Robustness Analysis of Feedback Linearisation for Uncertain Rational Systems

Thesis submitted for the degree of

Doctor of Philosophy

at the University of Leicester

by

Peter David Norton

Department of Engineering
University of Leicester

December 2015

Abstract

Robustness Analysis of Feedback Linearisation for Uncertain Rational Systems

Peter Norton

Feedback Linearisation (FL) is a nonlinear control technique that has gained a lot of attention in the past 30 years. Due to its relatively simple synthesis, the use of FL has been investigated particularly in the aerospace community, because aircraft models are often highly nonlinear and a controller is needed that can guarantee good performance over a wide range of operating conditions. However, mathematical models of real-life physical systems always have a level of uncertainty on them, as they are only ever approximations to the real system. In the current literature, the robustness of FL control has been analysed by extensive simulations, which may miss some worst-case combinations of uncertainties in the model. Alternatively, the robustness of the controlled system has been analysed on simplified linear models, using techniques from classical control, which do not well represent the inherently nonlinear dynamics of the system. This thesis contributes to the literature by using more recent techniques for analysis of nonlinear systems to assess robust stability under FL control. We apply advanced robust and nonlinear analysis techniques without the assumption that the controller has direct access to all the states of the system, by including state estimation or sensors in the closed loop for analysis. We also develop an existing analysis technique in the literature to show that a system under approximate FL control does not violate position limits of the actuator, despite uncertainty in the model, improving the rigour of the analysis. This is applied to a high-fidelity model of an aircraft, designed for use in industry and academia.

Acknowledgements

Thanks to my supervisor Dr. Emmanuel Prempain for giving me this opportunity and for suggesting this direction of research. I am grateful to him and to Prof. Matt Turner for guidance at points during my time in the department.

Thanks to the EPSRC for the DTA studentship which gave me financial support for the first three years of this degree.

Thanks and appreciation to friends currently and formerly of the Control Research Group: Bharani, Emre, Xiaoxing, Hasan, Indira, Mirza, The Captain, and Tomas.

Thanks in particular to Dr. Paresh Deshpande for his straight talking and whisky, which were beneficial in times of stress.

My great love and appreciation to the gallery of anachronisms Fatstone, Twoddle, Five Bellies, Three Plates, Bonehead, Dr. Beer and of course Simon Corn. You were exactly what you needed to be. Der Weg ist das Ziel.

Above all, thanks to my family for their love, support and patience during busy years.

Contents

List of Figures

List of Tables

Acronyms

AoA	Angle of Attack
AoS	Angle of Sideslip
ADMIRE	Aero-Data Model In a Research Environment
DA	Domain of Attraction
DI	Dynamic Inversion
FL	Feedback Linearisation
GA	Genetic Algorithm
GAM	Generic Aerodata Model
GM	Gain Margin
IOL	Input-Output Linearisation
IQC	Integral Quadratic Constraint
ISS	Input-to-State Stability
LDI	Linear Differential Inclusion
LFT	Linear Fractional Transformation
LMI	Linear Matrix Inequality
LPV	Linear Parameter-Varying
LTI	Linear Time-Invariant
MIMO	Multi-Input Multi-Output
NDI	Nonlinear Dynamic Inversion
PM	Phase Margin
QLPV	Quasi-Linear Parameter Varying
RDA	Robust Domain of Attraction
SISO	Single-Input Single-Output
SL	Straight and Level
TI	Time-Invariant
TSS	Time Scale Separation
TV	Time-Varying

Mathematical Notation

$\mathbb{R}$	real scalars
$\mathbb{R}^n$	$n \times 1$ vectors with real entries
$\mathbb{R}^{n \times m}$	$n \times m$ matrices with real entries
$\|x\|$	euclidean norm

Matrices

$\|X\|$	induced L_2 norm/ max. singular value
X^T	transpose
X^{-1}	inverse
$X^\dagger$	right-inverse $X^T(XX^T)^{-1}$
$X > 0$	symmetric, positive-definite
I_n	$n \times n$ identity matrix
$0_{n \times m}$	$n \times m$ matrix of zeroes

Chapter 1

Introduction

1.1 Overview

Much of technology today relies on control systems for its operation from aerospace applications such as aircraft, satellites and missiles, to control of nuclear reactors. There has been a huge increase in the use of automatic control systems in technology in general, much of it based on linear approximations to plant models, using methods such as gain-scheduling to cover a wider range of operating conditions.

While control theory began with the development of controllers for linear systems, it has increasingly been accepted that real-world systems are inherently nonlinear, and more complex control techniques have therefore been developed. Nonlinear controllers take account of more complicated system dynamics and can potentially provide good performance over a wider range of operating conditions than linear controllers, without the need for relatively computationally demanding gain scheduling techniques. However, rigorous robust stability guarantees for nonlinear systems, particularly with uncertainty on the model, have not yet been developed fully.

This thesis concentrates on nonlinear controllers designed by feedback linearisation, which is designed to cancel out the nonlinear behaviour of the plant and replace it with the desired, generally linear, behaviour. This control method has been developed in the literature over the past 30 years and has become a fairly mature technique, however the question of robust stability of the closed loop subject

to imperfect knowledge of the plant dynamics and measurements remains an open problem.

We focus here on using robust and nonlinear systems analysis techniques to provide stability guarantees for restricted classes of nonlinear systems, namely those in which the plant states and uncertain parameters appear rationally. We focus on aerospace applications, as this control method is widely used in modern aerospace control research, although in principle the techniques used in this thesis can be applied to ther types of systems.

1.2 Thesis Structure

- Chapter 2 presents background on the development of feedback linearisation and in particular its application in aerospace. This chapter sets out a review of the literature in FL and raises the question of robust stability of systems controlled by feedback linearisation (FL), in the presence of model uncertainty, output feedback or actuator dynamics, something which is yet to be addressed rigorously in the literature. This then motivates the research presented in this thesis.

- In chapter 3 a nonlinear missile model with time-varying uncertain parameters is controlled with a simple feedback linearisation and time-scale separation design, with synthesis based on the nominal model and full state feedback. The closed-loop system is then represented as a linear fractional transformation (LFT). A robust H_∞ filter is designed for the controlled plant, to estimate unknown states. Robust stability of the closed-loop system is then verified by using a scaled linear differential inclusion (LDI) technique.

- Chapter 4 presents the Aero-Data Model in a Research Environment (ADMIRE), which is a highly nonlinear aircraft model that has been studied in the past particularly for applications of nonlinear control techniques. This is the model that will be used in later chapters for robust stability analysis of FL.

- In chapter 5 a pitch rate controller is designed for a linear parameter-varying (LPV) model of the short-period longitudinal dynamics of ADMIRE, using input-output linearisation. A scaled linear differential inclusion (LDI) technique is applied to verify stability of the parameter-varying zero-dynamics. The parameters (Mach and altitude) are allowed to be time-varying. The proposed controller is simulated in the nonlinear model of ADMIRE and on the LPV model, with sensors, actuators and time-varying uncertainties on the polynomial surface fits. A robust performance LMI condition is used to verify an upper bound on the L_2 gain of the full-order closed loop system, subject to time-varying parametric uncertainty. This uses a linear fractional transformation (LFT) representation of the closed loop with block-diagonal structured uncertainty.

- Chapter 6 develops a nonlinear, 3-axes polynomial model of ADMIRE's short-period dynamics. This is done to put the model, in particular the aerodynamic force and moment coefficients, in a form that is conducive to robust stability analysis for rational systems.

- In chapter 7 we use the polynomial model of ADMIRE to design a FL controller, using the principle of time-scale-separation (TSS). The controller is tuned on ADMIRE using a genetic algorithm, which optimises the performance in a series of aggressive manoeuvres. The main contribution of this chapter is that we then find a robust domain of attraction for the closed loop, which provides a region of safe initial conditions within which the aircraft is stable, and in which we guarantee that the actuators do not saturate.

- Chapter 8 consists of concluding remarks on the work in this thesis, its limitations and possible directions for future research.

Chapter 2

Background on Feedback Linearisation and Analysis

Feedback linearisation (FL) is a method for either fully or partially transforming a nonlinear system into a linear one, by appropriate choice of a nonlinear control law. This is distinct from Jacobian linearisation (JL), which is a method for obtaining a first-order Taylor expansion, which is an *approximation* of the original nonlinear system, around some operating point (the term *linearisation* in control literature usually refers to the latter, although it is usually clear from the context which method is being referred to).

Feedback linearisation essentially appears in two forms: *input-to-state linearisation* or *full-state linearisation* refer to a situation where a nonlinear control law can be found such that the resulting closed-loop system is entirely linear in the states, in which case stability and performance of the closed loop can be analysed using standard techniques for linear systems; *input-output linearisation* (IOL) refers to finding a nonlinear control law that makes the input-output map of the closed loop linear, but leaves part of the system nonlinear in the states, in which case stability and performance of the closed loop depends on the behaviour of the remaining nonlinear 'internal' dynamics.

Rigorous development of FL began in the early 1980s, which followed from earlier work on linearisability of a nonlinear system, using an invertible nonlinear coordinate

change of the states (a *diffeomorphism*) [3]. Work in [3] developed conditions under which a nonlinear control system could be transformed into an equivalent linear one, by means of a suitable choice of diffeomorphism.

This lead to the derivation of conditions under which an appropriate choice of nonlinear control law could transform a SISO nonlinear system into a linear closed-loop system (input-to-state linearisation) [4], followed shortly by conditions for more general multi-input systems [5] [6]. At around the same time it was shown in [7] that nonlinear feedback could make the input-output map of a nonlinear (possibly open-loop unstable) plant linear. The earlier work on input-to-state linearisation was then quickly followed in [8] by necessary and sufficient conditions under which feedback could produce a linear input-output response (IOL), where the system is only partially linearised. *Extended* linearisation developed in [9] [10] [11] deals with a slightly weaker goal - that of designing a control law such that the JL of the closed-loop (nonlinear) system around some operating point (or a set of linearisations around a set of operating points) is invariant and equal to the composition of the JL of the plant with the JL of the controller.

2.1 Feedback Linearisation and Normal Form for SISO Systems

In order to introduce the FL technique, we restrict our attention to a single-input, single-output system, with state-feedback control,:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}) + \mathbf{g}(\mathbf{x})u(\mathbf{x}) \tag{2.1a}$$

$$y = h(\mathbf{x}) \tag{2.1b}$$

where $\mathbf{f}(\mathbf{x})$, $\mathbf{g}(\mathbf{x})$, $u(\mathbf{x})$ and $h(\mathbf{x})$ are smooth nonlinear functions of the states:

$$\mathbf{x} \in \mathbb{R}^n$$

$$u : \mathbb{R}^n \mapsto \mathbb{R}$$

$$h : \mathbb{R}^n \mapsto \mathbb{R}$$

In this section, we show how to design a linearising controller for the system (2.1). We also show how to transform the coordinates into "normal form", where the control input appears only in one place. Note, we assume here a SISO system, i.e. u and y in (2.1) are scalar. Putting a system in normal form separates the system into internal dynamics and the input-output external dynamics. All the theory on linearisation and normal form presented here is from [12].

2.1.1 Input-Output Linearisation

We seek a state-feedback controller of the form

$$u = \alpha(\mathbf{x}) + \beta(\mathbf{x})v \tag{2.2}$$

such that the input-output map of (2.1) is linear. v is our new control input, called the *equivalent control*. More generally, we cannot linearise the relation between y and v directly, but we can linearise the relation between v and the ρ -th time derivative of y, i.e.

$$y^{(\rho)} = v \tag{2.3}$$

where ρ is called the *relative degree* of the system. Hence, the input-output map is actually a chain of integrators.

To find the linearising controller, we take the output (2.1b) and differentiate:

$$\dot{y} = \frac{\partial h}{\partial \mathbf{x}}\dot{\mathbf{x}} = \frac{\partial h}{\partial \mathbf{x}}\mathbf{f} + \frac{\partial h}{\partial \mathbf{x}}\mathbf{g}u \tag{2.4}$$

If the coefficient of u in (2.4) is non-zero, we can use (2.2) to linearise the relation between $\dot{y}$ and v. Otherwise, we must continue to differentiate:

$$\ddot{y} = \frac{\partial \dot{y}}{\partial \mathbf{x}}\dot{\mathbf{x}} = \left(\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial h}{\partial \mathbf{x}}\mathbf{f}\right)\right)\mathbf{f} + \left(\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial h}{\partial \mathbf{x}}\mathbf{f}\right)\right)\mathbf{g}u$$

Again, if the second term is non-zero, we can perform linearisation, otherwise we must differentiate again:

$$\dddot{y} = \frac{\partial \ddot{y}}{\partial \mathbf{x}}\dot{\mathbf{x}} = \left(\frac{\partial}{\partial \mathbf{x}}\left(\left(\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial h}{\partial \mathbf{x}}\mathbf{f}\right)\right)\mathbf{f}\right)\right)\mathbf{f} + \left(\frac{\partial}{\partial \mathbf{x}}\left(\left(\frac{\partial}{\partial \mathbf{x}}\left(\frac{\partial h}{\partial \mathbf{x}}\mathbf{f}\right)\right)\mathbf{f}\right)\right)\mathbf{g}u$$

Obviously, this notation quickly becomes cumbersome, so we use the Lie derivative notation [12]:

$$y^{(\rho)} = L_f^\rho h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x}) u \tag{2.5}$$

If ρ is the order of the derivative at which the control term u appears with non-zero coefficient, then we can see from (2.5) that a linearising controller is given by:

$$u = \frac{1}{L_g L_f^{\rho-1} h(\mathbf{x})} \left[-L_f^\rho h(\mathbf{x}) + v \right] \tag{2.6}$$

Which will give us (2.3).

Hence, $\alpha(\mathbf{x})$ and $\beta(\mathbf{x})$ in (2.2) are given by:

$$\alpha(\mathbf{x}) := -\frac{L_f^\rho h(\mathbf{x})}{L_g L_f^{\rho-1} h(\mathbf{x})} \tag{2.7}$$

and

$$\beta(\mathbf{x}) := (\gamma(\mathbf{x}))^{-1} \tag{2.8}$$

where:

$$\gamma(\mathbf{x}) := L_g L_f^{\rho-1} h(\mathbf{x}) \tag{2.9}$$

The controller is not necessarily valid globally. We can see from (2.6) that any values of $\mathbf{x}$ such that $\gamma(\mathbf{x}) = 0$ result in a control signal that is not well-defined. Correspondingly, if $\gamma(\mathbf{x}) \to 0$ anywhere in the domain, the control signal becomes arbitrarily large.

Internal Dynamics

Although application of (2.2) linearises the input-output relation (the external dynamics), some of the system dynamics become unobservable (the internal dynamics). The internal dynamics may be unstable.

Stability of the internal dynamics is analysed by looking at the zero dynamics. The zero dynamics result from setting the output identically to zero:

$$y(t) \equiv 0 \Rightarrow h(\mathbf{x}) = L_f h(\mathbf{x}) = \cdots = L_f^{\rho-1} h(\mathbf{x}) = 0 \tag{2.10}$$

We can see from (2.5) that this implies the control is given by:

$$u^z(\mathbf{x}) = \alpha(\mathbf{x}) \tag{2.11}$$

Therefore, substituting (2.11) in (2.1a) gives the zero dynamics of the system. The system is said to be minimum phase if the origin of the zero dynamics is asymptotically stable [12].

Assessing the internal stability is not in general trivial. Stability of the origin of the zero dynamics does not imply global stability of the internal dynamics [12]. It is only a test for local stability. The internal dynamics are generally nonlinear and therefore advanced stability analysis methods are required.

2.1.2 Normal Form

Putting a system in normal form is an attempt to find a state transformation such that the input only appears in one place in the new state equations. Following the procedure for input-otuput linearisation, it can be seen [12] that if we define a new state vector $\boldsymbol{\xi} \in \mathbb{R}^{\rho}$:

$$\boldsymbol{\xi} = \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ y^{\rho-1} \end{bmatrix} \quad \Rightarrow \dot{\boldsymbol{\xi}} = \begin{bmatrix} L_f h(\mathbf{x}) \\ \vdots \\ L_f^{\rho} h(\mathbf{x}) + L_g L_f^{\rho-1} h(\mathbf{x}) u \end{bmatrix} \tag{2.12}$$

then we can design u as described in the previous section to linearise the input-output map. This state vector describes only the external behaviour of the system. The remaining (internal) dynamics are described by state vector $\boldsymbol{\eta} \in \mathbb{R}^{n-\rho}$:

$$\boldsymbol{\eta} = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \vdots \\ \phi_{n-\rho}(\mathbf{x}) \end{bmatrix} \tag{2.13}$$

where $\phi_i(\mathbf{x})$ are designed so that the input does not appear. Therefore, each component of $\boldsymbol{\eta}$ must be a solution of:

$$\frac{\partial \phi_i}{\partial \mathbf{x}} \mathbf{g}(\mathbf{x}) = 0 \tag{2.14}$$

So that when we calculate $\dot{\boldsymbol{\eta}}$, the coefficient of u is always zero. This results in $n-\rho$ PDEs:

$$\sum_{j=1}^{n} \frac{\partial \phi_i}{\partial x_j} g_j(\mathbf{x}) = 0 \tag{2.15}$$

The solution of these PDEs presents a difficulty when putting the system in normal form. In simple cases, each PDE may be separable, but generally their solution is not systematic.

Having put the system in normal form, the zero dynamics (setting $y \equiv 0$) are then given by setting $\boldsymbol{\xi} \equiv 0$. This results in the zero dynamics:

$$\dot{\boldsymbol{\eta}} = \mathbf{f}(\boldsymbol{\eta}, 0) \tag{2.16}$$

If the zero dynamics are stable, the system is minimum phase. Generally, we will not be able prove global stability of the internal dynamics. Therefore we will have to explicitly state the equivalent control $v(\boldsymbol{\eta}, \boldsymbol{\xi})$ for a particular objective (e.g. stabilisation or tracking [12]) and consider quadratic stability of the full system.

Summary of normal form

- Advantages
  - Normal form separates the dynamics into internal and external parts, each of lower order than the original system.
  - The procedure transforms the coordinates such that the control input appears in only one place.
  - The stability of the internal dynamics may be analysed to some extent without having to explicitly state the equivalent control v. Only broad assumptions about the behaviour of ξ are required, i.e. that ξ is bounded and asymtotically approaches zero.
- Disadvantages
  - The solution of the PDEs (2.15) is not in general systematic.
  - Solution of the PDEs may introduce nonrational terms.

– If the control u does not exactly linearise the external dynamics, then terms involving η might be introduced into the equation for $\dot{\xi}$.

2.1.3 Missile example

We present here a model of the longitudinal dynamics of a missile. The missile model presented here is from the instruction manual for [13], which is based on the Reichert missile model [14].

This example is used to show how to put a system in normal form and find the internal dynamics.

Missile in original coordinates

The following longitudinal missile model is based on :

$$\dot{x}_1 = K_1 M C_z(x_1, M, u)\cos(x_1) + x_2 \tag{2.17a}$$

$$\dot{x}_2 = K_2 M^2 C_m(x_1, M, u) \tag{2.17b}$$

where:

$$C_z(x_1, M, u) = z_3 x_1^3 + z_2 x_1^2 + z_1(2 - 1/3M)x_1 + z_0 u \tag{2.18a}$$

$$C_m(x_1, M, u) = m_3 x_1^3 + m_2 x_1^2 + m_1(-7 + 8/3M)x_1 + m_0 u \tag{2.18b}$$

where x_1 = angle of attack, x_2 = pitch rate and M = Mach number (which is generally time-varying). The other coefficients are constants, defined in [13].

The first assumption will be that the angle of attack x_1 is small enough such that the term $\cos(x_1) \simeq 1$ and may therefore be neglected *(this could be included later by either series expansion as a rational approximation or by representing it as an unknown, time-varying scalar. It is not included for now.)*.

Suppose the output to be controlled is the angle of attack: $y = x_1$.

Differentiating the output once, we get the equation for $\dot{y} = \dot{x}_1$, hence an IOL control law is given by

$$u = \frac{1}{K_1 M z_0}(-K_1 M(z_3 x_1^3 + \dots) - x_2 + v) \tag{2.19}$$

This control law results in the closed-loop

$$\dot{x}_1 = v \tag{2.20}$$

$$\dot{x}_2 = K_2 M^2(m_3 x_1^3 + \dots) + \frac{K_2 M^2 m_0}{K_1 M z_0}(-K_1 M(z_3 x_1^3 + \dots) - x_2 + v) \tag{2.21}$$

where we could choose $v = kx_1$, $k < 0$ for example, to stabilise the origin of the output dynamics.

The zero-dynamics of this system are defined by setting $\dot{x}_1 = 0$, $x_1 \equiv 0$, $v \equiv 0$:

$$\dot{x}_1 = 0 \tag{2.22}$$

$$\dot{x}_2 = -\frac{K_2 M^2 m_0}{K_1 M z_0} x_2 \tag{2.23}$$

Clearly the origin of the zero-dynamics is globally stable if and only if $-\frac{K_2 M^2 m_0}{K_1 M z_0} < 0$, in which case the system is minimum phase with this choice of output. However, this would not guarantee that the full closed loop dynamics are globally stable, as the closed-loop system is nonlinear and has nonlinear internal dynamics. Furthermore, this assumes that the Mach number M is constant, which it is not (although we can reasonably expect it to evolve much more slowly than the angle of attack and pitch rate). The next section shows the nonlinear internal dynamics clearly, by putting the system in normal form.

Missile in normal form

To put the missile model into normal form, first define ξ:

$$\xi = y = x_1 \tag{2.24a}$$

$$\dot{\xi} = \dot{y} = \dot{x}_1 \quad \Rightarrow \tag{2.24b}$$

$$\rho = 1, \quad \xi \in \mathbb{R}^1, \quad \eta \in \mathbb{R}^{n-\rho} = \mathbb{R}^1 \tag{2.24c}$$

The system has relative degree $\rho = 1 < n$, system order, so the system has nontrivial internal dynamics.

Next, we want to complete the state transformation $\mathbf{z} = T(x_1, x_2) = \begin{bmatrix} \xi \\ \eta \end{bmatrix}$ that puts the system in normal form.

Now find $\eta = \phi(\mathbf{x})$ such that[1] :

$$\frac{\partial \phi}{\partial \mathbf{x}} \mathbf{g} = \begin{bmatrix} \frac{\partial \phi}{\partial x_1} & \frac{\partial \phi}{\partial x_2} \end{bmatrix} \begin{bmatrix} K_1 M z_0 \\ K_2 M^2 m_0 \end{bmatrix} = 0 \quad \Rightarrow \tag{2.25a}$$

$$K_1 M z_0 \frac{\partial \phi}{\partial x_1} + K_2 M^2 m_0 \frac{\partial \phi}{\partial x_2} = 0 \tag{2.25b}$$

which is separable, of the form:

$$\phi(x_1, x_2) = X_1(x_1) + X_2(x_2) = \frac{1}{K_1 M z_0} x_1 - \frac{1}{K_2 M^2 m_0} x_2 \tag{2.26}$$

which satisfies $\phi(0, 0) = 0$. Hence, substituting (2.24a) :

$$\eta = \frac{1}{K_1 M z_0} \xi - \frac{1}{K_2 M^2 m_0} x_2 \quad \Rightarrow \tag{2.27a}$$

$$\mathbf{z} = T(x_1, x_2) = \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} x_1 \\ \frac{1}{K_1 M z_0} x_1 - \frac{1}{K_2 M^2 m_0} x_2 \end{bmatrix} \quad \Rightarrow \tag{2.27b}$$

$$\mathbf{x} = T^{-1}(\xi, \eta) = \begin{bmatrix} \xi \\ K_2 M^2 m_0 \left(\frac{1}{K_1 M z_0} \xi - \eta \right) \end{bmatrix} \tag{2.27c}$$

We can therefore express the dynamics of the system in normal form as:

$$\begin{aligned} \dot{\xi} =& K_1 M (z_3 \xi^3 + z_2 \xi^2 + z_1 (2 - M/3) \xi) + K_2 M^2 m_0 \left(\frac{1}{K_1 M z_0} \xi - \eta \right) + K_1 M z_0 u \\ \dot{\eta} =& \frac{1}{z_0} (z_3 \xi^3 + z_2 \xi^2 + z_1 (2 - M/3) \xi) + \frac{K_2 M^2 m_0}{K_1 M z_0} \left(\frac{1}{K_1 M z_0} \xi - \eta \right) \dots \\ & \dots - \frac{1}{m_0} (m_3 \xi^3 + m_2 \xi^2 + m_1 (-7 + 8M/3) \xi) \end{aligned}$$

Note, the equation for $\dot{\eta}$ is independent of u and represents the internal dynamics of the system. These dynamics are nonlinear and are not affected by our choice of

[1] If the $\cos x_1$ term is left in the original equation, this leads to $\phi(x) = K_2 M m_0 \ln |\sec x_1 + \tan x_1| - K_1 z_0 x_2$

u. The internal dynamics are nonlinear in ξ, but affine in η, hence we can see why we have linear zero-dynamics, which we can get by setting $\xi \equiv 0$:

$$\dot{\eta} = -\frac{K_2 M^2 m_0}{K_1 M z_0}\eta \tag{2.28}$$

Another point to note here is that we can define the IOL control law in the same way as before, in which case $\dot{\xi} = v$ is independent of η. This means that the internal dynamics are unobservable from the output. However, although the zero-dynamics are by definition independent of ξ, the internal dynamics are not and we cannot therefore infer global stability of the closed loop just from looking at the zero-dynamics.

Normal form and output redefinition

There is no guarantee that a system which is rational in the original states, will be rational when put in normal form (see for example [12], p.518-9).

The solution of the PDEs to put a system in normal form is not generally systematic. Their solution was simple in the missile example dealt with here, because they were separable. In general, there are $n - \rho$ PDEs to solve, one for each element of η. Each PDE is first-order, quasi-linear (linear in the partial derivatives) and homogeneous (right hand side is zero), in n variables.

Hence there are two problems associated with putting the system in normal form, both of which are to do with solving the PDEs for the internal dynamics. The relative degree of the system is dependent on the choice of output, so it is natural to ask whether the output can be redefined so that $\rho = n$, hence no internal dynamics. This would not only eliminate the problems mentioned above, but would also eliminate the test for input-to-state stability of the internal dynamics. Conditions for the existence of an output that results in $\rho = n$ can be found in [12]. However, this assumes that we are free to choose the output. Also, in order to be useful, the new output would have to imply the desired behaviour of the original output.

The technique described here for feedback linearisation of SISO nonlinear systems is a specific case of more general FL for MIMO systems. The reader is referred to [15] and to a greater extent [16] for a treatment of FL for MIMO systems.

2.2 Robust Feedback Linearisation in the Literature

All of the previous introduction to FL theory has assumed that the plant model is known exactly and that sensors or actuators are either not present or are included in the plant model. In reality, a plant model is never an exact representation of the system. It has also been assumed that the states are available for feedback, which in practice may not be the case. There is no limit on the size of the control signal in a FL controller, however it is more realistic to assume that an actuator will have position and rate limits. When an actuator is at the point of saturation, the actual control effect is no longer performing feedback linearisation of the plant, and therefore stability of the closed loop cannot be assumed (it is also well-known that saturation can lead to wind-up of integrators which is by itself an active area of research).

Feedback linearisation has emerged as a popular control technique in aerospace applications [**?**,17–80], where the plant is often highly nonlinear. FL, often referred to in aerospace applications as nonlinear dynamic inversion (NDI), is an attractive method because of its relatively simple synthesis when compared to other techniques for control of nonlinear systems.

The literature on feedback linearisation/ dynamic inversion is vast. Here we will give an overview of selected papers on FL in aerospace applications in particular, with some overall comments to follow.

Aircraft applications of NDI *Notation: α, β, p, q and r denote repsectively angle of attack, angle of sidelsip, and roll, pitch and yaw rates.*

[47] NDI is used to control a nonlinear aircraft model. The effect of the control

surface deflections is included only on the moment coefficients for the NDI synthesis, but is applied to the actual aircraft model. Success of the proposed controller is demonstrated by simulation.

[48] The (slightly) non-minimum phase characteristic of aircraft is discussed, with particular reference to the vertical/ short take-off and landing (V/STOL) Harrier. The control approach used is to neglect the contribution of the elevators to the lift coefficient when performing NDI synthesis, which is the effect that makes the system non-minimum phase. Hence the NDI synthesis is based on a minimum phase approximation to the actual system. They then apply this controller to the actual aircraft model, *with* that effect included. They show that this leads to good control. There is some analysis showing that the tracking error should still be bounded. The description of the problem includes discussion of the fact that directly applying NDI means trying to control the aircraft in a way is was not designed to be controlled; in effect trying to directly control a variable that only depends weakly on the control surfaces.

[49] NDI is used for control ofpitch rate. μ-synthesis [81] is used for robust design of the external controller (the required behaviour of pitch rate), subject to noise on sensor measurements and model uncertainty. Short-period and phugoid modes are considered. P and PI designs are compared. The stability and performance of the proposed controller is verified by simulation.

[50] An NDI control law, based on time scale separation, is used to decouple the control inputs of a three-axes aircraft model into independent control channels. Robust stability of the closed-loop system is verified using μ-analysis and deGaston-Safonov real multiloop stability margin. The engine throttle setting is assumed to be pilot-controlled.

[51] A short-period model of an F-16 is considered. The aircraft speed and throttle setting are treated as constant. The pilot command chosen for the longitudinal controller is the pitch rate. Control inputs are elevator deflection and thrust vector, which are combined into a single control input. The control inputs are combined such that thrust vectoring is only implemented when elevator deflection is insuffi-

cient to meet the required change in pitching moment. The control surface deflection is assumed to contribute to the pitching moment only and not to the forces. The control structure is NDI, with PI and feedforward compensation. The pitch rate dynamical equation is perturbed with uncertain parameters representing uncertainty in the dynamic equation and unmodelled dynamics. Robust stability and performance is verified using μ-analysis on the linear aircraft model and simulations.

[52] Missile pitch and yaw accelerations and roll rate are controlled by NDI, with time-scale separation, for a three-axes model. There are two main methods of robustness analysis: gain & phase margins [81] and nonlinear simulations. Firstly, linearisation of the missile and controller and using classical and vector gain and phase margins. Secondly, nonlinear simulations, introducing gain variations and delays at the plant input until limit cycle behaviour was observed. Thirdly, Monte Carlo simulations, perturbing some aerodynamic parameters randomly by up to 25% and observing steady-state error in acceleration and body axis roll rate.

[53] NDI is applied to an aircraft model. Uncertain parameters are included. The controller is tuned by performing Monte Carlo simulations, and evaluating a performance cost function. In the end, stochastic robustness guarantees are given. For the choice of outputs in this paper, there are no zero-dynamics. (see also [54])

[55] Robustness of an NDI controller, for the X-38 hypersonic recovery aircraft, is improved by an LQG [82] synthesis for the external controller. Robustness refers to parametric uncertainties, process noise and measurement noise. Two robust stability analyses are employed. Linear robustness analysis is performed, based on a quadratic performance index, of [83]. Nonlinear analysis is based on [84], which searches for the largest domain of stability around an operating point, in the presence of position and rate limits. The controller is designed for a transonic flight condition, then analysed at two additional flight conditions. The roll and yaw rates are controlled using ailerons and rudder. The plant state vector is $\begin{bmatrix}\beta & p & r & \phi\end{bmatrix}$. Only β and ϕ are assumed to be measured. An observer is designed (a Kalman filter) to estimate p and r. A regulator is designed (assuming full state feedback) and combined with the observer, to form an LQG controller. This is all based on a linear plant model.

"Nonlinear" analysis refers to saturations, rather than a nonlinear plant model.

[56] An adaptive DI based control law is designed to control body axis angular rates. An outer loop generates p, q and r commands, which are passed to the inner loop DI. The control derivatives are based on polynomial fits to data tables. The model is affine in the controls, by this approximation. The system is non-square. There are more control effectors than controlled variables, so a control allocation algorithm is used. As noted by the authors, the online control allocation routine can fail to converge in a small number of cases. The main focus is on the control allocation scheme in the presence of various types of failure. Performance of the controller is shown by simulation.

[57] A disturbance observer is used in combination with NDI to mitigate the effect of external disturbances on the performance. The model used is the Reichert longitudinal short-period missile model [85]. The controlled output is a linear combination of α and q. The disturbance is a single scalar (slowly) time-varying input to the pitch acceleration equation. Tuning of the output tracking error dynamics is not discussed. Stability is discussed in terms of classical GM and PM of the linearised closed-loop. Poor performance of the initial design (without observer) in the presence of constant disturbance is shown by simulation. To improve performance, a nonlinear observer is designed to estimate the disturbance, which is assumed to be slowly time-varying. This design procedure is shown to give good performance not only in the case of external disturbance, but also when the disturbances are actually uncertainty in the plant model and unmodelled dynamics, even though the stv assumption is not valid. No general robust stability proof is given. Stability and performance are shown by simulation and by GM and PM.

[58] Attitude control of the X-38 is effected using elevons. Application of DI directly to control bank angle leads to unsatisfactory, nearly nonminimum phase behaviour of zero-dynamics. Output is redefined. Robust stability of the zero dynamics to parametric modelling errors is improved by optimal pole placement. A neural network based adaptive control scheme is used to ensure approx linear i-o behaviour. The closed loop system has internal dynamics and is 3-axes. The plant

state vector is $\begin{bmatrix}\alpha & \beta & \mu & p & q & r\end{bmatrix}$ (where μ is bank angle). Short-period dynamics are considered, so other variables are assumed constant. A linear approximation of the dynamics is used. Effect of controls on forces is neglected in the design process; only effect on moments is considered. Force effects are included in the simulations. Initial FL design to control α and μ leads to vector relative degree 4 and therefore internal dynamics of degree 2. The internal dynamics are shown to be bounded but very lightly damped, leading to unsatisfactory behaviour. This is shown by standard pole analysis and simulation. The output is redefined to deal with this problem. This results in nonlinear internal dynamics, of degree 3. In tuning the tracking of the controller, poles are placed such that the sensitivity of pole locations to parametric modelling errors is minimised, using zero dynamics based on a linearisation of the internal dynamics. An adaptive neural network scheme is then used to achieve as close as possible the desired i-o behaviour in the presence of parametric modelling errors. The stability proof that follows is for local stability of the internal dynamics, using a converse Lyapunov theorem.

[59] A general procedure is outlined for robust stability analysis (finite L_2 gain) of a quasi-LPV system that depends not only on the endogenous parameters (states) but also on some exogenous time-varying parameters. The resulting stability theorem is very similar to the LMIs in Boyd et.al [86]. The exception is that the system matrices in the LMI condition in [59] are parameter-dependent, so making the problem numerically tractable requires gridding. The endogenous nature of the QLPV parameters is dealt with rigorously by looking for a stable ellipsoidal region defined by a quadratic Lyapunov function, within which the endogenous parameters obey the a-priori assumption on their bounds. This is very reminiscent of [87]. This is all illustrated with analysis of two NDI control laws (one P, one PI) for the F/A-18 short-period longitudinal dynamics. The state vector is $\begin{bmatrix}\alpha & q\end{bmatrix}$. Control input is elevator deflection. The QLPV model used is derived from least squares polynomial fits to look-up tables, for a fixed V and h. Controlled output is q. The main work is *a-posteriori* robustness analysis of controllers that are designed for the nominal system. Tuning of the external control is not discussed beyond specifying some desired

dynamics for the ideal case. It is shown that the PI design allows a larger ellipsoidal stability region and is better at maintaining a level of performance (specified by L_2 gain from reference input to performance output) as uncertainty increases.

[60] DI is applied to a nonlinear F-18 aircraft model. The 7 controls are 3-axes thrust vectoring, elevator, aileron, differential horizontal stabilizer and rudder. Controlled outputs are p, q and r. The plant state vector is $\begin{bmatrix}\alpha & \beta & \mu & p & q & r\end{bmatrix}$, where μ is rotation angle over velocity vector, γ is flight path angle and χ is ground following angle. The desired dynamics v is specified based on a linearisation of the model. The 7 controls are amalgamated into just 3 generalised inputs, using a control allocation scheme that is not specified. The resulting system is therefore square. Flight conditions are fixed at $h = 1500$m, $M = 0.6$. μ-synthesis is used to balance performance and robustness requirements in designing the controller. The desired dynamics are specified as first-order responses, with each axis decoupled from the others. The body angular rates are not directly measured, but have noise and pass through second-order filters. Performance and robustness of the closed loop are shown by a doublet command separately on each output, with the others commanded to zero. These are also shown to obey actuator position and rate limits.

[61] Linear Model Predictive Control, rather than PID control, is used after feedback linearisation. LMPC can take account of saturation constraints and therefore avoid them, as well as constraints on state and output. It also performs an input optimization based on a performance index. The model is a nonlinear X-38. The state vector is $\begin{bmatrix}\alpha & \beta & \sigma & p & q & r\end{bmatrix}$, where σ is bank angle. The controls are elevator, ailerons, rudder and 3-axes thrusts. The envelope is divided into 5 phases. Not all of the controls are available, depending on dynamics pressure. Aerodynamic moment coefficients are modelled as linear functions of the states and control surfaces. This model is obtained from look-up tables. Aerodynamic control surface effects on lift and drag are neglected. Force coefficients are neglected. Actuator assignment is defined (differently for each flight phase) such that the system is in fact square, which is important because a state transformation is used to put the system in normal form. The controlled outputs are $\begin{bmatrix}\alpha & \beta & \sigma\end{bmatrix}$, except in one flight phase

where only 2 control surfaces are available, in which case the controlled outputs are changed to β is dropped because the system needs to be square. The system has no internal dynamics, except for one flight phase, where the system has linear zero dynamics of order 2. The performance of the design is shown by simulation on the full model, not a simplified version. Stability during the flight phase where internal dynamics are present is verified purely by showing the eigenvalue history of the linear zero dynamics. The performance of the FL-MPC controller is compared with two FL-PID controllers.

[88] McFarlane-Glover loop shaping is used to give a robustness guarantee for a DI controller for *linear* plants. The design process is for SISO linear plants. The application is for a group of linear models for the short-period longitudinal dynamics of a civil aircraft. Plant states are α and q. q is the controlled output. There is no control surface effect on aerodynamic force. The loop shaping procedure is also shown working in the case where it also has to estimate α. Two different actuator cases are also shown. The design process treats C as a tuning parameter. The simulation results use linearised plant models.

[63] Incremental dynamic inversion is used to command velocity vector angles α, β and μ of a nonlinear aircraft model at high angle of attack. The control law uses inner and outer loops, controlling pqr. Using INDI means the control law is more robust to model errors, however the state derivatives are also required. The performance is shown by simulation, with noise, model error and measurement errors.

[64] Input-output linearisation is made more robust by incorporating an uncertainty and disturbance estimator. The system is assumed to be square. A stability proof for the *output error* dynamics *only* is given and is therefore a proof based on linear system theory. Stability of internal dynamics is not addressed. The disturbances are assumed to be well-behaved in the sense that the ith temporal derivative should be negligible. This is illustrated on stabilistaion and tracking of body roll angle ϕ, using a model of wing-rock motion. The open loop model exhibits limit cycle behaviour. The model is of order 2 and is nonlinear in one equation ($\dot{p}$). With $y = \phi$

the system is relative degree 2, therefore there are no internal dynamics. There is one input (differential aileron), therefore the example is SISO. Time-varying uncertainty is introduced into the aerodynamic parameters. Simulations show the closed loop system to be unstable under IOL alone, whereas with the IOL+UDE it is stable and with good performance.

[65] INDI is used, which is more robust to model mismatch than NDI. INDI requires angular accelerations to be fed back. This is shown to be sensitive to sensor measurement time delays. However, the authors reduce this sensitivity by using a linear filter that predicts angular accelerations. The model is 3-axes, with states (12): body fixed velocity components, body fixed rotational rates, attitude, and position. Sensor and actuator dynamics are included. Sensor measurements have noise and delay. Actuators have position and rate limits. Controlled outputs are the angular rates p, q and r. The control law has inner and outer loops based on the TSS principle - control of β. Thrust is assumed to be controlled to maintain constant airspeed and is not treated. For the simulations, constant speed and altitude are assumed. Tuning the controller is not discussed. The P and I gains of the inner and outer loops are simply stated and are different for the NDI, INDI and PINDI (INDI with prediction). Comparisons of the different control schemes and their robust performances are shown by simulation plots and by Monte-Carlo simulation.

[66]This uses the disturbance estimation technique of [64] on a MIMO system to deal with input disturbance. The authors give a proof that all internal signals of the closed-loop system are bounded, with the input disturbance being the only non-nominal part of the system. Internal dynamics are nonlinear and unstable. Robust tracking performance is shown by simulations.

[67] Hierarchy-Structured Dynamic Inversion (HSDI) (this is basically TSS) is used, together with extensive Monte-Carlo simulation, to tune the nested P gains. Root Sum Square Analysis is used to find out which model uncertainties are influential and the controller tuned accordingly. The model is a 6-dof Automatic Landing Flight Experiment.

2.3 Summary

Much of literature on FL concerns methods to make the closed loop system more robust, by adding to the outer loop to make the system less sensitive to disturbances or uncertainties. Robust stability analysis is generally by linear techniques or by simulations:

FL with linear synthesis techniques Papers that combine FL with linear robust synthesis techniques include those that use

H_∞ [26, 89]

Pole-Placement [43, 47, 52, 58, 90, 91]

μ-synthesis [20, 35, 40, 49, 50, 92, 93]

McFarlane-Glover loop shaping [**?**, 44, 51]

LQ [22, 37, 42, 55, 94, 95]

FL with other synthesis methods FL with sliding mode control is considered in [90, 96, 97].

Adaptive methods in combination with FL [18, 19, 23, 24, 31, 33, 56, 58, 98–100], in some cases in combination with neural networks [19, 33, 58].

TSS and INDI Early applications of TSS are in [35, 47] other papers that adopt this technique are [27, 31–35, 38, 39, 43, 47, 52, 54, 63, 65, 67, 101].

INDI is designed to increase robustness to model uncertainty. Papers include [63, 65, 66].

Analysis of FL Many papers primarily use simulations for analysis [19, 22–28, 30–38, 42, 44, 46, 47, 49, 51, 56–58, 60, 61, 63, 64, 66, 89–95, 97–100, 102–107]

More extensive Monte Carlo Simulations are used in [18, 52, 53, 65, 67], including [54, 108], which provide a stochastic proof of robust stability.

General proofs Some papers attempt general stability proof for FL control.

Ideal TSS is assumed in [32].

A linear system with nonlinear actuator is considered in [106].

a-priori assumption that nonlinear functions are bounded is employed in [34] and [101].

Other papers require much more restricive assumptions to prove stability. These papers require assumptions such as assuming the zero-dynamics are stable, linear or bounded, or that there is no uncertainty on the model [17, 32, 34, 46, 48, 48, 90, 95, 101, 106, 109–117], whereas [58] gives a general proof of stability only for the JL of the internal dynamics.

Linear analysis Many papers employ only linear techniques of stability analysis, which are not rigorous for nonlinear models.

Classical gain and phase margins are used in [**?**, 42–44, 52, 57], whereas μ-analysis is used in [20, 40, 43, 50, 51]. The assumption that the system can be modelled a linear time-varying is used in [118].

Other analysis techniques A sum-of-squares technique is used in [21] for robust stability anaysis of the closed loop.

[29, 59] use a technique that models the system as QLPV, and uses gridding over the associated parameter space of the LMIs to assess stability.

IQCs are used in [39] to assess robust stability of a QLPV missile model subject to TV parametric uncertainty under FL-TSS control, on the assumption that the endogenous part of the Δ-block is bounded.

The rather brief overview above is intended to motivate the research presented in this thesis by pointing out the restrictive assumptions that are used to give general stability proofs. Linear analysis techniques do not provide rigorous robust stability proofs for nonlinear systems. As uncertainty is inevitably present on a system model, a nonlinear system under FL will not generally be transformed into an ideal linear system. This is the case whether or not the plant is in principle full-state linearisable, as modelling and state measurement errors will inevitably mean that the system is not exactly linearised.

We set out to give robust stability proofs for uncertain SISO and MIMO systems

in this thesis, in some cases using output rather than full state feedback, via state estimation or sensors. The class of systems considered are those that are rational in the states and uncertain parameters and can admit a linear fractional tranformation (LFT). Under this assumption, we aim to give robust stability analysis of the closed loop systems, by using block-diagonal scalings that represent knowledge of the structure of the Δ-block for these kind of systems. For the reasons given in the previous section, i.e. solution of PDEs and introduction of non-rational terms, we will generally give the plant dynamics in terms of the original system states, rather than transforming it into normal form.

Chapter 3

State Estimation and FL for a Missile

Work in this chapter has been published in the conference proceedings of ECC 2013 [119].

3.1 Introduction

In the synthesis of a feedback linearising controller, it is common to assume that all the necessary states and controlled outputs are available to the controller. In practice, the available measurements will only be a subset of these. In aerospace systems, models are often highly nonlinear and contain several uncertainties. Therefore, robustness of the state observer and robust stability analysis of the closed-loop system is important. Even if inversion was performed exactly, the system may still be unstable due to the nonminimum phase nature of its (generally nonlinear) internal dynamics [12]. For a robust stability guarantee in the presence of uncertainties in the plant model, we cannot assume that inversion is performed exactly.

Literature on dynamic inversion for missile control [36] uses FL control on a MIMO missile model with time-varying uncertainty. State feedback is used. Robust stability of the closed loop is verified using simulations.

[34] provides a stability analysis of a nominal (no uncertainty) MIMO missile model under NDI-TSS, assuming full state feedback. Importantly, this also assumes *a-priori* that functions in the missile model state equations are bounded.

Output feedback is used with NDI-TSS missile control in [38] and [52], where observers are used to estimate unknown states. Stability of the closed loop is verified in [38] by simulation, whereas [52] uses more extensive Monte-Carlo Simulation (MCS), plus linear analysis techniques (GM/PM).

[45] uses a parameter-varying quadratic Lyapunov function to analyse robust stability of a SISO missile model under NDI control, to time-invariant uncertain parameters. Using an uncertain-parameter-dependent Lyapunov function leads to the assumption that uncertainties are time-invariant, in order to be able to write an expression for the time derivative of the Lyapunov function. This leads to a parameter-dependent LMI.

[57] NDI control is used on a SISO nonlinear missile model, with time-varying uncertainty. State feedback is used and the system has nonlinear internal dynamics, however linear stability analysis techniques (GM/PM) are used, along with simulations. This uses a disturbance observer to improve the closed loop robustness to external disturbance signals. A disturbance observer is used in combination with NDI to mitigate the effect of external disturbances on the performance. The model used is the Reichert longitudinal short-period missile model [85]. The controlled output is a linear combination of α and q. The disturbance is a single scalar (slowly) time-varying input to the pitch acceleration equation. Tuning of the output tracking error dynamics is not discussed. Stability is discussed in terms of classical GM and PM of the linearised closed-loop. Poor performance of the initial design (without observer) in the presence of constant disturbance is shown by simulation. To improve performance, a nonlinear observer is designed to estimate the disturbance, which is assumed to be slowly time-varying. This design procedure is shown to give good performance not only in the case of external disturbance, but also when the disturbances are actually uncertainty in the plant model and unmodelled dynamics, even though the stv assumption is not valid. No robust stability proof is given.

Stability and performance are shown by simulation and by GM and PM.

Other missile control techniques in the literature [120] compares the use of H_∞ and μ synthesis for robust control of a pitch-axis missile model. The model used is a JL of the nonlinear missile model around a single flight condition. The controller designed using μ synthesis is shown to give better performance than the H_∞ controller. Robust performance of the controllers on the nonlinear model is demonstrated by simulation only. In [14] similar synthesis techniques are used, but this time scheduled by Mach to cover more of the flight envelope and using observer-based feedback. Again, as the plant is nonlinear, robust performance on the nonlinear model is verified by simulation. [121] demonstrates an improved gain-scheduled H_∞ controller design for the same missile model, again using simulation to show robust preformance on the nonlinear model. [122] again uses this missile model to demonstrate observer-based output feedback H_2 controller synthesis, for parameter-dependent systems. This generates an LPV controller with robust performance demonstrated by simulation on the nonlinear missile model.

[123] uses LPV control for a SISO nonlinear missile model, which includes actuators with saturation. Output feedback is used and the controller is tuned according to H_2/H_∞ performance specifications. Stability of the closed loop is verified by simulation.

Robust stability analysis of systems controlled using feedback linearisation is generally based on simulations e.g. [105], [23]. This also applies to linear parameter varying (LPV) systems e.g. [124]. A systematic, stochastic approach to the nonlinear dynamic inversion synthesis is the focus of [54]. It is possible with these approaches that there is some "worst-case" scenario that is missed, which is why we aim for an analysis technique that will give a robust stability guarantee for all allowed combinations of uncertain parameters. Another interesting approach is observer-based feedback linearisation designed to alleviate the estimated disturbance [105], [90]. These place some restrictive assumptions on the form of the system e.g. full-state linearisable, or uncertainty only on the input channel.

We aim for a simple NDI-TSS controller design. In [39], integral quadratic

constraints (IQCs) are used to perform a robust stability analysis for this system, controlled using feedback linearisation and time scale separation. However, there is no observer i.e. full state feedback is assumed. In [34] the stability analysis assumes the fast subsystem inversion is performed exactly. The assumption that control deflections affect only the moments is carried through from controller synthesis to stability analysis. Also, there is no robustness guarantee, which is the main aim of this chapter.

We derive a method for the synthesis of a robust linear time invariant (LTI) filter to estimate unknown states, by solving a system of linear matrix inequalities (LMIs), which is based on [125]. The filter is designed to minimise the L_2 gain from an external input to the estimation error. Solution of the associated LMIs requires that the plant is stable. Therefore, we assume that we are designing a filter for the controlled plant, because the model in question is only marginally stable. As we do not assume *a priori* knowledge of the reference signal, we cannot use the H_2 method given in [125]. We therefore derive the H_∞ condition for a system with structured uncertainty. We also verify robust stability of the closed-loop system, including the filter, which is again a sufficient LMI condition using a linear differential inclusion (LDI) and scalings, based on a quasi-LPV/LFT form for the system.

This chapter is organised as follows: In Section 3.2 we present a nonlinear missile model. In Section 3.3 we give a simple input-ouput linearising controller using time-scale separation. In Section 3.4, we move from a nonlinear to a quasi-LPV/LFT model, and derive an LMI condition for synthesis of a robust filter. In Section 3.4.2 we show an LMI condition for giving a robust upper-bound on the L_2 gain of the closed-loop system. Nonlinear simulation with uncertainties and filter are given in Section 3.5. Here we also give robust stability analysis using the LMI of Section 3.4.2. Conclusions are given in Section 3.6. Notation: $\|X\|$ means the maximum singular value of X. $F_l(X, \Delta)$ means the lower linear fractional transformation of X with Δ [126].

Table 3.1: Notation - chapter 3

Symbol	Meaning
α	angle of attack
q	pitch rate
η, η_c	normal acceleration, commanded value
u	control input (tail fin deflection)
M	Mach
C_z, C_m	normal force and moment coeffs.
$\tilde{C}_z, \tilde{C}_m$	zero-input forms of the above
y_m	measured output (pitch rate)
x	states of the closed loop plant-controller-actuator
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$	closed loop system matrices
θ	a vector of time-varying parameters
z	the output to estimate with the filter
$\hat{x}$	state vector of the filter
A_f, B_f, L_f	filter matrices

3.2 The Plant

The nonlinear missile model from [121], has states angle of attack and pitch rate α (rad) and q (rad/s), output normal acceleration η (m/s^2) and input tail fin deflection u (rad):

$$\dot{\alpha}(t) = K_1 M(t) C_z(\alpha(t), M(t), u(t)) \cos(\alpha) + q(t) \tag{3.1a}$$

$$\dot{q}(t) = K_2 M^2(t) C_m(\alpha(t), M(t), u(t)) \tag{3.1b}$$

$$\eta(t) = K_3 M^2(t) C_z(\alpha(t), M(t), u(t)) \tag{3.1c}$$

The Mach number $M(t)$ is treated as an exogenous variable. The term $\cos(\alpha) \approx 1$ for the operating range and is therefore neglected from hereon. The aerodynamic

coefficients are given by:

$$\begin{aligned} C_z(\alpha, M, u) =& z_3|\alpha|^2\alpha + z_2|\alpha|\alpha \\ &+ z_1(2 - M/3)\alpha + z_0 u \\ C_m(\alpha, M, u) =& m_3|\alpha|^2\alpha + m_2|\alpha|\alpha \\ &+ m_1(-7 + 8M/3)\alpha + m_0 u \end{aligned} \tag{3.2}$$

Measurements available to the controller are q and η. Physical data is given in Table 3.3.

The plant is augmented with a second-order actuator with input commanded tail fin deflection $u_c(t)$(rad) and output $u(t)$:

$$\begin{bmatrix} \dot{u}(t) \\ \ddot{u}(t) \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -{\omega_a}^2 & -2\xi_a\omega_a \end{bmatrix} \begin{bmatrix} u(t) \\ \dot{u}(t) \end{bmatrix} + \begin{bmatrix} 0 \\ {\omega_a}^2 \end{bmatrix} u_c(t) \tag{3.3}$$

The operating range is given by $|\alpha(t)| \leq 20^\circ$ and $1.5 \leq M(t) \leq 3$. The controller should achieve robust stability over the operating range, to uncertainty in the α and u dependent parts of C_m that can vary independently by $\pm 25\%$. Performance specifications are that the controller should track step commands η_c with maximum time constant 350ms, overshoot 10% and steady-state error 1%. The maximum tail fin deflection rate should meet $|\dot{u}(t)| \leq 25^\circ/\text{s}$ for step command $\eta_c = 1g$.

3.3 Controller Synthesis

We follow the method in [39] for controller synthesis. The model is nonminimum phase, hence a time-scale separation technique is used. Neglecting the actuator, the plant is split into slow and fast subsystems. The u-dependent term in C_z is neglected in the slow subsystem.

Slow subsystem The slow subsystem has one state α, input q_c (commanded value for pitch rate) and output η. Defining $\tilde{C}_z := C_z(\alpha, M, 0)$, $e_\eta := \eta - \eta_c$ and following standard input-output linearising controller synthesis [12], the slow subsystem controller is

$$q_c = -K_1 M \tilde{C}_z + \left(K_3 M^2 \frac{\partial C_z}{\partial \alpha} \right)^{-1} (-k_1 e_\eta + \dot{\eta}_c) \tag{3.4}$$

which will, for the *approximate* slow subsystem, achieve asymptotic tracking of η_c with first-order dynamics. The term $\partial C_z(|\alpha|, M)/\partial\alpha$ has one positive root $|\alpha| = 73 - 69^\circ$ for $M = 1.5 - 3$, which is far outside the operating range, hence the controller is well-defined.

Fast subsystem The fast subsystem has one state q, input u_c and output q. Defining $\tilde{C}_m := C_m(\alpha, M, 0)$, $e_q := q - q_c$ an input-output linearising controller with second-order dynamics is given by

$$u_c = \frac{-\tilde{C}_m}{m_0} + \left(K_2 M^2 m_0\right)^{-1} \left(-k_2 q - k_3 \int e_q dt\right) \tag{3.5}$$

to achieve asymptotic tracking of q_c.

The controller gains k_1, k_2 and k_3 were tuned in [39] using a genetic algorithm, for particular values of Mach. Here, we use the gains calculated in [39] for the nominal Mach value $M = 2.25$: $k_1 = 4.69$, $k_2 = 18.3$, $k_3 = 211$.

3.4 Filter Synthesis and Stability Analysis

3.4.1 Robust Filter Synthesis

This section derives an LMI for robust filter synthesis for a system which is represented as an LFT. This is the same kind of system representation as is used for robust stability analysis of the closed loop system. Although the filter is initially designed in open loop, the filter will be used to provide an estimate of the angle of attack to the controller when we perform robust stability analysis and simulations in the final section. Hence the filter is being used in place of a more traditional observer.

It is more common for observers to be used for state estimation, however tuning of observers is not automatic. An observer in the form introduced by Luenberger [127] for linear systems is a system whose state matrices are the same as that of the plant, but with an additional input vector. This extra input is the difference between the plant measured output and the observer estimate of the measured output. This

error is called the *residual*. The design variable in a Luenberger-type observer is the distribution matrix of the residual, which is commonly called the observer gain matrix; all other observer matrices being the same as that of the plant [128].

It is well known that for linear systems, with appropriate choice of observer gain matrix, the observer state will asymptotically approach the plant state [128], or equivalently the error dynamics will approach zero. Therefore, choice of observer gain matrix is the crucial design variable and some standard tools have been developed for designing it in software packages such as MATLAB.

Another nice feature of the Luenberger-type observer is that a *separation principle* applies [128], i.e. the observer and controller for a given plant can be designed independently. It is important to note that for this to apply rigorously, the plant is assumed to be linear and its matrices well-known.

The analogue of the Luenberger observer for a nonlinear system was developed by Thau [129]. In this case, we have the same basic idea as a Luenberger observer i.e. the observer has the same structure as the plant, disturbed by the residual as an extra input. Note we still have the observer gain matrix as a design variable. Other types of function have been explored to introduce the residual to the observer, for example the use of a nonlinear function of the residual in [130]. Note that even in the relatively simple case of a Thau-type observer, we have a matrix design variable and the choice of it is more complicated than in the case of a linear plant.

Appendix C gives a derivation of how the design of a traditional (Thau-type) nonlinear observer for an LFT system leads to a bilinear matrix inequality (BMI).

BMIs have been proved to be more difficult to solve than LMIs [131]. Although control problems such as controller synthesis can be formulated as BMIs, there is significant numerical difficulty in solving them. There has been considerable research effort in the past twenty years into algorithms for efficient solution of BMI problems [132–135] and there is even software available for doing so such as that presented in [134].

This has led to several examples of successful application of some of these algorithms to solve controller synthesis problems for various systems [135–139], including

output feedback [140] and robust controller synthesis [141].

However despite the successful applications listed above, none of the available methods for the solution of BMI problems guarantee a globally optimal solution and the existing methods require a high computational burden [142]. On the other hand efficient methods for solution of LMIs have been available for some time [86] and relatively user-friendly and well-developed software packages are available, including for MATLAB [143].

The filter synthesis presented here is appealing because the filter matrices are produced by solving an LMI and because the LFT system representation is the same kind of representation that we use for robust stability analysis. The filter lacks the observer's residual term, and hence may be more susceptible to performance problems when the initial measurement error is large. However we show a successful application of it to the missile presented here, in estimating an unknown state. Note the results on filter synthesis may be of interest by themselves for an open-loop filter, not necessarily just for state estimation for feedback to a controller.

In [125], LMI conditions are derived for synthesis of an LTI filter for a system with uncertainty that can be represented as an LFT. For a system with structured uncertainty, only the H_2 result is given in [125]. However, it is stated in [125] that the H_∞ result can be derived using the methods given in that paper, which is what we do here.

The controlled plant with actuator can be written in quasi-LPV form, treating a reference demand r as the external input to the system and measured output y_m:

$$\begin{bmatrix} \dot{x}(t) \\ y_m(t) \end{bmatrix} = \begin{bmatrix} \mathcal{A}(\theta(t)) & \mathcal{B}(\theta(t) \\ \mathcal{C}(\theta(t)) & \mathcal{D} \end{bmatrix} \begin{bmatrix} x(t) \\ r(t) \end{bmatrix} =: M(\Delta(t)) \begin{bmatrix} x(t) \\ r(t) \end{bmatrix} \tag{3.6}$$

$x \in \mathbb{R}^n$, $y_m \in \mathbb{R}^{n_y}$, $r \in \mathbb{R}^{n_r}$, $\theta \in \mathbb{R}^{n_\theta}$, where x contains controller, plant and actuator states and $\theta(t)$ contains time-varying parameters and states in which the plant is nonlinear. We assume that the bound on the endogenous part of the quasi-LPV form is valid. It is generally necessary [59] to find a bound on the energy of the input r for which we can find a bound on the states that form the endogenous part of $\theta(t)$. This can be made computationally tractable by finding an ellipsoidal region

that lies within a polytope defined by the bound on Δ (a condition for which is given in [144], [87]). This is something we will address in chapter 7.

Note $\mathcal{D}$ is assumed to be constant for the available measured output. In the sequel, the measured output is actually just one of the states, in which case $\mathcal{D}$ is zero and $\mathcal{C}$ is constant.

With the system in quasi-LPV form, the uncertainties and nonlinear terms can be pulled out, and the system written in LFT form. The process of doing this is to normalise each of the parameters in $\theta(t)$ such that each each time-varying parameter is given by $\theta_i(t) = \bar{\theta}_i + s_i\delta_i(t)$, where $\bar{\theta}_i$ is the nominal value of $\theta_i(t)$ and $|\delta_i(t)| \leq \sigma^{-1}$ is the (bounded) variation of each parameter. Note the scaling factor s_i can be chosen such that the variation of all the parameters is normalised with the same bound. The Δ-block is therefore bounded by $\|\Delta\| \leq \sigma^{-1}$. This is a standard process of 'pulling out' time-varying parameters to produce an LFT model and can be achieved efficiently using the MATLAB toolbox [13].

$$\begin{aligned} M(\Delta(t)) &= F_l(H, \Delta) \\ &= \begin{bmatrix} A & B_r \\ C_y & D_{yr} \end{bmatrix} + \begin{bmatrix} B_p \\ D_{yp} \end{bmatrix} \Delta(t)(I - D_{qp}\Delta(t))^{-1} \begin{bmatrix} C_q & D_{qr} \end{bmatrix} \end{aligned} \tag{3.7}$$

where

$$H = \left[\begin{array}{cc:c} A & B_r & B_p \\ C_y & D_{yr} & D_{yp} \\ \hdashline C_q & D_{qr} & D_{qp} \end{array}\right] \tag{3.8}$$

and $\Delta(t)$ is a block-diagonal matrix with n_θ blocks. Here, each diagonal block consists of a repeated scalar $\delta_i(t)$; the normalised variation in one of the elements of $\theta(t)$. The size of each block, k_i, depends on the degree of nonlinearity with which that scalar appears in the system equations. $\Delta(t)$ therefore represents structured uncertainty and is norm-bounded, with allowed values in the set (here $l = n_\theta$):

$$\begin{aligned} \mathbf{\Delta} := \{\mathrm{diag}(\delta_1 I_{k_1}, \ldots, \delta_l I_{k_l}) : \|\Delta\| \leq \sigma^{-1}, \\ \delta_i \in \mathbb{R},\ \sigma > 0\} \subset \mathbb{R}^{n_p \times n_p} \end{aligned} \tag{3.9}$$

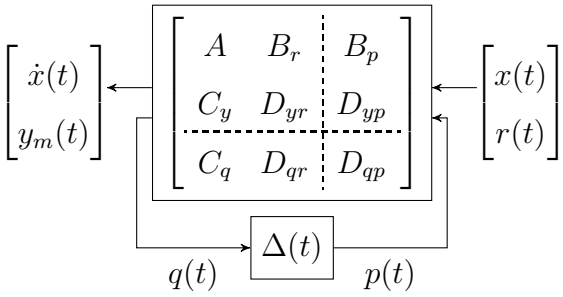


Figure 3.1: Linear fractional representation of the controlled plant (3.6).

In order for the following filter synthesis matrix inequality to be affine in the variables, we have to make a simplification of the LFT model of the system. Specifically, we set $D_{qr} = 0$, which means we neglect the variation of $\mathcal{B}(\theta(t))$ from its nominal value. Hence we effectively assume that $\mathcal{B}$ is constant. Although we do this at the stage of filter synthesis, this assumption is not carried through to the final robust stability analysis when the filter is used in the closed loop.

$$\dot{x}(t) = Ax(t) + B_r r(t) + B_p p(t) \tag{3.10a}$$

$$y_m(t) = C_y x(t) + D_{yr} r(t) + D_{yp} p(t) \tag{3.10b}$$

$$q(t) = C_q x(t) + D_{qp} p(t) \tag{3.10c}$$

$$p(t) = \Delta(t) q(t) \tag{3.10d}$$

with p, $q \in \mathbb{R}^{n_p}$. We want to estimate

$$z(t) := Lx(t), \quad L \in \mathbb{R}^{n_z \times n}$$

with a full-order LTI filter of the form

$$\dot{\hat{x}}(t) = A_f \hat{x}(t) + B_f y_m(t) \tag{3.11a}$$

$$\hat{z}(t) = L_f \hat{x}(t) \tag{3.11b}$$

where $A_f \in \mathbb{R}^{n \times n}$, $B_f \in \mathbb{R}^{n \times n_y}$ and $L_f \in \mathbb{R}^{n_z \times n}$ are the state-space matrices of the filter, to be found. Substitute (3.10b), in (3.11a):

$$\dot{\hat{x}} = A_f \hat{x}(t) + B_f \left[C_y x(t) + D_{yr} r(t) + D_{yp} p(t) \right] \tag{3.12}$$

Define the augmented state vector $\mu := \begin{bmatrix} x^T & \hat{x}^T \end{bmatrix}^T \in \mathbb{R}^{2n}$, then:

$$\begin{aligned} \dot{\mu} = \begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} &= \begin{bmatrix} A & 0 \\ B_f C_y & A_f \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B_r \\ B_f D_{yr} \end{bmatrix} r + \begin{bmatrix} B_p \\ B_f D_{yp} \end{bmatrix} p \\ &=: \bar{A}\mu + \bar{B}r + \bar{L}p \end{aligned} \tag{3.13}$$

Define the output of the augmented system as the estimation error:

$$e_z := z - \hat{z} = Lx - L_f\hat{x} = \begin{bmatrix} L & -L_f \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} =: \bar{C}\mu \tag{3.14}$$

Then, defined in terms of μ, (3.10c) becomes:

$$q = \begin{bmatrix} C_q & 0 \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + D_{qp}p =: \bar{E}\mu + D_{qp}p \tag{3.15}$$

and (3.10d), as before (augmenting with the filter does not change the p, q relation). We can represent the augmented system as an LFT, analogously to the controlled plant, however as $\mathcal{B}$ and $\mathcal{D}$ are known constant matrices, we can represent the system in the following LFT:

$$\begin{aligned} \bar{A}_\Delta &:= F_l\left(\begin{bmatrix} \bar{A} & \bar{L} \\ \bar{E} & D_{qp} \end{bmatrix}, \Delta(t)\right) \\ &= \bar{A} + \bar{L}\Delta(t)(I - D_{qp}\Delta(t))^{-1}\bar{E} \end{aligned} \tag{3.16}$$

This means that (3.13), (3.14), (3.15), (3.10d) are assumed to be equivalent to:

$$\dot{\mu}(t) = \bar{A}_\Delta\mu(t) + \bar{B}r(t) \tag{3.17a}$$

$$e_z(t) = \bar{C}\eta(t) \tag{3.17b}$$

Note that the representation of the system in (3.17) is not unique and for now we have not proved that such a representation exists. Note in particular that if the term $(I - D_{qp}\Delta(t))$ in (3.16) is not invertible for some allowed value of $\Delta(t)$, then the representation is not valid.

If we can prove that $(I - D_{qp}\Delta(t))$ is invertible for all $\Delta(t) \in \mathbf{\Delta}$ then the system representation in (3.17) is valid and is therefore said to be *well-posed*.

We would like to find the state-space filter matrices A_f, B_f and L_f that minimise, in a H_∞ sense, the L_2 gain from r to e_z. This and the well-posedness of (3.17) is addressed in the following theorem.

Theorem 1. *Consider the LDI* (3.17) *and let* $\sigma > 0$ *be given. Assume that the energy of the input is such that the bound on the endogenous part of* $\theta(t)$ *is valid. Then there exists a robust LTI filter such that the* L_2 *gain from* r *to* e_z *is less than* β *if the following matrix inequalities are satisfied:* $P_0 > 0$, $P_1 - P_0 > 0$ *and* (3.18) *(below)*

$$X := \begin{bmatrix} X_1 & X_3 \\ \star & X_2 \end{bmatrix} < 0 \tag{3.18}$$

where

$$X_1 := \begin{bmatrix} \Psi_{11} & \Psi_{12} & \Psi_{13} \\ \star & \Psi_{22} & \Psi_{23} \\ \star & \star & \Psi_{33} \end{bmatrix}$$

$$\begin{aligned} \Psi_{11} &= P_1 A + M_B C_y + A^T P_1 + C_y^T M_B^T + C_q^T S C_q \\ \Psi_{12} &= M_A + A^T P_0 + C_y^T M_B^T \\ \Psi_{13} &= P_1 B_p + M_B D_{yp} + C_q^T S D_{qp} + C_q^T G \\ \Psi_{22} &= M_A + M_A^T, \quad \Psi_{23} = P_0 B_p + M_B D_{yp} \\ \Psi_{33} &= D_{qp}^T S D_{qp} + D_{qp}^T G - G D_{qp} - \sigma^2 S \end{aligned}$$

$$X_2 := \begin{bmatrix} -\beta^2 I_{n_r} & 0 \\ 0 & -I_{n_z} \end{bmatrix}$$

$$X_3 := \begin{bmatrix} P_1 B_r + M_B D_{yr} & L^T \\ P_0 B_r + M_B D_{yr} & -M_L^T \\ 0 & 0 \end{bmatrix}$$

Furthermore

- *The filter state-space matrices are given by*

$$A_f = P_3^{-1} M_A P_3^{-1} \tag{3.19}$$

$$B_f = P_3^{-1} M_B \tag{3.20}$$

$$L_f = M_L P_3^{-1} \tag{3.21}$$

  where $P_3 = P_0^{1/2}$.

- *The filter is stable.*

- *The LDI* (3.17) *is well-posed.*

Remark 1: A Lyapunov function proving the stability and L_2 gain of the filter is given by $V(\mu) = \mu^T P \mu$, where $P = \begin{bmatrix} P_1 & P_3 \\ \star & I_n \end{bmatrix} \in \mathbb{R}^{2n \times 2n}$.

Remark 2: The variables in (3.18) are P_1, P_0, $M_A \in \mathbb{R}^{n \times n}$, $M_B \in \mathbb{R}^{n \times n_y}$, $M_L \in \mathbb{R}^{n_z \times n}$ and $\beta \in \mathbb{R}$. Notice that the matrix inequalities are linear in the decision variables. Therefore, standard MATLAB tools [143] can be used to obtain filter matrices minimising an upper bound on the L_2 gain.

Proof. By standard control system theory [86], if $\exists\, 0 < P \in \mathbb{R}^{2n \times 2n}$ and $\beta > 0$ such that

$$\begin{bmatrix} \bar{A}_\Delta^T P + P \bar{A}_\Delta + \bar{C}^T \bar{C} & P\bar{B} \\ \star & -\beta^2 I \end{bmatrix} < 0,\ \forall\, \Delta(t) \in \mathbf{\Delta} \tag{3.22}$$

then the induced L_2 gain from reference r to estimation error e_z for the augmented system (3.17) is less than β for all permitted values of Δ.

By Schur complement [86], (3.22) is equivalent to $-\beta^2 I < 0$ (which is obvious), together with

$$T_1 + T_2 \Delta (I - T_4 \Delta)^{-1} T_3 + T_3^T (I - T_4 \Delta)^{-T} \Delta^T T_2^T < 0 \tag{3.23}$$

where

$$\begin{aligned} T_1 &:= \bar{A}_\Delta^T P + P \bar{A}_\Delta + \bar{C}^T \bar{C} + \frac{1}{\beta^2} P \bar{B} \bar{B}^T P \\ T_2 &:= P\bar{L},\ T_3 := \bar{E},\ T_4 := D_{qp} \end{aligned} \tag{3.24}$$

We now associate with $\mathbf{\Delta}$ the subspaces of block-diagonal scaling matrices (representing relationships for real, structured uncertainty):

$$\mathbf{S} := \left\{\text{diag}(S_1, \ldots, S_l) : 0 < S_i \in \mathbb{R}^{k_i \times k_i}\right\} \subset \mathbb{R}^{n_p \times n_p}$$
$$\mathbf{G} := \left\{\text{diag}(G_1, \ldots, G_l) : G_i = -G_i^T \in \mathbb{R}^{k_i \times k_i}\right\} \subset \mathbb{R}^{n_p \times n_p}$$

Then [145], [87] (3.23) holds and is well-posed $\forall\, \Delta(t) \in \mathbf{\Delta}$, i.e. $\det(I - T_4\Delta) \neq 0$, if $\exists\, S \in \mathbf{S}$ and $G \in \mathbf{G}$ such that

$$Z := \begin{bmatrix} T_1 + T_3^T S T_3 & T_2 + T_3^T S T_4 + T_3^T G \\ \star & T_4^T S T_4 + T_4^T G - G T_4 - \sigma^2 S \end{bmatrix} < 0 \tag{3.25}$$

This is a sufficient condition and therefore introduces conservatism.

The augmented system matrices, as defined in (3.13), (3.14) and (3.15), are substituted into (3.24) and in turn to (3.25), with P partitioned as $\begin{bmatrix} P_1 & P_3 \\ \star & P_2 \end{bmatrix}$. The filter matrices are not fixed and therefore, without loss of generality [146], we can assume that $P_2 = I_n$. This results in a nonaffine matrix inequality $Z < 0$, with variables P_1, P_3, S, G, A_f, B_f, L_f and β. The goal is to find the filter matrices that minimise β, subject to $Z < 0$, $P > 0$, $\beta > 0$, $S \in \mathbf{S}$ and $G \in \mathbf{G}$.

Making Z affine requires a nonlinear change of variables and the use of Schur complement. First, define $P_0 := P_3 P_3^T$, so $P_3 = P_0^{1/2}$. By Schur complement on P, we require $P_0 > 0$ and $P_1 - P_0 > 0$. Next, we define the new variables $M_A := P_3 A_f P_3^T$, $M_B := P_3 B_f$ and $M_L := L_f P_3^T$. Then, we pre and post multiply Z by J^T and J respectively, where $J := \text{diag}\left(I_n, P_3^T, I_{n_p}\right)$. This results in the new matrix inequality

$$\bar{Z} := J^T Z J = \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ \star & Z_{22} & Z_{23} \\ \star & \star & Z_{33} \end{bmatrix} < 0 \tag{3.26}$$

where

$$Z_{11} = P_1 A + M_B C_y + A^T P_1 + C_y^T M_B^T + L^T L + \ldots$$
$$+ C_q^T S C_q + \beta^{-2}(P_1 B_r + M_B D_{yr})(B_r^T P_1 + D_{yr}^T M_B^T)$$

$$Z_{12} = M_A + A^T P_0 + C_y^T M_B^T - L^T M_L + \ldots$$
$$\cdots + \beta^{-2}(P_1 B_r + M_B D_{yr})(B_r^T P_0 + D_{yr}^T M_B^T)$$

$$Z_{13} = P_1 B_p + M_B D_{yp} + C_q^T S D_{qp} + C_q^T G$$

$$Z_{22} = M_A + M_A^T + M_L^T M_L \ldots$$
$$\cdots + \beta^{-2}(P_0 B_r + M_B D_{yr})(B_r^T P_0 + D_{yr}^T M_B^T)$$

$$Z_{23} = P_0 B_p + M_B D_{yp} \quad \text{and} \quad Z_{33} = \Psi_{33}$$

It can be seen that $\bar{Z}$ is still not affine in the variables, so we use Schur complement. We have $\bar{Z} = X_1 - X_3 X_2^{-1} X_3^T < 0$, with X_1, X_2 and X_3 defined as in (3.18) and $X_2 < 0$, hence by Schur complement $\bar{Z} < 0$ is equivalent to $\begin{bmatrix} X_1 & X_3 \\ \star & X_2 \end{bmatrix} < 0$.

Note if the LMI is feasible then $\Psi_{22} = M_A + M_A^T < 0$. This implies that the filter is stable, because the filter state matrix is $A_f = P_3^{-1} M_A P_3^{-1}$ and P_3 (and hence its inverse) are positive definite. □

3.4.2 Robust L_2 Gain

In this section, we present a method for robust stability analysis, by solving a system of LMIs. The proof is straightforward to derive from the literature [86], [87], [144]. We give a more detailed proof in chapter 5. In Section 3.5, we will apply this analysis to the closed-loop system formed by the plant, controller, actuator and filter.

The system under consideration is again an LDI, similarly to the case for filter synthesis, with $\mathbf{\Delta}$, $\mathbf{S}$ and $\mathbf{G}$ defined analogously. The external input is $r \in \mathbb{R}^{n_r}$, output for performance analysis is $e \in \mathbb{R}^{n_e}$ and states $x \in \mathbb{R}^n$:

$$\dot{x}(t) = Ax(t) + B_r r(t) + B_p p(t) \tag{3.28a}$$
$$e(t) = C_e x(t) + D_{er} r(t) + D_{ep} p(t) \tag{3.28b}$$
$$q(t) = C_q x(t) + D_{qr} r(t) + D_{qp} p(t) \tag{3.28c}$$
$$p(t) = \Delta(t) q(t) \tag{3.28d}$$

where $q, p \in \mathbb{R}^{n_p}$.

Theorem 2. *For a given $\sigma > 0$ and assuming the energy of the input is such that the bound on the endogenous part of $\theta(t)$ is valid, if $\exists\ P > 0$, $S \in \mathbf{S}$, $G \in \mathbf{G}$ and $\gamma > 0$, such that LMI (3.29) holds, then the LFT system (3.28) has a finite L_2 gain from input r to output e, with upper bound γ, $\forall\ \Delta \in \mathbf{\Delta}$. Moreover, the LDI is well-posed, i.e.* $\det\left(I - D_{qp}\Delta(t)\right) \neq 0$.

$$\Pi := \begin{bmatrix} \Pi_{11} & \Pi_{12} & \Pi_{13} \\ \star & \Pi_{22} & \Pi_{23} \\ \star & \star & \Pi_{33} \end{bmatrix} < 0 \tag{3.29}$$

where

$$\Pi_{11} = PA + A^T P + C_e^T C_e + C_q^T S C_q$$

$$\Pi_{12} = PB_r + C_e^T D_{er} + C_q^T S D_{qr}$$

$$\Pi_{13} = PB_p + C_e^T D_{ep} + C_q^T S D_{qp} + C_q^T G$$

$$\Pi_{22} = D_{er}^T D_{er} - \gamma^2 I_{n_r} + D_{qr}^T S D_{qr}$$

$$\Pi_{23} = D_{er}^T D_{ep} + D_{qr}^T S D_{qp} + D_{qr}^T G$$

$$\Pi_{33} = D_{ep}^T D_{ep} + D_{qp}^T S D_{qp} - \sigma^2 S + D_{qp}^T G - G D_{qp}$$

Given that we only have a sufficient condition for stability, we may find that the LMIs are not feasible with $\sigma = 1$. This means that we cannot find, by this method, a single quadratic Lyapunov function that guarantees stability over all allowed values of $\Delta(t)$. By increasing σ iteratively, we may be able to satisfy the LMIs, at the cost of reducing the bound on Δ.

3.5 Simulation & Results

Filter We aim for a filter that will give an estimate of α, using pitch rate q as the measured output from the plant. The controlled plant with actuator can be

written in quasi-LPV form, treating the reference demand as the external input to the system, $r := \begin{bmatrix} \eta_c & \dot{\eta}_c \end{bmatrix}^T$ and measured output $y_m := q$. In order to do this, we define $|\alpha(t)| \in [0, 0.349\text{rad}]$, $C_m = d_1(t)\tilde{C}_m(\alpha(t), M(t)) + d_2(t)m_0u(t)$, with d_1 and $d_2 \in [0.75, 1.25]$. Then $x := \begin{bmatrix} \alpha & q & u & \dot{u} & e_q \end{bmatrix}^T$ and $\theta(t) = \begin{bmatrix} |\alpha| & M & d_1 & d_2 \end{bmatrix}^T$. We create the LFT using the free MATLAB toolbox [13] and the LFT is normalised such that $\sigma = 1 \Leftrightarrow \|\Delta\| = 1$. We obtain an LFT with $\Delta(t) \in \mathbb{R}^{17\times 17}$: 4 in $\delta_\alpha(t)$, 11 in $\delta_M(t)$, 1 in $\delta_{d1}(t)$ and 1 in $\delta_{d2}(t)$, where these are the normalised variations in the elements of $\theta(t)$ about their nominal values.

Theorem 1 is used to obtain the filter state-space matrices. We find that the LMIs are not feasible with $\sigma^2 = 1$, which means we cannot find a filter that guarantees a robust L_2 gain from r to e_z for all allowed values of $\theta(t)$. We therefore increase σ^2 iteratively, until we find the smallest value for which the LMIs are feasible. The smallest value is $\sigma^2 = 1.36$, which implies $\|\Delta\| = 0.86$, with a corresponding decrease in the range of $\theta(t)$ over which robust stability is guaranteed. The filter matrices obtained are:

$$A_f = \begin{bmatrix} -8245 & 3127 & 1492 & -155.5 & 43700 \\ 4003 & -1692 & -797.0 & 58.93 & -21200 \\ 1190 & -429.6 & -313.9 & 85.77 & -6308 \\ 182.3 & -126.5 & -51.79 & -18.87 & -960.1 \\ 43590 & -16500 & -7873 & 821.6 & -231100 \end{bmatrix}$$

$$B_f = \begin{bmatrix} -35150 & 17050 & 5074 & 773.4 & 185900 \end{bmatrix}^T$$

$$L_f = \begin{bmatrix} 0.9901 & -0.2175 & 0.2883 & 0.1374 & 0.1906 \end{bmatrix}$$

A Bode plot of the filter response is given in figure 3.2. It can be seen that the filter is essentially low-pass, with a cut-off frequency (at -3 dB) of approximately 4 rad/s.

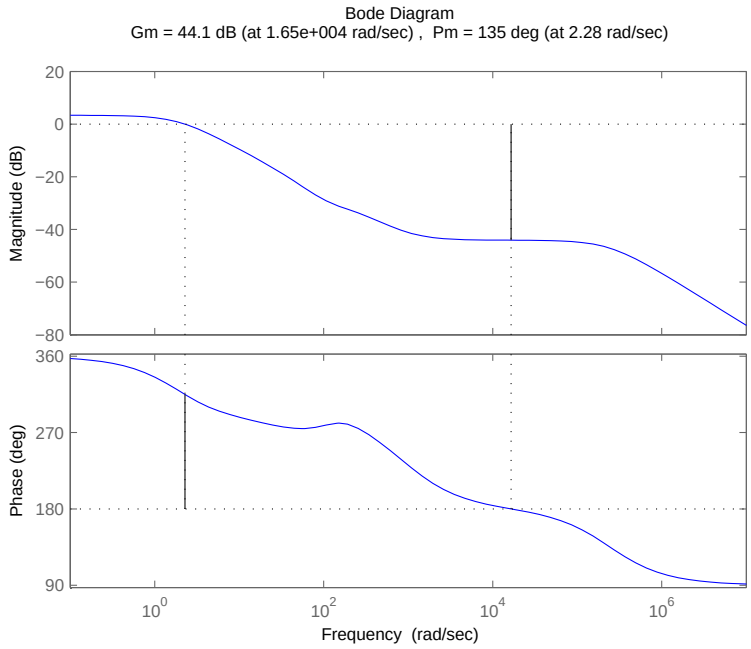


Figure 3.2: Bode plot of the filter.

Robust Stability & Performance Results for the L_2 performance analysis using Theorem 2 are given in Table 3.2. The output for performance analysis is $e := \eta - \eta_c$. We minimise γ, for a given value of σ^2. As with the filter synthesis, this is a linear objective with LMI constraints and is solved using the MATLAB toolbox [143]. Our goal is to minimise σ^2 whilst still being able to find a finite L_2 gain (much like the approach in [147] for robust stability analysis). We find that the LMIs are not feasible with $\sigma^2 = 1$. The smallest value for which we can obtain a result is $\sigma^2 = 1.42$, which implies a robust L_2 gain, not over all allowed values of $\theta(t)$, but for $M \in [1.62, 2.88]$, $|\alpha| \in [1.61, 18.4°]$ and d_1, d_2 represent $\pm 21\%$ on the α and u dependent parts of C_m, respectively. We obtain $\gamma = 198$, which of course does not guarantee good robust performance in tracking η_c. By increasing σ^2 we can obtain better values of γ, at the cost of further reducing the region over which we can give a robustness guarantee. We note that increasing σ^2 beyond about 10 does not reduce γ significantly.

Note that although we have found finite L_2 gains for restricted regions of the model envelope in tabel 3.2, these only imply robust *stability* of the closed-loop

Table 3.2: Robust Stability

σ^2	$\|\Delta\|$	γ
1	1	infeasible
1.42	0.84	198
2	0.71	4.57
10	0.32	1.77
100	0.1	1.47
1000	0.032	1.42

system with the filter. These L_2 gains do not imply satisfactory *performance* of the closed loop system. Checking robust performance really requires extensive nonlinear simulations, which is outside the scope of this chapter. However, some nonlinear simulations with time-varying uncertainty are given in the next section.

Nonlinear simulation The simulation results are given in figs. 3.3 to 3.6 for a series of constant step demands in η_c. This is the same series of steps carried out in [121] and [39].

There are four simulations shown: case 1 is the nominal model without filter; cases 2 to 4 include the filter. All simulations were done at $M = 3$ (constant) and included the actuator.

Case 2 is the nominal model with the filter (no uncertainty on C_m, i.e. $d_1 = d_2 = 1$).

Cases 3 and 4 include independent time-varying uncertainties $d_1(t)$ and $d_2(t)$.
For case 3, $d_1 = 1 + 0.25\sin(2\pi t/4.5 + 0.1)$, $d_2 = 1 + 0.25\sin(2\pi t/4.5 + 0.2)$.
For case 4, $d_1 = 1 + 0.25\sin(2\pi t/4.5 + 0.3)$, $d_2 = 1 + 0.25\sin(2\pi t/9 - 0.8)$.
These were chosen such that for case 3 the initial values of the uncertainties are small, whereas for case 4 the initial values are larger. It can be seen that for case 4, the performance for the initial $30g$ step is significantly poorer.

The approximation made in the slow subsystem control synthesis produces a

steady-state error, which is quite noticeable for the larger step commands. This error essentially comes from the fact that the slow subsystem command for q_c is not the correct value in order to achieve $\dot{e}_\eta = -ke_\eta$. This means that there is an equilibrium $\dot{e}_\eta = 0$ when $e_\eta \neq 0$. This error is present even for the nominal case and is given by (for constant Mach):

$$e_{\eta,ss} = -\frac{K_1 K_3 M^3 z_0}{k_1 m_0} \frac{\partial C_z}{\partial \alpha}(|\alpha_{ss}|, M)\tilde{C}_m(\alpha_{ss}, M)$$

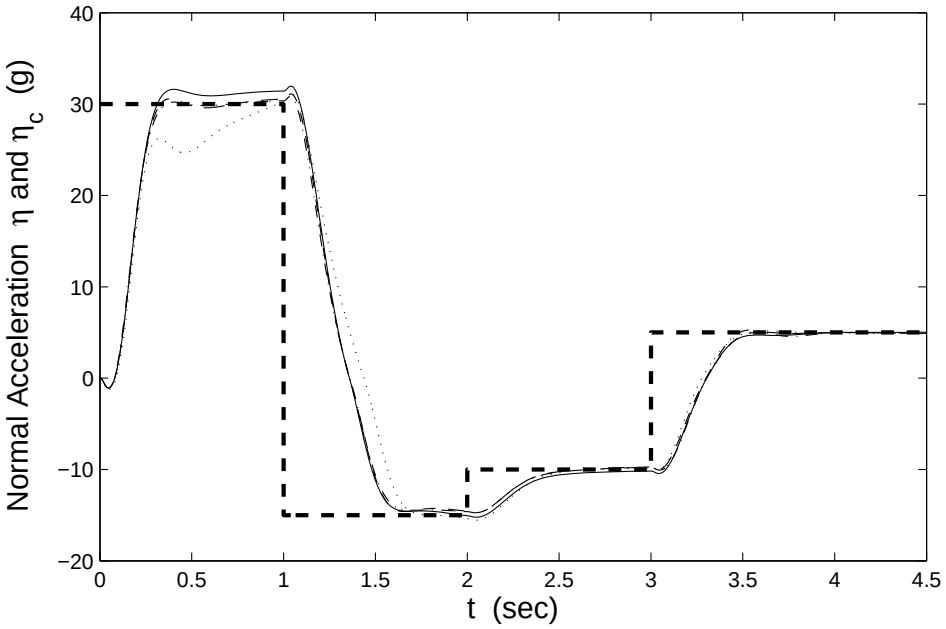


Figure 3.3: Normal acceleration η for case 1 (solid line), case 2 ('- -'), case 3 ('-.') and case 4 ('..'), for the series of step commands (thick dashed line).

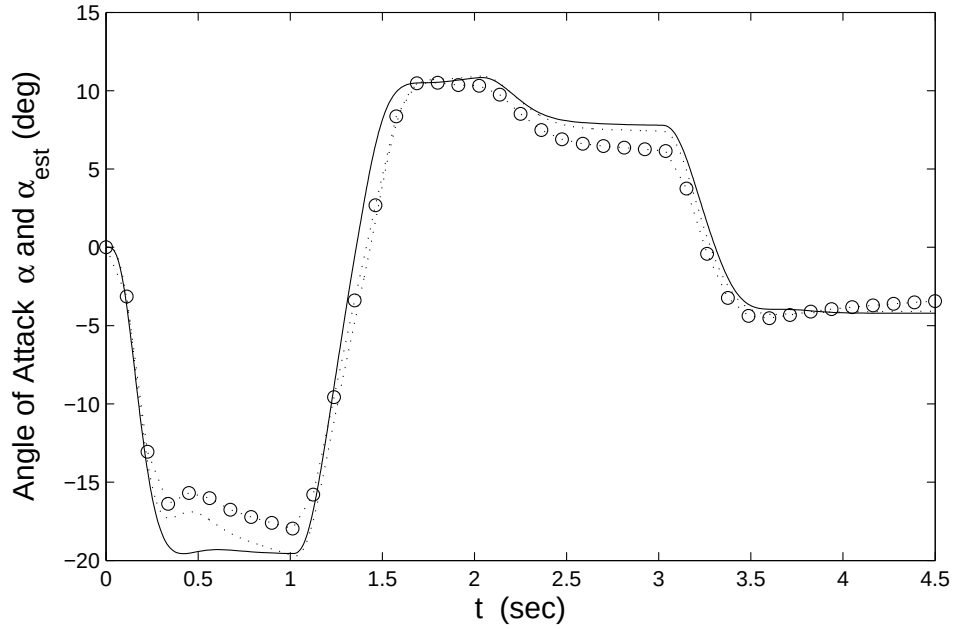


Figure 3.4: Angle of attack for case 1 α (solid line), case 4 $\alpha, \hat{\alpha}$ ('..', '.o.').

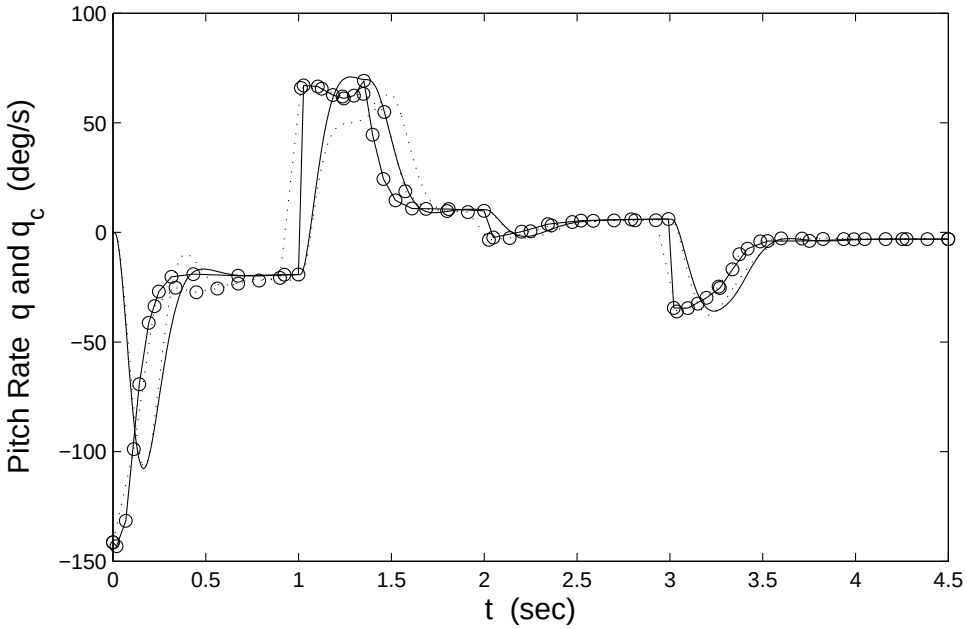


Figure 3.5: Pitch rate q and q_c for case 1 (solid line, '-o-') and for case 2 ('..','.o.').

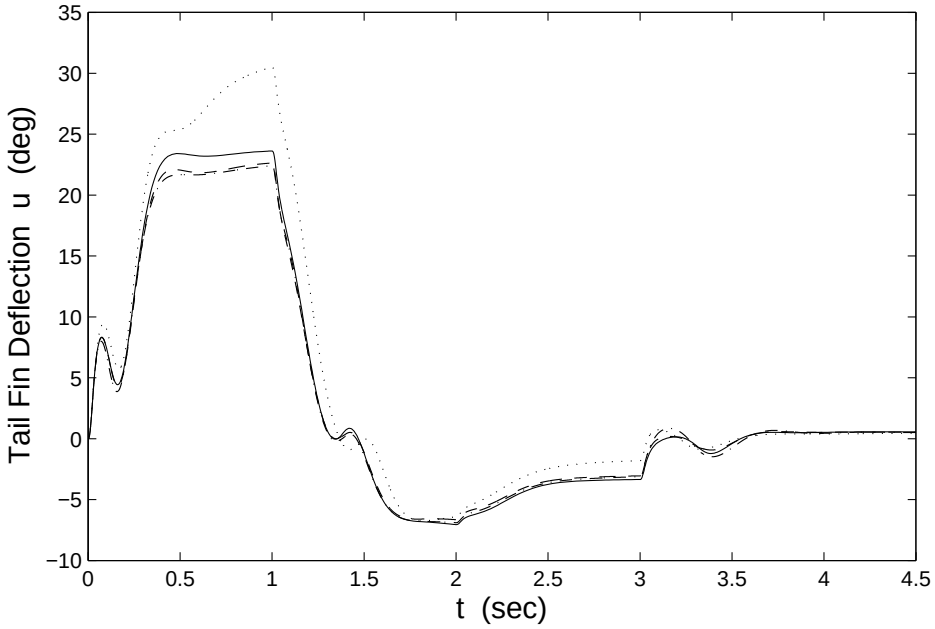


Figure 3.6: Tail fin deflection for case 1 (solid line), case 2 ('- -'), case 3 ('-.') and case 4 ('..').

3.6 Concluding Remarks

This chapter has developed an existing filter synthesis method in the literature to the case of H_∞ synthesis, where the external input signal is not assumed to be white noise. As the synthesis is based on a conservative LMI, we have also introduced an extra variable to decrease the size of the norm bound on the Δ-block, which allows us to get a result at the cost of reducing the size of the uncertainty for which the filter is designed.

The benefit of the methods presented here is that they require similar LFT system representations for the robust filter synthesis and for the robust performance analysis. We have shown that these methods can be applied successfully to a nonlinear, uncertain system with a simple feedback linearisation controller. Also, we have found that the filter performs well in all simulations, even in the presence of time-varying uncertainty.

We have used a conservative LMI technique for robust stability analysis of the

resulting closed-loop rational system, in order to show that the closed loop is robustly stable in the presence of time-varying uncertain parameters.

By using the filter to estimate the unknown states that are not available for feedback, together with robust stability analysis of the closed loop, we have been able to build on previous work in the literature for robust stability analysis for FL control, without relying purely on simulation to prove stability.

The stability analysis is conservative in that it searches for a quadratic Lyapunov function for the nonlinear system, and we have found that we cannot cover the entire flight envelope with a single Lyapunov function.

Although we have a rigorous method for robust stability analysis with output feedback and time-varying uncertainty, we are not able to guarantee good robust performance of the closed loop just by finding an L_2 gain from external input to tracking error. This method also does not guarantee that the actuator position and rate limits are obeyed. Hence, nonlinear simulations are still useful for assessing performance.

Also, the stability analysis presented here has required the assumption that the states in which the plant is nonlinear are bounded. This is an issue which we will address in chapter 7 by finding a robust domain of attraction for the system within which we can guarantee that the bound on the states holds.

Table 3.3: Physical Data

P_0	46602 Pa	(static pressure at 20,000 ft)
v	315.9 m/s	(speed of sound at 20,000 ft)
S	0.04 m^2	(surface area)
m	204.02 kg	(mass)
d	0.23 m	(diameter)
I_y	247.44 kg/m^2	(pitch moment of inertia)
K_1	$0.7P_0S/mv$	
K_2	$0.7P_0Sd/I_y$	
K_3	$0.7P_0S/m$	
z_3	19.3470 rad^{-3}	
z_2	-31.0084 rad^{-2}	
z_1	-9.7174 rad^{-1}	
z_0	-1.9481 rad^{-1}	
m_3	40.4847 rad^{-3}	
m_2	-64.1657 rad^{-2}	
m_1	2.9221 rad^{-1}	
m_0	-11.8029 rad^{-1}	
ξ_a	0.7	(actuator damping ratio)
ω_a	150 rad/s	(actuator undamped natural frequency)

Chapter 4

Introduction to ADMIRE

The Aero-Data Model In a Research Environment (ADMIRE) [2] is a Simulink model of a small single-engined fighter aircraft, developed in collaboration between researchers in industry and academia. The model is based on the Generic Aerodata Model (GAM) developed by Saab [1]. The goal of the development of ADMIRE was to have a model of a military aircraft that would be available to researchers and for use in simulators.

ADMIRE builds on aerodynamic model GAM by including an engine model, sensors, and actuators (as LTI blocks) together with a basic flight control system (FCS) and delay between the FCS and actuators. The model also includes rate and position limits on the actuators. Another aspect of ADMIRE is to include errors on the aerodynamic force and moment derivatives and sensor measurements available to the control system, in order to investigate robustness of controllers proposed as part of the GARTEUR Flight Clearance project FM (AG-11).

The bundled FCS is based on linearisation of the model around discrete trim points in the flight envelope and pole-placement, and is scheduled by Mach and altitude. This provides basic longitudinal and lateral stability and speed control. Longitudinal control is pitch rate, for Mach below 0.5, and normal load factor for higher Mach. Lateral control is roll command around the velocity vector and angle of sidelsip (for more details, see [2]).

ADMIRE is implemented in Simulink with embedded C-files to evaluate state

derivatives for numerical integration. The numerical simulation uses discrete time steps, so slightly different simulation results will be obtained by using steps of a different size. The C-files also direct the algorithm to look-up tables to calculate the aerodynamic force and moment coefficients, dynamic pressure, and thrust from the engine.

ADMIRE was used as a benchmark in [148], to investigate the use of more complex nonlinear control techniques and analysis, and a more detailed model description and design challenge can be found there. Techniques used were LPV control in [124], block-backstepping control in [149], evolutionary algorithms for clearance of flight control laws in [150], and qualitative (simulation-based) analysis of an NDI control law in [151].

In this chapter we give a description of the short-period dynamics of ADMIRE. This will be used in chapter 5 to develop a longitudinal LPV model and dynamic inversion controller and in chapter 7 for a more complex FL-TSS control of all three axes. The latter will be based on a nonlinear polynomial approximation to ADMIRE's dynamics which we build in chapter 6.

Physical data for the aircraft are given in table 4.1. Notation used for state and control variables is given in table 4.2. Notation used for other outputs and auxiliary variables used in calculations are given in table 4.3.

4.1 Definitions of symbols used in ADMIRE

Table 4.1: List of aircraft data and physical constants

Name	Notation	Value	Units
Accn. due to gravity	g	9.81	m s^{-2}
Aircraft data:			
Wing area	S_{ref}	45	m^2
Wing span	b_{ref}	10	m
Wing chord (mean)	c_{ref}	5.2	m^2
Mass	m	9100	kg
Centre of gravity position	x_{cg}, y_{cg}, z_{cg}	0, 0, -0.15	m
Aircraft body axis moments and products of inertia:			
	I_x	21000	kg m^2
	I_y	81000	kg m^2
	I_z	101000	kg m^2
	I_{xz}	2500	kg m^2
Auxiliary constants:			
	Γ	$I_xI_z - I_{xz}^2$	kg^2 m^4
	C_1	$((I_y - I_z)I_z - I_{xz}^2)/\Gamma$	-
	C_2	$(I_x - I_y + I_z)I_{xz}/\Gamma$	-
	C_3	I_z/Γ	$kg^{-1}m^{-2}$
	C_4	I_{xz}/Γ	$kg^{-1}m^{-2}$
	C_5	$(I_z - I_x)/I_y$	-
	C_6	I_{xz}/I_y	-
	C_7	$1/I_y$	$kg^{-1}m^{-2}$
	C_8	$((I_x - I_y)I_x - I_{xz}^2)/\Gamma$	-
	C_9	I_x/Γ	$kg^{-1}m^{-2}$

Table 4.2: List of state and control variables. All angles (rates) are in rad (rad/s) unless otherwise stated.

Symbol	Description
Short-period states *see figs.7.2,4.2*:	
α	Angle of attack
β	Angle of sideslip
p_b	Body-fixed roll rate
q_b	Body-fixed pitch rate
r_b	Body-fixed yaw rate
Long-period states:	
V_T	Total velocity (m/s)
ψ	Heading angle
θ	Pitch angle
ϕ	Bank angle
x_v, y_v, z_v	Positions in vehicle-carried ref. frame (m)
Controls *see fig.4.3* :	
δ_{rc}	Right canard deflection
δ_{lc}	Left canard deflection
δ_{roe}	Right outer elevon deflection
δ_{rie}	Right inner elevon deflection
δ_{lie}	Left inner elevon deflection
δ_{loe}	Left outer elevon deflection
δ_r	Rudder deflection

Table 4.3: List of auxiliary and aerodynamic variables.

Symbol	Description
Auxiliary variables or outputs:	
u_b, v_b, w_b	Body-fixed velocities (m/s)
n_x, n_y, n_z	Load factors (g)
F_x, F_y, F_z	Forces (N)
M_x, M_y, M_z	Moments (Nm)
Aerodynamic variables:	
M	Mach number
h	Altitude (m)
a	Speed of sound (m/s)
$\bar{q}$	Dynamic pressure (N/m^2)
ρ	Density of air (kg/m^3)
C_T, C_N, C_C	Tangential, normal, side force coefficients *see fig.4.1*
C_l, C_m, C_n	Rolling, pitching, yawing moment coefficients *see fig.4.1*

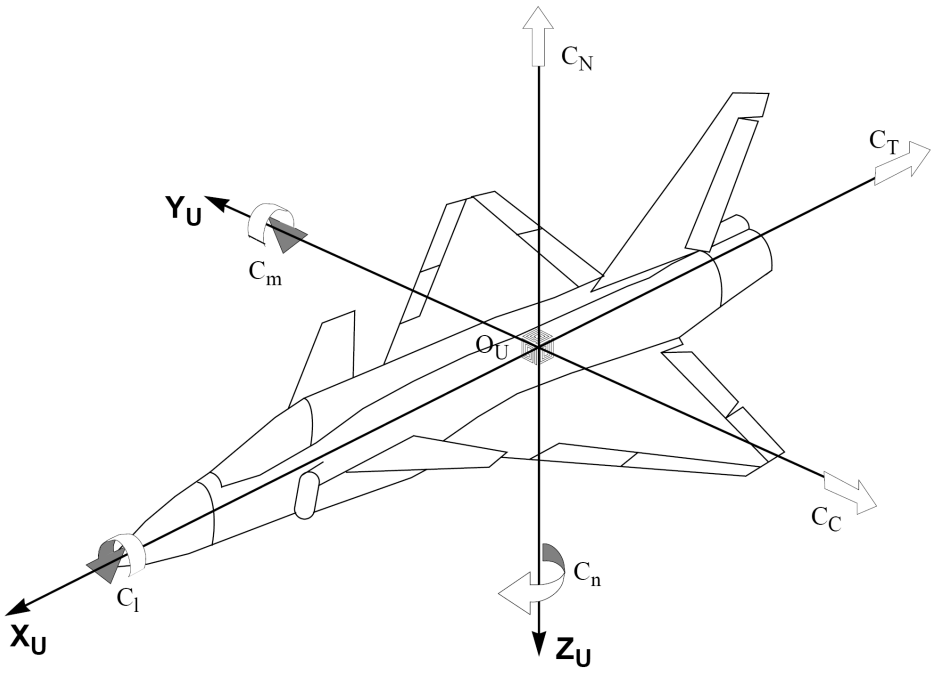


Figure 4.1: Frame S_U to define aerodata [1]

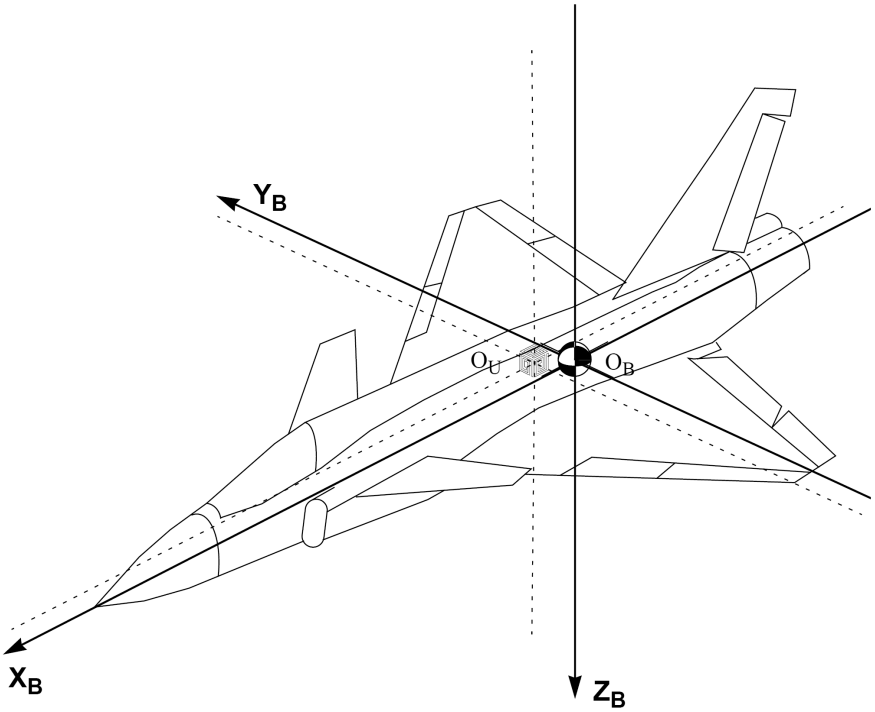


Figure 4.2: Body fixed frame S_B, including reference point for aerodata [1]

Figures 4.1 and 4.2 show the S_U and S_B frames respectively. S_U is the aerodynamic reference frame of the aircraft, which is fixed. The aerodynamic force and moment coefficients are calculated with respect to this reference frame. S_B is the 'body-fixed' reference frame, with respect to which the total forces and moments on the aircraft are calculated. In the nominal case ($x_{cg} = y_{cg} = z_{cg} = 0$), these two reference frames have the same origin, however in general the aircraft's centre of gravity can change so that these do not coincide, which produces extra turning effects in the moment equations (4.3).

The controls available in ADMIRE are left and right canard and inner and outer elevon deflections, and rudder see fig.4.3. In addition, ADMIRE can include throttle setting, leading edge flap, thrust vectoring and landing gear effects. The throttle setting is used as part of the bundled FCS for speed control, however the leading edge, thrust vectoring and landing gear are not used.

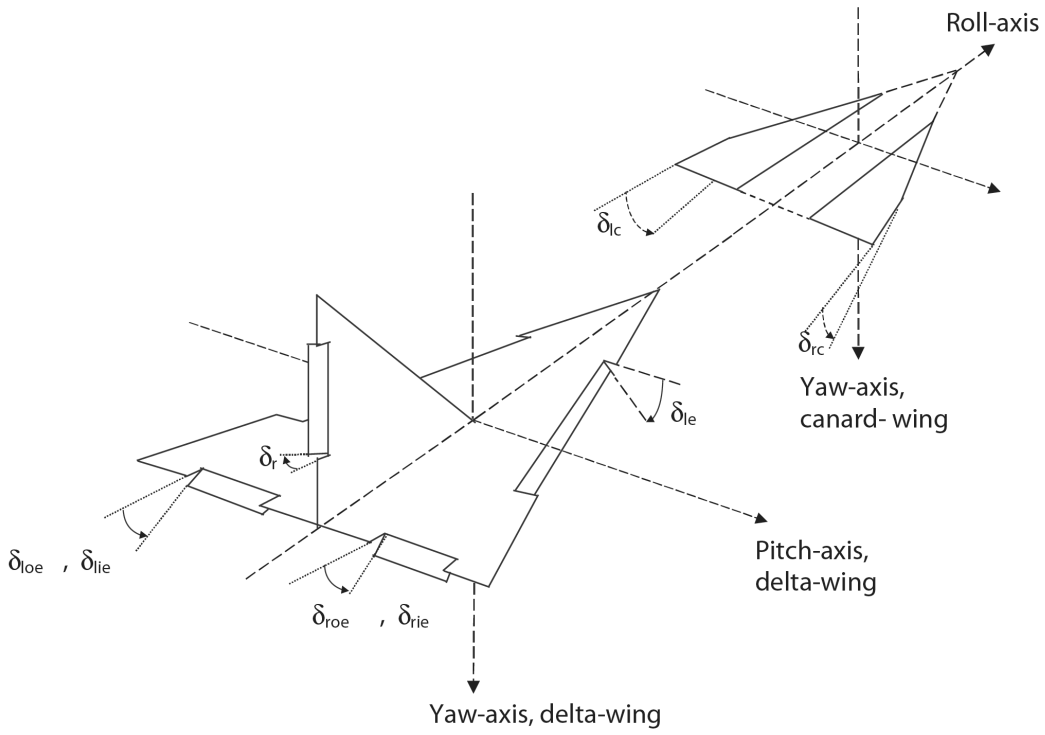


Figure 4.3: Definition of the control surface deflections. [2]

4.2 Short-period model description

In the sequel, we consider only the short-period dynamics of ADMIRE, a desription for which we give here. A more complete description including the phugoid mode can be found in [2] and [148].

Short-period 3-axes state equations:

$$\dot{\alpha} = (u_b\,\dot{w}_b - w_b\,\dot{u}_b)/(u_b^2 + w_b^2) \tag{4.1a}$$

$$\dot{\beta} = (\dot{v}_b\,V_T - v_b\,\dot{V}_T)/(V_T^2\,\cos\beta) \tag{4.1b}$$

$$\dot{p}_b = (C_1\,r_b + C_2\,p_b)q_b + C_3\,M_x + C_4\,M_z \tag{4.1c}$$

$$\dot{q}_b = C_5\,p_b\,r_b - C_6(p_b^2 - r_b^2) + C_7\,M_y \tag{4.1d}$$

$$\dot{r}_b = (C_8\,p_b - C_2\,r_b)q_b + C_4\,M_x + C_9\,M_z \tag{4.1e}$$

The auxiliary velocity and acceleration equations are given by:

$$u_b = V_T\,\cos\alpha\,\cos\beta \tag{4.2a}$$

$$v_b = V_T\,\sin\beta \tag{4.2b}$$

$$w_b = V_T\,\sin\alpha\,\cos\beta \tag{4.2c}$$

$$\dot{u}_b = r_b\,v_b - q_b\,w_b - g\,\sin\theta + F_x/m \tag{4.2d}$$

$$\dot{v}_b = -r_b\,u_b + p_b\,w_b + g\,\sin\phi\,\cos\theta + F_y/m \tag{4.2e}$$

$$\dot{w}_b = q_b\,u_b - p_b\,v_b + g\,\cos\phi\,\cos\theta + F_z/m \tag{4.2f}$$

$$\dot{V}_T = (u_b\,\dot{u}_b + v_b\,\dot{v}_b + w_b\,\dot{w}_b)/V_T \tag{4.2g}$$

The aerodynamic forces and moments in (4.1) and (4.2) are defined as:

$$F_x = \bar{q}\,S_{ref}\,C_T \tag{4.3a}$$

$$F_y = \bar{q}\,S_{ref}\,C_C \tag{4.3b}$$

$$F_z = -\bar{q}\,S_{ref}\,C_N \tag{4.3c}$$

$$M_x = \bar{q}\,S_{ref}\,b_{ref}\,C_l - z_{cg}\,F_y + y_{cg}\,F_z \tag{4.3d}$$

$$M_y = \bar{q}\,S_{ref}\,c_{ref}\,C_m - x_{cg}\,F_z + z_{cg}\,F_x \tag{4.3e}$$

$$M_z = \bar{q}\,S_{ref}\,b_{ref}\,C_n - y_{cg}\,F_x + x_{cg}\,F_y \tag{4.3f}$$

The control vector for the short-period model consists of a truncated set of the controls available in ADMIRE; canards, inner and outer elevons and rudder:

$$u_{adm} = \begin{bmatrix} \delta_{rc} & \delta_{lc} & \delta_{roe} & \delta_{rie} & \delta_{lie} & \delta_{loe} & \delta_r \end{bmatrix}^T \tag{4.4}$$

Note that the controls do not appear explicitly in the state equations (4.1), but enter via the aerodynamic force C_T, C_N, C_C and moment C_l, C_m, C_n coefficients in (4.3), which are calculated using look-up tables of aerodynamic data from [1]. The forces F_x etc affect the angular rates $\dot{\alpha}$ and $\dot{\beta}$, whereas the moments affect the body-axis angular accelerations $\dot{p}_b$ etc (with additional force effects in (4.3) due to the difference between the aerodynamic centre and centre of gravity).

In chapter 5 we restrict attention to a simplified, short-period longitudinal LPV model of ADMIRE which is SISO. Correspondingly, the controls used are only the elevons δ_{roe}, δ_{rie}, δ_{lie}, δ_{loe}, which are all slaved together to give a single control input which is identical symmetrical deflection of all elevons.

In chapters 6 and 7 we consider a short period model of all three axes which is MIMO. As such we use *all* of the control surfaces in (4.4). However the canards δ_{rc} and δ_{lc} are slaved together symmetrically, so the length of the control vector is effectively reduced to 6.

In general, there are additional thrust effects in (4.3), but we will neglect these in the description of the short-period model.

Descriptions of the sensors and actuators and the model envelope can be found in [2] and [148].

Chapter 5

LPV modelling and robust DI

Work in this chapter has been published in the conference proceedings of SICE 2013 [152].

5.1 Introduction

Dynamic inversion (DI) is a popular method of control synthesis in aerospace systems. This method of synthesis can have an advantage over linear parameter varying (LPV) (for example [153], [154], [124]) or gain scheduling [155] techniques, as the latter can be computationally demanding.

As discussed in chapter 2, it is generally assumed that all of the states are available to the controller, see for example [148], [34]. Robust stability of the closed-loop system to parametric uncertainties is generally verified by simulation or stochastic analysis, such as [54] and design [108], a drawback of which is the possibility of missing some worst case scenario.

Advanced techniques for robust stability analysis of nominally stable nonlinear and uncertain systems have been available for some time, for example linear differential inclusions (LDIs) with scalings [86], [87] and integral quadratic constraints (IQCs) [156]. In [39], robust stability analysis of a missile controlled by feedback linearisation and time scale separation is carried out by IQCs. The recent sum-of-squares technique has been used in [157] for robust stability analysis of aircraft

pitch axis with a dynamic inversion-based control law, with the assumption that the uncertain parameter is constant. The LDI analysis considered in this chapter does not set a bound on the rate of variation of the uncertain parameters.

In practice, it is generally the case that only a subset of the states are available for feedback or that only sensed values are available. The approach presented here is to design a DI controller for the nominal LPV plant. We derive a linear fractional transformation [126] (LFT) representation of the zero-dynamics, based on an LFT of the plant. We then perform a linear matrix inequality (LMI) stability analysis, based on existing scaled LDI theory [87]. The model considered is an LPV representation of the short-period ADMIRE [2] longitudinal dynamics.

When sensors are included in the loop, the controller does not have access to the actual plant states. An actuator is also included in the loop. Hence, the order of the system is in fact higher than the DI-LPV controller has been designed for. Time-varying uncertainties on the polynomial surface fits for the LPV model is also included in the analysis. The final step is to check the robust stability of the full closed-loop system by checking that there is a finite L_2 gain from a reference input to a performance output. The closed-loop system with controller, sensors, actuator and uncertainties is represented as an LFT, which leads naturally to the LDI analysis.

This chapter is organised as follows: In Section 5.2 we present an LPV model for the longitudinal short-period dynamics of ADMIRE and synthesis of a DI-LPV controller. We also move from an LPV to an LFT model, which is more suitable for stability analysis. In Section 5.3, we derive an LFT form for the zero-dynamics, based on the plant LFT and give an LMI condition for stability of the zero-dynamics. In Section 5.4 nonlinear and LPV simulation results are presented. We also perform LMI-based robust stability analysis for the closed-loop system, by finding an upper-bound on the L_2 gain. Robust stability guarantees are found for the closed-loop system with the controller, even when sensors, actuators and time-varying parametric uncertainties are included in the analysis. Conclusions are given in Section 5.5.

Table 5.1: Notation - chapter 5

Symbol	Meaning
α	angle of attack
q	pitch rate
u	control input (symmetrical elevon deflection)
M, h	Mach, altitude
y	output to control (pitch rate)
$\mathbf{A}, \mathbf{B}, \mathbf{C}, \mathbf{D}$	plant matrices (parameter-varying)
$\boldsymbol{\theta}$	vector of time-varying parameters (M and h)
$\tilde{x}$	truncated plant state vector
u_0	control signal at zero output
$\mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$	LFT matrices of the system with zero-dynamics
$\boldsymbol{\theta}_{ext}$	an extended parameter vector including TV uncertainty
r	reference signal
x_m	measured (sensed) state vector
x_s	sensor states
e_m, ϵ_m	measured value of tracking error, integral error
u_{com}	control command (input to actuator)

5.2 ADMIRE LPV Model and Controller

The Aero-Data Model in Research Environment (ADMIRE) [2] is a highly nonlinear model of a high-performance aircraft. As well as the aerodynamic data, in the form of look-up tables, the model includes sensors, actuators, rate-limiters and delays. Here we consider the short-period longitudinal dynamics, in the form of an LPV model. The ADMIRE is trimmed and linearised at $N_{data} = 441$ points, gridded to cover the flight envelope in terms of Mach, M, and altitiude, h:

$0.3 \leq M \leq 1.2$, in 21 steps of 0.045 and $100m \leq h \leq 6000m$, in 21 steps of

$295m$.

This results in an LTI state space model for each grid point. It is then proposed that an LPV model can be constructed for the short-period dynamics, covering the flight envelope, such that the state space matrices depend on the parameters M and h.

We write each element of the state space matrices as a polynomial in these two variables, by finding a best fit 2-variable polynomial to the 441 grid points. We note that in the transonic region, there is an abrupt change in several of the aerodynamic coefficients, which in some cases is not well-represented by the surface fit. The maximum order polynomial fit we use is of order 3 in M and order 1 in h. We therefore expect some difference between the ADMIRE simulation and the LPV simulation. An example of such a surface fit is given in fig.5.1, which corresponds with the fits found in [154]. We write this in the form:

$$\dot{x}(t) = \mathbf{A}(\boldsymbol{\theta}(t))x(t) + \mathbf{B}(\boldsymbol{\theta}(t))u(t) \tag{5.1a}$$

$$y(t) = \mathbf{C}x(t) \tag{5.1b}$$

where $x \in \mathbb{R}^n$, $u \in \mathbb{R}$ is the control signal and $y \in \mathbb{R}$ is the output to control. In the case of the short-period longitudinal dynamics of ADMIRE, we take $x = \begin{bmatrix} \alpha & q \end{bmatrix}^T$, where α and q are Angle of Attack and pitch rate and u is the average of all four elevons. The parameter-varying system matrices are given by:

$$\mathbf{A}(\boldsymbol{\theta}(t)) = \begin{bmatrix} Z_\alpha(\boldsymbol{\theta}(t)) & Z_q(\boldsymbol{\theta}(t)) \\ M_\alpha(\boldsymbol{\theta}(t)) & M_q(\boldsymbol{\theta}(t)) \end{bmatrix} \tag{5.2a}$$

$$\mathbf{B}(\boldsymbol{\theta}(t)) = \begin{bmatrix} Z_u(\boldsymbol{\theta}(t)) \\ M_u(\boldsymbol{\theta}(t)) \end{bmatrix}, \quad \boldsymbol{\theta}(t) = \begin{bmatrix} M(t) & h(t) \end{bmatrix}^T \tag{5.2b}$$

Similarly, the normal load factor n_z can be given in LPV form (as this is not used in the controller, we do not give it here - it is only used for checking the size of the load factor in simulations). We will take the control output to be the pitch rate q, hence $\mathbf{C} = \begin{bmatrix} 0 & 1 \end{bmatrix}$. The envelope for ADMIRE is given by the bounds on M and h given previously and: $-3g \leq n_z \leq 9g$, $-5^\circ \leq \alpha \leq 15^\circ$, $-25^\circ \leq u \leq 25^\circ$ and $-50^\circ/s \leq \dot{u} \leq 50^\circ/s$.

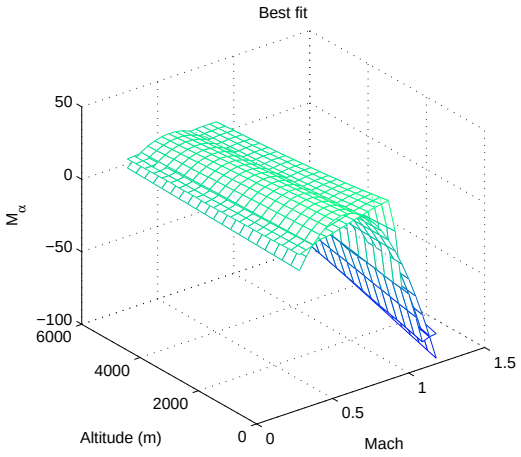


Figure 5.1: Best fit surface for $M_\alpha(M, h)$

DI-LPV Controller Here we give the controller synthesis for a system in the general form (5.1), with the assumption that the system is relative degree one with respect to the control output (5.1b), i.e. $\mathbf{CB}(\boldsymbol{\theta}(t))$ is invertible, for all $\boldsymbol{\theta}(t)$. From hereon, explicit dependence on t is dropped, for clarity of presentation. Differentiating (5.1b):

$$\dot{y} = \mathbf{CA}(\boldsymbol{\theta})x + \mathbf{CB}(\boldsymbol{\theta})u \tag{5.3}$$

Following a standard approach to design the external controller (see for example [12], [15]) so that the tracking error $e(t) := y(t) - r$ asymptotically approaches zero, we want: $v(y, r) = \dot{e} = -K_P e - K_I \int e\, dt$, with positive gains K_P and K_I and $r \in \mathbb{R}$ is a constant reference demand. If we use a control input of the form:

$$u = [\mathbf{CB}(\boldsymbol{\theta})]^{-1}[v(y, r) - \mathbf{CA}(\boldsymbol{\theta})x] \tag{5.4}$$

(where, for a SISO model, $\mathbf{CB}(\boldsymbol{\theta})$ will be a time-varying scalar) substituting (5.4) in (5.3):

$$\begin{aligned} \dot{y} &= \mathbf{CA}(\boldsymbol{\theta})x + \mathbf{CB}(\boldsymbol{\theta})[\mathbf{CB}(\boldsymbol{\theta})]^{-1}[v(y, r) - \mathbf{CA}(\boldsymbol{\theta})x] \\ &= v(y, r) \end{aligned} \tag{5.5}$$

which shows that (5.4) is an input-output linearising controller.

LFT Representation We assume the plant (5.1a) admits the LFT representation:

$$\dot{x} = \left[A + B_w \Delta(t) \left(I - D_{zw}\Delta(t)\right)^{-1} C_z\right] x \tag{5.6}$$

or:

$$\dot{x}(t) = Ax(t) + B_u u(t) + B_w w(t) \tag{5.7a}$$

$$z(t) = C_z x(t) + D_{zu} u(t) + D_{zw} w(t) \tag{5.7b}$$

$$w(t) = \Delta(t) z(t) \tag{5.7c}$$

where $z, w \in \mathbb{R}^{n_w}$ and $\Delta(t)$ is a block-diagonal matrix consisting of l diagonal blocks, one for each of the time-varying parameters. Each diagonal block consists of a repeated scalar $\delta_i(t)$, $i = 1, \ldots, l$; the normalised variation in one of the elements of $\boldsymbol{\theta}(t)$. The size of each block, k_i, depends on the polynomial degree with which that scalar appears in the system equations. $\Delta(t)$ therefore represents structured uncertainty and is norm-bounded, with allowed values in the set:

$$\begin{aligned}\boldsymbol{\Delta} := \{&\mathrm{diag}(\delta_1 I_{k_1}, \ldots, \delta_l I_{k_l}) : \\ &\|\Delta\| \leq \sigma^{-1},\ \delta_i \in \mathbb{R}\} \subset \mathbb{R}^{n_w \times n_w}\end{aligned}$$

where $\sigma > 0$. We also associate with $\boldsymbol{\Delta}$ the sets of block-diagonal scaling matrices [87]:

$$\mathbf{S} := \left\{\mathrm{diag}(S_1, \ldots, S_l) : 0 < S_i \in \mathbb{R}^{k_i \times k_i}\right\} \subset \mathbb{R}^{n_w \times n_w}$$

$$\begin{aligned}\mathbf{G} := \{&\mathrm{diag}(G_1, \ldots, G_l) : \\ &G_i = -G_i^T \in \mathbb{R}^{k_i \times k_i}\} \subset \mathbb{R}^{n_w \times n_w}\end{aligned}$$

which represent the conditions $w^T S w \leq \sigma^{-2} z^T S z$ for the bound on $\Delta(t)$ and $w^T G z - z^T G w = 0$ for the realness of $\Delta(t)$. Note that $\mathbf{S}$ and $\mathbf{G}$ consist of blocks that correspond in size to each block of $\boldsymbol{\Delta}$.

5.3 Zero-dynamics Stability Analysis

As the zero-dynamics will be time-varying, we will search for a quadratic Lyapunov function, using a scaled LDI technique. We first derive an LFT representation of the zero-dynamics, from the LFT of the plant (5.7).

The zero-dynamics of the plant are the dynamics when the output is held at zero, i.e. y, $\dot{y}$ and r are all identically 0; $\Rightarrow v(y, r) \equiv 0$ [12]. The zero-dynamics will be of order $n - 1$.

Assuming $\mathbf{C}$ is row j of I_n, one of the states is held at 0 (this is the case for our choice of output for the LPV model of ADMIRE; for a more general choice of output, the system could be transformed into normal form, then this assumption would hold [12]). We therefore truncate the state vector, removing the state which is the output. We define $\Pi \in \mathbb{R}^{(n-1)\times n}$ is I_n with row j removed. Then the truncated form of (5.7a) and (5.7b) is given by:

$$\dot{\tilde{x}}(t) = \tilde{A}\tilde{x}(t) + \tilde{B}_u u_0(t) + \tilde{B}_w w(t) \tag{5.8a}$$

$$z(t) = \tilde{C}_z\tilde{x}(t) + D_{zu}u_0(t) + D_{zw}w(t) \tag{5.8b}$$

where $\tilde{x} = \Pi x$, $\tilde{A} = \Pi A\Pi^T$, $\tilde{B}_u = \Pi B_u$, $\tilde{B}_w = \Pi B_w$ and $\tilde{C}_z = C_z\Pi^T$. $u_0(t)$ is the control signal that holds the output at 0.

In terms of the LFT of the plant (5.7), the control required to keep the output at zero is:

$$u_0 = -[\mathbf{C}B_u]^{-1}[\mathbf{C}Ax + \mathbf{C}B_w w] \tag{5.9}$$

which is just the control (5.4), with $v(y, r) = 0$, in terms of the plant LFT matrices.

We can also give the control (5.9) corresponding to the zero-dynamics in terms of the reduced-order state vector $\tilde{x}$:

$$u_0 = -[\mathbf{C}B_u]^{-1}[\mathbf{C}A\Pi^T\tilde{x} + \mathbf{C}B_w w] \tag{5.10}$$

An LFT form for the zero-dynamics are then given by substituting (5.10) into (5.8):

$$\dot{\tilde{x}}(t) = \mathcal{A}\tilde{x}(t) + \mathcal{B}w(t) \tag{5.11a}$$

$$z(t) = \mathcal{C}\tilde{x}(t) + \mathcal{D}w(t) \tag{5.11b}$$

where:

$$\mathcal{A} = \tilde{A} - \tilde{B}_u[\mathbf{C}B_u]^{-1}\mathbf{C}A\Pi^T$$

$$\mathcal{B} = \tilde{B}_w - \tilde{B}_u[\mathbf{C}B_u]^{-1}\mathbf{C}B_w$$

$$\mathcal{C} = \tilde{C}_z - D_{zu}[\mathbf{C}B_u]^{-1}\mathbf{C}A\Pi^T$$

$$\mathcal{D} = D_{zw} - D_{zu}[\mathbf{C}B_u]^{-1}\mathbf{C}B_w$$

Note (5.11) is in fact an LDI representation of the truncated form of:

$$\dot{x} = \left[I_n - \mathbf{B}(\boldsymbol{\theta})[\mathbf{CB}(\boldsymbol{\theta})]^{-1}\mathbf{C}\right]\mathbf{A}(\boldsymbol{\theta})x \tag{5.12}$$

We are now ready to state a condition for quadratic stability of the zero-dynamics (5.11):

Lemma 1. *[87] For a fixed* $\mathbb{R} \ni \sigma > 0$*, if* $\exists\ 0 < S \in \mathbf{S}$*,* $G \in \mathbf{G}$ *and* $\mathbb{R}^{(n-1)\times(n-1)} \ni P > 0$ *such that the following LMI* (5.13) *holds, then the LDI* (5.11) *is well-posed and is quadratically stable, for all* $\Delta(t) \in \mathbf{\Delta}$*.* $V(\tilde{x}) = \tilde{x}^T P \tilde{x}$ *is a Lyapunov function that proves it.*

$$\begin{bmatrix} \mathcal{A}^T P + P\mathcal{A} + \mathcal{C}^T S \mathcal{C} & P\mathcal{B} + \mathcal{C}^T S \mathcal{D} + \mathcal{C}^T G \\ \star & \mathcal{D}^T S \mathcal{D} - \sigma^2 S + \mathcal{D}^T G - G\mathcal{D} \end{bmatrix} < 0 \tag{5.13}$$

For ADMIRE, we aim to show that the zero-dynamics are quadratically stable, for all allowed Mach and altitude. Note that for the case of the short-period longitudinal dynamics of ADMIRE, the zero-dynamics are actually of order 1, because the pitch rate is kept at zero.

We normalise the LFT, such that $\sigma = 1$ corresponds to $M(t) = 0.75(1 + (3/5)\delta_M(t))$, $|\delta_M(t)| \leq 1$ and $h(t) = 3050(1 + (59/61)\delta_h(t))$, $|\delta_h(t)| \leq 1$, in order to cover the flight envelope. The Δ-block is then 8×8: 5 in δ_M and 3 in δ_h. We find that the conditions of Lemma 1 hold with $\sigma = 1$, which proves that the zero-dynamics are quadratically stable.

5.4 Robust Tracking Simulation and Analysis

Simulation We now deal with the case where there is uncertainty in the plant matrices, the controller has access only to sensed values of the states and the control signal now goes via an actuator.

By observing that the polynomial surface fits have uncertainty on them, depending on the order of the polynomials used for the fit, we estimate that there is 20%

parametric uncertainty $\delta_{M\alpha}(t)$ and $\delta_{Mq}(t)$ on each of M_α and M_q and 10% uncertainty $\delta_{Zu}(t)$ and $\delta_{Mu}(t)$ on each of Z_u and M_u. As the fits for Z_α and Z_q are much better, we neglect any uncertainty on these. These uncertainties are estimated by evaluating the surface fit for a given aerodynamic coefficient (i.e. $Z_\alpha(M,h)$, etc) at each of the N_{data} data points. At each data point, the difference between the actual value of that aerodynamic coefficient, from the linearisation of ADMIRE, and the estimated value from the surface fit, is considered as an error in the fit. The relative value of the largest error (generally in the transonic region) over the data points is taken as a reasonable estimate for a parametric uncertainty in that aerodynamic coefficient. In this case, the plant is given by:

$$\dot{x}(t) = \mathbf{A}(\boldsymbol{\theta}_{ext}(t))x(t) + \mathbf{B}(\boldsymbol{\theta}_{ext}(t))u(t) \tag{5.14}$$

where

$$\boldsymbol{\theta}_{ext} = [M(t),\, h(t),\, \delta_{M\alpha}(t),\, \delta_{Mq}(t),\, \delta_{Zu}(t),\, \delta_{Mu}(t)]^T \tag{5.15}$$

We assume that the controller has the structure of (5.4), as in the nominal case, with

$$v(y_m, r) = -K_P e_m - K_I \epsilon_m \tag{5.16}$$

where $e_m = \mathbf{C}x_m - r$ is the measured value of the tracking error, using the sensed values of the states, $x_m = [\alpha_m\ q_m]^T$ and $\epsilon_m = \int e_m\, dt$.

The two LTI sensors specified in the ADMIRE [2], are treated as a single LTI block with states $x_s \in \mathbb{R}^3$, input $x = [\alpha\ q]^T$ and output $x_m = [\alpha_m\ q_m]^T$.

The actuator (treating all four elevons as having a single actuator) is of order 1 (input u_{com}, which is the output of the controller and output u, the actual elevon deflection). The closed-loop dynamics are given in part by (5.14), with $u(t)$ being the output of the actuator.

The control command signal is given by:

$$\begin{aligned} u_{com} = -[\mathbf{CB}(\boldsymbol{\theta}_{ext})]^{-1}[\mathbf{CA}(\boldsymbol{\theta}_{ext})x_m \dots \\ \dots + K_P\mathbf{C}x_m - K_P r + K_I\epsilon_m] \end{aligned} \tag{5.17}$$

The closed-loop system is therefore of order seven: 2 plant states (α and q), 3 sensor states x_s, the actuator state u and the error state ϵ_m.

We denote the closed-loop state vector as $\bar{x} = [x^T \; x_s^T \; u \; \epsilon_m]^T \in \mathbb{R}^7$.

The interconnection and the normalisation of the time-varying parameters is performed using the Simulink interface of the LFR Toolbox [13]. Our goal is to analyse the robust stability and performance of this closed-loop, where the sensors, actuator and time-varying uncertainties have not been included in the control structure synthesis. We effectively have a control structure that is designed for the nominal plant (5.1a), but is applied to an uncertain plant (5.14), that is augmented with extra dynamics from the sensors and the actuator. This is an interesting problem, because clearly the controller will not result in exact input-output linearisation.

The next step is to find appropriate gains K_P and K_I. We perform LPV simulations of the perturbed plant (5.14) with control (5.17), including time-varying uncertainties and the extra dynamics, gridding over K_P and K_I. The integral square tracking error is evaluated over each simulation. We take the "best" gains to be those which give the smallest integral square error over most of the simulations. We arrive at $K_P = 16$ and $K_I = 40$.

Simulation results are given in fig.5.2 for the nonlinear Simulink model of ADMIRE [2] and for the LPV model, for a doublet command on q; from $+10°/s$ to $-10°/s$. The nonlinear simulations are performed first, then the observed change in $M(t)$ and $h(t)$ are approximately matched in the LPV simulation, as these are treated as independent variables in the LPV model. The nonlinear model is trimmed at the stated values of Mach and altitude. All of the bundled ADMIRE control system commands to the control surfaces are disconnected, with one exception: the command to the engine. This is necessary to keep the aircraft at a steady speed, or else it will become unstable. The controller elevon command is then sent equally to all four (left, right, inner & outer) elevons, via the existing delays, ratelimiters and actuators. The controller inputs are the sensed values of the states, Mach and altitude (using the standard sensors) and the reference r. For the LPV simulation, as well as including the sensors and actuator, we also perturb the plant with slowly time-varying parameters, of magnitude stated earlier. We note that, as the Mach number increases, so the normal acceleration can exceed the specified bound, for the

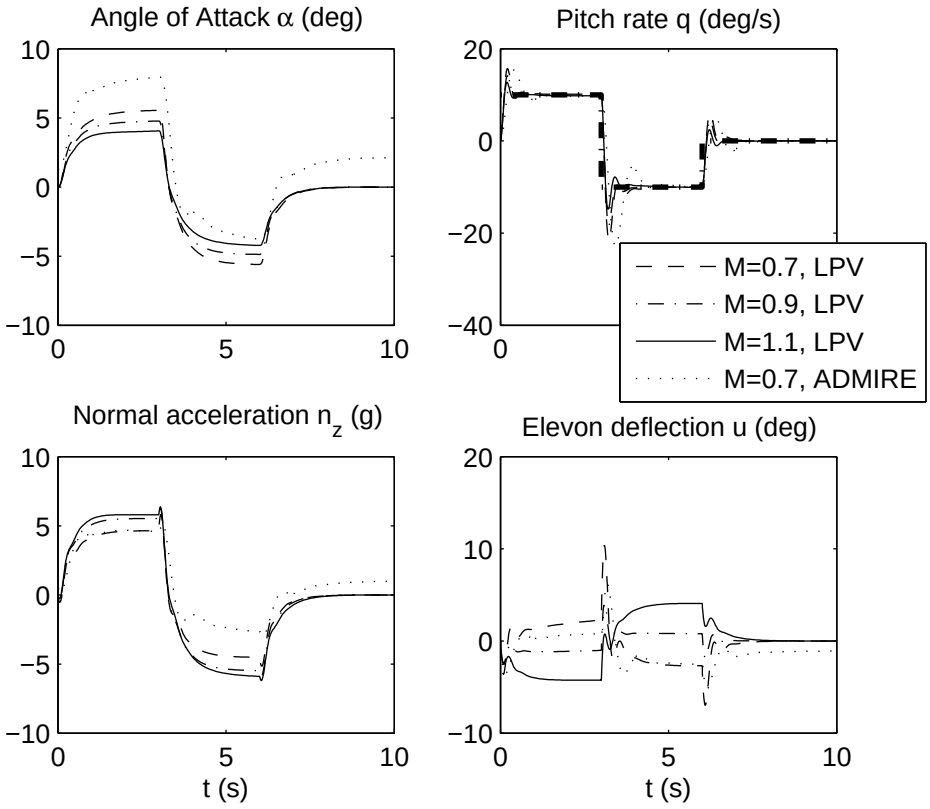


Figure 5.2: ADMIRE Nonlinear and LPV simulations, $h \approx 4$km

size of pitch rate command simulated. As noted in [2], the pitch rate is generally used as the output to control for small values of Mach. A pitch rate controller is however used as an inner loop at higher values of Mach, where the output controlled is the normal acceleration. Clearly, there is some discrepancy between the simulation of the LPV model and the ADMIRE, as we would expect not only because of the errors in the surface fits for the LPV model, but because the ADMIRE also includes rate-limiters and time delay representing computation time.

Robust Performance We aim to give a robust stability guarantee for the closed-loop LPV model, with the sensors, actuator and time-varying uncertainties, by finding a finite L_2 gain from reference input r to sensed tracking error e_m, by an LMI technique. We assume the closed-loop model admits an LFT as illustrated in fig.5.3, which is performed using [13].

We normalise the LFT and denote by σ^{-1} the bound on $\Delta(t)$ (similar to the procedure for the zero-dynamics). The closed-loop dynamics is therefore contained in the LDI:

$$\dot{\bar{x}}(t) = A\bar{x}(t) + B_r r + B_w w(t) \tag{5.18a}$$

$$e_m(t) = C_e\bar{x}(t) + D_{er} r + D_{ew} w(t) \tag{5.18b}$$

$$z(t) = C_z\bar{x}(t) + D_{zr} r + D_{zw} w(t) \tag{5.18c}$$

$$w(t) = \Delta(t) z(t) \tag{5.18d}$$

where the LFT matrices are extracted from the LFR Toolbox [13].

We have $\bar{x} \in \mathbb{R}^7$, $r \in \mathbb{R}$, $e_m \in \mathbb{R}$ and $z, w \in \mathbb{R}^{n_w}$. The subspaces $\mathbf{\Delta}$, $\mathbf{S}$ and $\mathbf{G}$ are defined analogously to Section 5.2.

Lemma 2. *[87], [144] For a given $\sigma > 0$, if $\exists\ P > 0$, $S \in \mathbf{S}$, $G \in \mathbf{G}$ and $\gamma^2 > 0$, such that* (5.19) *(below) holds, then the LFT system* (5.18) *has a finite L_2 gain from input r to output e_m, with upper bound γ, $\forall\ \Delta(t) \in \mathbf{\Delta}$. Moreover, the LDI is well-posed, i.e.* $\det\left(I - D_{zw}\Delta(t)\right) \neq 0$.

$$Z := \begin{bmatrix} Z_{11} & Z_{12} & Z_{13} \\ \star & Z_{22} & Z_{23} \\ \star & \star & Z_{33} \end{bmatrix} < 0 \tag{5.19}$$

where

$$Z_{11} = PA + A^T P + C_e^T C_e + C_z^T S C_z$$

$$Z_{12} = PB_r + C_e^T D_{er} + C_z^T S D_{zr}$$

$$Z_{13} = PB_w + C_e^T D_{ew} + C_z^T S D_{zw} + C_z^T G$$

$$Z_{22} = D_{er}^T D_{er} - \gamma^2 I + D_{zr}^T S D_{zr}$$

$$Z_{23} = D_{er}^T D_{ew} + D_{zr}^T S D_{zw} + D_{zr}^T G$$

$$Z_{33} = D_{ew}^T D_{ew} + D_{zw}^T S D_{zw} - \sigma^2 S + D_{zw}^T G - G D_{zw}$$

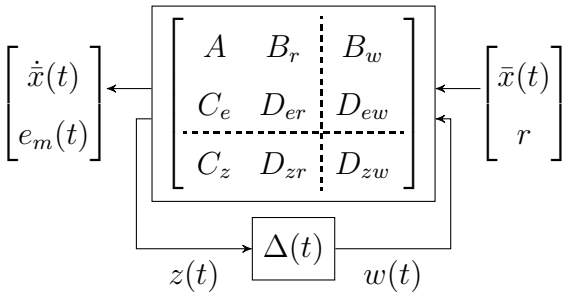


Figure 5.3: Linear fractional representation of the closed-loop

Minimising γ^2, for a given value of σ, is a linear objective with LMI constraints and is solved using the MATLAB toolbox [143].

The Δ-block is 26×26: 16 in δ_M, 6 in δ_h, 1 in each of $\delta_{M\alpha}(t)$, $\delta_{Mq}(t)$ (representing 20% uncertainty) and 1 in each of $\delta_{Zu}(t)$ and $\delta_{Mu}(t)$ (representing 10% uncertainty).

We find that we cannot find a single L_2 gain bound γ covering the entire envelope in one go. This is not a surprise, as the LDI condition allows the parameters $\delta_i(t)$ to be arbitrarily fast time-varying. Instead, we grid over small regions of Mach, in steps of 0.045 and altitude, in steps of 590m. The uncertainties on the aerodynamic coefficients are not reduced, however. For each region, the LFT is normalised with

$\sigma = 1$ and the best γ is found by Lemma 2. This gives a grid of L_2 gains over the flight envelope, in terms of Mach and altitude, which are plotted against the mid-point of each region in fig.5.4. The L_2 gain for each region is therefore an upper bound which is robust to the full $\pm 20\%$ and $\pm 10\%$ time-varying uncertainty on the relevant aerodynamic coefficients. This means we will have a robust stability

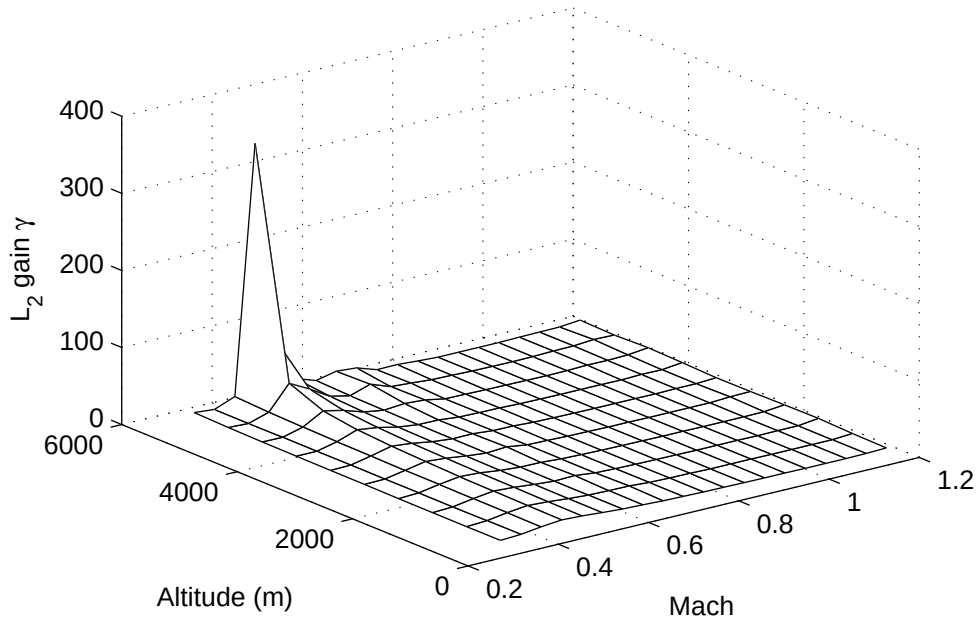


Figure 5.4: Robust upper bound on L_2 gain over the flight envelope

guarantee for the controlled plant with time-varying parametric uncertainties, with the controller having access to sensed rather than actual values of the states and the control signal entering the plant via the actuator.

We also wish to make clear that the stability analysis is performed using the full-order LPV model of the closed-loop with all 7 states, rather than a reduced order normal form. Hence we avoid the issue of having to explicitly separate the system into external and internal dynamics [12].

It is important to note that the L_2 gains found only apply to the LPV model. There is no robust stability guarantee for the actual nonlinear model of ADMIRE. That would require a quasi-LPV model, representing the actual nonlinear dynamics, perhaps using multivariable polynomial fits to the aero-data tables (we anticipate

this would lead to a very large Δ-block for the LFR).

Gridding over the flight envelope, each time normalising the LFT using [13] and then minimising γ subject to Lemma 2 took 200 regions to cover the flight envelope, which took 29 min on a 2.5 GHz i5 PC with 4 GB RAM. It is not desirable to grid over Mach and altitude, because we obtain a different Lyapunov function for each region and do not have a rigorous robust stability guarantee over the entire flight envelope. However, the steps in Mach and altitude are actually larger than the variation in those parameters over the simulations in fig.5.2. If we assume that for a given set of initial conditions, the Mach and altitude do not leave one of these regions, we have a robust stability guarantee. Of course, this depends on the stability of the phugoid mode and the manoeuvre being performed by the aircraft.

5.5 Concluding Remarks

In this chapter we have used DI-LPV control on an LPV model of ADMIRE's short-period longitudinal dynamics.

We have successfully verified that the zero-dynamics of the system are stable, subject to the time-varying parameters Mach and altitude. The derivation of the LFT of the zero-dynamics is given in a general form for the SISO case.

Despite the lack of an in-built robustness guarantee in the controller synthesis, we have shown that the controller gives good responses even with an actuator, sensors and time-varying uncertainties, which are not considered in the synthesis. This applies to both the LPV model and the nonlinear simulation. These are comparable with those in, for example, [148] or [154], where more computationally demanding synthesis techniques are employed.

We have shown that with the controlled system in LPV/LFT form, we can provide robust stability guarantees over regions of the envelope and this can be applied more generally to LPV models. However, we have only provided a robust stability analysis for the LPV model. We cannot rigorously guarantee that the actual ADMIRE will be robustly stable using this method. This analysis method

could be extended to a quasi-LPV model of a nonlinear system.

We will develop a nonlinear polynomial of ADMIRE in the following chapter, with the aim to more accurately capture the short period dynamics of all three axes. This will be used as an example in chapter 7 to give a more rigorous robust stability analysis of FL-TSS control, where we find a robust domain of attraction.

It may be possible to reduce conservatism in the closed loop analysis by using IQCs (a reduction in conservatism has been demonstrated in [39], comparing IQCs for arbitrarily fast vs. slowly time-varying parameters) or using a sum-of-squares technique.

Chapter 6

Polynomial Modelling of ADMIRE

In chapter 5, we presented an LPV model for the short-period longitudinal dynamics of ADMIRE, covering the whole flight envelope in terms of Mach and altitude. The model was based on Jacobian linearisation around the SL flight condition, at points gridded over the allowed envelope of Mach and altitude. The variation in each of the elements of the associated state-space matrices were then described approximately as polynomial functions of Mach and altitude, giving us an LPV model approximating the dynamics of ADMIRE.

As the LPV model is based on Jacobian linearisation around SL flight, it does not capture the coupling between lateral-directional and longitudinal axes, or nonlinear dynamics when the aircraft is not close to SL. We can see from equations (4.1)-(4.3) that ADMIRE is actually nonlinear and includes coupling between axes. Also, although it is not explicit in equations (4.1)-(4.3), we might reasonably expect the effectiveness of control surfaces to vary with AoA, and for coupling between axes to manifest as a change in control surface effectiveness when the aircraft is sideslipping (e.g. we would not expect symmetric elevon deflection to produce zero roll if AoS is non-zero, because control surface effectiveness would not be identical on each side - this is significant because the aircraft's moment of inertia is relatively small around the roll axis). Therefore, we should not expect the control distribution matrix to be constant, even for constant Mach and altitude. This is important because in FL design, we invert the control distribution matrix. Several other nonlinear and

coupling effects are well-studied in the literature, such as yaw instability at high angle of attack. In order to apply FL, we require a control-affine model of the plant with an explicit control distribution matrix, which we do not have in (4.1)-(4.3).

Another reason to improve on our approximate model of ADMIRE is not just for the controller design process, but in order to have a more convincing robust stability analysis of the closed loop. The LFT/LDI method used earlier requires that the system under consideration is rational, so it is not immediately applicable to ADMIRE which includes trigonometric functions and look-up tables. However if we can find a polynomial/ rational approximation to ADMIRE that captures the nonlinear dynamics and coupling between axes, it makes robust stability analysis of the closed loop more meaningful.

We focus on Mach = 0.5 and altitude = 4000 m.

6.0.1 Trimming and Linearisation

The trimming routine in ADMIRE finds values of α, δ_n and δ_e such that the equilibrium point (for 'straight and level') is as close as possible to $(x, u) = (0, 0)$. This equilibrium point 'trim' will be denoted (x_{trim}, u_{trim}). The linearisation routine then linearises around this point. The ss matrices that are output $A_{bare}, B_{bare}, \ldots$ are therefore for the system:

$$\dot{\tilde{x}} = A_{bare}\tilde{x} + B_{bare}\tilde{u} \tag{6.1}$$

$$\dot{x} - \dot{x}_{trim} = A_{bare}(x - x_{trim}) + B_{bare}(u - u_{trim}) \tag{6.2}$$

$$\dot{x} = A_{bare}(x - x_{trim}) + B_{bare}(u - u_{trim}) \tag{6.3}$$

$$\dot{x} = A_{bare}x + B_{bare}u - (A_{bare}x_{trim} + B_{bare}u_{trim}) \tag{6.4}$$

The state of the trimmed model is actually $\tilde{x} = x - x_{trim}$, so if we want to know the actual value of the plant state x then we need to add the trim value to the output:

$$y_{actual} = \tilde{y} + y_{trim} \tag{6.5}$$

$$= C\tilde{x} + Cx_{trim} \tag{6.6}$$

$$= Cx \tag{6.7}$$

with $C = I_n$ to get the actual plant state vector.

The bundled trimming and linearisation algorithms allow us to find the Jacobian linearisation of ADMIRE around the trim point for straight and level flight:

$$\dot{x} = A_{lin}\tilde{x} + B_{lin}\tilde{u} \tag{6.8}$$

where $\tilde{x} = x - x_{trim}$, $\tilde{u} = u - u_{trim}$, $x = \begin{bmatrix} \alpha & \beta & p & q & r \end{bmatrix}^T$ and $u = \begin{bmatrix} \delta_{rc} & \delta_{lc} & \delta_{roe} & \delta_{rie} & \delta_{lie} & \delta_{loe} & \delta_r \end{bmatrix}^T$.

x_{trim} is zero for all the short-period states apart from α_{trim}, which is approximately 3.4862°. The trim values of u are small and will be neglected from hereon.

We will use only symmetric deflection for the canards and use the control transformation in appendix E: $u = \begin{bmatrix} \delta_n & \delta_{ei} & \delta_{ey} & \delta_{ai} & \delta_{ay} & \delta_r \end{bmatrix}^T$.

The polynomial model here is developed by proposing a set of nonlinear monomials which are selected by looking at a second-order Taylor expansion of the short-period state equations in symbolic form, and considering the coupling between axes described in the C-files that calculate the aerodynamic coefficients in [2]. Some trial-and-error is used to remove monomials that do not give a significant contribution to each state equation.

The coefficients of each monomial are considered as variables, which are fit to open-loop simulation results of ADMIRE, using a standard Vandermonde technique and a least-squares fit. Approximately 10000 simulation data points were used. Simulations were conducted by gridding over fixed control surface deflections, in order to perturb the dynamics of ADMIRE. The C-file was edited to give the value of $\dot{x}$ for each of the short-period states at each time step, and the polynomial with variable coefficients was fit to a matrix that concatenated all of the simulation results, removing data points at which ADMIRE left the envelope. Each simulation was carried out from the trimmed flight condition for SL flight at Mach 0.5 and altitude 4000 m, hence the polynomial model given below and used in the following chapter is only valid for deviations from SL flight in this part of the flight envelope. In principle, the method could be extended to include monomials in Mach and/or altitude, however this would result in a very high-order polynomial model and be very computationally demanding.

6.0.2 Final Polynomial Model

$$
\begin{aligned}
\dot{\tilde{\alpha}} = \\
&- 1.0768\tilde{\alpha} + 0.9728q_b \\
&- 0.95208\beta p_b \\
&- 0.2264\delta_{ei} - 0.1361\delta_{ey} - 0.0126\delta_n \\
&+ 0.055501\tilde{\alpha}\delta_{ei} + 0.11199\tilde{\alpha}\delta_{ey} + 0.0089154\tilde{\alpha}\delta_n \\
&+ 0.34806\beta\delta_{ai} + 0.022782\beta\delta_{ay} + 0.03423\beta\delta_r
\end{aligned}
\tag{6.9a}
$$

$$
\begin{aligned}
\dot{\beta} = \\
&- 0.2438\beta + 0.0609p_b - 0.9893r_b \\
&+ 0.058055\tilde{\alpha}\beta + 0.95075\tilde{\alpha}p_b + 0.021665\beta q_b - 0.26768\tilde{\alpha}r_b \\
&+ 0.0383\delta_{ai} + 0.0077\delta_{ay} + 0.0666\delta_r \\
&- 0.45514\tilde{\alpha}\delta_{ai} - 0.19667\tilde{\alpha}\delta_{ay} - 0.45611\tilde{\alpha}\delta_r \\
&- 0.020724\beta\delta_{ei} + 0.0043252\beta\delta_{ey} - 0.023841\beta\delta_n
\end{aligned}
\tag{6.9b}
$$

$$
\begin{aligned}
\dot{p_b} = \\
&- 23.613\beta - 2.0784p_b + 0.5423r_b \\
&- 130.28\tilde{\alpha}\beta - 3.8229\tilde{\alpha}p_b - 13.793\beta q_b - 5.0752\tilde{\alpha}r_b + 0.10389p_b q_b \\
&- 2.2525q_b r_b \\
&+ 5.236\delta_r - 20.693\delta_{ai} - 23.706\delta_{ay} \\
&- 17.275\tilde{\alpha}\delta_{ai} - 25.901\tilde{\alpha}\delta_{ay} - 7.9192\tilde{\alpha}\delta_r \\
&- 20.065\beta\delta_{ei} - 15.582\beta\delta_{ey} - 10.674\beta\delta_n
\end{aligned}
\tag{6.9c}
$$

$$
\begin{aligned}
\dot{q_b} = \\
& 9.1771\tilde{\alpha} - 1.079q_b \\
& + 0.28991\beta p_b - 0.2791\tilde{\alpha}q_b + 0.15552\beta r_b + 1.052p_b r_b + 5.9292\tilde{\alpha}^2 - 30.32\tilde{\alpha}^3 \\
& - 0.033186{p_b}^2 - 0.074713{r_b}^2 \\
& - 8.7692\delta_{ei} - 5.3918\delta_{ey} + 5.6652\delta_n \\
& + 0.064492\tilde{\alpha}\delta_{ei} + 2.9578\tilde{\alpha}\delta_{ey} + 2.2686\tilde{\alpha}\delta_n \\
& + 0.17627\beta\delta_{ai} + 0.17224\beta\delta_{ay} + 0.25352\beta\delta_r
\end{aligned}
\tag{6.9d}
$$

$$
\begin{aligned}
\dot{r_b} = \\
& 1.7719\beta - 0.1217p_b - 0.416r_b \\
& - 6.4015\tilde{\alpha}\beta - 0.78091\tilde{\alpha}p_b - 1.5148\beta q_b + 0.37012\tilde{\alpha}r_b - 0.62512p_b q_b \\
& - 0.36707q_b r_b \\
& - 2.7575\delta_{ai} - 1.1538\delta_{ay} - 4.2585\delta_r \\
& + 0.48663\tilde{\alpha}\delta_{ai} - 1.039\tilde{\alpha}\delta_{ay} + 0.38327\tilde{\alpha}\delta_r \\
& - 1.4258\beta\delta_{ei} - 1.1976\beta\delta_{ey} + 1.5842\beta\delta_n
\end{aligned}
\tag{6.9e}
$$

Chapter 7

Robust FL-TSS Synthesis and Analysis

This chapter presents robust FL controller synthesis for the short-period dynamics of ADMIRE. Robust stability of the closed-loop system subject to parametric uncertainty on the polynomial model is analysed using the LFT matrices of the open-loop plant.

The proposed controller is based on the TSS methodology, which achieves approximate feedback linearisation. The robust stability analysis of the closed-loop does not assume that the fast subsystem states are identical to their commanded values.

Table 7.1: Notation - chapter 7. Other symbols used are standard aircraft notation as described in chapter 4.

Symbol	Meaning
x	short-period model state vector
η, Ω	slow, fast states of the short period model, $x^T = [\eta^T \ \Omega^T]$
u	control input to the plant
u_c, Ω_c	control commands, based on the idealised TSS model and FL
r	reference command for η
e_η, e_Ω	tracking error in r, Ω_c
ϵ, ε	integrated values of the above
K_{1-4}	diagonal gain matrices (design variables)
Ψ	a vector of TV uncertain parameters
$\tilde{\mathbf{A}}, \tilde{\mathbf{G}}$	uncertain QLPV plant matrices
$\bar{\Psi}$	nominal value of Ψ
$\bar{\mathbf{A}}, \bar{\mathbf{G}}$	nominal QLPV plant matrices (no uncertainty)
ξ, A_ξ, B_ξ, C_ξ	state & matrices of LTI actuator block
C_η, C_Ω	defined such that $\eta = C_\eta x$, $\Omega = C_\Omega x$
$\bar{\mathbf{A}}_{\mathbf{11}}, \bar{\mathbf{A}}_{\mathbf{12}}$	upper $n_\eta \times n_\eta$ and $n_\eta \times n_\Omega$ submatrices of $\bar{\mathbf{A}}$
$\bar{x}, \mathcal{A}, \mathcal{B}, \mathcal{C}, \mathcal{D}$	state & LFT matrices of the closed loop when $r = 0$
$\mathcal{T}$	transformation matrix
$\mathcal{S}, \mathcal{G}$	block-diagonal scaling matrices (LMI variables)
P	Lyapunov matrix (LMI variable)
σ_p^{-1}	bound on the normalised variation of each plant state
σ_w^{-1}	bound on the normalised variation of each uncertain parameter

7.1 Controller Synthesis via FL-TSS and GA

The previous chapter proposed that the short-period dynamics of ADMIRE around altitude 4000 m and Mach 0.5 can be represented approximately by the polynomial model given by equations (6.9). In this section we put this system into the standard control-affine form for FL controller synthesis. This system is then subdivided into 'fast' and 'slow' subsystems, which are approximations of the polynomial model. A nonlinear FL-TSS control structure is designed based on this model.

The controller is tuned to give acceptable performance in a number of fairly aggressive manoeuvres similar to those in [148] that excite the dynamics of all three axes. The tuning is performed using a GA that minimses a cost function (or 'fitness' rating) associated with each candidate set of controller gains. Having proposed a suitable controller, the closed-loop responses of ADMIRE, the polynomial model (6.9) and a standard JL model are compared. We also give an approximate domain of attraction for the closed-loop system, including an indication of initial conditions that cause saturation of control surfaces to occur.

7.1.1 TSS Controller Structure

For controller synthesis, we will use the FL-TSS technique

$$\dot{x}(t) = \mathbf{f}(x(t)) + \mathbf{G}(x(t))u(t) \tag{7.1}$$

where $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^{n_u}$.

A TSS representation of the plant splits the state vector into 'fast' and 'slow' parts: $x = \begin{bmatrix} \eta \\ \Omega \end{bmatrix} = \begin{bmatrix} C_\eta \\ C_\Omega \end{bmatrix} x$ where η is the slow state vector (wind axis angles) and Ω is the fast state vector (body axis angular rates). For simplicity of exposition, we assume that η consists of the first n_η elements of x and that Ω consists of the last $n_\Omega = n - n_\eta$ states:

$$\dot{\eta} = C_\eta(\mathbf{f}(x) + \mathbf{G}(x)u) = \mathbf{f_1}(x) + \mathbf{G_1}(x)u \tag{7.2a}$$

$$\dot{\Omega} = C_\Omega(\mathbf{f}(x) + \mathbf{G}(x)u) = \mathbf{f_2}(x) + \mathbf{G_2}(x)u \tag{7.2b}$$

The primary outputs to control are the slow states η. Assuming that states Ω evolve significantly faster than η, the standard approach is to neglect the effect of the control surfaces u in (7.2a) and instead treat Ω as control inputs. This defines the commanded value of Ω, which is Ω_c. Neglecting the actuator so that $u \equiv u_c$, the commanded control surface deflection u_c is defined by the fast subsystem (7.2b), in order to achieve the commanded value of Ω.

The control command for each subsystem will be designed by feedback linearisation (FL). An important assumption is that in order for the control command to each subsystem to be well-defined, we assume that not only is the plant affine in u, but that the slow subsystem (7.2a) is also affine in Ω. We also assume that the control distribution matrix is a function only of the slow states.

The idealised representation of the plant is given by:

$$\dot{\eta}_{ideal} = \mathbf{A_{11}}(\eta)\eta + \mathbf{A_{12}}(\eta)\Omega_c \tag{7.3a}$$

$$\dot{\Omega}_{ideal} = \mathbf{f_2}(x) + \mathbf{G_2}(\eta)u_c \tag{7.3b}$$

Using feedback linearisation [12], the controller design for each subsystem is:

$$\Omega_c = -\mathbf{A_{12}}(\eta)^{\dagger}(\mathbf{A_{11}}(\eta)\eta - v_\eta(\eta, r)) \tag{7.4a}$$

$$u_c = -\mathbf{G_2}(\eta)^{\dagger}(\mathbf{f_2}(x) - v_\Omega(\Omega, \Omega_c)) \tag{7.4b}$$

where the external signal $r \in \mathbb{R}^{n_\eta}$ is the reference value for η. The 'equivalent control' v for each subsystem is defined below.

Before we define the equivalent control, we should consider under what conditions the controller in (7.4) is well-defined. Both the fast and slow subsystems require a right psuedo-inverse. Assuming the states η are restricted to some domain $\eta \in \mathcal{X} \subset \mathbb{R}^{n_\eta}$, the slow subsystem inversion requires that $\mathbf{A_{12}}(\eta)^{\dagger}$ should be well-defined $\forall\, \eta \in \mathcal{X}$. A necessary condition for the right inverse to exist is that $n_\Omega \geq n_\eta$ (which is the same as $\mathbf{A_{12}}(\eta)^{-1}$ if $n_\Omega = n_\eta$). Likewise, the fast subsystem inversion requires that $\mathbf{G_2}(\eta)^{\dagger}$ should be well-defined $\forall\, \eta \in \mathcal{X}$, a necessary condition for which is that $n_u \geq n_\Omega$. Physically in an aircraft application, this implies that we cannot control more wind-axis angles (η) than we have body-angle rates (Ω). These are both of

order 3 in practice. Similarly, we cannot control more body-angle rates than we have available control surfaces (u). In practice, we would expect $n_u > n_\Omega$, particularly if inner and outer elevons are deflected independently. We will not deal with cases of control surface failure here.

In order to complete the control structure, we need to define the equivalent controls in (7.4). First we define the tracking errors $e_\eta = \eta - r$ and $e_\Omega = \Omega - \Omega_c$ and their respective integrals $\epsilon = \int e_\eta \, \mathrm{d}t$ and $\varepsilon = \int e_\Omega \, \mathrm{d}t$. In order to stabilise the error dynamics of each subsystem, the functions $v_\eta(\eta, r)$ and $v_\Omega(\Omega, \Omega_c)$ are designed as follows:

$$v_\eta(\eta, r)) = -K_3 e_\eta - K_4 \epsilon \tag{7.5a}$$

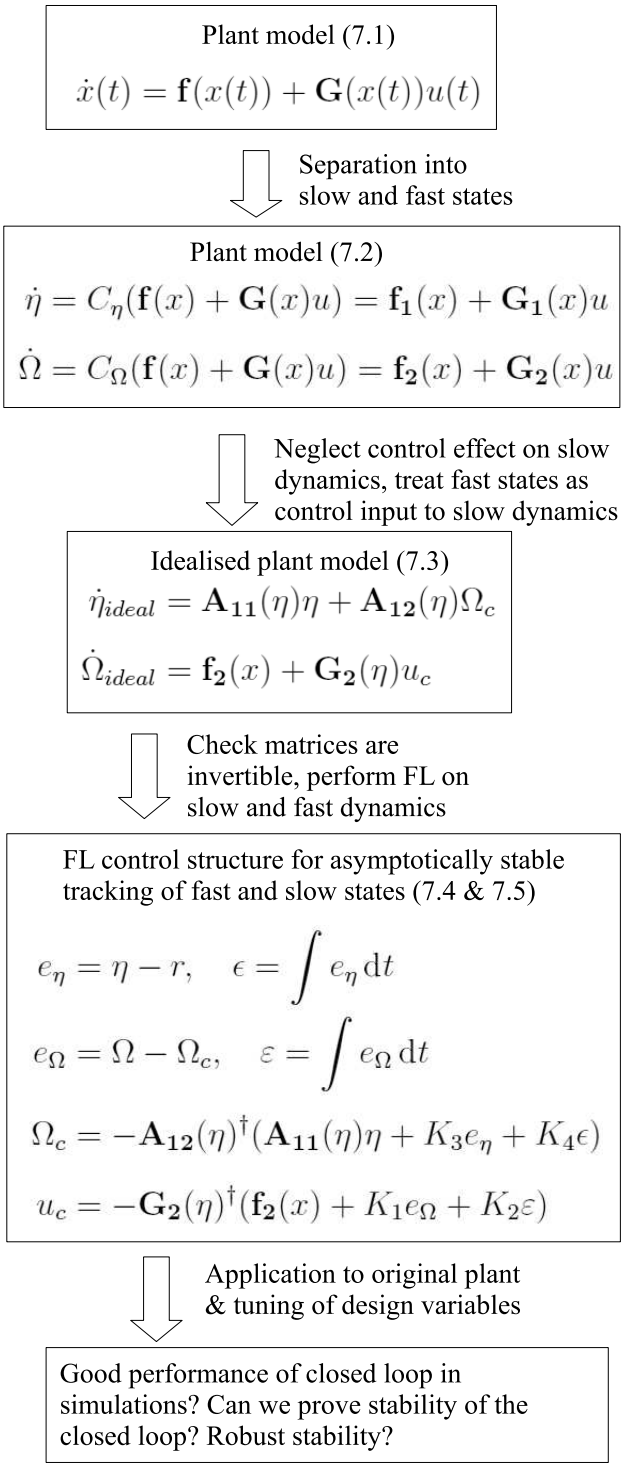
$$v_\Omega(\Omega, \Omega_c) = -K_1 e_\Omega - K_2 \varepsilon \tag{7.5b}$$

where $K_{3/4} \in \mathbb{R}^{n_\eta \times n_\eta}$ and $K_{1/2} \in \mathbb{R}^{n_\Omega \times n_\Omega}$ are diagonal matrices with positive elements on the diagonal. These are tuning variables which will affect the performance and robustness of the closed loop. Hence for this type of controller there are $2(n_\eta + n_\Omega)$ scalar variables to tune.

The controller desribed above defines approximately proportional-integral, stable error dynamics for each subsystem, when the reference commands are slowly time-varying. Not only that, but defining K_i to be diagonal decouples the error responses from one another, although this is only approximately achieved in practice due to modelling errors.

The design process is outlined in figure 7.1

Figure 7.1: Feedback Linearisation-Time Scale Separation controller design process



The preceding feedback linearisation-time scale separation (FL-TSS) controller description seems to imply that the last design step, tuning the design variables,

involves only setting the speed of the error dynamics through appropriate choice of K_i. However in reality the closed loop system may not behave as expected. The control structure is based on applying FL separately to the fast and slow dynamics, assuming that the time scales are sufficiently different that the fast states can be considered as control inputs to the slow dynamics. This is of course only an approximation in reality. Also, the TSS technique neglects the actual control effect on the slow dynamics during the design process. Not only that, but the control input to the fast dynamics is assumed to be the actual output of the controller, but we know in reality there will be actuator dynamics so that the control input to the plant is not identical to the output command of the controller. We also have to consider control surface position and rate limits and robustness to modelling errors, state measurement errors and noise, which has no simple analytic solution.

We want a closed loop system that behaves as required over some quite aggressive flying manoeuvres. A simple performance objective alone such as obtaining a small L_2 gain from some external input to tracking error would not guarantee that the aircraft actually behaves well in simulations. Not only that but a naïve approach, simply setting the control gains to give an arbitrarily fast response, on the assumption that FL is performed exactly, would not take account of actuator position or rate limits (we could expect these to saturate if the gains are set too high).

There are many simplifying assumptions made and neglected dynamics during the controller design process. Failure of any one of these assumptions or the effect of neglected dynamics could lead to very poor performance when the FL-TSS controller is applied to the plant. This motivates the use of a technique that can improve the tuning of the design variables K_i.

Tuning the controller is a highly complex and nonconvex problem. A brute-force approach such as gridding over the space of design variables, clearly would require a very large computational burden. The limits of the parameter space would have to be set in advance. Without any intuition into what the limits should be the optimal solution could lie outside these pre-defined limits and it could even be the case that

no satisfactory solution lies within those limits. This kind of approach is clearly very inefficient and time-consuming, spending a large amount of computation time evaluating performance of the controller in parts of the parameter space that are not useful. There is no memory in such an approach, i.e. no regions of the design variable space are rejected as the performance evaluations run. The computational burden involved means that even for just a few design variables, a gridding approach is not practical.

Perhaps the most well-known method for dealing with complex problems like this is the Monte-Carlo (MC) technique [158–161]. This is essentially a random walk over the space of design variables, evaluating performance at each point. This is clearly more efficient than a gridding approach because it requires fewer simulations to be run, scattering randomly over the variable space. This will not in general lead to an optimal solution, however it has been used extensively with success to give sub-optimal solutions that are satisfactory. Note that the MC method shares some of the drawbacks of gridding. The limits of the design parameter space are set in advance and there is no memory built in to such an algorithm, so again many simulations may be run exploring parts of the parameter space that could have been ruled out early on.

Genetic algorithms (GAs) [162] mimick the process of evolution by natural selection. The advantage of this approach is that simulations are run over a relatively small selection of design variables, generation-by-generation. The characteristics of the population of each generation depend on the performance of the parameters in the previous generation. A particular set design variables is more likely to become a parent to the next generation if it performs well in simulations. Conversely, poor performing sets of design variables get naturally eliminated in this process, so the algorithm does not waste as much time exploring parts of the parameter space that are not useful.

Although the next generation is derived from the previous one, random mutations can also be included for evaluation, which mean that the space of design variables is not limited by the characteristics of the initial population. There is no need to

set limits on the space of design variables in advance, hence this method can find a sub-optimal solution anywhere, if left to run for a large number of generations. Interest in GA methods has increased as it has been found to be a relatively efficient method for finding a satisfactory solution to a highly complex problem [163–165].

GAs have been used in control applications, to tune controllers according to complex performance criteria, and have been shown to be very effective. Examples of this in the literature include [166–177]. The interested reader is referred to the survey [178] for applications of GAs in the control community. With reference to ADMIRE in particular, GAs were used in [150] for the flight control law clearance problem, using GAs to search for worst-case pilot inputs.

In [179], random search and genetic algorithm techniques were directly compared for robust controller synthesis. The GA method was found to require significantly fewer evaluations in order to arrive at a robust controller with similar performance (evaluated by a stochastic cost function).

The following section describes how the FL-TSS methodology is applied to the short-period dynamics of ADMIRE. We then move to tuning the controller using a genetic algorithm, which is applied to closed loop simulations of demanding manoeuvres, in order to help overcome the issues discussed above. Later in this chapter we address the final step in figure 7.1, checking robust stability of the closed loop by searching for a robust domain of attraction.

7.1.2 Application to ADMIRE

In this section we address the problem of finding an appropriate controller for the short-period dynamics of ADMIRE. The primary control objective is to have acceptable control of angle of attack, angle of sideslip and wind axis roll. The FL-TSS control structure defined in the previous section is constructed based on the polynomial model proposed in chapter 6 for deviation from straight and level (SL) trim, at Mach 0.5 and altitude 4000 m.

First we will define wind axis roll, as this is not explicitly given in the model of ADMIRE. Figure 7.2 shows the body axes X_b, Y_b, Z_b and stability axes X_s, Y_s, Z_s

of the aircraft, where plane OX_bZ_b and OX_sZ_s both define the 'plane of stability'. The angle of attack α is defined by the angle between X_b and the projection of the velocity vector v onto the plane of stability. The stability axes are obtained by rotating the body axes through angle α around Y_b. Y_s is therefore aligned with Y_b. The body axes rotation rates $\Omega = \begin{bmatrix} p_b & q_b & r_b \end{bmatrix}^T$ are aligned with the body axes as shown.

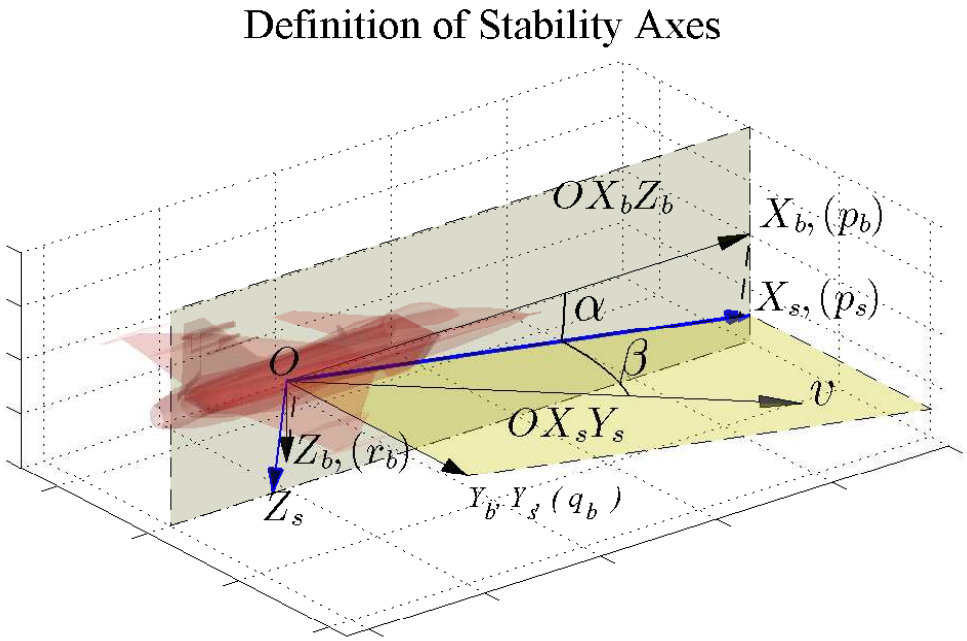


Figure 7.2: Definition of stability axis roll

For a given value of α, the magnitude (with sign) of the component of Ω along the X_s-axis defines the stability axis roll rate:

$$p_s = p_b \cos\alpha + r_b \sin\alpha \tag{7.6}$$

The sideslip angle β is defined by the angle between the velocity vector and X_s, on the OX_sY_s plane. The wind axes X_w, Y_w, Z_w are obtained by rotating the

stability axes through angle β around the Z_s-axis, such that the X_w-axis is aligned with the velocity vector v.

For given α and β, wind axis roll rate $p_w = \dot{\mu}$ is the magnitude (with sign) of the component of Ω along the X_w-axis:

$$\dot{\mu} = p_b \cos\alpha \cos\beta + q_b \sin\beta + r_b \sin\alpha \cos\beta \tag{7.7}$$

As X_w is aligned with v, this is sometimes referred to as the velocity vector roll rate.

Note that in the case of zero sideslip angle, wind axes and stability axes roll rate are the same. As it is often the case that roll manoeuvres around the velocity vector are performed at approximately zero sideslip, (7.6) is referred to in some of the literature as wind axis roll.

For controller synthesis, the polynomial approximation to ADMIRE given by (6.9) and wind axis roll defined by (7.7) is written in the control-affine form

$$\dot{x}(t) = \mathbf{f}(x(t)) + \mathbf{G}(\tilde{\alpha}(t), \beta(t))u(t) \tag{7.8}$$

where $x = \begin{bmatrix} \tilde{\alpha} & \beta & \mu & p_b & q_b & r_b \end{bmatrix}^T$ and $u = \begin{bmatrix} \delta_n & \delta_{ei} & \delta_{ey} & \delta_{ai} & \delta_{ay} & \delta_r \end{bmatrix}^T$.

The primary control outputs are $\eta = \begin{bmatrix} \tilde{\alpha} & \beta & \mu \end{bmatrix}^T = C_\eta x$, which define the states of the slow subsystem. The fast subsystem has states $\Omega = \begin{bmatrix} p_b & q_b & r_b \end{bmatrix}^T = C_\Omega x$. Note we have $n_\eta = n_\Omega = 3$, $n_u = 6 > n_\Omega$ and $n = 6$.

The envelope is the same as that of the polynomial model which is $\tilde{\alpha} = (\alpha - \alpha_{trim}) \in [-13.5, 16.5°]$, where we assume that $\alpha_{trim} = 3.5°$ is constant and $\beta \in [-15, 15°]$. Note that $(x, u) = 0$ is an equilibrium of (7.8).

In order to show that the controller (7.4) is well defined over this envelope, the determinants of $\mathbf{G_2}(\tilde{\alpha}, \beta)\mathbf{G_2}(\tilde{\alpha}, \beta)^T$ and $\mathbf{A_{12}}(\tilde{\alpha}, \beta)\mathbf{A_{12}}(\tilde{\alpha}, \beta)^T$ are shown in figures 7.3 and 7.4 respectively.

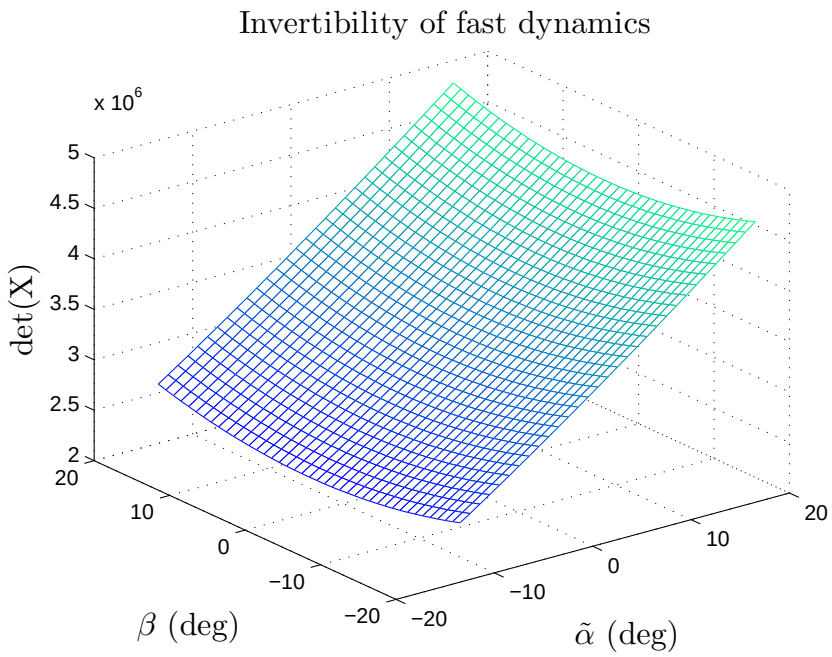


Figure 7.3: Determinant of $X = \mathbf{G_2}(\tilde{\alpha}, \beta)\mathbf{G_2}(\tilde{\alpha}, \beta)^T$

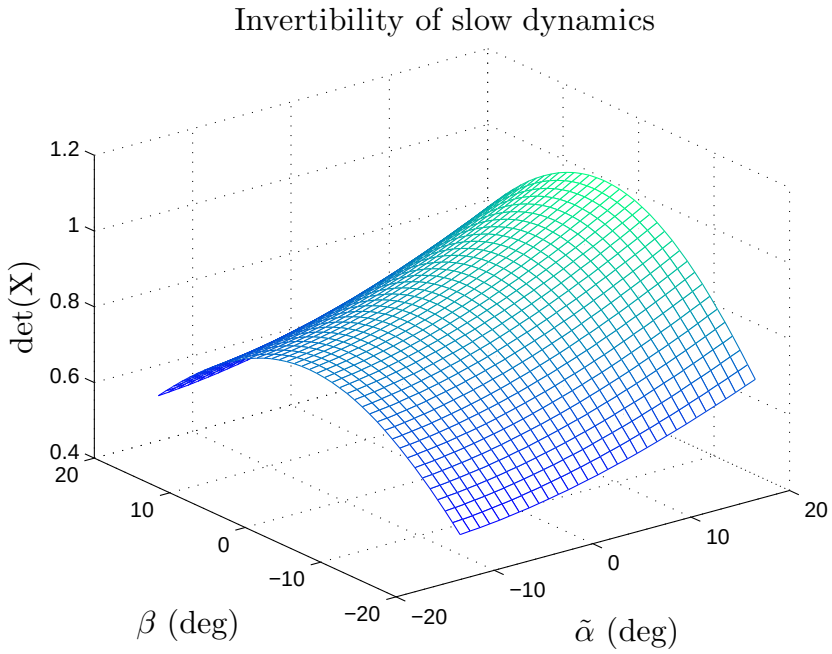


Figure 7.4: Determinant of $X = \mathbf{A_{12}}(\tilde{\alpha}, \beta)\mathbf{A_{12}}(\tilde{\alpha}, \beta)^T$

Reference model In the context of military aircraft, in order to meet Level 1 handling requirements [180] [181], the desired response of α to a command is usually

shaped as the output of a reference model:

$$\frac{\alpha}{\alpha_c} = \frac{\omega^2}{s^2 + 2\zeta\omega s + \omega^2} \tag{7.9}$$

where ω = frequency and ζ = damping ratio.

A similar model is usually used for β. This is a very common method of specifying the desired dynamics [124] [149] [151], and similar transfer functions have been used as prefilters to shape the stick/pedal commands into appropriate smooth reference values to the control system [20] [23] [24].

In the sequel we will use a reference model of the form (7.9) to shape part of the reference value for r. We use (7.9) with $\omega = 2.5$ rad/s and $\zeta = 0.8$ for α and β. For μ, we will assume that the lateral stick command is converted to a ramp (with maximum gradient 300 deg/s).

7.1.3 Tuning Using GA

Here we give a description of how the FL-TSS controller is tuned, using a genetic algorithm.

Genetic algorithms (GAs) were proposed in [162], as a method of mimicking the process of evolution by natural selection, and popularised in [182]. The basic process is to evaluate the performance of a population of variables according to some fitness criteria in simulations. The choice of fitness criteria is therefore key to the process. The next generation is formed by using the previous generation as parents, which are recombined by crossover (a combination of two or more parents), mutation (a random change in one parent), or directly taking a high-performing candidate into the next generation (an elite). This process is repeated for successive generations, until some convergence critera has been met, a maximum number of generations have been reached, or a maximum computation time limit is reached.

Genetic algorithms (GAs) have been applied for tuning controllers in aerospace applications. In [39], a GA is used to tune a NDI-TSS controller for a nonlinear missile model, using time response, damping, actuator deflection rate and GM/PM of the linearised model to define the fitness of each candidate. [53] and [54] use

GAs to tune an NDI controller for hypersonic aircraft. In this case, the fitness of each candidate is a weighted sum of binary indicators according to whether certain fixed design requirements (stability, settling time, overshoot) have or have not been met. Monte-Carlo simulations are used to evaluate each candidate with uncertain parameters.

The fitness function used in this section is somewhat more similar to that used in [171], where the fitness of each candidate is a weighted sum of total control effort, integral and absolute output tracking error.

Algorithm Description

For the controller defined in (7.4) and (7.5), there are 12 scalar parameters to tune when applied to ADMIRE. Each candidate set of controller gains k_{1-4} will be referred to as a gene.

Each gene has the form $k = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}$, where k_{1-4} are each of length 3. These define the controller gains in (7.5) as $K_1 = diag(k_1)$, $K_2 = diag(k_2)$, etc.

The GA requires an initial population, which will consist of N genes. The population size N will be fixed in each generation. The initial population is made from an initial guess for k, which we will call the seed k_0. The seed is then used to produce $N - 1$ other genes, by randomly changing each parameter of k_0 by up to 100%, which make up the remainder of the initial population.

For the initial population, and every subsequent generation, the fitness of each gene is evaluated using simulations. The controller is applied to ADMIRE (not the polynomial model of ADMIRE). The output of the controller u_c enters the actuators, which have rate and position limits specified in [2]. The output values of the states of ADMIRE are perturbed with multiplicative noise and enter the controller via the standard LTI sensors in [2].

Applying the controller to the original model of ADMIRE, rather than the polynomial model, should help to reduce sensitivity of the controller to modelling errors, as the polynomial and ADMIRE are not identical. Noise is introduced and the sensors are used rather than direct state feedback, in order to reduce sensitivity of the

controller to noise and state measurement errors.

This is illustrated in figure 7.5.

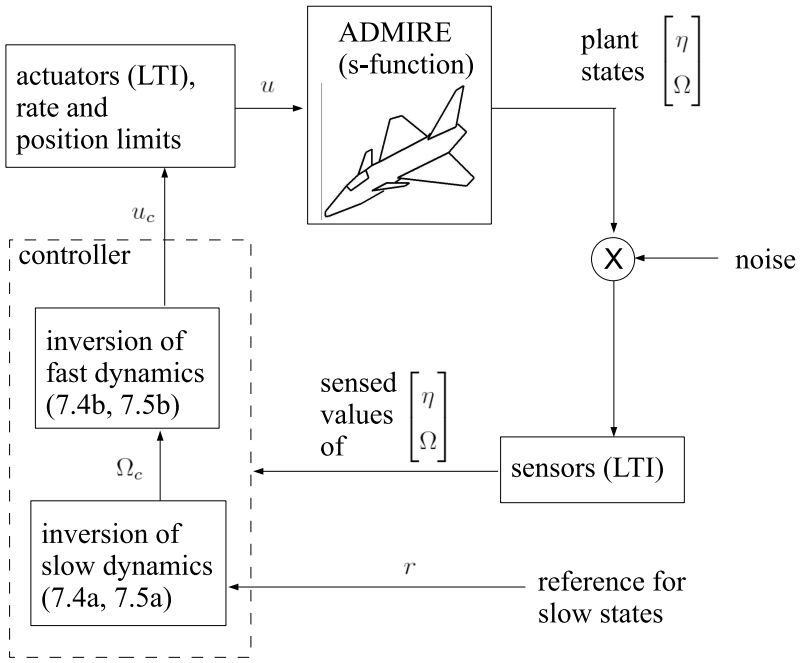


Figure 7.5: Control loop for tuning

The manoeuvres used for evaluation of each gene are given below. These are carried out from SL:

1. PULL - full longitudinal stick command $\alpha_c = 20°$ in 0.1 sec. β and μ commanded to zero.

2. PUSH - full longitudinal stick command $\alpha_c = -10°$ in 0.1 sec. β and μ commanded to zero.

3. ROLL - full lateral stick command, which is a ramp input to μ_c with a slope of 300 deg/s. This is held at $\alpha_c = 5°$. β is commanded to zero.

4. TURN - full lateral stick command to roll around the velocity vector by approximately 90°, with full longitudinal stick pull. This is held for 5 sec, followed by longitudinal and lateral stick commands in the opposite direction. This manoeuvre changes the heading of the aircraft by approximately 90°. β is commanded to zero.

All of the simulations above are run for 10 sec (where possible). For these simulations, the envelope check bundled with ADMIRE is disabled. The values $|\alpha| = 50°$ and $|\beta| = 50°$ are used to terminate the simulation early, as an indication of instability. The only control input from the bundled ADMIRE FCS is the engine throttle setting, although fixing the value at zero makes little difference in these simulations.

Each gene therefore has its fitness assessed over four simulations. For each gene, four fitness values f_{1-4} (scalars) are derived from the simulated manoeuvres. The total fitness f of a gene is the sum of the fitness in all four simulations $f = f_1 + f_2 + f_3 + f_4$.

The length of the simulation is t_m, which should be 10 sec unless the simulation terminates early due to instability.

The fitness in each simulation f_i is calculated as follows:

- IF the simulation terminates early ($t_m < 10$s), then $f_i = 50+100(10-t_m)$ (this penalises genes that cause instability, according to how early the simulation terminates).
- ELSE $f_i = w \left[|e|_{max} \quad \int_0^{10} \varepsilon_i^2 \, \mathrm{d}t \quad \int_0^{10} \epsilon_i^2 \, \mathrm{d}t \quad |u_{adm,c}|_{max} \quad \int_0^{10} u_{adm}^T u_{adm} \, \mathrm{d}t \quad |n_z|_{max} \right]^T$.

where w is a vector of weightings of length 15, and:

- $|e|_{max}$ is the maximum absolute tracking error in each state (vector of length 6)
- $\int_0^{10} \varepsilon_i^2 \, \mathrm{d}t$ and $\int_0^{10} \epsilon_i^2 \, \mathrm{d}t$ are the integral square tracking errors in the fast and slow states (total length 6)
- $|u_{adm,c}|_{max}$ is the maximum absolute value of all the commanded control surface deflections (scalar)
- $\int_0^{10} u_{adm}^T u_{adm} \, \mathrm{d}t$ is the integral control effort (scalar)
- $|n_z|_{max}$ is the maximum absolute value of the normal acceleration in the simulation (scalar)

Each of the N genes therefore has a fitness f associated with it, which we aim to minimise. Note that the GA does not require the seed or any member of the initial population to be stable.

Making the next generation The genes in the current generation are selected as *parents* according to their fitness f. The likelihood of a particular gene being selected as a parent is inversely proportional to its fitness. The genes that make up the next generation are the *children*. The children have 3 types: *elites*, *mutants* and *crossovers*.

Elites are the N_{elite} members of the current generation with the best (lowest) fitness. The elites are carried through to the next generation with no change. Keeping a small number of elites in the selection means that the best fitness cannot get worse from one generation to the next.

Mutants are N_{mutant} members of the next generation that have just one parent. A mutant is made by selecting a parent and randomly changing one parameter by up to 100%. Having mutants in the selection helps to add diversity to the population and reduces the tendency for the algorithm to get stuck in a local minimum.

Crossovers are the remaining $N-N_{elite}-N_{mutant}$ members of the next generation. Crossovers have 2 parents. A crossover child is made by a randomly-weighted average of its parents.

Initial trials The population size was fixed at $N = 14$, with $N_{elite} = 2$ and $N_{mutant} = 6$. After some initial trial runs, an integral gain k_2 =[0 0 0] was fixed (the integral tracking error on the fast states). Good results were obtained without these 3 variables, hence this was fixed in order to reduce the total number of variables. This makes sense as an integral gain is generally used to eliminate steady-state error, which is important for the slow states (the primary control outputs) but less so for the fast states.

The weighting vector used was

w =[5 5 1 1 1 1 100 100 1 0 0 0 1 1 0.1]

The seed used was

$k_0 = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}$ =[10 10 10 0 0 0 5 5 5 2 2 2]

Simulations of the 4 manoeuvres with the initial seed follow.

Simulations with Initial Seed

Figures 7.6-7.9 show the 4 manoeuvres that are used for fitness evaluation, with the initial seed set of controller gains. These figures show the states of ADMIRE α, β, μ, p_b, q_b, r_b, together with wind axis roll rate $\dot{\mu}$ and normal acceleration n_z.

Note, the seed produces instability in all the manoeuvres apart from manoeuvre 3: wind axis roll.

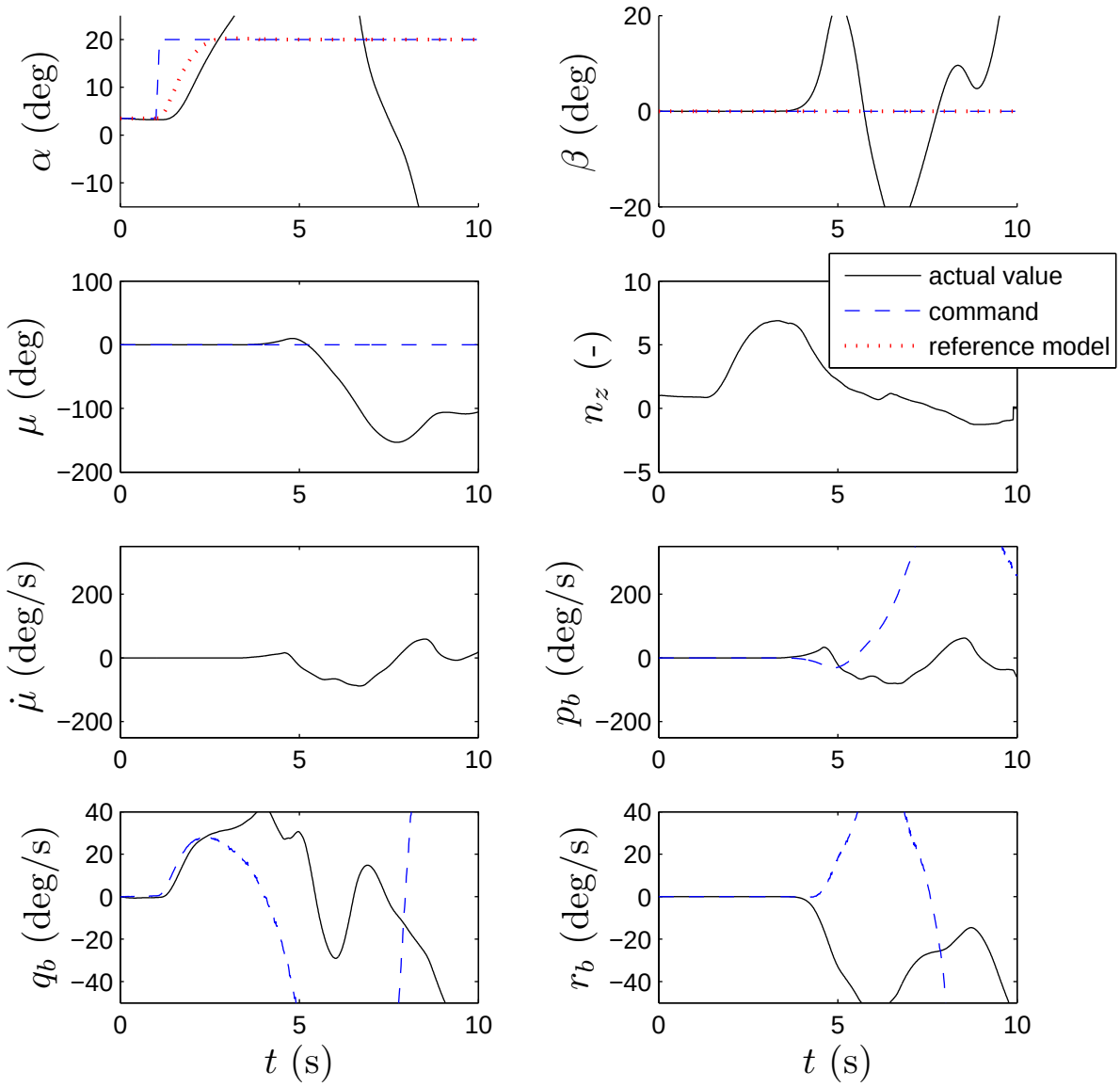


Figure 7.6: Pull manoeuvre initial seed

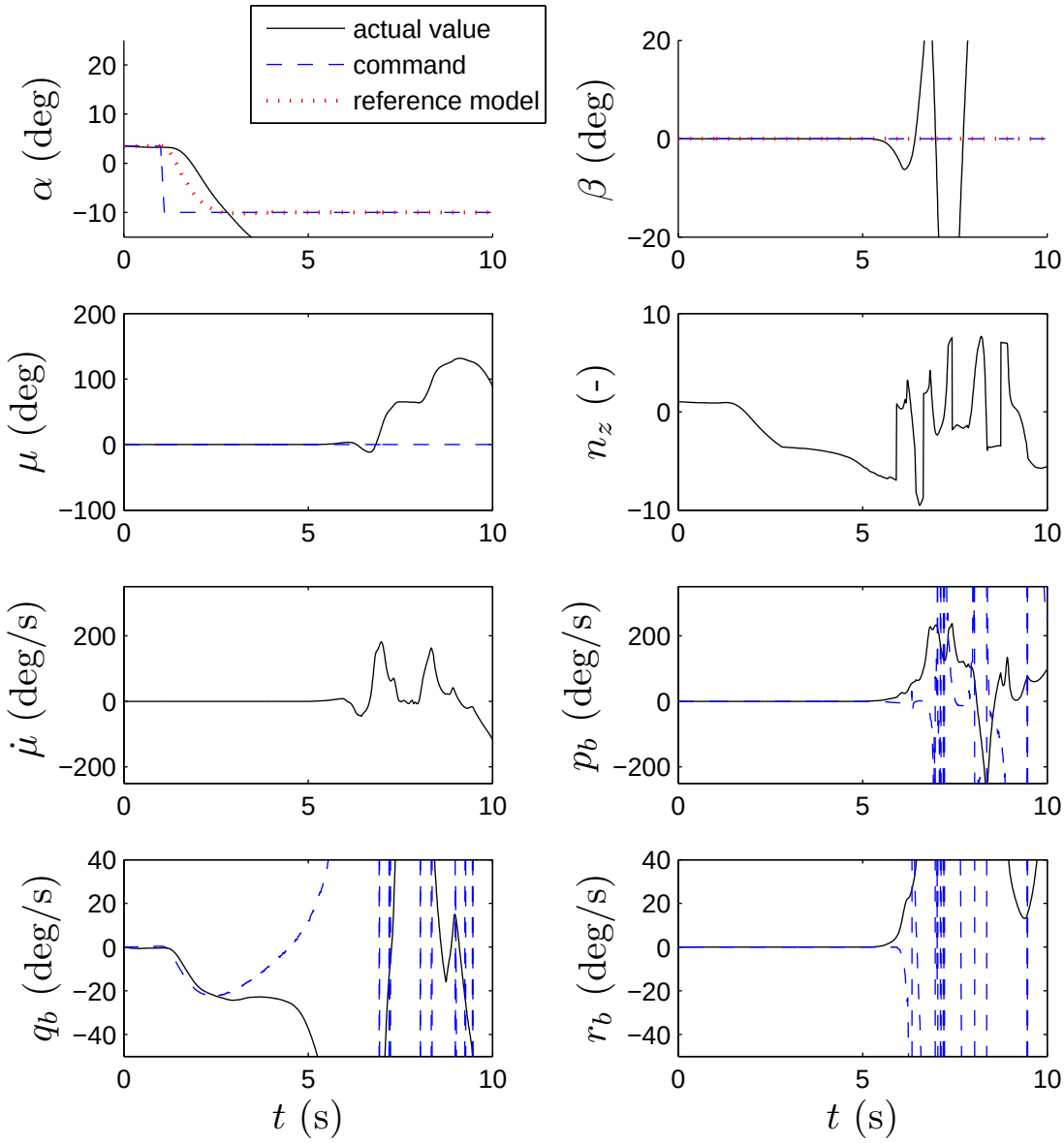


Figure 7.7: Push manoeuvre initial seed

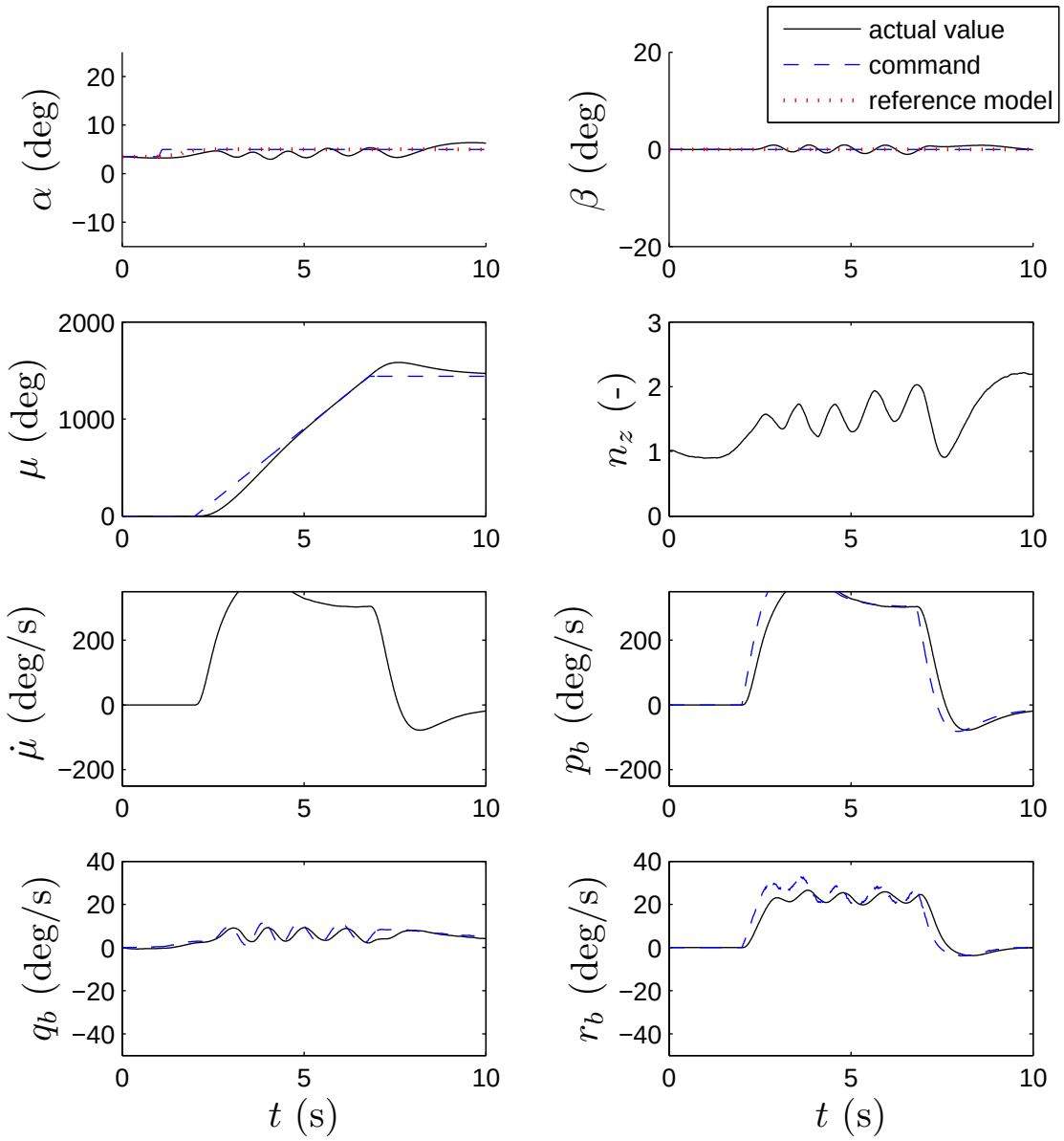


Figure 7.8: Roll manoeuvre initial seed

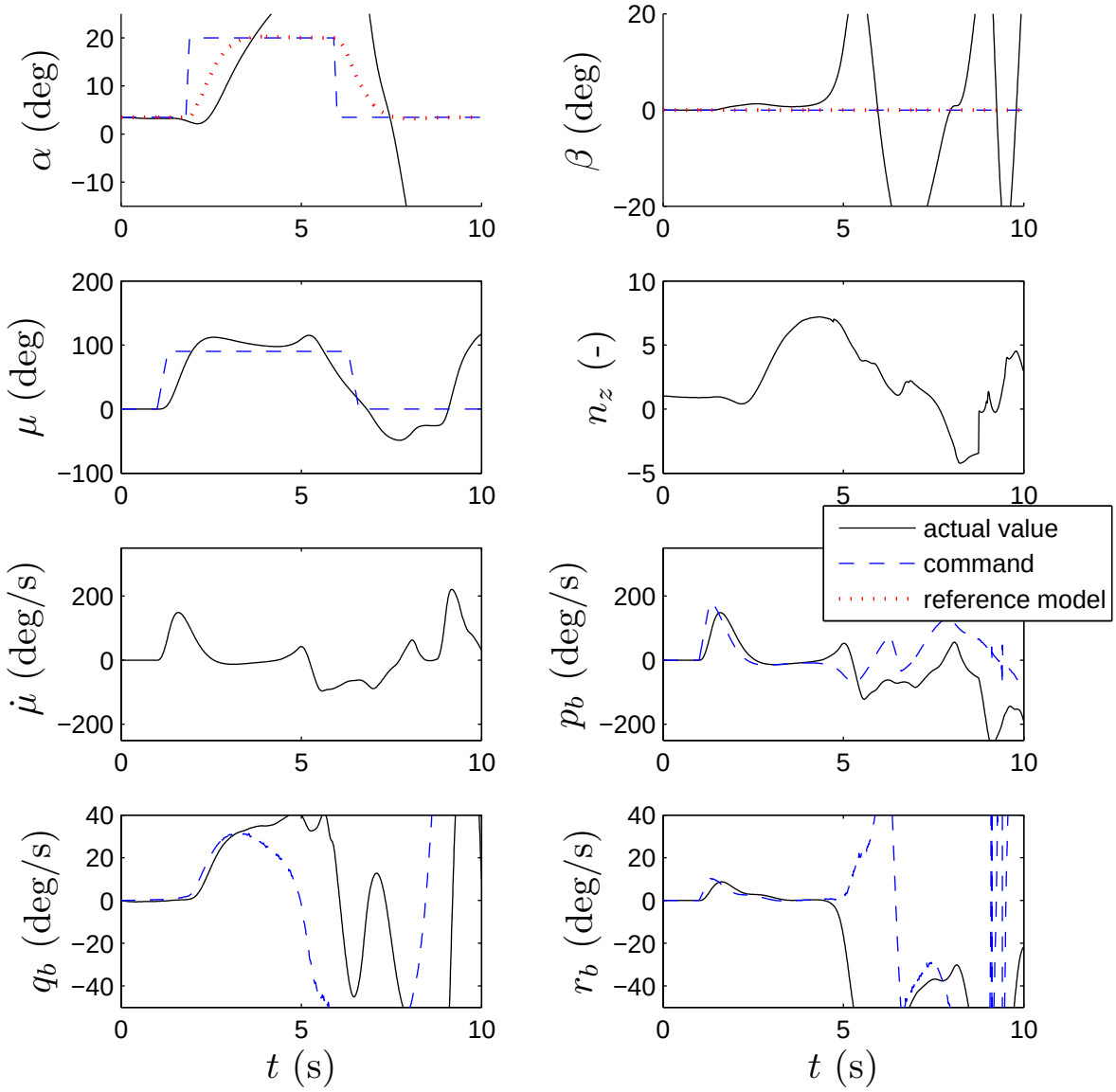
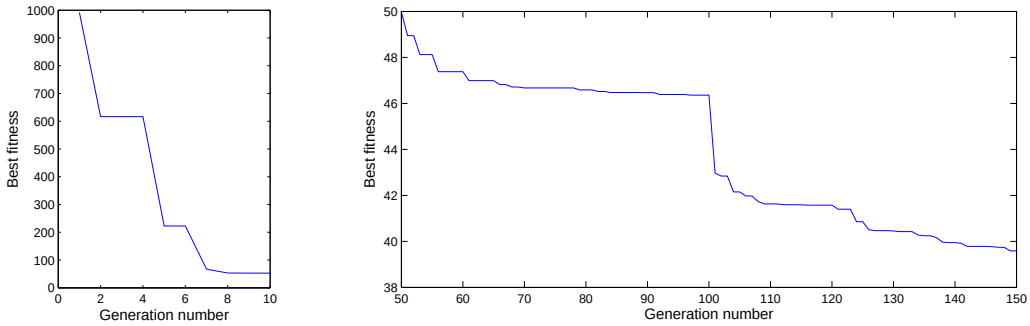


Figure 7.9: Turn manoeuvre initial seed

GA Progress

The GA was run for 150 generations. Figure 7.10 shows the progress of the GA. The fitness value of the best gene (the gene with the lowest fitness) in each generation is plotted vs. the generation number. Note, the high value of the best fitness in the first few generations shows that the best gene was not producing stability

in all 4 manoeuvres. However, within 10 generations the best fitness has dropped dramatically, showing the the GA has found a gene that is stable for all 4 manoeuvres (though not with good performance). After the 10th generation, progress is somewhat slower, with plateaus followed by sudden drops in fitness (perhaps due to an effective mutant).



(a) First 10 generations (b) Last 100 generations

Figure 7.10: GA best (lowest) fitness in each generation

Value of best gene after 150 generations:

$k_{best} = \begin{bmatrix} k_1 & k_2 & k_3 & k_4 \end{bmatrix}$, where

$$k_1 = \begin{bmatrix} 10.8628 & 12.5506 & 5.2597 \end{bmatrix} \tag{7.10a}$$

$$k_2 = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad \text{(fixed a-priori)} \tag{7.10b}$$

$$k_3 = \begin{bmatrix} 5.7819 & 3.8931 & 2.4111 \end{bmatrix} \tag{7.10c}$$

$$k_4 = \begin{bmatrix} 2.1074 & 0.6279 & 0.4137 \end{bmatrix} \tag{7.10d}$$

7.1.4 Closed Loop Simulation Results

Responses with the final controller

Figures 7.11-7.25 show the response of ADMIRE with the 'best' controller following the GA optimisation. The 4 manoeuvres used for evaluation are shown, plus an extra sideslip manoeuvre that was not included in the fitness evaluation.

Figs 7.11-7.13 show the state response, other outputs and control response respectively, for the pull manoeuvre. Figs 7.14-7.16 show the push manoeuvre. Figs

7.17-7.19 show the wind axis roll. Figs 7.20-7.22 show the turn manoeuvre. Figs 7.23-7.25 show a sideslip doublet.

In each case, the responses shown are the actual outputs of ADMIRE. Where appropriate, command and reference signals are also shown.

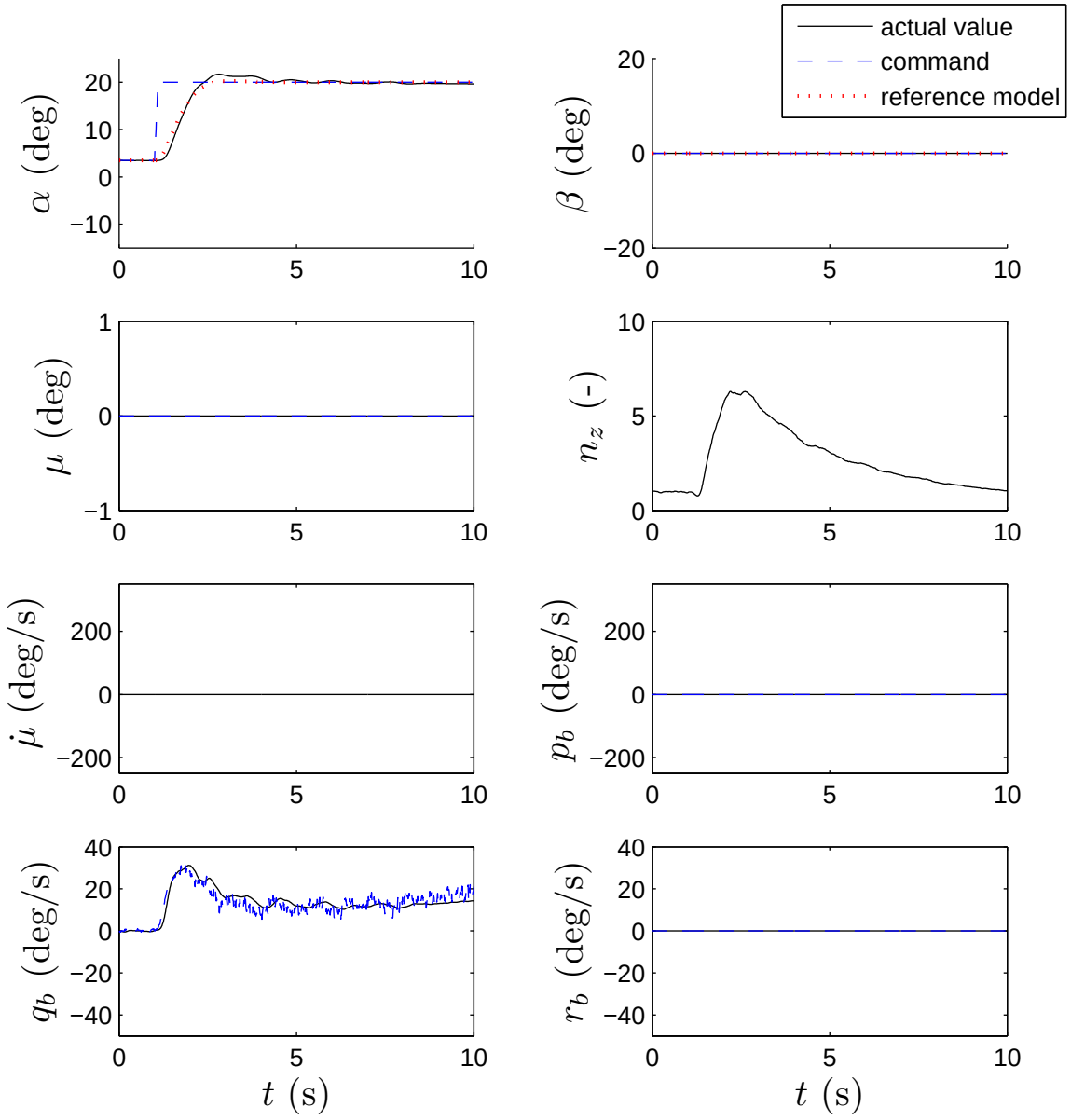


Figure 7.11: Pull manoeuvre with final controller - state response

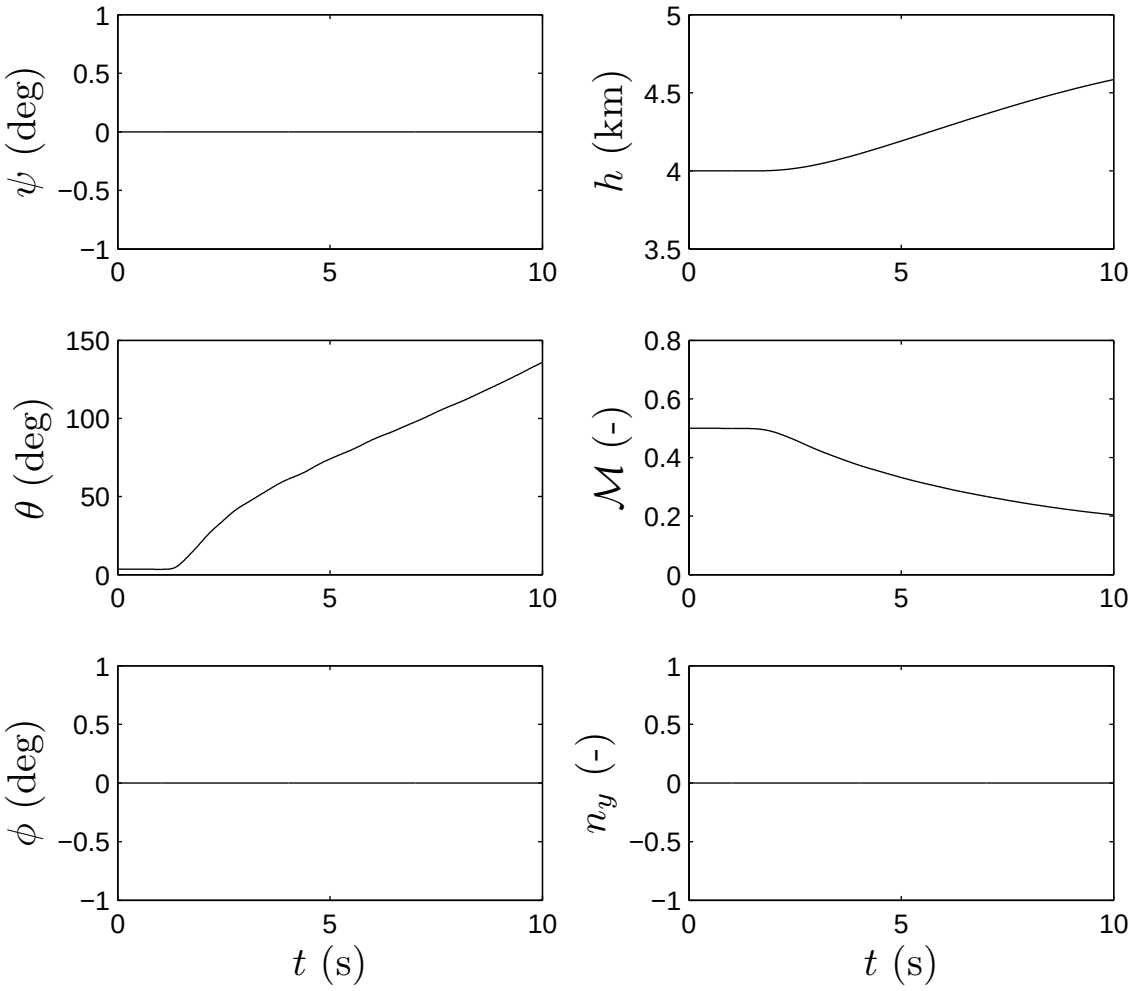


Figure 7.12: Pull manoeuvre with final controller - other outputs

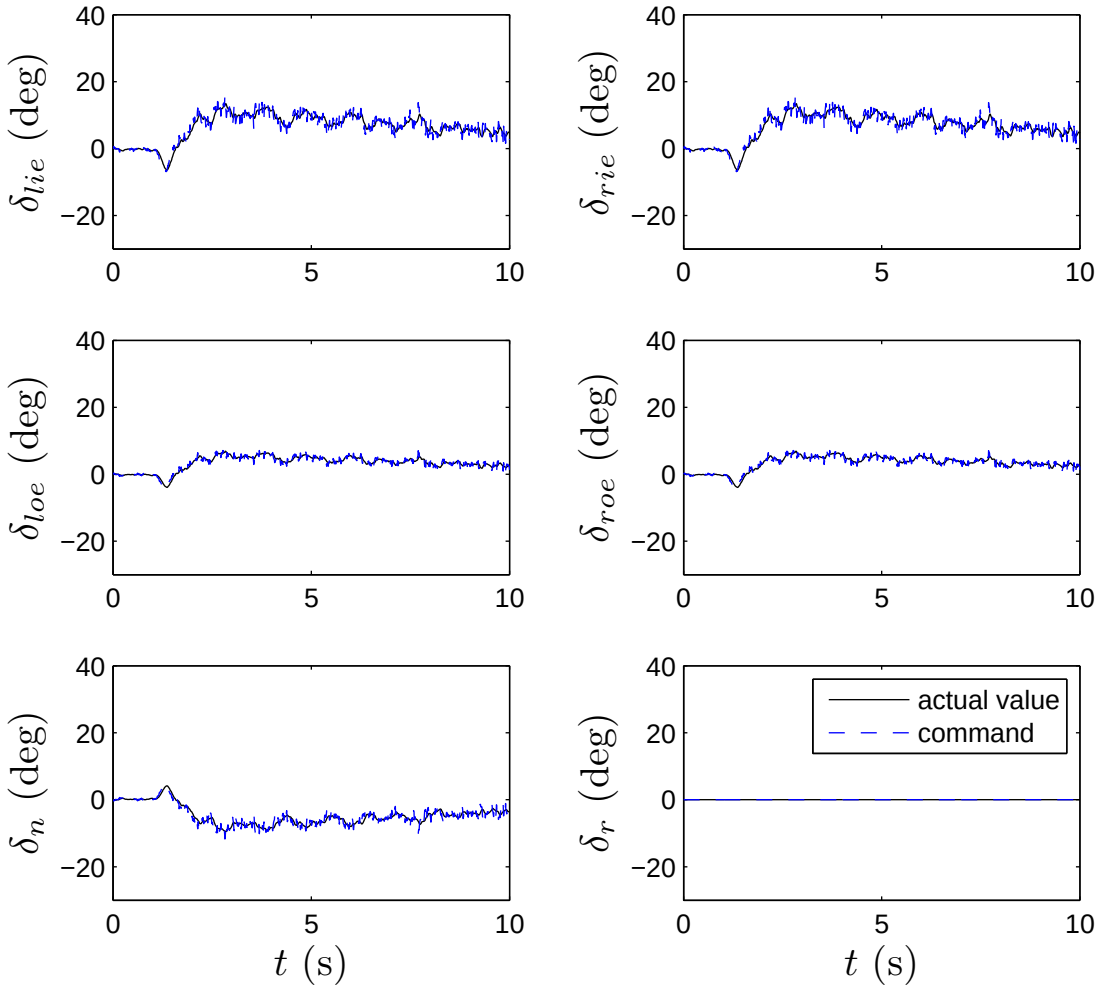


Figure 7.13: Pull manoeuvre with final controller - control response

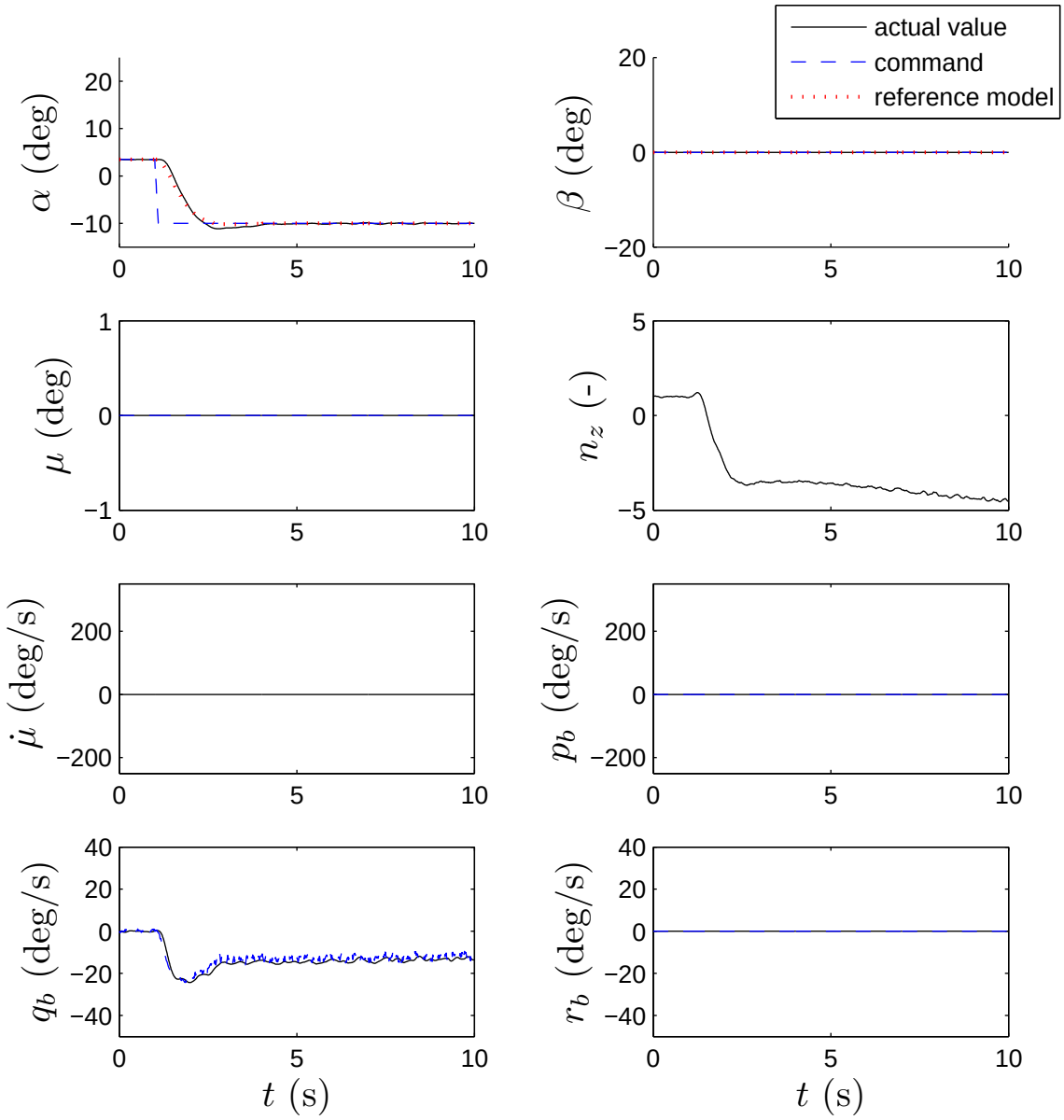


Figure 7.14: Push manoeuvre with final controller - state response

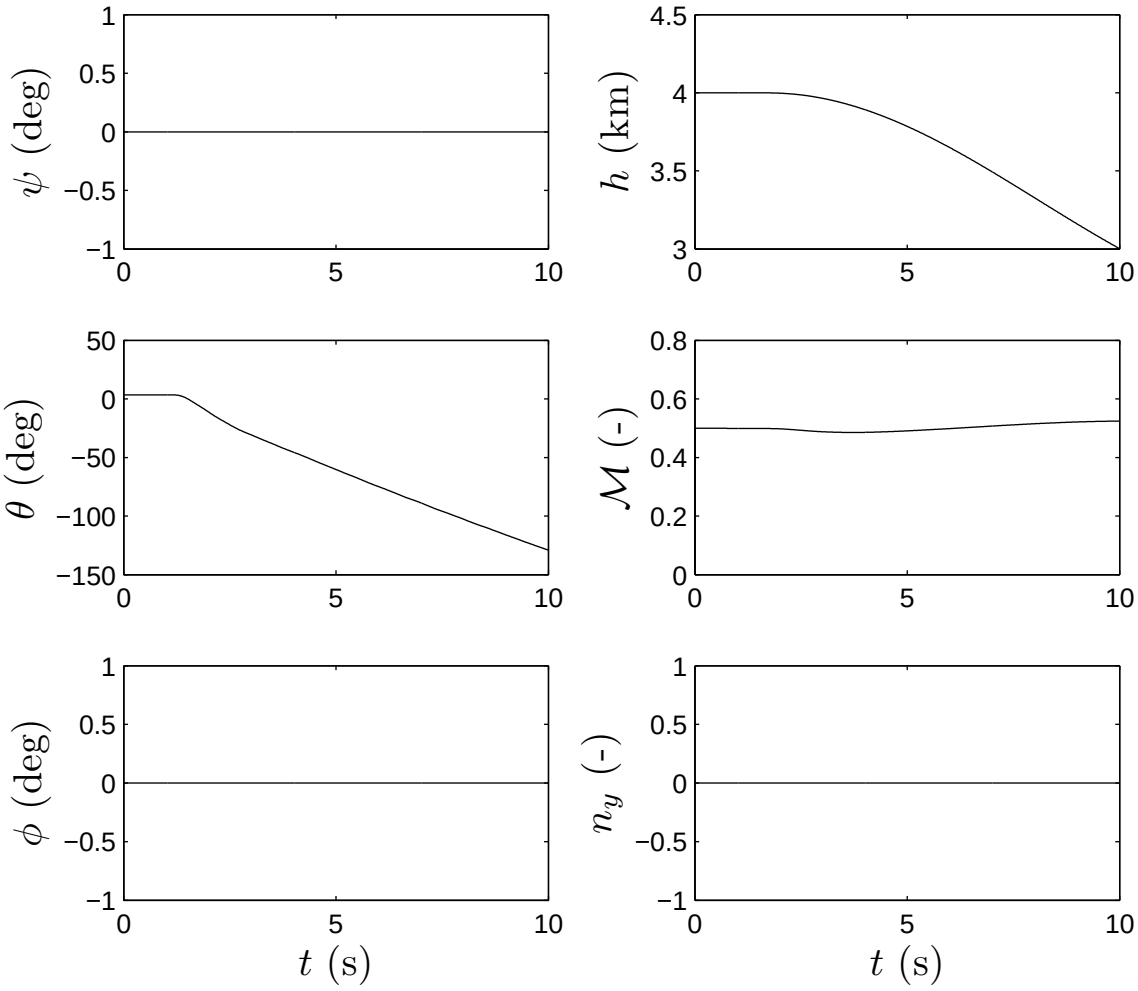


Figure 7.15: Push manoeuvre with final controller - other outputs

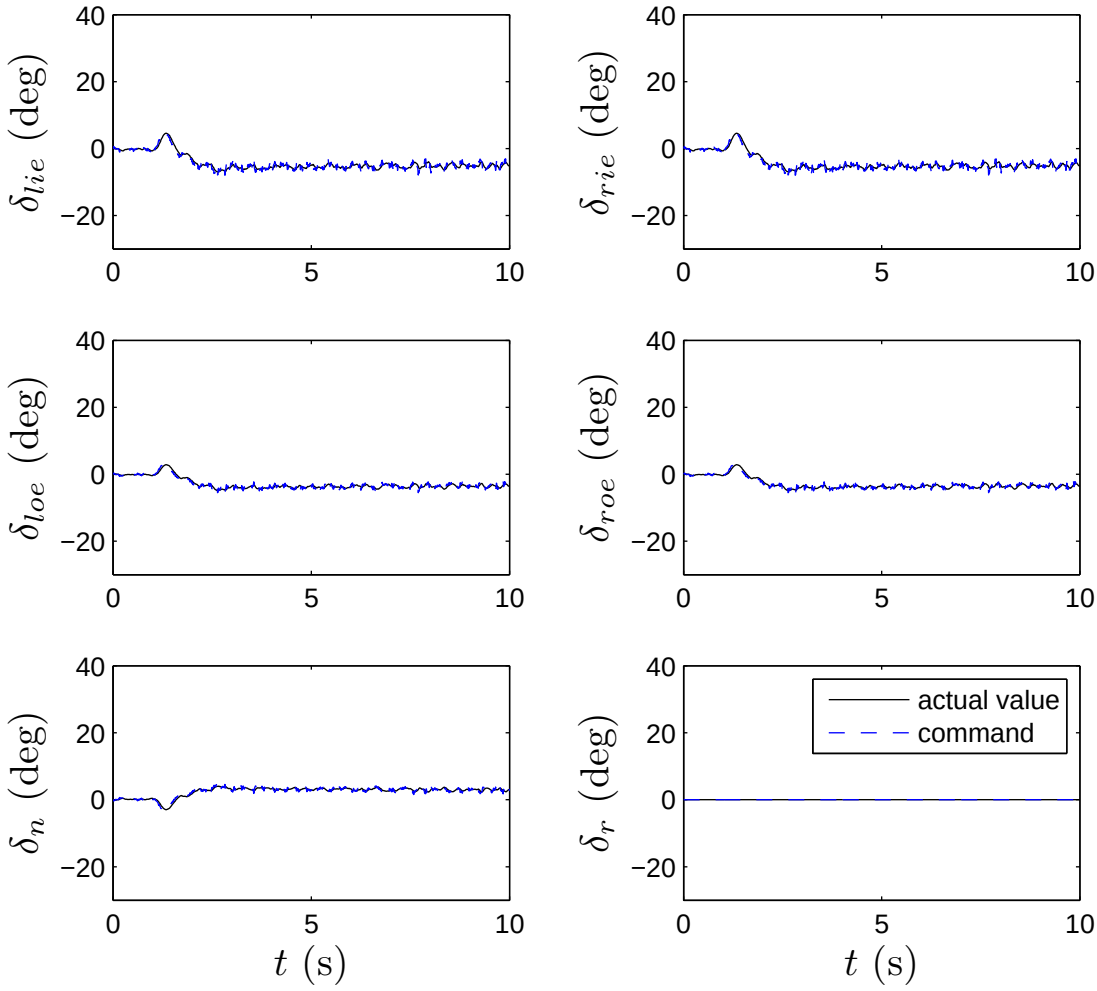


Figure 7.16: Push manoeuvre with final controller - control response

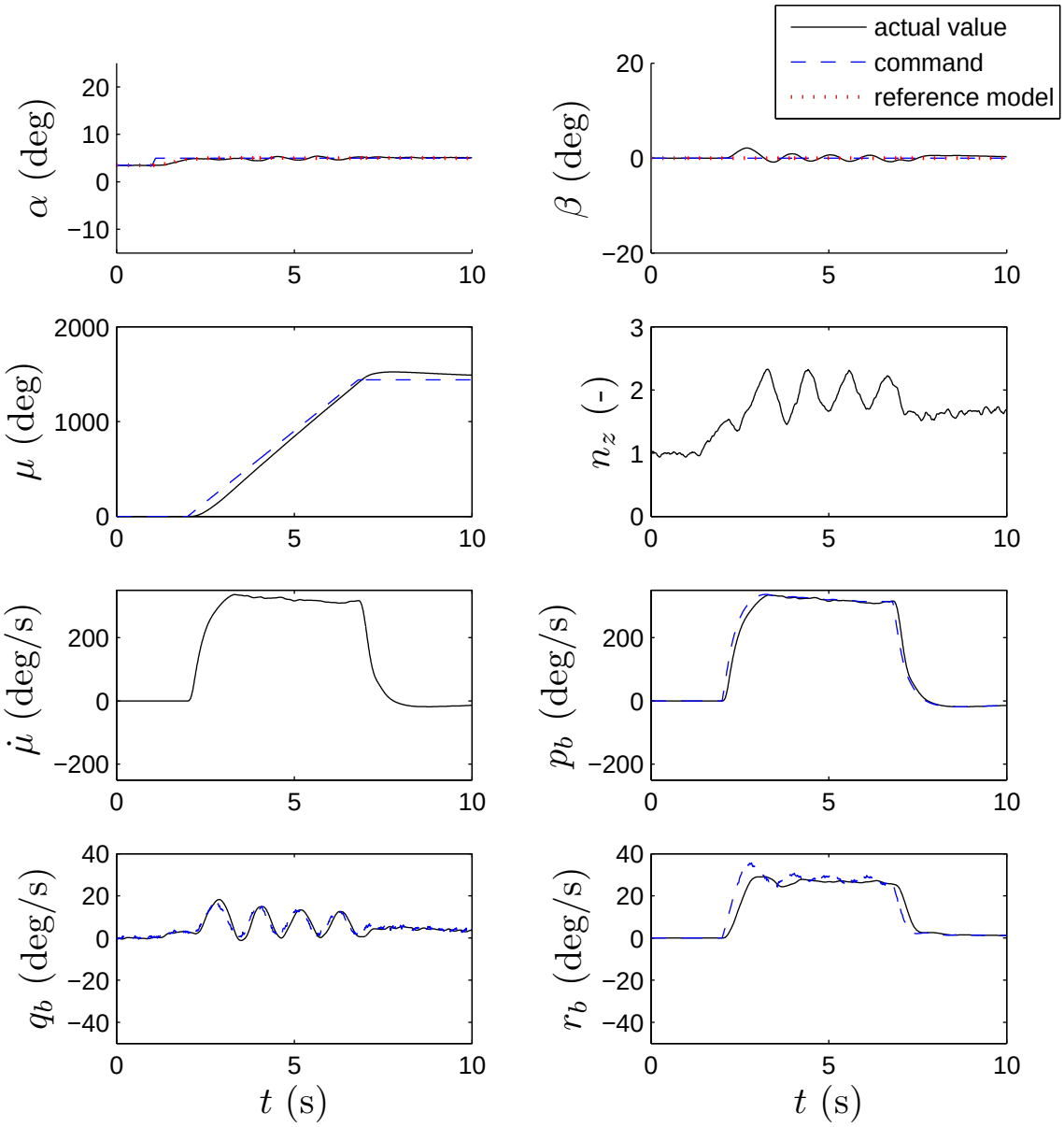


Figure 7.17: Roll manoeuvre with final controller - state response

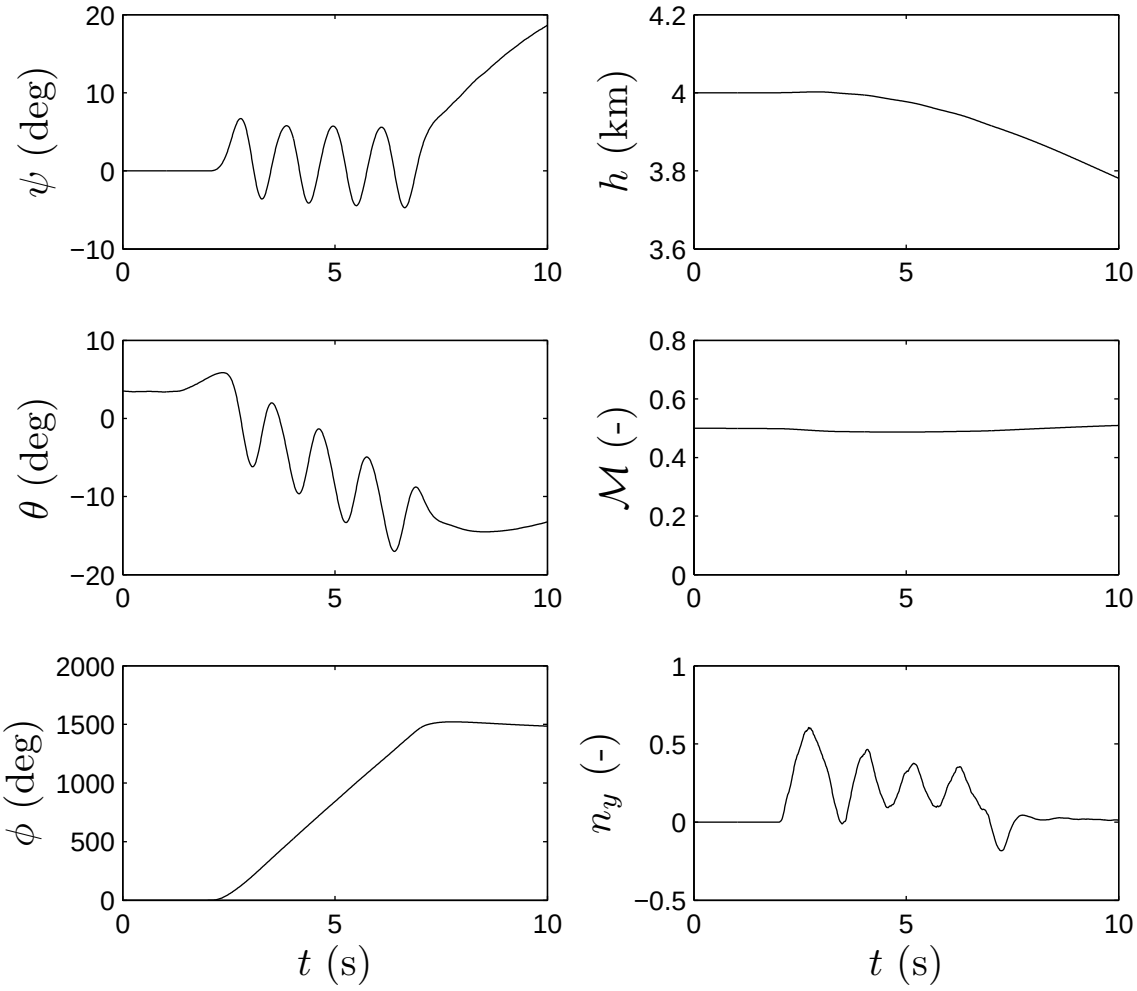


Figure 7.18: Roll manoeuvre with final controller - other outputs

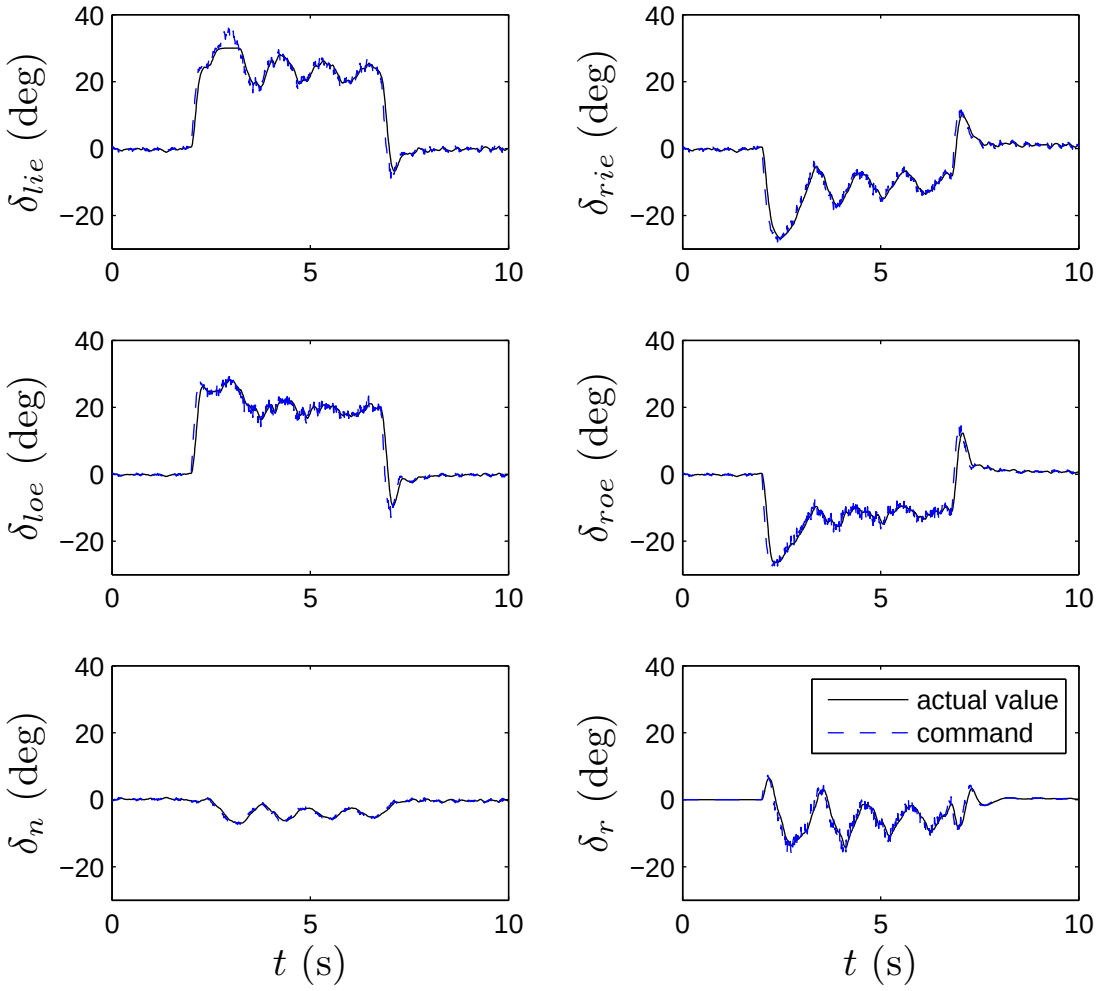


Figure 7.19: Roll manoeuvre with final controller - control response

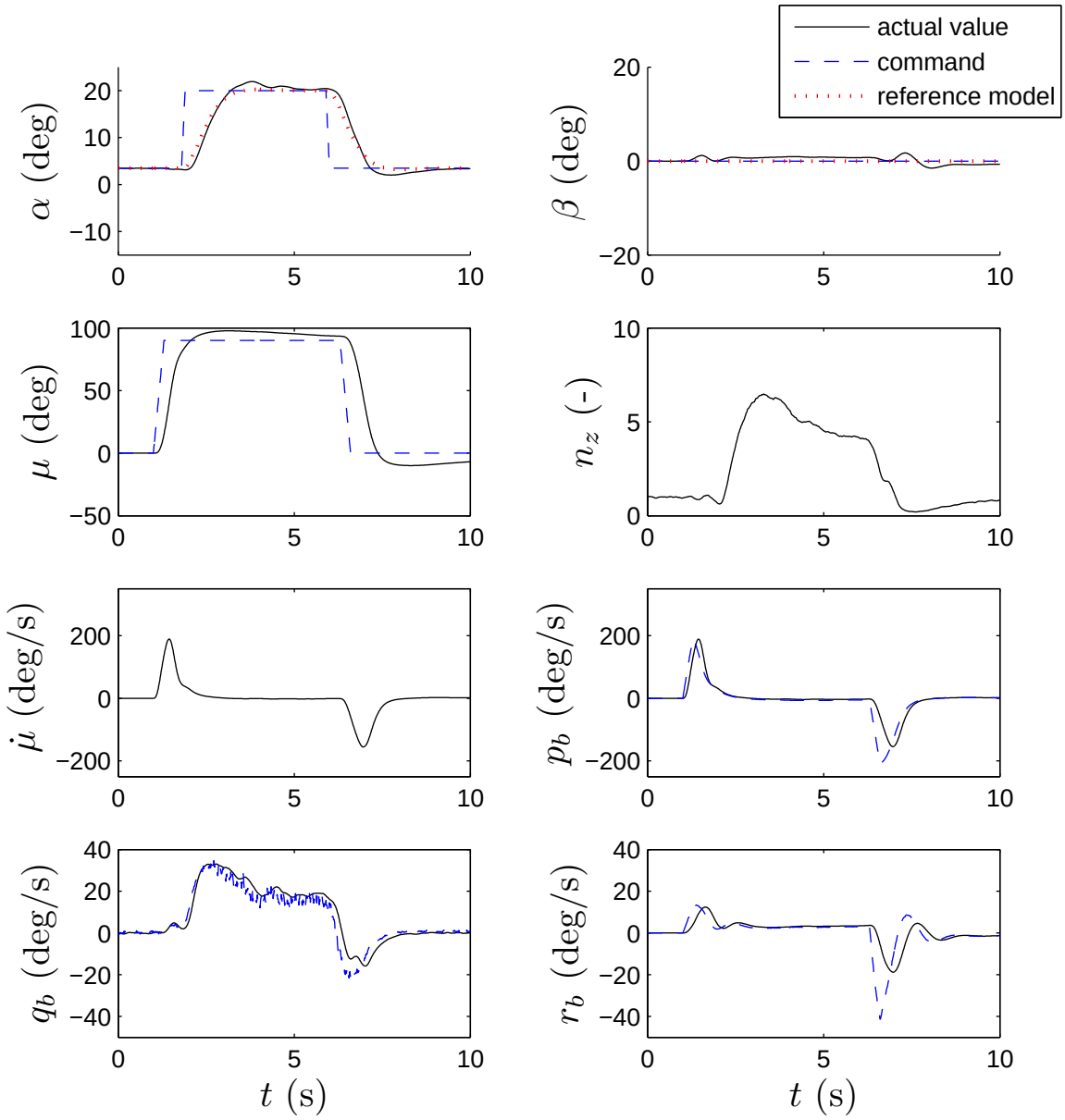


Figure 7.20: Turn manoeuvre with final controller - state response

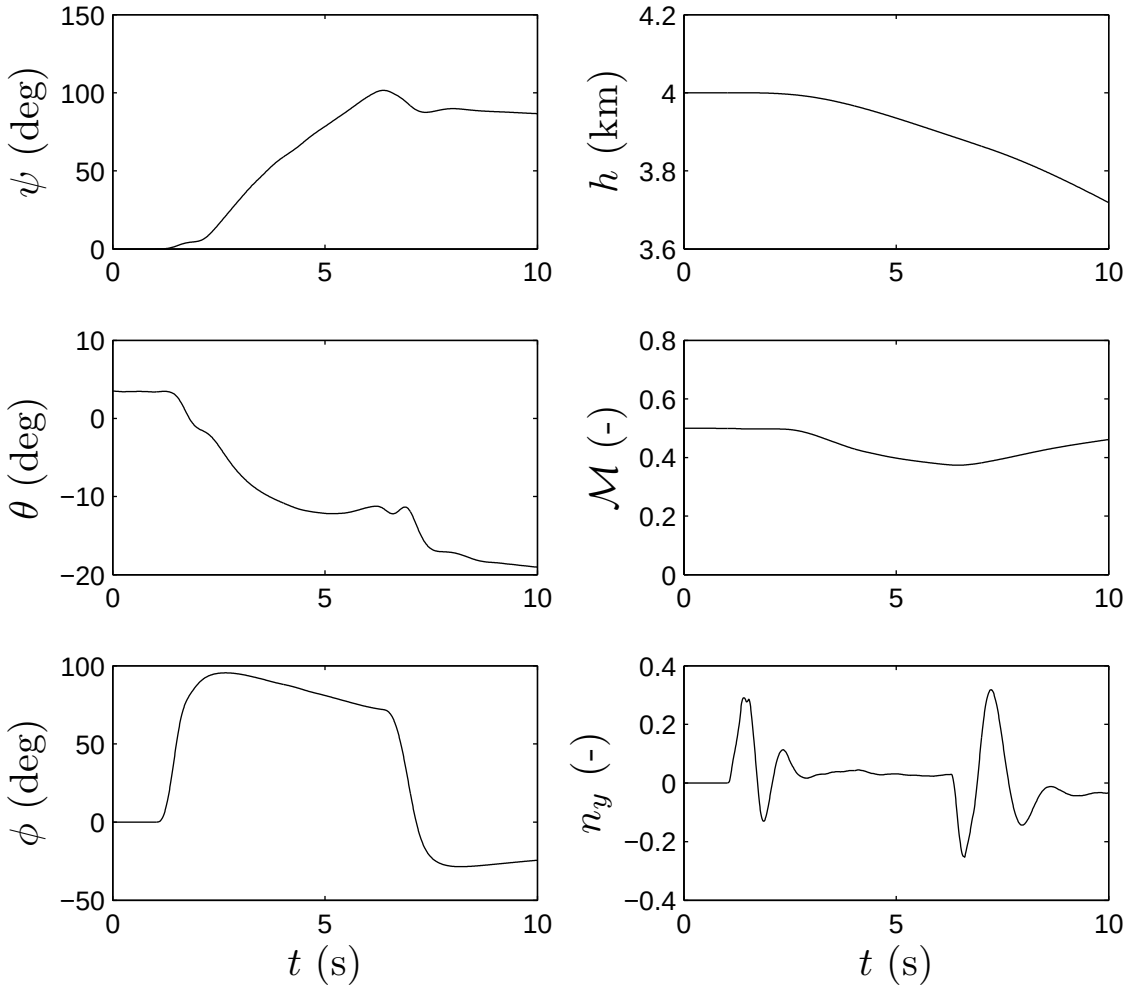


Figure 7.21: Turn manoeuvre with final controller - other outputs

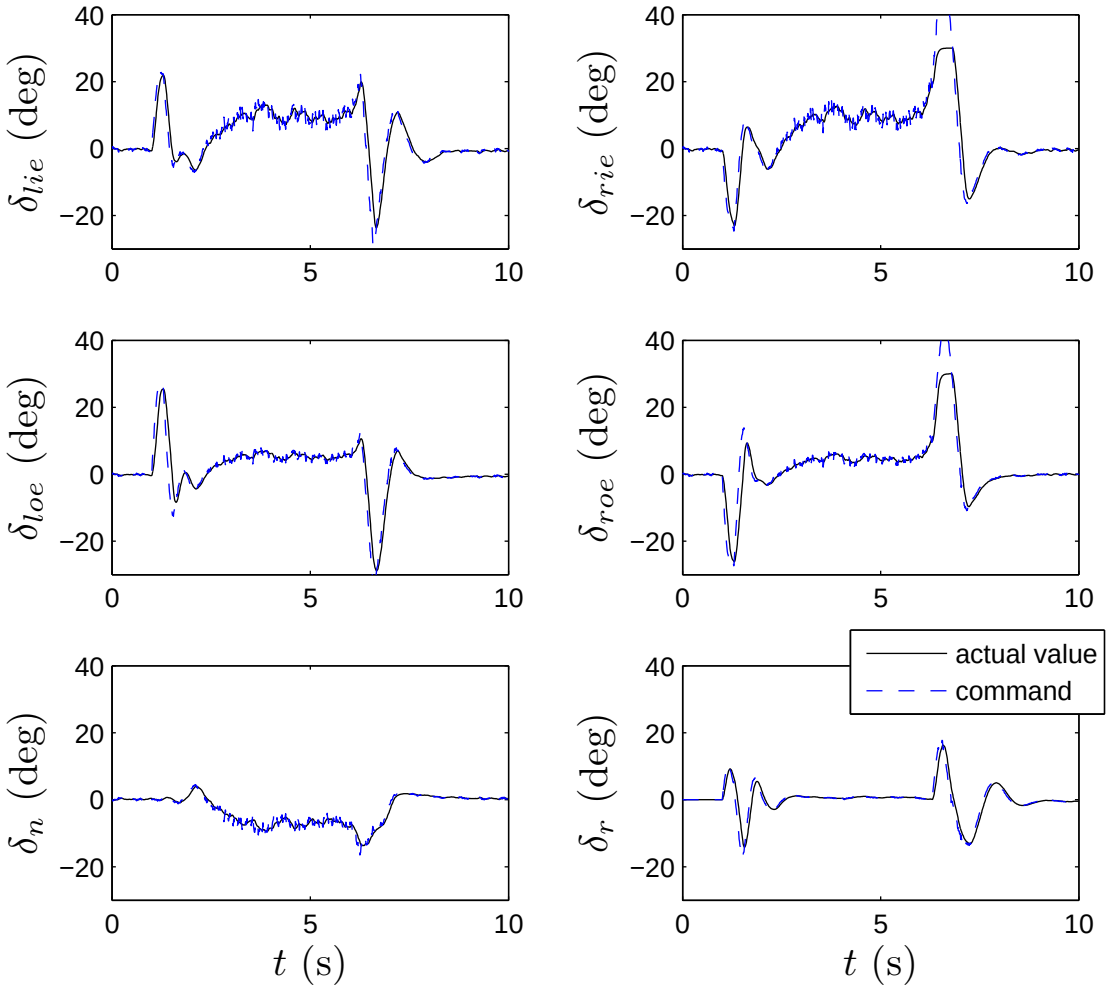


Figure 7.22: Turn manoeuvre with final controller - control response

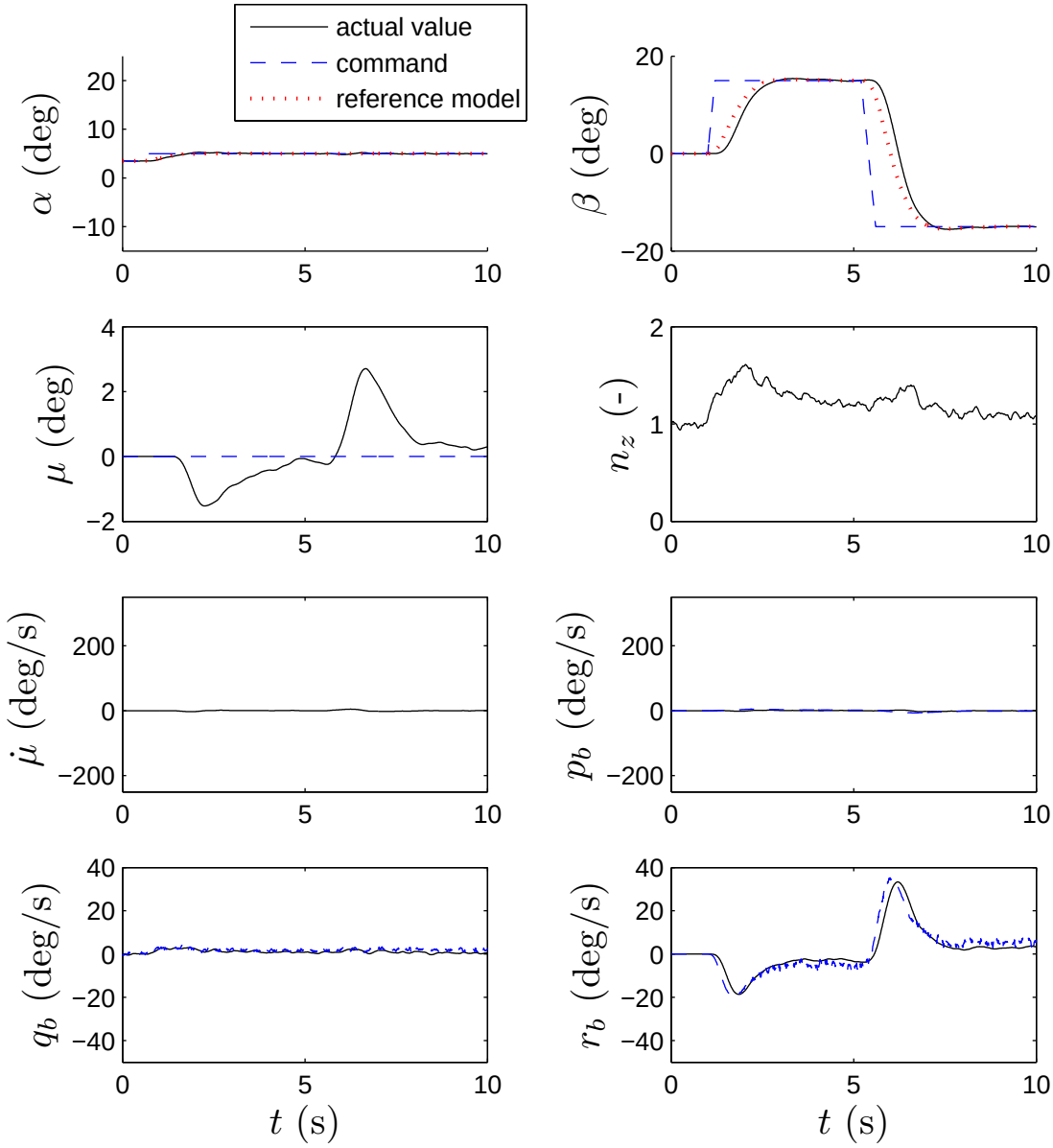


Figure 7.23: Sideslip manoeuvre with final controller - state response

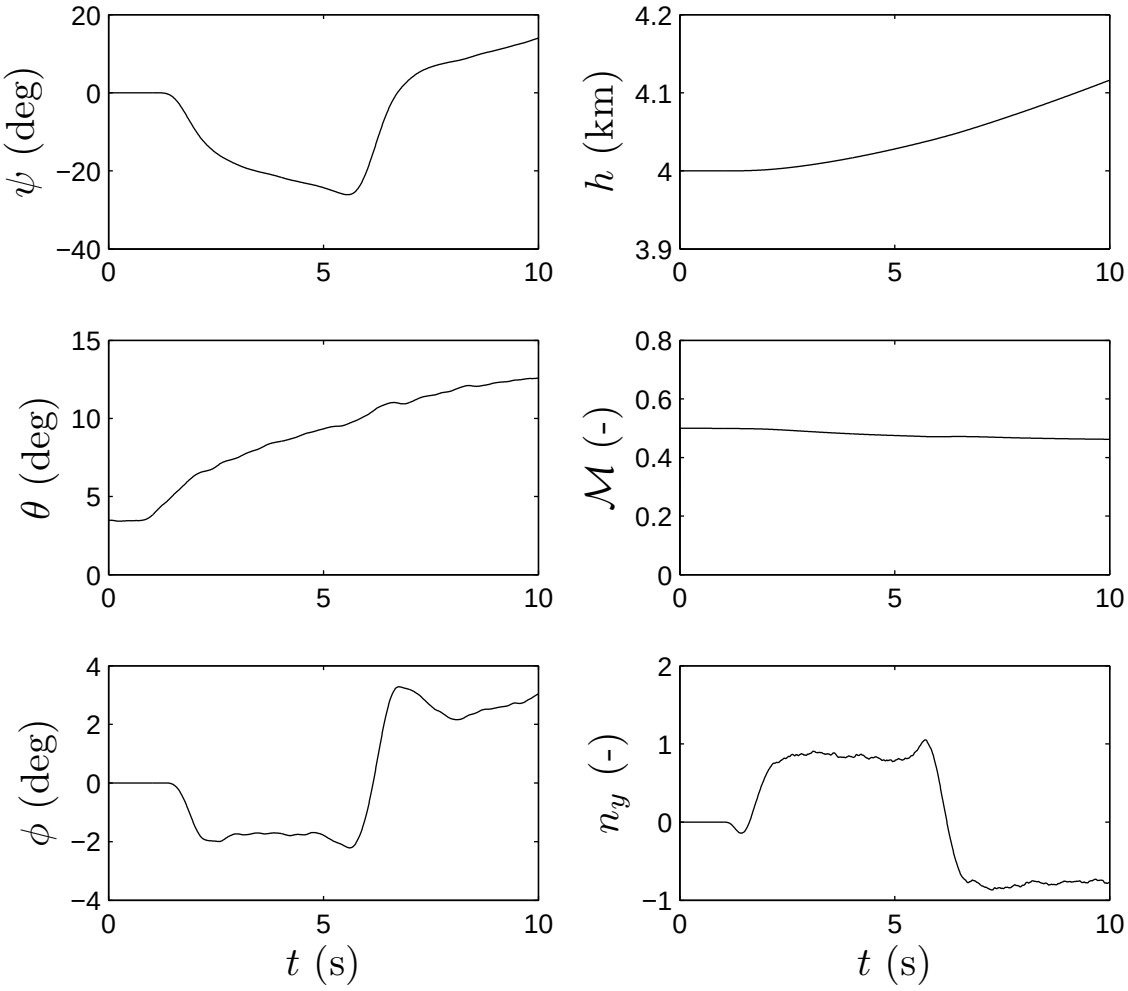


Figure 7.24: Sideslip manoeuvre with final controller - other outputs

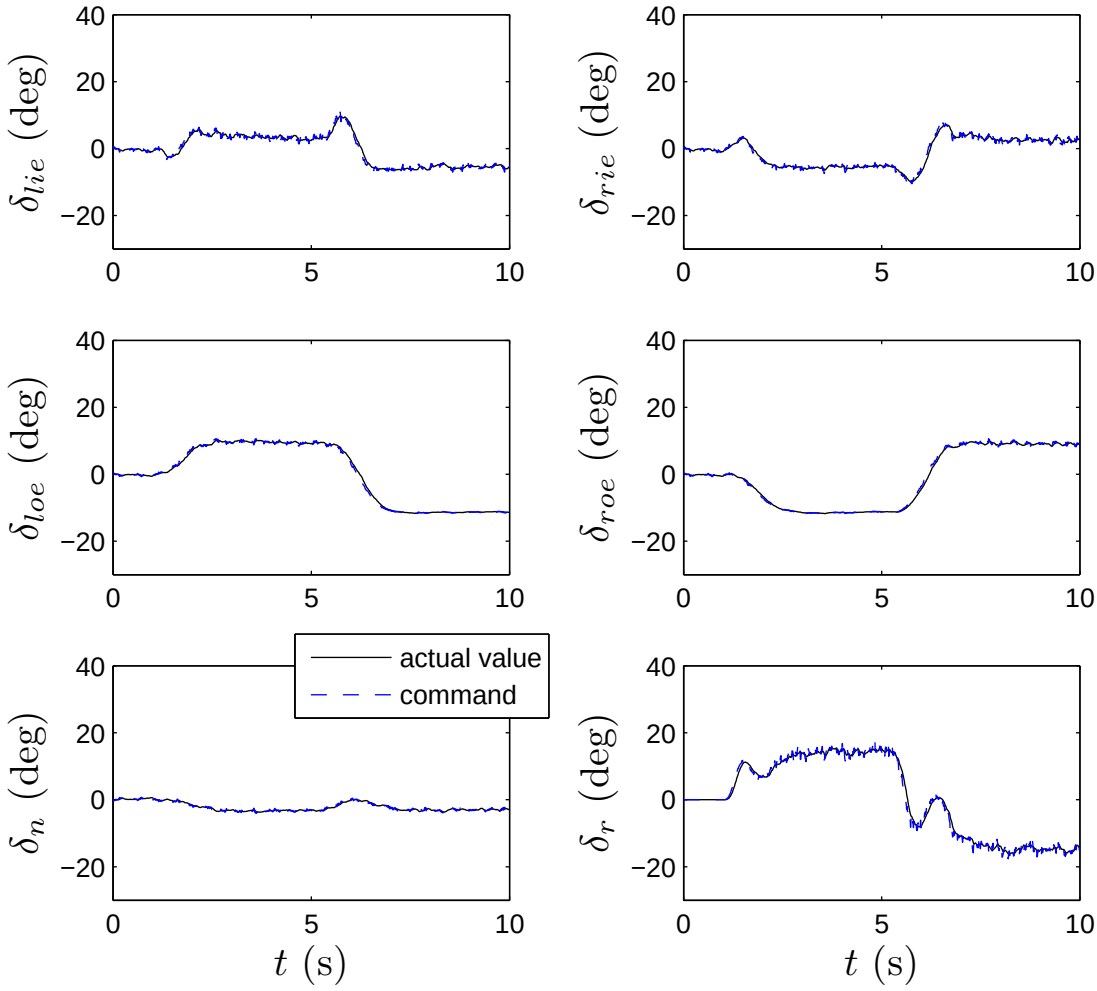


Figure 7.25: Sideslip manoeuvre with final controller - control response

Each manoeuvre displays good tracking of α, β and μ. Noise is not unduly amplified by the controller.

In the pull manoeuvre, good tracking is achieved despite the large change in Mach (see fig 7.12), which suggests the controller is not very sensitive to modelling inaccuracies in making the polynomial model (which was made for $\mathcal{M} = 0.5$). Note that the system is actually outside the envelope for $\mathcal{M} < 0.3$, around $t = 7$s. In the push manoeuvre again good tracking is achieved, but this manoeuvre requires the normal acceleration to drop below -3g, which is outside the envelope. The roll manoeuvre shows good tracking and only very small induced sideslip (see fig 7.17).

The turn manoeuvre shows slightly less good tracking of μ, but good tracking of α (fig 7.20). Again, the induced sideslip is very small. Note, the turn manoeuvre described in [148] requires a pilot model to adjust the control command to maintain approximately constant altitude, which we did not do here.

The only manoeuvres during which any control surfaces saturate are the roll fig 7.19 and the turn fig 7.22. The left inner elevon saturates for around 0.5 sec near the beginning of the roll, however this does not unduly effect performance or cause instability. Right inner and outer elevons saturate for less than 0.5 sec during the latter part of the turn manoeuvre. Again, this does not cause instability although there is a degredation of the tracking error on μ. Note this is a very aggressive manoeuvre that requires simultaneous commands on α and μ at the maximum rates. During the sidelsip doublet, there is negligible induced change in angle of attack and only small induced μ (around 2 deg) when the sideslip command changes.

Comparing Models - closed loop

The controller is defined by FL-TSS with the polynomial model, and was tuned by applying this to ADMIRE. In the sequel, we want to give a robust stability analysis of the closed loop, which requires using the polynomial model. The following simulation results are to find out whether the closed loop behaviour of the polynomial is very different to that of ADMIRE. μ and $\dot{\mu}$ are derived from the other plant states, hence only α, β, p_b, q_b and r_b are shown. Closed loop simulations with the controller applied to the (Jacobian) linear model are also given for comparison.

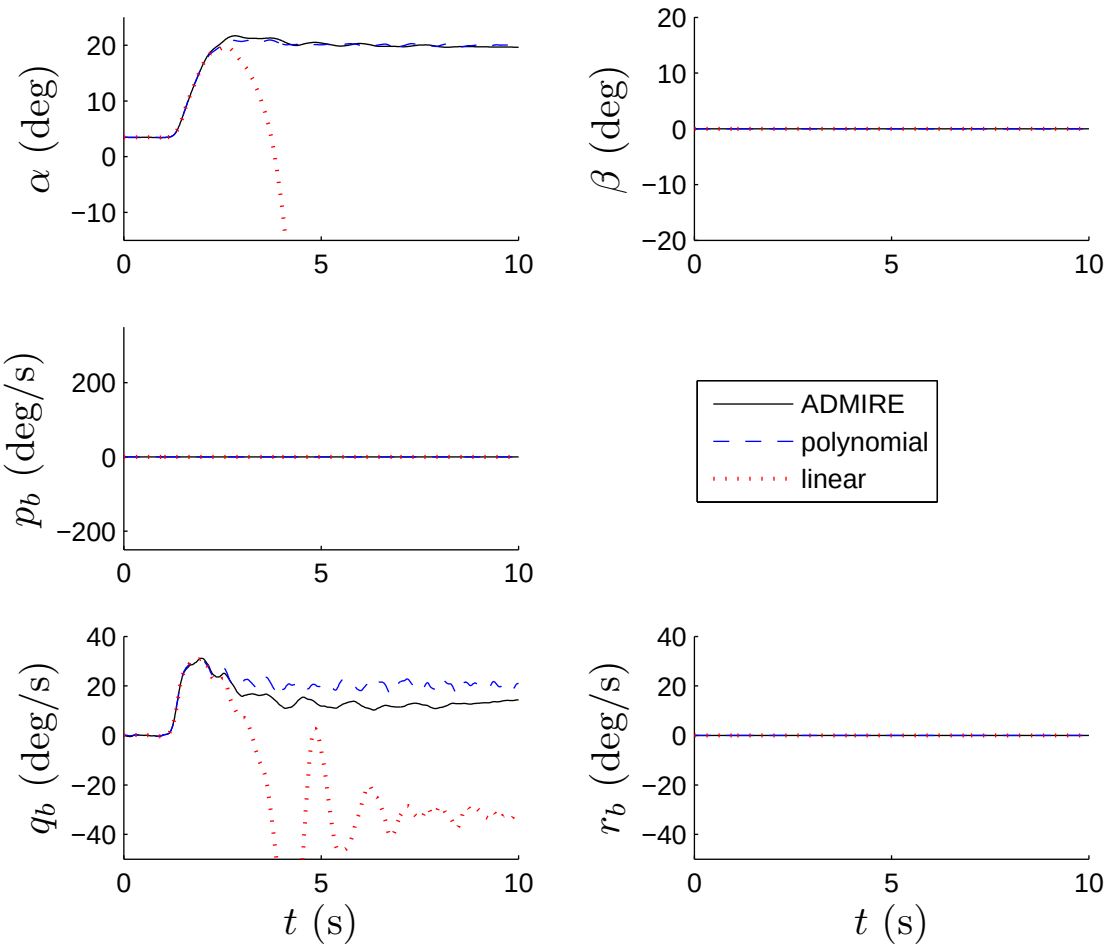


Figure 7.26: Comparison of pull-up manoeuvre state responses

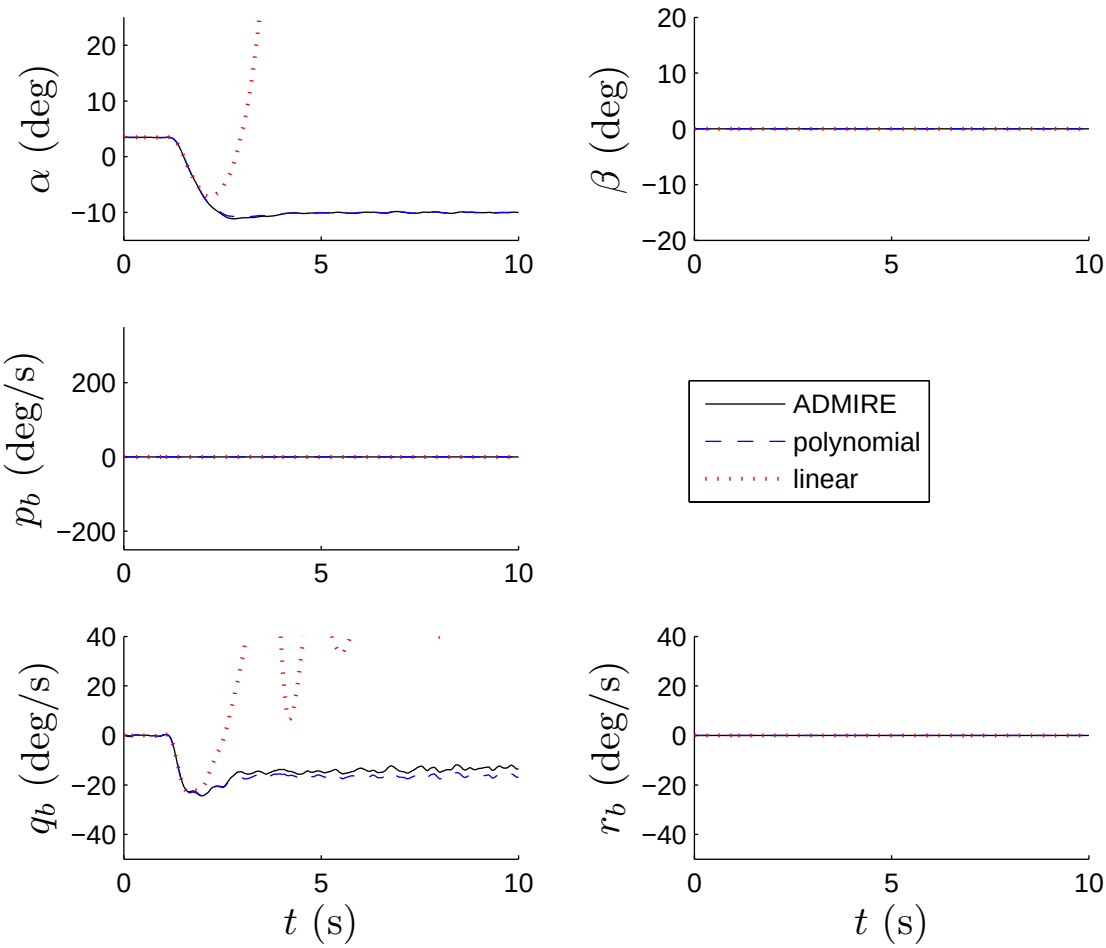


Figure 7.27: Comparison of push-over manoeuvre state responses

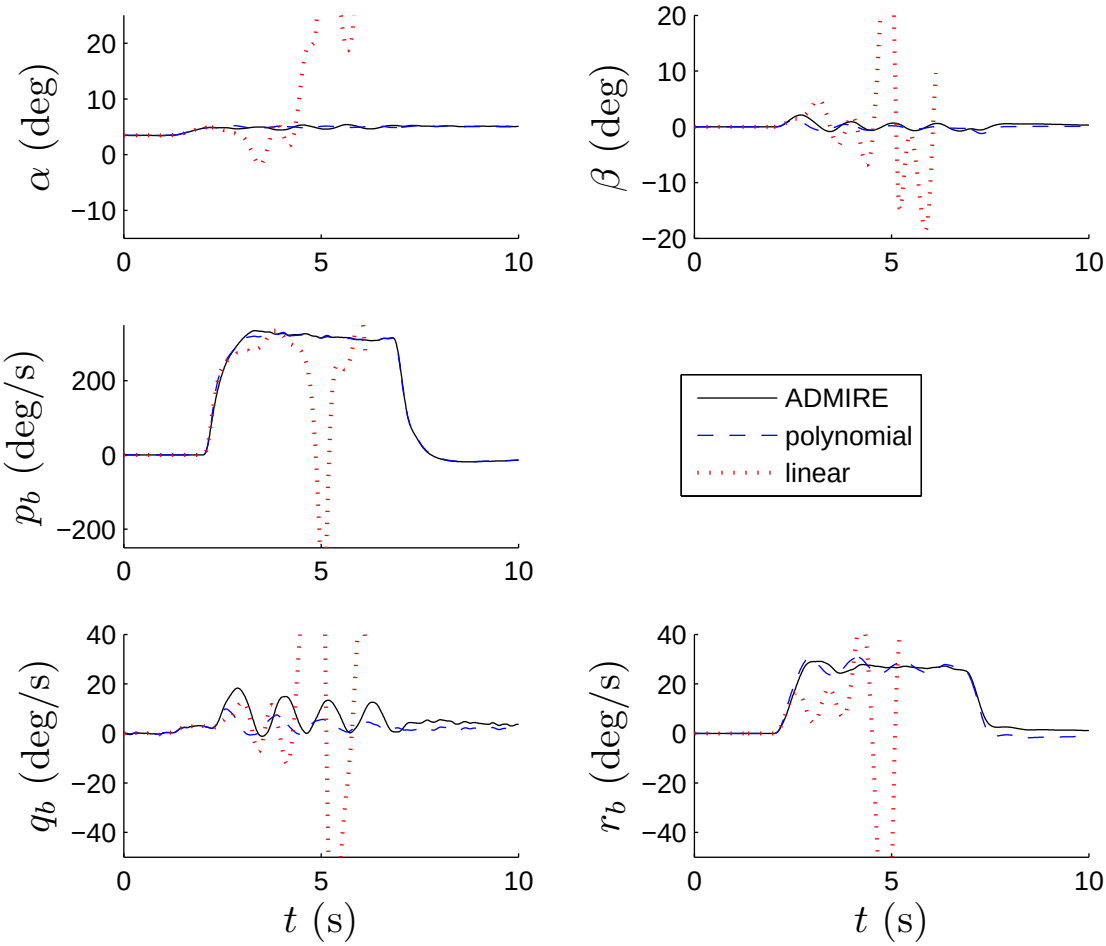


Figure 7.28: Comparison of roll manoeuvre state responses

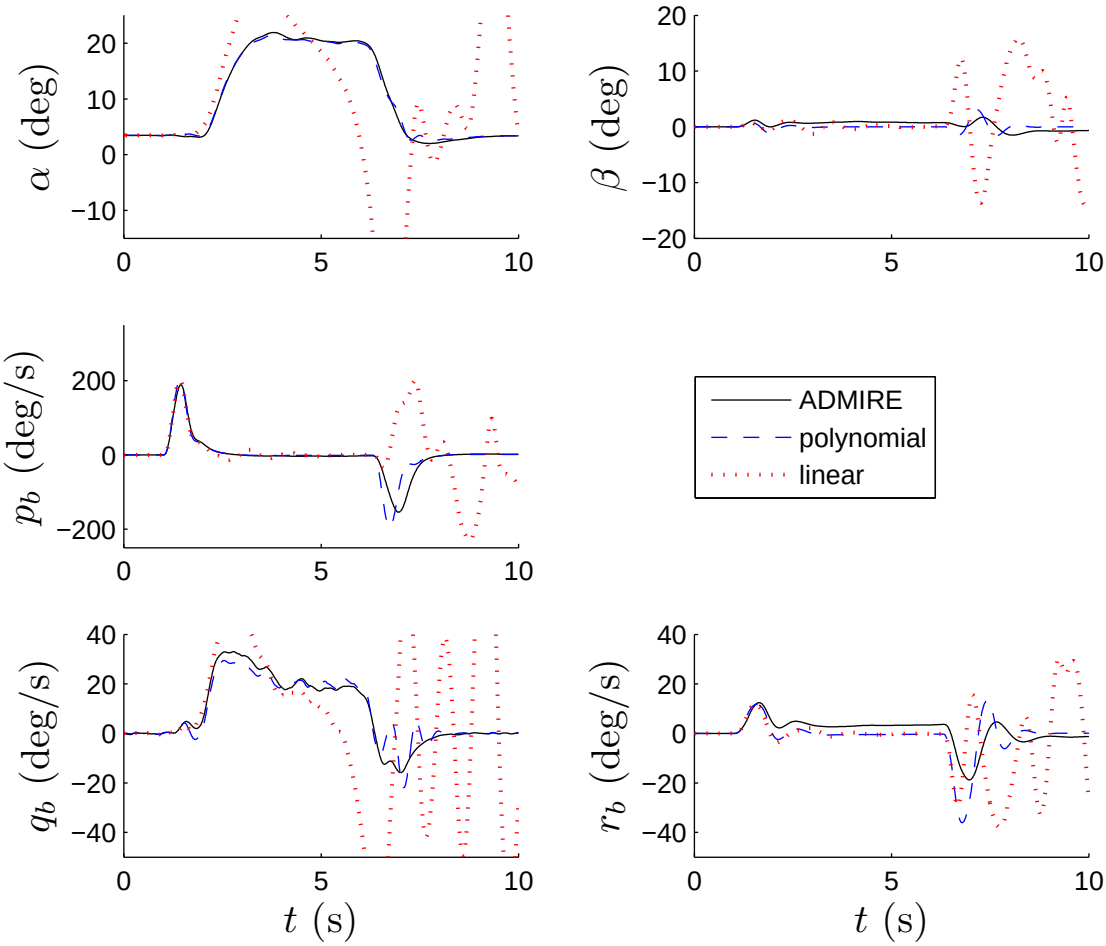


Figure 7.29: Comparison of turn manoeuvre state responses

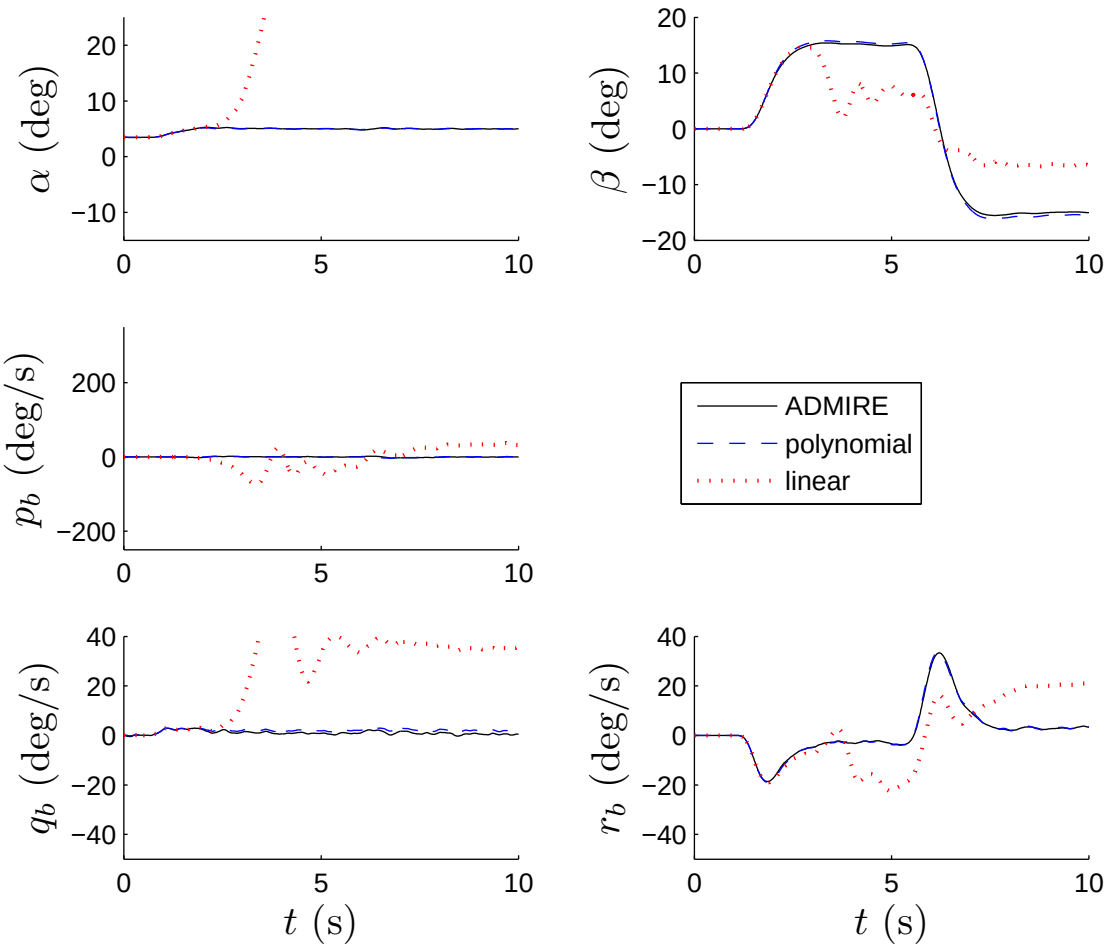


Figure 7.30: Comparison of sideslip manoeuvre state responses

In each manoeuvre, the closed loop using the polynomial model as the plant agrees well with the case where the actual ADMIRE is used. There is particularly close agreement between the polynomial model and ADMIRE for α, β and p_b. Agreement for q_b is not quite so good in the pull manoeuvre, when the angle of attack becomes large, and in the roll manoeuvre when p_b is large. There is a noticeable difference between the polynomial and ADMIRE body angle rates during the latter part of the turn manoeuvre, when elevons saturate. However, overall the polynomial model with the controller describes the closed-loop characteristics of ADMIRE well. In all these manoeuvres, the JL model (from the bundled trimming and linearisation routines) departs significantly from ADMIRE. As predicted, this

is especially noticeable in the last 3 manoeuvres (figs 7.28-7.30), where all 3 axes are perturbed. Agreement between the JL model and ADMIRE is better over a small timescale, when only the longitudinal axis is perturbed (figs 7.26-7.27).

Overall, these comparisons suggest that the polynomial model can provide meaningful stability analysis of the closed loop, particularly if we can guarantee that control surfaces do not saturate, although clearly we need some robustness guarantee to account for modelling inaccuracy.

This discussion motivates the next section, where we search for a domain of attraction (DA) of the origin of the closed-loop of ADMIRE, in which control surfaces do not saturate.

7.1.5 Domain of Attraction and Saturation

In this section we perform some numerical simulations with various initial conditions, in order to get an approximation of the domain of attraction (DA) of the closed loop system when our controller is applied to ADMIRE.

The definition of a DA is given in [12] as follows:

Given the autonomous system

$$\dot{x} = f(x) \tag{7.11}$$

suppose the state starts at x when $t = 0$ and trajectory $\phi(t, x)$ is the corresponding solution of (7.11). If the origin $x = 0$ is stable, then the domain of attraction (of the origin) is defined as the set of all initial conditions x such that the trajectory $\phi(t, x)$ is defined for all $t \geqslant 0$ and $\lim_{t\to\infty} \phi(t, x) = 0$ (i.e. the states asymptotically approach zero).

The corresponding definition of a robust domain of attraction is that the conditions stated above must hold for all allowed values of the uncertain parameters.

The following figures 7.31-7.35 show the DA of the closed-loop, using ADMIRE as the plant. This includes sensors, actuators and rate and position limits. The aim is to find initial conditions for $\alpha(0)$, $\beta(0)$ and $p_b(0)$ from which the controller recovers stability, i.e. $\begin{bmatrix}\alpha_c & \beta_c & \mu_c\end{bmatrix} = 0$. This is done by gridding over initial values

of α, β and p_b and recording if the resulting response was stable and if saturation of any control surface occurred. We also record whether the response stayed within the envelope $-12 < \alpha < 24°$, $-18 < \beta < 18°$, $-3.6 < n_z < 10.8$g. This envelope is actually 20% larger than that of the polynomial model, to allow for some overshoot at the beginning of the response.

In figs 7.31-7.35, green points indicate initial conditions from which the response stayed within the envelope described above and no saturation occurred. Orange points are likewise, but with saturation of at least one control surface. In figs 7.33-7.35 we give cross-sections for given initial values of β. The red points indicate stability, but the envelope is exceeded. Black points indicate instability.

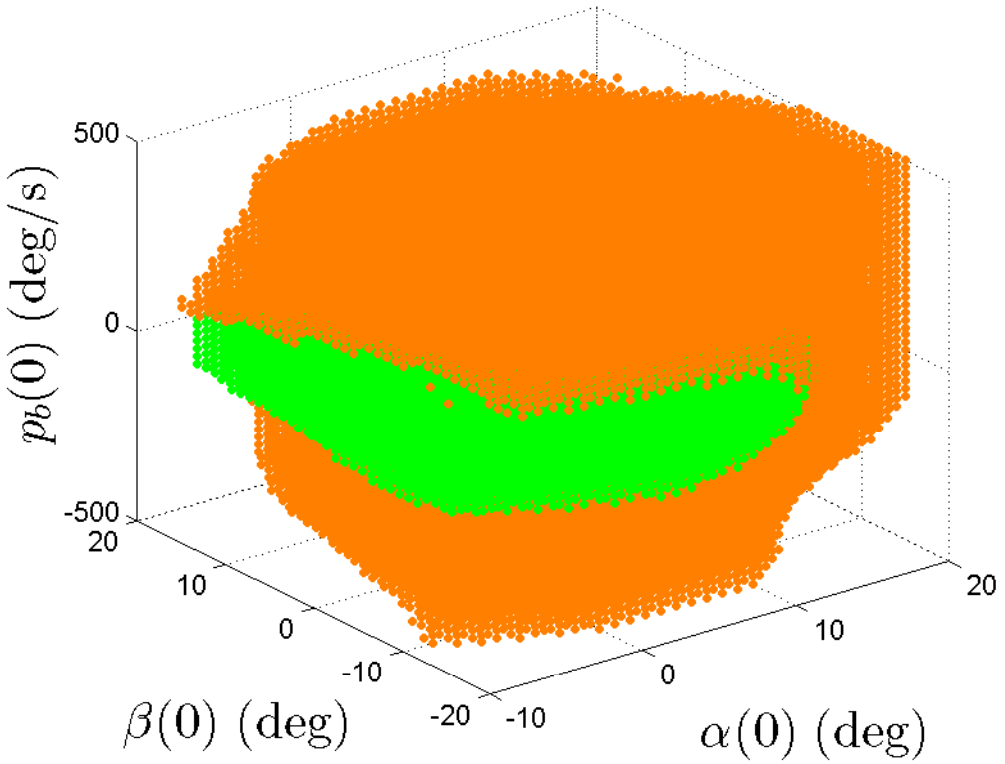


Figure 7.31: DA for the closed-loop. Green=stable, no saturation. Orange=stable, but at least one control surface saturates.

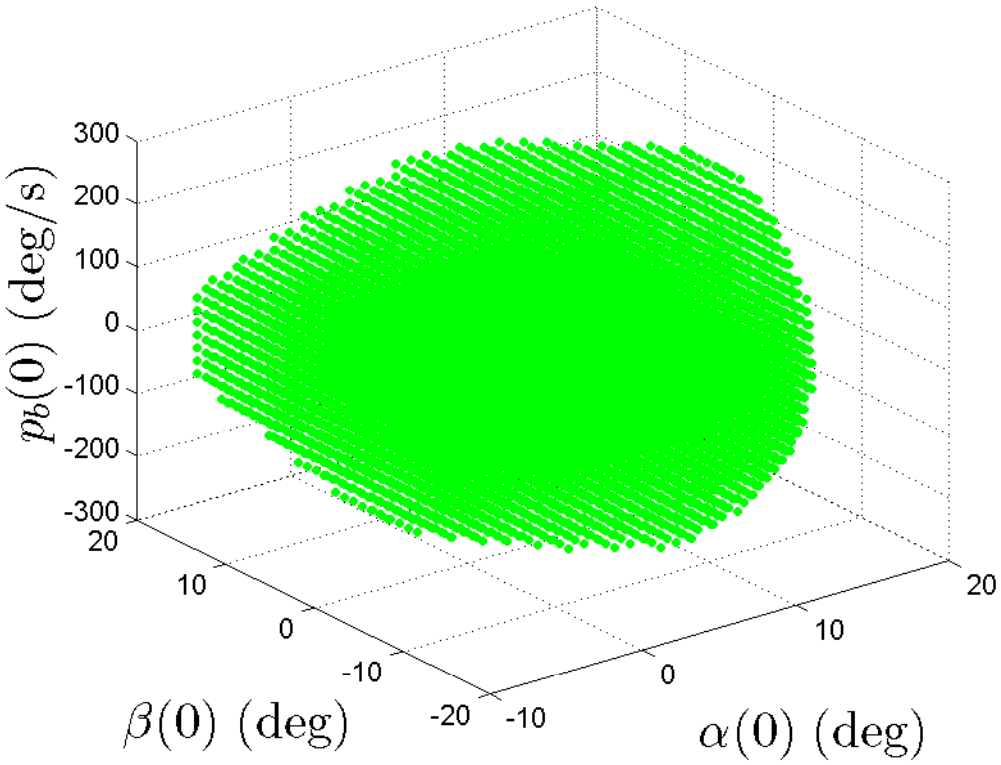


Figure 7.32: DA for the closed-loop. Green=stable, no saturation. Orange=stable, but at least one control surface saturates.

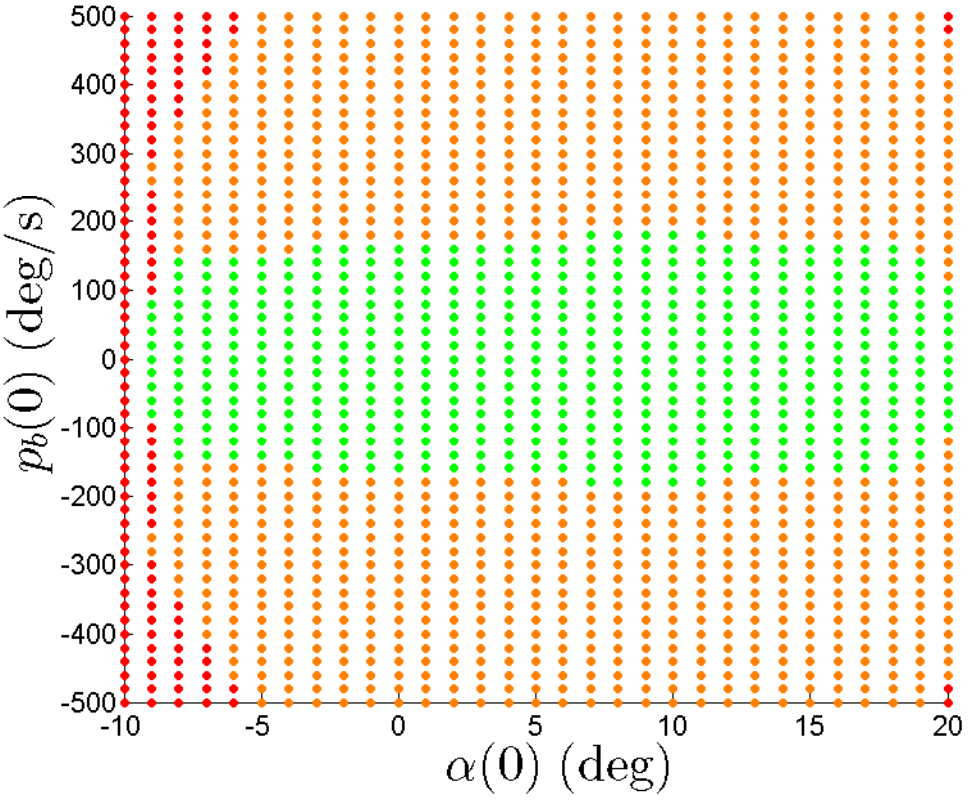


Figure 7.33: DA for the closed-loop, for $\beta(0) = 0$

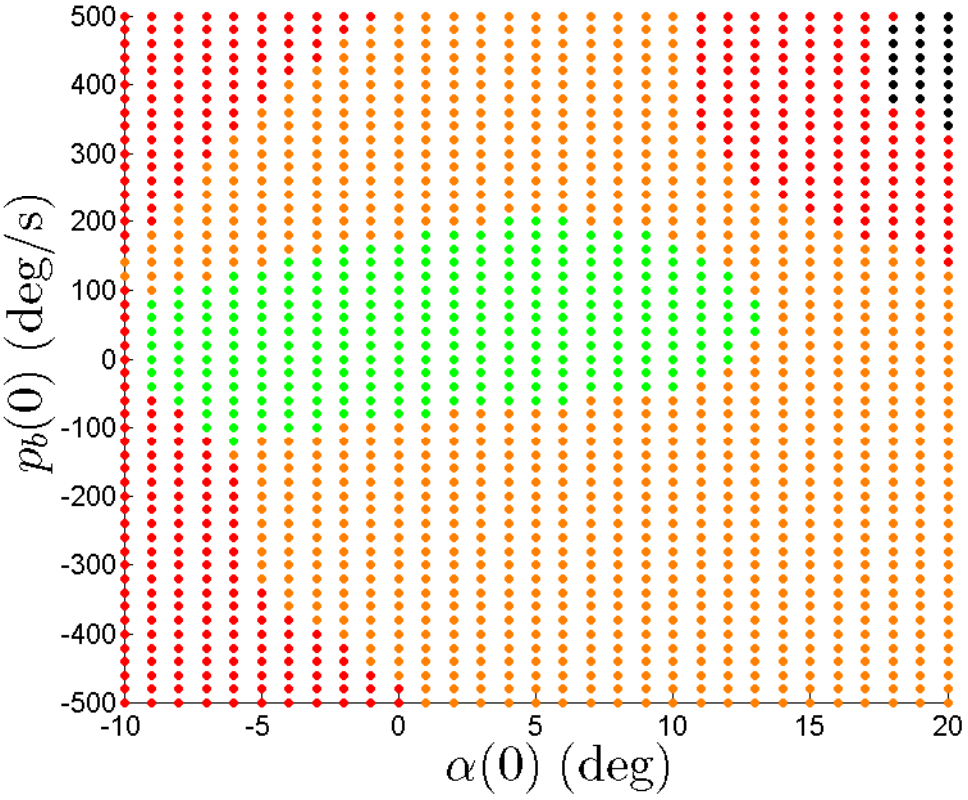


Figure 7.34: DA for the closed-loop, for $\beta(0) = 15°$

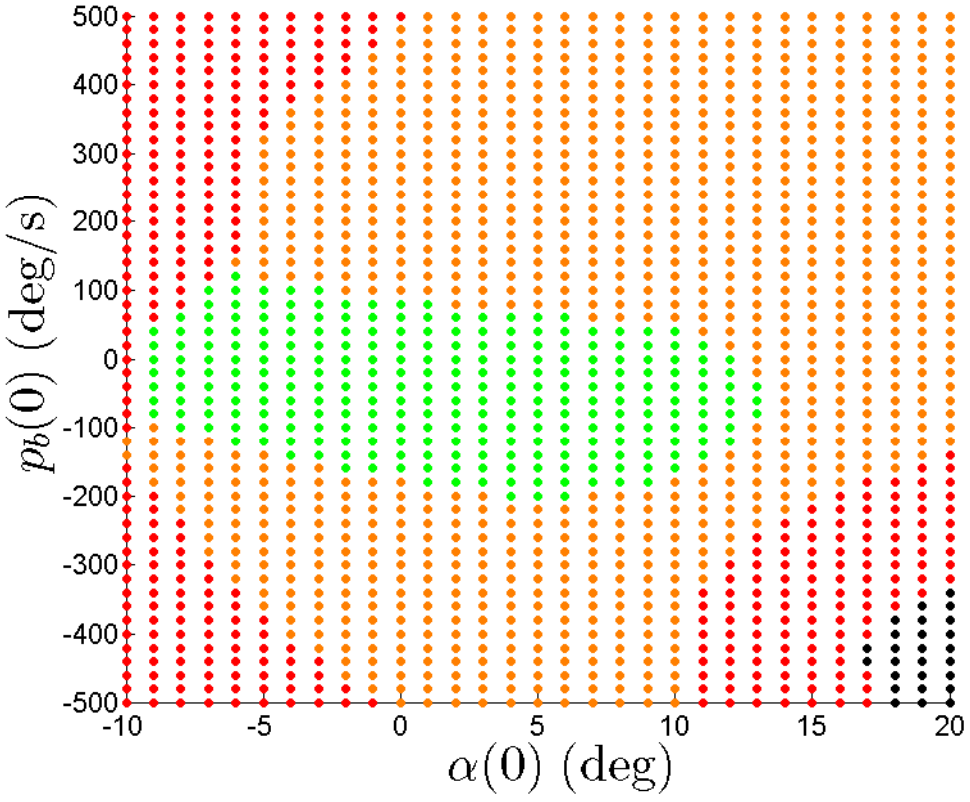


Figure 7.35: DA for the closed-loop, for $\beta(0) = -15°$

In the sequel, we will analyse robust stability of the closed-loop, assuming the plant can be given by the polynomial model with some TV parametric uncertainty. We analyse stability and robust stability using an LFT/LDI representation of the closed loop (similar to the representation used in previous chapters) and conservative LMI conditions to find a DA and RDA. We will also guarantee no saturation for initial conditions within the DA and RDA. The resulting conservative ellipsoidal domains of attraction can be compared with 7.32.

7.2 Closed-Loop Robust Stability Analysis

7.2.1 LFT Representation of the Uncertain Plant

The aim of this section is to provide a method for obtaining a domain of attraction (DA) and robust domain of attraction (RDA) for an uncertain rational system, controlled by FL-TSS. This will be applied to finding a RDA for the TSS controller on the polynomial model of ADMIRE, within which we can guarantee that control surfaces do not saturate.

Whereas the previous section used numerical simulations of the controller on ADMIRE to approximate the DA, in this section we search for an ellipsoidal estimate of the true DA using a quadratic Lyapunov function.

As stated in [12], a level set of a Lyapunov function provides an ellipsoidal inner estimate of the true domain of attraction, and may be very conservative.

The other difference between this and the previous section is that the technique used here requires that the closed-loop system admits an LFT representation, hence the controller is applied not to ADMIRE, but to the polynomial model derived from it.

Suppose a time-varying uncertain plant has the (non-unique) quasi-LPV representation

$$\dot{x}(t) = \tilde{\mathbf{A}}(x(t), \Psi(t))x(t) + \tilde{\mathbf{G}}(x(t), \Psi(t))u(t) \tag{7.12}$$

where $\Psi \in \mathbb{R}^{n_\Psi}$ is a vector of time-varying parameters representing parametric uncertainty in the model. We assume that $(x, u) = (0, 0)$ is an equilibrium of (7.12), for all allowed values of $\Psi(t)$.

We assume that the n_Ψ uncertain parameters are known *a-priori* to be bounded and can be normalised in the form

$$\Psi_i(t) = \bar{\Psi}_i + s_i \delta\Psi_i(t) \tag{7.13}$$

where $\bar{\Psi}$ is the nominal (or 'best guess') value of Ψ, which is given by $\bar{\Psi} = \frac{\Psi_{i,max}+\Psi_{i,min}}{2}$. The scaling constant is $s_i = \frac{\Psi_{i,max}-\Psi_{i,min}}{2}$. The normalised variation in each param-

eter is $\delta\Psi_i(t)$, and we will assume these all have the bound $|\delta\Psi_i(t)| \leq \sigma_w^{-1}$, where $\sigma_w > 0$ is a real scalar constant.

The nominal plant is therefore given by

$$\dot{x}(t) = \bar{\mathbf{A}}(x(t), \bar{\Psi})x(t) + \bar{\mathbf{G}}(x(t), \bar{\Psi})u(t) \tag{7.14}$$

Likewise, we make the *a-priori* assumption that each of the n_k states that appears in the QLPV matrices of (7.12) is bounded and can be written in the form

$$x_k(t) = s_k \delta x_k(t) \tag{7.15}$$

where we have assumed that each state is bounded symmetrically around 0, such that $|\delta x_k(t)| \leq \sigma_p^{-1}$. Note, bounding states is a strong assumption to make and needs to be verified in the later analysis. An important part of the later analysis is therefore to verify the bound $-s_k\sigma_p^{-1} < x_k < s_k\sigma_p^{-1}$ or $x_k^2 < s_k^2\sigma_p^{-2}$.

Assume the plant (7.12) admits an LFT, given by

$$\dot{x} = Ax + B_u u + B_p p + B_w w \tag{7.16a}$$

$$q = C_q x + D_{qu} u + D_{qp} p \tag{7.16b}$$

$$z = C_z x + D_{zu} u + D_{zp} p + D_{zw} w \tag{7.16c}$$

where $p = \Delta_p q$, $\|\Delta_p\| \leq \sigma_p^{-1}$, $\sigma_p > 0$ and similarly $w = \Delta_w z$, $\|\Delta_w\| \leq \sigma_w^{-1}$, $\sigma_w > 0$.

$\Delta_p(\delta x(t))$ contains the normalised variation in the states and $\Delta_w(\delta\Psi(t))$ contains the normalised variation in the uncertain parameters.

(7.16) with $w, z = 0$ is therefore an LFT representation of the nominal plant (7.14) (no uncertainty).

In addition, the control input u to the plant is not the output of the controller, but the output of the actuators (assumed LTI), described by

$$\dot{\xi} = A_\xi \xi + B_\xi u_c \tag{7.17a}$$

$$u = C_\xi \xi \tag{7.17b}$$

where u_c is the output of the controller and ξ is the actuator state vector.

7.2.2 Representing the TSS Controller in LFT Form

An idealised representation of the nominal plant neglects the actuator and splits the state vector into 'fast' and 'slow' parts: $x = \begin{bmatrix} \eta \\ \Omega \end{bmatrix} = \begin{bmatrix} C_\eta \\ C_\Omega \end{bmatrix} x$ where η is the slow state vector (wind axis angles) and Ω is the fast state vector (body axis angular rates).

Fast Subsystem The fast subsystem is an idealised model of the dynamics of states Ω. The control input is assumed to be equal to the commanded control signal, i.e. the actuator is neglected.

$$\dot{\Omega}_{ideal} = C_\Omega(\bar{\mathbf{A}}(x, \bar{\Psi})x + \bar{\mathbf{G}}(x, \bar{\Psi})u_c) \tag{7.18}$$

In LFT form

$$\dot{\Omega}_{ideal} = A_{21}\eta + A_{22}\Omega + G_2 u_c + B_\Omega p_\Omega \tag{7.19a}$$

$$q_\Omega = C_{\Omega q} \begin{bmatrix} \eta \\ \Omega \end{bmatrix} + D_{\Omega u} u_c + D_{\Omega p} p_\Omega \tag{7.19b}$$

$$p_\Omega = \Delta_\Omega(\delta x) q_\Omega \tag{7.19c}$$

$$\|\Delta_\Omega\| \leq \sigma_p^{-1} \tag{7.19d}$$

Leading to

$$u_c = -G_2^\dagger[A_{21}\eta + A_{22}\Omega + B_\Omega p_\Omega + K_1(\Omega - \Omega_c) + K_2\varepsilon] \tag{7.20}$$

where $\dot{\varepsilon} = \Omega - \Omega_c$.

Slow Subsystem Neglecting the control effect on the slow dynamics and instead treating the body angle rates as control inputs leads to:

$$\dot{\eta}_{ideal} = C_\eta \bar{\mathbf{A}}(x, \bar{\Psi}) \begin{bmatrix} \eta \\ \Omega_c \end{bmatrix} = \bar{\mathbf{A}}_{11}(\eta, \bar{\Psi})\eta + \bar{\mathbf{A}}_{12}(\eta, \bar{\Psi})\Omega_c \tag{7.21}$$

Note: This step requires the crucial assumption that the slow dynamics are affine in Ω, as well as the assumption that the fast states can be treated as control inputs equal to their commanded values.

In LFT form

$$\dot{\eta}_{ideal} = A_{11}\eta + A_{12}\Omega_c + B_\eta p_\eta \tag{7.22a}$$

$$q_\eta = C_{\eta q}\eta + D_{\eta\Omega}\Omega_c + D_{\eta p}p_\eta \tag{7.22b}$$

$$p_\eta = \Delta_\eta(\delta\eta)q_\eta \tag{7.22c}$$

$$\|\Delta_\eta\| \leq \sigma_p^{-1} \tag{7.22d}$$

Leading to

$$\Omega_c = -A_{12}^{\dagger}[A_{11}\eta + B_\eta p_\eta + K_3(\eta - r) + K_4\epsilon] \tag{7.23}$$

where $\dot{\epsilon} = \eta - r$ and r is the reference value of η.

The final controller is then given by substituting (7.23) into (7.20), associated with the LFT fictitious feedback signals defined by (7.19b), (7.19c) and (7.22b), (7.22c).

$$\begin{aligned} u_c = -G_2^{\dagger}\Big[&A_{21}\eta + (A_{22} + K_1)\Omega + B_\Omega p_\Omega + K_2\varepsilon \ \ldots \\ &\ldots + K_1 A_{12}^{\dagger}[A_{11}\eta + B_\eta p_\eta + K_3(\eta - r) + K_4\epsilon]\Big] \end{aligned} \tag{7.24}$$

The system interconnection is illustrated in figure 7.36.

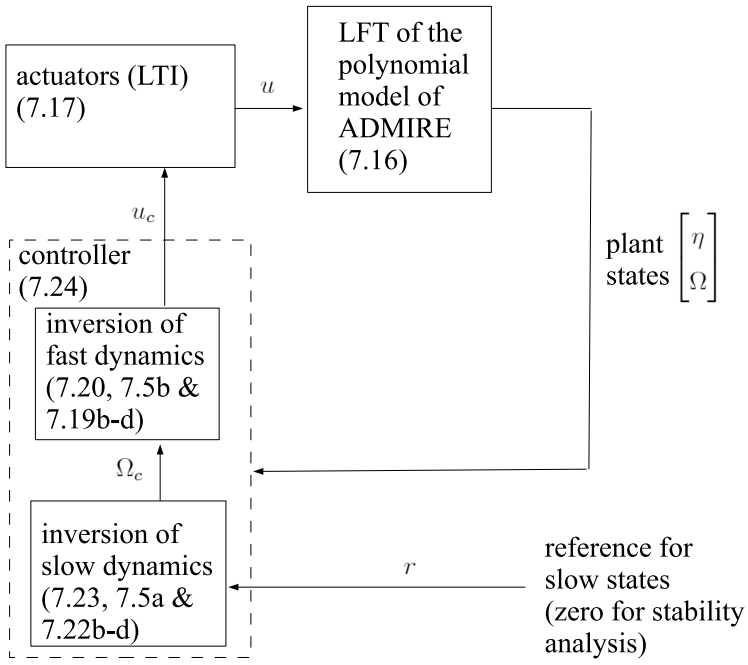


Figure 7.36: System interconnection for stability analysis in LFT form

7.2.3 Closing the control loop

Given that we want to find a robust domain of attraction (RDA) for the closed loop system, we set the external reference input $r = 0$ and close the control loop by substituting the controller (7.24) into (7.19b) and (7.17a) and substitute (7.23) into (7.22b). We also substitute the actuator output (7.17b) into the LFT equations of the plant (7.16).

We get a closed-loop LFT system of the form

$$\dot{\bar{x}} = \mathcal{A}\bar{x} + \mathcal{B}\bar{p} \tag{7.25a}$$

$$\bar{q} = \mathcal{C}\bar{x} + \mathcal{D}\bar{p} \tag{7.25b}$$

$$\bar{p} = \bar{\Delta}\bar{q} \tag{7.25c}$$

where $\bar{x} = \begin{bmatrix} x^T & \xi^T & \varepsilon^T & \epsilon^T \end{bmatrix}^T$, $\bar{q} = \begin{bmatrix} q^T & q_\eta^T & q_\Omega^T & z^T \end{bmatrix}^T$ and $\bar{p} = \begin{bmatrix} p^T & p_\eta^T & p_\Omega^T & w^T \end{bmatrix}^T$ and $\bar{\Delta} = diag(\Delta_p, \Delta_\eta, \Delta_\Omega, \Delta_w)$.

Note, $\Delta_p, \Delta_\eta, \Delta_\Omega$ all have the same norm bound: σ_p^{-1}. The norm bound on Δ_w is σ_w^{-1}.

The closed-loop matrices in (7.25) are given below:

$$\mathcal{A} = \begin{bmatrix} A & B_u C_\xi & 0 & 0 \\ -B_\xi \Phi & A_\xi & -B_\xi G_2^\dagger K_2 & -B_\xi G_2^\dagger K_1 A_{12}^\dagger K_4 \\ C_\Omega + A_{12}^\dagger (A_{11} + K_3) C_\eta & 0 & 0 & A_{12}^\dagger K_4 \\ C_\eta & 0 & 0 & 0 \end{bmatrix} \tag{7.26}$$

$$\mathcal{B} = \begin{bmatrix} B_p & 0 & 0 & B_w \\ 0 & -B_\xi G_2^\dagger K_1 A_{12}^\dagger B_\eta & -B_\xi G_2^\dagger B_\Omega & 0 \\ 0 & A_{12}^\dagger B_\eta & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \tag{7.27}$$

$$\mathcal{C} = \begin{bmatrix} C_q & D_{qu}C_\xi & 0 & 0 \\ [C_{\eta q} - D_{\eta\Omega}A_{12}^\dagger(A_{11}+K_3)]C_\eta & 0 & 0 & -D_{\eta\Omega}A_{12}^\dagger K_4 \\ C_{\Omega q} - D_{\Omega u}\Phi & 0 & -D_{\Omega u}G_2^\dagger K_2 & -D_{\Omega u}G_2^\dagger K_1 A_{12}^\dagger K_4 \\ C_z & D_{zu}C_\xi & 0 & 0 \end{bmatrix} \tag{7.28}$$

$$\mathcal{D} = \begin{bmatrix} D_{qp} & 0 & 0 & 0 \\ 0 & D_{\eta p} - D_{\eta\Omega}A_{12}^\dagger B_\eta & 0 & 0 \\ 0 & -D_{\Omega u}G_2^\dagger K_1 A_{12}^\dagger B_\eta & D_{\Omega p} - D_{\Omega u}G_2^\dagger B_\Omega & 0 \\ D_{zp} & 0 & 0 & D_{zw} \end{bmatrix} \tag{7.29}$$

where

$$\Phi = G_2^\dagger[C_\Omega A + K_1 C_\Omega + K_1 A_{12}^\dagger(A_{11}+K_3)C_\eta] \tag{7.30}$$

The matrices above represent the closed loop LFT, in feedback with with block $\bar{\Delta}$.

The final step we want to perform in deriving the closed loop system matrices is a simple transformation of $\bar{\Delta}$, which reorders the entries on the diagonal of $\bar{\Delta}$ such that like terms appear in a single block. Note that Δ_p, Δ_η and Δ_Ω may contain entries on their diagonals that represent the same scalar variable, although in blocks of different sizes. What we want to do is reorder the diagonal so that each scalar variable appears in a single block. The reason for doing this is more clear when we consider the quadratic forms that we will use to describe what we know about the $\bar{\Delta}$-block (we have a choice either to perform this transformation of the $\bar{\Delta}$-block or to include off-diagonal blocks in our scaling matrices, otherwise the analysis will be more conservative).

If we define $\Delta^\# = \mathcal{T}\bar{\Delta}$, where $\mathcal{T}$ is a simple invertible transformation matrix such that like parameters appear in single diagonal blocks of $\Delta^\#$, then our final closed-loop is described by

$$\dot{\bar{x}} = \mathcal{A}\bar{x} + \mathcal{B}\mathcal{T}^{-1}p^{\#} \tag{7.31}$$

$$\bar{q} = \mathcal{C}\bar{x} + \mathcal{D}\mathcal{T}^{-1}p^{\#} \tag{7.32}$$

$$p^{\#} = \Delta^{\#}q \tag{7.33}$$

Now, $\Delta^{\#} = diag(\Delta_1, \Delta_w)$, where Δ_1 belongs to the set

$$\mathbf{\Delta} := \{\text{diag}(\delta_1 I_{k_1}, \ldots, \delta_l I_{k_l}) : \|\Delta\| \leq \sigma_p^{-1},\ \delta_i \in \mathbb{R}\}$$

where $\sigma_p > 0$ and Δ_w belongs to a similar set, but with norm bound σ_w^{-1}.

Writing the closed-loop Δ-block this way separates the Δ-block of the closed-loop system into $\Delta_1(\delta x(t))$, which contains the normalised variation in the states, and $\Delta_w(\delta\Psi(t))$ which contains the normalised variation in the uncertain time-varying parameters, giving the extra terms that appear in the closed loop system dynamics due to poor knowledge of the aerodynamic coefficients. Hence when $\Delta_w(\delta\Psi(t)) = 0$, we have the *nominal* closed loop, i.e. the closed loop dynamics we would obtain if the aerodynamic coefficients were known exactly.

Note that for now, we are *assuming* $|\delta x_i(t)| \leq \sigma_p^{-1}$. We aim to find a robust domain of attraction for the closed loop system, the boundary of which will be a level set of a quadratic Lyapunov function. In order for this to be rigorous, we have to show that initial conditions on the level set satisfy $|\delta x_i(t)| \leq \sigma_p^{-1}$.

Explicitly separating the Δ-block into these two parts allows us to investigate the effect of increasing the size of the uncertainty on the size of the provable RDA, by decreasing the value of σ_w. We can accomplish this by fixing a value for σ_w which simultaneously sets a norm bound on the size of all the uncertain parameters, with an iterative search over σ_p to increase the largest initial value of the states.

Given what we know about the structure of $\Delta^{\#}$, we know that $p^{\#}$ and $\bar{q}$ are related by the quadratic forms

$$p^{\#T}\mathcal{X}\mathcal{S}p^{\#} \leq \bar{q}^T\mathcal{S}\bar{q} \tag{7.34}$$

$$p^{\#T}\mathcal{G}q - \bar{q}^T\mathcal{G}p^{\#} = 0 \tag{7.35}$$

where $\mathcal{S}$ and $\mathcal{G}$ belong to the sets [87]:

$$\mathbf{S} := \left\{\text{diag}(S_1, \ldots, S_l) : 0 < S_i \in \mathbb{R}^{k_i \times k_i}\right\}$$

$$\mathbf{G} := \{\text{diag}(G_1, \ldots, G_l) : G_i = -G_i^T \in \mathbb{R}^{k_i \times k_i}\}$$

respectively, similar to those used in chapters 3 and 5, and $\mathcal{X}$ is given by $\mathcal{X} = diag(\sigma_p^2 I, \sigma_w^2 I_{n_w})$.

We now give a condition for the set

$$\mathcal{E}_a := \{\bar{x} | \bar{x}^T P \bar{x} \leq a\} \tag{7.36}$$

to be a robust domain of attraction for the uncertain closed-loop system (7.31), which is a slight generalisation on work in [87] to the case of finding a robust domain of attraction:

Theorem 3. *If, for a given $\sigma_p > 0$ and $\sigma_w > 0$, $\exists\ P = P^T > 0$, $\mathcal{S} \in \mathbf{S}$ and $\mathcal{G} \in \mathbf{G}$ such that*

$$\begin{bmatrix} \mathcal{A}^T P + P\mathcal{A} + \mathcal{C}^T \mathcal{S} \mathcal{C} & P\mathcal{B}\mathcal{T}^{-1} + \mathcal{C}^T \mathcal{S} \mathcal{D} \mathcal{T}^{-1} + \mathcal{C}^T \mathcal{G} \\ \star & (\mathcal{D}\mathcal{T}^{-1})^T \mathcal{S} \mathcal{D} \mathcal{T}^{-1} - \mathcal{X}\mathcal{S} + (\mathcal{D}\mathcal{T}^{-1})^T \mathcal{G} - \mathcal{G}\mathcal{D}\mathcal{T}^{-1} \end{bmatrix} < 0 \tag{7.37}$$

simultaneously with n_k inequalities of the form

$$\begin{bmatrix} \frac{1}{a} s_k^2 \sigma_p^{-2} & e_k \\ \star & P \end{bmatrix} > 0 \tag{7.38}$$

hold, then the set $\mathcal{E}_a$ (7.36) is a robust domain of attraction for the closed-loop system (7.31) under approximate FL-TSS and the LFT is well-posed. Note, e_k is a row of the identity matrix of appropriate size, that each state that appears in the QLPV state matrices of the plant is can be given by $x_k = e_k \bar{x}$.

Proof. LMI (7.37) shows that $V(\bar{x}) = \bar{x}^T P \bar{x}$ is a Lyapunov function for the closed-loop, under the assumption that each state x_k that appears on the QLPV matrices of (7.12) is bounded by $x_k^2 < s_k^2 \sigma_p^{-2}$, as described in (7.15). This follows from [87] and is just a zero-input version of lemma 2 in chapter 5.

Now, by Schur complement (see Appendix D), each LMI (7.38) is equivalent to $P > 0$ and

$$P - e_k^T \frac{a}{s_k^2 \sigma_p^{-2}} e_k > 0 \tag{7.39}$$

$$\bar{x}^T P \bar{x} - \bar{x}^T e_k^T \frac{a}{s_k^2 \sigma_p^{-2}} e_k \bar{x} > 0 \tag{7.40}$$

$$\frac{1}{a} \bar{x}^T P \bar{x} - \frac{1}{s_k^2 \sigma_p^{-2}} \bar{x}_k^2 > 0 \tag{7.41}$$

$$\bar{x}_k^2 < s_k^2 \sigma_p^{-2} \frac{1}{a} \bar{x}^T P \bar{x} \tag{7.42}$$

Hence $\bar{x}_k^2 < s_k^2 \sigma_p^{-2}$ if $\bar{x}^T P \bar{x} \leq a$, which verifies the bound described in (7.15), for values of $\bar{x}$ in $\mathcal{E}_a$. If the closed-loop system (7.31) has initial conditions that lie in $\mathcal{E}_a$, then $\mathcal{E}_a$ contains all future trajectories because (7.37) guarantees that $\dot{V} < 0$ for all allowed values of $\Psi(t)$. $\mathcal{E}_a$ is therefore a robust domain of attraction for (7.31).

Proof of well-posedness follows from [87] (see also appendix A) □

If Theorem 3 is satisfied, then we can conclude that the system is robustly stable under FL-TSS control, despite the fact that FL-TSS is only performed approximately, due to time-varying parametric uncertainty and the fact that the slow/fast subsystems used for control design are only approximate, idealised models of the plant dynamics.

Corollary - Output bounds and actuator position limits

Given a single output $z = C_z \bar{x}$ (where C_z is a row vector) of the closed-loop system, we can also bound the magnitude of z, using LMIs similar to (7.38). If the LMIs in Theorem 3 are feasible, together with

$$\begin{bmatrix} \frac{1}{a} z_{max}^2 & C_z \\ \star & P \end{bmatrix} > 0 \tag{7.43}$$

then, following a proof similar to the one given above, $|z(t)| < z_{max}$ in $\mathcal{E}_a$.

As the actuator states ξ are states of the closed-loop system, this allows us to verify that actuator position limits are obeyed in $\mathcal{E}_a$.

This was noted in [87] for the case of state feedback control of the form $u = Kx$. Explicitly including the actuator dynamics in the closed loop LFT matrices allows

us to verify that control surfaces do not saturate for the FL-TSS controller proposed here.

7.2.4 Application to ADMIRE and Robust Domain of Attraction

This section aims to assess the robust stability of the polynomial model of ADMIRE found in chapter 6, under the FL-TSS control designed in the previous section.

We first write the polynomial model of ADMIRE (6.9) and (7.7) in a control-affine form of (7.14), which has state vector

$x = \begin{bmatrix} \tilde{\alpha} & \beta & \mu & p_b & q_b & r_b \end{bmatrix}^T$,

using standard rational approximations to the trigonometric functions in (7.7) and control vector

$u = \begin{bmatrix} \delta_n & \delta_{ei} & \delta_{ey} & \delta_{ai} & \delta_{ay} & \delta_r \end{bmatrix}^T$

This is given in appendix F.

As we want to assess robust stability, we describe the plant as a system with parametric uncertainty (7.12), where we allow independent uncertainty on each row of the state equations. We have independent time-varying uncertainties:

$$\Psi(t) = \begin{bmatrix} \Psi_1(t) & \Psi_2(t) & \Psi_3(t) & \Psi_4(t) & \Psi_5(t) & \Psi_6(t) \end{bmatrix}^T \tag{7.44}$$

and write the uncertain plant as

$$\dot{x}(t) = Z(\bar{\mathbf{A}}(x(t))x(t) + \bar{\mathbf{G}}(x(t))u(t)) \tag{7.45}$$

with

$$Z = diag(\Psi_1(t), \Psi_2(t), \Psi_3(t), \Psi_4(t), \Psi_5(t), \Psi_6(t)) \tag{7.46}$$

Ψ is normalised such that

$\bar{\Psi} = \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}^T$,

Ψ_{1-3} have min/max values 1 ± 0.1 and Ψ_{4-6} have min/max values 1 ± 0.5. Hence, we have a greater level of uncertainty (50%) on the body axis roll accelerations than on the wind axis angular rates (10%).

Note that $\tilde{\alpha}$, β, p_b, q_b and r_b all appear in the QLPV state matrices (in appendix F), hence these elements on the state matrices need to be normalised. We normalise these such that their nominal values are zero and

$$\tilde{\alpha} \in [-13.5, 13.5^\circ] \tag{7.47}$$

$$\beta \in [-15, 15^\circ] \tag{7.48}$$

$$p_b \in [-100, 100^\circ/\text{s}] \tag{7.49}$$

$$q_b \in [-50, 50^\circ/\text{s}] \tag{7.50}$$

$$r_b \in [-50, 50^\circ/\text{s}] \tag{7.51}$$

Note, given that the trim value of α is approximately 3.5° at Mach 0.5 and altitude 4000 m, the symmetrical normalisation around zero of $\tilde{\alpha}$ actually allows for an asymmetrical variation of $\alpha \in [-10, 17^\circ]$.

Following this, we find LFTs for the perturbed plant described above and for the idealised fast and slow subsystems, as described by (7.19) and (7.22), using the LFR Toolbox [13]. Note that for the slow subsystem, we neglect the control effect.

The resulting Δ-blocks are given below:

$$\Delta_p = diag(\delta\alpha I_8, \delta\beta I_{10}, \delta p_b, \delta q_b I_2, \delta r_b) \tag{7.52a}$$

$$\Delta_w = diag(\delta\Psi_1, \delta\Psi_2, \delta\Psi_3, \delta\Psi_4, \delta\Psi_5, \delta\Psi_6) \tag{7.52b}$$

$$\Delta_\eta = diag(\delta\alpha I_3, \delta\beta I_5) \tag{7.52c}$$

$$\Delta_\Omega = diag(\delta\alpha I_4, \delta\beta I_3, \delta p_b, \delta q_b I_2, \delta r_b) \tag{7.52d}$$

Note, the length of $\Delta_p \in \mathbb{R}^{22\times22}$ (the Δ-block of the nominal plant) is longer than that of $diag(\Delta_\eta, \Delta_\Omega) \in \mathbb{R}^{19\times19}$ (the Δ-block of the controller), because the latter is based on an idealised model that neglects the control effect on the slow states and hence includes fewer nonlinear terms.

The control surface variables do not appear in any of (7.52), because the system is (modelled as) control-affine. Likewise, δp_b, δq_b and δr_b do not appear in Δ_η, because the slow subsystem is (modelled as) affine in the fast states Ω.

Next, we describe the LFT of the closed-loop system with zero reference input in terms of the original plant and idealised subsystem matrices, and the control gains found in the previous section, as described in (7.26)-(7.29).

We then transform the closed-loop $\bar{\Delta}$ to put like terms in single diagonal blocks, resulting in a closed-loop LFT for robust stability anaysis as in (7.31). The closed-loop LFT state matrix is given in appendix G, as are the Lyapunov matrix results.

The closed loop system has 15 states $\bar{x}$: 6 from the original plant model x, 6 from the actuators ξ, and 3 ϵ are the integral tracking errors on the slow states (we set $K_2 = 0$ in the previous section, so there is no integral tracking error state ε for the fast states):

$\bar{x} = \begin{bmatrix} x^T & \xi^T & \epsilon^T \end{bmatrix}^T$.

We apply Theorem 3 to the closed loop, together with 6 LMIs of the form (7.43), to verify that each control surface is below 25°, to avoid saturation. Note, each row vector C_z in (7.43) is a row of the control transformation matrix T, given in appendix E (we do not need to use the second row of T as well as the first, as we have chosen only to use symmetric canard deflection $\delta_{rc} = \delta_{lc}$).

As the Lyapunov matrix P is a variable, we arbitrarily set $a = 1$ in (7.38) and (7.43).

We minimise trace(P), subject to the conditions of Theorem 3, in order to increase the size of the ellipsoid $\mathcal{E}_a$ by maximising its semi-axes lengths. This is a convex minimisation problem [86].

7.2.5 RDA Results

Initially, we set $\sigma_w^2 = 1$ and iteratively find the smallest value of σ_p^2 such that the LMI problem is feasible, with the maximum size of uncertainty in the plant model described earlier. The ellipsoidal RDAs shown in figures 7.37 and 7.38 are respectively for intial conditions $\tilde{\alpha}(0)$, $\beta(0)$, $p_b(0)$ (other initial states zero) and for longitudinal-only initial conditions $\tilde{\alpha}(0)$, $q_b(0)$ (other initial states zero).

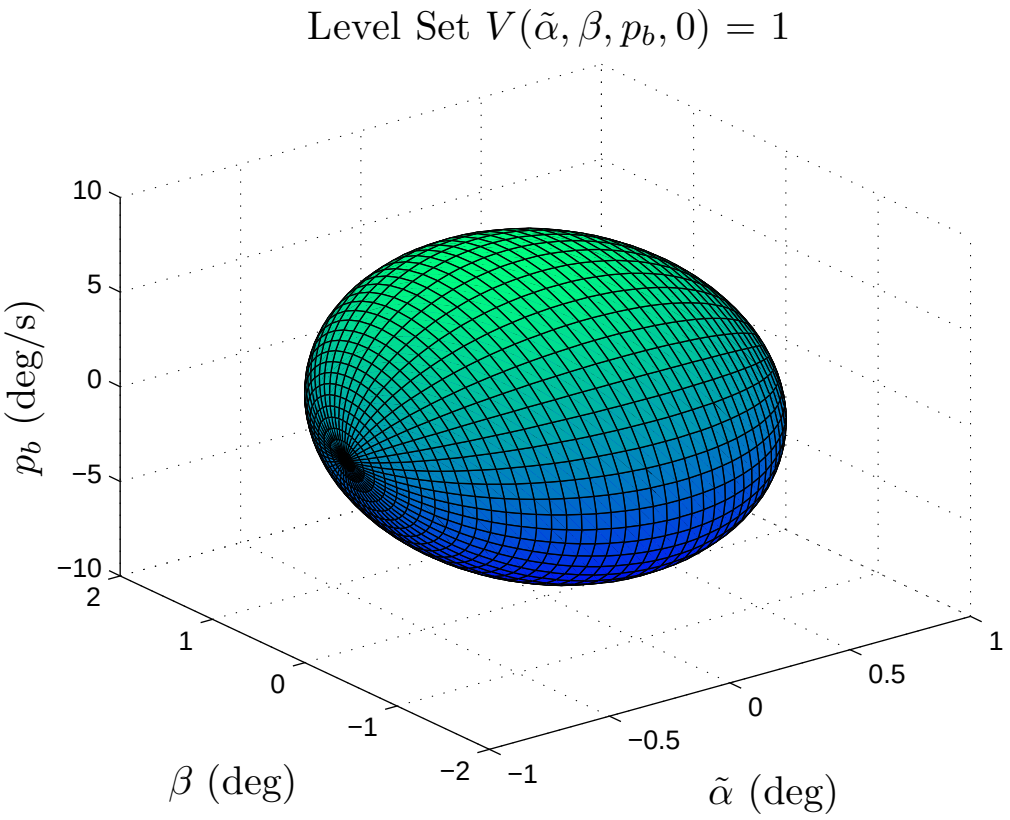


Figure 7.37: RDA for the closed-loop, for $\sigma_p^2 = 73$, $\sigma_w^2 = 1$

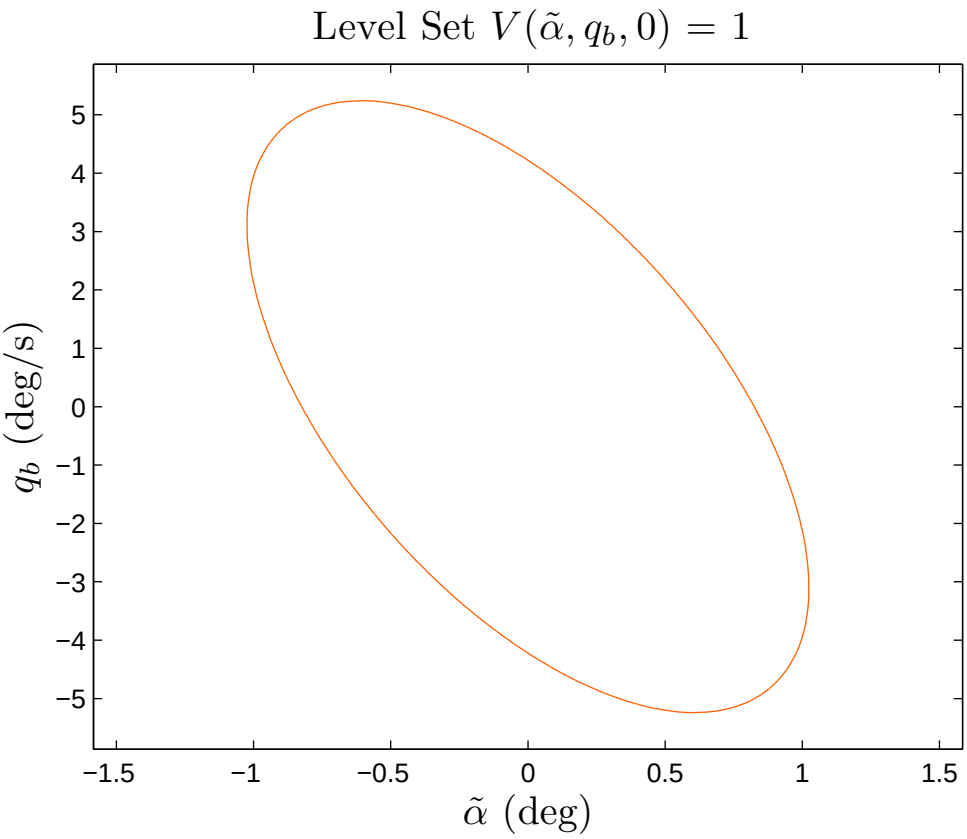


Figure 7.38: Longitudinal RDA for the closed-loop, for $\sigma_p^2 = 73$, $\sigma_w^2 = 1$

Next, in order to investigate the effect of decreasing the size of the parametric uncertainty on the size of RDA we can get, we set $\sigma_w^2 = 4$ and again iteratively find the smallest possible value of σ_p^2. Note, this time we can get a smaller σ_p^2 and a correspondingly larger RDA, which is shown in figures 7.39 and 7.40.

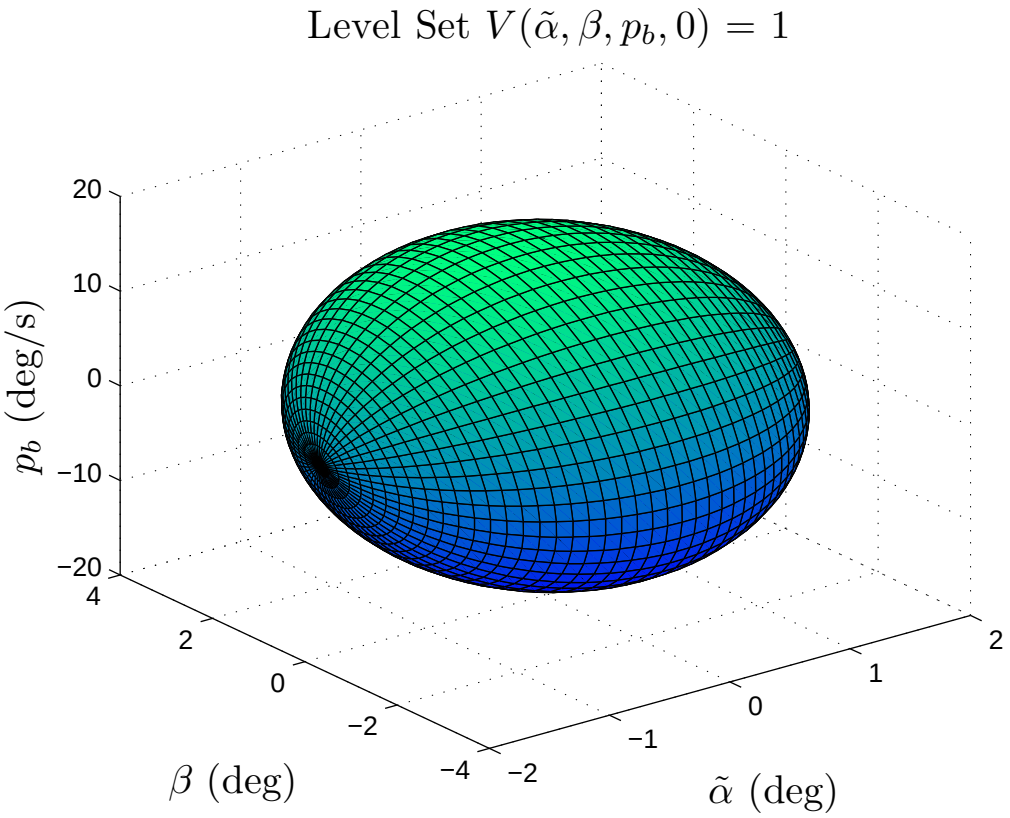


Figure 7.39: RDA for the closed-loop, for $\sigma_p^2 = 20$, $\sigma_w^2 = 4$

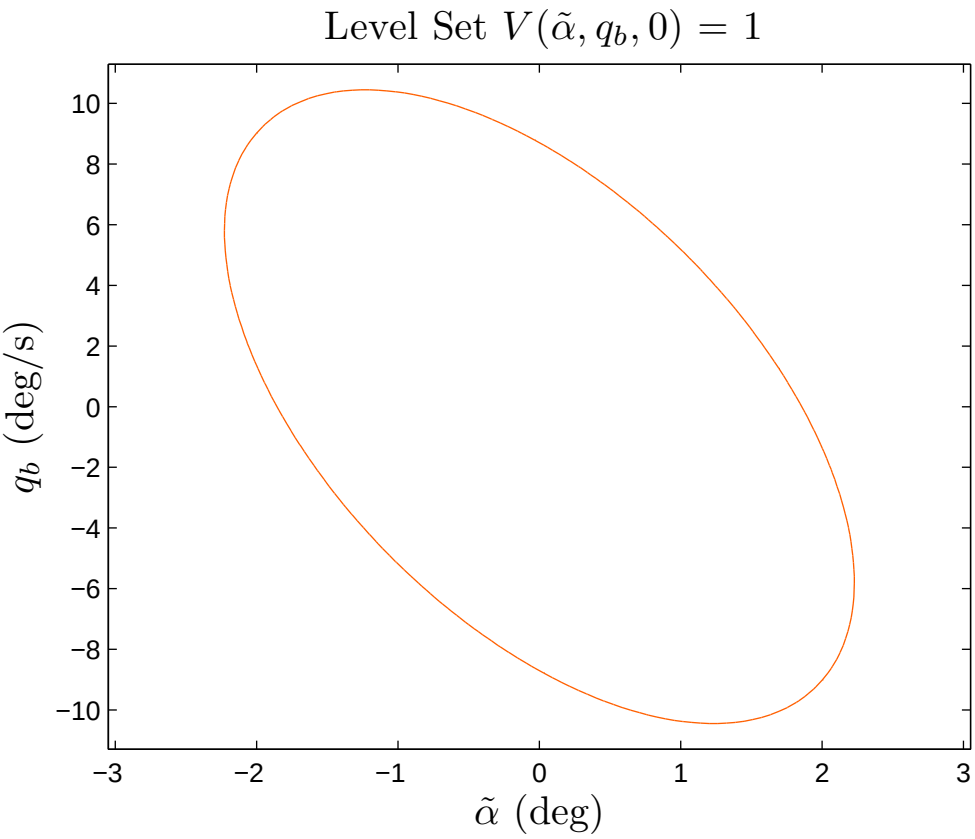


Figure 7.40: Longitudinal RDA for the closed-loop, for $\sigma_p^2 = 20$, $\sigma_w^2 = 4$

Finally, in order to find a DA with no uncertainty in the model, we truncate the LFT matrices to eliminate the block Δ_w completely, which is shown in figures 7.41 and 7.42. Note we find a yet smaller value of σ_p^2 and a correspondingly larger DA. However, this DA only applies to the nominal model with our FL-TSS controller.

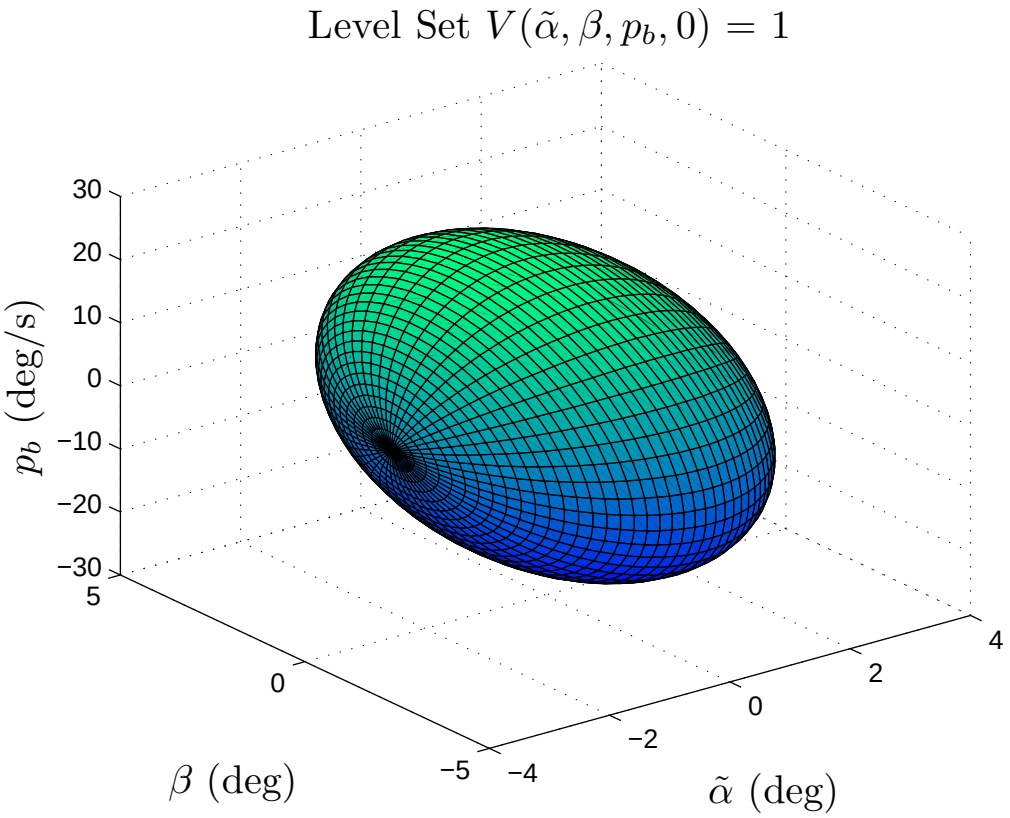


Figure 7.41: DA for the closed-loop, for $\sigma_p^2 = 9.73$

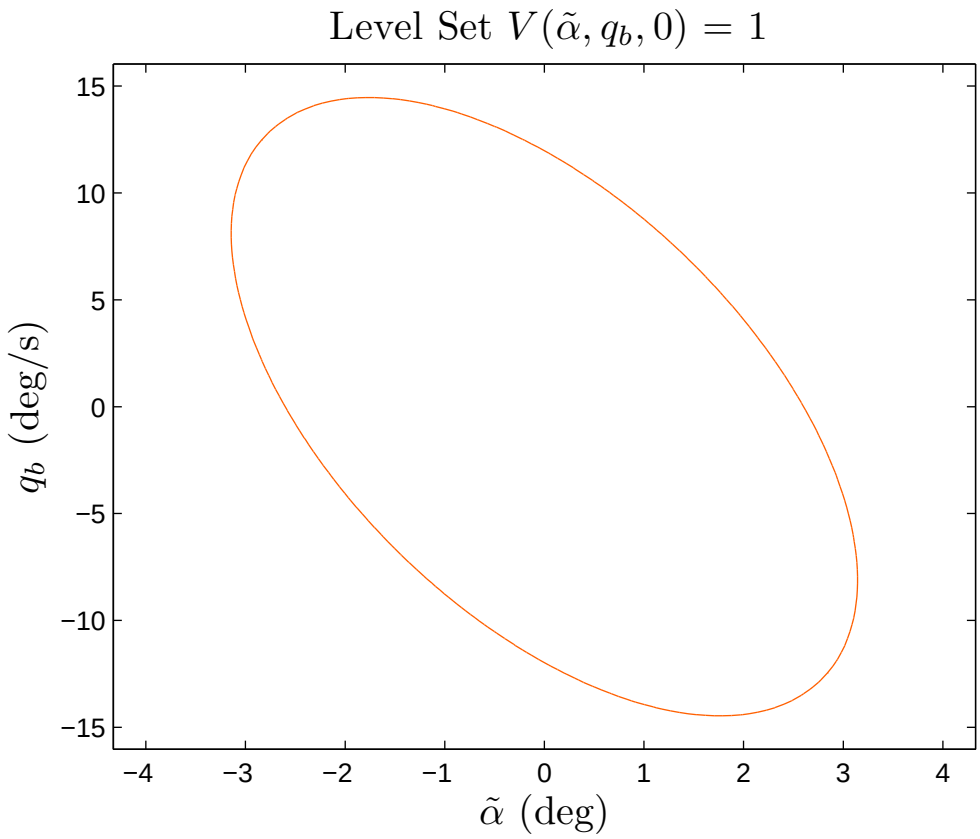


Figure 7.42: Longitudinal DA for the closed-loop, for $\sigma_p^2 = 9.73$

7.3 Concluding Remarks

In this chapter we have designed a FL-TSS controller that achieves approximate input-output linearisation for the short-period dynamics of ADMIRE. The controller was tuned using a genetic algorithm that optimises the performance of the system in a series of aggressive manoeuvres.

The most important part of this chapter is that we have developed a technique for analysing robust stability of a nonlinear rational system under FL-TSS control, with time-varying uncertainty. We find a robust domain of attraction for the closed loop system. For initial conditions in the RDA, we can guarantee that actuator position limits are not violated, and that the *a-priori* bounds set on part of the state vector hold, hence the analysis is rigorous. This has been applied to the ADMIRE benchmark model and we have investigated the effect of changing the

size of uncertainty on the plant model due to poor knowledge of the aerodynamic coefficients.

However, the domains of attraction found are fairly small and do not cover the entire allowed envelope, as the analysis is very conservative. We have only been able to search for quadratic Lyapunov functions to prove closed loop stability using the analysis outlined here; in general, a Lyapunov function for a nonlinear system does not have to be quadratic. More recent sum-of-squares techniques [183] [184] may be a fruitful method of reducing conservatism, as they are able to search for higher-order Lyapunov functions.

The analysis presented here is also conservative in that it does not set a bound on the rate of variation of parameters on the Δ-block. The IQC methodology [156] [147] provides a way of setting *a-priori* bounds on parameter rates, which may reduce conservatism, together with a bank of quadratic forms to represent other Δ-block descriptions.

Our analysis here also does not guarantee that actuator *rates* are not violated, hence extensive simulation is still an important tool in analysis. Although we did not include sensors in the analysis in this chapter, they can in principle be included as LTI blocks in the closed loop LFT matrices.

Chapter 8

Conclusions and Future Research

This thesis has investigated methods for providing robust stability guarantees for uncertain nonlinear systems controlled using feedback linearisation/ dynamic inversion. Since feedback linearisation was first proposed as a control technique, it has undergone a surge in popularity, particularly in the field of aerospace.

Although there has been a huge amount of research on FL, dealing with systems that are uncertain or include external disturbances, rigorous robust stability guarantees have not been found. The literature on FL, whether in aerospace or otherwise, has tended to focus on linear methods of robust stability analysis, which are not rigorous for nonlinear systems, or on simulation to assess robustness.

In this thesis, we have successfully applied a combination of robust and nonlinear systems analysis for rational systems in order to provide robust stability guarantees for the closed loop. This is the first time these techniques have been applied to FL systems, as there has been very little overlap between the field of FL research and the field of robust and nonlinear systems analysis.

We have been able to assess robust stability of the closed loop in the case where not all states are available for feedback, using LMI filter synthesis to estimate unknown states and conservative LMI conditions to assess robust stability of the closed loop.

With ADMIRE, we have first applied dynamic inversion to an LPV model of the system, in order to assess robust stability over the whole flight envelope. As usual

with dynamic inversion, the controller was designed on the assumption that it has access to all the states, however in practice only sensed values are available. We have been able to assess robust stability of the LPV model, including sensor and actuator dynamics in the loop and time-varying parametric uncertainty that represents poor knowledge of the aerodynamic force and moment coefficients, something which is usually assessed in the literature by simulation alone or by using linear techniques.

The latter part of the thesis has dealt with obtaining a nonlinear polynomial model of ADMIRE which better represents the system than the LPV model. It has been shown that this model closely matches the actual dynamics of ADMIRE and in particular is far better than a linear model in cases where all three axes are perturbed. Feedback linearisation has been applied to this model in order to assess robust closed loop stability. Of particular interest is the fact that we have been able to find a robust domain of attraction for the closed loop. This provides a region of safe initial conditions wthin which we can guarantee that the system is stable. Additionally, we are able to guarantee that the actuators do not saturate within the domain of attraction, which makes our robust stability analysis more rigorous.

Although we have been able to find LMI methods to analyse robust stability, these are conservative methods. The techniques presented in this thesis do not set a bound *a-priori* on the rate of variation of parameters that appear on the Δ-block, though in reality uncertainties and parameters such as Mach and Altitude could be considered as slowly time-varying, when considering short period dynamics. Also, we have only searched for quadratic Lyapunov functions for rational nonlinear systems, which increases the conservativeness of the analysis.

We have been able to obtain rigorous robust stability analyses for system models under approximate feedback linearisation, in the presence of time-varying uncertainty, actuator dynamics and output feedback either via sensors or via state estimation using a filter. In addition, we have been able to make the stability analysis more rigorous by finding a RDA within which we can guarantee that actuators do not saturate. Analysis has been applied to both LPV models and quasi-LPV models of rational nonlinear systems. These have been so far missing from the literature on

FL.

However, the work on proving robust quadratic stability in the previous chapter applies to a polynomial model, not to the ADMIRE itself. More importantly, none of the analyses presented here should be considered as replacement for extensive simulation and real-world testing. All of our analyses apply to approximate mathematical models of nonlinear systems, rather than the actual system itself, hence any rigorous robust stability guarantees obtained apply to the model, rather than to a real-world application. Not only that, but the analyses presented here are not sufficient to guarantee good performance of the closed loop to all design requirements, and are mainly aimed at robust stability.

Hence, there are several directions that future research can take:

- Reducing conservativeness of the analysis.
- Accounting for other types of nonlinearity
- Rate as well as position limits of actuators
- Improving robust performance guarantees
- Improving state estimation (observer) design in connection with FL

Appendix A

Linear Fractional Transformation and well-posedness

A linear fractional transformation (LFT) may be used to represent a system as a mapping from x to $\dot{x}$. The linear fractional transformation (LFT) matrix function is given by [126] :

$$\mathcal{F}_u(M, \Delta) := M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12} \quad \text{(upper LFT)} \tag{A.1}$$

$$\mathcal{F}_l(M, \Delta) := M_{11} + M_{12}\Delta(I - M_{22}\Delta)^{-1}M_{21} \quad \text{(lower LFT)} \tag{A.2}$$

where M is a partitioned matrix:

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \tag{A.3}$$

An LFT is well-posed, provided the inverse term is well-defined, i.e.

$$\det\left(I - M_{11/22}\Delta\right) \neq 0 \tag{A.4}$$

From now on, we will only be concerned with lower LFTs, unless otherwise stated.

Define the system matrix:

$$H = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \tag{A.5}$$

Where the system is in the form:

$$\dot{x} = Ax + Bp \tag{A.6a}$$

$$q = Cx + Dp \tag{A.6b}$$

$$p = \Delta q \tag{A.6c}$$

so that:

Ax contains the terms linear in the states and Bp is used to describe the nonlinear and parameter-varying terms. Δ is a diagonal matrix. The terms on the diagonal are the states in which the system is nonlinear, and the time-varying parameters.

If the system can be written in the form (A.6), then it can be written as the (lower) LFT of H, with a block Δ, where Δ contains all the nonlinear and/or time-varying terms:

$$\dot{x} = \mathcal{F}_l(H, \Delta)x = \left(A + B\Delta(I - D\Delta)^{-1}C\right)x \tag{A.7}$$

This represents an LTI block H, with fictitious inputs and outputs p and q, where p and q are connected by feedback with Δ.

Well-Posedness If $\exists S \in \mathbf{S}$ and $G \in \mathbf{G}$ such that

$$p^T(D^T S D - \sigma^2 S + D^T G - GD)p < 0 \tag{A.8}$$

then the LDI (and hence the LFT) is well-posed for all $\Delta(t) \in \mathbf{\Delta}$ with $\|\Delta\| \leq \sigma^{-1}$, i.e. $\det(I - D\Delta) \neq 0$.

Proof

Assume the LDI is *not* well-posed, i.e. $\exists x = 0$, $q \neq 0$ such that $(I - D\Delta)q = 0$, with $q = Dp$.

Then (A.8) becomes

$$
\begin{aligned}
& p^T D^T S D p - \sigma^2 p^T S p + p^T D^T G p - p^T G D p \\
& = q^T S q - \sigma^2 p^T S p + q^T G p - p^T G q, \quad (\because Dp = q) \\
& = q^T S q - \sigma^2 (\Delta q)^T S \Delta q + q^T G \Delta q - (\Delta q)^T G q, \quad (\because p = \Delta q) \\
& = q^T S q - \sigma^2 q^T \Delta^T S \Delta q + q^T G \Delta q - q^T \Delta^T G q \\
& = q^T S q - \sigma^2 q^T \Delta S \Delta q + q^T G \Delta q - q^T \Delta G q, \quad (\because \Delta \text{ is symmetric }) \\
& = q^T S q - \sigma^2 q^T \Delta^2 S q + q^T \Delta G q - q^T \Delta G q, \quad (\because \Delta \text{ commutes with } S \text{ and } G) \\
& = q^T S q - \sigma^2 q^T \Delta^2 S q < 0
\end{aligned}
$$

where the fact that Δ commutes with both S and G follows from the fact that they are all block diagonal, where each block of Δ is a repeated scalar, of the same size as each block of both S and G.

The inequality above implies $I < \sigma^2 \Delta^2 \Rightarrow \lambda_{max}(\Delta^2) > \sigma^{-2} \Rightarrow \|\Delta\| > \sigma^{-1}$, which is a contradiction. Hence the LDI is well-posed.

Appendix B

S-Procedure

Here we introduce the S-procedure for quadratic forms [86]. The S-procedure is a means of combining several inequalities on quadratic functions into one inequality. This is used in the next section, where we have an LMI condition to show quadratic stability, and another LMI representing the constraint on the Δ block. Using the S-procedure, we can combine these to form one LMI proving quadratic stability of the system, subject to the constraint on Δ. The S-procedure for quadratic forms is as follows [86] :

Suppose we have the quadratic forms

$$\mathbf{z}^T T_0 \mathbf{z} > 0, \quad \mathbf{z} \neq 0, \quad \text{such that} \quad \mathbf{z}^T T_i \mathbf{z} \geqslant 0, \quad i = 1, ...p \tag{B.1}$$

where

$$T_0, ..., T_p \quad \text{are symmetric matrices} \tag{B.2}$$

then, if $\exists \quad \tau_1 \geqslant 0, ..., \tau_p \geqslant 0$ such that

$$T_0 - \sum_{i=1}^{p} \tau_i T_i > 0 \tag{B.3}$$

then (B.1) holds. This is generally just a sufficient condition for feasibility of the LMI system. However, when $p = 1$, the converse also holds [86] (this is called the lossless S-procedure).

Appendix C

Observer Synthesis

The purpose of this note is to present a synthesis method for a robust nonlinear observer for a nonlinear, uncertain system which can be represented in quasi-LPV form and show that this leads to a BMI.

The observer is the nonlinear analogue of a Luenberger observer; essentially it is a copy of the form of the plant, plus a residual term. The gain on this residual term, H, is the design variable.

We try to design H such that the (robust) L_2 gain from a known exogenous input to the performance output (error in the states we need to estimate) is minimised.

L_2 Gain

Assume $\exists\, V(\eta) = \eta^T P \eta > 0$ such that:

$$\frac{d}{dt}V(\eta(t)) + e(t)^T e(t) - \gamma^2 u(t)^T u(t) < 0 \tag{C.1}$$

Then integrating from $t = 0$ to $t = T$ gives:

$$V(\eta(t)) - V(\eta(0)) + \int_0^T e(t)^T e(t)\, dt - \gamma^2 \int_0^T u(t)^T u(t)\, dt < 0 \tag{C.2}$$

If $V(\eta(t)) - V(\eta(0)) > 0$, which is clearly the case if $\eta(0) = 0$, then this implies that:

$$\frac{\|e\|_2}{\|u\|_2} < \gamma \tag{C.3}$$

i.e. the L_2 gain from exogenous input u to estimation error e has upper bound γ.

Form of plant and observer

Suppose the plant is nonlinear, of the form

$$\dot{x} = f(x, \theta) + g(x, \theta)u \tag{C.4a}$$

$$y = C_y x \tag{C.4b}$$

where $x \in \mathbb{R}^n$ are the plant states, $y \in \mathbb{R}^{n_y}$ are measured outputs, $u \in \mathbb{R}^{n_u}$ are control inputs and θ is a vector of uncertain parameters that are not available to the observer.

Assume this system admits the the quasi-LPV form

$$\begin{bmatrix} \dot{x} \\ y \end{bmatrix} = \begin{bmatrix} \mathbb{A}(x, \theta) & \mathbb{B}(x, \theta) \\ C_y & 0_{n_y \times n_u} \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix} \tag{C.5}$$

and suppose this in turn can be written in the LFT form, similar to the precedure used in earlier chapters:

$$\dot{x} = Ax + B_u u + B_p p \tag{C.6a}$$

$$q = C_q x + D_{qu} u + D_{qp} p \tag{C.6b}$$

$$y = C_y x \tag{C.6c}$$

$$p = \Delta(x, \theta) q, \quad \|\Delta\| \leq \sigma^{-1} \tag{C.6d}$$

Suppose we want to design a nonlinear observer for this plant. It is standard practice [128] for the observer to have the same form as the plant, but perturbed by the difference between the estimated output $\hat{y}$ and the actual measured output y (residual):

$$\dot{\hat{x}} = \hat{f}(\hat{x}) + \hat{g}(\hat{x})u + H[\hat{y} - y] \tag{C.7a}$$

$$\hat{y} = C_y \hat{x} \tag{C.7b}$$

this leads to the observer having a similar LFT form to the plant:

$$\dot{\hat{x}} = -HC_y x + [\hat{A} + HC_y]\hat{x} + \hat{B}_u u + \hat{B}_p \hat{p} \tag{C.8a}$$

$$\hat{q} = \hat{C}_q \hat{x} + \hat{D}_{qu} u + \hat{D}_{qp} \hat{p} \tag{C.8b}$$

$$\hat{y} = C_y \hat{x} \tag{C.8c}$$

$$\hat{p} = \hat{\Delta}(\hat{x}) \hat{q}, \quad \|\hat{\Delta}\| \leq \sigma^{-1} \tag{C.8d}$$

where $H \in \mathbb{R}^{n \times n_y}$ "observer gain" is to be determined.

Note this is more complicated than the linear case. For a linear plant and observer we would be able to consider the error dynamics alone. This doesn't work if either the plant or the observer are nonlinear, because $f(\hat{x}) - f(x) \neq f(\hat{x} - x)$.

The plant augmented with the observer is given by:

$$\begin{bmatrix} \dot{x} \\ \dot{\hat{x}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ -HC_y & \hat{A} + HC_y \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} B_u \\ \hat{B}_u \end{bmatrix} u + \begin{bmatrix} B_p & 0 \\ 0 & \hat{B}_p \end{bmatrix} \begin{bmatrix} p \\ \hat{p} \end{bmatrix} \tag{C.9}$$

Or:

$$\dot{\eta} = \bar{A}\eta + \bar{B}_u u + \bar{B}_p \bar{p} \tag{C.10}$$

in feedback with

$$\begin{bmatrix} q \\ \hat{q} \end{bmatrix} = \begin{bmatrix} C_q & 0 \\ 0 & \hat{C}_q \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} + \begin{bmatrix} D_{qu} \\ \hat{D}_{qu} \end{bmatrix} u + \begin{bmatrix} D_{qp} & 0 \\ 0 & \hat{D}_{qp} \end{bmatrix} \begin{bmatrix} p \\ \hat{p} \end{bmatrix} \tag{C.11}$$

Or:

$$\bar{q} = \bar{C}_q \eta + \bar{D}_{qu} u + \bar{D}_{qp} \bar{p} \tag{C.12}$$

Via the block:

$$\begin{bmatrix} p \\ \hat{p} \end{bmatrix} = \begin{bmatrix} \Delta(x, \theta) & 0 \\ 0 & \hat{\Delta}(\hat{x}) \end{bmatrix} \begin{bmatrix} q \\ \hat{q} \end{bmatrix}, \quad \text{or} \quad \bar{p} = \bar{\Delta}(x, \hat{x}, \theta)\bar{q} \tag{C.13}$$

If we define the output of the augmented system to be the estimation error e, then this is given by:

$$e = L\hat{x} - Lx = \begin{bmatrix} -L & L \end{bmatrix} \begin{bmatrix} x \\ \hat{x} \end{bmatrix} =: \bar{L}\eta \tag{C.14}$$

So if the objective is to estimate all of the plant states, then we would take $L = I_n$.

Now say we want to find the observer gain matrix H that minimises the L_2 gain from the plant input u to the performance output e.

Setting $\xi := \begin{bmatrix} x^T & \hat{x}^T & u^T & p^T & \hat{p}^T \end{bmatrix}^T$, (C.1) holds if $\exists\ P \in \mathbb{R}^{2n \times 2n} > 0$ and $\gamma > 0$ such that $\xi^T X_P \xi < 0$ where $P := \begin{bmatrix} P_1 & P_3 \\ \star & P_2 \end{bmatrix} > 0$, $P_i \in \mathbb{R}^{n \times n}$ and:

$$X_P := \begin{bmatrix} \bar{A}^T P + P\bar{A} + \bar{L}^T \bar{L} & P\bar{B}_u & P\bar{B}_p \\ \bar{B}_u^T P & -\gamma^2 I_{n_u} & 0_{n_u \times 2n_p} \\ \bar{B}_p^T P & 0_{n_p \times n_u} & 0_{2n_p \times 2n_p} \end{bmatrix} \begin{matrix} 2n \\ n_u \\ 2n_p \end{matrix} < 0 \tag{C.15}$$

Scaling matrices

The scaling matrices for a block-diagonal Δ-block comprising blocks of real scalars has been discussed in previous chapters.

From $\bar{p} = \bar{\Delta}\bar{q}$ and $\|\bar{\Delta}\| \leq \sigma^{-1}$ we get

$$\bar{p}^T \bar{S} \bar{p} \leq \sigma^{-2} \bar{q}^T \bar{S} \bar{q}, \quad \bar{S}_i > 0 \tag{C.16}$$

(if we have the relations $p^T S p \leq \sigma^{-2} q^T S q, \ S_i > 0$ for the plant and $\hat{p}^T \hat{S} \hat{p} \leq \sigma^{-2} \hat{q}^T \hat{S} \hat{q}, \ \hat{S}_i > 0$ for the observer, then (C.16) above applies, where $\bar{S} = diag(S, \hat{S})$) which leads to $\xi^T X_S \xi \leq 0$, where

$$X_S := - \begin{bmatrix} \bar{C}_q^T \bar{S} \bar{C}_q & \bar{C}_q^T \bar{S} \bar{D}_{qu} & \bar{C}_q^T \bar{S} \bar{D}_{qp} \\ \bar{D}_{qu}^T \bar{S} \bar{C}_q & \bar{D}_{qu}^T \bar{S} \bar{D}_{qu} & \bar{D}_{qu}^T \bar{S} \bar{D}_{qp} \\ \bar{D}_{qp}^T \bar{S} \bar{C}_q & \bar{D}_{qp}^T \bar{S} \bar{D}_{qu} & \bar{D}_{qp}^T \bar{S} \bar{D}_{qp} - \sigma^2 \bar{S} \end{bmatrix} \leq 0 \tag{C.17}$$

Using the fact that signals are real (with a similar argument for combining the plant and observer scaling matrices as above, $\bar{G} = diag(G, \hat{G})$):

$$\bar{p}^T \bar{G} \bar{q} - \bar{q}^T \bar{G} \bar{p} = 0, \quad G_i = -G_i^T \tag{C.18}$$

which leads to $\xi^T X_G \xi = 0$, where

$$X_G := \begin{bmatrix} 0_{n \times n} & 0_{n \times n_u} & -\bar{C}_q^T \bar{G} \\ 0_{n_u \times n} & 0_{n_u \times n_u} & -\bar{D}_{qu}^T \bar{G} \\ \bar{G} \bar{C}_q & \bar{G} \bar{D}_{qu} & \bar{G} \bar{D}_{qp} - \bar{D}_{qp}^T \bar{G} \end{bmatrix} \tag{C.19}$$

Combining all three quadratic forms using the $\mathcal{S}$-procedure leads to $\xi^T Z \xi < 0$, where $Z < 0$ is the large BMI (C.20) on the next page.

The variables are blocks of the Lyapunov matrix: P_1, P_2 and $P_3 \in \mathbb{R}^{n \times n}$, where $P = \begin{bmatrix} P_1 & P_3 \\ \star & P_2 \end{bmatrix} > 0$; the observer gain matrix $H \in \mathbb{R}^{n \times n_y}$; the square of the L_2 gain, $\gamma^2 > 0$ and the scaling matrices $S \in \mathbf{S}$ and $G \in \mathbf{G}$.

We assume σ^2 is fixed (we assume that the LFT of the plant and observer is normalised such that $\sigma^2 = 1$). The objective is to find H that minimises γ^2.

$$Z := \left[\begin{array}{cc|c|cc} A^TP_1 + P_1A - P_3HC_y - C_y^TH^TP_3^T + L^TL + C_q^TSC_q & A^TP_3 + P_3\hat{A} + P_3HC_y - C_y^TH^TP_2 - L^TL & P_1B_u + P_3\hat{B}_u + C_q^TSD_{qu} & P_1B_p + C_q^TSD_{qp} + C_q^TG & P_3\hat{B}_p \\ \star & \hat{A}^TP_2 + P_2\hat{A} + P_2HC_y + C_y^TH^TP_2 + L^TL + \hat{C}_q^T\hat{S}\hat{C}_q & P_3^TB_u + P_2\hat{B}_u + \hat{C}_q^T\hat{S}\hat{D}_{qu} & P_3^TB_p & P_2\hat{B}_p + \hat{C}_q^T\hat{S}\hat{D}_{qp} + \hat{C}_q^T\hat{G} \\ \hline \star & \star & D_{qu}^TSD_{qu} + \hat{D}_{qu}^T\hat{S}\hat{D}_{qu} - \gamma^2 I_{n_u} & D_{qu}^TSD_{qp} + D_{qu}^TG & \hat{D}_{qu}^T\hat{S}\hat{D}_{qp} + \hat{D}_{qu}^T\hat{G} \\ \hline \star & \star & \star & D_{qp}^TSD_{qp} - \sigma^2 S + D_{qp}^TG - GD_{qp} & 0_{n_p \times n_p} \\ \star & \star & \star & \star & \hat{D}_{qp}^T\hat{S}\hat{D}_{qp} - \sigma^2\hat{S} + \hat{D}_{qp}^T\hat{G} - \hat{G}\hat{D}_{qp} \end{array}\right] \tag{C.20}$$

It can be seen that $Z < 0$ is a bilinear matrix inequality.

If we restrict the structure of the Lyapunov matrix to be block diagonal, i.e. $P_3 = 0_{n\times n}$, then this can be made an LMI by setting $Y := P_2 H \in \mathbb{R}^{n\times n_y}$. We can recover the original variable $H = P_2^{-1}Y$, assuming P_2 is invertible. By the definition of the Lyapunov matrix $P > 0$, we should also have $P_2 > 0$, so it will be invertible. Restricting the structure of the Lyapunov matrix introduces more conservatism.

If we set H first, calculating it by some other method, then clearly this is an LMI so standard tools can be used [143]. In that case, the LMI (C.20) would be simply to show robust stability of the augmented plant and is not for the observer synthesis.

Derivation of the large BMI (C.20)

From the definition of L_2 gain (C.15) Upper left block of (C.15):

$$\begin{aligned} P\bar{A} &= \begin{bmatrix} P_1 & P_3 \\ P_3^T & P_2 \end{bmatrix} \begin{bmatrix} A & 0_{n\times n} \\ -HC_y & \hat{A} + HC_y \end{bmatrix} \\ &= \begin{bmatrix} P_1 A - P_3 HC_y & P_3\hat{A} + P_3 HC_y \\ P_3^T A - P_2 HC_y & P_2\hat{A} + P_2 HC_y \end{bmatrix} \end{aligned} \tag{C.21}$$

plus transpose terms:

$$\begin{aligned} \bar{A}^T P &= \begin{bmatrix} A^T & -C_y^T H^T \\ 0_{n\times n} & \hat{A}^T + C_y^T H^T \end{bmatrix} \begin{bmatrix} P_1 & P_3 \\ P_3^T & P_2 \end{bmatrix} \\ &= \begin{bmatrix} A^T P_1 - C_y^T H^T P_3^T & A^T P_3 - C_y^T H^T P_2 \\ \hat{A}^T P_3^T + C_y^T H^T P_3^T & \hat{A}^T P_2 + C_y^T H^T P_2 \end{bmatrix} \end{aligned} \tag{C.22}$$

$$\bar{L}^T \bar{L} = \begin{bmatrix} -L^T \\ L^T \end{bmatrix} \begin{bmatrix} -L & L \end{bmatrix} = \begin{bmatrix} L^T L & -L^T L \\ -L^T L & L^T L \end{bmatrix} \tag{C.23}$$

(1,2) block of (C.15)

$$P\bar{B}_u = \begin{bmatrix} P_1 & P_3 \\ P_3^T & P_2 \end{bmatrix} \begin{bmatrix} B_u \\ \hat{B}_u \end{bmatrix} = \begin{bmatrix} P_1 B_u + P_3 \hat{B}_u \\ P_3^T B_u + P_2 \hat{B}_u \end{bmatrix} \tag{C.24}$$

(1,3) block of (C.15)

$$P\bar{B}_p = \begin{bmatrix} P_1 & P_3 \\ P_3^T & P_2 \end{bmatrix} \begin{bmatrix} B_p & 0 \\ 0 & \hat{B}_p \end{bmatrix} = \begin{bmatrix} P_1 B_p & P_3 \hat{B}_p \\ P_3^T B_p & P_2 \hat{B}_p \end{bmatrix} \tag{C.25}$$

From (C.17) **(S scaling)** The (1,1) block of (C.17):

$$\bar{C}_q^T \bar{S} \bar{C}_q = \begin{bmatrix} C_q^T & 0 \\ 0 & \hat{C}_q^T \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & \hat{S} \end{bmatrix} \begin{bmatrix} C_q & 0 \\ 0 & \hat{C}_q \end{bmatrix} = \begin{bmatrix} C_q^T S C_q & 0 \\ 0 & \hat{C}_q^T \hat{S} \hat{C}_q \end{bmatrix} \tag{C.26}$$

The (1,2) block:

$$\bar{C}_q^T \bar{S} \bar{D}_{qu} = \begin{bmatrix} C_q^T & 0 \\ 0 & \hat{C}_q^T \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & \hat{S} \end{bmatrix} \begin{bmatrix} D_{qu} \\ \hat{D}_{qu} \end{bmatrix} = \begin{bmatrix} C_q^T S D_{qu} \\ \hat{C}_q^T \hat{S} \hat{D}_{qu} \end{bmatrix} \tag{C.27}$$

The (1,3) block:

$$\bar{C}_q^T \bar{S} \bar{D}_{qp} = \begin{bmatrix} C_q^T & 0 \\ 0 & \hat{C}_q^T \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & \hat{S} \end{bmatrix} \begin{bmatrix} D_{qp} & 0 \\ 0 & \hat{D}_{qp} \end{bmatrix} = \begin{bmatrix} C_q^T S D_{qp} & 0 \\ 0 & \hat{C}_q^T \hat{S} \hat{D}_{qp} \end{bmatrix} \tag{C.28}$$

The (2,2) block:

$$\bar{D}_{qu}^T \bar{S} \bar{D}_{qu} = \begin{bmatrix} D_{qu}^T & \hat{D}_{qu}^T \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & \hat{S} \end{bmatrix} \begin{bmatrix} D_{qu} \\ \hat{D}_{qu} \end{bmatrix} = D_{qu}^T S D_{qu} + \hat{D}_{qu}^T \hat{S} \hat{D}_{qu} \tag{C.29}$$

The (2,3) block:

$$\bar{D}_{qu}^T \bar{S} \bar{D}_{qp} = \begin{bmatrix} D_{qu}^T & \hat{D}_{qu}^T \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & \hat{S} \end{bmatrix} \begin{bmatrix} D_{qp} & 0 \\ 0 & \hat{D}_{qp} \end{bmatrix} = \begin{bmatrix} D_{qu}^T S D_{qp} & \hat{D}_{qu}^T \hat{S} \hat{D}_{qp} \end{bmatrix} \tag{C.30}$$

The (3,3) block:

$$\begin{aligned} \bar{D}_{qp}^T \bar{S} \bar{D}_{qp} - \sigma^2 \bar{S} &= \begin{bmatrix} D_{qp}^T & 0 \\ 0 & \hat{D}_{qp}^T \end{bmatrix} \begin{bmatrix} S & 0 \\ 0 & \hat{S} \end{bmatrix} \begin{bmatrix} D_{qp} & 0 \\ 0 & \hat{D}_{qp} \end{bmatrix} - \begin{bmatrix} \sigma^2 S & 0 \\ 0 & \sigma^2 \hat{S} \end{bmatrix} \\ &= \begin{bmatrix} D_{qp}^T S D_{qp} - \sigma^2 S & 0 \\ 0 & \hat{D}_{qp}^T \hat{S} \hat{D}_{qp} - \sigma^2 \hat{S} \end{bmatrix} \end{aligned} \tag{C.31}$$

From (C.19) **(G scaling):** the (1,1), (1,2) and (2,2) blocks are zeros.

The (1,3) block of (C.19):

$$\bar{C}_q^T \bar{G} = \begin{bmatrix} C_q^T & 0 \\ 0 & \hat{C}_q^T \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & \hat{G} \end{bmatrix} = \begin{bmatrix} C_q^T G & 0 \\ 0 & \hat{C}_q^T \hat{G} \end{bmatrix} \tag{C.32}$$

The (2,3) block:

$$\bar{D}_{qu}^T \bar{G} = \begin{bmatrix} D_{qu}^T & \hat{D}_{qu}^T \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & \hat{G} \end{bmatrix} = \begin{bmatrix} D_{qu}^T G & \hat{D}_{qu}^T \hat{G} \end{bmatrix} \tag{C.33}$$

The (3,3) block:

$$\begin{aligned} \bar{D}_{qp}^T \bar{G} - \bar{G} \bar{D}_{qp} &= \begin{bmatrix} D_{qp}^T & 0 \\ 0 & \hat{D}_{qp}^T \end{bmatrix} \begin{bmatrix} G & 0 \\ 0 & \hat{G} \end{bmatrix} - \begin{bmatrix} G & 0 \\ 0 & \hat{G} \end{bmatrix} \begin{bmatrix} D_{qp} & 0 \\ 0 & \hat{D}_{qp} \end{bmatrix} \\ &= \begin{bmatrix} D_{qp}^T G - G D_{qp} & 0 \\ 0 & \hat{D}_{qp}^T \hat{G} - \hat{G} \hat{D}_{qp} \end{bmatrix} \end{aligned} \tag{C.34}$$

Appendix D

Schur Complement Lemma

The Schur complement lemma is given [185], and can be used to transform some non-affine matrix inequalities into equivalent affine ones.

Schur complement

$$\begin{bmatrix} Q & S \\ S^T & R \end{bmatrix} < 0 \quad \text{iff} \quad \begin{matrix} Q < 0 & \text{and} & R - S^T Q^{-1} S < 0 \\ R < 0 & \text{and} & Q - S R^{-1} S^T < 0 \end{matrix} \tag{D.1}$$

Appendix E

ADMIRE control transformation

ADMIRE control surface deflections defined in table 4.2:

$u_{ADM} = \begin{bmatrix} \delta_{rc} & \delta_{lc} & \delta_{roe} & \delta_{rie} & \delta_{lie} & \delta_{loe} & \delta_r \end{bmatrix}^T, \quad \delta_{rc} = \delta_{lc}$

These are transformed into the commonly-used symmetric/ asymmetric controls:

$u = \begin{bmatrix} \delta_n & \delta_{ei} & \delta_{ey} & \delta_{ai} & \delta_{ay} & \delta_r \end{bmatrix}^T$

using $u_{ADM} = Tu$ and $u = T^{\#}u_{ADM}$

where

$$T^{\#} = \frac{1}{2}\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 \end{bmatrix} \tag{E.1}$$

$$T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{E.2}$$

Appendix F

Polynomial model of ADMIRE short-period dynamics in control-affine form

$$x = \begin{bmatrix} \tilde{\alpha} & \beta & \mu & p_b & q_b & r_b \end{bmatrix}^T$$

$$f(x) = \begin{bmatrix} -1.0768\tilde{\alpha} + 0.9728q_b - 0.95208\beta p_b \\ -0.2438\beta + 0.0609p_b - 0.9893r_b + 0.058055\tilde{\alpha}\beta + 0.95075\tilde{\alpha}p_b + 0.021665\beta q_b - 0.26768\tilde{\alpha}r_b \\ p_b(1 - 0.5(\tilde{\alpha} + \alpha_{trim})^2)(1 - 0.5\beta^2) + q_b\beta + r_b(\tilde{\alpha} + \alpha_{trim})(1 - 0.5\beta^2) \\ -23.613\beta - 2.0784p_b + 0.5423r_b - 130.28\tilde{\alpha}\beta - 3.8229\tilde{\alpha}p_b - 13.793\beta q_b - 5.0752\tilde{\alpha}r_b + 0.10389p_b q_b - 2.2525q_b r_b \\ 9.1771\tilde{\alpha} - 1.079q_b + 0.28991\beta p_b - 0.2791\tilde{\alpha}q_b + 0.15552\beta r_b + 1.052p_b r_b + 5.9292\tilde{\alpha}^2 - 30.32\tilde{\alpha}^3 - 0.033186{p_b}^2 - 0.074713{r_b}^2 \\ 1.7719\beta - 0.1217p_b - 0.416r_b - 6.4015\tilde{\alpha}\beta - 0.78091\tilde{\alpha}p_b - 1.5148\beta q_b + 0.37012\tilde{\alpha}r_b - 0.62512p_b q_b - 0.36707q_b r_b \end{bmatrix} \tag{F.1}$$

$$A(x) = \begin{bmatrix} -1.0768 & 0 & 0 & -0.95208\beta & 0.9728 & 0 \\ 0 & -0.2438 + 0.058055\tilde{\alpha} & 0 & 0.0609 + 0.95075\tilde{\alpha} & 0.021665\beta & -0.9893 - 0.26768\tilde{\alpha} \\ 0 & 0 & 0 & (1 - 0.5(\tilde{\alpha} + \alpha_{trim})^2)(1 - 0.5\beta^2) & \beta & (\tilde{\alpha} + \alpha_{trim})(1 - 0.5\beta^2) \\ 0 & -23.613 - 130.28\tilde{\alpha} & 0 & -2.0784 - 3.8229\tilde{\alpha} + 0.10389q_b & -13.793\beta & 0.5423 - 5.0752\tilde{\alpha} - 2.2525q_b \\ 9.1771 + 5.9292\tilde{\alpha} - 30.32\tilde{\alpha}^2 & 0 & 0 & 0.28991\beta + 1.052r_b - 0.033186p_b & -1.079 - 0.2791\tilde{\alpha} & 0.15552\beta - 0.074713r_b \\ 0 & 1.7719 - 6.4015\tilde{\alpha} & 0 & -0.1217 - 0.78091\tilde{\alpha} - 0.62512q_b & -1.5148\beta & -0.416 + 0.37012\tilde{\alpha} - 0.36707q_b \end{bmatrix} \tag{F.2}$$

$$u = \begin{bmatrix} \delta_n & \delta_{ei} & \delta_{ey} & \delta_{ai} & \delta_{ay} & \delta_r \end{bmatrix}^T$$

$$G(\tilde{\alpha},\beta) = \begin{bmatrix} -0.0126 + 0.0089154\tilde{\alpha} & -0.2264 + 0.055501\tilde{\alpha} & -0.1361 + 0.11199\tilde{\alpha} & 0.34806\beta & 0.022782\beta & 0.03423\beta \\ -0.023841\beta & -0.020724\beta & 0.0043252\beta & 0.0383 - 0.45514\tilde{\alpha} & 0.0077 - 0.19667\tilde{\alpha} & 0.0666 - 0.45611\tilde{\alpha} \\ 0 & 0 & 0 & 0 & 0 & 0 \\ -10.674\beta & -20.065\beta & -15.582\beta & -20.693 - 17.275\tilde{\alpha} & -23.706 - 25.901\tilde{\alpha} & 5.236 - 7.9192\tilde{\alpha} \\ 5.6652 + 2.2686\tilde{\alpha} & -8.7692 + 0.064492\tilde{\alpha} & -5.3918 + 2.9578\tilde{\alpha} & 0.17627\beta & 0.17224\beta & 0.25352\beta \\ 1.5842\beta & -1.4258\beta & -1.1976\beta & -2.7575 + 0.48663\tilde{\alpha} & -1.1538 - 1.039\tilde{\alpha} & -4.2585 + 0.38327\tilde{\alpha} \end{bmatrix} \tag{F.3}$$

Appendix G

LFT matrices and LMI matrix variables

LFT state matrix of the closed loop $\mathcal{A} =$

$$\begin{pmatrix}
-1.08 & 0 & 0 & 0 & 0.973 & 0 & -0.0126 & -0.226 & -0.136 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -0.244 & 0 & 0.0609 & 0 & -0.989 & 0 & 0 & 0 & 0.0383 & 0.0077 & 0.0666 & 0 & 0 & 0 \\
0 & 0 & 0 & 0.998 & 0 & 0.0608 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & -23.6 & 0 & -2.08 & 0 & 0.542 & 0 & 0 & 0 & -20.7 & -23.7 & 5.24 & 0 & 0 & 0 \\
9.18 & 0 & 0 & 0 & -1.08 & 0 & 5.67 & -8.77 & -5.39 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.77 & 0 & -0.122 & 0 & -0.416 & 0 & 0 & 0 & -2.76 & -1.15 & -4.26 & 0 & 0 & 0 \\
-57.3 & 0 & 0 & 0 & -9.41 & 0 & -20.0 & 0 & 0 & 0 & 0 & 0 & -22.3 & 0 & 0 \\
88.8 & 0 & 0 & 0 & 14.6 & 0 & 0 & -20.0 & 0 & 0 & 0 & 0 & 34.5 & 0 & 0 \\
54.6 & 0 & 0 & 0 & 8.96 & 0 & 0 & 0 & -20.0 & 0 & 0 & 0 & 21.2 & 0 & 0 \\
0 & -29.5 & 9.58 & 2.73 & 0 & 6.4 & 0 & 0 & 0 & -20.0 & 0 & 0 & 0 & -4.13 & 1.64 \\
0 & -5.77 & 12.4 & 4.26 & 0 & -0.959 & 0 & 0 & 0 & 0 & -20.0 & 0 & 0 & 1.04 & 2.13 \\
0 & -61.8 & -5.9 & -3.49 & 0 & 18.9 & 0 & 0 & 0 & 0 & 0 & -20.0 & 0 & -13.2 & -1.01 \\
1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 1.0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}$$

Lyapunov matrix for the case $\sigma_p^2 = 73$, $\sigma_w^2 = 1$:

$$P = \begin{pmatrix}
4820.0 & -63.8 & 93.4 & 35.4 & 559.0 & -40.0 & 82.1 & -157.0 & -85.8 & -67.9 & -79.9 & 12.6 & 2160.0 & 0.378 & 16.2 \\
-63.8 & 1510.0 & -186.0 & -47.3 & -7.28 & -199.0 & -1.24 & 26.0 & 13.3 & 134.0 & 137.0 & 5.57 & -1.98 & 251.0 & -29.0 \\
93.4 & -186.0 & 1020.0 & 148.0 & -13.6 & -7.66 & -13.9 & -32.4 & -21.0 & -210.0 & -231.0 & 40.0 & -127.0 & -19.4 & 173.0 \\
35.4 & -47.3 & 148.0 & 54.1 & -5.62 & -5.19 & -0.638 & -9.36 & -5.3 & -59.5 & -67.2 & 14.0 & -19.8 & 3.91 & 25.2 \\
559.0 & -7.28 & -13.6 & -5.62 & 184.0 & 0.364 & 28.3 & -43.6 & -26.0 & 14.1 & 16.5 & -5.51 & 419.0 & -6.7 & -1.79 \\
-40.0 & -199.0 & -7.66 & -5.19 & 0.364 & 144.0 & -0.224 & 3.08 & 1.83 & -8.66 & 0.238 & -23.2 & 20.2 & -106.0 & 4.33 \\
82.1 & -1.24 & -13.9 & -0.638 & 28.3 & -0.224 & 2480000.0 & 1160000.0 & 715000.0 & 0.758 & 0.247 & -0.514 & 28.7 & -29.6 & 10.6 \\
-157.0 & 26.0 & -32.4 & -9.36 & -43.6 & 3.08 & 1160000.0 & 1430000.0 & -1110000.0 & 17.7 & 19.5 & -2.17 & -83.1 & -14.2 & 0.638 \\
-85.8 & 13.3 & -21.0 & -5.3 & -26.0 & 1.83 & 715000.0 & -1110000.0 & 2550000.0 & 10.2 & 11.3 & -1.05 & -93.3 & -7.03 & -0.83 \\
-67.9 & 134.0 & -210.0 & -59.5 & 14.1 & -8.66 & 0.758 & 17.7 & 10.2 & 1560000.0 & -1500000.0 & -606000.0 & 55.5 & 9.64 & -33.0 \\
-79.9 & 137.0 & -231.0 & -67.2 & 16.5 & 0.238 & 0.247 & 19.5 & 11.3 & -1500000.0 & 1440000.0 & 581000.0 & 109.0 & 4.73 & -38.6 \\
12.6 & 5.57 & 40.0 & 14.0 & -5.51 & -23.2 & -0.514 & -2.17 & -1.05 & -606000.0 & 581000.0 & 235000.0 & -36.7 & 14.8 & 5.17 \\
2160.0 & -1.98 & -127.0 & -19.8 & 419.0 & 20.2 & 28.7 & -83.1 & -93.3 & 55.5 & 109.0 & -36.7 & 11300.0 & -350.0 & -18.0 \\
0.378 & 251.0 & -19.4 & 3.91 & -6.7 & -106.0 & -29.6 & -14.2 & -7.03 & 9.64 & 4.73 & 14.8 & -350.0 & 1090.0 & -61.0 \\
16.2 & -29.0 & 173.0 & 25.2 & -1.79 & 4.33 & 10.6 & 0.638 & -0.83 & -33.0 & -38.6 & 5.17 & -18.0 & -61.0 & 85.3
\end{pmatrix}$$

Lyapunov matrix for the case $\sigma_p^2 = 20$, $\sigma_w^2 = 4$:

$$P = \begin{pmatrix}
953.0 & -18.2 & 8.25 & 7.1 & 112.0 & -10.9 & 24.0 & -21.3 & -8.48 & -19.2 & -22.5 & 1.57 & 422.0 & -0.295 & 1.41 \\
-18.2 & 339.0 & -36.9 & -9.39 & -3.94 & -31.3 & 0.107 & 8.05 & 3.5 & 26.7 & 28.8 & -0.769 & 3.25 & 42.0 & -6.02 \\
8.25 & -36.9 & 342.0 & 34.3 & -3.91 & -6.12 & -4.89 & -7.78 & -4.41 & -56.7 & -63.8 & 12.6 & -37.0 & 4.13 & 53.1 \\
7.1 & -9.39 & 34.3 & 12.4 & -1.23 & -1.67 & 0.701 & -2.1 & -1.05 & -14.2 & -16.0 & 3.23 & 0.845 & 1.88 & 5.63 \\
112.0 & -3.94 & -3.91 & -1.23 & 43.3 & -0.231 & 6.56 & -6.4 & -3.52 & 3.85 & 4.52 & -1.69 & 75.9 & -2.87 & -0.564 \\
-10.9 & -31.3 & -6.12 & -1.67 & -0.231 & 32.5 & -1.59 & 0.362 & 0.109 & 0.516 & 2.03 & -3.69 & 0.425 & -24.0 & 0.482 \\
24.0 & 0.107 & -4.89 & 0.701 & 6.56 & -1.59 & 222000.0 & 104000.0 & 63900.0 & -0.964 & -2.74 & 0.964 & -8.14 & -2.79 & 2.33 \\
-21.3 & 8.05 & -7.78 & -2.1 & -6.4 & 0.362 & 104000.0 & 128000.0 & -98900.0 & 4.75 & 4.77 & 0.0804 & -14.5 & -1.8 & 0.0916 \\
-8.48 & 3.5 & -4.41 & -1.05 & -3.52 & 0.109 & 63900.0 & -98900.0 & 228000.0 & 2.2 & 1.77 & 0.311 & -14.9 & -0.129 & 0.0309 \\
-19.2 & 26.7 & -56.7 & -14.2 & 3.85 & 0.516 & -0.964 & 4.75 & 2.2 & 140000.0 & -134000.0 & -54200.0 & 5.11 & -1.15 & -7.36 \\
-22.5 & 28.8 & -63.8 & -16.0 & 4.52 & 2.03 & -2.74 & 4.77 & 1.77 & -134000.0 & 128000.0 & 51900.0 & 14.5 & -1.69 & -9.03 \\
1.57 & -0.769 & 12.6 & 3.23 & -1.69 & -3.69 & 0.964 & 0.0804 & 0.311 & -54200.0 & 51900.0 & 21000.0 & -5.2 & 2.52 & 1.34 \\
422.0 & 3.25 & -37.0 & 0.845 & 75.9 & 0.425 & -8.14 & -14.5 & -14.9 & 5.11 & 14.5 & -5.2 & 1500.0 & -91.3 & -17.6 \\
-0.295 & 42.0 & 4.13 & 1.88 & -2.87 & -24.0 & -2.79 & -1.8 & -0.129 & -1.15 & -1.69 & 2.52 & -91.3 & 259.0 & -6.86 \\
1.41 & -6.02 & 53.1 & 5.63 & -0.564 & 0.482 & 2.33 & 0.0916 & 0.0309 & -7.36 & -9.03 & 1.34 & -17.6 & -6.86 & 31.3
\end{pmatrix}$$

Lyapunov matrix for the case $\sigma_p^2 = 9.73$:

$$P = \begin{pmatrix}
953.0 & -18.2 & 8.25 & 7.1 & 112.0 & -10.9 & 24.0 & -21.3 & -8.48 & -19.2 & -22.5 & 1.57 & 422.0 & -0.295 & 1.41 \\
-18.2 & 339.0 & -36.9 & -9.39 & -3.94 & -31.3 & 0.107 & 8.05 & 3.5 & 26.7 & 28.8 & -0.769 & 3.25 & 42.0 & -6.02 \\
8.25 & -36.9 & 342.0 & 34.3 & -3.91 & -6.12 & -4.89 & -7.78 & -4.41 & -56.7 & -63.8 & 12.6 & -37.0 & 4.13 & 53.1 \\
7.1 & -9.39 & 34.3 & 12.4 & -1.23 & -1.67 & 0.701 & -2.1 & -1.05 & -14.2 & -16.0 & 3.23 & 0.845 & 1.88 & 5.63 \\
112.0 & -3.94 & -3.91 & -1.23 & 43.3 & -0.231 & 6.56 & -6.4 & -3.52 & 3.85 & 4.52 & -1.69 & 75.9 & -2.87 & -0.564 \\
-10.9 & -31.3 & -6.12 & -1.67 & -0.231 & 32.5 & -1.59 & 0.362 & 0.109 & 0.516 & 2.03 & -3.69 & 0.425 & -24.0 & 0.482 \\
24.0 & 0.107 & -4.89 & 0.701 & 6.56 & -1.59 & 222000.0 & 104000.0 & 63900.0 & -0.964 & -2.74 & 0.964 & -8.14 & -2.79 & 2.33 \\
-21.3 & 8.05 & -7.78 & -2.1 & -6.4 & 0.362 & 104000.0 & 128000.0 & -98900.0 & 4.75 & 4.77 & 0.0804 & -14.5 & -1.8 & 0.0916 \\
-8.48 & 3.5 & -4.41 & -1.05 & -3.52 & 0.109 & 63900.0 & -98900.0 & 228000.0 & 2.2 & 1.77 & 0.311 & -14.9 & -0.129 & 0.0309 \\
-19.2 & 26.7 & -56.7 & -14.2 & 3.85 & 0.516 & -0.964 & 4.75 & 2.2 & 140000.0 & -134000.0 & -54200.0 & 5.11 & -1.15 & -7.36 \\
-22.5 & 28.8 & -63.8 & -16.0 & 4.52 & 2.03 & -2.74 & 4.77 & 1.77 & -134000.0 & 128000.0 & 51900.0 & 14.5 & -1.69 & -9.03 \\
1.57 & -0.769 & 12.6 & 3.23 & -1.69 & -3.69 & 0.964 & 0.0804 & 0.311 & -54200.0 & 51900.0 & 21000.0 & -5.2 & 2.52 & 1.34 \\
422.0 & 3.25 & -37.0 & 0.845 & 75.9 & 0.425 & -8.14 & -14.5 & -14.9 & 5.11 & 14.5 & -5.2 & 1500.0 & -91.3 & -17.6 \\
-0.295 & 42.0 & 4.13 & 1.88 & -2.87 & -24.0 & -2.79 & -1.8 & -0.129 & -1.15 & -1.69 & 2.52 & -91.3 & 259.0 & -6.86 \\
1.41 & -6.02 & 53.1 & 5.63 & -0.564 & 0.482 & 2.33 & 0.0916 & 0.0309 & -7.36 & -9.03 & 1.34 & -17.6 & -6.86 & 31.3
\end{pmatrix}$$

Bibliography

[1] Hans Backström. Report on the usage of the generic aerodata model. Technical report, Saab Aircraft AB, Saab Military Aircraft, S-581 88 Linköping Sweden, 1996.

[2] L. Forssell and U. Nilsson. ADMIRE: The Aero-Data Model in a Research Environment version 4.0 model description. Technical Report FOI-R–1624–SE, Swedish Defence Research Agency, Stockholm, December 2005.

[3] Arthur J. Krener. On the equivalence of control systems and the linearization of nonlinear systems. *SIAM Journal on Control*, 11(4):670–676, 1973.

[4] B. Jacubczyk and W. Respondek. On linearization of control systems. *Bulletindel Academic Polonaise Sciences Serie Sciences Mathematiques*, 28:517–522, 1980.

[5] Renjeng Su. On the linear equivalents of nonlinear systems. *Systems & Control Letters*, 2(1):48 – 52, 1982.

[6] L. Hunt, Renjeng Su, and G. Meyer. Global transformations of nonlinear systems. *Automatic Control, IEEE Transactions on*, 28(1):24–31, Jan 1983.

[7] C.A. Desoer and Yung-Terng Wang. Foundations of feedback theory for nonlinear dynamical systems. *Circuits and Systems, IEEE Transactions on*, 27(2):104–123, Feb 1980.

[8] A. Isidori and A. Ruberti. On the synthesis of linear input-output responses for nonlinear systems. *Systems & Control Letters*, 4(1):17 – 22, 1984.

[9] W.T. Baumann and W.J. Rugh. Feedback control of nonlinear systems by extended linearization. *Automatic Control, IEEE Transactions on*, 31(1):40–46, Jan 1986.

[10] J. Wang and W.J. Rugh. Feedback linearization families for nonlinear systems. *Automatic Control, IEEE Transactions on*, 32(10):935–940, Oct 1987.

[11] W.T. Baumann. Feedback control of multiinput nonlinear systems by extended linearization. *Automatic Control, IEEE Transactions on*, 33(2):193–197, Feb 1988.

[12] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.

[13] J.-F. Magni. User Manual of the Linear Fractional Representation Toolbox: Version 2.0. Technical Report TR 5/10403.01F DCSD, ONERA, Toulouse, October 2005.

[14] R.T. Reichert. Dynamic scheduling of modern-robust-control autopilot designs for missiles. In *Decision and Control, 1991., Proceedings of the 30th IEEE Conference on*, pages 2994 vol.3–, Dec 1991.

[15] J.J.E. Slotine and W. Li. *Applied Nonlinear Control*. Prentice Hall, 1991.

[16] A. Isidori. *Nonlinear Control Systems*. Communications and Control Engineering. Springer London, 2013.

[17] C. J. Tomlin and S. S. Sastry. Bounded tracking for non-minimum phase nonlinear systems with fast zero dynamics. *International Journal of Control*, 68(4):819–848, 1997.

[18] Gang Gao, Jinzhi Wang, and Xianghua Wang. Adaptive fault-tolerant control for feedback linearizable systems with an aircraft application. *International Journal of Robust and Nonlinear Control*, 25(9):1301–1326, 2015.

[19] Kevin A. Wise, Joseph S. Brinker, Anthony J. Calise, Dale F. Enns, Michael R. Elgersma, and Petros Voulgaris. Direct adaptive reconfigurable flight control

for a tailless advanced fighter aircraft. *International Journal of Robust and Nonlinear Control*, 9(14):999–1012, 1999.

[20] William C. Reigelsperger, Kelly D. Hammett, and Siva S. Banda. Robust control law design for lateral-directional modes of an f-16/matv using -synthesis and dynamic inversion. *International Journal of Robust and Nonlinear Control*, 7(8):777–795, 1997.

[21] James Anderson and Antonis Papachristodoulou. Robust nonlinear stability and performance analysis of an f/a-18 aircraft model using sum of squares programming. *International Journal of Robust and Nonlinear Control*, 23(10):1099–1114, 2013.

[22] O.U. Rehman, B. Fidan, and I.R. Petersen. Robust minimax optimal control of nonlinear uncertain systems using feedback linearization with application to hypersonic flight vehicles. In *Decision and Control, 2009 held jointly with the 2009 28th Chinese Control Conference. CDC/CCC 2009. Proceedings of the 48th IEEE Conference on*, pages 720–726, Dec 2009.

[23] W. MacKunis, P.M. Patre, M.K. Kaiser, and W.E. Dixon. Asymptotic tracking for aircraft via robust and adaptive dynamic inversion methods. *Control Systems Technology, IEEE Transactions on*, 18(6):1448–1456, Nov 2010.

[24] W. MacKunis, M.K. Kaiser, P.M. Patre, and W.E. Dixon. Asymptotic tracking for aircraft via an uncertain dynamic inversion method. In *American Control Conference, 2008*, pages 3482–3487, June 2008.

[25] Ruyi Yuan, Guoliang Fan, Jianqiang Yi, and Wensheng Yu. Robust attitude controller for unmanned aerial vehicle using dynamic inversion and extended state observer. In *Intelligent Computation Technology and Automation, 2009. ICICTA '09. Second International Conference on*, volume 1, pages 850–853, Oct 2009.

[26] Dou Liqian, Zong Qun, and Ji Yuehui. Robust dynamic inversion control for near space vehicle. In *Systems and Control in Aeronautics and Astronautics (ISSCAA), 2010 3rd International Symposium on*, pages 744–749, June 2010.

[27] G. Ducard and H.P. Geering. Stability analysis of a dynamic inversion based pitch rate controller for an unmanned aircraft. In *Intelligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on*, pages 360–366, Sept 2008.

[28] C. Tomlin, J. Lygeros, L. Benvenuti, and S. Sastry. Output tracking for a non-minimum phase dynamic ctol aircraft model. In *Decision and Control, 1995., Proceedings of the 34th IEEE Conference on*, volume 2, pages 1867–1872 vol.2, Dec 1995.

[29] C. Papageorgiou and Keith Glover. Robustness analysis of nonlinear dynamic inversion control laws with application to flight control. In *Decision and Control, 2004. CDC. 43rd IEEE Conference on*, volume 4, pages 3485–3490 Vol.4, Dec 2004.

[30] Prathyush P. Menon, Mark Lowenberg, Guido Herrmann, Matthew C. Turner, Declan G. Bates, and Ian Postlethwaite. Experimental implementation of a nonlinear dynamic inversion controller with antiwindup. *Journal of Guidance, Control, and Dynamics*, 36(4):1035–1046, May 2013.

[31] Lisa Fiorentini, Andrea Serrani, Michael A. Bolender, and David B. Doman. Nonlinear robust adaptive control of flexible air-breathing hypersonic vehicles. *Journal of Guidance, Control, and Dynamics*, 32(2):402–417, Mar 2009.

[32] Saif A. Al-Hiddabi. Semiglobal trajectory tracking control law for a nonlinear nonminimum phase three-degree-of-freedom flight vehicle. *Journal of Guidance, Control, and Dynamics*, 32(2):693–699, Mar 2009.

[33] Yoonghyun Shin, Anthony J. Calise, and Matthew Johnson. Adaptive control of advanced fighter aircraft in nonlinear flight regimes. *Journal of Guidance, Control, and Dynamics*, 31(5):1464–1477, Sep 2008.

[34] Corey Schumacher and Pramod P. Khargonekar. Stability analysis of a missile control system with a dynamic inversion controller. *Journal of Guidance, Control, and Dynamics*, 21(3):508–515, May 1998.

[35] J. Reiner, G.J. Balas, and W.L. Garrard. Flight control design using robust dynamic inversion and time-scale separation. *Automatica*, 32(11):1493–1504, 1996.

[36] J. Huang, C.F. Lin, J.R. Cloutier, J.H. Evers, and C. D'Souza. Robust feedback linearization approach to autopilot design. In *Control Applications, 1992., First IEEE Conference on*, pages 220–225 vol.1, Sep 1992.

[37] S.A. Al-Hiddabi and N.H. McClamroch. Output tracking for nonlinear non-minimum phase vtol aircraft. In *Decision and Control, 1998. Proceedings of the 37th IEEE Conference on*, volume 4, pages 4573–4578 vol.4, Dec 1998.

[38] E. Devaud, J.P. Harcaut, and H. Siguerdidjane. Three-axes missile autopilot design: From linear to nonlinear control strategies. *Journal of Guidance, Control, and Dynamics*, 24(1):64–71, 2001.

[39] I.J. Adounkpé, E. Godoy, and J.P. Harcaut. Robustness analysis of a nonlinear missile autopilot : an integral quadratic constraint approach. In *5th Asian Control Conference.*, volume 3, pages 1833 –1840, July 2004.

[40] D. Enns, D. Bugajski, R. Hendrick, and G. Stein. Dynamic inversion: An evolving methodology for flight control design. *International Journal of Control*, 59(1):71–91, 1994.

[41] B. Morton, D. Enns, and B. Zhang. Stability of dynamic inversion control laws applied to nonlinear aircraft pitch-axis models. *International Journal of Control*, 63(1):1–25, 1996.

[42] H. Fer and D. Enns. Nonlinear longitudinal axis regulation of f-18 harv using feedback linearization and lqr. In *Proceedings of the 36th IEEE Conference on Decision and Control*, pages 4173–4178, 1997.

[43] D. Enns and T. Keviczky. Dynamic inversion based flight control for autonomous rmax helicopter. In *Proceedings of the American Control Conference*, pages 3916–3923, June 2006.

[44] J. Tierno and S. Glavaški. Phase-lead compensation of pitch-axis control laws via mcfarlane-glover hinf loop shaping. In *Proceedings o the 38th IEEE Conference on Decision and Control*, pages 1538–1543, December 1999.

[45] A. Tsourdos, D.J. Leith, W.E. Leithead, and B.A. White. A velocity-based framework for the robust stability analysis of dynamic inversion flight controllers. In *Proceedings of the American Control Conference*, volume 5, pages 3341–3345, June 25-27 2001.

[46] L. Benvenuti, M.D. Di Benedetto, and J.W. Grizzle. Approximate output tracking for nonlinear non-minimum phase systems with an application to flight control. *International Journal of Robust and Nonlinear Control*, 4:397 – 414, 1994.

[47] Stephen H. Lane and Robert F. Stengel. Flight control design using non-linear inverse dynamics. *Automatica*, 24(4):471 – 483, 1988.

[48] John Hauser, Shankar Sastry, and George Meyer. Nonlinear control design for slightly non-minimum phase systems: Application to V/STOL aircraft. *Automatica*, 28(4):665 – 679, 1992.

[49] Jacob Reiner, Gary J. Balas, and William L. Garrard. Robust dynamic inversion for control of a highly maneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 18(1):18–24, Jan-Feb 1995.

[50] Joseph S. Brinker and Kevin A. Wise. Stability and flying qualities robustness of a dynamic inversion aircraft control law. *Journal of Guidance, Control, and Dynamics*, 19(6):1270–1277, Nov-Dec 1996.

[51] James M. Buffington and Dale F. Enns. Flight control for mixed-amplitude commands. *International Journal of Control*, 68(6):1209–1230, 1997.

[52] Michael B. McFarland and Shaheen M. Hoque. Robustness of a nonlinear missile autopilot designed using dynamic inversion. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, pages 258–267, Denver, CO, 14-17 August 2000.

[53] Qian Wang and Robert F. Stengel. Robust nonlinear control of a hypersonic aircraft. *Journal of Guidance, Control, and Dynamics*, 23(4):577–585, Jul-Aug 2000.

[54] Qian Wang and R.F. Stengel. Robust nonlinear flight control of a high-performance aircraft. *Control Systems Technology, IEEE Transactions on*, 13(1):15–26, 2005.

[55] Daigoro Ito, Donald T. Ward, and John Valasek. Robust dynamic inversion controller design and analysis for the X-38. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1871–1881, Montreal, Canada, 6-9 August 2001.

[56] David B. Doman and Anhtuan D. Ngo. Dynamic inversion-based adaptive/reconfigurable control of the X-33 on ascent. *Journal of Guidance, Control, and Dynamics*, 25(2):275–284, Mar-Apr 2002.

[57] Wen-Hua Chen. Nonlinear disturbance observer-enhanced dynamic inversion control of missiles. *Journal of Guidance, Control, and Dynamics*, 26(1):161–166, Jan-Feb 2003.

[58] E.M. Wallner and K.H. Well. Attitude control of a reentry vehicle with internal dynamics. *Journal of Guidance, Control, and Dynamics*, 26(6):846–854, Nov-Dec 2003.

[59] C. Papageorgiou and K. Glover. Robustness analysis of nonlinear flight controllers. *Journal of Guidance, Control, and Dynamics*, 28(4):639–648, Jul-Aug 2005.

[60] M.M. Moghaddam and S.F. Moosavi. Robust maneuvering control design of an aircraft via dynamic inversion and μ -synthesis. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 219(1):11–18, 2005.

[61] W.R. van Soest, Q.P. Chu, and J.A. Mulder. Combined feedback linearization and constrained model predictive control for entry flight. *Journal of Guidance, Control, and Dynamics*, 29(2):427–434, Mar-Apr 2006.

[62] E. Mooij, K.D. Mease, and J. Benito. Robust re-entry guidance and control system design and analysis. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, 2007.

[63] H.B. Chen and S.G. Zhang. Robust dynamic inversion flight control law design. In *2nd International Symposium on Systems and Control in Aerospace and Astronautics.* ISSCAA, 10-12 Dec 2008.

[64] S.E. Talole and S.B. Phadke. Robust inputoutput linearisation using uncertainty and disturbance estimation. *International Journal of Control*, 82(10):1794–1803, 2009.

[65] S. Sieberling, Q.P. Chu, and J.A. Mulder. Robust flight control using incremental nonlinear dynamic inversion and angular acceleration prediction. *Journal of Guidance, Control, and Dynamics*, 33(6):1732–1742, Nov-Dec 2010.

[66] Shanwei Su and Yan Lin. Robust output tracking control of a class of nonminimum phase systems and application to vtol aircraft. *International Journal of Control*, 84(11):1858–1872, 2011.

[67] J. Kawaguchi, T. Ninomiya, and Y. Miyazawa. Stochastic approach to robust flight control design using hierarchy-structured dynamic inversion. *Journal of Guidance, Control, and Dynamics*, 34(5):1573–1576, 2011.

[68] S.A. Snell, D.F. Enns, and W.L. Garrard. Nonlinear inversion flight control for a supermaneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 15(4):976–984, 1992.

[69] S.N. Singh, M.L. Steinberg, and R.D. DiGirolamo. Nonlinear predictive control of feedback linearizable systems and flight control design. *Journal of Guidance, Control, and Dynamics*, 18(5):1023–1028, 1995.

[70] M.M. Monahemi and M. Krstic. Control of wing rock motion using adaptive feedback linearization. *Journal of Guidance, Control, and Dynamics*, 19:905–912, 1996.

[71] S. Bennani and G. Looye. Flight control law design for a civil aircraft using robust dynamic inversion. In *Proceedings of the IEEE/SMC CESA '98*, volume 1, pages 998–1004. IEEE Publications, 1998.

[72] Y. Patel and P.R. Smith. Translational motion control of vertical takeoff aircraft using nonlinear dynamic inversion. *Journal of Guidance, Control, and Dynamics*, 21(1):179–182, 1998.

[73] C. Schumacher and P.P. Khargonekar. Missile autopilot designs using hinf control with gain scheduling and dynamic inversion. *Journal of Guidance, Control, and Dynamics*, 21(2):234–243, 1998.

[74] D.M. Littleboy and P.R. Smith. Using bifurcation methods to aid nonlinear dynamic inversion control law design. *Journal of Guidance, Control, and Dynamics*, 21(4):632–638, 1998.

[75] S. Bharadwaj, A.V. Rao, and K.D. Mease. Entry trajectory tracking law via feedback linearization. *Journal of Guidance, Control, and Dynamics*, 21(5):726–732, 1998.

[76] Z.M. Smit and I.K. Craig. Robust flight controller design using loop-shaping and dynamic inversion techniques, 1998. AIAA 98-4132.

[77] P.R. Smith and A. Berry. Flight test experience of a non-linear dynamic inversion control law on the vaac harrier, August 2000.

[78] G. Papageorgiou and R.A. Hyde. Analysing the stability of ndi-based flight controllers with lpv methods. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, August 2001. AIAA Paper 2001-4039.

[79] R. Wacker, D. Enns, D. Bugajski, S. Munday, and S. Merkle. X-38 application of dynamic inversion flight control. *Advances in the Astronautical Sciences*, 107:233–250, 2001.

[80] R.R. da Costa, Q.P. Chu, and J.A. Mulder. Re-entry flight controller design using nonlinear dynamic inversion. *Journal of Spacecraft and Rockets*, 40(1):64–70, 2003.

[81] S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control: Analysis and Design*. Wiley, 2005.

[82] M. Athans. A tutorial on the LQG/LTR method. In *Proceedings of the American Control Conference*, volume 2, pages 1289–1296, Seattle, WA, 1986. IEEE.

[83] L.E. Ghaoui, A. Carrier, and A.E. Bryson. Linear quadratic minimax controllers. *Journal of Guidance, Control, and Dynamics*, 15(4):953–961, Jul-Aug 1992.

[84] S. Tarbouriech, G. Garcia, and D. Henrion. Local stabilization of linear systems with position and rate bounded actuators. In *Proceedings of the 14th Triennial World Congress*, pages 459–464, Beijing, P.R.China, 1999. F-2e-02-5.

[85] R. Reichert. Modern robust control for missile autopilot design. In *Proceedings of Automatic Control Conference*, pages 2368–2373. American Automatic Control Council, 1990.

[86] S. Boyd, L.E. Ghaoui, E. Feron, and V. Balakrishnan. *Linear Matrix Inequalities in System and Control Theory*. SIAM Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994.

[87] Laurent El Ghaoui and Gérard Scorletti. Control of rational systems using linear-fractional representations and linear matrix inequalities. *Automatica*, 32(9):1273 – 1284, 1996.

[88] George Papageorgiou and Michal Polansky. Tuning a dynamic inversion pitch axis autopilot using McFarlane - Glover loop shaping. *Optimal Control Applications and Methods*, 30(3):287–308, 2009.

[89] L.B. Freidovich and H.K. Khalil. Robust feedback linearization using extended high-gain observers. In *Decision and Control, 2006 45th IEEE Conference on*, pages 983–988, Dec 2006.

[90] S. Iqbal, C. Edwards, and A.I. Bhatti. Robust feedback linearization using higher order sliding mode observer. In *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*, pages 7968–7973, Dec 2011.

[91] M.A. Henson and M.J. Kurtz. Input-output linearizing control of constrained nonlinear processes. *Journal of Process Control*, 7(1):3–17, 1997.

[92] Shr-Shiung Hu, Pao-Hwa Yang, J. Y. Juang, and B. C. Chang. Robust nonlinear ship course-keeping control by h i/o linearization and -synthesis. *International Journal of Robust and Nonlinear Control*, 13(1):55–70, 2003.

[93] B.C. Chang, H. Kwatny, and Shr-Shiung Hu. An application of robust feedback linearization to a ball and beam control problem. In *Control Applications, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 1, pages 694–698 vol.1, Sep 1998.

[94] Toshiharu Sugie and Kenji Fujimoto. Controller design for an inverted pendulum based on approximate linearization. *International Journal of Robust and Nonlinear Control*, 8(7):585–597, 1998.

[95] Yan Liu and Dirk Söffker. Robust control approach for inputoutput linearizable nonlinear systems using high-gain disturbance observer. *International Journal of Robust and Nonlinear Control*, 24(2):326–339, 2014.

[96] J.J.E. Slotine and J.K. Hedrick. Robust input-output feedback linearization. *International Journal of Control*, 57:1133–1139, 1993.

[97] R.B. Fernandez. Robust feedback linearization through sliding mode control. In *Proceedings of the 29th IEEE Conference on Decision and Control*, pages 3398–3399, December 1990.

[98] Hebertt Sira-Ramírez, Miguel Rios-Bolivar, and Alan S. I. Zinober. Adaptive dynamical inputoutput linearization of dc to dc power converters: A backstepping approach. *International Journal of Robust and Nonlinear Control*, 7(3):279–296, 1997.

[99] M. Barkhordary, S.M.H. Nabavi, M.R.J. Motlagh, and A. Kazemi. A robust feedback linearization controller design for thyristor control series capacitor to damp electromechanical oscillations. In *Industrial Technology, 2006. ICIT 2006. IEEE International Conference on*, pages 2587–2592, Dec 2006.

[100] Kamesh Subbarao and Sam James Welsh. Nonlinear control of motion synchronization for satellite proximity operations. *Journal of Guidance, Control, and Dynamics*, 31(5):1284–1294, Sep 2008.

[101] Corey J. Schumacher, Pramrod P. Khargonekar, and N. Harris McClamroch. Stability analysis of dynamic inversion controllers using time-scale separation. In *AIAA Guidance, Navigation and Control Conference and Exhibit*, Guidance, Navigation, and Control and Co-located Conferences, pages 1217–1223. American Institute of Aeronautics and Astronautics, American Institute of Aeronautics and Astronautics, August 10-12 1998. AIAA-98-4322.

[102] Rafael Martínez-Guerra, Rodolfo Suarez, and Jesús De León-Morales. Asymptotic output tracking of a class of nonlinear systems by means of an observer. *International Journal of Robust and Nonlinear Control*, 11(4):373–391, 2001.

[103] G. Herrmann, P. P. Menon, M. C. Turner, D. G. Bates, and I. Postlethwaite. Anti-windup synthesis for nonlinear dynamic inversion control schemes. *International Journal of Robust and Nonlinear Control*, 20(13):1465–1482, 2010.

[104] H.R. Karimi and M.R.J. Motlagh. Robust feedback linearization control for a non linearizable mimo nonlinear system in the presence of model uncertainties. In *Service Operations and Logistics, and Informatics, 2006. SOLI '06. IEEE International Conference on*, pages 965–970, June 2006.

[105] S.F. Atashzar, H.A. Talebi, and F. Towhidkhah. A robust feedback linearization approach for tracking control of flexible-link manipulators using an ekf disturbance estimator. In *Industrial Electronics (ISIE), 2010 IEEE International Symposium on*, pages 1791–1796, July 2010.

[106] Tetsujiro Ninomiya, Isao Yamaguchi, and Takashi Kida. Feedback control of plants driven by nonlinear actuators via input-state linearization. *Journal of Guidance, Control, and Dynamics*, 29(1):20–24, Jan 2006.

[107] P. Leland, R. Robust feedback linearization in the presence of fuzzy uncertainty. In *Proceedings of the American Control Conference*, pages 3751–3755, June 1997.

[108] Qian Wang and Robert F. Stengel. Robust control of nonlinear systems with parametric uncertainty. *Automatica*, 38(9):1591 – 1599, 2002.

[109] B. Charlet, J. Lévine, and R. Marino. On dynamic feedback linearization. *Systems & Control Letters*, 13(2):143 – 151, 1989.

[110] Wei Kang. Approximate linearization of nonlinear control systems. *Systems & Control Letters*, 23(1):43 – 52, 1994.

[111] A. Isidori and A.J. Krener. On feedback equivalence of nonlinear systems. *Systems & Control Letters*, 2(2):118 – 121, 1982.

[112] Arthur J Krener and Alberto Isidori. Linearization by output injection and nonlinear observers. *Systems & Control Letters*, 3(1):47 – 52, 1983.

[113] Riccardo Marino and Patrizio Tomei. Dynamic output feedback linearization and global stabilization. *Systems & Control Letters*, 17(2):115 – 121, 1991.

[114] S. Devasia. Robust inversion-based feedforward controllers for output tracking under plant uncertainty. In *American Control Conference, 2000. Proceedings of the 2000*, volume 1, pages 497–502 vol.1, Sep 2000.

[115] L.R. Hunt and G. Meyer. Stable inversion for nonlinear systems. *Automatica*, 33(8):1549–1554, 1997.

[116] A. Pavlov and K.Y. Pettersen. Stable inversion of non-minimum phase nonlinear systems: a convergent systems approach. In *Proceedings of the 46th Conference on Decision and Control*, pages 3995–4000, Dec 12-14 2007.

[117] S. Devasia, Degang Chen, and B. Paden. Nonlinear inversion-based output tracking. *Automatic Control, IEEE Transactions on*, 41(7):930–942, Jul 1996.

[118] Ruyi Yuan, Yafei Chang, Guoliang Fan, and Jianqiang Yi. Stability robustness of nonlinear dynamic inversion based controller. In *Robotics and Biomimetics (ROBIO), 2013 IEEE International Conference on*, pages 328–333, Dec 2013.

[119] P. Norton and E. Prempain. Robustness analysis of feedback linearisation with robust state estimation for a nonlinear missile model. In *Control Conference (ECC), 2013 European*, pages 4220–4225, July 2013.

[120] R.T. Reichert. Robust autopilot design using μ-synthesis. In *American Control Conference, 1990*, pages 2368–2373, May 1990.

[121] R.A. Nichols, R.T. Reichert, and W.J. Rugh. Gain scheduling for h-infinity controllers: a flight control example. *Control Systems Technology, IEEE Transactions on*, 1(2):69–79, Jun 1993.

[122] Emmanuel Prempain and Ian Postlethwaite. and performance analysis and gain-scheduling synthesis for parameter-dependent systems. *Automatica*, 44(8):2081 – 2089, 2008.

[123] Paulo C. Pellanda, Pierre Apkarian, and Hoang Duong Tuan. Missile autopilot design via a multi-channel lft/lpv control method. *International Journal of Robust and Nonlinear Control*, 12(1):1–20, 2002.

[124] M. Sidoryuk, M. Goman, S. Kendrick, D. Walker, and P. Perfect. *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*, chapter An LPV Control Law Design and Evaluation for the ADMIRE Model, pages 197–229. Springer, 2007.

[125] Kunpeng Sun and A. Packard. Robust H2 and H∞ filters for uncertain LFT systems. *IEEE Trans. Autom. Control*, 50(5):715 – 720, May 2005.

[126] K. Zhou, J.C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice Hall, 1996.

[127] D.G. Luenberger. Observers for multivariable systems. *Automatic Control, IEEE Transactions on*, 11(2):190–197, Apr 1966.

[128] W.S. Levine. *The Control Systems Handbook, Second Edition: Control System Advanced Methods*. Electrical Engineering Handbook. CRC Press, 2011.

[129] F.E. Thau. Observing the state of nonlinear dynamic systems. *International Journal of Control*, 17():471 – 479, 1973.

[130] Shauying R. Kou, David L. Elliott, and Tzyh Jong Tarn. Exponential observers for nonlinear dynamic systems. *Information and Control*, 29(3):204 – 216, 1975.

[131] O. Toker and H. Ozbay. On the np-hardness of solving bilinear matrix inequalities and simultaneous stabilization with static output feedback. In *American Control Conference, Proceedings of the 1995*, volume 4, pages 2525–2526 vol.4, Jun 1995.

[132] Aharon Ben-Tal and Michael Zibulevsky. Penalty/barrier multiplier methods for convex programming problems. *SIAM Journal on Optimization*, 7(2):347–366, 1997.

[133] H. D. Tuan and P. Apkarian. Low nonconvexity-rank bilinear matrix inequalities: algorithms and applications in robust controller and structure designs. *IEEE Transactions on Automatic Control*, 45(11):2111–2117, Nov 2000.

[134] Michal Kovara and Michael Stingl. Pennon: A code for convex nonlinear and semidefinite programming. *Optimization Methods and Software*, 18(3):317–333, 2003.

[135] C. Papageorgiou and M. C. Smith. Positive real synthesis using matrix inequalities for mechanical networks: application to vehicle suspension. *IEEE Transactions on Control Systems Technology*, 14(3):423–435, May 2006.

[136] M. G. Safonov, K. C. Goh, and J. H. Ly. Control system synthesis via bilinear matrix inequalities. In *American Control Conference, 1994*, volume 1, pages 45–49 vol.1, June 1994.

[137] K.-C. Goh, M. G. Safonov, and J. H. Ly. Robust synthesis via bilinear matrix inequalities. *International Journal of Robust and Nonlinear Control*, 6(9-10):1079–1095, 1996.

[138] K. Kiriakidis. Robust stabilization of the takagi-sugeno fuzzy model via bilinear matrix inequalities. *IEEE Transactions on Fuzzy Systems*, 9(2):269–277, Apr 2001.

[139] C. H. Mayr, C. Hametner, M. Kozek, and S. Jakubek. Nonlinear stable pid controller design using local model networks. In *2012 20th Mediterranean Conference on Control Automation (MED)*, pages 842–847, July 2012.

[140] D. Henrion, J. Lofberg, M. Kocvara, and M. Stingl. Solving polynomial static output feedback problems with penbmi. In *Proceedings of the 44th IEEE Conference on Decision and Control*, pages 7581–7586, Dec 2005.

[141] D. D. Simfukwe and B. C. Pal. Robust and low order power oscillation damper design through polynomial control. *IEEE Transactions on Power Systems*, 28(2):1599–1608, May 2013.

[142] M. Dettori and C. W. Scherer. Mimo control design for a compact disc player with multiple norm specifications. *IEEE Transactions on Control Systems Technology*, 10(5):635–645, Sep 2002.

[143] G. Balas, R. Chiang, A. Packard, and M. Safonov. *Robust Control Toolbox 3: User's Guide*. The MathWorks, Inc., March 2009.

[144] L. El Ghaoui and G. Scorletti. Performance control of rational systems using linear-fractional representations and lmis. In *Decision and Control, 1994., Proceedings of the 33rd IEEE Conference on*, volume 3, pages 2792–2797 vol.3, Dec 1994.

[145] L. El Ghaoui and J.P. Folcher. Multiobjective robust control of {LTI} systems subject to unstructured perturbations. *Systems & Control Letters*, 28(1):23 – 30, 1996.

[146] Carlos E. de Souza and Alexandre Trofino. *Advances in Linear Matrix Inequality Methods in Control*, chapter A Linear Matrix Inequality Approach to the Design of Robust H2 Filters, pages 175–185. SIAM, 2000.

[147] U. Jönsson, C.-Y. Kao, A. Megretski, and A. Rantzer. *A Guide to IQC-β: A MATLAB Toolbox for Robust Stability and Performance Analysis*, August 2004.

[148] D. Bates and M. Hagström. *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*. Lecture Notes in Control and Information Sciences. Springer-Verlag Berlin Heidelberg, 2007.

[149] J. Robinson. *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*, chapter Block Backstepping for Nonlinear Flight Control Law Design, pages 231–258. Springer, 2007.

[150] P.P. Menon, D.G. Bates, and I. Postlethwaite. *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*, chapter Optimisation-Based Flight Control Law Clearance, pages 259–300. Springer, 2007.

[151] M.G. Goman, A.V. Khramtsovsky, and E.N. Kolesnikov. *Nonlinear Analysis and Synthesis Techniques for Aircraft Control*, chapter Investigation of the ADMIRE Manoeuvring Capabilities Using Qualitative Methods, pages 301–324. Springer, 2007.

[152] Peter Norton and Emmanuel Prempain. Robust input-output linearisation for an lpv model of admire. In *SICE Annual Conference (SICE), 2013 Proceedings of*, pages 685–690, Sept 2013.

[153] C.W. Scherer. LPV control and full block multipliers. *Automatica*, 37(3):361 – 375, 2001.

[154] Y. Ameho and E. Prempain. Linear parameter varying controllers for the admire aircraft longitudinal dynamics. In *American Control Conference (ACC), 2011*, pages 1315–1320, June 2011.

[155] P. Apkarian and R.J. Adams. Advanced gain-scheduling techniques for uncertain systems. *Control Systems Technology, IEEE Transactions on*, 6(1):21 –32, jan 1998.

[156] A. Megretski and A. Rantzer. System analysis via integral quadratic constraints. *IEEE Trans. Autom. Control*, 42(6):819 –830, jun 1997.

[157] K. Krishnaswamy, G. Papageorgiou, S. Glavaski, and A. Papachristodoulou. Analysis of aircraft pitch axis stability augmentation system using sum of squares optimization. In *American Control Conference, 2005. Proceedings of the 2005*, pages 1497–1502 vol. 3, June.

[158] R. F. Stengel and C. I. Marrison. Stochastic robustness synthesis for a benchmark problem. In *1992 American Control Conference*, pages 2421–2422, June 1992.

[159] E. Siva, P. Goulart, J. Maciejowski, and N. Kantas. Stability of model predictive control using markov chain monte carlo optimisation. In *2009 European Control Conference (ECC)*, pages 2851–2856, Aug 2009.

[160] O. Chaojie, J. Ju, W. Hongxin, and Z. Ziyang. Monte carlo approach to the analysis of uavs control system. In *Proceedings of 2014 IEEE Chinese Guidance, Navigation and Control Conference*, pages 458–462, Aug 2014.

[161] P. D. Vouzis, S. Collange, M. G. Arnold, and M. V. Kothare. Improving model predictive control arithmetic robustness by monte carlo simulations. *IET Control Theory Applications*, 6(8):1064–1070, May 2012.

[162] John H. Holland. *Adaptation in Natural and Artificial Systems.* Ann Arbor: The University of Michigan Press, 1975.

[163] Kerry Gallagher, Malcolm Sambridge, and Guy Drijkoningen. Genetic algorithms: An evolution from monte carlo methods for strongly nonlinear geophysical optimization problems. *Geophysical Research Letters*, 18(12):2177–2180, 12 1991.

[164] L. Davis, editor. *Genetic Algorithms and Simulated Annealing.* Pitman, London, 1990.

[165] John A Scales, Martin L Smith, and Terri L Fischer. Global optimization methods for highly multimodal inverse problems. *Journal of Computational Physics*, 101(1):228 – 229, 1992.

[166] R. Dimeo and K.Y. Lee. Boiler-turbine control system design using a genetic algorithm. *Energy Conversion, IEEE Transactions on*, 10(4):752–759, Dec 1995.

[167] A. Trebi-Ollennu and B.A. White. Multiobjective fuzzy genetic algorithm optimisation approach to nonlinear control system design. *Control Theory and Applications, IEE Proceedings* -, 144(2):137–142, Mar 1997.

[168] C. Onnen, R. Babuka, U. Kaymak, J.M. Sousa, H.B. Verbruggen, and R. Isermann. Genetic algorithms for optimization in predictive control. *Control Engineering Practice*, 5(10):1363 – 1372, 1997.

[169] Christopher I. Marrison and R.F. Stengel. Robust control system design using random search and genetic algorithms. *Automatic Control, IEEE Transactions on*, 42(6):835–839, Jun 1997.

[170] W.M. Schubert and R.F. Stengel. Parallel synthesis of robust control systems. *Control Systems Technology, IEEE Transactions on*, 6(6):701–706, Nov 1998.

[171] Marc L. Steinberg and Anthony B. Page. Nonlinear adaptive flight control with genetic algorithm design optimization. *International Journal of Robust and Nonlinear Control*, 9(14):1097–1115, 1999.

[172] Euan W. McGookin, David J. Murray-Smith, Yun Li, and Thor I. Fossen. Ship steering control system optimisation using genetic algorithms. *Control Engineering Practice*, 8(4):429 – 443, 2000.

[173] D.R. Lewin and A. Parag. A constrained genetic algorithm for decentralized control system structure selection and optimization. *Automatica*, 39(10):1801 – 1807, 2003.

[174] Leehter Yao, Wen-Chi Chang, and Rong-Liang Yen. An iterative deepening genetic algorithm for scheduling of direct load control. *Power Systems, IEEE Transactions on*, 20(3):1414–1421, Aug 2005.

[175] Rong-Jong Wai and Kuo-Ho Su. Supervisory control for linear piezoelectric ceramic motor drive using genetic algorithm. *Industrial Electronics, IEEE Transactions on*, 53(2):657–673, April 2006.

[176] M.F. Miranda, R.H.C. Takahashi, and F.G. Jota. Hierarchical approach for H_∞ robust control design: S/KS mixed sensitivity with genetic algorithm. *Control Theory Applications, IET*, 1(1):18–24, January 2007.

[177] Rong-Jong Wai and Ching-Hsiang Tu. Design of total sliding-mode-based genetic algorithm control for hybrid resonant-driven linear piezoelectric ceramic motor. *Power Electronics, IEEE Transactions on*, 22(2):563–575, March 2007.

[178] P.J Fleming and R.C Purshouse. Evolutionary algorithms in control systems engineering: a survey. *Control Engineering Practice*, 10(11):1223 – 1241, 2002.

[179] C. I. Marrison and R. F. Stengel. The use of random search and genetic algorithms to optimize stochastic robustness functions. In *American Control Conference, 1994*, volume 2, pages 1484–1489 vol.2, June 1994.

[180] M.V. Cook. *Flight Dynamics Principles: A Linear Systems Approach to Aircraft Stability and Control*. Elsevier Aerospace Engineering. Elsevier Science, 2007.

[181] R.J. Adams, J.M. Buffington, A.G. Sparks, and S.S. Banda. *Robust multivariable flight control*. Advances in industrial control. Springer-Verlag GmbH, 1994.

[182] D.E. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Artificial Intelligence. Addison-Wesley Publishing Company, 1989.

[183] G. Chesi. *Domain of Attraction: Analysis and Control via SOS Programming*. Lecture Notes in Control and Information Sciences. Springer London, 2011.

[184] G. Chesi, A. Garulli, A. Tesi, and A. Vicino. *Homogeneous Polynomial Forms for Robustness Analysis of Uncertain Systems*. Lecture Notes in Control and Information Sciences. Springer London, 2009.

[185] R.A. Horn and C.R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.