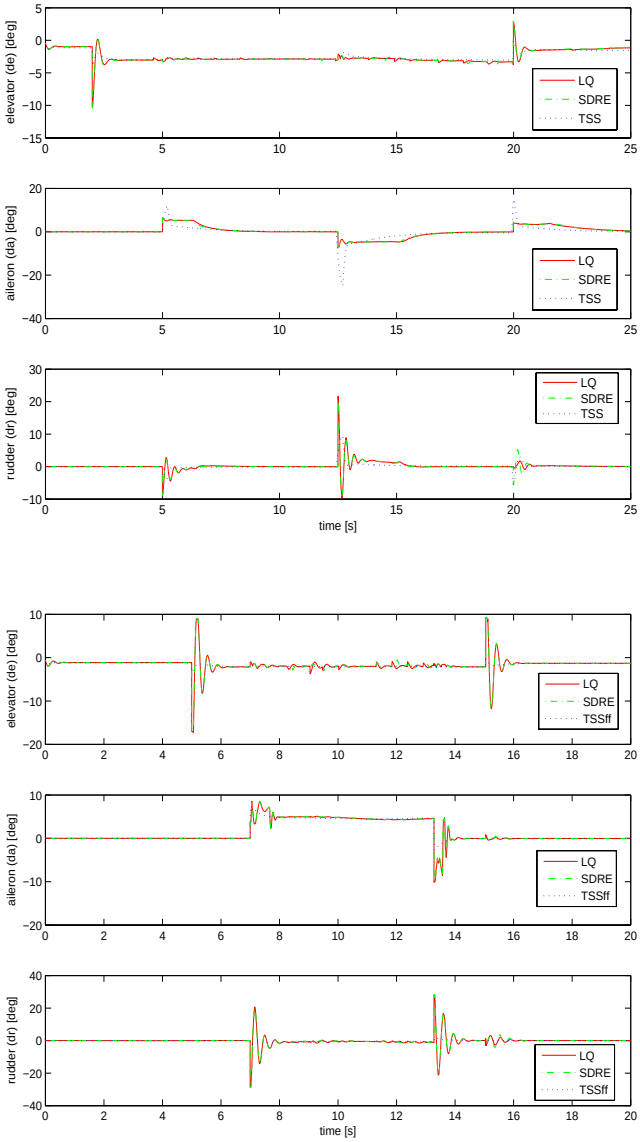

Figure 4.21: δ_a, δ_e, δ_r. Above: commanding ϕ for SDRE LQ and TSS. Below: commanding ξ for SDRE, LQ and TSS.

Flying condition	Maneuver	Maneuver index
M = 0.5 altitude = 5000m	Step in ξ, 1 rad/s	1
M = 0.8 altitude = 7000m	Step in α, 0.2 rad performin a partial looping	2

Table 4.2: Maneuvers performed in the Montecarlo simulation

j	Maneuver index	Weight in J	Design requirement
1	1	100	Overall stability
2	1	10	Overshoot max 10%
3	1	1	Settling time 1 second
4	1	1	Steady state error max 5%
5	2	100	Overall stability
6	2	10	Overshoot max 10%
7	2	1	Settling time 1 second
8	2	1	Steady state error max 5%

Table 4.3: Criterias for Monte Carlo simulation

P	Maneuver 1 SDRE	LQ
Overall stability	0	0
Overshoot	0.015	0
Settling time	0	0.025
Steady state error	0.105	0
	Maneuver 2	
Overall stability	0.02	0.0114
Overshoot	0	0
Settling time	0	0
Steady state error	0	0

Table 4.4: Probabilities P in (4.2) for the Monte Carlo simulation

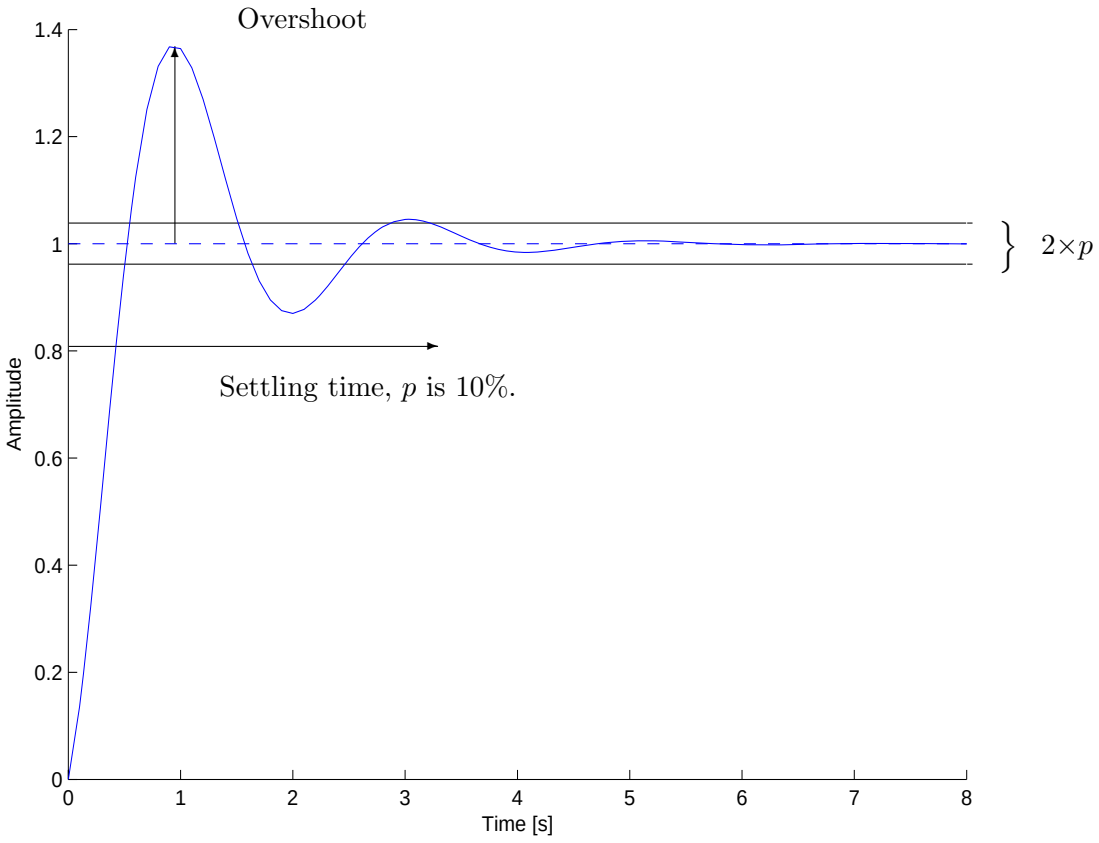

Figure 4.22: Definitions of Overshoot and Settling time.

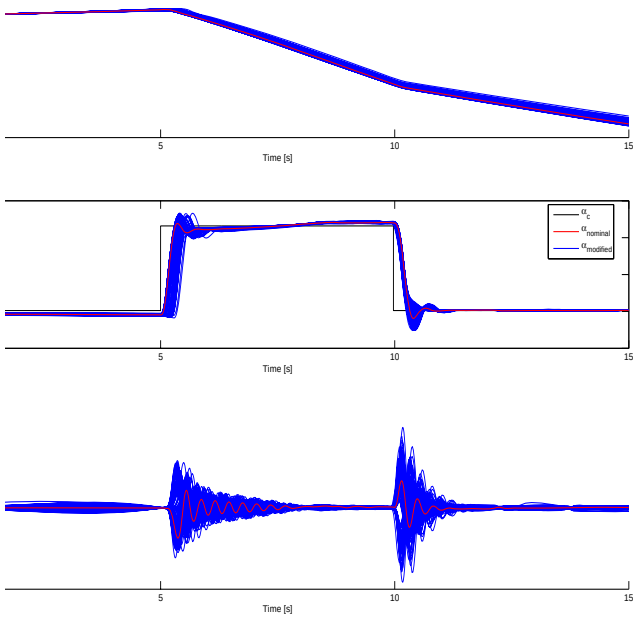

Figure 4.23: Monte Carlo simulation for a partial looping.

Chapter 5

Conclusions

The difference between SDRE and LQ is not as high as predicted. This very similar results may be explained by several facts. First, the nonlinear equations of motion are during flyable conditions (small α and β) relatively linear, i.e $\cos\alpha$ and $\sin\alpha$ are with good accuracy equal to 1 and α respectively and $\cos\beta$ and $\sin\beta$ is with even higher accuracy equal to 1 and β respectively. Further it can be aircraft specific since a study of the aerodynamic data shows a strong linear behaviour for moderate and relatively high angles of attacks and sideslip. This might be different for other aircrafts though. For lowspeed flight, i.e landing and starting conditions which require high angle of attacks, SDRE may show a clearer better performance but these flying conditions are not considered in this thesis. This phenomena is emphasized when considering that the vectorfield works with variables that are deviations from trimmed conditions. These deviations are often small and when they $\rightarrow 0$ SDRE and LQ asymptotically becomes exactly the same method. The use of high frequency CPU:s calculating the correct feedback online at each iteration simplifies the design considerably and this is regarded as the most important conclusion of this work. Of course, when using hardware capable of realtime calculations, it is possible to let LQ(as well as it is possible for SDRE) calculate the feedbacks online instead of using scheduling between precalculated gains. This makes the difference between SDRE and LQ even narrower. The use of state dependent gain matrices may also be implemented $\mathbf{Q}(\mathbf{x})$ and $\mathbf{R}(\mathbf{x})$ using realtime calculations of LQ. Now, the only difference between SDRE and LQ is the matrix $\mathbf{A}$, which for SDRE gives higher accuracy between $\mathbf{f}(\mathbf{x})$ and $\mathbf{A}$. Since the calculation of the this matrix is quite simple and only needs to be scripted once, SDRE is preferable over LQ. One disadvantage with SDRE and LQ is that it stabilizes the vectorfield around trimmed conditions which must be precalculated(or scripted to be calculated online requiring extra CPU power). Further the accuracy in these trimmed condition (3.7) is vital for stability and for the steady state error. A non-zero result, $\mathbf{f}(\mathbf{x}_0)+\mathbf{B}(\mathbf{x}_0)\mathbf{u}_0 \approx 0$, will give a steady

state error. This may however be counteracted by the use of integrators. Using TSS has its benefits since the result is quite predictable following a first order system behaviour. Also no precalculations of trimmed states are needed Despite these advantages its slower, poorer performance is decisive to the recommendation of SDRE. Regarding robustness the results shows clearly that the model is robust even for as high model modifications as 40%.

Appendix A

The matrix A(x)

$A(x)$ is built up by submatrices as stated below.

$$
\begin{aligned}
A_{11} = & \frac{qSC_{Z_0}}{Vm}\Big[-\frac{\tilde{\alpha}+2\alpha_0}{2!}+\frac{\tilde{\alpha}^3+4\tilde{\alpha}^2\alpha_0+6\tilde{\alpha}\alpha_0^2+4\alpha_0^3}{4!}\Big] \\
& + \frac{qSC_{Z_\alpha}}{Vm}\Big[1-\frac{\tilde{\alpha}^2+3\tilde{\alpha}\alpha_0+3\alpha_0^2}{2!}+\frac{\tilde{\alpha}^4+5\tilde{\alpha}^3\alpha_0+10\tilde{\alpha}^2\alpha_0^2+10\tilde{\alpha}\alpha_0^3+5\alpha_0^4}{4!}\Big] \\
& + \frac{qSC_{X_0}}{Vm}\Big[-1+\frac{\tilde{\alpha}^2+3\tilde{\alpha}\alpha_0+3\alpha_0^2}{3!}\Big] \\
& + \frac{qSC_{X_\alpha}}{Vm}\Big[-(\tilde{\alpha}+2\alpha_0)+\frac{\tilde{\alpha}^3+4\tilde{\alpha}^2\alpha_0+6\tilde{\alpha}\alpha_0^2+4\alpha_0^3}{3!}\Big] \\
& + \frac{qSC_{X_{\alpha^2}}}{Vm}\Big[-(\tilde{\alpha}^2+3\tilde{\alpha}\alpha_0+3\alpha_0^2\Big)] \\
& + \frac{g}{V}[\cos(\theta)\cos(\phi)(-\frac{\tilde{\alpha}+2\alpha_0}{2!})-\sin(\theta)]
\end{aligned}
\tag{A.1}
$$

$$
\begin{aligned}
A_{12} = & -p_0\cos(\tilde{\alpha}+\alpha_0)[1-\frac{\tilde{\beta}^2}{3!}]-r_0\sin(\tilde{\alpha}+\alpha_0)[1-\frac{\tilde{\beta}^2}{3!}] \\
& + \cos(\tilde{\alpha}+\alpha_0)\frac{qS}{Vm}C_{Z_{\alpha|\beta|}}(\tilde{\alpha}+\alpha_0)sgn(\tilde{\beta}) \\
& + \cos(\tilde{\alpha}+\alpha_0)\frac{qS}{Vm}C_{Z_{\beta^3}}\tilde{\beta}^2
\end{aligned}
\tag{A.2}
$$

$$
A_{21} = -p_0[1]+r_0[\frac{\tilde{\alpha}+2\alpha_0}{2!}]
$$

(A.3)

$$A_{22} = qSC_{Y_\beta} + \cos(\tilde{\alpha}+\alpha_0)\Big[1 - \frac{\tilde{\beta}^2}{3!} + \frac{\tilde{\beta}^4}{5!}\Big]\frac{X(\tilde{\mathbf{x}}+\mathbf{x}_0)}{Vm} - \sin(\tilde{\alpha}+\alpha_0)\Big[1 - \frac{\tilde{\beta}^2}{3!} + \frac{\tilde{\beta}^4}{5!}\Big]\frac{Z(\tilde{\mathbf{x}}+\mathbf{x}_0)}{Vm} \tag{A.4}$$

$$\begin{pmatrix} A_{13} & A_{14} \\ A_{23} & A_{24} \end{pmatrix} = \begin{pmatrix} -\cos(\tilde{\alpha}+\alpha_0))\sin(\tilde{\beta}) & 1+\cos(\tilde{\alpha}+\alpha_0)\frac{qS}{Vm}C_{Z_q} \\ \sin(\tilde{\alpha}+\alpha_0) & 0 \end{pmatrix} \tag{A.5}$$

$$\begin{pmatrix} A_{15} \\ A_{25} \end{pmatrix} = \begin{pmatrix} -\sin(\tilde{\alpha}+\alpha_0)\sin(\tilde{\beta}) \\ -\cos(\tilde{\alpha}+\alpha_0) \end{pmatrix} \tag{A.6}$$

$$\begin{pmatrix} A_{31} \\ A_{41} \\ A_{51} \end{pmatrix} = qS \begin{pmatrix} 0 \\ c\frac{(C_{m_\alpha}+C_{m_{\alpha^2}}(\tilde{\alpha}+2\alpha_0)+C_{m_{\alpha^3}}(\tilde{\alpha}^2+3\tilde{\alpha}\alpha_0+3\alpha_0^2))}{I_{yy}} \\ 0 \end{pmatrix} \tag{A.7}$$

$$\begin{pmatrix} A_{32} \\ A_{42} \\ A_{52} \end{pmatrix} = qS \begin{pmatrix} b\frac{(C_{l_\beta}+C_{l_{\alpha\beta}}(\tilde{\alpha}+\alpha_0))}{I_{xx}} \\ c\frac{C_{m_{\alpha|\beta|}}(\tilde{\alpha}+\alpha_0)sgn(\tilde{\beta})}{I_{yy}} \\ b\frac{(C_{n_\beta}+C_{n_{\alpha\beta}}(\tilde{\alpha}+\alpha_0))}{I_{zz}} \end{pmatrix} \tag{A.8}$$

$$\begin{pmatrix} A_{33} & A_{34} \\ A_{43} & A_{44} \\ A_{53} & A_{54} \end{pmatrix} = \begin{pmatrix} L_{\tilde{p}} & (\tilde{r}+r_0)\frac{(I_{yy}-I_{zz})}{I_{xx}} \\ r_0\frac{(I_{zz}-I_{xx})}{I_{yy}} & M_{\tilde{q}} \\ (\tilde{q}+q_0)\frac{(I_{xx}-I_{yy})}{I_{zz}} & p_0\frac{(I_{xx}-I_{yy})}{I_{zz}} \end{pmatrix} \tag{A.9}$$

$$\begin{pmatrix} A_{35} \\ A_{45} \\ A_{55} \end{pmatrix} = \begin{pmatrix} q_0\frac{(I_{yy}-I_{zz})}{I_{xx}} \\ (\tilde{p}+p_0)\frac{(I_{zz}-I_{xx})}{I_{yy}} \\ N_{\tilde{r}} \end{pmatrix} \tag{A.10}$$

$$
\begin{aligned}
A_{61} = {} & p_0 q_0 [1 - \frac{\tilde{\alpha}^2 + 3\tilde{\alpha}\alpha_0 + 3\alpha_0^2}{3!}](\frac{I_{xx} - I_{yy}}{I_{zz}}) \\
& + r_0 q_0 [-\frac{\tilde{\alpha} + 2\alpha_0}{2!}](\frac{I_{yy} - I_{zz}}{I_{xx}}) \\
& + p_0 \frac{qS}{Vm}[C_{X_\alpha} + C_{X_\alpha^2}(\tilde{\alpha} + 2\alpha_0) + C_{X_\alpha^3}(\tilde{\alpha}^2 + 3\tilde{\alpha}\alpha_0 + 3\alpha_0^2)] \\
& + r_0 \frac{qS}{Vm}[C_{Z_\alpha} + C_{Z_\alpha^2}(\tilde{\alpha} + 2\alpha_0)] \\
& + \frac{qS}{2I_{xx}} C_{l_p} p_0 (-(\tilde{\alpha} + 2\alpha_0)) + \frac{qS}{I_{zz}} C_{n_r} r_0 (1 - \frac{(\tilde{\alpha}^2 + 3\tilde{\alpha}\alpha_0 + 3\alpha_0^2}{3!}))
\end{aligned} \tag{A.11}
$$

$$
\begin{aligned}
A_{62} = {} & qS(\frac{q_0 CY_\beta}{Vm} + \frac{\cos(\tilde{\alpha} + \alpha_0) Cl_\beta}{VI_{xx}} + \frac{\sin(\tilde{\alpha} + \alpha_0) Cn_\beta}{VI_{zz}}) \\
& + [1 - \frac{\tilde{\beta}^2}{3!}][(\tilde{r} + r_0)(\tilde{p} + p_0)(I_{zz} - I_{xx})/I_{yy} + \frac{M(\tilde{\mathbf{x}} + \mathbf{x}_0)}{I_{yy}}]
\end{aligned} \tag{A.12}
$$

$$
A_{63} = \frac{X(\tilde{\mathbf{x}} + \mathbf{x}_0)}{Vm} + \frac{g}{V}\cos(\theta)\sin(\phi) + (\tilde{q} + q_0)\sin(\tilde{\alpha} + \alpha_0)\frac{I_{xx} - I_{yy}}{I_{zz}} \tag{A.13}
$$

$$
\begin{aligned}
A_{64} = {} & \frac{Y(\tilde{\mathbf{x}} + \mathbf{x}_0)}{Vm} - \frac{g}{V}\sin(\theta) \\
& + (\tilde{r} + r_0)\cos(\tilde{\alpha} + \alpha_0)\frac{I_{yy} - I_{zz}}{I_{xx}} + q_0 \sin(\tilde{\alpha} + \alpha_0)\frac{I_{xx} - I_{yy}}{I_{zz}}
\end{aligned} \tag{A.14}
$$

$$
A_{65} = \frac{Z(\tilde{\mathbf{x}} + \mathbf{x}_0)}{Vm} + \frac{g}{V}\cos(\theta)\cos(\phi) + q_0 \cos(\tilde{\alpha} + \alpha_0)\frac{I_{yy} - I_{zz}}{I_{xx}} \tag{A.15}
$$

$$\begin{pmatrix} A_{61} \\ A_{62} \\ A_{63} \\ A_{64} \\ A_{65} \\ A_{66} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \tag{A.16}$$

Appendix B

Nonlinear Dynamic Inversion via Two-Timescale Separation

Here we give a summary of the TSS method and the extension of it to the feedforward case, the TSSff method which is used in some simulations. The TSSff method is described further in [18, 19].

B.1 Underlying Idea

The system considered for the basic TSS method is the following

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{y}, \tag{B.1}$$

$$\dot{\mathbf{y}} = \mathbf{h}(\mathbf{y}) + \mathbf{k}\mathbf{u}, \tag{B.2}$$

where $\mathbf{f}, \mathbf{h}$ are continuous vector fields in $\mathbb{R}^n$ and $\mathbf{g}$ is a continuous function on $\mathbb{R}^n$ with values in $\mathbb{R}^{n\times n}$, and $\mathbf{k}$ is an invertible constant matrix in $\mathbb{R}^{n\times n}$. It is moreover assumed that $\mathbf{f}(0) = 0$ and $\mathbf{h}(0)$, and that $\mathbf{g}(\mathbf{x})$ is invertible for all $\mathbf{x}$. The object is to stabilize the equilibrium at 0 simultaneously for the $\mathbf{x}$ and $\mathbf{y}$-systems in (B.2).

The first step in deriving the TSS control law is to decide on some desired dynamics

$$\dot{\mathbf{z}} = \mathbf{f}_d(\mathbf{z}) \tag{B.3}$$

for the $\mathbf{x}$-system in (B.1). Next, the dynamics for the $\mathbf{x}$-system is rewritten as

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{y} \\ &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{y}_d(\mathbf{x}) + \mathbf{g}(\mathbf{x})(\mathbf{y} - \mathbf{y}_d(\mathbf{x})) \\ &= \mathbf{f}_d(\mathbf{x}) + \mathbf{g}(\mathbf{x})\boldsymbol{\epsilon}(\mathbf{x}),\end{aligned} \tag{B.4}$$

where $\mathbf{y}_d(\mathbf{x})$ is a "virtual control law" such that

$$\mathbf{f}_d(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})\mathbf{y}_d(\mathbf{x}) \tag{B.5}$$

and the error term $\boldsymbol{\epsilon}(\mathbf{x}, \mathbf{y})$ is defined as

$$\boldsymbol{\epsilon}(\mathbf{x}, \mathbf{y}) = \mathbf{y} - \mathbf{y}_d(\mathbf{x}). \tag{B.6}$$

The idea is to try to make the term $\mathbf{g}(\mathbf{x})\boldsymbol{\epsilon}(\mathbf{x}, \mathbf{y})$ in (B.4) small so that the desired dynamics (B.3) for $\mathbf{x}$ is approximately realized. In the applications of TSS the desired dynamics $\mathbf{f}_d$ for the $\mathbf{x}$-system are often of the linear decoupled (diagonal) type. The virtual control law can be solved for in (B.5), which yields

$$\mathbf{y}_d(\mathbf{x}) = \mathbf{g}(\mathbf{x})^{-1}(\mathbf{f}_d(\mathbf{x}) - \mathbf{f}(\mathbf{x})). \tag{B.7}$$

In the second step of TSS one proceeds analogously as in the first step and derives the control law $\mathbf{u}(\mathbf{x}, \mathbf{y})$ from the relations

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{h}(\mathbf{y}) + \mathbf{k}\mathbf{u}(\mathbf{x}, \mathbf{y}) \\ &= \mathbf{f}_d^{(\mathbf{y})}(\mathbf{y} - \mathbf{y}_d(\mathbf{x})), \end{aligned} \tag{B.8}$$

where $\mathbf{f}_d^{(\mathbf{y})}$ represents desired dynamics for the $\mathbf{y}$-system in (B.2). The control law $\mathbf{u}(\mathbf{x}, \mathbf{y})$ can now be written down using (B.7) and (B.8) explicitly as

$$\begin{aligned} \mathbf{u}(\mathbf{x}, \mathbf{y}) &= \mathbf{k}^{-1}(\mathbf{f}_d^{(\mathbf{y})}(\mathbf{y} - \mathbf{y}_d(\mathbf{x})) - \mathbf{h}(\mathbf{y})) \\ &= \mathbf{k}^{-1}\big(\mathbf{f}_d^{(\mathbf{y})}(\mathbf{y} - \mathbf{g}(\mathbf{x})^{-1}(\mathbf{f}_d(\mathbf{x}) - \mathbf{f}(\mathbf{x}))) - \mathbf{h}(\mathbf{y})\big) \end{aligned} \tag{B.9}$$

Often the desired dynamics for both $\mathbf{f}_d$ and $\mathbf{f}_d^{(\mathbf{y})}$ are chosen as linear dynamics so that

$$\begin{aligned} \mathbf{f}_d(\mathbf{x}) &= -\mathbf{A}_\mathbf{x}\mathbf{x}, \\ \mathbf{f}_d^{(\mathbf{y})}(\mathbf{y} - \mathbf{y}_d(\mathbf{x})) &= -\mathbf{A}_\mathbf{y}(\mathbf{y} - \mathbf{y}_d(\mathbf{x})), \end{aligned} \tag{B.10}$$

where $\mathbf{A}_\mathbf{x}$ and $\mathbf{A}_\mathbf{y}$ are two positive definite matrices, in particular diagonal matrices for decoupled dynamics.

B.2 Stability of Two-Timescale Separation

A stability proof for TSS has been given in [20]. It is based on Lyapunov techniques and requires, among other things, that the time scale separation between the loops (the $\mathbf{x}$-system in (B.1) and the $\mathbf{y}$-system in (B.2)) is sufficient. It moreover assumes infinitely fast actuator responses, i.e. a requirement that the dynamics of the "fast" loop (the $\mathbf{y}$-system) can be set arbitrarily fast. Both of these assumptions limit the applicability of the technique in practical cases.

B.3 Extension to the Feed-Forward Case

The basic TSS control law (B.9) can be extended to the case of a feed-forward system as in

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x},\mathbf{y})\mathbf{y} + \mathbf{n}(\mathbf{x},\mathbf{y})\mathbf{u}, && \text{(B.11)} \\ \dot{\mathbf{y}} &= \mathbf{h}(\mathbf{y}) + \mathbf{k}\mathbf{u}, && \text{(B.12)} \end{aligned}$$

where $\mathbf{f}, \mathbf{h}$ and $\mathbf{k}$ are as in (B.1), (B.2), and $\mathbf{g}, \mathbf{n}$ are continuous functions on $\mathbb{R}^{n\times n}$ with values in $\mathbb{R}^{n\times n}$. It is not hard to see that for the case of wind vector roll rate control the model in (2.16) is of this form.

We assume that the desired dynamics for $\mathbf{f}_d^{(\mathbf{y})}$ are of the linear form in (B.10). The object is to find a virtual control law $\mathbf{y}_d(\mathbf{x},\mathbf{y})$ and a control law $\mathbf{u}(\mathbf{x},\mathbf{y})$ such that some suitable error dynamics are obtained around the equilibrium at 0 for the $\mathbf{x}$ and $\mathbf{y}$ systems in (B.11), (B.11). In order to achieve this, we seek a virtual control law $\mathbf{y}_d(\mathbf{x},\mathbf{y})$ and a control law $\mathbf{u}(\mathbf{x},\mathbf{y})$ such that

$$\begin{aligned} \mathbf{f}_d(\mathbf{x}) &= \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x},\mathbf{y})\mathbf{y}_d(\mathbf{x},\mathbf{y}) + \mathbf{n}(\mathbf{x},\mathbf{y})\mathbf{u}(\mathbf{x},\mathbf{y}), && \text{(B.13)} \\ -\mathbf{A}_\mathbf{y}(\mathbf{y} - \mathbf{y}_d(\mathbf{x},\mathbf{y})) &= \mathbf{h}(\mathbf{y}) + \mathbf{k}\mathbf{u}(\mathbf{x},\mathbf{y}). && \text{(B.14)} \end{aligned}$$

From (B.14) we can solve for the virtual control law $\mathbf{y}_d(\mathbf{x},\mathbf{y})$ as

$$\mathbf{y}_d(\mathbf{x},\mathbf{y}) = \mathbf{y} + \mathbf{A}_\mathbf{y}^{-1}(\mathbf{h}(\mathbf{y}) + \mathbf{k}\mathbf{u}(\mathbf{x},\mathbf{y})), \tag{B.15}$$

and if we insert this expression for $\mathbf{y}_d(\mathbf{x},\mathbf{y})$ into (B.13) we obtain

$$\mathbf{f}_d(\mathbf{x}) = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x},\mathbf{y})(\mathbf{y} + \mathbf{A}_\mathbf{y}^{-1}\mathbf{h}(\mathbf{y})) + (\mathbf{n}(\mathbf{x},\mathbf{y}) + \mathbf{g}(\mathbf{x},\mathbf{y})\mathbf{A}_\mathbf{y}^{-1}\mathbf{k})\mathbf{u}(\mathbf{x},\mathbf{y}). \tag{B.16}$$

Thus, if the matrix

$$\mathbf{n}(\mathbf{x},\mathbf{y}) + \mathbf{g}(\mathbf{x},\mathbf{y})\mathbf{A}_\mathbf{y}^{-1}\mathbf{k} \tag{B.17}$$

is invertible for all $\mathbf{x}, \mathbf{y}$ we can always solve for $\mathbf{u}(\mathbf{x},\mathbf{y})$ here and define the control law $\mathbf{u}(\mathbf{x},\mathbf{y})$ by

$$\mathbf{u}(\mathbf{x},\mathbf{y}) = (\mathbf{n}(\mathbf{x},\mathbf{y}) + \mathbf{g}(\mathbf{x},\mathbf{y})\mathbf{A}_\mathbf{y}^{-1}\mathbf{k})^{-1}(\mathbf{f}_d(\mathbf{x}) - \mathbf{f}(\mathbf{x}) - \mathbf{g}(\mathbf{x},\mathbf{y})(\mathbf{y} + \mathbf{A}_\mathbf{y}^{-1}\mathbf{h}(\mathbf{y})). \tag{B.18}$$

B.3.1 On the rank condition in feed-forward TSS

In [18] it is shown that the rank condition (B.17) is fulfilled for diagonal $\mathbf{A}_\mathbf{y}$ when $\dot{V} = 0$ for the system in (2.16) with wind vector roll rate control.

Appendix C

Rootlocus plots

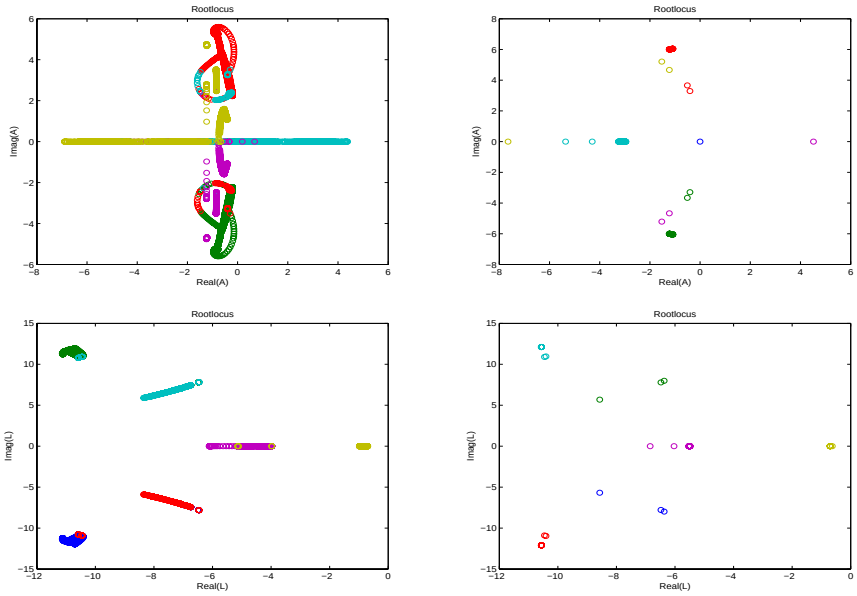

Figure C.1: Rootlocus plots for $\mathbf{A}(\mathbf{x})$, $\mathbf{A}$, $\mathbf{A}(\mathbf{x}) - \mathbf{BK}$, $\mathbf{A} - \mathbf{BK}$ a looping at M = 0.8 and altitude 7000m

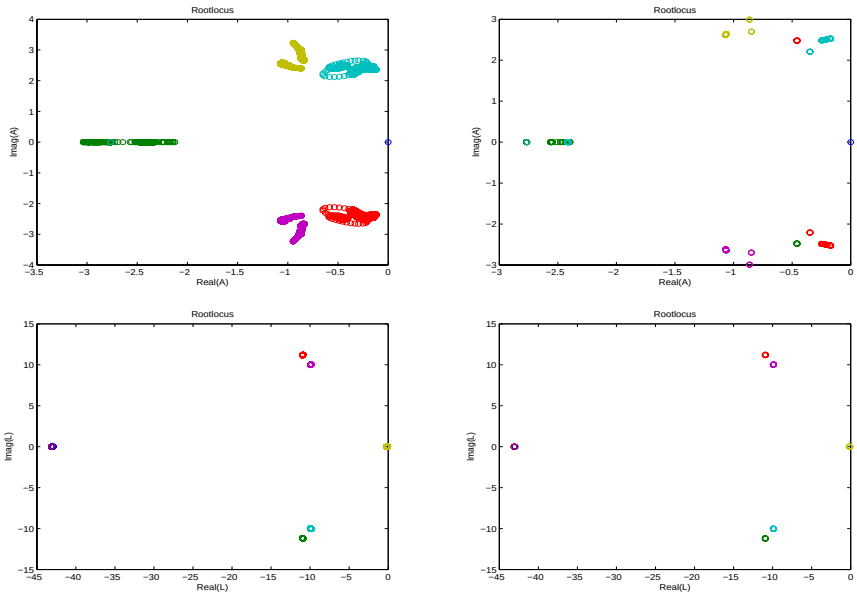

Figure C.2: Rootlocus plots for $\mathbf{A}(\mathbf{x})$, $\mathbf{A}$, $\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x})\mathbf{K}$, $\mathbf{A} - \mathbf{BK}$ for high angle of attack windvector roll, commanding $\xi \rightarrow 0.1rad/s \rightarrow -0.1rad/s \rightarrow 0rad/s$. .

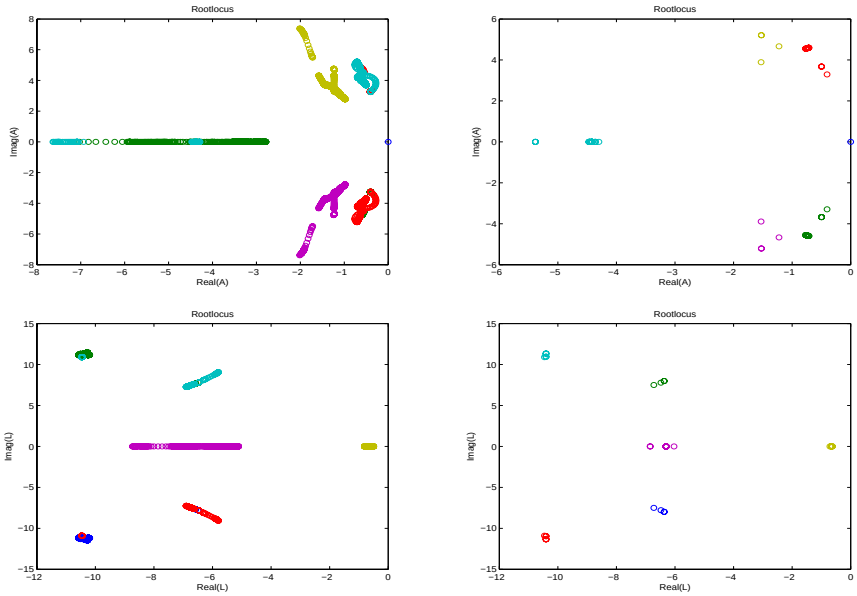

Figure C.3: Rootlocus plots for $\mathbf{A}(\mathbf{x})$, $\mathbf{A}$, $\mathbf{A}(\mathbf{x}) - \mathbf{B}(\mathbf{x})\mathbf{K}$, $\mathbf{A} - \mathbf{BK}$ for high angle of attack windvector roll, commanding $\Phi \rightarrow 180° \rightarrow -180° \rightarrow 0°$.

Bibliography

[1] B.D.O. Anderson and J.B. Moore, *Optimal Control: Linear Quadratic Methods,* Prentice Hall, 1989.

[2] T. Ivarsson, "Gripen – A Fourth Generation Combat Aircraft," *Proc. AIAA/ICAS Intl. Air and Space Symposium and Exposition,* Dayton, OH, July 14–16 2003. AIAA paper 2003-2862.

[3] B.L. Stevens and F.L. Lewis, *Aircraft Control and Simulation,* Wiley, Hoboken, NJ, 2003.

[4] J.R Cloutier, C. N. D'Souza and C. P. Mracek, "Nonlinear regulation and nonlinear H_∞ control via the state-dependent Riccati equation technique; part 1, theory; part 2, examples," *Proc. Intl. Conference on Nonlinear Problems in Aviation and Aerospace,* Daytona Beach, FL, May 1996, pp. 117–142.

[5] A. Bracci, M. Innocenti and L. Pollini, "A Newton Algorithm for Implementation of SDRE Controllers," *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit*, San Francisco, CA, Aug. 15–18 2005. AIAA paper 2005-6386.

[6] A. Bogdanov, M. Carlsson, G. Harvey, J. Hunt D. Kieburtz and R. van der Merwe, "State-Dependent Riccati Equation Control of a Small Unmanned Helicopter," *Proc. AIAA Guidance, Navigation, and Control Conference and Exhibit,* Austin, TX, Aug. 11–14 2003. AIAA paper 2003-5672.

[7] J. R. Cloutier "State-Dependent Riccati Equation Techniques: An Overview," *Proc. 1997 American Control Conference,* Albuquerque, NM, June 4–6 1997.

[8] H. T. Banks, B. M. Lewis, and H. T. Tran, *Nonlinear Feedback Controllers and Compensators: A State-Dependent Riccati Equation Approach,* Final Technical Report, CRSC-TR03-26, Center for Research in Scientific Computation, North Carolina State University, Raleigh, NC, July, 2001.

[9] P.K. Menon, T. Lam, L.S. Crawford and V. H. L. Cheng, "Real-Time Computational Methods for SDRE Nonlinear Control of Missiles," *Proc. American Control Conference,* Anchorage, AK, May 8–10 2002.

[10] D. Ito, J. Georgie, J. Vlasek and D.T. Ward, *Re-Entry Vehicle Flight Controls Design Guidlines,* Final Technical Report, NAG9-1085, Flight Simulation Laboratory, Texas Engineering Experiment Station, Texas A&M University, 2001.

[11] S.H. Lane and R.F. Stengel, "Flight Control Design using Nonlinear Inverse Dynamics," *Proc. 1986 American Control Conference,* Seattle, WA, June 1986, pp. 387–396.

[12] D.J. Bugajski, D.F. Enns and M.R. Elgersma, "A Dynamic Inversion Based Control Law with Application to the High Angle-of-Attack Research Vehicle," *Proc. AIAA Guidance, Navigation, and Control Conference,* Portland, OR, Aug. 1990, pp. 826–839. AIAA paper 90-3407-CP.

[13] G.J. Balas, W.L. Garrard and J. Reiner, "Robust Dynamic Inversion Control Laws for Aircraft Control," *Proc. AIAA Guidance, Navigation and Control Conference,* Hilton Head Island, SC, Aug. 10–12, 1992, pp. 192–205. AIAA paper 92-4329-CP.

[14] J-F. Magni, S. Bennani and J. Terlouw, Eds, *Robust Flight Control: A Design Challenge (GARTEUR),* Volume 224 of the Lecture Notes in Control and Information Sciences, Springer-Verlag, 1997.

[15] Q. Wang and R. F. Stengel, "Robust Nonlinear Flight Control of a High-Performance Aircraft," *IEEE Trans. Control Systems Technology,* Vol. 13, No. 1, Jan. 2005, pp. 15-26.

[16] ADMIRE aircraft model `http://www.foi.se/admire` (2005-12-20).

[17] The Mathworks Inc. `http://www.mathworks.com` (2005-12-20).

[18] J.W.C. Robinson, S.-L. Wirkander, M. Hammar and L. Forssell, *Nonlinear Flight Control Design for a UAV,* Swedish Defence Research Agency, FOI report, FOI-R–1863–SE, Oct. 2005.

[19] J.W.C. Robinson and M. Högberg, *Block Backstepping and Two-Timescale Control for Aircraft and Missiles,* Swedish Defence Research Agency, FOI report, Manuscript in preparation.

[20] C. Schumacher and P.P. Khargonekar, "Stability Analysis of a Missile Control System with a Dynamic Inversion Controller," *J. Guidance, Control and Dynamics*, Vol. 21, No. 3, 1998, pp. 508–515.

[21] M. Athans and P. Falb, *Optimal Control,* McGraw-Hill, New York, NY, 1966.

[22] F.L. Lewis, *Optimal Control,* Wiley, New York, NY, 1986.

[23] U. Jönsson, C. Trygger and P. Ögren, *Lecture Notes on Optimal Control,* Dept. Optimization & Systems Theory Report, Royal Inst. Technology 2002.

[24] M.H. Kalos and P.A. Whitlock, *Monte Carlo Methods, Volume I: Basics,* Wiley-Interscience Publications, John Wiley and Sons, New York, 1986.

[25] J. M. Hammersley and D. C. Handscomb. *Monte Carlo Methods,* Chapman and Hall, London & New York, 1964.