Linköping Studies in Science and Technology. Dissertations
No. 820

Backstepping and Control Allocation with Applications to Flight Control

Ola Härkegård

Department of Electrical Engineering
Linköping University, SE–581 83 Linköping, Sweden

Linköping 2003

Backstepping and Control Allocation with Applications to Flight Control

ola@isy.liu.se
http://www.control.isy.liu.se
Division of Automatic Control
Department of Electrical Engineering
Linköping University
SE–581 83 Linköping
Sweden

ISBN 91-7373-647-3 ISSN 0345-7524

Printed by UniTryck, Linköping, Sweden 2003

To Eva

Abstract

In this thesis we study a number of nonlinear control problems motivated by their appearance in flight control. The results are presented in a general framework and can also be applied to other areas. The two main topics are backstepping and control allocation.

Backstepping is a nonlinear control design method that provides an alternative to feedback linearization. Here, backstepping is used to derive robust linear control laws for two nonlinear systems, related to angle of attack control and flight path angle control, respectively. The resulting control laws require less modeling information than corresponding designs based on feedback linearization, and achieve global stability in cases where feedback linearization can only be performed locally. Further, a method for backstepping control of a rigid body is developed, based on a vector description of the dynamics. We also discuss how to augment an existing nonlinear controller to suppress constant input disturbances. Two methods, based on adaptive backstepping and nonlinear observer design, are proposed.

Control allocation deals with actuator utilization for overactuated systems. In this thesis we pose the control allocation problem as a constrained least squares problem to account for actuator position and rate constraints. Efficient solvers based on active set methods are developed with similar complexity to existing, approximate, pseudoinverse methods. A method for dynamic control allocation is also proposed which enables a frequency dependent control distribution among the actuators to be designed. Further, the relationship between control allocation and linear quadratic control is investigated. It is shown that under certain circumstances, the two techniques give the same freedom in distributing the control effort among the actuators. An advantage of control allocation, however, is that since the actuator constraints are considered, the control capabilities of the actuator suite can be fully exploited.

Acknowledgments

Many people have contributed in a variety of ways to the production of this thesis.

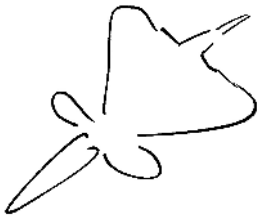
First of all, I would like to thank Professor Lennart Ljung for drafting me to the Control and Communication group in Linköping, hereby giving me the opportunity to perform research within a most professional, ambitious, and inspiring group of people. I also want to thank my supervisor Professor Torkel Glad for introducing me to backstepping control design and for giving me great freedom in selecting my research topics, and Karin Ståhl Gunnarsson at SAAB AB for guidance and expertise within aircraft flight control.

Mikael Norrlöf, Thomas Schön, Fredrik Tjärnström, Johan Sjöberg, Jonas Jansson, and Magnus Åkerblad "closed the loop" by proofreading the thesis, providing valuable feedback that significantly increased the quality and robustness of the result.

Professor Emeritus Åke Björck at the Department of Mathematics provided much appreciated input on numerical methods for constrained least squares problems.

Jacob Roll, Mikael Norrlöf, Måns [mawns] Östring, and Martin Enqvist all had the misfortune of having their office near mine, and have therefore been the targets of many ill-posed questions over the years. Fredrik "Running Man" Tjärnström got his fair share too, despite his more strategic position down the hall. Although I know he doesn't want me to, I must also thank my good friend Björn Johansson at the Computer Vision Laboratory for always having five minutes to spare, and for his great skills in matrix massage.

This work was sponsored by the graduate school ECSEL.



Ola Härkegård

Linköping, April 2003

Contents

II Control Allocation 103

Chapter 1

Introduction

Modern fighter aircraft offer a wealth of interesting and challenging control problems. The governing dynamics are nonlinear, the aerodynamics are uncertain, and the control input is constrained by position and rate limits. Despite these conditions, the performance requirements on a flight control system are high. Stability is crucial, the aircraft must be able to operate under a wide range of conditions, and for maximum maneuverability the control system should utilize the full control capabilities of the actuator suite.

In this thesis we consider some of these control problems and develop theories and methods to solve them. Although the primary application is flight control, the results are presented in a general framework and can be applied also to other areas. The two main topics are

1. backstepping control of nonlinear systems;

2. actuator redundancy management using control allocation.

1.1 Background

The interplay between automatic control and manned flight goes back a long time. As a consequence, a large number of design methods have been applied to flight control, ranging from PID control to model predictive control, see, e.g., Magni et al. (1997). Two of the most successful methods are linear quadratic (LQ) control, used in the Swedish fighter JAS 39 Gripen, and feedback linearization, also known as nonlinear dynamic inversion (NDI).

In LQ control, a linear aircraft model is first constructed by linearizing the dynamics around some operating point. A linear feedback law is then designed by minimizing a quadratic performance index involving the aircraft state and the control inputs, given by the deflections of the aerodynamic control surfaces. Since the aircraft dynamics vary with speed and altitude, this procedure is repeated for a number of operating points, rendering several linear control laws, each tailored

for a specific flight case. So called gain scheduling (Rugh and Shamma 2000) is then used to blend these control laws together using interpolation.

The main benefit of this strategy is that it is based on linear control theory. This allows the designer to utilize all the standard tools for frequency analysis, robustness analysis, etc. The price to pay is that nonlinear effects such as nonlinearities in the aerodynamics, occurring in particular at high angles of attack, and the cross-couplings between longitudinal and lateral motion, are neglected in the model and therefore not accounted for in the control design. This motivates the use of nonlinear design methods.

Feedback linearization (Isidori 1995) is a nonlinear design method that can explicitly handle these types of nonlinearities. Using nonlinear feedback, the influence of the nonlinearities on the controlled variables are cancelled and a linear closed loop system is achieved. The variations of the dynamics with speed and altitude can also be dealt with this way, which means that a single controller can be used for all flight cases. This method has received much attention by the flight control community including the works of Meyer et al. (1984), Lane and Stengel (1988), and Enns et al. (1994).

To perform feedback linearization, the system nonlinearities must be completely known, including their derivatives up to some order depending on how they enter the dynamics. This is a potential problem in flight control since the aerodynamic forces and moments cannot be modeled precisely. To achieve robustness against such model errors, Reiner et al. (1996) propose to augment the feedback linearization controller with a linear, robust controller. A different approach is to design control laws that rely on less precise model information.

Backstepping control design (Krstić et al. 1995) constitutes an alternative to feedback linearization. With backstepping, system nonlinearities do not have to be cancelled in the control law. How to deal with nonlinearities instead becomes a design choice. If a nonlinearity acts stabilizing, and thus in a sense is useful, it may be retained in the closed loop system. This leads to robustness to model errors and less control effort may be needed to control the system. This was illustrated in Krstić et al. (1998) where backstepping was applied to jet engine control.

A weakness of backstepping as well as feedback linearization is that they lack support for dealing with actuator redundancy. The resulting control laws specify which total control effort to produce, but not how to produce it. For performance reasons, and also for safety reasons, modern aircraft are typically over-actuated in the sense that there are several combinations of control surface deflections that will give the same aircraft response. In LQ control, the control distribution among the actuators is determined by a weighting matrix in the optimization criterion. To distribute the total control demand from a nonlinear controller among the available actuators, control allocation can be used.

In flight control, performing control allocation means to compute control surface deflections such that some specified aerodynamic moments in pitch, roll, and yaw are produced, see, e.g., Durham (1993). If there are more than three control surfaces, and the aerodynamic moments are assumed to be affine in the control deflections, this gives an underdetermined linear system of equations to be solved.

A common way to make the choice of control input unique is to pick the combination that minimizes some quadratic cost, besides producing the desired moments. A motivation for this is that the optimal solution then can be written in closed form as a weighted pseudoinverse solution.

An advantage of performing control allocation separately, rather than letting the control distribution be decided by the feedback law as in LQ control, is that actuator position and rate limits can be considered. If one actuator saturates, the remaining actuators can be used to make up the difference. Including such limits gives a constrained least squares problem to be solved at each sampling instant. Most solvers proposed for this problem start from the nominal pseudoinverse solution of the unconstrained problem and then try to adjust it to the constraints in an iterative manner, see, e.g., Virnig and Bodden (1994) and Bordignon (1996). However, none of these so called pseudoinverse methods are guaranteed to find the optimal solution in general.

To use control allocation, the actuator dynamics must be neglected so that the relationship between the control inputs and the resulting total control effort becomes a static mapping. To compensate for this approximation, filtering can be incorporated into the control allocation procedure, so that the high frequency components of the total control effort are produced by the fastest actuators. Davidson et al. (2001) propose such a strategy where the total control effort is partitioned into high and low frequency components. These components are then allocated separately which means that the full control capabilities of the actuator suite may not be utilized.

1.2 Objectives of the Thesis

There are two primary objectives of this thesis, namely (a) to investigate the use of backstepping for flight control design, and (b) to develop new tools and efficient solvers for optimization based control allocation.

In the first part of the thesis the following problems are treated:

- How can backstepping be used to design flight control laws that require minimal modeling information and achieve stability even at high angles of attack?
- Can backstepping be used to control the motion of a generic rigid body?
- How can an existing nonlinear controller be augmented to suppress constant input disturbances?

In the second part, the following control allocation issues are dealt with within a least squares framework:

- Can standard methods from numerical optimization be used for real-time control allocation?
- How can filtering be incorporated into the control allocation procedure while taking actuator constraints into account?

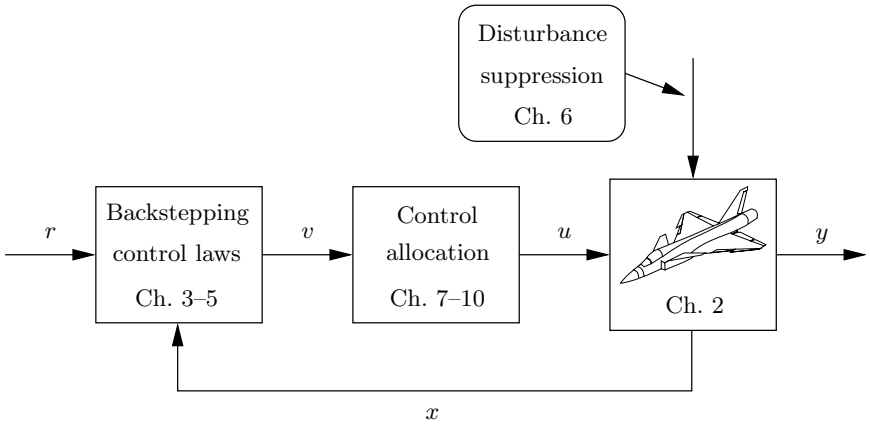


Figure 1.1: Block diagram representation of the organization of the thesis.

- What is the relationship between using control allocation and linear quadratic control to distribute the control effort among the actuators?

1.3 Thesis Outline

This thesis consists of three parts, preceded by an introduction to aircraft control in Chapter 2. Part I and Part II contain the theoretical contributions of the thesis and deal with backstepping and control allocation, respectively. The design tools developed in these parts are then combined and evaluated in Part III. With the exception of Part III, the organization of the thesis is shown in Figure 1.1.

The first two parts have the same structure. The underlying theory is presented and an introductory chapter, where relevant publications are also reviewed. The following three chapters then each deal with one of the problems stated above. A more detailed outline of these parts can be found in Section 3.7 and Section 7.5, respectively. In the last part, a simplified flight control system is implemented and evaluated using the ADMIRE model (ADMIRE ver. 3.4h 2003), maintained by the Swedish Defence Research Agency (FOI). In the final chapter, some general conclusions regarding the results presented in this thesis are stated.

Although this thesis is written as a monograph, it can be viewed as a collection of edited versions of previously published papers, listed on the next page. As a result, these chapters can be read in any order that the reader may prefer.

Connections between thesis chapters and previously published papers

Chapter 4: O. Härkegård and S. T. Glad. Flight control design using backstepping. In *Proc. of the 5th IFAC Symposium on Nonlinear Control Systems*, pages 259–264, St. Petersburg, Russia, July 2001b,

and

O. Härkegård and S. T. Glad. A backstepping design for flight path angle control. In *Proc. of the 39th Conference on Decision and Control*, pages 3570–3575, Sydney, Australia, Dec. 2000.

Chapter 5: S. T. Glad and O. Härkegård. Backstepping control of a rigid body. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 3944–3945, Las Vegas, NV, Dec. 2002.

Chapter 6: O. Härkegård and S. T. Glad. Control of systems with input nonlinearities and uncertainties: An adaptive approach. In *Proc. of the European Control Conference*, pages 1912–1917, Porto, Portugal, Sept. 2001a.

Chapter 8: O. Härkegård. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 1295–1300, Las Vegas, NV, Dec. 2002b.

Chapter 9: O. Härkegård. Dynamic control allocation using constrained quadratic programming. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002a.

Chapter 10: O. Härkegård. Resolving actuator redundancy—control allocation vs linear quadratic control. In *Proc. of the European Control Conference*, Cambridge, UK, Sept. 2003. Accepted for presentation.

Chapter 2

Aircraft Primer

The purpose of this aircraft primer is to introduce the reader to modern fighter aircraft from an automatic control perspective. This includes developing a dynamical model, describing the control variables available, and reviewing the most common control objectives. Based on this we also discuss which model approximations must be made in order to use backstepping and control allocation, and compare these with the approximations made when linear control or feedback linearization is used. The nomenclature introduced in this chapter is summarized in Appendix A.

There is a substantial literature on flight dynamics and the presentation in this chapter is mainly based on the textbooks by Stevens and Lewis (1992), Nelson (1998), and Boiffier (1998).

In Section 2.1, the standard variables used to describe the motion of an aircraft are introduced and a nonlinear dynamical model is developed. In Section 2.2, the choice of controlled variables for different purposes is considered. Model approximations for different control design methods are discussed in Section 2.3. Finally, the ADMIRE model, a non-classified realistic model of a fighter aircraft, is presented in Section 2.4.

2.1 The Dynamics of Flight

We begin by deriving a nonlinear dynamical model of an aircraft. We will consider the aircraft as a rigid body and neglect any structural flexibilities. Earth is considered flat, and regarded as an inertial system so that Newton's laws of motion can be applied.

2.1.1 Vector Notation and Differentiation

In our presentation we will make a distinction between a vector and its representation in a certain coordinate frame, where a frame is a right-handed triple of unit

vectors. In a frame A, a vector $\mathbf{v}$ can be expressed as

$$\mathbf{v} = x_A\hat{\mathbf{x}}_A + y_A\hat{\mathbf{y}}_A + z_A\hat{\mathbf{z}}_A = \begin{pmatrix}\hat{\mathbf{x}}_A & \hat{\mathbf{y}}_A & \hat{\mathbf{z}}_A\end{pmatrix}\begin{pmatrix}x_A\\ y_A\\ z_A\end{pmatrix} = \mathbf{e}_A v_A$$

where $\mathbf{e}_A$ contains the basis vectors of the frame, and the component vector v_A contains the coordinates of $\mathbf{v}$ in frame A. In a different frame B, the same vector can be expressed as

$$\mathbf{v} = x_B\hat{\mathbf{x}}_B + y_B\hat{\mathbf{y}}_B + z_B\hat{\mathbf{z}}_B = \mathbf{e}_B v_B$$

We will use bold face style for vectors and italic style for component vectors.

Assume now that frame B rotates with an angular velocity of $\boldsymbol{\omega}$ with respect to frame A. Then the theorem of Coriolis (Stevens and Lewis 1992, p. 17) gives us

$$\left.\frac{d}{dt}\right|_A \mathbf{v} = \left.\frac{d}{dt}\right|_B \mathbf{v} + \boldsymbol{\omega}\times\mathbf{v} \tag{2.1}$$

where the subscript on the derivative operator indicates with respect to which frame the derivative is taken. Since the basis vectors in $\mathbf{e}_B$ are fixed with respect to frame B we have that

$$\left.\frac{d}{dt}\right|_B \mathbf{v} = \left.\frac{d}{dt}\right|_B (\mathbf{e}_B v_B) = \mathbf{e}_B\dot{v}_B$$

where

$$\dot{v}_B = \begin{pmatrix}\dot{x}_B\\ \dot{y}_B\\ \dot{z}_B\end{pmatrix}$$

contains the time derivatives of the components of $\mathbf{v}$ in frame B.

2.1.2 Coordinate Frames

The two coordinate frames most frequently used to describe the motion of an aircraft are the Earth-fixed frame i, and the body-fixed frame b, see Figure 2.1. In the Earth-fixed frame, the axes point north, east, and down. This frame is useful for describing the position and orientation of the aircraft. In the body-fixed frame, the origin is at the aircraft center of gravity and the axes point forward, over the right wing, and down (relative to the pilot). In this frame, the inertia matrix of the aircraft is fixed which makes this frame suitable for describing angular motions.

Another coordinate frame of interest is the wind-axes coordinate frame w. In this frame the x-axis is directed along the velocity vector of the aircraft, $\mathbf{V}$, as depicted in Figure 2.1. The orientation of this frame relative to the body-fixed frame is determined by the angle of attack α and the sideslip angle β. Given any

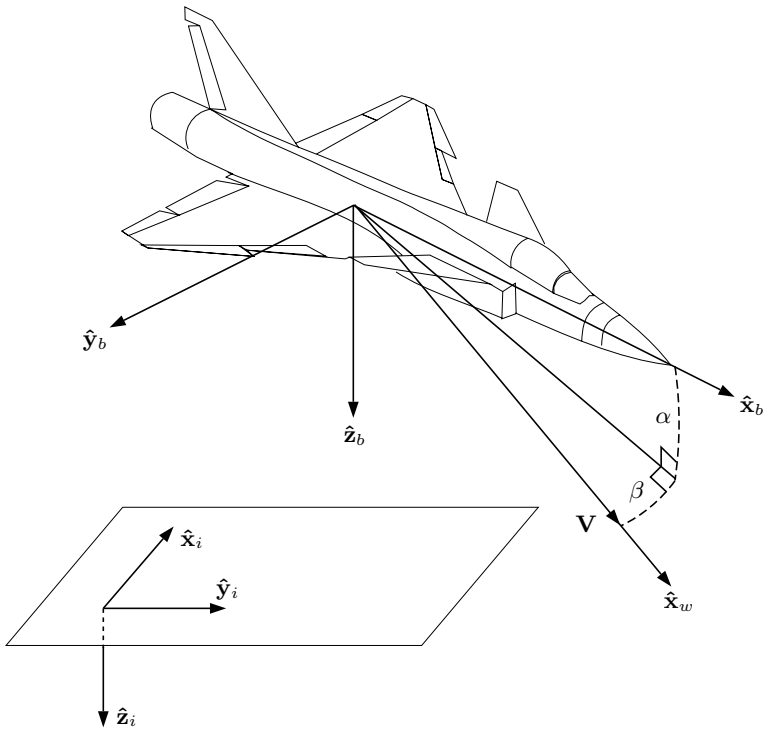


Figure 2.1: Illustration of the inertial, Earth-fixed coordinate frame i, and the body-fixed frame b. Also shown is the x-axis of the wind-axes frame w. In the figure, α and β are both positive.

vector

$$\mathbf{v} = \mathbf{e}_b v_b = \mathbf{e}_w v_w$$

its component vectors in the two frames are related by

$$\begin{aligned} v_w &= T_{wb} v_b \\ v_b &= T_{bw} v_w = T_{wb}^T v_w \end{aligned} \tag{2.2}$$

where

$$T_{wb} = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix}$$

Since the body-fixed frame is the most frequently used, we will drop the subscript b for component vectors related to this frame and simply write $\mathbf{v} = \mathbf{e}_b v$.

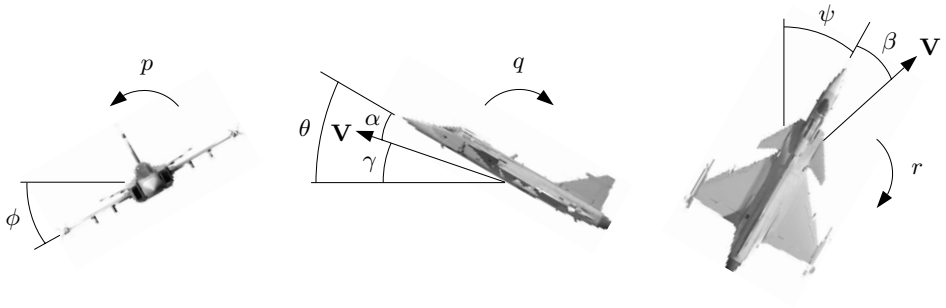


Figure 2.2: Illustration of the aircraft orientation angles ϕ, θ, and ψ, the aerodynamic angles α and β, and the angular rates p, q, and r. In the figure, all angles are positive.

2.1.3 Aircraft Variables

Considering the aircraft as a rigid body, its motion can be described by its position, orientation, velocity, and angular velocity over time.

Position The position vector $\mathbf{p}$ is given by

$$\mathbf{p} = \mathbf{e}_i \begin{pmatrix} p_N & p_E & -h \end{pmatrix}^T$$

in the Earth-fixed coordinate frame, where p_N = position north, p_E = position east, and h = altitude.

Orientation The orientation of the aircraft can be represented by the Euler angles

$$\Phi = \begin{pmatrix} \phi & \theta & \psi \end{pmatrix}^T$$

where ϕ = roll angle, θ = pitch angle, and ψ = yaw angle, see Figure 2.2. These angles relate the body-fixed frame to the Earth-fixed frame.

Velocity The velocity vector $\mathbf{V}$ is given by

$$\mathbf{V} = \mathbf{e}_b V = \mathbf{e}_w V_w$$
$$V = \begin{pmatrix} u & v & w \end{pmatrix}^T$$
$$V_w = \begin{pmatrix} V_T & 0 & 0 \end{pmatrix}^T$$

in the body-fixed and in the wind-axes coordinate frames, respectively. Here, u = longitudinal velocity, v = lateral velocity, w = normal velocity, and V_T = total velocity (true airspeed). Using (2.2) gives

$$V = T_{bw} V_w = V_T \begin{pmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \end{pmatrix}^T$$

Conversely, we have that

$$\begin{aligned} V_T &= \sqrt{u^2 + v^2 + w^2} \\ \alpha &= \arctan \frac{w}{u} \\ \beta &= \arcsin \frac{v}{V_T} \end{aligned} \tag{2.3}$$

When $\beta = \phi = 0$ we can also define the flight path angle

$$\gamma = \theta - \alpha$$

illustrated in Figure 2.2. A general definition of the flight path angle can be found in Stevens and Lewis (1992, p. 131).

Angular velocity The angular velocity vector $\boldsymbol{\omega}$ is given by

$$\begin{aligned} \boldsymbol{\omega} &= \mathbf{e}_b \omega = \mathbf{e}_w \omega_w \\ \omega &= \begin{pmatrix} p & q & r \end{pmatrix}^T \\ \omega_w &= T_{wb} \omega = \begin{pmatrix} p_w & q_w & r_w \end{pmatrix}^T \end{aligned}$$

in the body-fixed and in the wind-axes coordinate frames, respectively. Here, $p =$ roll rate, $q =$ pitch rate, and $r =$ yaw rate. The wind-axes roll rate p_w is also known as the velocity vector roll rate since $\hat{\mathbf{x}}_w$ is parallel to the velocity vector $\mathbf{V}$, see Figure 2.1.

2.1.4 Control Variables

The control variables of an aircraft consist of the engine throttle setting and the deflections the aerodynamic control surfaces, δ. The control surfaces divert the airflow to produce aerodynamic forces and moments.

In traditional aircraft configurations, the engine provides speed control while the motions in pitch, roll, and yaw are goverened by the elevators, the ailerons, and the rudder, respectively. Modern aircraft typically have more that three control surfaces, see, e.g., Moir and Seabridge (1992, p. 36–39). This is motivated by performance issues and redundancy aspects. Figure 2.3 shows an example of a modern delta canard fighter configuration (Claréus 2001). With this layout, roll control is achieved by deflecting the elevons[1] differentially. Pitch control is achieved by combining symmetric elevon deflection, which generates a non-minimum phase response, with deflection of the canards, which produces a response in the commanded direction immediately.

The interest in high angle of attack flight has also led to the invention of thrust vectored control (TVC), see, e.g., Enns et al. (1994). Deflectable vanes are then

[1] The word "elevon" comes from merging "elevator" with "aileron".

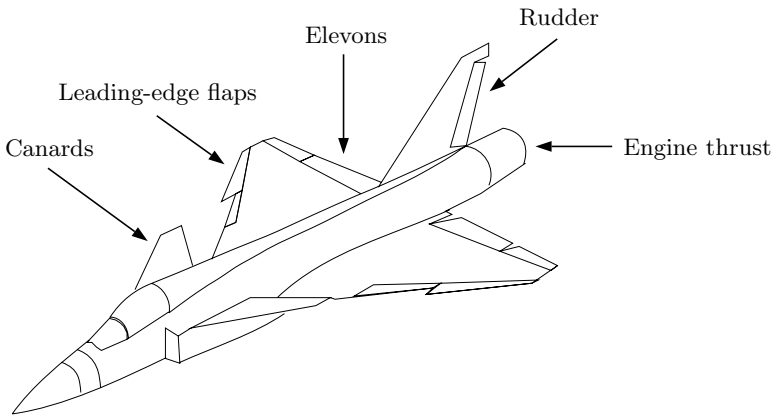


Figure 2.3: A delta canard fighter aircraft configuration.

mounted at the engine exhaust so that the engine thrust can be directed to provide additional pitching and yawing moments.

The control surfaces of a fly-by-wire aircraft are driven by servo controlled actuators to produce the deflections commanded by the flight control system, u, which are the true control variables. The servo dynamics are typically modeled as first or second order linear systems such that the deflection of the i:th control surface satisfies

$$\delta_i = G_i(s)u_i, \quad G_i(0) = 1 \tag{2.4}$$

where G_i has low-pass characteristics.

2.1.5 Rigid Body Motion

Let us now derive a model for the aircraft dynamics in terms of the variables introduced in the previous sections. Considering the aircraft as a rigid body allows us to use Newton's laws of motion to investigate the effects of external forces and moments. In the inertial, Earth-fixed coordinate frame i, Newton's second law states that (Stevens and Lewis 1992, p. 19, 28)

$$\begin{aligned} \mathbf{F} &= \left.\frac{d}{dt}\right|_i (m\mathbf{V}) \\ \mathbf{T} &= \left.\frac{d}{dt}\right|_i \mathbf{H} \end{aligned} \tag{2.5}$$

where $\mathbf{F}$ = total force, $\mathbf{T}$ = total torque, m = aircraft mass, and $\mathbf{H}$ = angular momentum of the aircraft. Using (2.1) allows us to perform the differentiation in

the body-fixed frame instead:

$$\mathbf{F} = \frac{d}{dt}\bigg|_b (m\mathbf{V}) + \boldsymbol{\omega} \times m\mathbf{V}$$
$$\mathbf{T} = \frac{d}{dt}\bigg|_b \mathbf{H} + \boldsymbol{\omega} \times \mathbf{H}$$

Since this frame is fixed relative to the aircraft, the inertia matrix I is constant. This means that the angular momentum can be expressed as

$$\mathbf{H} = \mathbf{e}_b I \omega$$
$$I = \begin{pmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{pmatrix}$$

where the zero entries are due to that the body-fixed xz-plane is a symmetry plane for most aircraft. Expressing all vectors in the body-fixed frame thus gives the following standard equations for rigid body motion in terms of velocity and angular velocity:

$$\begin{aligned} F &= m(\dot{V} + \omega \times V) \\ T &= I\dot{\omega} + \omega \times I\omega \end{aligned} \tag{2.6}$$

Corresponding equations can be derived also for the position and orientation dynamics, see Stevens and Lewis (1992). In this thesis we only need the pitch angle dynamics during level flight ($\phi = 0$) given by

$$\dot{\theta} = q$$

2.1.6 Forces and Moments

In (2.5), $\mathbf{F}$ and $\mathbf{T}$ represent the sum of the forces and moments acting on the aircraft at the center of gravity. These forces and moments spring from three major sources: gravity, engine thrust, and aerodynamic effects. Introducing

$$\begin{aligned} \mathbf{F} &= \mathbf{F}_G + \mathbf{F}_E + \mathbf{F}_A \\ \mathbf{T} &= \mathbf{T}_E + \mathbf{T}_A \end{aligned} \tag{2.7}$$

we will now investigate each of these components.

Gravity

Gravity only gives a force contribution since it acts at the aircraft center of gravity. The gravitational force mg is directed along the normal of the Earth plane and is

considered to be independent of the altitude. This yields

$$\mathbf{F}_G = \mathbf{e}_i \begin{pmatrix} 0 \\ 0 \\ mg \end{pmatrix} = \mathbf{e}_b\, mg \begin{pmatrix} -\sin\theta \\ \sin\phi\cos\theta \\ \cos\phi\cos\theta \end{pmatrix} = \mathbf{e}_w m \begin{pmatrix} g_1 \\ g_2 \\ g_3 \end{pmatrix}$$

where

$$\begin{aligned} g_1 &= g(-\cos\alpha\cos\beta\sin\theta + \sin\beta\sin\phi\cos\theta + \sin\alpha\cos\beta\cos\phi\cos\theta) \\ g_2 &= g(\cos\alpha\sin\beta\sin\theta + \cos\beta\sin\phi\cos\theta - \sin\alpha\sin\beta\cos\phi\cos\theta) \\ g_3 &= g(\sin\alpha\sin\theta + \cos\alpha\cos\phi\cos\theta) \end{aligned} \tag{2.8}$$

using (2.2).

Engine

The thrust force produced by the engine is denoted by F_T. Assuming the engine to be positioned so that the thrust acts parallel to the aircraft body x-axis yields

$$\mathbf{F}_E = \mathbf{e}_b \begin{pmatrix} F_T \\ 0 \\ 0 \end{pmatrix}$$

Also assuming the engine to be mounted so that the thrust point lies in the xz-plane of the body-fixed frame, offset from the center of gravity by z_{TP} along the z-axis gives a pitching moment

$$\mathbf{T}_E = \mathbf{e}_b \begin{pmatrix} 0 \\ F_T z_{TP} \\ 0 \end{pmatrix}$$

If TVC is used, these expressions will be somewhat different and also depend on the engine nozzle deflections.

Aerodynamics

The aerodynamic forces and moments, or the aerodynamic efforts for short, are due to the interaction between the aircraft body and the surrounding air. The size and direction of the aerodynamic efforts are determined by the amount of air diverted by the aircraft in different directions (see Anderson and Eberhardt (1999) for a discussion on various explanations to aerodynamic lift). The amount of air diverted by the aircraft is mainly decided by

- the speed and density of the airflow: V_T, ρ
- the geometry of the aircraft: S (wing area), b (wing span), $\bar{c}$ (mean aerodynamic chord)

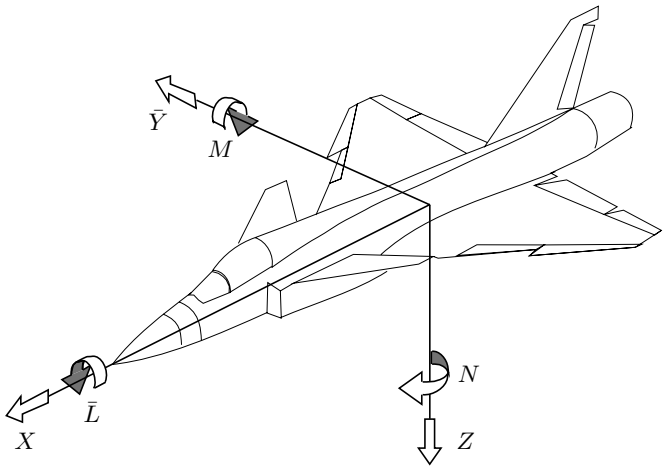


Figure 2.4: Aerodynamic forces and moments in the body-fixed coordinate frame.

- the orientation of the aircraft relative to the airflow: α, β
- the control surface deflections: δ

The aerodynamic efforts also depend on other variables, like the angular rates (p, q, r) and the time derivatives of the aerodynamic angles ($\dot{\alpha}$, $\dot{\beta}$), but these effects are not as prominent.

This motivates the standard way of modeling scalar aerodynamic forces and moments:

$$\begin{aligned} \text{Force} &= \bar{q}SC_F(\delta, \alpha, \beta, p, q, r, \dot{\alpha}, \dot{\beta}, \ldots) \\ \text{Moment} &= \bar{q}SlC_M(\delta, \alpha, \beta, p, q, r, \dot{\alpha}, \dot{\beta}, \ldots) \end{aligned} \tag{2.9}$$

where the aerodynamic pressure

$$\bar{q} = \frac{1}{2}\rho(h)V_T^2 \tag{2.10}$$

captures the density dependence and most of the speed dependence, and l is either b or $\bar{c}$. The remaining aerodynamic effects are determined by the dimensionless aerodynamic coefficients C_F and C_M. These are difficult to model analytically but can be estimated empirically through wind tunnel experiments and actual flight tests. Typically, each coefficient is written as the sum of several components, each capturing the dependence of one or more of the variables involved. These components can be represented in several ways. A common approach is to store them in look-up tables and use interpolation to compute intermediate values. In other approaches one tries to fit the data to some parameterized function.

In the body-fixed frame we introduce the components

$$\mathbf{F}_A = \mathbf{e}_b \begin{pmatrix} X \\ \bar{Y} \\ Z \end{pmatrix} \quad \text{where} \quad \begin{aligned} X &= \bar{q}SC_x \\ \bar{Y} &= \bar{q}SC_y \\ Z &= \bar{q}SC_z \end{aligned} \tag{2.11}$$

$$\mathbf{T}_A = \mathbf{e}_b \begin{pmatrix} \bar{L} \\ M \\ N \end{pmatrix} \quad \text{where} \quad \begin{aligned} \bar{L} &= \bar{q}SbC_l && \text{rolling moment} \\ M &= \bar{q}S\bar{c}C_m && \text{pitching moment} \\ N &= \bar{q}SbC_n && \text{yawing moment} \end{aligned} \tag{2.12}$$

These are illustrated in Figure 2.4. The aerodynamic forces are often expressed in the wind-axes coordinate frame:

$$\mathbf{F}_A = \mathbf{e}_w \begin{pmatrix} -D \\ Y \\ -L \end{pmatrix} \quad \text{where} \quad \begin{aligned} D &= \bar{q}SC_D && \text{drag force} \\ Y &= \bar{q}SC_Y && \text{side force} \\ L &= \bar{q}SC_L && \text{lift force} \end{aligned} \tag{2.13}$$

The sign convention is such that the drag force acts along the negative x_w-axis in Figure 2.1 while the lift force is directed "upwards", perpendicular to the velocity vector. Using (2.2), the force components in the two frames are related by

$$\begin{aligned} D &= -X\cos\alpha\cos\beta - \bar{Y}\sin\beta - Z\sin\alpha\cos\beta \\ Y &= -X\cos\alpha\sin\beta + \bar{Y}\cos\beta - Z\sin\alpha\sin\beta \\ L &= X\sin\alpha - Z\cos\alpha \end{aligned}$$

The lift force L opposes gravity and prevents the aircraft from falling down. The lift generated is mainly decided by the angle of attack, α. Figure 2.5 shows the lift coefficient C_L as a function of α for the ADMIRE model (see Section 2.4). An increase in α leads to an increase in C_L up to $\alpha = 32^\circ$ where C_L reaches its maximum. This angle of attack is known as the stall angle. Beyond the stall angle, C_L starts to decrease. In most aircraft applications, in particular for civil airplanes, one wants to avoid stall for safety reasons (Roskam 1989). However, in military applications it has been shown that by utilizing high angles of attack, certain tactical advantages can be achieved (Herbst 1980, Well et al. 1982).

2.1.7 Gathering the Equations

Let us now state the equations that govern the motion of an aircraft by combining the rigid body dynamics from Section 2.1.5 with the forces and moments from Section 2.1.6.

Combining (2.6) with (2.7) gives the body-axes equations summarized in Table 2.1. The force equations can also be expressed in terms of V_T, α, β, and ω_w

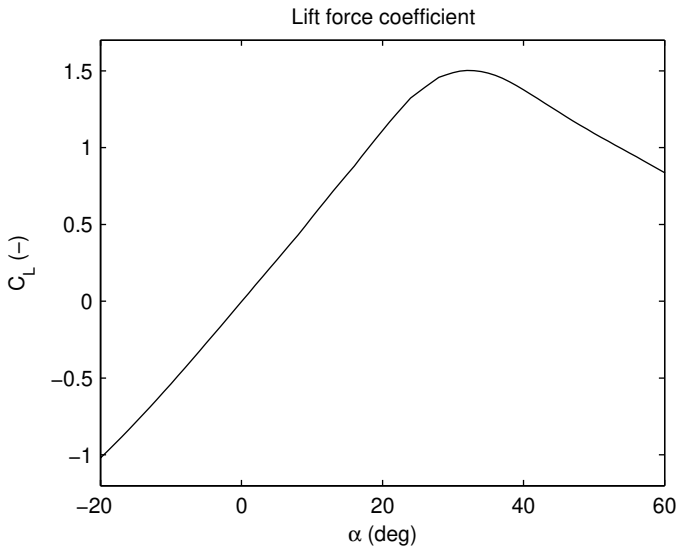


Figure 2.5: Typical lift coefficient vs angle of attack relationship. The lift force increases up to the stall angle, beyond which it starts to decrease.

which gives the wind-axes equations in Table 2.2 (Stevens and Lewis 1992, p. 85). Finally, Table 2.3 contains the longitudinal equations of motion that result in the absence of lateral motion, i.e., when $p = r = \phi = \beta = 0$ (Stevens and Lewis 1992, p. 88–89).

2.2 Control Objectives

Flight control systems can be designed for several types of control objectives. Let us first consider general maneuvering. In the longitudinal direction, the normal acceleration (Stevens and Lewis 1992, p. 263)

$$n_z = -\frac{Z}{mg}$$

or the pitch rate q make up suitable controlled variables. The normal acceleration, or load factor[2], is the normalized aerodynamic force along the negative body-fixed z-axis, expressed as a multiple of the gravitational acceleration g. The normal acceleration experienced by the pilot is given by (Stevens and Lewis 1992, p. 263)

$$n_{zp} = n_z + \frac{\dot{q}x_P}{g} \tag{2.18}$$

[2]The term "load factor" is also often used for the lift-to-weight ratio $n = \frac{L}{mg}$, see Stevens and Lewis (1992, p. 225).

Body-Axes Force Equations

$$\begin{aligned} X + F_T - mg\sin\theta &= m(\dot{u} + qw - rv) \\ \bar{Y} + mg\sin\phi\cos\theta &= m(\dot{v} + ru - pw) \\ Z + mg\cos\phi\cos\theta &= m(\dot{w} + pv - qu) \end{aligned} \tag{2.14}$$

Body-Axes Moment Equations

$$\begin{aligned} \bar{L} &= I_x\dot{p} - I_{xz}\dot{r} + (I_z - I_y)qr - I_{xz}pq \\ M + F_T z_{TP} &= I_y\dot{q} + (I_x - I_z)pr + I_{xz}(p^2 - r^2) \\ N &= I_z\dot{r} - I_{xz}\dot{p} + (I_y - I_x)pq + I_{xz}qr \end{aligned} \tag{2.15}$$

Table 2.1: Force and moment equations expressed in the body-fixed frame.

Wind-Axes Force Equations

$$\begin{aligned} \dot{V}_T &= \frac{1}{m}(-D + F_T\cos\alpha\cos\beta + mg_1) \\ \dot{\alpha} &= \frac{1}{\cos\beta}(q_w + \frac{1}{mV_T}(-L - F_T\sin\alpha + mg_3)) \\ \dot{\beta} &= -r_w + \frac{1}{mV_T}(Y - F_T\cos\alpha\sin\beta + mg_2) \end{aligned} \tag{2.16}$$

Table 2.2: Force equations expressed in the wind-axes frame.

Longitudinal Motion

$$\begin{aligned} \dot{V}_T &= \frac{1}{m}(-D + F_T\cos\alpha - mg\sin\gamma) \\ \dot{\alpha} &= q + \frac{1}{mV_T}(-L - F_T\sin\alpha + mg\cos\gamma) \\ \dot{\gamma} &= \frac{1}{mV_T}(L + F_T\sin\alpha - mg\cos\gamma) \\ \dot{\theta} &= q \\ \dot{q} &= \frac{1}{I_y}(M + F_T z_{TP}) \end{aligned} \tag{2.17}$$

Table 2.3: Longitudinal equations of motion.

where x_P is the distance between the pilot and the aircraft center of gravity measured along the body-fixed x-axis. The normal acceleration is closely coupled to the angle of attack α. Since α appears naturally in the equations of motion (2.16), angle of attack command control designs are also common, in particular for nonlinear approaches.

In the lateral directions, roll rate and sideslip command control systems are prevalent. For roll control, the body-fixed x-axis may be selected as the rotation axis, and p as the controlled variable. However, at high angles of attack, this choice has the disadvantage that angle of attack is turned into sideslip and vice versa during a roll maneuver. This may not be tolerable since the largest acceptable amount of sideslip during a roll is in the order of 3–5 degrees (Durham et al. 1994). To remove this effect, the rotation axis can instead be selected as the x-axis of the wind-axes frame, which means p_w is the controlled variable. Ideally, α and β then remain constant during a roll. The resulting maneuver is known as a velocity vector roll.

There also exist situations where other control objectives are of interest. Autopilot functions like altitude, heading, and speed hold are vital to assist the pilot during long distance flight. For firing on-board weapons, the orientation of the aircraft is crucial. To benefit from the drag reduction that can be accomplished during close formation flight, the position of the wingman relative to the leader must be controlled precisely, preferably automatically to relieve the workload of the wingman pilot (Hall and Pachter 2000). Furthermore, in order to perform an automated landing of the aircraft it may be of interest to control its descent through the flight path angle γ.

2.3 Approximations for Control

Equations (2.14)–(2.17) together with the expressions for the aerodynamic forces and moments (2.9) and the servo dynamics (2.4) constitute a detailed dynamical model of an aircraft. In fact, most control design methods require some approximations to be made before they can be applied. In this section we review the approximations needed to use linear control, feedback linearization, backstepping, and control allocation.

2.3.1 Linear Control

To use linear methods for flight control design, linear approximations of the rigid body equations (2.14)–(2.17) and the aerodynamic efforts (2.9) must be used. Linearizing the aerodynamics implies that the model is not valid at high angles of attack where, e.g., the lift force starts to decrease, see Figure 2.5. Linearizing the rigid body equations includes linearizing the cross-product $\omega \times I\omega$ in (2.6). As a consequence, $(I_x - I_z)pr + I_{xz}(p^2 - r^2)$ in (2.15) is approximated by zero and not accounted for in the control design. During a rapid roll, corresponding to a high value of p, these terms induce an undesired angular acceleration in pitch. This

effect is known as inertia coupling (Stevens and Lewis 1992, p. 270–271).

A benefit of linearizing the equations of motion is that the longitudinal and the lateral-directional equations become decoupled. This reduces the complexity of the control design which can be performed separately for longitudinal and lateral motion.

2.3.2 Backstepping and Feedback Linearization

Backstepping and feedback linearization (nonlinear dynamic inversion) both assume full-state feedback, and hence require all the variables entering the aircraft model to be measured. Since most modern aircraft carry a full sensor suite, reliable direct or observer-based measurements of all relevant variables are available (Enns et al. 1994).

These methods also require the system to be in a lower-triangular form, see Section 3.3.2. In particular, the control input must enter only at the "bottom" of the system model. If α and β are controlled variables, this implies that the influence of the control surface deflections on the aerodynamic forces must be neglected, so that the control surfaces are seen as pure moment generators affecting only $\dot{\omega}$, see, e.g., Lane and Stengel (1988), Enns et al. (1994) and Reiner et al. (1996). Fortunately, this is a good approximation for many types of aircraft, see, e.g., Etkin and Reid (1996, p. 33).

Two more approximations are frequently used in feedback linearization designs, and will also be used for backstepping control design. First, the dependence of the aerodynamic forces on the angular velocity ω is neglected. Second, the actuator dynamics are neglected, i.e., (2.4) is replaced by $\delta = u$. These assumptions are not structurally necessary but substantially simplify the design procedure.

2.3.3 Control Allocation

In flight control applications, control allocation means computing control surface deflections such that the demanded aerodynamic moments, and possible also forces, are produced. This requires a static relationship between the commanded control deflections and the resulting forces and moments, i.e., the servo dynamics (2.4) need to be neglected.

Further, for linear control allocation methods to be applicable, the aerodynamic forces and moments must be affine in the control deflections. In terms of the aerodynamic coefficients in (2.9) this means that

$$\begin{aligned} C_F(\delta, x) &= a_F(x) + b_F(x)\delta \\ C_M(\delta, x) &= a_M(x) + b_M(x)\delta \end{aligned}$$

most hold, where $x = (\alpha, \beta, p, q, r, \dots)$.

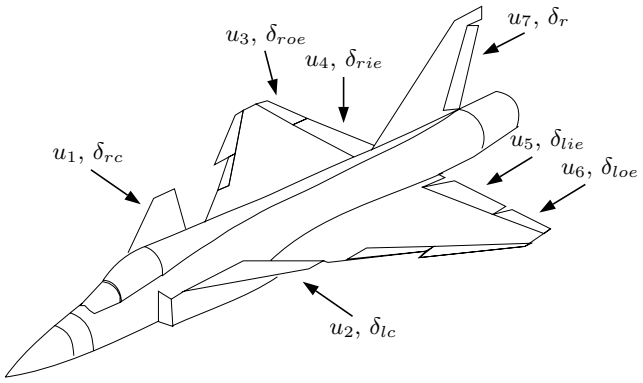


Figure 2.6: ADMIRE control surface configuration. u_i are the commanded deflections and δ_* are the actual deflections.

2.4 The ADMIRE Model

To illustrate the design tools developed in this thesis, and to evaluate their usefulness in flight control design, we will use the ADMIRE [3] aircraft model (ADMIRE ver. 3.4h 2003), implemented in MATLAB/Simulink and maintained by the Department of Autonomous Systems of the Swedish Research Agency (FOI). The model describes a small single engine fighter aircraft with a delta canard configuration and with the following characteristics:

- *Dynamics:* The dynamic model consists of the nonlinear rigid body equations (2.6) along with the corresponding equations for the position and orientation. Actuator and sensor dynamics are included, see below.

- *Aerodynamics:* The aerodata model is based on the Generic Aerodata Model (GAM) developed by Saab AB (Backström 1997), and was recently extended for high angles of attack.

- *Control surfaces:* The actuator suite consists of canards (left and right), leading-edge flaps, elevons (inboard and outboard, left and right), a rudder, and also thrust vectoring capabilities. In our control designs, the leading-edge flaps will not be used due to their low effectiveness for maneuvering. Thrust vectoring will not be used either due to lack of documentation regarding its function. The remaining seven actuators—the two canards, the four elevons, and the rudder—are depicted in Figure 2.6, where u and δ denote the commanded and the actual deflections, respectively.

- *Actuator models:* The servo dynamics of the utilized control surfaces are given by first order systems with a time constant of 0.05 s, corresponding to

[3] Aerodata Model in Research Environment

Control surface	Minimum deflection (deg)	Maximum deflection (deg)	Maximum rate (deg/s)
Canards	−55	25	50
Elevons	−30	30	150
Rudder	−30	30	100

Table 2.4: ADMIRE control surface position and rate limits below Mach 0.5.

a bandwidth of 20 rad/s. Actuator position and rate constraints are also included. The maximum allowable deflections and deflection rates, valid below Mach 0.5, are summarized in Table 2.4.

- *Flight envelope:* The flight envelope covers Mach numbers up to 1.2 and altitudes up to 6000 m. Longitudinal aerodata are available up to 90 degrees angles of attack, but lateral aerodata only exist for angles of attack up to 30 degrees.

Part I

Backstepping

Chapter 3

Introduction to Part I

Lyapunov theory has for a long time been an important tool in linear as well as in nonlinear control theory. However, its use within nonlinear control is often hampered by the difficulties to find a Lyapunov function for a given system. If one can be found, the system is known to be stable, but the task of finding such a function is often left to the imagination and experience of the designer.

Backstepping is a systematic method for nonlinear control design, which can be applied to a broad class of systems. The name "backstepping" refers to the recursive nature of the design procedure. First, only a small subsystem is considered, for which a "virtual" control law is constructed. Then, the design is extended in several steps until a control law for the full system has been constructed. Along with the control law, a Lyapunov function for the controlled system is successively constructed.

An important feature of backstepping is that nonlinearities can be dealt with in several ways. Useful nonlinearities, which act stabilizing, can be retained, and sector bounded nonlinearities may dealt with using linear control. This is in contrast to feedback linearizing control (Isidori 1995) where nonlinearities are cancelled using nonlinear feedback. Retaining nonlinearities instead of cancelling them requires less precise models and may also require less control effort. Further, the resulting control laws can sometimes be shown to be optimal with respect to a meaningful performance index, which guarantees certain robustness properties.

The origin of backstepping is not quite clear due to its simultaneous and often implicit appearance in several papers in the late 1980's. However, it is fair to say that backstepping has recieved much attention thanks to the work of Professor Petar V. Kokotović and coworkers. The 1991 Bode lecture at the IEEE Conference on Decision and Control, published in Kokotović (1992), was devoted to the evolving subject and the year after, Kanellakopoulos et al. (1992) presented a mathematical "toolkit" for designing control laws for various nonlinear systems using backstepping. During the following years, the textbooks by Krstić et al. (1995), Freeman and Kokotović (1996), and Sepulchre et al. (1997a) were published. The progress of backstepping and other nonlinear control tools during the 1990's were

surveyed by Kokotović (1999) at the 1999 IFAC World Congress in Beijing.

In this part of the thesis we will use backstepping to develop control laws for some nonlinear systems related to flight control. Previous nonlinear flight control designs are typically based on feedback linearization, or nonlinear dynamic inversion (NDI) as the method is called in the flight control community, see, e.g., Meyer et al. (1984), Lane and Stengel (1988), and Enns et al. (1994). In comparison, the backstepping control laws proposed in Chapter 4 rely on less precise aerodynamic model information, and possess certain robustness properties. The rigid body control design in Chapter 5 uses a vector form description of the dynamics, rather than the typically used component form description, and offers a compact way to design flight control laws.

This chapter introduces the backstepping technique and contains no new material. Section 3.1 reviews some concepts and results from Lyapunov theory, and Section 3.2 introduces the basic ideas in Lyapunov based control design. The backstepping method is presented Section 3.3, along with a discussion regarding which systems it can be applied to and which design choices there are. Some related designs methods are reviewed in Section 3.4, and previously reported applications of backstepping are listed in Section 3.5. Section 3.6 deals with inverse optimal control, i.e., how to find the performance index minimized by a certain control law. Finally, Section 3.7 outlines the remaining chapters of this part of the thesis.

3.1 Lyapunov Theory

Backstepping control design is based on Lyapunov theory. The aim is to construct a control law that brings the system to, or at least near, some desired state. That is to say, we wish to make this state a stable equilibrium of the closed loop system. In this section we define the notion of stability in the Lyapunov sense, and review the main tools for proving stability of an equilibrium. This section is based on Slotine and Li (1991, chap. 3) and Khalil (2002, chap. 4) to which we refer for proofs of the stability theorems.

Consider the autonomous system

$$\dot{x} = f(x) \tag{3.1}$$

where x is the system state vector. This can be thought of as the closed loop dynamics of a controlled system. Let $x = x_e$ be an equilibrium of the system, that is, let $f(x_e) = 0$. The stability properties if this equilibrium are characterized by the following definition.

Definition 3.1 (Lyapunov stability) *The equilibrium point $x = x_e$ of (3.1) is*

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

$$\big\|x(0) - x_e\big\| < \delta \Rightarrow \big\|x(t) - x_e\big\| < \epsilon, \quad \forall t \geq 0$$

- *unstable if it is not stable*

- *asymptotically stable if it is stable and in addition there exists* $r > 0$ *such that*

$$\|x(0) - x_e\| < r \Rightarrow \lim_{t\to\infty} x(t) = x_e$$

- *globally asymptotically stable (GAS) if it is asymptotically stable for all initial states, that is, if*

$$\lim_{t\to\infty} x(t) = x_e, \quad \forall x(0)$$

These definitions involve the trajectory $x(t)$, the solution to (3.1). In general, $x(t)$ cannot be found analytically. Fortunately there are other ways of proving stability.

The Russian mathematician and engineer A. M. Lyapunov introduced the idea of condensing the state vector $x(t)$ into a scalar function $V(x)$, measuring how far from the equilibrium the system is. If $V(x)$ decreases over time, then the system must be moving towards the equilibrium. This approach to showing stability is called Lyapunov's direct method (or second method). Lyapunov's original work can be found in Lyapunov (1992).

Let us first introduce some useful concepts.

Definition 3.2 *A scalar function* $V(x)$ *is*

- *positive definite if* $V(0) = 0$ *and* $V(x) > 0, \quad x \neq 0$
- *positive semidefinite if* $V(0) = 0$ *and* $V(x) \geq 0, \quad x \neq 0$
- *negative (semi-)definite if* $-V(x)$ *is positive (semi-)definite*
- *radially unbounded if* $V(x) \to \infty$ *as* $\|x\| \to \infty$

We now state the main theorem to be used for proving global asymptotic stability (Khalil 2002, thm. 4.2).

Theorem 3.1 *Consider the system (3.1) and let* $f(0) = 0$*. Let* $V(x)$ *be a positive definite, radially unbounded, continuously differentiable scalar function. If*

$$\dot{V}(x) = V_x(x)f(x) < 0, \quad x \neq 0$$

then $x = 0$ *is a globally asymptotically stable (GAS) equilibrium.*

A positive definite function $V(x)$ that satisfies $\dot{V}(x) \leq 0$ is called a Lyapunov function of the system. If such a function can be found, $x = 0$ is a stable equilibrium. In the theorem, the stronger condition $\dot{V}(x) < 0$ gives asymptotic stability. The radial unboundedness of $V(x)$ means that all level curves of $V(x)$ are closed. This is necessary to guarantee that asymptotic stability holds globally.

In some cases, global asymptotic stability can be shown when $\dot{V}(x)$ is only negative semidefinite (Khalil 2002, cor. 4.2).

Theorem 3.2 *Consider the system (3.1) and let $f(0) = 0$. Let $V(x)$ be a positive definite, radially unbounded, continuously differentiable scalar function such that*

$$\dot{V}(x) = V_x(x)f(x) \leq 0, \quad \forall x$$

Let $S = \{x : \dot{V}(x) = 0\}$ and suppose that no other solution than $x(t) = 0$ can stay forever in S. Then, $x = 0$ is a globally asymptotically stable (GAS) equilibrium.

Note that both these theorems are non-constructive, in the sense that they give no clue about how to find the function V satisfying the conditions necessary to show GAS.

3.2 Lyapunov Based Control Design

Let us now turn to control design using Lyapunov theory. Consider the system

$$\dot{x} = f(x, u) \tag{3.2}$$

where x is the system state and u is the control input, and let $x = 0$ be the control objective. Stated differently, we want to design a control law

$$u = k(x)$$

such that $x = 0$ is a GAS equilibrium of the closed loop system

$$\dot{x} = f(x, k(x))$$

To show GAS we need to construct a Lyapunov function $V(x)$ satisfying the conditions in Theorem 3.1 or Theorem 3.2. Constructing a control law $k(x)$ and a Lyapunov function $V(x)$ to go with it is what Lyapunov based control design is about.

A straightforward approach to find $k(x)$ is to pick a positive definite, radially unbounded function $V(x)$ and then choose $k(x)$ such that

$$\dot{V} = V_x(x)f(x, k(x)) < 0, \quad x \neq 0 \tag{3.3}$$

For this approach to succeed, V must be carefully selected, or (3.3) will not be solvable. This motivates the following definition (Krstić et al. 1995, def. 2.4):

Definition 3.3 (Control Lyapunov function) *A positive definite, radially unbounded, smooth scalar function $V(x)$ is called a control Lyapunov function (clf) for (3.2) if*

$$\inf_u V_x f(x, u) < 0, \quad x \neq 0$$

Given a clf for the system, we can thus find a globally stabilizing control law. In fact, the existence of a globally stabilizing control law is equivalent to the existence

of a clf. This means that for each globally stabilizing control law, a corresponding clf can be found, and vice versa. This is known as Artstein's theorem (Artstein 1983).

To illustrate the approach, consider the system

$$\dot{x} = f(x) + g(x)u \tag{3.4}$$

which is affine in the control input, and assume that a clf for the system is known. For this case Sontag (1989) proposes a particular choice of control law given by

$$u = k(x) = -\frac{a + \sqrt{a^2 + b^2}}{b} \tag{3.5}$$

where

$$a = V_x(x)f(x)$$
$$b = V_x(x)g(x)$$

This control law gives

$$\dot{V} = V_x(x)(f(x) + g(x)u) = a + b\left(-\frac{a + \sqrt{a^2 + b^2}}{b}\right) = -\sqrt{a^2 + b^2} \tag{3.6}$$

and thus renders the origin GAS. Equation (3.5) is known as Sontag's formula.

Freeman and Primbs (1996) propose a related approach where u is chosen to minimize the control effort necessary to satisfy

$$\dot{V} \leq -W(x)$$

for some W. The minimization is carried out pointwise in time (and not over some horizon). Using an inequality constraint rather than asking for equality (as in (3.6)) makes it possible to benefit from the system's inherent stability properties. If $f(x)$ alone drives the system (3.4) towards the equilibrium such that

$$\dot{V}|_{u=0} = V_x(x)f(x) < -W(x)$$

it would be a waste of control effort to achieve $\dot{V} = -W(x)$.

3.3 Backstepping

The methods above assume that a clf is known for the system to be controlled. What if this is not the case? How can a control law be constructed along with a Lyapunov function to show closed loop stability? Backstepping solves this problem through a recursive design for a class of nonlinear systems. In this section, based on Krstić et al. (1995), we review the backstepping method, to which systems it can be applied, and what the design choices are.

3.3.1 Main Result

The main idea in backstepping is to let certain states act as "virtual controls" of others. The same idea can be found in cascaded control design and also in singular perturbation theory (Kokotović et al. 1986).

Consider the system

$$\dot{x} = f(x, \xi) \tag{3.7a}$$

$$\dot{\xi} = u \tag{3.7b}$$

where $x \in \mathbb{R}^n$ and $\xi \in \mathbb{R}$ are state variables and $u \in \mathbb{R}$ is the control input. Assume that if ξ were the control input, the control law

$$\xi = \xi^{\text{des}}(x) \tag{3.8}$$

would make the origin $x = 0$ a GAS equilibrium, shown by the Lyapunov function $W(x)$. Since ξ is not the true control input, (3.8) is called a *virtual control law*. The following theorem, based on Sepulchre et al. (1997c), shows how to "step back" through the model, and construct a true control law in terms of u given ξ^{des} and $W(x)$.

Theorem 3.3 (Backstepping) *Consider the system (3.7). Assume that a clf $W(x)$ and a virtual control law $\xi = \xi^{\text{des}}(x)$ are known for the subsystem (3.7a) such that*

$$\dot{W}|_{\xi=\xi^{\text{des}}} = W_x(x) f(x, \xi^{\text{des}}(x)) < 0, \quad x \neq 0$$

Then, a clf for the augmented system (3.7) is given by

$$V(x, \xi) = W(x) + \frac{1}{2}(\xi - \xi^{\text{des}}(x))^2 \tag{3.9}$$

Moreover, the control law

$$u = \frac{\partial \xi^{\text{des}}}{\partial x}(x) f(x, \xi) - W_x(x) \frac{f(x, \xi) - f(x, \xi^{\text{des}}(x))}{\xi - \xi^{\text{des}}(x)} + \xi^{\text{des}}(x) - \xi \tag{3.10}$$

achieves

$$\dot{V} = W_x(x) f(x, \xi^{\text{des}}(x)) - (\xi - \xi^{\text{des}}(x))^2 < 0, \quad x \neq 0,\ \xi \neq \xi^{\text{des}}(0)$$

and makes $x = 0$, $\xi = \xi^{\text{des}}(0)$ a GAS equilibrium.

The control law (3.10) is neither the only, nor necessarily the best globally stabilizing control law for (3.7). The value of the theorem is that it shows the existence of at least one globally stabilizing control law for this type of augmented systems.

Proof: We will conduct the proof in a constructive manner to show which design choices that can be made during the control law construction.

The key idea is to utilize that the virtual control law (3.8) would stabilize the subsystem (3.7a) if ξ were a control variable. Since we are not in direct control of ξ we introduce the residual

$$\tilde{\xi} = \xi - \xi^{\text{des}}(x)$$

and use the true control input u to steer $\tilde{\xi}$ to zero. If $\tilde{\xi}$ goes to zero, ξ will go to the desired value ξ^{des} and the entire system will be stabilized. In terms of $\tilde{\xi}$, the system dynamics (3.7) become

$$\dot{x} = f(x, \tilde{\xi} + \xi^{\text{des}}(x)) \triangleq \underbrace{f(x, \xi^{\text{des}}(x))}_{\text{desired dynamics}} + \psi(x, \tilde{\xi})\tilde{\xi} \tag{3.11a}$$

$$\dot{\tilde{\xi}} = u - \frac{\partial \xi^{\text{des}}}{\partial x}(x) f(x, \tilde{\xi} + \xi^{\text{des}}(x)) \tag{3.11b}$$

where

$$\psi(x, \tilde{\xi}) = \frac{f(x, \tilde{\xi} + \xi^{\text{des}}(x)) - f(x, \xi^{\text{des}}(x))}{\tilde{\xi}}$$

In (3.11a) we have separated the desired dynamics from the dynamics due to $\tilde{\xi} \neq 0$.

To find a clf for the augmented system it is natural to take the clf for the subsystem, $W(x)$, and add a term penalizing the residual $\tilde{\xi}$. We therefore select

$$V(x, \tilde{\xi}) = W(x) + \frac{1}{2}\tilde{\xi}^2$$

and find a globally stabilizing control law by making $\dot{V}$ negative definite.

$$\begin{aligned} \dot{V} &= W_x(x)\Big[f(x, \xi^{\text{des}}(x)) + \psi(x, \tilde{\xi})\tilde{\xi}\Big] + \tilde{\xi}\Big[u - \frac{\partial \xi^{\text{des}}}{\partial x}(x) f(x, \tilde{\xi} + \xi^{\text{des}}(x))\Big] \\ &= \underbrace{W_x(x) f(x, \xi^{\text{des}}(x))}_{<0,\ x \neq 0} + \tilde{\xi}\Big[W_x(x)\psi(x, \tilde{\xi}) + u - \frac{\partial \xi^{\text{des}}}{\partial x}(x) f(x, \tilde{\xi} + \xi^{\text{des}}(x))\Big] \end{aligned} \tag{3.12}$$

The first term is negative definite according to the assumptions. The second term, and thus $\dot{V}$, can be made negative definite by selecting

$$\begin{aligned} u &= -W_x(x)\psi(x, \tilde{\xi}) + \frac{\partial \xi^{\text{des}}}{\partial x}(x) f(x, \tilde{\xi} + \xi^{\text{des}}(x)) - \tilde{\xi} \\ &= \frac{\partial \xi^{\text{des}}}{\partial x}(x) f(x, \xi) - W_x(x)\frac{f(x, \xi) - f(x, \xi^{\text{des}}(x))}{\xi - \xi^{\text{des}}(x)} + \xi^{\text{des}}(x) - \xi \end{aligned}$$

This yields

$$\dot{V} = W_x(x)(f(x, \xi^{\text{des}}(x)) - \tilde{\xi}^2 < 0, \quad x \neq 0,\ \tilde{\xi} \neq 0$$

Hence this control law makes $x = 0$, $\tilde{\xi} = 0$ a GAS equilibrium. In the original variables, this implies $\xi = \xi^{\text{des}}(0)$. □

Let us now deal with some issues related to practical control design using backstepping.

3.3.2 Which Systems Can Be Handled?

The backstepping technique can be extended to other nonlinear systems than (3.7).

Input Nonlinearities

An immediate extension of Theorem 3.3 is to handle systems with input nonlinearities (Krstić et al. 1995, p. 61):

$$\begin{aligned}\dot{x} &= f(x,\xi)\\ \dot{\xi} &= g(x,\xi,u)\end{aligned}$$

Introducing $\dot{\xi} = \tilde{u}$, Theorem 3.3 can be used to find a control law in terms of $\tilde{u}$. Then u can be determined given that

$$g(x,\xi,u) = \tilde{u}$$

can be solved for u. If this is possible, we say that g is invertible w.r.t. u.

Feedback Form Systems

If the system (3.7) is augmented with additional integrators at the input, Theorem 3.3 can be applied recursively. Assume that u is not the actual control input, but a state variable with the dynamics

$$\dot{u} = v \tag{3.13}$$

Then (3.10) becomes a virtual control law, which along with the clf (3.9) can be used to find a globally stabilizing control law in terms of v for the system (3.7) augmented by (3.13).

Now, either v is yet another state variable, in which case the backstepping procedure is repeated once again, or v is indeed the control input, in which case we have arrived at a globally stabilizing control law.

Thus, by recursively applying backstepping, globally stabilizing control laws can be constructed for systems of the following lower triangular form, known as *pure-feedback systems* (Krstić et al. 1995, p. 61):

$$\begin{aligned}\dot{x} &= f(x,\xi_1)\\ \dot{\xi}_1 &= g_1(x,\xi_1,\xi_2)\\ &\vdots\\ \dot{\xi}_i &= g_i(x,\xi_1,\ldots,\xi_i,\xi_{i+1})\\ &\vdots\\ \dot{\xi}_m &= g_m(x,\xi_1,\ldots,\xi_m,u)\end{aligned} \tag{3.14}$$

For the design to succeed, a globally stabilizing virtual control law $\xi_1 = \xi_1^{\text{des}}(x)$, along with a clf, must be known for the x-subsystem. Also, $g_i,\ i = 1, \ldots, m-1$ must be invertible w.r.t. ξ_{i+1} and g_m must be invertible w.r.t. u.

Systems for which the "new" variables enter in an affine way, are known as *strict-feedback systems* (Krstić et al. 1995, p. 58):

$$\begin{aligned}
\dot{x} &= a(x) + b(x)\xi_1 \\
\dot{\xi}_1 &= a_1(x, \xi_1) + b_1(x, \xi_1)\xi_2 \\
&\vdots \\
\dot{\xi}_i &= a_i(x, \xi_1, \ldots, \xi_i) + b_i(x, \xi_1, \ldots, \xi_i)\xi_{i+1} \\
&\vdots \\
\dot{\xi}_m &= a_m(x, \xi_1, \ldots, \xi_m) + b_m(x, \xi_1, \ldots, \xi_m)u
\end{aligned} \tag{3.15}$$

Strict-feedback systems are nice to deal with and often used for deriving results related to backstepping. First, the invertability condition imposed above is satisfied given that $b_i \neq 0$, although this is not a necessary condition, see Krstić et al. (1995, ex. 2.9). Second, if (3.7a) is affine in ξ and the dynamics are given by

$$\begin{aligned}
\dot{x} &= a(x) + b(x)\xi \\
\dot{\xi} &= u
\end{aligned}$$

then the control law (3.10) reduces to

$$u = \frac{\partial \xi^{\text{des}}}{\partial x}(x)(a(x) + b(x)\xi) - W_x(x)b(x) + \xi^{\text{des}}(x) - \xi \tag{3.16}$$

Dynamic Backstepping

Even for certain systems which do not fit into a lower triangular feedback form, there exist backstepping designs. Fontaine and Kokotović (1998) consider a two dimensional system where both states are affected by the control input:

$$\begin{aligned}
\dot{x}_1 &= \psi(x_1) + x_2 + \phi(u) \\
\dot{x}_2 &= u
\end{aligned}$$

Their approach is to first design a globally stabilizing virtual control law for the x_1-subsystem, considering $\eta = x_2 + \phi(u)$ as the input. Then backstepping is used to convert this virtual control law into a realizable one in terms of u. Their design results in a dynamic control law, and hence the term dynamic backstepping is used.

3.3.3 Which Design Choices Are There?

The proof of Theorem 3.3 leaves a lot of room for variations. Let us now exploit some of these to illustrate the different kinds of design freedom in backstepping.

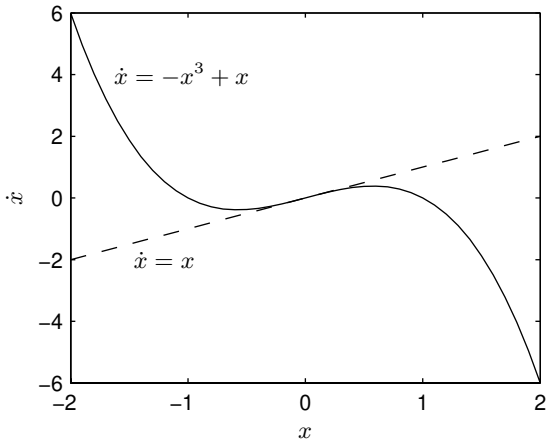


Figure 3.1: The dynamics of the uncontrolled system $\dot{x} = -x^3 + x$. The linear term acts destabilizing around the origin.

Dealing with Nonlinearities

A trademark of backstepping is that it allows the designer benefit from "useful" nonlinearities, naturally stabilizing the system. This can be done by choosing the virtual control laws properly. The following example demonstrates this fundamental difference to feedback linearization.

Example 3.1 (A useful nonlinearity) Consider the system

$$\dot{x} = -x^3 + x + u$$

and let $x = 0$ be the desired equilibrium. The uncontrolled dynamics, $\dot{x} = -x^3 + x$, are plotted in Figure 3.1. For the origin to be asymptotically stable, the sign of $\dot{x}$ should be opposite that of x for all x. For the uncontrolled system, this holds for large values of $|x|$ where the cubic term $-x^3$ dominates the dynamics, but near the origin, the linear term x dominates and destabilizes the origin.

Thus, to make the origin GAS, only the linear dynamics need to be counteracted by the control input. This can be achieved by selecting

$$u = -x \tag{3.17}$$

A clf is given by

$$W = \frac{1}{2}x^2 \tag{3.18}$$

which yields

$$\dot{W} = -x(-x^3 + x + u) = -x^4$$

proving the origin to be GAS according to Theorem 3.1.

Applying feedback linearization instead renders the control law

$$u = x^3 - kx, \quad k > 1$$

Obviously, this control law does not recognize the beneficial cubic nonlinearity but counteracts it, thus wasting control effort.

Also, the feedback linearizing design is dangerous from a robustness perspective. If the true system dynamics are $\dot{x} = -0.9x^3 + x + u$, the feedback linearization control law yields the closed loop dynamics

$$\dot{x} = 0.1x^3 - (k-1)x$$

for which the origin is only locally asymptotically stable since regardless of k, the cubic, destabilizing term will dominate for large values of $|x|$. The backstepping control law yields

$$\dot{x} = -0.9x^3$$

and the origin remains a GAS equilibrium.

Weighting the CLF

When constructing the combined clf (3.9), we can choose any weighted sum of the two terms,

$$V = cW + \frac{1}{2}(\xi - \xi^{\text{des}})^2, \; c > 0$$

In the designs in Chapter 4, c will be used to cancel certain terms in Equation (3.12). A technical hint is to put the weight on W since it yields nicer expressions.

Non-quadratic CLF

Although quadratic clf:s are frequently used in backstepping, they do not always constitute the best choice as the following example demonstrates.

Example 3.2 (A useful nonlinearity, cont.) Consider the system in Example 3.1 augmented by an integrator:

$$\begin{aligned} \dot{x}_1 &= -x_1^3 + x_1 + x_2 \\ \dot{x}_2 &= u \end{aligned}$$

To benefit from the useful nonlinearity $-x_1^3$, let us reuse (3.17) as our virtual control law, i.e.,

$$x_2^{\text{des}}(x_1) = -x_1$$

The choice of clf for the x_1-subsystem will strongly affect the resulting control law. To see this we first reuse (3.18) and pick

$$W = \frac{1}{2}x_1^2$$

Introducing

$$\tilde{x}_2 = x_2 - x_2^{\text{des}}(x_1) = x_2 + x_1$$

we can rewrite the system as

$$\begin{aligned}\dot{x}_1 &= -x_1^3 + \tilde{x}_2\\ \dot{\tilde{x}}_2 &= u - x_1^3 + \tilde{x}_2\end{aligned}$$

Following the proof of Theorem 3.3, we add a quadratic term to W, to penalize the deviation from the suggested virtual control law:

$$V(x_1, x_2) = W(x_1) + \frac{1}{2}\big(x_2 - x_2^{\text{des}}(x_1)\big)^2 = \frac{1}{2}x_1^2 + \frac{1}{2}\tilde{x}_2^2$$

Differentiating w.r.t. time yields

$$\dot{V} = x_1(-x_1^3 + \tilde{x}_2) + \tilde{x}_2(u - x_1^3 + \tilde{x}_2) = -x_1^4 + \tilde{x}_2(x_1 + u - x_1^3 + \tilde{x}_2)$$

To render $\dot{V}$ negative definite, u must clearly dominate the $\tilde{x}_2$ term using a control input of, e.g., $-3\tilde{x}_2$. In addition, since the mixed terms between x_1 and $\tilde{x}_2$ are indefinite, there seems to be no other choice than to cancel them using the control law

$$u = x_1^3 - x_1 - 3\tilde{x}_2 = x_1^3 - 4x_1 - 3\tilde{x}_2$$

We note that this control law does not recognize the fact that x_1-subsystem is naturally stabilized for high values of x_1 but instead counteracts this effect, thereby wasting control effort.

So how should we pick $W(x_1)$ to avoid this cancellation? One idea is not to specify the clf beforehand, but instead let the choice of W be decided by the backstepping design. Thus, we let W be any function satisfying Definition 3.3. As before, use

$$V(x_1, x_2) = W(x_1) + \frac{1}{2}\tilde{x}_2^2$$

and compute its time derivative.

$$\begin{aligned}\dot{V} &= W'(x_1)(-x_1^3 + \tilde{x}_2) + \tilde{x}_2(u - x_1^3 + \tilde{x}_2)\\ &= -W'(x_1)x_1^3 + \tilde{x}_2(W'(x_1) + u - x_1^3 + \tilde{x}_2)\end{aligned}$$

We now use our extended design freedom and select a W so that the indefinite mixed terms cancel each other. This is satisfied by

$$W'(x_1) = x_1^3, \quad W(x_1) = \frac{1}{4}x_1^4$$

which indeed is a valid choice. We now have

$$\dot{V} = -x_1^6 + \tilde{x}_2(u + \tilde{x}_2)$$

Clearly, the control u no longer has to cancel the cubic x_1 term but can be chosen linear in x_1 and x_2. The control law

$$u = -3\tilde{x}_2 = -3x_1 - 3x_2$$

renders $\dot{V} = -x_1^6 - 2\tilde{x}_2^2$ negative definite and thus makes the origin GAS.

This refinement of backstepping is due to Krstić et al. (1998). The technique of designing a non-quadratic clf will be used for flight control design in Chapter 4, where some of the aerodynamic forces also have the property of being nonlinear, but still stabilizing.

Choice of Virtual Control Input

For the system (3.7) there are cases where it is more convenient to use some function of ξ as the virtual control input rather than ξ itself.

Example 3.3 Consider the system

$$\begin{aligned}\dot{x}_1 &= x_1^3 + x_2^5 + x_2\\ \dot{x}_2 &= u\end{aligned}$$

For this system it is easier to find a virtual control law in terms of $\eta = x_2^5 + x_2$ than in terms of x_2 itself. With this change of variables the dynamics become

$$\begin{aligned}\dot{x}_1 &= x_1^3 + \eta\\ \dot{\eta} &= (5x_2^4 + 1)u \triangleq \tilde{u}\end{aligned}$$

A clf for the x_1-subsystem is given by $W(x_1) = \frac{1}{2}x_1^2$ which gives

$$\dot{W} = x_1(x_1^3 + x_2^5 + x_2) = x_1(x_1^3 + \eta)$$

We can now pick $\eta^{\text{des}} = -2x_1^3$ which yields $\dot{W} = -x_1^4$ negative definite if $\eta = \eta^{\text{des}}$. Using (3.16) gives the control law

$$u = \frac{1}{5x_2^4 + 1}\Big[-6x_1^2(x_1^3 + x_2^5 + x_2) - x_1 - 2x_1^3 - x_2^5 - x_2\Big]$$

Alrifai et al. (1998) use this technique for speed control of a switched reluctance motor where it is convenient to formulate the virtual control law in terms of the square current i^2.

3.4 Related Lyapunov Designs

Besides state feedback backstepping, several other constructive nonlinear control designs exist. We will now outline some of these.

3.4.1 Forwarding

The backstepping philosophy applies to systems of the form (3.7). Another class of nonlinear systems for which one can also construct globally stabilizing control laws are those that can be written

$$\dot{x} = f(x, u) \tag{3.19a}$$

$$\dot{\xi} = g(x, u) \tag{3.19b}$$

A clf and a globally stabilizing control law for the dynamics of x in (3.19a) are assumed to be known. The question is how to augment this control law to also stabilize the integrator state ξ in (3.19b). This problem, which can be seen as a dual to the one in backstepping, can be solved using so called forwarding (Sepulchre et al. 1997b).

By combining feedback (3.7) and feedforward (3.19) systems, interlaced systems can be constructed. Using backstepping in combination with forwarding, such systems can also be systematically stabilized (Sepulchre et al. 1997c).

3.4.2 Adaptive, Robust, and Observer Backstepping

So far we have only considered the case where all the state variables are available for feedback and where the model is completely known. For the non-ideal cases where this is not true, there are other flavors of backstepping to resort to.

For systems with parametric uncertainties, there exists adaptive backstepping (Krstić et al. 1995). Here, a parameter estimate update law is designed such that closed loop stability is guaranteed when the parameter estimate is used by the controller. In Chapter 6 we will see how this technique can be used to estimate and cancel unknown additive disturbances on the control input.

Robust backstepping (Freeman and Kokotović 1996) designs exist for systems with imperfect model information. Here, the idea is to select a control law such that a Lyapunov function decreases for all systems comprised by the given model uncertainty.

In cases where not all the state variables can be measured, the need for observers arises. The separation principle valid for linear systems does not hold for nonlinear systems in general. Therefore, care must be taken when designing the feedback law based on the state estimates. This is the topic of observer backstepping (Kanellakopoulos et al. 1992, Krstić et al. 1995).

3.5 Applications of Backstepping

Although backstepping theory has a rather short history, numerous practical applications can be found in the literature. This fact indicates that the need for a nonlinear design methodology handling a number of practical problems, as discussed in the previous section, has existed for a long time. We now survey some publications regarding applied backstepping. This survey is by no means complete, but is intended to show the broad spectrum of engineering disciplines in which backstepping has been used.

Backstepping designs can be found for a wide variety of electrical motors, see, e.g., Carroll et al. (1993, 1995), Hu et al. (1995, 1996), Alrifai et al. (1998). Turbocharged diesel engines are considered in Fredriksson (2002), Jankovic et al. (2000) while jet engines are the subject of Krstić et al. (1998). In Grøvlen and Fossen (1996), Strand et al. (1998), backstepping is used for automatic ship positioning. In Strand et al. (1998), the controller is made locally $\mathcal{H}_\infty$-optimal based on results in Ezal et al. (2000), see Section 3.6. Robotics is another field where backstepping designs can be found. Tracking control is considered in Jiang and Nijmeijer (1997) and Bridges et al. (1995) where the latter is a survey of approaches valid for various assumptions regarding the knowledge of the model.

There also exist some publications combining flight control and backstepping. Singh et al. (2000) treat formation flight control of unmanned aerial vehicles. Singh and Steinberg (1996) and Steinberg and Page (1999) use backstepping to design flight control laws which are adaptive to changes in the aerodynamic forces and moments due to, e.g., actuator failures. Here, the Lyapunov functions contain a term penalizing the integral of the tracking error, enhancing the robustness. Sharma and Ward (2002) consider adaptive flight path angle control using neural nets. Wänström (2001) and Dahlgren (2002) use backstepping for robust missile flight control design.

3.6 Inverse Optimal Control

In linear control, one often seeks control laws that are optimal in some sense, due to their ability to suppress external disturbances and to function despite model errors, as in the case of $\mathcal{H}_\infty$ and linear quadratic control (Zhou et al. 1996). It is therefore natural that efforts have been made to extend these designs to nonlinear control. The difficulty lies in the Hamilton-Jacobi-Bellman equation that needs to be solved in order to find the control law. However, the reverse task – to find the performance index minimized by a certain control law – is easy to perform in some cases. This route of first deriving a control law and then determining which cost it minimizes, and thus in which sense it is optimal, is known as *inverse optimal control*.

Ezal et al. (2000) apply this technique to strict-feedback systems to construct backstepping controllers which are locally $\mathcal{H}_\infty$-optimal, and globally optimal according to some performance index that the designer cannot control precisely.

Löfberg (2000) extends these results and considers a linear quadratic performance index around the equilibrium.

In this section, we develop tools for inverse optimal control. These will be used in Chapter 4 to show that some of the proposed backstepping controllers minimize meaningful performance indices. The material in this section will be presented in a rather intuitive manner. A mathematically strict treatment of the subject can be found in Sepulchre et al. (1997a).

In Section 3.6.1 the general infinite horizon optimal control problem is introduced. In Section 3.6.2, systems which are affine in the control input are considered, and some standard inverse results are derived for cost functionals which are quadratic in the input. Finally, the well known gain margin result of optimal control is shown in Section 3.6.3.

3.6.1 Optimal Control

Let us first consider infinite horizon optimal control for nonlinear systems. Given a dynamic system

$$\dot{x} = f(x, u)$$

where $x \in \mathbb{R}^n$ is the state vector and $u \in \mathbb{R}^m$ is the control input, we seek the control law $u(x)$ that minimizes the cost functional

$$J = \int_0^\infty L(x, u)dt$$

By choosing L properly, the system is guaranteed to reach steady state as $t \to \infty$. The cost to get there, J, the *cost-to-go*, will depend on the initial state of the system, $x(0)$. We therefore write this cost $J(x)$.

The optimal control law is denoted $u^*(x)$. When this optimal control law is applied, $J(x)$ will decrease along the trajectory, since the cost-to-go must continuously decrease by the principle of optimality (Bertsekas 1995). This means that $V(x) = J(x)$ is a Lyapunov function for the controlled system. At steady state it must hold that $V = 0$. Hence, the following holds:

$$V(x(0)) = \int_0^\infty L(x, u^*)dt = -\big[\underbrace{V(x(\infty))}_{=0} - V(x(0))\big] = -\int_0^\infty \dot{V}(x)dt$$

Clearly, when the optimal control law is used, L and $-\dot{V}$ coincide. This motivates the Hamilton-Jacobi-Bellman (HJB) equation

$$0 = \min_u \big[L(x, u) + V_x(x)f(x, u)\big] \tag{3.20}$$

for finding the optimal control law u^* along with a Lyapunov function $V(x)$ for the controlled system.

3.6.2 Inverse Optimal Control

It is well known that in general, it is not feasible to solve the HJB equation (3.20). We therefore restrict our discussion to dynamic systems which are affine in the control input:

$$\dot{x} = f(x) + g(x)u \tag{3.21}$$

For these systems, the HJB equation is greatly simplified if L is chosen quadratic in u according to

$$L(x,u) = q(x) + u^T R(x) u$$

where $q(x)$ is positive definite and $R(x)$ is a symmetric matrix, positive definite for all x. Inserting this into (3.20) yields

$$0 = \min_u \big[q(x) + u^T R(x) u + V_x(x)(f(x) + g(x)u) \big] \tag{3.22}$$

The equation is solved in two steps. First we find the minimizing u, and then we solve for equality to zero. The minimization can be done by completion of squares:

$$\begin{aligned} &q + u^T R u + V_x f + V_x g u = \\ &\qquad q + V_x f + \big[u + \frac{1}{2} R^{-1} (V_x g)^T\big]^T R \big[u + \frac{1}{2} R^{-1} (V_x g)^T\big] - \frac{1}{4} V_x g R^{-1} (V_x g)^T \end{aligned}$$

The control input u only appears in the "square", positive definite term. The minimum therefore occurs when this term is set to zero, which is achieved by

$$u^* = k(x) = -\frac{1}{2} R^{-1}(x)(V_x(x) g(x))^T \tag{3.23}$$

What remains is to insert this control law into (3.22). This gives us

$$0 = q(x) + V_x(x) f(x) - \frac{1}{4} V_x(x) g(x) R^{-1}(x) (V_x(x) g(x))^T \tag{3.24}$$

Equations (3.23) and (3.24) provide the connection between the cost functional, given by $q(x)$ and $R(x)$, and the optimal control strategy, in terms of $k(x)$ and $V(x)$. As for the general problem in the previous section, it is in general not feasible to solve for $k(x)$ and $V(x)$ given $q(x)$ and $R(x)$ of the designer's choice. However, we see that the reverse task is simpler. Given a control law $k(x)$ and a clf $V(x)$ (corresponding to a Lyapunov function for the controlled system), $q(x)$ and $R(x)$, determining the cost functional that is minimized, can be found by solving

$$k(x) = -\frac{1}{2} R^{-1}(x)(V_x(x) g(x))^T \tag{3.25}$$

$$q(x) = -V_x(x) f(x) - \frac{1}{2} V_x(x) g(x) k(x) \tag{3.26}$$

For a single input system we can explicitly solve for $R(x)$:

$$R(x) = -\frac{V_x(x) g(x)}{2 k(x)} \tag{3.27}$$

To illustrate this technique we revisit Example 3.2 and show that the derived backstepping control law is optimal w.r.t. a meaningful cost functional, in the sense that $q(x)$ and $R(x)$ both become positive definite.

Example 3.4 (Exploiting useful nonlinearities is optimal) Consider the system in Example 3.2. The original dynamics

$$\begin{aligned}\dot{x}_1 &= -x_1^3 + x_1 + x_2\\ \dot{x}_2 &= u\end{aligned}$$

can be written in the form (3.21) with

$$f(x) = \begin{pmatrix} -x_1^3 + x_1 + x_2 \\ 0 \end{pmatrix}, \quad g(x) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

In Example 3.2,

$$u = k(x) = -3(x_1 + x_2)$$

was shown to make the origin GAS using the Lyapunov function

$$V(x) = \frac{1}{4}x_1^4 + \frac{1}{2}(x_1 + x_2)^2$$

which satisfies

$$V_x(x) = \begin{pmatrix} x_1^3 + x_1 + x_2 & x_1 + x_2 \end{pmatrix}$$

Inserting this into (3.26) and (3.27) yields

$$\begin{aligned}R(x) &= \frac{x_1 + x_2}{2 \cdot 3(x_1 + x_2)} = \frac{1}{6}\\ q(x) &= -(x_1^3 + x_1 + x_2)(-x_1^3 + x_1 + x_2) + \frac{1}{2}(x_1 + x_2) \cdot 3(x_1 + x_2)\\ &= x_1^6 + \frac{1}{2}(x_1 + x_2)^2\end{aligned}$$

Thus, the suggested control law minimizes the cost functional

$$J = \int_0^\infty (x_1^6 + \frac{1}{2}(x_1 + x_2)^2 + \frac{1}{6}u^2)dt$$

3.6.3 Robustness of Optimal Control

Optimal control is not only intuitively appealing, the resulting control laws inherently possess certain robustness properties (Glad 1987). One important property regards the gain margin.

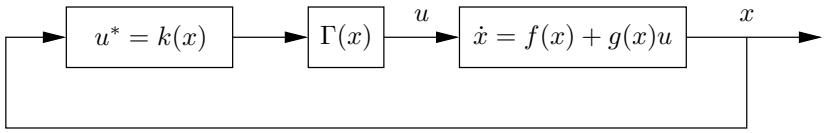


Figure 3.2: The optimal control law $u^* = k(x)$ remains globally stabilizing for any scalar gain perturbation $\Gamma(x) \geq \frac{1}{2}$.

Assume that the prescribed optimal control input (3.23) cannot be produced exactly, but that the actual control input is

$$u = \Gamma(x)u^* \tag{3.28}$$

where $\Gamma(x) > 0$ is a scalar, see Figure 3.2. Actuator saturation, for example, can be modeled as gain reduction, $\Gamma(x) < 1$. Are optimal controllers robust to such changes in the gain? The control law (3.28) is globally stabilizing provided that

$$\dot{V} = V_x f + V_x g u = V_x f + \Gamma V_x g u^*$$

is negative definite. From the assumptions and (3.26) we know that

$$-q = V_x f + \frac{1}{2} V_x g u^*$$

is negative definite. Combining these two equations yields

$$\dot{V} = -q + (\Gamma - \frac{1}{2}) V_x g u^* = -q(x) - (\Gamma(x) - \frac{1}{2}) \cdot \underbrace{\frac{1}{2} V_x(x) g(x) R^{-1}(x) (V_x(x) g(x))^T}_{\text{positive (semi-)definite}}$$

Apparently, $\dot{V}$ is negative definite (at least) for all $\Gamma(x) \geq \frac{1}{2}$. Thus, all state feedback control laws which solve an optimal control problem of the type considered in Section 3.6.2, have infinite gain margin and a 50% gain reduction tolerance.

Note that the actual tolerable gain reduction may be more than 50%. In Example 3.2, any control law $u = -k\tilde{x}_2$ where $k > 1$ makes $\dot{V}$ negative definite and hence is globally stabilizing. The selected control law $u = -3\tilde{x}_2$ thus remains globally stabilizing for any gain perturbation $\Gamma(x) > \frac{1}{3}$.

3.7 Outline of Part I

In the following three chapters, backstepping is used to solve a number of nonlinear control problems. In Chapters 4 and 5 we design control laws for some specific nonlinear systems related to flight control, and in Chapter 6 we discuss how to modify these control laws to suppress the effects of external disturbances.

In Chapter 4 we consider two nonlinear SISO systems appearing in angle of attack control and flight path angle control, respectively. We show that for both

these systems, backstepping can be used to design linear control laws with infinite gain margin and a certain amount of gain reduction tolerance. These control laws are compared with the corresponding feedback linearization designs.

In Chapter 5 we propose a control law for general rigid body motion, treating the dynamics as a MIMO system. Under certain assumptions, the resulting control law can be used to control the speed, angle of attack, sideslip angle, and velocity vector roll rate of an aircraft.

Chapter 6 deals with output regulation in the presence of constant disturbances entering at the input of a nonlinear system. Two methods for robustifying a nominal controller are developed, based on adaptive backstepping and nonlinear observer techniques, respectively, and compared with adding regular integral control.

Chapter 4

Two Backstepping Designs

A trademark of backstepping control design is the possibility to benefit from naturally stabilizing nonlinear terms in the system dynamics. This was illustrated in Example 3.2. To retain such nonlinearities, rather than cancel them using feedback as in feedback linearization, has several potential benefits. Less control effort may be needed to control the system, the resulting control law may depend on less precise model information which improves the robustness against modeling errors, and global stability may be achieved in cases where feedback linearization can only be performed locally.

In this chapter, all of these features will be illustrated as we design backstepping controllers for two generic nonlinear systems, one second order and one third order system. The two systems considered are motivated by their appearance in aircraft flight control—in angle of attack control and flight path angle control, respectively.

The second order system is given by

$$\begin{aligned} \dot{x}_1 &= \phi(x_1) + x_2 \\ \dot{x}_2 &= u \end{aligned} \tag{4.1}$$

In angle of attack control, u is the pitch angular acceleration produced by the control surfaces, $x_1 = \alpha$ is the angle of attack, $x_2 = q$ is the pitch rate, and ϕ is determined by the lift force. Another nonlinear system of this type appears in jet engine control, see Krstić et al. (1998).

Several nonlinear control designs have been suggested for this problem, considering x_1 as the controlled variable. Lane and Stengel (1988) use feedback linearization to cancel the effects of ϕ and obtain a linear closed loop system. The resulting control law depends on $\phi(x_1)$ as well as on $\phi'(x_1)$. Snell et al. (1992), Bugajski and Enns (1992), Reiner et al. (1996) and Adams et al. (1994) use feedback linearization combined with time-scale separation (Lu and Shen 2002). With this technique, the control law only depends on $\phi(x_1)$, but no precise bounds on the feedback gains for stability to hold can be given. Krstić et al. (1998) design a backstepping controller which relies only on the knowledge of the maximum positive slope of $\phi(x_1)$. The authors only consider regulation about the origin, and a particular choice of ϕ.

In this chapter, we extend the results of Krstić et al. (1998) to set-point regulation for a class of nonlinear functions ϕ. We also use backstepping to show that for the previously proposed control laws based on feedback linearization and time-scale separation, simple bounds on the feedback gains exist that guarantee global asymptotic stability.

The third order system that we consider is given by

$$\begin{aligned} \dot{x}_1 &= \phi(x_2 - x_1) \\ \dot{x}_2 &= x_3 \\ \dot{x}_3 &= u \end{aligned} \tag{4.2}$$

In flight path angle control, u is again the pitch angular acceleration, $x_1 = \gamma$ is the flight path angle, $x_2 = \theta$ is the pitch angle, $x_3 = q$ is the pitch rate, and as above, ϕ is determined by the lift force. Applying feedback linearization, as suggested by Lane and Stengel (1988), leads to a control law that depends on $\phi(x)$, $\phi'(x)$, and $\phi''(x)$. Further, this control law becomes singular when $\phi'(x) = 0$ which occurs around the stall angle in flight path angle control. In this chapter, a new backstepping design is proposed, which depends only on the zero crossing of ϕ and achieves global stability even in cases where the system is not globally feedback linearizable.

This chapter is organized as follows. In Section 4.1 and Section 4.2, backstepping control laws for the second order system (4.1), and for the third order system (4.2), are derived. Their properties are investigated and the control laws are compared with feedback linearizing control laws for the same systems. In Section 4.3, we apply the designs to flight control, and develop control laws for maneuvering flight control (angle of attack control, sideslip regulation, and roll control) and for flight path angle control. Section 4.4 completes the chapter, and contains the conclusions.

Section 4.1 is based on

> O. Härkegård and S. T. Glad. Flight control design using backstepping. In *Proc. of the 5th IFAC Symposium on Nonlinear Control Systems*, pages 259–264, St. Petersburg, Russia, July 2001b,

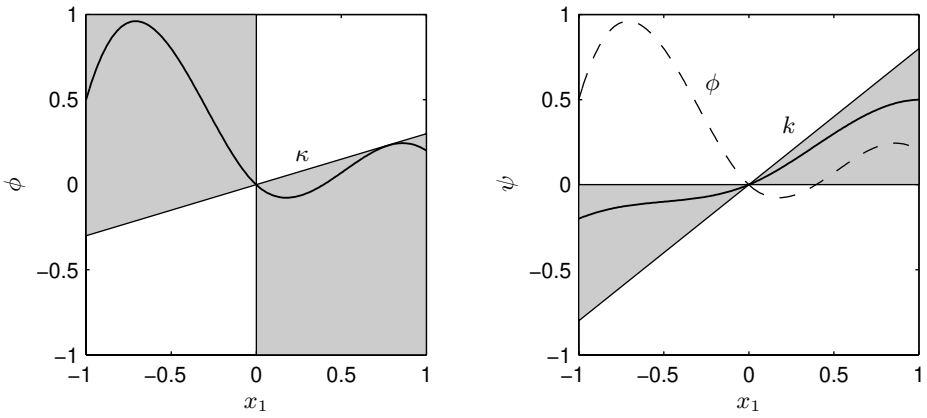
and Section 4.2 is based on

> O. Härkegård and S. T. Glad. A backstepping design for flight path angle control. In *Proc. of the 39th Conference on Decision and Control*, pages 3570–3575, Sydney, Australia, Dec. 2000.

4.1 A Second Order System

Consider the second order nonlinear system

$$\begin{aligned} \dot{x}_1 &= \phi(x_1) + x_2 \\ \dot{x}_2 &= u \\ y &= x_1 \end{aligned} \tag{4.3}$$



(a) System nonlinearity, ϕ.

(b) Virtual control law, ψ.

Figure 4.1: Examples of ϕ and ψ that satisfy (4.4), (4.6)–(4.8). Both functions must be sector bounded for a globally stabilizing control law of the form (4.5) to exist.

where the nonlinearity ϕ is characterized below. This system is on strict-feedback form, see (3.15), and has relative degree two. Hence, backstepping as well as feedback linearization can be applied. In Sections 4.1.1–4.1.5 we consider stabilization around the origin and in Section 4.1.6 we extend the designs to set-point regulation.

4.1.1 Backstepping Design

To begin with, let the control objective be to achieve GAS at the origin, $x_1 = x_2 = 0$. The nonlinearity ϕ is assumed to satisfy the following conditions, illustrated in Figure 4.1(a).

Assumption 4.1 *Let $\phi(0) = 0$ and let ϕ satisfy the growth condition*

$$\kappa = \max_{x_1 \neq 0} \frac{\phi(x_1)}{x_1} < \infty \tag{4.4}$$

The proposed backstepping control law is given by the following theorem. The proof of the theorem contains the actual backstepping design.

Theorem 4.1 *Consider the system (4.3) and let Assumption 4.1 hold. Then the control law*

$$u = -k(x_2 + \psi(x_1)) \tag{4.5}$$

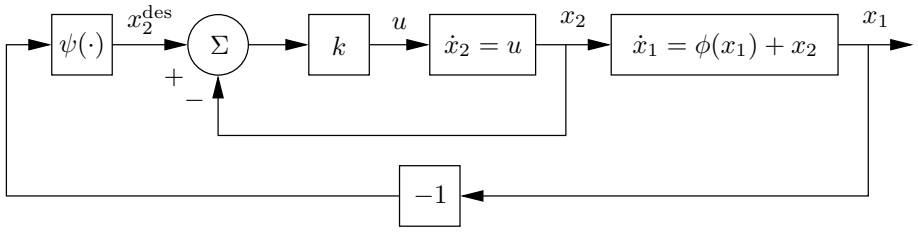


Figure 4.2: The nonlinear system (4.3) can be globally stabilized using a cascaded control law.

where $k \in \mathbb{R}$ *and* $\psi : \mathbb{R} \mapsto \mathbb{R}$ *satisfy*

$$(\phi(x_1) - \psi(x_1))x_1 < 0, \quad x_1 \neq 0 \tag{4.6}$$

$$\psi'(x_1) > 0, \quad x_1 \neq 0 \tag{4.7}$$

$$k > \psi'(x_1), \quad \forall x_1 \tag{4.8}$$

makes the origin GAS.

Figure 4.2 illustrates the control law (4.5) implemented as a cascaded controller, where the outer loop generates the desired value of x_2, $x_2^{\text{des}} = -\psi(x_1)$, which the inner loop tracks. Figure 4.1 illustrates the conditions on ψ and k in the theorem, which can be interpreted as follows.

- (4.6): When $x_2 = x_2^{\text{des}}$, the resulting outer loop dynamics, $\dot{x}_1 = \phi(x_1) - \psi(x_1)$, must be stable.
- (4.7): $\partial u / \partial x_1 = -k\psi'(x_1) < 0$, that is, the control law must provide negative feedback from x_1 everywhere, even "locally".
- (4.8): The inner control loop in Figure 4.2 must be faster than the outer loop.

Proof: The proof is based on Krstić et al. (1998).

Step 1: Start by finding a globally stabilizing virtual control law for the dynamics of x_1 in (4.3), regarding x_2 as the control variable. Pick a virtual control law of the form

$$x_2^{\text{des}} = -\psi(x_1) \tag{4.9}$$

along with the clf

$$W(x_1) = \frac{1}{2}x_1^2$$

Taking the time derivative of W yields

$$\dot{W}|_{x_2 = x_2^{\text{des}}} = \big(\phi(x_1) - \psi(x_1)\big)x_1$$

which is negative definite if (4.6) holds.

Step 2: Continue by introducing the residual

$$\tilde{x}_2 = x_2 - x_2^{\text{des}} = x_2 + \psi(x_1)$$

and rewrite the dynamics (4.3) in terms of x_1 and $\tilde{x}_2$.

$$\begin{aligned} \dot{x}_1 &= \phi(x_1) - \psi(x_1) + \tilde{x}_2 \\ \dot{\tilde{x}}_2 &= u + \psi'(x_1)\big(\phi(x_1) - \psi(x_1) + \tilde{x}_2\big) \end{aligned} \tag{4.10}$$

As in Example 3.2, we use a nonquadratic clf to avoid having to cancel the x_1 dependencies in $\dot{\tilde{x}}_2$ with the control signal u, and pick the clf

$$V(x_1, \tilde{x}_2) = F(x_1) + \frac{1}{2}\tilde{x}_2^2$$

for the total system (4.10). Here, F is any valid clf for the x_1-subsystem, see Definition 3.3. Specifically this means that

$$\dot{F}(x_1)|_{x_2 = x_2^{\text{des}}} = F'(x_1)\big(\phi(x_1) - \psi(x_1)\big) = -U(x_1) \tag{4.11}$$

where $U(x_1)$ is positive definite. Differentiating V w.r.t. time we get

$$\begin{aligned} \dot{V} &= F'(x_1)\big[\phi(x_1) - \psi(x_1) + \tilde{x}_2\big] + \tilde{x}_2\big[u + \psi'(x_1)\big(\phi(x_1) - \psi(x_1) + \tilde{x}_2\big)\big] \\ &= -U(x_1) + \tilde{x}_2\big[F'(x_1) + u + \psi'(x_1)\big(\phi(x_1) - \psi(x_1)\big) + \psi'(x_1)\tilde{x}_2\big] \end{aligned}$$

The complexity of the second term is reduced by selecting F such that the x_1 terms inside the brackets cancel each other. This is achieved by

$$F'(x_1) = -\psi'(x_1)\big(\phi(x_1) - \psi(x_1)\big), \quad F(0) = 0 \tag{4.12}$$

Inserting this into (4.11) yields

$$U(x_1) = \psi'(x_1)\big(\phi(x_1) - \psi(x_1)\big)^2$$

For $U(x_1)$ to be positive definite, (4.7) must hold. Note that $\phi(x_1) \neq \psi(x_1)$ holds according to (4.6). We now have that

$$\dot{V} = -U(x_1) + \tilde{x}_2\big[u + \psi'(x_1)\tilde{x}_2\big]$$

To make $\dot{V}$ negative definite, select u such that the positive definite, destabilizing term $\psi'(x_1)\tilde{x}_2^2$ is cancelled. If $\psi'(x_1)$ is bounded from above, then u can be selected linear in $\tilde{x}_2$.

$$u = -k\tilde{x}_2 = -k\big(x_2 + \psi(x_1)\big) \tag{4.13}$$

renders

$$\dot{V} = -U(x_1) - (k - \psi'(x_1))\tilde{x}_2^2$$

negative definite if (4.8) holds, thus making the origin of (4.3) GAS. Assumption 4.1 must hold or $\psi'(x_1)$ will not be bounded since

$$\psi'(x_1) \geq \frac{\psi(x_1)}{x_1} > \frac{\phi(x_1)}{x_1}$$

according to (4.6). □

4.1.2 Inverse Optimality

Before discussing which choices of ψ that might be of interest, let us examine the robustness properties of the derived control law (4.5). Using the tools in Section 3.6, the control law can be shown to be optimal with respect to a meaningful cost functional, provided that k is chosen properly.

Theorem 4.2 *If*

$$k > 2 \cdot \max_{x_1} \psi'(x_1) \tag{4.14}$$

then the control law (4.5) is inverse optimal w.r.t. the cost functional

$$\int_0^\infty \Big(\psi'(x_1)(\phi(x_1) - \psi(x_1))^2 + (\frac{k}{2} - \psi'(x_1))(x_2 + \psi(x_1))^2 + \frac{1}{2k}u^2\Big)dt$$

Proof: The system (4.3) is affine in the control input, hence the tools in Section 3.6 can be used. In this proof, we will use the transformed system description (4.10) to compute the cost functional that is minimized since the expressions then become simpler. Comparing (4.10) with (3.21), we have that

$$f(x) = \begin{pmatrix} \phi(x_1) - \psi(x_1) + \tilde{x}_2 \\ \psi'(x_1)(\phi(x_1) - \psi(x_1) + \tilde{x}_2) \end{pmatrix}, \quad g(x) = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

From (4.12) we also have that

$$V_x = \begin{pmatrix} F'(x_1) & \tilde{x}_2 \end{pmatrix} = \begin{pmatrix} -\psi'(x_1)(\phi(x_1) - \psi(x_1)) & \tilde{x}_2 \end{pmatrix}$$

Inserting this into (3.26) and (3.27) yields

$$\begin{aligned} R(x) &= \frac{\tilde{x}_2}{2k\tilde{x}_2} = \frac{1}{2k} \\ q(x) &= \psi'(x_1)(\phi(x_1) - \psi(x_1))(\phi(x_1) - \psi(x_1) + \tilde{x}_2) \\ &\quad - \tilde{x}_2\psi'(x_1)(\phi(x_1) - \psi(x_1) + \tilde{x}_2) + \frac{1}{2}\tilde{x}_2 \cdot k\tilde{x}_2 \\ &= \psi'(x_1)(\phi(x_1) - \psi(x_1))^2 + (\frac{k}{2} - \psi'(x_1))\tilde{x}_2^2 \end{aligned}$$

Apparently, to make $q(x)$ positive definite, which is required for the cost functional to be "meaningful", k should be chosen such that

$$k > 2 \cdot \max_{x_1} \psi'(x_1)$$

□

This inverse optimality property of the control law means that it has certain robustness properties, such as infinite gain margin, 50% gain reduction tolerance, and 60° phase margin (Glad 1987). Note that the gain reduction tolerance follows from that the lower limit for inverse optimality (4.14) is twice the limit for GAS (4.8).

4.1.3 Selecting the Virtual Control Law

Theorem 4.1 specifies a family of control laws, all of which globally asymptotically stabilize the system (4.3) at the origin. How should the virtual control law $x_2^{\text{des}} = -\psi(x_1)$ be chosen? Let us investigate two particular choices, one that has control law simplicity in focus, and one that aim at linearizing the dynamics.

Linear Control

First consider the linear control choice $\psi(x_1) = k_1 x_1$. This is the control law proposed by Krstić et al. (1998).

Corollary 4.1 *Consider the system (4.3) and let Assumption 4.1 hold. Then the linear control law*

$$u = -k_2(x_2 + k_1 x_1) \tag{4.15}$$

where

$$k_2 > k_1 > \max\{0, \kappa\}$$

with κ as defined in (4.4), makes the origin GAS. In addition, for $k_2 > 2k_1$ the control law minimizes the meaningful cost functional

$$\int_0^\infty \Big(k_1(\phi(x_1) - k_1 x_1)^2 + (\frac{k_2}{2} - k_1)(x_2 + k_1 x_1)^2 + \frac{1}{2k_2} u^2 \Big) dt$$

Proof: Selecting $\psi(x_1) = k_1 x_1$ and $k = k_2$ in Theorem 4.1 yields the control law $u = -k_2(x_2 + k_1 x_1)$ and leads to the conditions

- (4.6): $(\phi(x_1) - k_1 x_1)x_1 \leq (\kappa - k_1)x_1^2 < 0, \quad x_1 \neq 0 \quad \Longleftrightarrow \quad k_1 > \kappa$
- (4.7), (4.8): $k_2 > k_1 > 0$

The cost functional follows directly from Theorem 4.2. □

Note how little information about the system nonlinearity ϕ this control law is dependent of. Only an upper bound on its growth rate, κ, is needed. In particular, if ϕ is known to lie in the second and fourth quadrants only, thus intuitively being useful for stabilizing x_1, we do not need any further information, since then $\kappa < 0$ and the parameter restriction $k_1 > 0$ becomes active.

Linearizing Control

Next, consider the linearizing virtual control law $\psi(x_1) = k_1 x_1 + \phi(x_1)$, which aims at cancelling the natural dynamics of x_1.

Corollary 4.2 *Consider the system (4.3) and let Assumption 4.1 hold. Then the partially linearizing control law*

$$u = -k_2(x_2 + k_1 x_1 + \phi(x_1)) \tag{4.16}$$

where

$$\begin{aligned} k_1 &> \max\{0, -\min_{x_1} \phi'(x_1)\} \\ k_2 &> k_1 + \max_{x_1} \phi'(x_1) \end{aligned} \tag{4.17}$$

makes the origin GAS, provided that such upper and lower bounds on ϕ' exist. In addition, for $k_2 > 2(k_1 + \max_{x_1} \phi'(x_1))$ the control law minimizes the meaningful cost functional

$$\int_0^\infty \Big((k_1 + \phi'(x_1)) k_1^2 x_1^2 + (\frac{k_2}{2} - k_1 - \phi'(x_1))(x_2 + k_1 x_1 + \phi(x_1))^2 + \frac{1}{2k_2} u^2 \Big) dt$$

Proof: Selecting $\psi(x_1) = k_1 x_1 + \phi(x_1)$ and $k = k_2$ in Theorem 4.1 yields the control law $u = -k_2(x_2 + k_1 x_1 + \phi(x_1))$ and the conditions

- (4.6): $(\phi(x_1) - k_1 x_1 - \phi(x_1))x_1 = -k_1 x_1^2 < 0, \quad x_1 \neq 0 \quad \Longleftrightarrow \quad k_1 > 0$
- (4.7): $k_1 + \phi'(x_1) > 0 \quad \Longleftrightarrow \quad k_1 > -\min_{x_1} \phi'(x_1)$
- (4.8): $k_2 > k_1 + \phi'(x_1) \quad \Longleftrightarrow \quad k_2 > k_1 + \max_{x_1} \phi'(x_1)$

The cost functional follows directly from Theorem 4.2. □

This control law corresponds to selecting the virtual control law

$$x_2^{\text{des}} = -\phi(x_1) - k_1 x_1$$

If this virtual control law could be produced exactly it would cancel the nonlinear dynamics of x_1 and replace them with linear dynamics, $-k_1 x_1$. Condition (4.17) states that it is all right to cancel the natural dynamics, $\phi(x_1)$, as long as the new dynamics, $-k_1 x_1$, provide at least the same amount of negative feedback. This is sound also from an optimality point of view. Intuitively, it must be suboptimal to use the control signal to slow down the natural system dynamics.

4.1.4 Feedback Linearization

In the preceding sections, we have used backstepping to design globally stabilizing control laws for the system (4.3). The backstepping control laws are inverse optimal with respect to meaningful cost functionals, and depend on the nonlinearity ϕ at a varying extent.

We will now instead use feedback linearization for control design. We will use regular feedback linearization (Isidori 1995), as well as feedback linearization based on time-scale separation (Lu and Shen 2002) between the two dynamics of x_1 and x_2.

Feedback Linearization

To perform feedback linearization, we introduce the new variables

$$\begin{aligned}\dot{x}_1 &= \phi(x_1) + x_2 \triangleq \tilde{x}_2 \\ \dot{\tilde{x}}_2 &= \phi'(x_1)(\phi(x_1) + x_2) + u \triangleq \tilde{u}\end{aligned}$$

This leads to the system description

$$\begin{aligned}\dot{x}_1 &= \tilde{x}_2 \\ \dot{\tilde{x}}_2 &= \tilde{u}\end{aligned}$$

which is a double integrator from $\tilde{u}$ to x_1. To stabilize this system, any linear control law

$$\tilde{u} = -l_1 x_1 - l_2 \tilde{x}_2, \quad l_1 > 0,\ l_2 > 0$$

can be selected. In the original variables we get the control law

$$\begin{aligned}u &= -l_1 x_1 - l_2(\phi(x_1) + x_2) - \phi'(x_1)(\phi(x_1) + x_2) \\ &= -l_1 x_1 - (l_2 + \phi'(x_1))(\phi(x_1) + x_2)\end{aligned} \tag{4.18}$$

This control law differs from the backstepping control law (4.16) only in the last term $\phi'(x_1)(\phi(x_1) + x_2)$. The first two terms can be made equal to (4.16) by selecting $l_1 = k_1 k_2$, $l_2 = k_2$.

Feedback Linearization with Time-Scale Separation

Next, we consider feedback linearization coupled with time-scale separation. In time-scale separation, the idea is to stabilize one loop at the time, assuming the inner loops to be much faster than the outer loops (Lu and Shen 2002).

First consider the dynamics of x_1 and design a linearizing control law regarding x_2 as the control variable. This gives

$$x_2^{\text{des}} = -\phi(x_1) - k_1 x_1$$

Now consider x_2^{des} as a reference for x_2 and use u to steer x_2 towards x_2^{des}.

$$\dot{x}_2 = u = k_2(x_2^{\text{des}} - x_2)$$

which can be rewritten as

$$u = -k_2(x_2 + k_1 x_2 + \phi(x_1)) \tag{4.19}$$

To ensure stability based on time-scale separation arguments, $k_2 \gg k_1$ must be selected (Lu and Shen 2002). However, note that (4.19) is identical to the backstepping control law (4.16). For the latter control law we know from Corollary 4.2 that $k_2 > k_1 + \max_{x_1} \phi'(x_1)$ suffices to guarantee stability.

4.1.5 Example

Consider the system

$$\begin{aligned}\dot{x}_1 &= x_1 \sin x_1 + x_2\\ \dot{x}_2 &= u\end{aligned}$$

which matches the second order system (4.3) with $\phi(x_1) = x_1 \sin x_1$, illustrated in Figure 4.3. Let the control objective be to stabilize this system around the origin. ϕ satisfies Assumption 4.1 with

$$\kappa = \max_{x_1 \neq 0} \frac{\phi(x_1)}{x_1} = \max_{x_1 \neq 0} \sin x_1 = 1$$

Applying Corollary 4.1 gives the linear backstepping control law

$$u = -k_2(x_2 + k_1 x_1)$$

where $k_2 > k_1 > \kappa = 1$ to make the origin GAS. With $k_1 = 1.2$, $k_2 = 2.5$ this control law minimizes the cost functional

$$\int_0^\infty \Big(1.2(1.2 - \sin x_1)x_1^2 + 0.05(x_2 + 1.2x_1)^2 + 0.2u^2\Big)dt$$

The partially linearizing backstepping control law in Corollary 4.2 cannot be applied here since $\phi'(x_1) = \sin x_1 + x_1 \cos x_1$ is unbounded for $x_1 \in \mathbb{R}$. The feedback linearization control law (4.18) becomes

$$u = -l_1 x_1 - (l_2 + \sin x_1 + x_1 \cos x_1)(x_1 \sin x_1 + x_2)$$

which is quadratic in x_1. Selecting $l_1 = 2$, $l_2 = 3$ places the closed loop poles in -1 and -2.

Figure 4.4 shows the simulation results when the two control laws are applied. Each control law is tested for the initial conditions $x_1(0) = \pm 20$ and $x_2(0) = -\phi(x_1(0))$ so that $\dot{x}_1(0) = 0$.

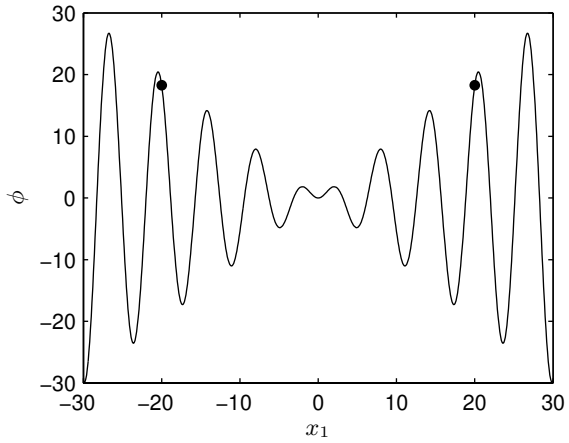


Figure 4.3: The nonlinearity $\phi(x_1) = x_1 \sin x_1$. The dots represent the two initial conditions in the simulations.

With feedback linearization, the x_1 response is that of a second order linear system with poles in -1 and -2. Note for example that the responses for $x_1(0) = \pm 20$ are the same but with opposite signs. The price to pay for this linearity is that x_2 and u become very oscillatory.

With backstepping, the response in x_1 is no longer linear, and its convergence properties vary with the initial condition. The benefit is that neither x_2 nor u oscillate, and that less overall control effort is needed to stabilize the system.

4.1.6 Set-Point Regulation

We now extend our results to set-point regulation and consider the problem of making $y = x_1 = r$ a GAS equilibrium. To do this, we need to update Assumption 4.1.

Assumption 4.2 *Let ϕ satisfy*

$$\kappa = \max_{\substack{x_1 \neq r \\ r \in \mathbb{R}}} \frac{\phi(x_1) - \phi(r)}{x_1 - r} = \max_{x_1} \phi'(x_1) < \infty$$

The following corollary extends Theorem 4.1.

Corollary 4.3 *Consider the system (4.3) and let Assumption 4.2 hold. Let r be a constant reference signal. Then the control law*

$$u = -k(\phi(r) + x_2 + \psi(x_1 - r)) \tag{4.20}$$

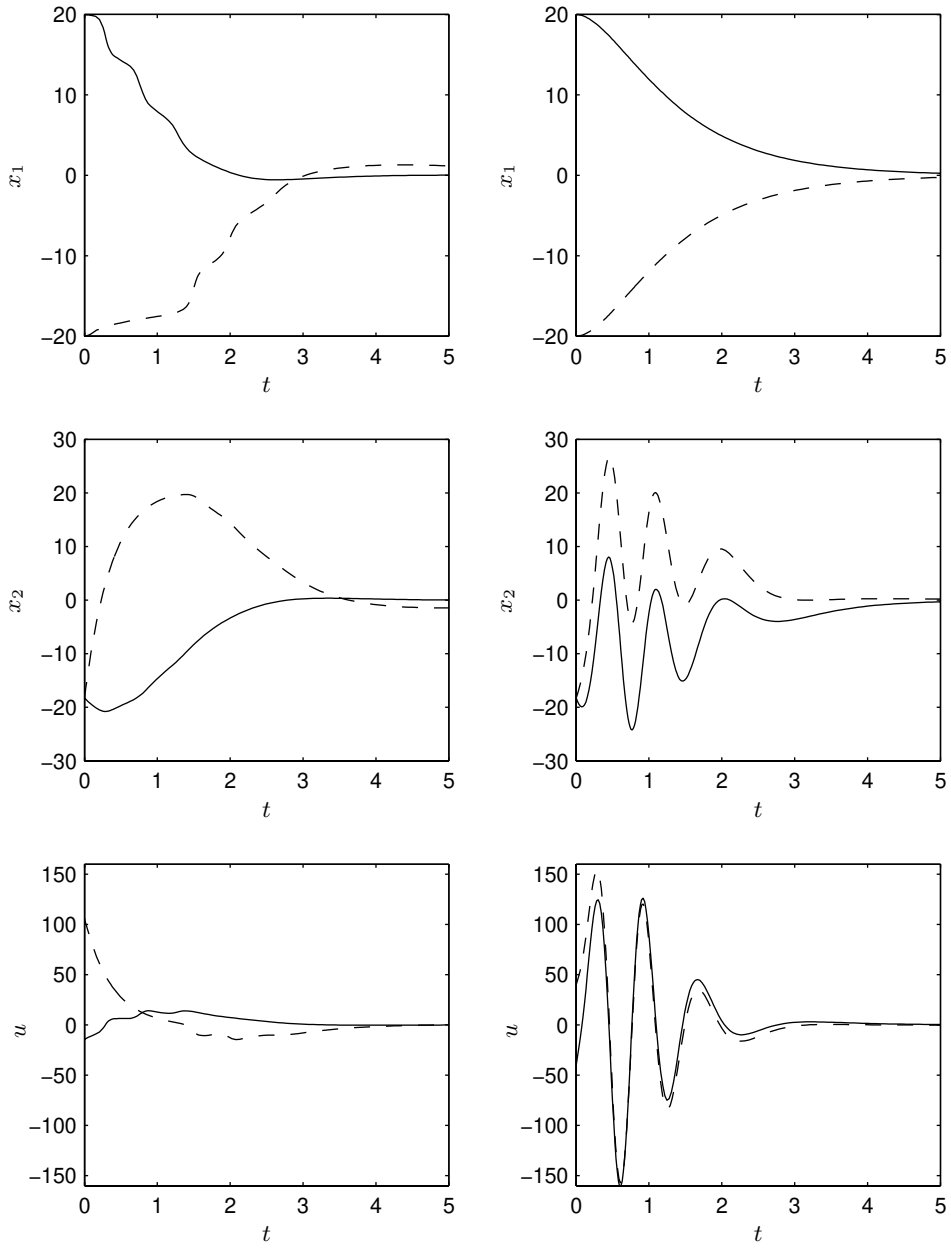


Figure 4.4: Simulation results using backstepping (left) and feedback linearization (right) for the initial conditions $x_1(0) = 20$ (solid) and $x_1(0) = -20$ (dashed). Feedback linearization gives x_1 linear convergence properties at the price of large oscillations in x_2 and u.

where

$$\begin{aligned}(\phi(x_1)-\phi(r)-\psi(x_1-r))(x_1-r)<0, \quad & x_1\neq r\\ \psi'(x_1)>0, \quad & x_1\neq r\\ k>\psi'(x_1), \quad & \forall x_1\end{aligned}$$

makes $y=r$ a GAS equilibrium.

Proof: Introduce

$$\begin{aligned}z_1&=x_1-r\\ z_2&=x_2+\phi(r)\end{aligned}$$

which gives the control objective $z_1=z_2=0$ and the dynamics

$$\begin{aligned}\dot{z}_1&=\underbrace{\phi(z_1+r)-\phi(r)}_{\eta_r(z_1)}+z_2\\ \dot{z}_2&=u\end{aligned}$$

where η_r satisfies Assumption 4.1 for all $r\in\mathbb{R}$. Applying Theorem 4.1 gives the control law

$$u=-k(z_2+\psi(z_1))=-k(x_2+\phi(r)+\psi(x_1-r))$$

The requirements on k and ψ follow directly from (4.6)–(4.8). □

It is straightforward to also extend Corollary 4.1 and Corollary 4.2 to set-point regulation. Note that κ, used in Corollary 4.1, should be redefined as in Assumption 4.2.

4.2 A Third Order System

In this section, we consider control of the third order system

$$\begin{aligned}\dot{x}_1&=\phi(x_2-x_1)\\ \dot{x}_2&=x_3\\ \dot{x}_3&=u\\ y&=x_1\end{aligned}\tag{4.21}$$

where ϕ is characterized below. For this system, we propose a new backstepping design, which also leads to a linear control law.

4.2.1 Backstepping Design

We start by considering stabilization around the origin.

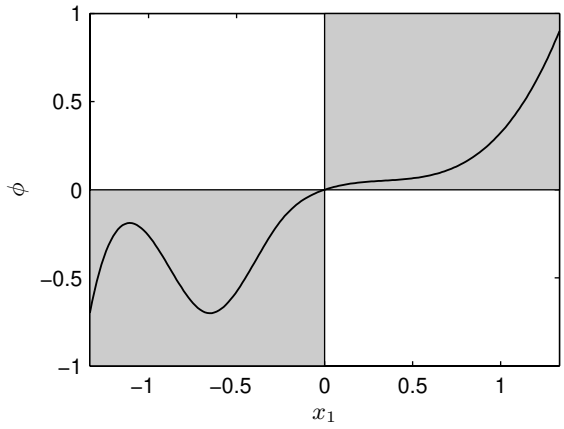


Figure 4.5: Example of a ϕ that satisfies Assumption 4.3. The linear control law (4.23) is globally stabilizing for all functions ϕ that only pass through the first and third quadrants.

Assumption 4.3 *Let ϕ satisfy the sign condition*

$$x\phi(x) > 0, \quad x \neq 0 \tag{4.22}$$

Figure 4.5 illustrates the assumption. The key idea in the backstepping design below is the following. For $x_2 = 0$, we get $\dot{x}_1 = \phi(-x_1)$, which acts stabilizing since $\phi(-x_1)$ is $\phi(x_1)$ mirrored about the y-axis, and thus lies in the second and fourth quadrants. Using backstepping, we will show how to utilize this inherent stability property.

Theorem 4.3 *Consider the system (4.21) and let Assumption 4.3 hold. Then the linear control law*

$$u = -k_3(x_3 + k_2(x_2 + k_1 x_1)) \tag{4.23}$$

where

$$\begin{aligned} &k_1 > -1 \\ &k_2 > 0 \\ &k_3 > \begin{cases} k_2 & \text{if } k_1 \leq 0 \\ k_2(1+k_1) & \text{if } k_1 > 0 \end{cases} \end{aligned} \tag{4.24}$$

makes the origin GAS.

Figure 4.6 illustrates the linear control law (4.23), implemented as a cascaded controller. Note that the theorem gives precise bounds on the feedback gains in the three loops for GAS to hold.

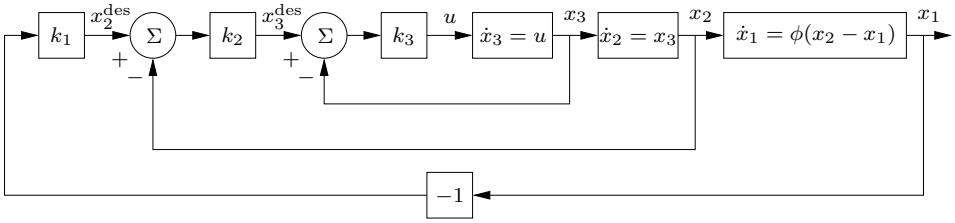


Figure 4.6: The nonlinear system (4.21) is globally stabilized by a linear cascaded controller.

Proof: In this proof, k_i are constants which parameterize the control law, while c_i are "dummy constants" whose values will be assigned during the derivation to simplify various expressions.

Step 1: As usual, start by considering only the x_1-subsystem of (4.21). To find a globally stabilizing virtual control law, we use the clf

$$V_1 = \frac{1}{2}x_1^2 \tag{4.25}$$

Considering x_2 as our virtual control input

$$\begin{aligned}\dot{V}_1 &= x_1\phi(x_2 - x_1)\\ &= x_1\phi(-(1+k_1)x_1 + x_2 + k_1x_1)\\ &= x_1\phi(-(1+k_1)x_1) < 0, \quad x_1 \neq 0\end{aligned}$$

can be achieved by selecting

$$x_2 = x_2^{\text{des}} = -k_1x_1, \quad k_1 > -1 \tag{4.26}$$

The fact that $k_1 = 0$ is a valid choice means that x_1 feedback is not necessary for the sake of stabilization. However, it provides an extra degree of freedom for tuning the closed loop performance.

Step 2: Since we cannot control x_2 directly, we continue by introducing the deviation from the virtual control law.

$$\tilde{x}_2 = x_2 - x_2^{\text{des}} = x_2 + k_1x_1$$

Including the x_2 dynamics in (4.21) we get

$$\begin{aligned}\dot{x}_1 &= \phi(\xi)\\ \dot{\tilde{x}}_2 &= x_3 + k_1\phi(\xi)\end{aligned}$$

where

$$\xi = -(1+k_1)x_1 + \tilde{x}_2 \tag{4.27}$$

has been introduced. We will also need

$$\dot{\xi} = -(1+k_1)\phi(\xi) + x_3 + k_1\phi(\xi) = -\phi(\xi) + x_3$$

A regular backstepping design would proceed by expanding the clf (4.25) with a term penalizing $\tilde{x}_2$. We do this, but also add a term $F(\xi)$ as an extra degree of freedom, where F is required to be positive definite. Hence,

$$V_2 = \frac{c_1}{2}x_1^2 + \frac{1}{2}\tilde{x}_2^2 + F(\xi), \quad c_1 > 0$$

We compute its time derivative to find a new virtual control law, x_3^{des}.

$$\begin{aligned}\dot{V}_2 &= c_1 x_1\phi(\xi) + \tilde{x}_2(x_3 + k_1\phi(\xi)) + F'(\xi)(-\phi(\xi) + x_3)\\ &= (c_1x_1 + k_1\tilde{x}_2 - F'(\xi))\phi(\xi) + (\tilde{x}_2 + F'(\xi))x_3^{\text{des}} + (\tilde{x}_2 + F'(\xi))(x_3 - x_3^{\text{des}})\end{aligned}$$

Although it may not be obvious, we can again find a stabilizing function independent of ϕ. Choosing

$$x_3^{\text{des}} = -k_2\tilde{x}_2, \quad k_2 > 0 \tag{4.28}$$

$$F'(\xi) = c_2\phi(\xi), \quad F(0) = 0, \quad c_2 > 0 \tag{4.29}$$

where (4.29) is an implicit but perfectly valid choice of F, yields

$$\dot{V}_2|_{x_3 = x_3^{\text{des}}} = \underbrace{(c_1x_1 + (k_1 - k_2c_2)\tilde{x}_2)}_{(k_1-k_2c_2)\xi}\phi(\xi) - c_2\phi(\xi)^2 - k_2\tilde{x}_2^2$$

To make the first term negative definite using the assumed property (4.22), we select c_1 to make the factor in front of $\phi(\xi)$ proportional to $-\xi$, see (4.27). This is achieved by

$$c_1 = -(1+k_1)(k_1 - k_2c_2), \quad k_2c_2 > k_1 \tag{4.30}$$

With this choice,

$$\dot{V}_2|_{x_3 = x_3^{\text{des}}} = (k_1 - k_2c_2)\xi\phi(\xi) - c_2\phi(\xi)^2 - k_2\tilde{x}_2^2$$

becomes negative definite. The benefit of using the extra term $F(\xi)$ shows up in Equation (4.30). $F(\xi) \equiv 0$ (or equally, $c_2 = 0$) leads to $c_1 = -(1+k_1)k_1 > 0$ and the severe restriction $-1 < k_1 < 0$ implying positive feedback from x_1.

Step 3: The final backstepping iteration begins with introducing the third residual

$$\tilde{x}_3 = x_3 - x_3^{\text{des}} = x_3 + k_2\tilde{x}_2$$

We also update the system description

$$\begin{aligned}\dot{x}_1 &= \phi(\xi)\\ \dot{\tilde{x}}_2 &= \tilde{x}_3 - k_2\tilde{x}_2 + k_1\phi(\xi)\\ \dot{\tilde{x}}_3 &= u + k_2(\tilde{x}_3 - k_2\tilde{x}_2 + k_1\phi(\xi))\end{aligned} \tag{4.31}$$

Furthermore,

$$\dot{\xi} = -\phi(\xi) + \tilde{x}_3 - k_2\tilde{x}_2$$

V_3 is constructed by adding a term penalizing $\tilde{x}_3$ to V_2.

$$V_3 = c_3V_2 + \frac{1}{2}\tilde{x}_3^2, \quad c_3 > 0$$

We get

$$\begin{aligned}\dot{V}_3 = c_3\big[&\underbrace{(k_1 - k_2c_2)\xi\phi(\xi)}_{\text{negative definite}} - c_2\phi(\xi)^2 - k_2\tilde{x}_2^2 + \tilde{x}_3(\tilde{x}_2 + c_2\phi(\xi))\big] \\ &+ \tilde{x}_3\big[u + k_2(\tilde{x}_3 - k_2\tilde{x}_2 + k_1\phi(\xi))\big] \\ \leq &- c_2c_3\phi^2(\xi) - k_2c_3\tilde{x}_2^2 + \tilde{x}_3\big[u + k_2\tilde{x}_3 + (c_3 - k_2^2)\tilde{x}_2 + (k_1k_2 + c_2c_3)\phi(\xi)\big]\end{aligned}$$

once again using (4.22). Select $c_3 = k_2^2$ to cancel the $\tilde{x}_2\tilde{x}_3$ cross-term and try yet another linear control law.

$$u = -k_3\tilde{x}_3, \quad k_3 > k_2 \tag{4.32}$$

is a natural candidate and with this we investigate the resulting clf time derivative.

$$\dot{V}_3 \leq -k_2^2c_2\phi^2(\xi) - k_2^3\tilde{x}_2^2 - (k_3 - k_2)\tilde{x}_3^2 + (k_1k_2 + k_2^2c_2)\tilde{x}_3\phi(\xi)$$

In order to investigate the impact of the last cross-term, we complete the squares.

$$\begin{aligned}\dot{V}_3 \leq &- k_2^3\tilde{x}_2^2 - (k_3 - k_2)(\tilde{x}_3 - \frac{k_1k_2 + k_2^2c_2}{2(k_3 - k_2)}\phi(\xi))^2 \\ &- (k_2^2c_2 - \frac{(k_1k_2 + k_2^2c_2)^2}{4(k_3 - k_2)})\phi^2(\xi)\end{aligned}$$

$\dot{V}_3$ is negative definite provided that the $\phi^2(\xi)$ coefficient is negative, which is true for

$$k_3 > k_2(1 + \frac{(k_1 + k_2c_2)^2}{4k_2c_2}) \tag{4.33}$$

We now pick c_2 to minimize this lower limit under the constraints $c_2 > 0$ and $k_2c_2 > k_1$.

For $k_1 \leq 0$, we can make $k_1 + k_2c_2$ arbitrarily small whereby Equation (4.33) reduces to

$$k_3 > k_2$$

i.e., the same restriction as in (4.32). For $k_1 > 0$ the optimal strategy can be shown to be selecting c_2 arbitrarily close to the bound k_1/k_2. This yields

$$k_3 > k_2(1 + k_1) \tag{4.34}$$

To summarize, under the parameter restrictions in (4.26), (4.28), (4.32), and (4.34), the control law (4.32) makes the origin GAS. □

4.2.2 Robustness

For this design we refrain from investigating inverse optimality due to the complicated expressions involved. Regardless of this, the control law (4.23) can be shown to provide some robustness. For example, in the case of $k_1 > 0$, we can introduce a multiplicative gain perturbation

$$\Gamma > \frac{k_2(1+k_1)}{k_3} \tag{4.35}$$

at the input (see Figure 3.2) without violating $\Gamma k_3 > k_2(1+k_1)$ from (4.24), hence retaining global asymptotic stability at the origin.

4.2.3 Feedback Linearization

It is rewarding to compare the preceding backstepping design with a control design based on feedback linearization. Such a design makes the open loop system a chain of integrators by defining new coordinates according to

$$\begin{aligned}
\dot{x}_1 &= \phi(x_2 - x_1) \triangleq \tilde{x}_2 \\
\dot{\tilde{x}}_2 &= \phi'(x_2 - x_1)(x_3 - \tilde{x}_2) \triangleq \tilde{x}_3 \\
\dot{\tilde{x}}_3 &= \phi''(x_2 - x_1)(x_3 - \tilde{x}_2)^2 + \phi'(x_2 - x_1)(u - \tilde{x}_3) \triangleq \tilde{u}
\end{aligned}$$

We can now select

$$\tilde{u} = -l_1 x_1 - l_2 \tilde{x}_2 - l_3 \tilde{x}_3$$

to achieve any desired linear relationship between $\tilde{u}$ and x_1. Solving for the actual control input u to be produced, we get

$$u = \tilde{x}_3 + \frac{\tilde{u} - \phi''(x_2 - x_1)(x_3 - \tilde{x}_2)^2}{\phi'(x_2 - x_1)} \tag{4.36}$$

Two things are worth noting about this expression.

- The feedback linearization control law depends not only on ϕ (through $\tilde{x}_2$), but also on its first and second derivatives, which therefore must be known.

- ϕ' is in the denominator of (4.36) implying that the control law has a singularity where $\phi' = 0$. Thus, global stability cannot be achieved using feedback linearization unless ϕ is a strictly increasing function.

Our backstepping design did not suffer from any of the problems above, since all we required from ϕ was for $x\phi(x)$ to be positive definite, see Assumption 4.3.

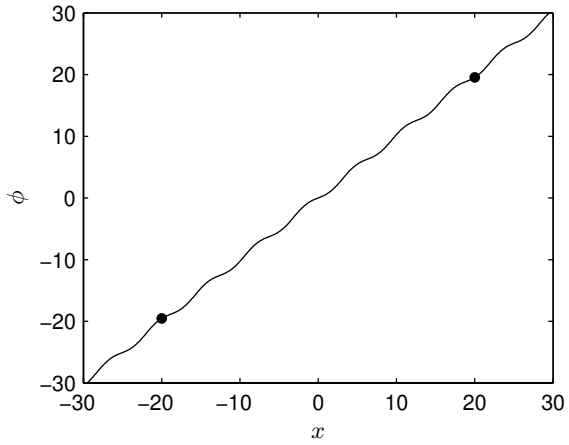


Figure 4.7: Illustration of the nonlinearity $\phi(x) = x - 0.5\sin x$. The dots represent the two initial conditions in the simulations.

4.2.4 Example

Consider the system

$$\begin{aligned}\dot{x}_1 &= x_2 - x_1 - 0.5\sin(x_2 - x_1)\\ \dot{x}_2 &= x_3\\ \dot{x}_3 &= u\end{aligned}$$

This system matches the generic system (4.21) with $\phi(x) = x - 0.5\sin x$ which satisfies Assumption 4.3, see Figure 4.7.

Theorem 4.3 gives the linear backstepping control law

$$u = -k_3(x_3 + k_2(x_2 + k_1 x_1))$$

The gains $k_1 = 0.5$, $k_2 = 0.5$, $k_3 = 1$ achieve GAS at the origin. The feedback linearization control law (4.36) can also be used since $\phi'(x) = 1 - 0.5\cos x > 0$ means that the system is globally feedback linearizable. We select the feedback gains $l_1 = 0.75$, $l_2 = 1.5$, $l_3 = 2$ to achieve a similar convergence in x_1 as for the backstepping control law. This choice gives the same closed loop poles as the backstepping control law would give for $\phi(x) = x$.

The simulation results are shown in Figure 4.8. The responses in x_1 for the two control laws are almost identical, but the further down the chain of integrators in the system one looks, the greater is the difference.

4.2.5 Set-Point Regulation

Let us now consider set-point regulation of system (4.21) with $y = x_1$ as the controlled variable. Here we allow for ϕ to cross zero at an arbitrary position, x_0.

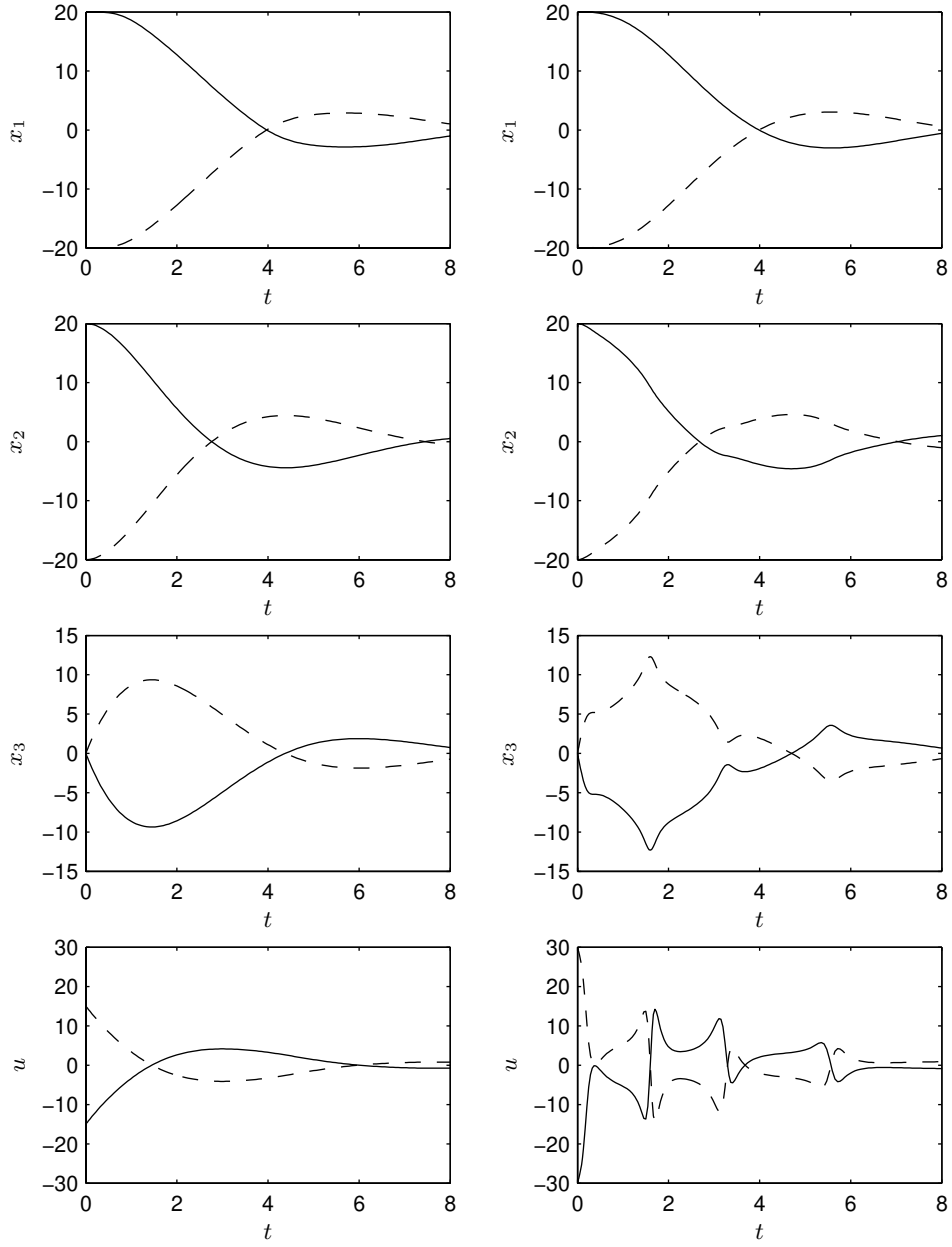


Figure 4.8: Simulation results using backstepping (left) and feedback linearization (right) for the initial conditions $x_1(0) = 20$ (solid) and $x_1(0) = -20$ (dashed). Although the responses in x_1 are very similar, the control signals differ a lot.

Assumption 4.4 *Let ϕ satisfy the sign condition*

$$(x - x_0)\phi(x - x_0) > 0, \quad x \neq x_0$$

Corollary 4.4 *Consider the system (4.21) and let Assumption 4.4 hold. Let r be a constant reference signal. Then the linear control law*

$$u = k_3k_2(1 + k_1)r - k_3(x_3 + k_2(x_2 + k_1x_1)) + k_3k_2x_0$$

where k_1, k_2, and k_3 satisfy (4.24), makes $y = r$ a GAS equilibrium.

Proof: At steady state, $\dot{x}_1 = \phi(x_2 - x_1) = 0$. Since $\phi(x_0) = 0$, $y = x_1 = r$ implies $x_2 = r + x_0$. Introduce the residual variables $z_1 = x_1 - r$, $z_2 = x_2 - r - x_0$, and the shifted function $\eta(x) = \phi(x + x_0)$, which satisfies Assumption 4.3. This gives the dynamics

$$\begin{aligned}\dot{z}_1 &= \eta(z_2 - z_1)\\ \dot{z}_2 &= x_3\\ \dot{x}_3 &= u\end{aligned}$$

Applying Theorem 4.3 now gives the control law

$$\begin{aligned}u &= -k_3(x_3 + k_2(z_2 + k_1z_1)) = -k_3(x_3 + k_2(x_2 - r - x_0 + k_1(x_1 - r)))\\ &= k_3k_2(1 + k_1)r - k_3(x_3 + k_2(x_2 + k_1x_1)) + k_3k_2x_0\end{aligned}$$

where k_1, k_2, and k_3 must satisfy (4.24). □

4.3 Applications to Flight Control

The two nonlinear systems studied in Section 4.1 and Section 4.2 are generalizations of two nonlinear control problems that appear in high angle of attack flight control. In this section, we apply the derived backstepping control laws to maneuvering flight control (angle of attack control, sideslip regulation, and roll control) and to flight path angle control.

4.3.1 Maneuvering Flight Control

First consider maneuvering flight control, where the aim is to control the aircraft variables involved in the fast flight modes—the angle of attack, α, the sideslip angle, β, and the components of the angular velocity vector ω_w, i.e., p_w, q_w, and r_w. We choose to express the angular velocity in the wind-axes coordinate frame in order to roll about the velocity vector rather than about the body x-axis, as discussed in Section 2.2. The control objective is to make

$$\begin{aligned}p_w &= p_w^{\text{ref}}\\ \alpha &= \alpha^{\text{ref}}\\ \beta &= 0\end{aligned}$$

a globally asymptotically stable equilibrium.

The dynamics of α and β are given by (2.16). We will assume the aerodynamic forces L and Y to depend only on α and β and not on ω_w or the control surface deflections, see Section 2.3. If we consider

$$u = \dot{\omega}_w$$

as the control input, the open loop dynamics are given by

$$\dot{p}_w = u_1 \tag{4.37a}$$
$$\dot{\alpha} = \frac{1}{\cos\beta}(q_w + \frac{1}{mV_T}(-L(\alpha,\beta) - F_T \sin\alpha + mg_3)) \tag{4.37b}$$
$$\dot{q}_w = u_2 \tag{4.37c}$$
$$\dot{\beta} = -r_w + \frac{1}{mV_T}(Y(\alpha,\beta) - F_T \cos\alpha \sin\beta + mg_2) \tag{4.37d}$$
$$\dot{r}_w = u_3 \tag{4.37e}$$

where

$$g_2 = g(\cos\alpha \sin\beta \sin\theta + \cos\beta \sin\phi \cos\theta - \sin\alpha \sin\beta \cos\phi \cos\theta)$$
$$g_3 = g(\sin\alpha \sin\theta + \cos\alpha \cos\phi \cos\theta)$$

are the gravitational contributions from (2.8). Given a control law in terms $u = \dot{\omega}_w$, we can determine which net torque T (in the body-fixed frame) that is required to produce this angular acceleration. Combining (2.2) with (2.6) and disregarding the dynamics of α and β in the transformation matrix T_{bw} yields

$$T = I\dot{\omega} + \omega \times I\omega \approx IT_{bw}u + \omega \times I\omega \tag{4.38}$$

Determining control surface deflections such that this torque is produced is known as control allocation, which is the topic of Part II.

The control design will be based on the backstepping designs in Section 4.1. These designs are not directly applicable since the generic system (4.3) is SISO whereas (4.37) is a MIMO system. However, (4.37) can be separated into three SISO systems if the couplings between the dynamics of α and β are neglected.

To achieve this separation, all variables except α and q_w will be regarded as constants in the angle of attack control design. Accordingly, in the sideslip control design, all variables except β and r_w will be regarded as constants. This includes slower varying variables such as the aircraft velocity, V_T, and the orientation of the aircraft, which affects the gravitational terms, g_2 and g_3. With this, the roll dynamics (4.37a), the angle of attack dynamics (4.37b)–(4.37c), and the sideslip dynamics (4.37d)–(4.37e), are each described by a SISO system.

The angle of attack and sideslip dynamics are both of second order and match those of the generic nonlinear system (4.3). Table 4.1 summarizes the relationships

General system (4.3)	α dynamics (4.37b)–(4.37c)	β dynamics (4.37d)–(4.37e)
x_1	α	β
x_2	$q_w/\cos\beta$	$-r_w$
ϕ	$f_\alpha/\cos\beta$	f_β
u	$u_2/\cos\beta$	$-u_3$
y	α	β
r	α^{ref}	0

Table 4.1: The relationships between the general nonlinear system (4.3) and the angle of attack and sideslip dynamics in (4.37).

between the systems. The nonlinearities are given by

$$f_\alpha(\alpha) = \frac{1}{mV_T}(-L(\alpha,\beta) - F_T\sin\alpha + mg_3) \tag{4.39}$$

$$f_\beta(\beta) = \frac{1}{mV_T}(Y(\alpha,\beta) - F_T\cos\alpha\sin\beta + mg_2) \tag{4.40}$$

where the dependence on variables that are regarded as constants are not included as arguments of f_α and f_β. The characteristics of these functions are decided mainly by the lift force, $L(\alpha,\beta)$, and the side force, $Y(\alpha,\beta)$. The corresponding aerodynamics coefficients, C_L and C_Y, are shown in Figure 4.9 for the ADMIRE model (ADMIRE ver. 3.4h 2003). Recall from (2.13) that $L = \bar{q}SC_L$ and $Y = \bar{q}SC_Y$.

We now propose state feedback control laws for each of the three control objectives. Each control law provides global asymptotic stability around the desired reference level provided that all other variables are held constant, as described above.

Velocity Vector Roll Control

Controlling the velocity vector roll rate is straightforward. Given the dynamics in (4.37a) and the roll rate command p_w^{ref}, the proportional control law

$$u_1 = k_p(p_w^{\text{ref}} - p_w) \tag{4.41}$$

makes $p_w = p_w^{\text{ref}}$ a GAS equilibrium if $k_p > 0$.

Angle of Attack Control

Combining Corollary 4.3 (set-point regulation) and Corollary 4.1 (linear control) yields the angle of attack control law

$$u_2 = -k_q(q_w + k_\alpha\cos\beta(\alpha - \alpha^{\text{ref}}) + f_\alpha(\alpha^{\text{ref}})) \tag{4.42}$$

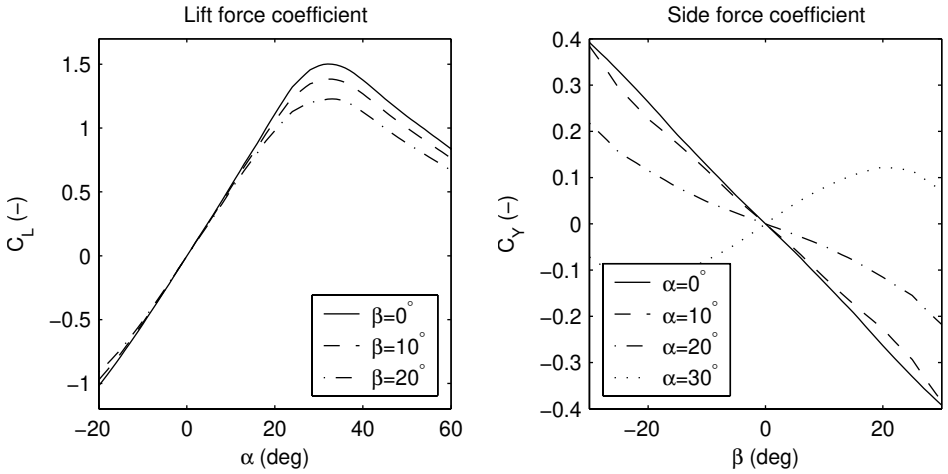


Figure 4.9: Lift force coefficient, C_L (left), and side force coefficient, C_Y (right), as functions of α and β for the ADMIRE model. At low angles of attack, both forces act stabilizing in the aircraft dynamics (4.37).

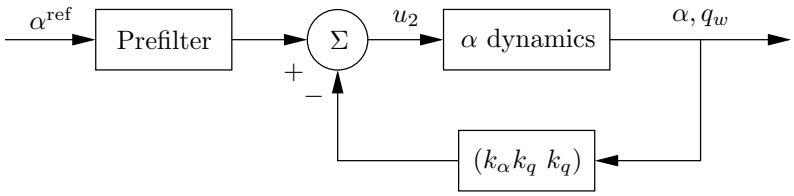


Figure 4.10: The backstepping control law (4.42) moves the dependence on C_L outside the control loop, thereby enhancing the robustness.

where f_α is given by (4.39). The control law is illustrated in Figure 4.10. Although implementing this control law requires knowledge of the lift force, and thereby the lift force coefficient, C_L, we note that the lift force dependent computation is performed in the prefilter *outside* the feedback loop. Therefore, imperfect knowledge of C_L does not jeopardize closed loop stability but only shifts the equilibrium.

The feedback gains k_α and k_q should satisfy

$$\begin{aligned} k_\alpha &> \max\{0, \kappa_\alpha\} \\ k_q &> 2 \cdot k_\alpha \end{aligned} \tag{4.43}$$

for the control law to be globally stabilizing and also minimize a meaningful cost functional. Setting $\beta = 0$, the growth coefficient κ_α can be approximately com-

puted as

$$\kappa_\alpha = \max_\alpha f'_\alpha(\alpha) \approx \max_\alpha -\frac{1}{mV_T}\frac{\partial L(\alpha,\beta)}{\partial \alpha} = -\frac{\bar{q}S}{mV_T}\min_\alpha \frac{\partial C_L(\alpha,\beta)}{\partial \alpha} \tag{4.44}$$

At low angles of attack, $\frac{\partial C_L}{\partial \alpha}$ is positive and at high angles of attack, in the post-stall region, $\frac{\partial C_L}{\partial \alpha}$ is negative, see Figure 4.9. This means that if the flight envelope does not include high angles of attack then $\kappa_\alpha < 0$, and $k_\alpha > 0$ suffices to guarantee stability. On the other hand, if the control law is to provide stability also in the post-stall region, then $k_\alpha > \kappa_\alpha > 0$ must hold.

Note that κ_α also depends on the speed and altitude of the aircraft due to the factor $\bar{q}/V_T$. To handle this, one can either select k_α and k_q such that (4.43) holds for all flight cases, or schedule k_α and k_q with speed and altitude.

An alternative control law can be found by applying Corollary 4.2, where a linearizing virtual control law was selected. This gives us

$$u_2 = -k_q(q_w + k_\alpha \cos\beta(\alpha - \alpha^{\text{ref}}) + f_\alpha(\alpha)) \tag{4.45}$$

The only difference to (4.42) is that now f_α depends on α rather than on α^{ref}. This causes the feedback loop to depend on C_L, and robustness against model errors in C_L becomes more difficult to analyze.

The control law (4.45) matches the angle of attack control law proposed by Snell et al. (1992), which was derived using feedback linearization and time-scale separation arguments. Using our Lyapunov based backstepping approach, we have thus shown this control law to be not only globally stabilizing, but also inverse optimal w.r.t. a meaningful cost functional, according to Corollary 4.2.

Sideslip Regulation

Finally, we consider sideslip regulation. Since $\beta = 0$ is the control objective, Theorem 4.1 (stabilization around the origin) can be used if we adjust for that

$$f_\beta(0) = \frac{1}{V_T} g \cos\theta \sin\phi$$

may not be zero. This gives the sideslip control law

$$u_3 = k_r(-r_w + k_\beta \beta + \frac{1}{V_T} g \cos\theta \sin\phi) \tag{4.46}$$

The feedback gain restrictions

$$\begin{aligned} k_\beta &> \max\{0, \kappa_\beta\} \\ k_r &> 2 \cdot k_\beta \end{aligned}$$

ensure the control law to be globally stabilizing and also optimal w.r.t. a meaningful cost functional. Here,

$$\kappa_\beta = \max_\beta \frac{f_\beta(\beta) - f_\beta(0)}{\beta} \approx \max_\beta \frac{Y(\alpha,\beta)}{\beta} = \frac{\bar{q}S}{mV_T}\max_\beta \frac{C_Y(\alpha,\beta)}{\beta}$$

At low angles of attack, $\kappa_\beta < 0$ holds according to Figure 4.9 and the parameter restrictions above reduce to

$$k_r > 2 \cdot k_\beta > 0$$

In this case no further knowledge of the side force is necessary to implement the control law (4.46). At high angles of attack, $\kappa_\beta > 0$, which means $k_\beta > \kappa_\beta$ must hold for yaw stability to be guaranteed.

4.3.2 Flight Path Angle Control

Let us now consider control of the flight path angle, γ, which for example may be of interest for controlling the ascent or descent of an unmanned aerial vehicle. Restricting ourselves to longitudinal control, the relevant dynamics are given by the last three equations of (2.17). If we again assume the lift force to depend only on the angle of attack, $\alpha = \theta - \gamma$, this system matches the third order system (4.21) considered in Section 4.2, except for the last term in $\dot{\gamma}$, $mg\cos\gamma$. To circumvent this problem, we make the approximation $\cos\gamma \approx \cos\gamma^{\text{ref}}$ which gives the dynamics

$$\begin{aligned} \dot{\gamma} &= \frac{1}{mV_T}\big(L(\alpha) + F_T \sin\alpha - mg\cos\gamma^{\text{ref}}\big) \\ \dot{\theta} &= q \\ \dot{q} &= \frac{1}{I_y}(M + F_T z_{TP}) \end{aligned} \tag{4.47}$$

This system matches (4.21) with

$$\begin{aligned} x_1 &= \gamma \\ x_2 &= \theta \\ x_3 &= q \\ u &= \frac{1}{I_y}(M + F_T z_{TP}) \\ \phi(x_2 - x_1) &= \frac{1}{mV_T}\big(L(\alpha) + F_T \sin\alpha - mg\cos\gamma^{\text{ref}}\big) \end{aligned}$$

From Figure 4.9 we see that Assumption 4.4 holds even at high angles of attack, since ϕ is dominated by the lift force term. The assumption breaks down around $\alpha = 90°$.

Applying Corollary 4.4 yields the flight path angle control law

$$u = -k_3(q + k_2(\theta + k_1(\gamma - \gamma^{\text{ref}}) - \gamma^{\text{ref}} - \alpha_0))$$

where α_0 is the angle of attack at steady state, solving $\dot{\gamma} = \phi(\alpha_0) = 0$, and where the feedback gains must satisfy (4.24). Alternatively, α can be used for feedback rather than θ. Using $\theta = \alpha + \gamma$ gives the equivalent control law

$$u = -k_3(q + k_2(\alpha - \alpha_0 + (1 + k_1)(\gamma - \gamma^{\text{ref}}))) \tag{4.48}$$

The aerodynamic pitching moment required to produce this angular acceleration is given by

$$M = I_y u - F_T z_{TP} \tag{4.49}$$

This backstepping control law guarantees stability up to and beyond the stall angle, where the lift force starts to decrease. In contrast, the feedback linearization control law (4.36) can only be used for low angles of attack, since (4.36) is singular when $\phi' = 0$, which occurs around the stall angle.

4.3.3 Simulation Results

In this section we use the ADMIRE model, described in Section 2.4, to illustrate the behavior of the proposed angle of attack and flight path angle control laws at high angles of attack. The considered flight case is Mach 0.3, 1000m. In addition to the backstepping control laws, the control system contains a control allocator, distributing the desired aerodynamic moment to the control surfaces. For γ control, integral control is also included to achieve set-point regulation despite model errors. For example, a fixed value of α_0, selected as the angle of attack at trimmed flight, is used in the control law. For α control, no integral action is included. In Chapter 11, the overall control system is presented in detail and other properties are evaluated.

Angle of Attack Control

Figure 4.11 shows the simulation results when the control law (4.42) is used with $k_q = 5.7$ and $k_\alpha = 1.3$. For $\alpha^{\text{ref}} = 20$ deg and $\alpha^{\text{ref}} = 30$ deg, the aircraft response is satisfactory. For $\alpha^{\text{ref}} = 40$ deg, which is in the post-stall region, the control system fails to achieve the control objective. The is due to that the control allocator fails to produce the aerodynamic pitching moment required to stabilize the aircraft. As can be seen from the control surface plots, the canards and the elevons all saturate temporarily, either in position or in rate, when $\alpha^{\text{ref}} = 40$ is applied. This means that no fair evaluation of the control law properties at high angles of attack can be made. A possible remedy for this would be to also use the thrust vectoring capabilities included in the model.

Flight Path Angle Control

Figure 4.12 shows the resulting flight trajectory when the control law (4.48) is used with $k_1 = 1.0$, $k_2 = 1.3$, and $k_3 = 5.7$, and $\gamma^{\text{ref}} = 30$ deg is commanded. Although the angle of attack reaches a maximum of 36 degrees, and the canards and the elevons saturate in rate and in position, respectively, stability is retained.

4.4 Conclusions

In this chapter, backstepping has been used to design linear, robust control laws for two generic nonlinear systems. With a linear control law, the closed loop system remains nonlinear. This is the converse of feedback linearization, in which

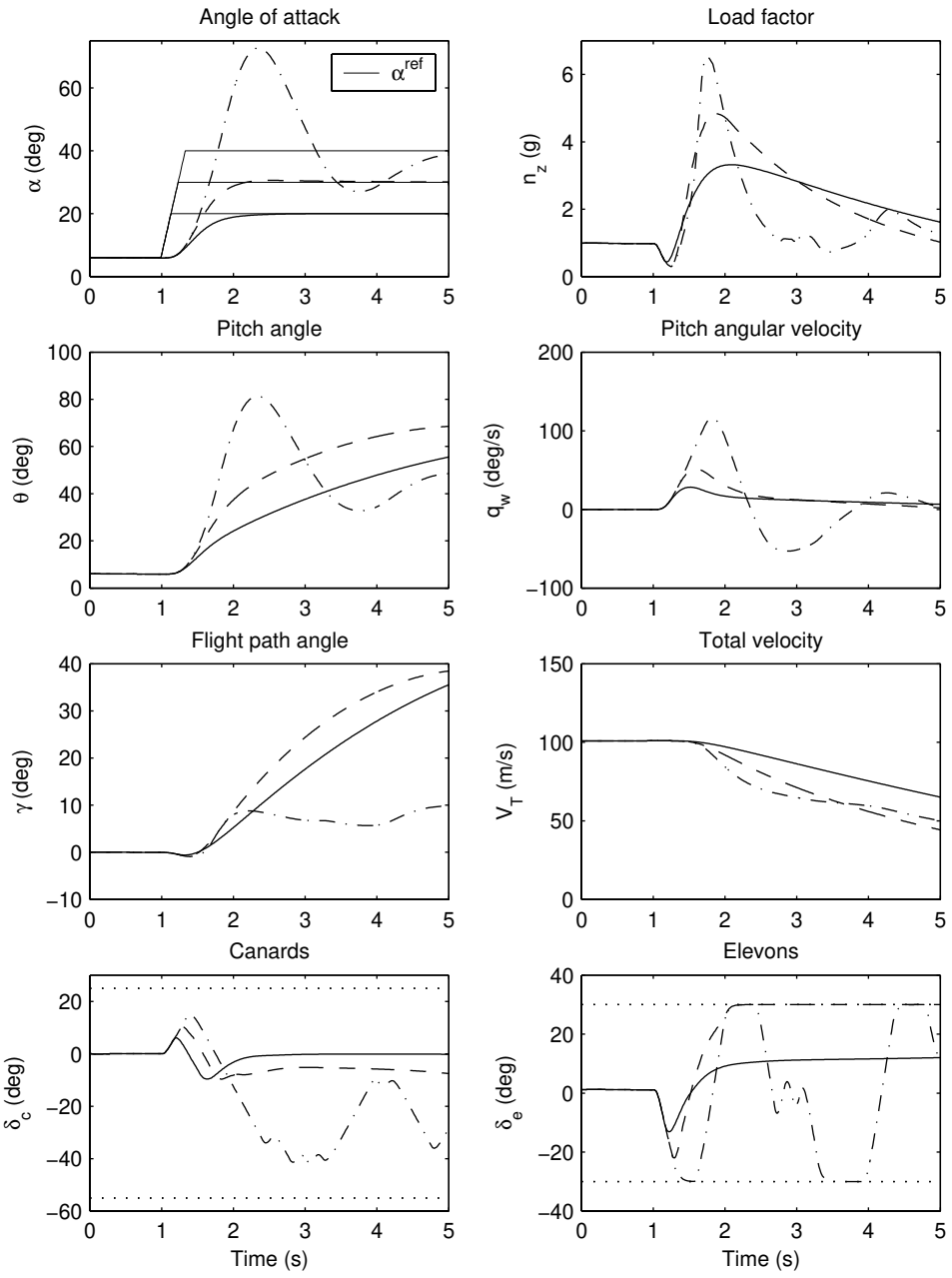


Figure 4.11: Simulation results for three different angle of attack commands: $\alpha^{\text{ref}} = 20$ deg (solid), $\alpha^{\text{ref}} = 30$ deg (dashed), and $\alpha^{\text{ref}} = 40$ deg (dash-dotted).

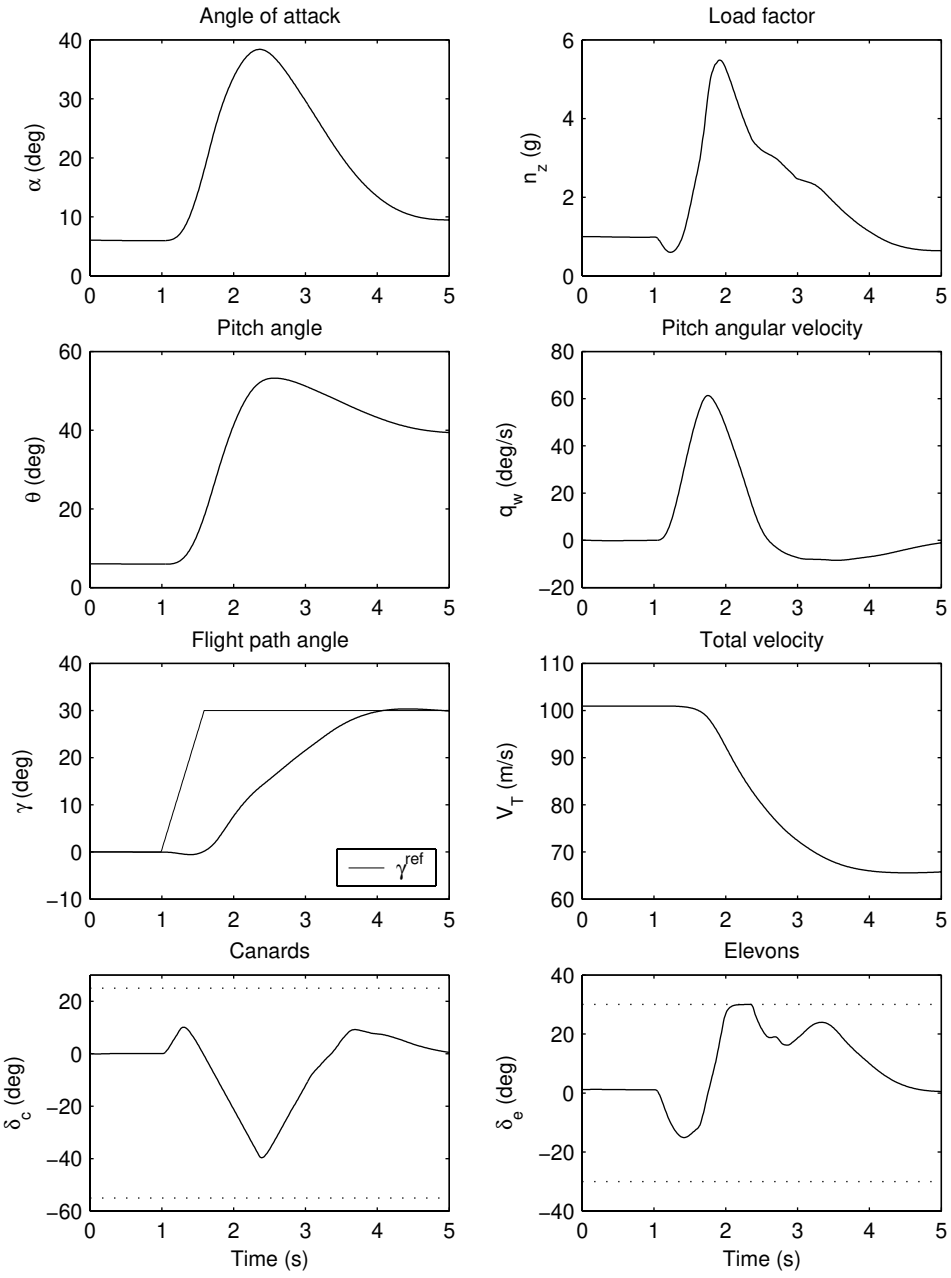


Figure 4.12: Simulation results when $\gamma^{\text{ref}} = 30$ deg is commanded.

the closed loop system is rendered linear using nonlinear feedback. A benefit of not forcing a linear output response is that less control effort may be needed, as illustrated in the examples.

We have also shown the connection between our backstepping designs and control design based on time-scale separation. Both render cascaded controllers, but with backstepping, actual bounds on the feedback gains in the different loops that guarantee global stability are achieved.

Unfortunately, it is not clear how these backstepping designs can be extended to other nonlinear systems. For example, it is an open question whether the second order system (4.3) can be augmented with an integrator at the input and still be controlled by a linear control law. This could be of interest for including actuator dynamics in the flight control case.

The application of backstepping to flight control is relevant since the nonlinearities due to the aerodynamic forces cannot be modeled precisely, and since these forces typically act stabilizing and therefore do not need to be cancelled. The proposed control law for maneuvering flight control and for flight path angle control give stability even for high angles of attack.

Chapter 5

Backstepping Control of a Rigid Body

Controlling the motion of a rigid body is important in many applications. For example aircraft, missiles, cars, ships, and submarines can often be regarded as rigid bodies, at least for the purpose of control. There are several ways of designing controllers for rigid bodies, see, e.g., Woolsey and Leonard (1999) and Woolsey et al. (2001). In this chapter we develop a method for backstepping control of rigid body motion. The design can then be specialized to aircraft control problems or the control of other types of vehicles.

Starting from the standard equations for rigid body motion (2.6),

$$\begin{aligned} \dot{V} &= -\omega \times V + \frac{1}{m} F \\ I\dot{\omega} &= -\omega \times I\omega + T \end{aligned}$$

we will derive a control law to make $V = V_o$, $\omega = \omega_o$ a globally asymptotically stable equilibrium. Controlling V, the components of the velocity vector expressed in body-fixed coordinates, can be interpreted as controlling the total velocity, V_T, the angle of attack, α, and the sideslip angle, β, since they are related through

$$V = V_T \begin{pmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \end{pmatrix}^T$$

We will retain the model on vector form for the control design. This idea can also be found in Fossen and Grøvlen (1998) who consider rigid body motion in two dimensions to derive control laws for ship control. In contrast, many previous designs, like Lane and Stengel (1988) and Grøvlen and Fossen (1996), treating flight control and ship control respectively, use a component form description of the model. The same goes for the backstepping designs in the previous chapter.

The vector form is convenient to work with and leads to reduced complexity of the expressions involved in the control design. Also, treating the problem as a multivariable control problem allows us to account for the cross-term $\omega \times V$

correctly, which was not done in the previous chapter. A disadvantage is that it less clear how to utilize the characteristics of F and T to achieve a robust controller.

This chapter is organized as follows. In Section 5.1, the rigid body dynamics are presented and the assumptions used in the control design are stated. In Section 5.2, possible steady states of the considered rigid body motion are investigated. The backstepping control design is performed in Section 5.3 and illustrated with an example in Section 5.4. Section 5.5 contains the conclusions of the chapter.

The chapter is based on

> S. T. Glad and O. Härkegård. Backstepping control of a rigid body. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 3944–3945, Las Vegas, NV, Dec. 2002.

5.1 Rigid Body Dynamics

We assume that the controlled object is a rigid body with mass m. We describe the motion in a body-fixed coordinate system with the origin at the center of mass and obtain the equations

$$\dot{V} = -\omega \times V + \frac{1}{m}F \tag{5.1a}$$

$$I\dot{\omega} = -\omega \times I\omega + T \tag{5.1b}$$

where V is the velocity, ω is the angular velocity, F is the external force, and T is the external torque (all these quantities are vectors with three components). I is the moment of inertia. We will assume that the force has the form

$$F = m(F_a(V) + u_V \hat{V}) \tag{5.2}$$

where $\hat{V} = \frac{1}{|V|}V$ and u_V is a control variable. The first part, F_a, corresponds to aerodynamic or hydrodynamic forces, and the second part approximately models the thrust action of an engine aligned with the velocity vector. The torque T is assumed to depend on V, ω, and control variables.

5.2 Stationary Motion

Let us first examine the possible steady states of the rigid body dynamics (5.1). The following relationships are useful for rearranging cross product expressions:

$$\begin{aligned} a \times b &= -b \times a & \qquad a \times a &= 0 \\ (a \times b) \times c &= (a^T c)b - (b^T c)a & \qquad a^T(a \times b) &= 0 \\ a \times (b \times c) &= (a^T c)b - (a^t b)c & \qquad a^T(b \times c) &= b^T(c \times a) \end{aligned}$$

Consider a motion with $V = V_o$, $\omega = \omega_o$ where V_o and ω_o are constant vectors. The velocity equation (5.1a) then becomes

$$\omega_o \times V_o = F_a(V_o) + u_V \hat{V}_o$$

Multiplying with $\hat{V}_o^T$ shows that u_V has to satisfy

$$u_V = -\hat{V}_o^T F_a(V_o) \tag{5.3}$$

To determine possible values of ω_o given V_o we rewrite ω_o as

$$\omega_o = \omega_\perp + \lambda \hat{V}_o$$

where $\omega_\perp$ and $\hat{V}_o$ are orthogonal and λ is a constant scalar. Inserting (5.3) this gives us

$$\omega_o \times V_o = \omega_\perp \times V_o = F_a(V_o) - [\hat{V}_o^T F_a(V_o)]\hat{V}_o$$

Since only $\omega_\perp$ is left, this shows that the angular velocity component parallel to the velocity vector, $\lambda \hat{V}_o$, can be selected arbitrarily. This corresponds to the rolling motion about the velocity vector.

To solve for $\omega_\perp$, we take the cross product with $\hat{V}_o$,

$$\hat{V}_o \times (\omega_\perp \times V_o) = \big|V_o\big|\omega_\perp = \hat{V}_o \times (F_a(V_o) - [\hat{V}_o^T F_a(V_o)]\hat{V}_o) = \hat{V}_o \times F_a(V_o)$$

which yields

$$\omega_\perp = \frac{1}{\big|V_o\big|}\hat{V}_o \times F_a(V_o)$$

To summarize, $V = V_o$, $\omega = \omega_o$ is a possible steady state if ω_o satisfies

$$\omega_o = \frac{1}{\big|V_o\big|}\hat{V}_o \times F_a(V_o) + \lambda \hat{V}_o$$

where λ is an arbitrary constant. Thus, the angular velocity at steady state is determined by the velocity except for the component parallel to the velocity, corresponding to the velocity vector roll rate.

5.3 Backstepping Design

In this section we develop a backstepping design to make V_o, ω_o a stable equilibrium. Introduce the angular acceleration

$$u_T = I^{-1}(T - \omega \times I\omega)$$

and regard u_T as the control signal. Then the dynamics are given by

$$\begin{aligned} \dot{V} &= -\omega \times V + F_a(V) + u_V \hat{V} \\ \dot{\omega} &= u_T \end{aligned}$$

Step 1: First regard the angular velocity ω (together with u_V) as the control variable. Introduce the Lyapunov candidate

$$W_1 = \frac{1}{2}(V - V_o)^T(V - V_o)$$

and the virtual control law $\omega = \omega^{\text{des}}$. This gives

$$\dot{W}_1 = (V - V_o)^T(-\omega^{\text{des}} \times V + F_a(V) + u_V\hat{V})$$

Since the characteristics of $F_a(V)$ are unknown, we cancel its influence by selecting control laws of the form

$$u_V = \bar{u}_V - \hat{V}^T F_a(V)$$
$$\omega^{\text{des}} = \bar{\omega} + \frac{1}{|V|}\hat{V} \times F_a(V)$$

This gives

$$\begin{aligned}\dot{W}_1 &= (V - V_o)^T\Big(-\bar{\omega} \times V + \bar{u}_V\hat{V} + F_a(V) - [\hat{V}^T F_a(V)]\hat{V} - \frac{1}{|V|}[\hat{V} \times F_a(V)] \times V\Big) \\ &= (V - V_o)^T(-\bar{\omega} \times V + \bar{u}_V\hat{V})\end{aligned}$$

since

$$\frac{1}{|V|}[\hat{V} \times F_a(V)] \times V = [\hat{V} \times F_a(V)] \times \hat{V} = F_a(V) - [\hat{V}^T F_a(V)]\hat{V}$$

We now select $\bar{\omega}$ and $\bar{u}_V$ to make $\dot{W}_1$ negative. $\dot{W}_1$ can be rewritten as

$$\begin{aligned}\dot{W}_1 &= (V - V_o)^T(-\bar{\omega} \times V + \bar{u}_V\hat{V}) \\ &= V_o^T(\bar{\omega} \times V) + \bar{u}_V(V - V_o)^T\hat{V} \\ &= \bar{\omega}^T(V \times V_o) + \bar{u}_V(V - V_o)^T\hat{V}\end{aligned}$$

The choices

$$\bar{u}_V = k_1(V_o - V)^T\hat{V}$$
$$\bar{\omega} = k_2(V_o \times V) + \lambda\hat{V}_o$$

where the term $\lambda\hat{V}_o$ reflects the velocity vector roll control objective, achieve

$$\dot{W}_1 = -k_1((V_o - V)^T\hat{V})^2 - k_2\big|V_o \times V\big|^2 \triangleq -U(V, V_o) \leq 0$$

In this expression $\dot{W}_1 = 0$ only if $V = V_o$, provided the singularity $V = 0$ is avoided. The control laws

$$u_V = k_1(V_o - V)^T\hat{V} - \hat{V}^T F_a(V)$$
$$\omega^{\text{des}} = k_2(V_o \times V) + \frac{1}{|V|}\hat{V} \times F_a(V) + \lambda\hat{V}_o$$

along with the Lyapunov function W_1 thus guarantee convergence to the desired $V = V_o$ if $\omega = \omega^{\text{des}}$ can be achieved.

Step 2: Define

$$\xi = \omega - \omega^{\text{des}} \iff \omega = \omega^{\text{des}} + \xi$$

In the new variables the dynamics are

$$\dot{V} = \underbrace{-\omega^{\text{des}} \times V + F_a(V) + u_V \hat{V}}_{\text{desired dynamics}} - \xi \times V$$
$$\dot{\xi} = u_T - \phi(V, \xi)$$

where

$$\phi(V, \xi) = \frac{d}{dt}\omega^{\text{des}}$$

Extending the Lyapunov function to

$$W_2 = k_3 W_1 + \frac{1}{2}\xi^T \xi = \frac{k_3}{2}(V - V_o)^T (V - V_o) + \frac{1}{2}\xi^T \xi \tag{5.4}$$

gives

$$\begin{aligned}
\dot{W}_2 &= k_3 (V - V_o)^T (-\omega^{\text{des}} \times V + F_a(V) + u_V \hat{V} - \xi \times V) + \xi^T (u_T - \phi) \\
&= -k_3 U(V, V_o) + k_3 \underbrace{V_o^T (\xi \times V)}_{\xi^T (V \times V_0)} + \xi^T (u_T - \phi) \\
&= -k_3 U(V, V_o) + \xi^T (u_T + k_3 (V \times V_o) - \phi) \\
&= -k_3 U(V, V_o) - k_4 \xi^T \xi \leq 0
\end{aligned}$$

if we select the control law

$$u_T = k_3 (V_o \times V) - k_4 \xi + \phi$$

Since $\dot{W}_2 = 0$ only occurs for $V = V_o$, $\xi = 0$ (except for the singular case $V = 0$) there will be convergence to $V = V_o$, $\xi = 0$, which implies $\omega = \omega_o$.

The following theorem summarizes our results.

Theorem 5.1 *Consider the rigid body dynamics (5.1). Let the force F have the form*

$$F = m(F_a(V) + u_V \hat{V})$$

and introduce the angular acceleration

$$u_T = I^{-1}(T - \omega \times I\omega)$$

Then the control laws

$$u_V = k_1(V_o - V)^T\hat{V} - \hat{V}^T F_a(V)$$
$$u_T = k_3(V_o \times V) + k_4(\omega^{\text{des}} - \omega) + \phi(V, \omega)$$

where

$$\omega^{\text{des}} = k_2(V_o \times V) + \frac{1}{|V|}\hat{V} \times F_a(V) + \lambda\hat{V}_o$$
$$\phi(V, \omega) = \frac{d}{dt}\omega^{\text{des}}$$

and $k_1, k_2, k_3, k_4 > 0$*, make*

$$V = V_o$$
$$\omega = \frac{1}{|V_o|}\hat{V}_o \times F_a(V_o) + \lambda\hat{V}_o$$

a GAS equilibrium except for the singular point $V = 0$*.*

Let us make a few comments regarding the backstepping design and the resulting control laws.

- *Design parameters.* The scalar design parameters k_i can be replaced by positive definite matrices K_i. This requires the expressions to be modified somewhat. For example, the total Lyapunov function in (5.4) can be replaced by
$$V = \frac{1}{2}(V - V_o)^T K_3(V - V_o) + \frac{1}{2}\xi^T\xi$$

- *Roll axis.* In the virtual control law ω^{des}, the term $\lambda\hat{V}_o$ implies that we wish to roll about the steady state velocity vector. A perhaps more natural choice would be to use $\lambda\hat{V}$ instead, to roll about the current velocity vector. The disadvantage of this choice is that the expression for $\phi = d\omega^{\text{des}}/dt$ becomes more complicated.

- *Control law interpretation.* The different terms in the control laws can be interpreted as follows.

$$\begin{aligned} u_V = {} & k_1(V_o - V)^T\hat{V} && \text{control of } |V| \\ & - \hat{V}^T F_a(V) && \text{cancel (parts of) } F_a \end{aligned}$$

$$\begin{aligned} u_T = {} & k_3(V_o \times V) && \text{control of } V\text{'s direction} \\ & + k_4(\omega^{\text{des}} - \omega) && \text{control of } \omega \\ & + \phi(V, \omega) && \text{compensate for non-constant } \omega^{\text{des}}(t) \end{aligned}$$

5.4 Example

In this section we illustrate the properties of the backstepping control. For simplicity we let $F_a = 0$, that is we consider a rigid body traveling in free space. This allows us to focus on the effects of the cross term $\omega \times V$ in the dynamics given by

$$\begin{aligned}\dot{V} &= -\omega \times V + u_V \hat{V} \\ \dot{\omega} &= u_T\end{aligned}$$

To make a connection to flight control, let the controlled variables be the total velocity, $V_T = |V|$, the angle of attack, α, the sideslip angle, β, and the roll rate about the velocity vector, p_w. Given the reference values of these variables, the desired equilibrium is given by

$$\begin{aligned}V_o &= V_T^{\text{ref}} \begin{pmatrix} \cos\alpha^{\text{ref}} \cos\beta^{\text{ref}} \\ \sin\beta^{\text{ref}} \\ \sin\alpha^{\text{ref}} \cos\beta^{\text{ref}} \end{pmatrix} \\ \omega_o &= p_w^{\text{ref}} \hat{V}_o\end{aligned}$$

From Theorem 5.1 we get the control law

$$\begin{aligned}u_V &= k_1 (V_o - V)^T \hat{V} \\ u_T &= k_3 (V_o \times V) + k_4 (\omega^{\text{des}} - \omega) + \phi(V, \omega)\end{aligned}$$

where

$$\begin{aligned}\omega^{\text{des}} &= k_2 (V_o \times V) + \lambda \hat{V}_o \\ \phi(V, \omega) &= \frac{d}{dt} \omega^{\text{des}} = k_2 V_o \times \left(-\omega \times V + (k_1 (V_o - V)^T \hat{V}) \hat{V} \right)\end{aligned}$$

The design parameters are selected as $k_1 = k_4 = 1$, $k_2 = k_3 = 0.2$.

Figure 5.1 illustrates the simulated maneuver. An increase in the total velocity is followed by a 30 degrees angle of attack command. Note that since $F_a = 0$ there is no external force affecting the object. This means that the object will continue to travel in the same direction as before but with a different orientation. Finally, a velocity vector roll is commanded. If the axis of rotation coincides exactly with the velocity vector, α and β will not be affected. The sideslip reference is zero throughout the maneuver.

The simulation results are shown in Figures 5.3 and 5.4. In Figure 5.3, the commands are applied sequentially as described above and in Figure 5.4, the commands are applied simultaneously. In both cases, the system converges to the specified states as expected. Further, the controlled variables are fairly decoupled in both cases. This can be regarded as a bonus since only stability was addressed in the backstepping design. Some more specific comments can also be made.

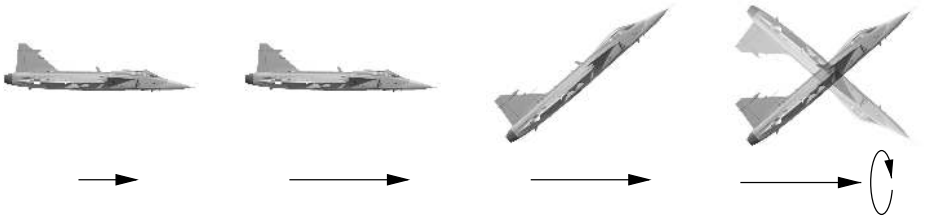


Figure 5.1: Maneuver used in the simulations. With $F_a = 0$, a change in α does not change the direction of motion but only the orientation of the object.

- In Figure 5.3, the total velocity decreases for a short period of time when the angle of attack command is applied. This is due to that $V(t)$ does not follow a circle segment with constant radius from $(4 \;\; 0 \;\; 0)^T$ to $(2\sqrt{3} \;\; 0 \;\; 2)^T$, see Figure 5.2. To achieve better decoupling between α and V_T, the control law for u_V could be adjusted so that $u_V = 0$ when $\big|V\big| = \big|V_o\big|$.
- In Figure 5.3, the velocity vector roll is perfectly performed without affecting α or β. In Figure 5.4, the result is somewhat degraded with a worst case sideslip of -5 degrees. The reason for this may be that the axis of rotation specified in ω^{des} is the desired velocity vector, $\hat{V}_o$, rather than the current one, $\hat{V}$, and that these two differ just after the commands are applied at $t = 2$.

5.5 Conclusions

We have proposed a backstepping control law that steers the velocity and angular velocity vectors of a rigid body to desired values. The control law uses external torques and a force along the velocity vector. This configuration is similar to, but not precisely equal to the one used in aircraft control, where control surfaces generate torques and the engine gives a longitudinal force. However, our proposed rigid body control could inspire new aircraft control designs. An interesting extension would be to take the orientation into account, which would make it possible to include the effect of gravity.

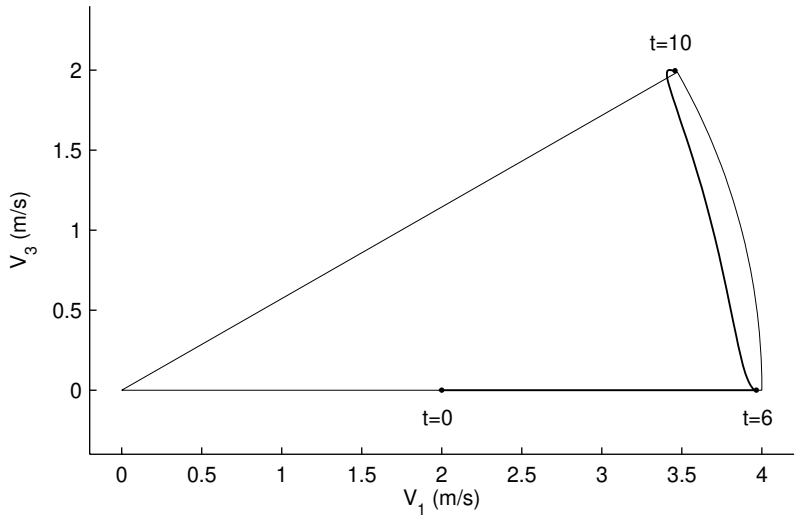


Figure 5.2: Trajectory generated by the first and third components of V when the commands are applied sequentially as in Figure 5.3. From $t = 6$ to $t = 10$, V does not follow the circle segment corresponding to a constant value of $V_T = \left|V\right|$.

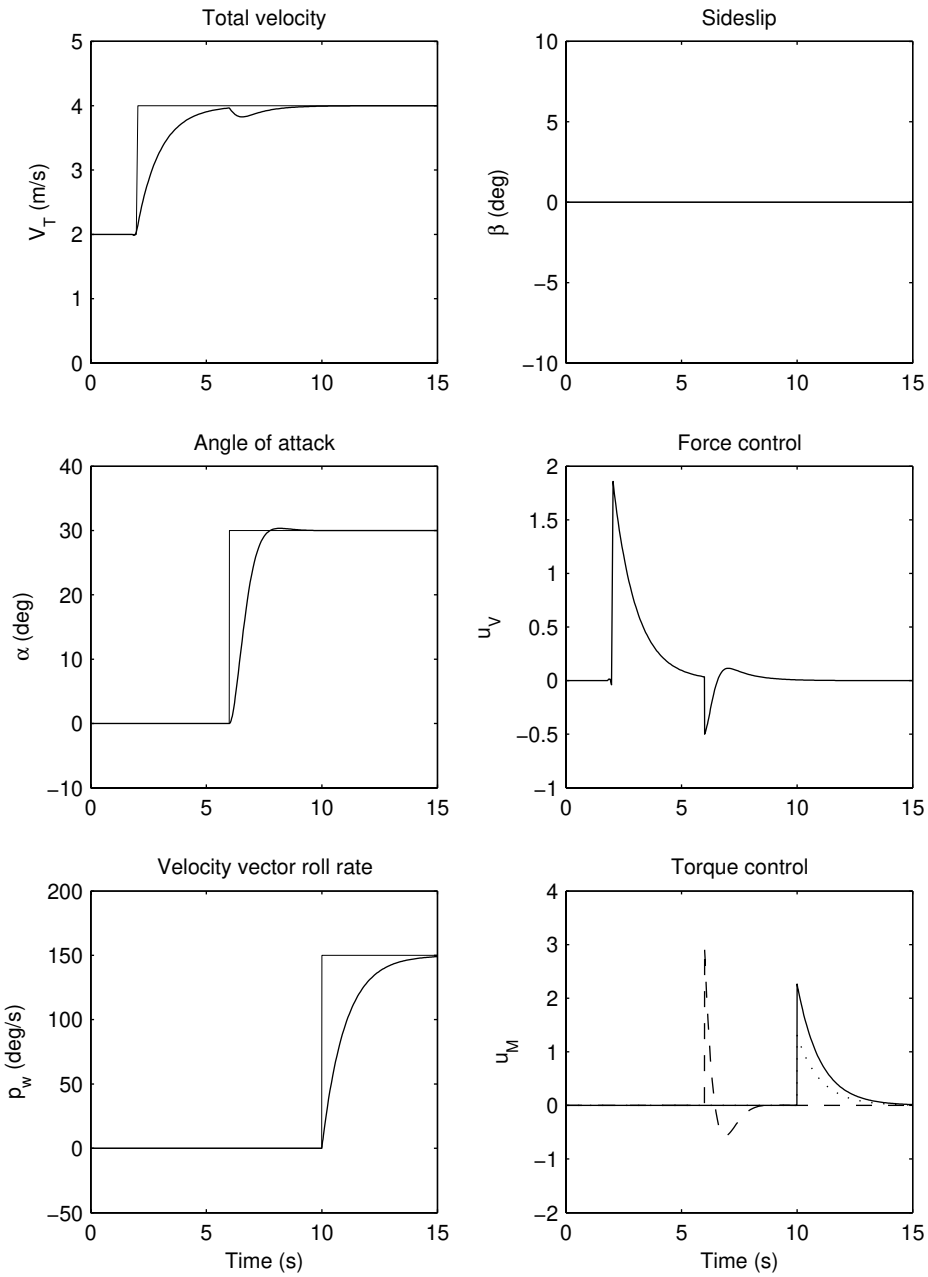


Figure 5.3: Simulation results when the reference commands (thin lines) in V_T, α, and p_w are applied sequentially. Note that u_V is used for control during the α step response, causing a dip in the V_T response.

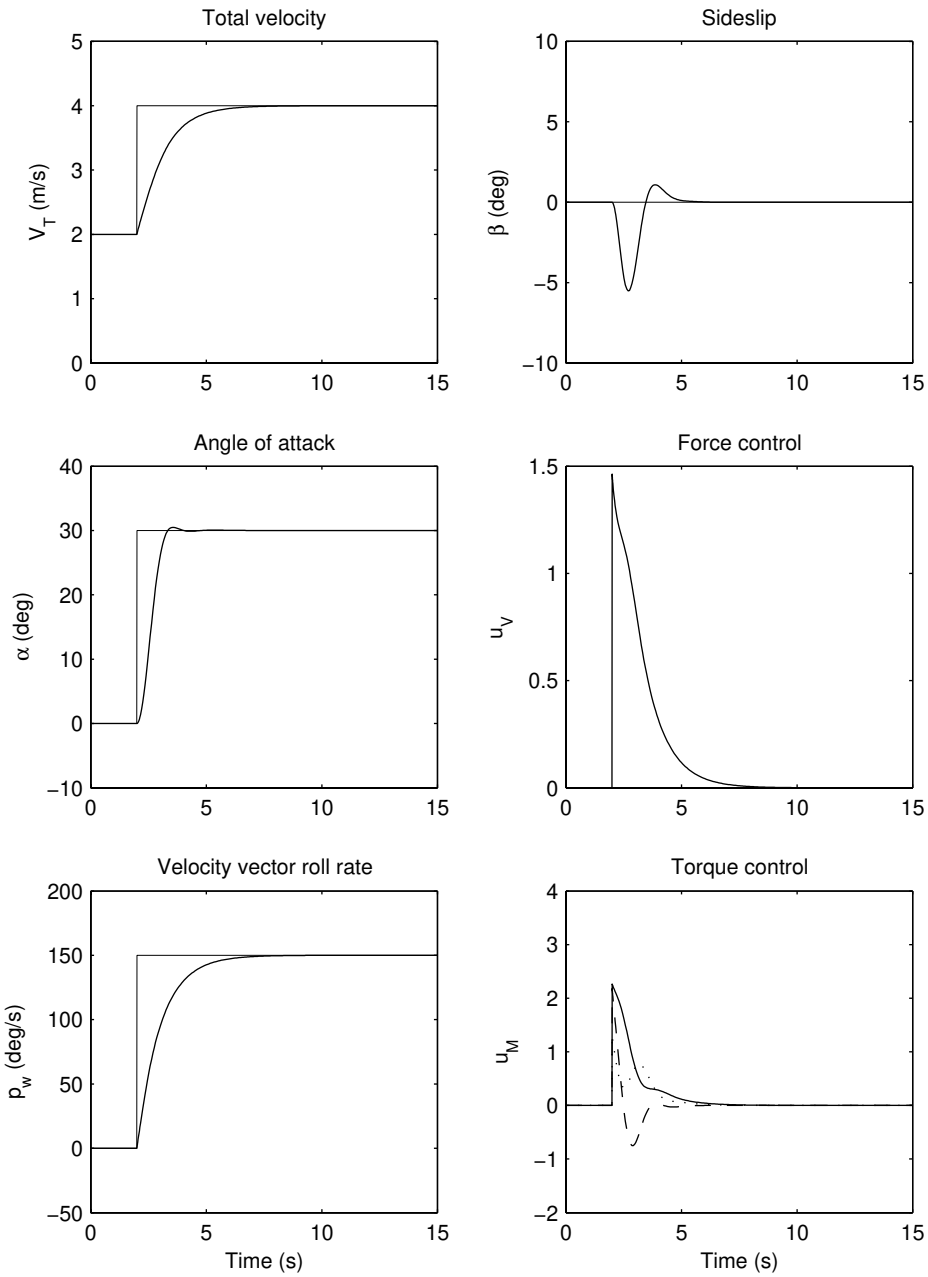


Figure 5.4: Simulation results when the reference commands (thin lines) in V_T, α, and p_w are applied simultaneously. The oscillations in β indicate that the axis of rotation in roll does not quite coincide with the velocity vector.

Chapter 6

Output Regulation with Constant Input Disturbances

For a controller to function in practice, it needs to be robust against uncertainties such as modeling errors and external disturbances. A special case is when the uncertainty and the control input are "matched" in the sense that they enter the dynamics in the same way. The system

$$\dot{x} = f(x) + B(u + \delta(x, u)) \tag{6.1}$$

satisfies this matching condition. In this chapter we consider the case where a controller has been designed for the nominal case $\delta = 0$, and we want to robustify this controller. In flight control the uncertainty δ corresponds to a torque disturbance acting on the aircraft, due to, e.g., unmodeled aerodynamic effects.

Two related methods for such robustification are Lyapunov redesign (Khalil 2002) and nonlinear damping (Krstić et al. 1995). In these methods, the nominal control law is augmented with a term that dominates the uncertainty δ. A weakness is that if a continuous control law is desired, and δ is nonzero at the desired operating point, only boundedness of the solution can be guaranteed, and not asymptotic stability. To achieve asymptotic stability, a discontinuous control law must be used which may lead to chatter in the control input.

In this chapter we focus on achieving asymptotic stability in the presence of a nonvanishing input uncertainty, using a continuous control law. We restrict our discussion to the case of a constant disturbance $\delta(x, u) = \theta$ and consider the system

$$\dot{x} = f(x) + B(u + \theta)$$

This can be interpreted as compensating for the effects of δ at the desired operating point and disregarding its variation with x and u.

A general result from output regulation theory is that to suppress a disturbance, the controller should incorporate an internal model of the disturbance (Byrnes and

Isidori 2000). For a constant disturbance θ, this means that an integrator should be included in the controller.

Su et al. (2001) consider a nonlinear system subject to input disturbances, and propose to augment the nominal control law with an integral term $\varepsilon\int_0^t(r-y)dt$ where r is the reference signal to be tracked by the system output y. Using singular perturbation methods, the authors show that given a Lyapunov function for the nominal system, an upper bound on ε can be computed such that the augmented controller yields complete disturbance suppression.

Here, two other methods for suppressing the disturbance θ are developed. One is based on adaptive backstepping and the other on nonlinear observer techniques. Compared to adding integral control as in Su et al. (2001), the proposed methods provide faster suppression of input disturbances, but do not achieve output regulation when disturbances are introduced elsewhere.

The remainder of this chapter is organized as follows. In Section 6.1 the control problem is stated in detail. In Section 6.2, the adaptive backstepping approach is presented, and in Section 6.3, the observer based approach is pursued. The methods are illustrated in Section 6.4 with a simulation example and also experiments using a magnetic levitation system. Section 6.5 contains the conclusion of the chapter.

This chapter is based on

> O. Härkegård and S. T. Glad. Control of systems with input nonlinearities and uncertainties: An adaptive approach. In *Proc. of the European Control Conference*, pages 1912–1917, Porto, Portugal, Sept. 2001a.

6.1 Problem Formulation

Consider the nonlinear system

$$\dot{x} = f(x) + B(u+\theta) \tag{6.2}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input, and $\theta \in \mathbb{R}^m$ is an unknown, constant disturbance. Let $u = k(x)$ be a control law satisfying the following assumption.

Assumption 6.1 *Assume that a nominal control law $u = k(x)$ and a Lyapunov function $V(x)$ are known such that*

$$\dot{V}(x) = V_x(x)(f(x) + Bk(x)) = -W(x) < 0, \quad x \neq 0$$

Thus, if θ was known, the control law

$$u = k(x) - \theta$$

would make $x = 0$ a GAS equilibrium of the closed loop system according to Theorem 3.1. How do we deal with the fact that θ is not available? A straightforward solution is to replace θ by an estimate $\hat{\theta}$ and form the certainty equivalence

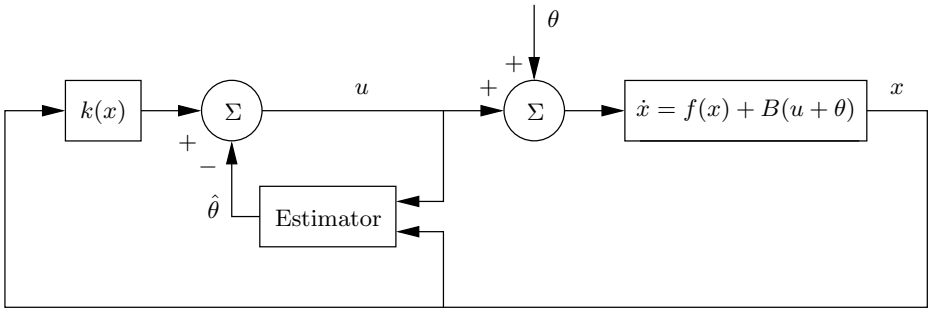


Figure 6.1: Illustration of the system (6.2) and the certainty equivalence controller (6.3). The aim is for the estimator to achieve $\hat{\theta} = \theta$ so that the input disturbance θ is completely cancelled.

controller

$$u = k(x) - \hat{\theta} \tag{6.3}$$

Figure 6.1 illustrates the approach. This strategy is intuitively appealing, but leads to two important questions:

- How can we estimate θ?
- Can we retain global stability using $\hat{\theta}$ for feedback?

Two approaches to the problem will be pursued. In Section 6.2, we use adaptive backstepping to find an estimator that guarantees closed loop stability without adjusting the control law (6.3). In Section 6.3, the starting point is that a converging estimator is given, based on nonlinear observer techniques. The question then is how to possibly adjust the control law to retain stability.

In both approaches, $B \in \mathbb{R}^{n\times m}$ is assumed to have rank m. If $\operatorname{rank} B = k < m$, then B can be factorized as $B = B_1 B_2$ where $B_1 \in \mathbb{R}^{n\times k}$ and $B_2 \in \mathbb{R}^{k\times m}$ both have rank k. This gives the dynamics

$$\dot{x} = f(x) + B_1(B_2 u + B_2\theta) \triangleq f(x) + B_1(\tilde{u} + \tilde{\theta})$$

The methods that we develop can be used to estimate $\tilde{\theta}$, which is what is needed for disturbance suppression since if $\tilde{\theta}$ is known, the control law

$$u = k(x) - G\tilde{\theta}$$

where G is any right inverse of B_2 such that $B_2 G = I$, achieves

$$\dot{x} = f(x) + B_1 B_2(k(x) - G\tilde{\theta}) + B_1\tilde{\theta} = f(x) + Bk(x)$$

and thus makes $x = 0$ a GAS equilibrium.

6.2 Adaptive Backstepping

In adaptive backstepping (Krstić et al. 1995), unknown parameters are dealt with by extending the Lyapunov function with a term that penalizes the parameter estimation error. The parameter update law is then designed to achieve a negative time derivative of the Lyapunov function. Using adaptive backstepping to estimate and adapt to the input disturbance in (6.2) gives the following result.

Theorem 6.1 *Consider the system (6.2) and assume that B has rank m. Let $k(x)$ and $V(x)$ satisfy Assumption 6.1. Then, the control law*

$$u = k(x) - \hat{\theta} \tag{6.4a}$$

$$\dot{\hat{\theta}} = \Gamma(V_x(x)B)^T \tag{6.4b}$$

where $\Gamma \in \mathbb{R}^{m\times m}$ is any symmetric, positive definite matrix, makes $x = 0$, $\hat{\theta} = \theta$ a GAS equilibrium.

Proof: Define the extended Lyapunov function

$$V_a(x, \tilde{\theta}) = V(x) + \frac{1}{2}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta}$$

where $\tilde{\theta} = \theta - \hat{\theta}$ and Γ is a positive definite matrix. Introduce the update rule

$$\dot{\hat{\theta}} = \tau(x, \hat{\theta})$$

When the control law (6.4a) is applied we get

$$\begin{aligned}\dot{V}_a &= V_x(x)(f(x) + B(k(x) - \hat{\theta} + \theta)) - \tau(x, \hat{\theta})^T\Gamma^{-1}\tilde{\theta} \\ &= -W(x) + (V_x(x)B - \tau(x, \hat{\theta})^T\Gamma^{-1})\tilde{\theta}\end{aligned}$$

The first term is negative definite according to the assumptions, and the second, mixed term is indefinite. Since $\tilde{\theta}$ is not available, we cancel the second term by selecting

$$\tau(x, \hat{\theta}) = \tau(x) = \Gamma(V_x(x)B)^T$$

The resulting closed loop system becomes

$$\begin{aligned}\dot{x} &= f(x) + B(k(x) + \tilde{\theta}) \\ \dot{\tilde{\theta}} &= -\Gamma(V_x(x)B)^T\end{aligned} \tag{6.5}$$

which satisfies

$$\dot{V}_a(x, \tilde{\theta}) = -W(x)$$

This expression is zero only when $x = 0$. For x to remain at the origin, $B\tilde{\theta} = 0$ must hold. This implies $\tilde{\theta} = 0$ since B has full column rank. Thus, $x = 0$, $\hat{\theta} = \theta$ is a GAS equilibrium according to Theorem 3.2. □

The control law (6.4a) can be rewritten as

$$u(t) = k(x(t)) - \int_0^t \Gamma(V_x(x(s))B)^T \, ds$$

Thus, using adaptive backstepping to estimate and compensate for a constant input disturbance corresponds to adding integral action from certain states, which depend on the Lyapunov function $V(x)$.

6.3 Observer Based Adaptation

In the previous section, the θ estimator was a result of assigning a negative Lyapunov function time derivative. In this section, we first design an estimator and then investigate how to adjust the certainty equivalence control law (6.3).

An exponentially converging θ estimate can be achieved by viewing θ as a constant state and using the output injection method proposed by Krener and Isidori (1983) to design a nonlinear observer. However, since the separation principle valid for linear systems does not hold for nonlinear systems in general, closed loop stability may be lost when $\hat{\theta}$ is used for feedback as in (6.3). The following theorem states how to augment the control law to guarantee global asymptotic stability.

Theorem 6.2 *Consider the system (6.2) and assume that B has rank m. Let $k(x)$ and $V(x)$ satisfy Assumption 6.1. Then, the control law*

$$u = k(x) - \Gamma(V_x(x)B)^T - \hat{\theta} \tag{6.6}$$

together with the observer

$$\begin{aligned} \dot{\hat{x}} &= f(x) + B(u + \hat{\theta}) + K_1(x - \hat{x}) \\ \dot{\hat{\theta}} &= K_2(x - \hat{x}) \end{aligned} \tag{6.7}$$

where $\Gamma \in \mathbb{R}^{m \times m}$ is any symmetric, positive definite matrix and the observer gains K_1, K_2 are such that $\begin{pmatrix} -K_1 & B \\ -K_2 & 0 \end{pmatrix}$ is Hurwitz[1], makes $x = 0$, $\hat{x} = x$, $\hat{\theta} = \theta$ a GAS equilibrium.

Proof: Viewing θ as a constant state yields the augmented dynamics

$$\begin{aligned} \frac{d}{dt}\begin{pmatrix} x \\ \theta \end{pmatrix} &= \begin{pmatrix} f(x) + B(u+\theta) \\ 0 \end{pmatrix} = \begin{pmatrix} f(x) \\ 0 \end{pmatrix} + \underbrace{\begin{pmatrix} 0 & B \\ 0 & 0 \end{pmatrix}}_{A} \begin{pmatrix} x \\ \theta \end{pmatrix} + \begin{pmatrix} B \\ 0 \end{pmatrix} u \\ y = x &= \underbrace{\begin{pmatrix} I & 0 \end{pmatrix}}_{C} \begin{pmatrix} x \\ \theta \end{pmatrix} \end{aligned} \tag{6.8}$$

[1] A matrix is Hurwitz if all its eigenvalues are in the open left half plane.

Since the nonlinearity $f(x)$ is a function of measurable states only, an observer for θ is given by

$$\frac{d}{dt}\begin{pmatrix}\hat{x}\\ \hat{\theta}\end{pmatrix} = \begin{pmatrix}f(x)+B(u+\hat{\theta})\\ 0\end{pmatrix} + \begin{pmatrix}K_1\\ K_2\end{pmatrix}(x-\hat{x}) \tag{6.9}$$

Including the nonlinear dynamics in the observer is known as output injection (Krener and Isidori 1983). The estimation error $\varepsilon = (\tilde{x} \quad \tilde{\theta})^T = (x-\hat{x} \quad \theta-\hat{\theta})^T$ has linear dynamics:

$$\dot{\varepsilon} = \begin{pmatrix}-K_1 & B\\ -K_2 & 0\end{pmatrix}\varepsilon = A_\varepsilon \varepsilon \tag{6.10}$$

Since $(A,\, C)$ is observable (due to that $\operatorname{rank} B = m$), the eigenvalues of A_ε can be placed arbitrarily.

Next, we determine a Lyapunov function for the observer. For any observer gains K_1, K_2 such that A_ε is Hurwitz, we can find a symmetric, positive definite matrix Q that satisfies

$$\frac{d}{dt}\varepsilon^T Q\varepsilon = -\varepsilon^T \underbrace{\begin{pmatrix}I_n & 0\\ 0 & \Gamma^{-1}\end{pmatrix}}_{M}\varepsilon = -\tilde{x}^T\tilde{x} - \tilde{\theta}^T\Gamma^{-1}\tilde{\theta} \tag{6.11}$$

where Γ is any symmetric, positive definite matrix, by solving the Lyapunov equation (Rugh 1996, p. 124)

$$QA_\varepsilon + A_\varepsilon^T Q = -M$$

To investigate closed loop stability, we combine the original Lyapunov function $V(x)$ with $\varepsilon^T Q\varepsilon$ and form

$$V_o(x,\varepsilon) = V(x) + \varepsilon^T Q\varepsilon$$

We also augment the control law (6.3) with an extra term $\bar{u}$ to compensate for using $\hat{\theta}$ for feedback. The resulting control law

$$u = k(x) + \bar{u} - \hat{\theta} \tag{6.12}$$

yields

$$\begin{aligned}\dot{V}_o &= V_x(x)(f(x) + B(k(x)+\bar{u}-\hat{\theta}+\theta)) - \varepsilon^T M\varepsilon\\ &= -W(x) - \tilde{x}^T\tilde{x} + V_x(x)B(\bar{u}+\tilde{\theta}) - \tilde{\theta}^T\Gamma^{-1}\tilde{\theta}\end{aligned}$$

By choosing

$$\bar{u} = -\Gamma(V_x(x)B)^T \tag{6.13}$$

we can perform a completion of squares.

$$\begin{aligned}\dot{V}_o &= -W(x) - \tilde{x}^T\tilde{x} - V_xB\Gamma(V_xB)^T + V_xB\tilde{\theta} - \tilde{\theta}^T\Gamma^{-1}\tilde{\theta}\\ &= -W(x) - \tilde{x}^T\tilde{x} - \frac{3}{4}\tilde{\theta}^T\Gamma^{-1}\tilde{\theta} - \big[(V_xB)^T - \frac{1}{2}\Gamma^{-1}\tilde{\theta}\big]^T\Gamma\big[(V_xB)^T - \frac{1}{2}\Gamma^{-1}\tilde{\theta}\big]\\ &< 0, \quad x\neq 0,\ \tilde{x}\neq 0,\ \tilde{\theta}\neq 0\end{aligned}$$

Thus, the control law (6.12) in combination with the observer (6.9) makes $x = 0$, $\hat{x} = x$, $\hat{\theta} = \theta$ a GAS equilibrium. □

From the proof one can note that the error dynamics (6.10) of the observer are independent of the system trajectory $x(t)$ and the control input $u(t)$. Since $\hat{\theta} = \theta + \tilde{\theta}$, the same goes for the estimate $\hat{\theta}$. This is not true for the adaptive backstepping estimate in (6.4b), which on the contrary is driven by $x(t)$.

In some cases, the extra term in (6.6) is unnecessary, and the original certainty equivalence controller (6.3) by itself achieves closed loop stability. One such case is when the nominal state feedback control law $k(x)$ solves a common optimal control problem.

Corollary 6.1 *Consider the system (6.2) and let B have rank m. Assume that for $\theta = 0$, $u = k(x)$ solves an optimal control problem of the form*

$$\min_u \int_0^\infty (q(x) + u^T R(x) u) dt \tag{6.14}$$

where $q(x)$ is a positive definite function and $R(x)$ is a symmetric positive definite matrix satisfying

$$u^T R(x) u \leq u^T R_0 u, \quad \forall x$$

where R_0 is positive definite. Then, the control law

$$u = k(x) - \hat{\theta}$$

together with the observer (6.7) makes $x = 0$, $\hat{x} = x$, $\hat{\theta} = \theta$ a GAS equilibrium.

Proof: From Section 3.6.2 we known that if $u = k(x)$ is optimal w.r.t. (6.14) it can be expressed as

$$k(x) = -\frac{1}{2} R^{-1}(x)(V_x B)^T$$

for some Lyapunov function $V(x)$ solving the corresponding Hamilton-Jacobi-Bellman equation. Furthermore, V satisfies

$$\dot{V} = -W(x) = -q(x) - \frac{1}{4} V_x B R^{-1}(x)(V_x B)^T$$

when the optimal control law is used.

Selecting $\Gamma = R_0^{-1}$ in (6.11) and $\bar{u} = 0$ in (6.12) yields

$$\begin{aligned}
\dot{V}_o &= -q(x) - \tilde{x}^T \tilde{x} - \frac{1}{4} V_x B R^{-1}(x)(V_x B)^T + V_x B \tilde{\theta} - \tilde{\theta}^T R_0 \tilde{\theta} \\
&= -q(x) - \tilde{x}^T \tilde{x} - \tilde{\theta}^T (R_0 - R(x)) \tilde{\theta} \\
&\quad - \Big[\frac{1}{2}(V_x B)^T - R(x)\tilde{\theta}\Big]^T R^{-1}(x) \Big[\frac{1}{2}(V_x B)^T - R(x)\tilde{\theta}\Big] \\
&< 0, \quad x \neq 0,\ \tilde{x} \neq 0,\ \tilde{\theta} \neq 0
\end{aligned}$$

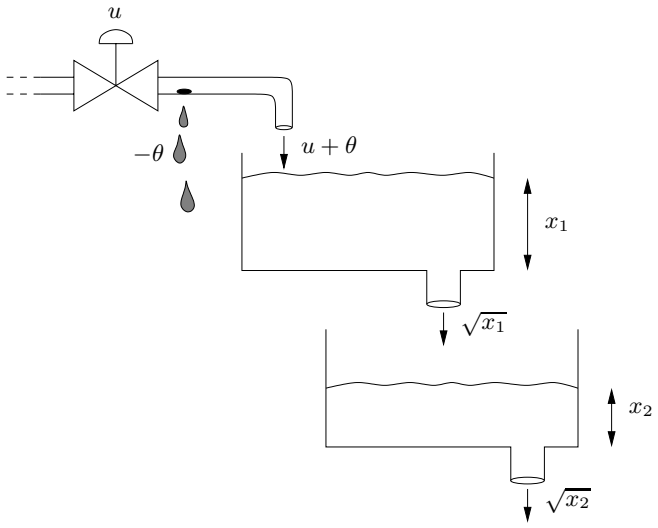


Figure 6.2: Two tanks connected in series. Due to the pipe leakage, the net inflow to the upper tank is $u + \theta$ where $\theta \leq 0$.

Hence the control law $u = k(x) - \hat{\theta}$ together with the observer (6.7) makes $x = 0$, $\hat{x} = x$, $\hat{\theta} = \theta$ a GAS equilibrium. □

Thus, in this case we do not need to augment the certainty equivalence control law (6.3) with an extra state feedback term in order to guarantee global stability. An intuitive interpretation of this result is that some of the optimal control effort can be sacrificed in order to compensate for using the estimate $\hat{\theta}$ for feedback.

6.4 Examples

We now illustrate the properties of the two strategies using a water tank simulation example, and experiments with a magnetic levitation system.

6.4.1 A Water Tank Example

Consider two water tanks connected as in Figure 6.2. The objective is to control the water level in the lower tank and achieve $x_2 = r$. The control input u is the waterflow through the valve of the input pipe. Due to the leakage $\theta \leq 0$ of the pipe, the net flow into the upper tank is $u + \theta$. Using Bernoulli's equation and setting all constants to unity, the system dynamics become

$$\begin{aligned} \dot{x}_1 &= -\sqrt{x_1} + u + \theta \\ \dot{x}_2 &= -\sqrt{x_2} + \sqrt{x_1} \end{aligned} \tag{6.15}$$

where x_1 and x_2 are the water levels of the upper and lower tank. respectively. This system fits (6.2) with $B = (1 \quad 0)^T$.

Let us first find a globally stabilizing control law $u = k(x)$ for the case $\theta = 0$. We do this using an *ad hoc* Lyapunov approach. At the desired steady state, $x_1 = x_2 = r$. This motivates the control Lyapunov function

$$V(x) = \frac{1}{2}(x_1 - r)^2 + \frac{a}{2}(x_2 - r)^2, \quad a > 0$$

which satisfies

$$\begin{aligned}
\dot{V}(x) &= (x_1 - r)(-\sqrt{x_1} + u) + a(x_2 - r)(-\sqrt{x_2} + \sqrt{x_1}) \\
&= \underbrace{-(x_1 - r)(\sqrt{x_1} - \sqrt{r}) - a(x_2 - r)(\sqrt{x_2} - \sqrt{r})}_{-W_1(x)} \\
&\quad + (x_1 - r)(u - \sqrt{r}) + a(x_2 - r)(\sqrt{x_1} - \sqrt{r}) \\
&= -W_1(x) + (\sqrt{x_1} - \sqrt{r})\Big[(\sqrt{x_1} + \sqrt{r})(u - \sqrt{r}) + a(x_2 - r)\Big] \\
&= -W_1(x) - b(x_1 - r)^2 < 0, \quad x_1 \neq r,\ x_2 \neq r
\end{aligned}$$

if we use the control law

$$u = k(x) = \sqrt{r} + b(r - x_1) + \frac{a}{\sqrt{x_1} + \sqrt{r}}(r - x_2) \quad a > 0,\ b \geq 0 \tag{6.16}$$

The parameters are selected as $a = 1$, $b = 0.5$.

Let us now evaluate the expressions involved in the two approaches for compensating for the leakage in the control law.

- *Observer based adaptation:* Since only x_1 is directly affected by θ we use the reduced observer

  $$\begin{aligned}
  \dot{\hat{x}}_1 &= -\sqrt{x_1} + u + \hat{\theta} + k_1(x_1 - \hat{x}_1) \\
  \dot{\hat{\theta}} &= k_2(x_1 - \hat{x}_1)
  \end{aligned}$$

  Computing the input-output description of this observer gives us

  $$\hat{\theta} = \frac{k_2}{s^2 + k_1 s + k_2}(sx_1 - (-\sqrt{x_1} + u)) = \frac{k_2}{s^2 + k_1 s + k_2}\theta$$

  if we replace sx_1 with the expression for $\dot{x}_1$. Thus, $\hat{\theta}$ is θ filtered through a low-pass filter. Choosing the observer gains $k_1 = 1$, $k_2 = 0.5$ places the observer poles in $-0.5 \pm 0.5i$.

  The additional term in the control law (6.6) becomes $\gamma_o(r - x_1)$. However, since $b = 0.5 > 0$ in the nominal controller (6.16), we can regard this term as included in the control law already, and we may select $\gamma_o = 0$ and use the control law

  $$u = k(x) - \hat{\theta}$$

- *Adaptive backstepping:* The update rule (6.4b) for estimating θ becomes

$$\dot{\hat{\theta}} = \gamma_a (V_x B)^T = \gamma_a (x_1 - r), \quad \gamma_a > 0 \tag{6.17}$$

With this, the control law (6.4a) becomes

$$u = k(x) + \gamma_a \int_0^t (r - x_1(s)) ds \tag{6.18}$$

Thus, using adaptive backstepping in this case corresponds to adding integral action from control error in the upper tank. Here we select $\gamma_a = 0.3$.

- *Integral control:* The results of Su et al. (2001) on nonlinear integral control can also be used to suppress θ. Since x_2 is the controlled variable this gives the control law

$$u = k(x) + \gamma_i \int_0^\infty (r - x_2(s)) ds$$

Comparing with (6.3) we can interpret this as using the estimator

$$\dot{\hat{\theta}} = \gamma_i (x_2 - r)$$

Here we select $\gamma_i = 0.1$.

The following scenario is simulated. The initial water level, which is also fed to the observer, is 1 m in both tanks. The control objective is for the lower tank x_2 to reach the reference level $r = 4$ m and maintain this level. At $t = 40$ s a pipe leakage of $\theta = -3\,\mathrm{m}^3/\mathrm{s}$ occurs. At $t = 70$ s, an extra inflow of $1\,\mathrm{m}^3/\mathrm{s}$ to the *lower* tank is introduced. Note that this disturbance was not included in the model (6.15).

Figure 6.3 shows the simulation results. The observer based approach does the best job in estimating and adapting to the pipe leakage θ. Note though that this method is "blind" to the extra inflow at $t = 70$ s, which does not affect $\hat{\theta}$, and gives a steady state error. Adaptive backstepping also does a good job in dealing with the pipe leakage, but displays poor performance in dealing with the inflow disturbance. This is due to that the control law (6.18) forces the upper tank to maintain its nominal level $x_1 = 4$.

The integral control approach is the least direct in estimating the pipe leakage θ, since it relies only on x_2 measurements, and gives the longest response time (increasing γ_i yields very oscillatory responses). However, it is the only method that achieves regulation of the lower tank level despite the extra inflow. To reduce the oscillations, the control law could be adjusted to only integrate the difference between x_2 and the nominal closed loop x_2 response that would result in the absence of model errors and disturbances.

6.4.2 Magnetic Levitation System

Consider next a magnetic levitation system, illustrated in Figure 6.4, in which an electromagnet is used to counteract gravity and make a steel ball levitate. The

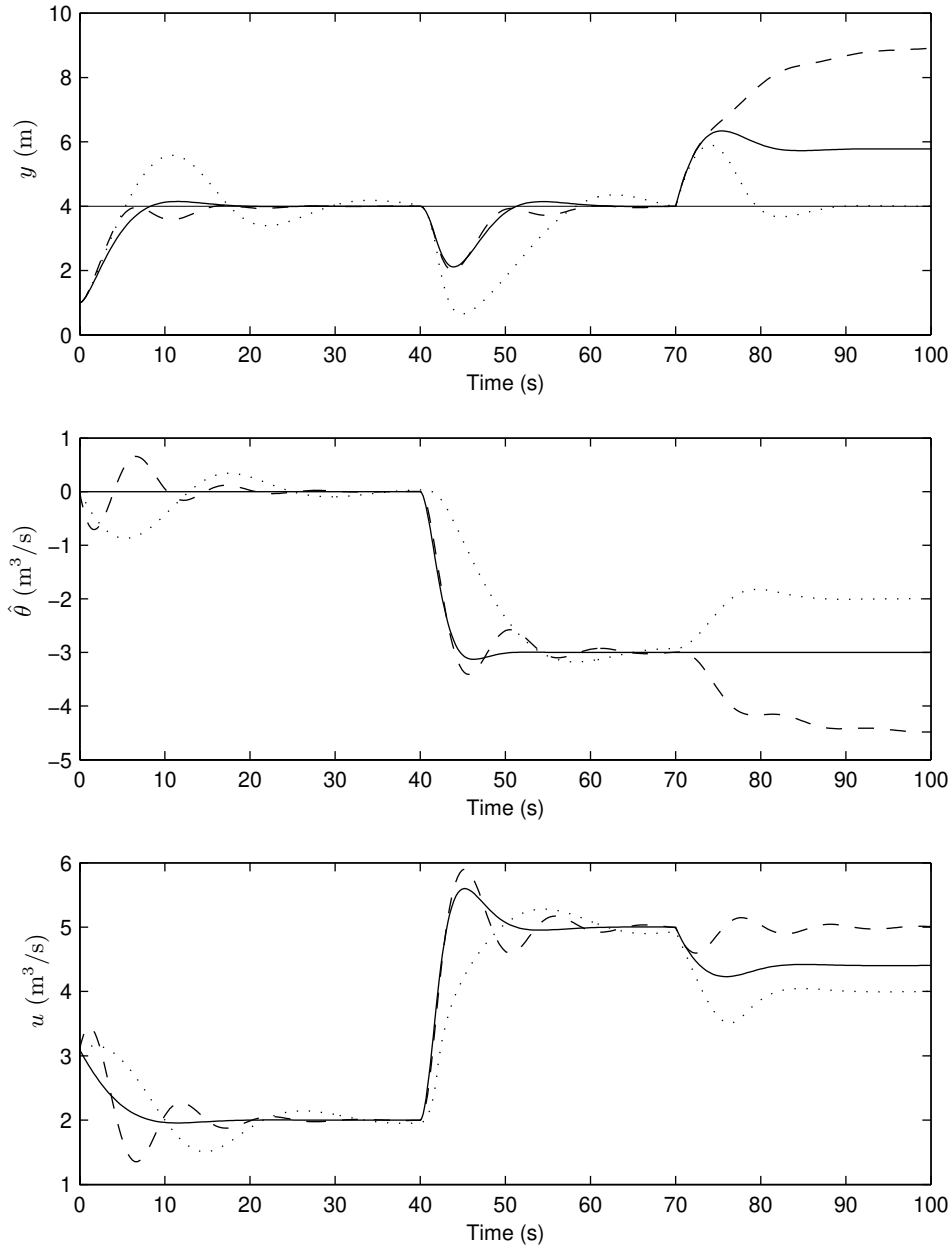


Figure 6.3: Lower tank level (top), leakage estimate (middle), and control input (bottom) when the observer based approach (solid), adaptive backstepping (dashed), and regular integral control (dotted) is used to suppress external disturbances. All methods handle the leakage at $t = 40$ s well, but only regular integral control suppresses the unmodeled inflow in the lower tank at $t = 70$ s.

Figure 6.4: A magnetic levitation system in which an electromagnet is used to make a steel ball levitate.

dynamics of the levitating ball are

$$\dot{x}_1 = x_2$$
$$\dot{x}_2 = g - \frac{F}{m}$$

where x_1 = ball position (air gap between ball and magnet plus offset), x_2 = ball velocity, g = gravitational acceleration, F = magnetic force, and m = ball mass. The system is equipped with a sensor measuring the air gap x_1. The quality of this sensor is high enough for the velocity x_2 to be determined by numerically differentiating the x_1 signal.

A model of the magnetic force, based on Gentili and Marconi (2001) and Yang and Miyazaki (2001), is given by

$$F = k\frac{(u+U_0)^2}{(x_1+X_0)^2}$$

where u = voltage input and k, U_0, and X_0 are constants. The following parameter estimates were achieved from simple experiments using the built-in hardware controller:

$$k/m = 23.9$$
$$X_0 = 0.88$$
$$U_0 = 0.46$$

To control the airgap x_1, a state feedback control law will not suffice, but needs to be complemented with some form of integral action. First of all, the parameter estimates above are rather uncertain. Thus, using a static control law will give a steady state position error. Secondly, the system properties are time varying and depend on how long the electromagnet has been switched on. This calls for some kind of adaptivity in the controller. We also want to the controller to handle variations in the mass resulting from, e.g., adding a second steel ball.

We therefore rewrite the dynamics as

$$\dot{x} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} (v + \theta)$$

where $v = g - F/m$ is nominal acceleration based on the parameter estimates above. The uncertainty θ represents additional, unmodeled contributions to the net acceleration. This is a very coarse way to model uncertainties in the individual parameters, but allows us to use the methods developed earlier in the chapter to achieve zero steady state position error.

The observer based approach from Section 6.3 is used to estimate θ. For the case $\theta = 0$, the following PD-controller brings the ball position to the reference level $x_1 = r$:

$$v = k(x) = K_P(r - x_1) - K_D x_2$$

With $K_P = 14$, $K_D = 100$ this control law is optimal w.r.t. the LQ performance index

$$\min_v \int_0^\infty (196(x_1 - r)^2 + 9972x_2^2 + v^2)dt$$

which according to Corollary 6.1 means that the certainty equivalence controller

$$v = k(x) - \hat{\theta} \tag{6.19}$$

can be used. Using only x_2 to estimate θ gives the observer

$$\dot{\hat{x}}_2 = v + \hat{\theta} + k_1(x_2 - \hat{x}_2)$$
$$\dot{\hat{\theta}} = k_2(x_2 - \hat{x}_2)$$

The observer gains $k_1 = 1.4$, $k_2 = 1.0$ place the observer poles in $-0.7 \pm 0.7i$. Solving for the true control input u gives

$$u = \sqrt{\frac{m}{k}(g - v)}(x_1 + X_0) - U_0 \tag{6.20}$$

Figures 6.5 and 6.6 show the results from two different experiments. In these figures, $-x_1$ is plotted instead of x_1 so that the curves illustrate the position of the ball relative to the ground instead of relative to the electromagnet.

In the first experiment, a single steel ball was inserted under the electromagnet. Measurement data were collected for a square wave reference signal using the control law (6.20), with and without the $\hat{\theta}$ bias correction term in (6.19). In Figure 6.5

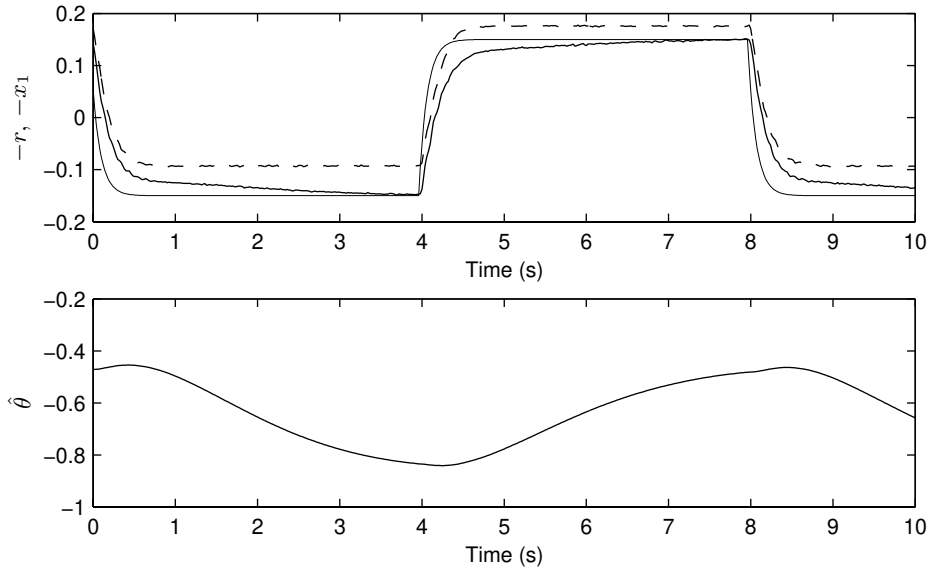


Figure 6.5: Top: Reference trajectory, $-r$ (thin line), together with ball position, $-x_1$, when adaptation is turned on (thick, solid line) and off (dashed line). Bottom: Bias estimate, $\hat{\theta}$. Without adaptation there is a steady state position error.

we see that without adaptation, there is a steady state position error (dashed line). This error vanishes when the adaptation is switched on (solid line). The figure also shows the bias estimate which varies between -0.45 and -0.85 depending on the operating point.

In the second experiment, the reference signal was held constant while a second steel ball was attached to an already levitating ball of the same size. The results are shown in Figure 6.6. Again, the steady state error converges to zero as the bias estimate picks up the effects of the second ball. Note that if the electromagnet had no effect on this second ball, attaching this ball would correspond to a twice as large gravitational pull as before, i.e., an increase in θ of 9.8. In the figure we see that $\hat{\theta}$ only increases by about 5.

6.5 Conclusions

In this chapter we have proposed two methods for estimating and adapting to a constant input disturbance, given a globally stabilizing control law for the undisturbed system. The nonlinear observer approach gives a control law which efficiently suppresses the input disturbance θ, but may be "blind" to disturbances entering the dynamics elsewhere. Adaptive backstepping results in adding integral control from those states directly affected by the input disturbance. The resulting controller may

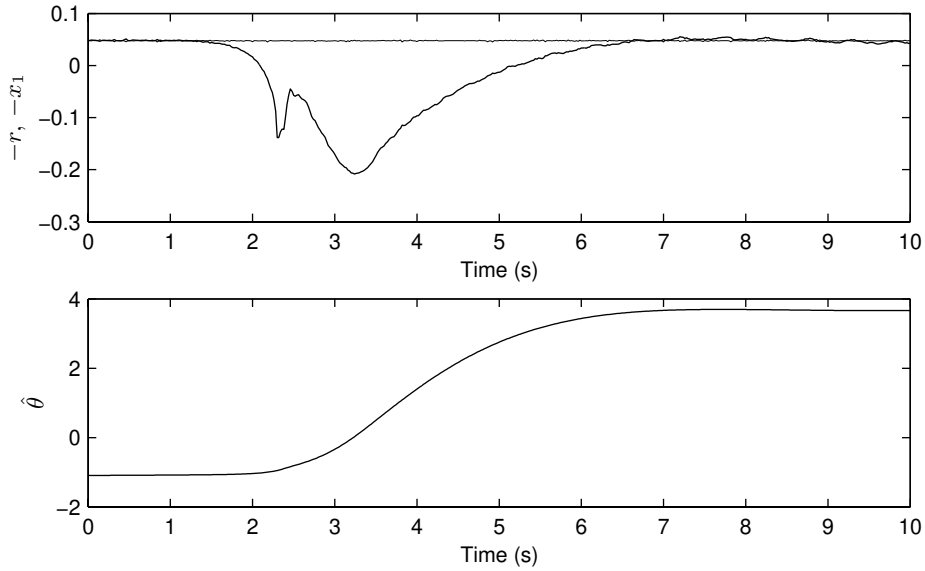


Figure 6.6: Top: Reference trajectory, $-r$ (thin line), and ball position, $-x_1$ (thick line). Bottom: Bias estimate, $\hat{\theta}$. Between $t = 2$ and $t = 3$, a second ball is attached to the first one. As the bias estimate converges, the position error returns to zero.

behave poorly when other disturbances are introduced. In comparison, adding integral control from the controlled variables, as in Su et al. (2001), in general gives slower suppression of input disturbances, but can also be used to suppress other types of disturbances.

Part II

Control Allocation

Chapter 7

Introduction to Part II

This part of the thesis is about control allocation. Control allocation is useful for control of overactuated systems, and deals with distributing the total control demand among the individual actuators. Using control allocation, the actuator selection task is separated from the regulation task in the control design.

To introduce the ideas behind control allocation, consider the system

$$\dot{x} = u_1 + u_2$$

where x is a scalar state variable, and u_1 and u_2 are control inputs. x can be thought of as the velocity of a unit mass object affected by a net force $v = u_1 + u_2$ produced by two actuators. Assume that to accelerate the object, the net force $v = 1$ is to be produced. There are several ways to achieve this. We can choose to utilize only the first actuator and select $u_1 = 1$, $u_2 = 0$, or to gang the actuators and use $u_1 = u_2 = 0.5$. We could even select $u_1 = -10$, $u_2 = 11$, although this might not be very practical. Which combination to pick is the control allocation problem.

This type of actuator redundancy can be found in several applications. In aerospace control, effectors like aerosurfaces, thrust vectoring vanes, and reaction control system jets are used to produce the net torque acting on the vehicle, governing its motion (Durham 1993, Shertzer et al. 2002). To perform dynamic positioning of marine vessels, a set of thrusters is used to produce translational forces and yawing moment, in order to keep the vessel in place with the heading in the desired direction (Lindfors 1993, Berge and Fossen 1997, Sørdalen 1997). In yaw control for cars, the net yawing moment depends on the individual brake forces generated by each wheel (Kiencke and Nielsen 2000, sec. 8.2). In cruise control for trucks, the engine, the wheel brakes, and the retarder contribute to the net acceleration (Axehill and Sjöberg 2003). Cement mixing is another example (Westerlund et al. 1980, Lundán and Mattila 1974). Here the composition of the cement raw meal, which is to be held constant, is governed by the output from five different silos containing raw material. Since the resulting composition is characterized by only three composition ratios, there is a certain degree of freedom.

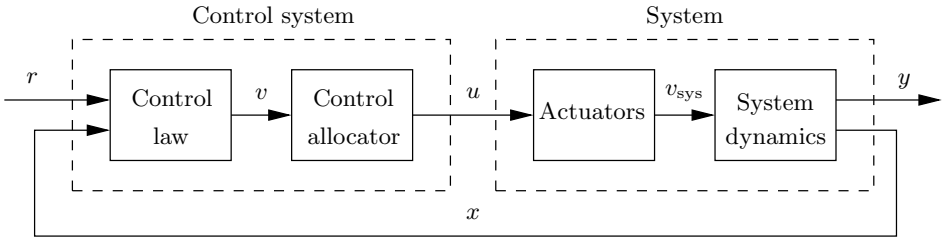


Figure 7.1: Control system structure when control allocation is used. The control system is made up by a control law, specifying which total control effect, v, should be produced, and a control allocator, which distributes this control demand among the individual actuators, u. In the system, the actuators generate a total control effect, v_{sys}, which determines the system behavior. If the control allocation is successful, $v_{\text{sys}} = v$.

Today, control allocation is a research topic in aerospace control and marine vessel control.

For these types of overactuated systems, it is possible to split the control design into the following two steps.

1. Design a control law specifying which total control effort to be produced (net torque, force, etc.).

2. Design a control allocator that maps the total control demand onto individual actuator settings (commanded aerosurfaces deflections, thrust forces, etc.).

Figure 7.1 illustrates the configuration of the overall control system.

So why should the control design be split into two separate tasks? In nonlinear control theory the separation comes naturally with design methods like feedback linearization and backstepping. The resulting control laws specify which total control effect to produce but not how to produce it, and hence the need for control allocation arises. In linear control theory, there is a wide range of control design methods, like LQ design and $\mathcal{H}_\infty$ control, which perform control allocation and regulation in one step (Zhou et al. 1996). Thus, the usefulness of control allocation for linear systems is not so obvious.

There are however other, more practical reasons to use a separate control allocation module, even for linear systems. One benefit is that actuator constraints can be taken into account. If one actuator saturates, and fails to produce its nominal control effect, another actuator may be used to make up the difference. This way, the control capabilities of the actuator suite are fully exploited before the closed loop performance is degraded (Durham 1993). The way the system performance degrades can also be affected. For example, in flight control, it might be crucial to maintain yaw control performance to avoid yaw departure, while roll control may be less important (Enns 1998).

Another benefit is that reconfiguration can be performed if the effectiveness of the actuators change over time, or in the event of an actuator failure, without

having to redesign the control law (Eberhardt and Ward 1999a, Wise et al. 1999).

A third major benefit is that the actuator utilization can be treated independently and can be optimized for the application considered. The actuator redundancy can be used for several purposes. Most commonly, the extra degrees of freedom are used to optimize some secondary objective, like total aerosurface deflections, drag, wing load, or radar signature in aircraft applications (Virnig and Bodden 1994, Wise et al. 1999, Eberhardt and Ward 1999a, Buffington 1997), or total thrust in ship control applications (Lindfors 1993, Sørdalen 1997). Another possibility is to include filtering in the control allocation procedure, to obtain different control distributions among the actuators at different frequencies (Davidson et al. 2001, Papageorgiou et al. 1997).

The remainder of this introductory chapter is organized as follows. In Section 7.1, we formalize the control allocation problem, investigate its properties and introduce some notation. In Section 7.2 we investigate to which classes of systems, linear and nonlinear, that control allocation can be applied in order to resolve actuator redundancy, and also point out two obstacles to using control allocation. Existing methods for control allocation are surveyed in Section 7.3, and in Section 7.4 corresponding numerical methods are presented. Finally, in Section 7.5, the contents of the remaining chapters on control allocation are outlined.

7.1 The Control Allocation Problem

Publications on control allocation are almost exclusively application driven. As a result, the notion and terminology used depend on the application in mind. In this section, an effort is made to develop a generic statement of the control allocation problem, and to introduce proper notation and terminology to be used throughout this thesis.

Mathematically, a *control allocator*[1] solves an underdetermined, typically constrained, system of equations. The input to the control allocator is the total control effect to be produced, the *virtual control input*[2] $v(t) \in \mathbb{R}^k$. The output of the control allocator is the *true control input* $u(t) \in \mathbb{R}^m$, where $m > k$.

Given $v(t)$, $u(t)$ is sought such that

$$g(u(t)) = v(t) \tag{7.1}$$

where $g : \mathbb{R}^m \mapsto \mathbb{R}^k$ is the mapping from the true to the virtual control input in the system to be controlled. In the control allocation literature, the linear case is almost exclusively studied, for which (7.1) becomes

$$Bu(t) = v(t) \tag{7.2}$$

where the *control effectiveness matrix* B is a $k \times m$ matrix with rank k.

[1] Also known as control mixer, control blender, control selector, or control distributor.
[2] Also known as generalized control input or pseudo-control input.

To incorporate actuator position constraints we require that

$$u_{\min} \leq u(t) \leq u_{\max} \tag{7.3}$$

where the inequalities apply componentwise. If actuator rate constraints also exist, we further require that

$$\rho_{\min} \leq \dot{u}(t) \leq \rho_{\max} \tag{7.4}$$

Since the control allocator is part of a digital control system, it is reasonable to approximate the time derivative as

$$\dot{u}(t) \approx \frac{u(t) - u(t-T)}{T} \tag{7.5}$$

where T is the sampling time (Durham and Bordignon 1996). This enables the rate constraints to be rewritten as position constraints. Combining (7.3)-(7.5) yields

$$\underline{u}(t) \leq u(t) \leq \overline{u}(t) \tag{7.6}$$

where

$$\begin{aligned} \underline{u}(t) &= \max\{u_{\min}, u(t-T) + T\rho_{\min}\} \\ \overline{u}(t) &= \min\{u_{\max}, u(t-T) + T\rho_{\max}\} \end{aligned} \tag{7.7}$$

are the overall position constraints at time t.

Equation (7.2) constrained by (7.6) constitute the standard formulation of the linear control allocation problem. Dropping the time dependence yields

$$Bu = v \tag{7.8a}$$
$$\underline{u} \leq u \leq \overline{u} \tag{7.8b}$$

which is the standard constrained linear control allocation problem.

The solution set is given by the intersection between the hyperplane $Bu = v$ and the position constraints hyperbox $\underline{u} \leq u \leq \overline{u}$. Since both of these are convex sets, the solution set will be convex. Thus, trying to solve (7.8) has three possible outcomes:

1. There is an infinite number of solutions,
2. there is one unique solution, or
3. no solution exists.

Example 7.1 Consider two control inputs, u_1 and u_2, contributing to one virtual control input, v, through

$$v = 2u_1 + u_2$$

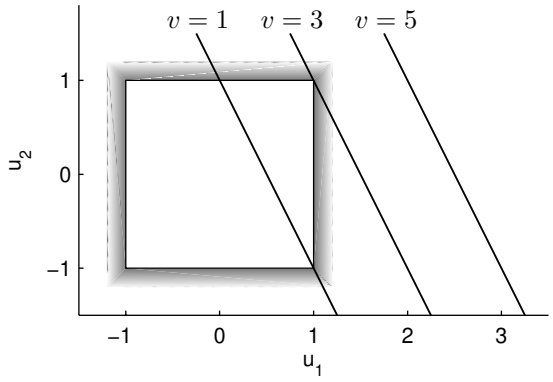


Figure 7.2: Virtual control demand $2u_1 + u_2 = v$ for different values of v (black lines), and position constraints $-1 \leq u_1, u_2 \leq 1$ (shaded box). For $v = 1$ there are infinitely many solutions, for $v = 3$ there is one unique solution, and for $v = 5$ there is no feasible solution.

Let there be position constraints given by

$$-1 \leq u_1 \leq 1$$
$$-1 \leq u_2 \leq 1$$

Using the notation of (7.8) we have

$$B = \begin{pmatrix} 2 & 1 \end{pmatrix}, \quad \underline{u} = \begin{pmatrix} -1 \\ -1 \end{pmatrix}, \quad \overline{u} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Figure 7.2 illustrates the situation for different values of v. For $v = 1$ there is a set of solutions given by the part of the line tagged "$v = 1$" that satisfies the box constraints. For $v = 3$, $u = (1 \quad 1)^T$ is the only solution. For $v = 5$ there is no solution to the problem, i.e., there is no actuator combination that yields the desired virtual control input.

In case 1, there is some degree of freedom in choosing the control input while satisfying (7.8), i.e., without affecting the closed loop dynamics. As we will see in our survey of control allocation methods in Section 7.3, this freedom can be used for example to optimize some objective like minimum use of control input (optimization based control allocation), or to prevent the use of certain actuators until all other actuators have saturated (daisy chaining).

In case 2, where there is one unique control input that produces the desired virtual control input, the task for the control allocator is to find this control input.

In case 3, the desired virtual control input cannot be produced, and one has to settle for a different (typically "lower") value of v which is attainable. This is known as command limiting. Buffington (1997) suggests to decompose v into a number of terms with different priority depending on their function (e.g., stabilization, reference following, etc.), and then start by limiting the term with the lowest priority. Bodson and Pohlchuk (1998) suggest a number of alternatives including scaling the reference inputs and reducing the control requirements, e.g., the closed loop bandwidth.

Command limiting can also be implemented as a part of the control allocation scheme. This is done by letting the control allocator determine a feasible u such that the produced virtual control input, Bu, approximates v well in some sense. The approximation can be such that some, possibly weighted, norm of $Bu - v$ is minimized (optimization based control allocation), or such that u produces the maximum virtual control input possible in the direction of v (direct control allocation).

7.2 When Can Control Allocation Be Used?

So when can control allocation be used to resolve actuator redundancy? Essentially, for control allocation to be applicable, the system needs to be separable as shown in Figure 7.1. This is not the case for all systems where the number of control inputs exceeds the number of controlled variables. Let us therefore investigate to which classes of linear and nonlinear systems that control allocation can be applied.

7.2.1 Linear Systems

Consider first a linear dynamic system on state space form,

$$\dot{x} = Ax + B_u u \tag{7.9}$$

where $x \in \mathbb{R}^n$ is the state vector, $u \in \mathbb{R}^m$ is the control input, $A \in \mathbb{R}^{n\times n}$, and $B_u \in \mathbb{R}^{n\times m}$. Assume that B_u has rank $k < m$. Then B_u has a nullspace of dimension $m - k$ in which we can perturb the control input without affecting $\dot{x}$. Thus, there are several choices of control input that gives the same system dynamics. This is the type of redundancy that can be resolved using control allocation.

Since B_u is rank deficient it can be factorized as

$$B_u = B_v B \tag{7.10}$$

where $B_v \in \mathbb{R}^{n\times k}$ and $B \in \mathbb{R}^{k\times m}$ both have rank k. Introducing the virtual control input

$$v = Bu$$

where $v \in \mathbb{R}^k$, we can rewrite the systems dynamics (7.9) as

$$\dot{x} = Ax + B_v v \tag{7.11}$$

Now, control design can be performed in two steps, as outlined in the introduction.

In cases where B_u is not rank deficient, but ill-conditioned, Virnig and Bodden (1994) suggest to use the singular value decomposition (see Appendix B) to approximately factorize B_u and enable for control allocation to be used. The authors apply this idea to aircraft flight control design.

7.2.2 Nonlinear Systems

The same ideas can be used to deal with nonlinear systems of the form

$$\dot{x} = f(x, g(x, u)) \tag{7.12}$$

where $f : \mathbb{R}^n \times \mathbb{R}^k \mapsto \mathbb{R}^n$ and $g : \mathbb{R}^n \times \mathbb{R}^m \mapsto \mathbb{R}^k$ where $k < m$. Introducing the virtual control input

$$v = g(x, u) \tag{7.13}$$

where $v \in \mathbb{R}^k$, we can rewrite (7.12) as

$$\dot{x} = f(x, v)$$

and again use a two-step control design.

A special class of nonlinear systems is systems of the form

$$\begin{aligned} \dot{x} &= f(x) + g_u(x, u) \\ g_u(x, u) &= B_v g(x, u) \end{aligned}$$

where $B_v \in \mathbb{R}^{n \times k}$ and f and g are nonlinear mappings as above. Again introducing $v = g(x, u)$ yields

$$\dot{x} = f(x) + B_v v$$

Note that these resulting dynamics are affine in v, which simplifies many nonlinear design methods like, for example, backstepping.

Solving (7.13) for u, while considering the actuator constraints (7.8b), amounts to performing constrained nonlinear programming. Since control allocation is to be performed in real time, this may not be computationally feasible. One way to resolve this problem is to approximate (7.13) locally with an affine mapping. Linearizing g around u_0 yields

$$g(x, u) \approx g(x, u_0) + \underbrace{\frac{\partial g}{\partial u}(x, u_0)}_{B(x)} \cdot (u - u_0) \tag{7.14}$$

which leads to the linear control allocation problem

$$\bar{v} = B(x)u$$

where

$$\bar{v} = v - g(x, u_0) + B(x)u_0 \tag{7.15}$$

and methods for linear control allocation can be used. In Bordignon (1996, chap. 12) and Doman and Oppenheimer (2002), u_0 is picked as the previously applied control input, $u(t - T)$.

7.2.3 Obstacles

In the preceding sections we have discussed techniques for dividing a control design for an overactuated system into a feedback design part and a linear control allocation part. In this section we discuss two important obstacles to using these techniques.

Actuator Dynamics

A first obstacle is actuator dynamics. Consider the linear case and assume that the system description (7.9) does not capture the dynamics of the actuators. That is, u represents the true actuator positions rather than the commanded ones. If the actuators are independently controlled and have first order dynamics, the actuator dynamics can be described by

$$\dot{u} = B_a(u^{\text{cmd}} - u) \tag{7.16}$$

where B_a is a diagonal $m \times m$ matrix and $u^{\text{cmd}} \in \mathbb{R}^m$ represents the commanded actuator positions, i.e., the true control input. Combining (7.9) and (7.16) we get

$$\begin{pmatrix} \dot{x} \\ \dot{u} \end{pmatrix} = \begin{pmatrix} A & B_u \\ 0 & -B_a \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} + \begin{pmatrix} 0 \\ B_a \end{pmatrix} u^{\text{cmd}}$$

We note that the resulting B-matrix, $(0 \quad B_a^T)^T$, has full column rank (m) and therefore cannot be factorized into two matrices with lower dimension as in (7.10). Hence, control allocation cannot be used directly.

Actuator dynamics are certainly present in aircraft applications where v typically represents the aerodynamic moments to be produced in pitch, roll, and yaw, and u^{cmd} and u represent the commanded and actual deflections of the aerodynamics control surfaces, respectively.

The most common solution to the problem is to simply neglect the actuator dynamics, and consider u to be the actual control input in (7.9). This works as long as the closed loop system is designed to be substantially slower than the actuator servo systems, since then (7.16) can be approximated by the steady state relationship $u = u^{\text{cmd}}$.

In cases where this assumption breaks down, Venkataraman and Doman (2001) and Page and Steinberg (2002) propose to speed up the actuator dynamics by augmenting the existing actuator servo with an additional controller.

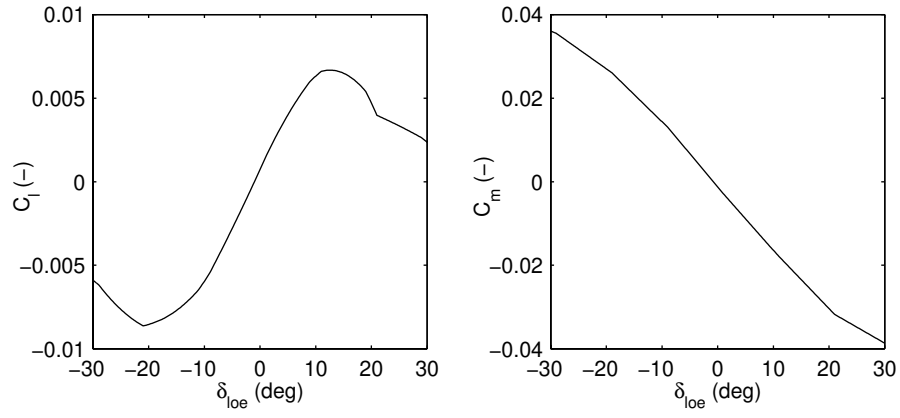


Figure 7.3: Aerodynamic moment coefficients for the ADMIRE model in roll, C_l (left), and in pitch, C_m (right), as functions of the left outboard elevon deflection, δ_{loe}, at 2000 m, Mach 0.8. The fact that $C_l(\delta_{loe})$ is nonmonotonic may be a problem if control allocation based on local linearization is used.

Nonmonotonic Nonlinearities

Nonmonotonic nonlinearities in the mapping (7.13) from u to v constitute another important obstacle (Doman and Oppenheimer 2002). If the mapping is not monotonic, the linearization (7.14) may be misleading.

There are several examples where nonmonotonic mappings appear in aircraft control allocation. Figure 7.3 shows an example of this for the ADMIRE model (ADMIRE ver. 3.4h 2003), based on aerodata from the Generic Aerodata Model (GAM) (Backström 1997). In the figure, the aerodynamic moment coefficients in roll, C_l, and pitch, C_m, are plotted as functions of the left outboard elevon deflection, δ_{loe}, at 2000 m, Mach 0.8. While $C_m(\delta_{loe})$ is monotonically decreasing, $C_l(\delta_{loe})$ is nonmonotonic. Consider now, for example, a situation where $\delta_{loe}(t - T) = -25^\circ$ and the new roll demand is $C_l(t) = 0$. Linearizing C_l around -25°, as in (7.14), tells us that to increase C_l, δ_{loe} should be decreased. However, since $\left|\delta_{loe}\right| \leq 30^\circ$ is a constraint, we will fail to achieve $C_l = 0$ with this strategy.

A remedy for this problem, proposed by Doman and Oppenheimer (2002), is to alter the position limits and use only the monotonic part of the mapping, in our case $-21^\circ \leq \delta_{loe} \leq 13^\circ$. However, as noted by the authors, altering the position limits may be undesirable in a multivariate case where using the full control authority may be beneficial for producing some other virtual control component. In our case we see that using the full δ_{loe} range is beneficial for producing pitching moment.

Another example of a nonmonotonic mapping is given in Bordignon (1996, chap. 12), based on F-18 simulation data. Here, the yawing moment coefficient, C_n, is plotted as a function of the right aileron deflection, δ_{ra}. The result is a V-shaped curve with $C_n > 0$ for all values of δ_{ra} except $\delta_{ra} = 0$.

7.3 Control Allocation Methods

Let us now survey the most common methods for control allocation appearing in the literature, all of which consider the constrained linear control allocation problem (7.8). Many proposed "methods" correspond to different ways of computing the solution for a certain control allocation objective, rather than different objectives. In this presentation, the aim is to make a clear distinction, for each control allocation method, between which solution is sought and how the solution can be computed numerically. The latter issue is the topic of Section 7.4.

7.3.1 Optimization Based Control Allocation

Optimization based methods rely on the following pragmatic interpretation of the control allocation problem. Given a virtual control command v, determine a feasible control input u such that $Bu = v$. If there are several solutions, pick the best one. If there is no solution, determine u such that Bu approximates v as well as possible.

Description of Method

As a measure of how "good" a solution or an approximation is, the l_p norm is used. For a particular p, we will refer to this as l_p-optimal control allocation. The l_p norm of a vector $u \in \mathbb{R}^m$ is defined as

$$\left\|u\right\|_p = \left(\sum_{i=1}^{m} \left|u_i\right|^p\right)^{1/p} \quad \text{for } 1 \leq p \leq \infty \tag{7.17}$$

The optimal control input is given by the solution to a two-step optimization problem.

$$\begin{aligned} u &= \arg\min_{u \in \Omega} \left\|W_u(u - u_d)\right\|_p \\ \Omega &= \arg\min_{\underline{u} \leq u \leq \overline{u}} \left\|W_v(Bu - v)\right\|_p \end{aligned} \tag{7.18}$$

Here, u_d is the desired control input and W_u and W_v are weighting matrices. Equation (7.18) should be interpreted as follows: Given Ω, the set of feasible control inputs that minimize $Bu - v$ (weighted by W_v), pick the control input that minimizes $u - u_d$ (weighted by W_u).

In (7.18), u_d, W_u, and W_v are design parameters. u_d is the desired control input to which solution of (7.18) is attracted if there is no unique feasible minimizer of $\left\|W_v(Bu - v)\right\|_p$. This situation certainly occurs when (7.8) has several solutions, but may also occur when (7.8) has no solution, as noted by Bodson (2002). The choice of u_d may correspond to minimum control deflections, drag, radar signature, or wing loading. W_u allows for actuator prioritization, i.e., which actuators should be used primarily. Similarly, W_v allows for prioritization among the virtual control inputs when (7.8) has no solution. In the aircraft case, for example, this corresponds to prioritizing among the moments produced in pitch, roll, and yaw.

On the Choice of Norm

The l_2 norm is the most frequently used (Virnig and Bodden 1994, Eberhardt and Ward 1999a, Buffington 1997, Enns 1998, Reiner et al. 1995, Snell et al. 1992). This is much due to that the unconstrained minimum norm control allocation problem

$$\begin{aligned} \min_u \quad & \|u\|_2 \\ \text{subject to} \quad & Bu = v \end{aligned}$$

has an explicit solution given by

$$u = B^\dagger v$$

where $B^\dagger = B^T(BB^T)^{-1}$ is the pseudoinverse of B, see Appendix B. A similar result can be derived for the general case, $u_d \neq 0$, $W_u \neq I$, see Lemma B.1. This fact is exploited in many of the numerical schemes for computing the solution to (7.18), see Section 7.4.3.

Using the l_1 norm has also been proposed (Lindfors 1993, Enns 1998, Buffington et al. 1999, Ikeda and Hood 2000, Shertzer et al. 2002). A motivation for this choice is that in general, a linear program can be solved faster than a quadratic one.

Let us compare the characteristics of these two norms by applying optimization based control allocation to the system in Example 7.1.

Example 7.2 Consider the control allocation problem in Example 7.1 where $B = \begin{pmatrix} 2 & 1 \end{pmatrix}$ and let $v = 1$ be the desired virtual control input. Let the optimization objective be given by (7.18) with $u_d = \begin{pmatrix} 0 & 0 \end{pmatrix}^T$, $W_u = I$, $W_v = 1$. Since $v = 1$ is attainable, (7.18) reduces to

$$\begin{aligned} \min_u \quad & \|u\|_p && \text{(7.19a)} \\ \text{subject to} \quad & Bu = 1 && \text{(7.19b)} \\ & \underline{u} \leq u \leq \overline{u} && \text{(7.19c)} \end{aligned}$$

Figure 7.4 illustrates the situation. The optimal solution is the point that lies inside the box (7.19c), on the line (7.19b), and on the level curve that corresponds to the lowest possible value of $\|u\|_p$. For $p = 1$ we get

$$u = \begin{pmatrix} 0.5 \\ 0 \end{pmatrix}$$

while for $p = 2$, the optimal solution is given by

$$u = B^\dagger \cdot 1 = \begin{pmatrix} 0.4 \\ 0.2 \end{pmatrix}$$

The example points out some characteristics of the two norms.

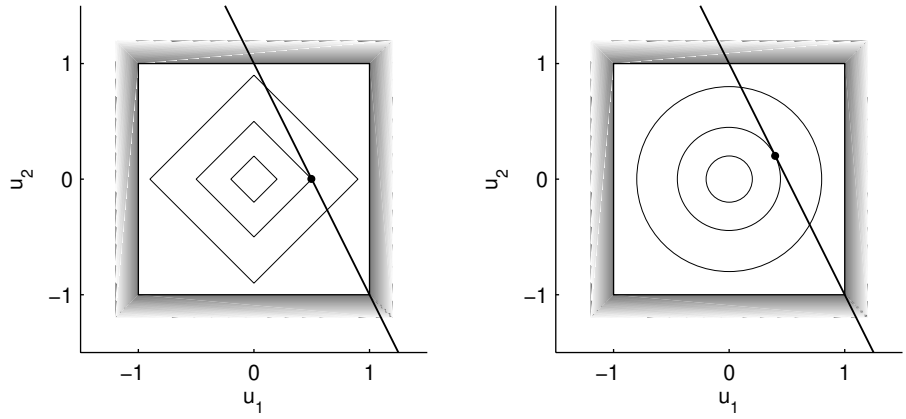


Figure 7.4: Virtual control demand $Bu = 1$ (thick line) and control constraints $\underline{u} \leq u \leq \overline{u}$ (shaded box) along with level curves (thin lines) of the cost function $\left\|u\right\|_p$ for $p = 1$ (left) and $p = 2$ (right). The black dots mark the optimal solutions.

- Both norms favor the use of effective control inputs. In the example, u_1 is more effective than u_2 since it has a higher entry in B. Consequently, $u_1 > u_2$ in both solutions. Note that in general the solution also depends on the choice of W_u.

- The l_2 norm distributes the virtual control demand among all of the control inputs, while the l_1 solution utilizes as few control inputs as possible to satisfy the virtual control demand. In general, using the l_1 norm, all but k control inputs will be either saturated or at their preferred position (given by u_d) (Enns 1998, Bodson 2002).

- The l_2 solution varies continuously with the problem parameters, while the l_1 solution does not. For example, with $B = (2 \quad b_2)$ in the example, where $b_2 \geq 0$, we get the l_2 solution

$$u = B^{\dagger} = \frac{1}{4 + b_2^2} \begin{pmatrix} 2 \\ b_2 \end{pmatrix}$$

which varies continuously with b_2. The l_1 solution,

$$u = \begin{cases} \begin{pmatrix} 0.5 & 0 \end{pmatrix}^T & \text{if } b_2 < 2 \\ \begin{pmatrix} 0 & b_2^{-1} \end{pmatrix}^T & \text{if } b_2 > 2 \end{cases}$$

achieved by inspecting Figure 7.4, has a discontinuity at $b_2 = 2$ for which the optimal solution is not unique.

- If W_u is nonsingular, (7.18) has a unique solution for $p = 2$. For $p = 1$, this is not always the case, as shown above. In both cases, Ω is convex since it results from minimizing a convex function over a convex set. For $p = 2$, $\left\|W_u(u - u_d)\right\|_p$ is strictly convex which leads to a unique minimum.

7.3.2 Direct Control Allocation

Direct control allocation was introduced by Durham (1993). In direct control allocation, the choice of control input is made unique by geometric reasoning.

Description of Method

In the original papers on direct control allocation (Durham 1993, 1994b), the method was described as follows. Given a virtual control demand v, first find the feasible control input u^* that generates the virtual control input $v^* = Bu^*$ of maximum magnitude in the direction of v. Let

$$a = \frac{\left\|v^*\right\|_2}{\left\|v\right\|_2}$$

and select the control input u according to

$$u = \begin{cases} \frac{1}{a}u^* & \text{if } a > 1 \\ u^* & \text{if } a \leq 1 \end{cases} \tag{7.20}$$

Bodson (2002) condensed this verbal description into the following optimization problem. Let a, u^* solve

$$\begin{aligned} \max_{a,u} \quad & a \\ \text{subject to} \quad & Bu = av \\ & \underline{u} \leq u \leq \overline{u} \end{aligned} \tag{7.21}$$

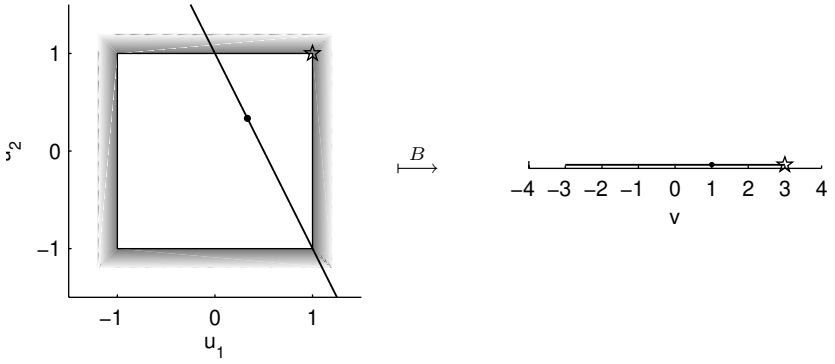
Then assign u as in (7.20).

Example 7.3 Consider Example 7.1 and let $v = 1$. Since v is scalar, finding the maximum virtual control input in v's direction corresponds to finding the largest positive v that can be generated. Figure 7.5 shows the set of feasible control inputs, $-1 \leq u_1, u_2 \leq 1$, mapped onto the set of feasible virtual control inputs, $-3 \leq v \leq 3$. This gives us

$$v^* = 3$$

which is achieved for

$$u^* = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$



(a) Control input, u. (b) Virtual control input, v.

Figure 7.5: The set of feasible control inputs (left) are mapped onto a set of feasible virtual control inputs (right). The black dots in (a) and (b) represent u and v, while the stars represent u^* and v^*.

u^* and v^* are represented by stars in Figure 7.5(a) and 7.5(b), respectively. Further, $a = 3/1 > 1$ gives the control input

$$u = \frac{1}{3}u^* = \begin{pmatrix} 1/3 \\ 1/3 \end{pmatrix}$$

according to (7.20).

Comments

Let us make a few comments regarding the direct control allocation method.

- There are no design variables to be selected. The solution to the control allocation problem is completely determined by the control effectiveness matrix B, and the control constraints $\underline{u}$ and $\overline{u}$.

  In Bordignon and Durham (1995, chap. 13), a modified version of direct allocation was developed in which the control redundancy is used for function optimization by taking incremental steps along the gradient of some user specified function. In Durham and Bolling (1997), these ideas were used for drag minimization.

- For $a > 1$, no element in u will be saturated (Durham 1994b). That is to say, if v lies strictly inside the set of feasible virtual control inputs, u will lie strictly inside the set of feasible control inputs.

- Direct control allocation performs command limiting by direction preservation. If v is not feasible, the generated virtual control input is $Bu = a^{-1}v$, which has the same direction as v, but smaller magnitude.
- The control constraints $\underline{u} \leq u \leq \overline{u}$ must include the origin (Bodson 2002). That is, $u = 0$ must be a feasible control input. This follows from the construction of u in (7.20). When rate constraints are included, as in (7.7), this is typically not the case. A modified method to also handle rate constraints was developed in Durham and Bordignon (1996).
- The choice of u^* is not always unique (Durham 1994a). For example, consider three control inputs, all constrained by -1 and 1, producing two virtual control inputs

$$v = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} u$$

For $v = (0 \quad 1)^T$, (7.21) is solved by $a = 1$, $u^* = (w \quad -w \quad 1)^T$, where $-1 \leq w \leq 1$. The reason why u^* is not unique is that the virtual control inputs produced by u_1 and u_2 are collinear. Durham (1994a) showed that this problem occurs if not all $k \times k$ submatrices of B have full rank.

Methods to make the choice of u^* unique include perturbing the elements of B to avoid rank deficiency (Durham 1994a), and averaging over (a subset of) the possible control inputs (Petersen and Bodson 2000), which in our case would lead to $u^* = (0 \quad 0 \quad 1)^T$.

7.3.3 Daisy Chain Control Allocation

In daisy chain control allocation (Buffington and Enns 1996, Bordignon 1996, Durham and Bordignon 1996), the allocator suite is divided into groups which are successively employed to generate the total control effort.

Description of Method

To begin with, the m control inputs are divided into M groups,

$$u = \begin{pmatrix} u^1 \\ \vdots \\ u^M \end{pmatrix}$$

after possibly reordering the control inputs. The control effectiveness matrix is partitioned accordingly,

$$B = \begin{pmatrix} B_1 & \dots & B_M \end{pmatrix}$$

The control allocation problem (7.8a) can now be restated as solving

$$B_1 u^1 + B_2 u^2 + \cdots + B_M u^M = v$$

The daisy chain idea is to first try and satisfy this virtual control demand using only the first group of actuators by solving

$$B_1 u^1 = v \tag{7.22}$$

for u^1. If $\operatorname{rank} B_1 \geq \dim v = k$, (7.22) is solved by

$$u^1 = P_1 v \tag{7.23}$$

where P_1 is any right inverse of B_1, see Appendix B. If $\operatorname{rank} B_1 < k$, v can, in general, not be produced using only the control inputs in u^1. In this case, we also pick a solution of the form (7.23), but have to settle for a P_1 producing only an approximate solution to (7.22).

If u^1 satisfies (7.22), as well as the actuator position and rate constraints, the allocation was successful and the procedure halts. Otherwise, u^1 is saturated according to its position and rate constraints,

$$u^1 = \operatorname{sat}_{u^1}(P_1 v)$$

and the second group of actuators is employed by solving

$$B_2 u^2 = v - B_1 u^1 \tag{7.24}$$

for u^2, yielding some, possibly approximate, solution $u_2 = P_2(v - B_1 u_1)$. Again, if u^2 fails to satisfy (7.24) or violates some constraint, the solution is saturated and u^3 is employed to make up the difference. This procedure is repeated until either the virtual control demand is met, or all actuator groups have been employed.

The daisy chain control allocation procedure can be summarized as

$$\begin{aligned} u^1 &= \operatorname{sat}_{u^1}(P_1 v) \\ u^2 &= \operatorname{sat}_{u^2}\Big(P_2\big(v - B_1 u^1\big)\Big) \\ &\vdots \\ u^M &= \operatorname{sat}_{u^M}\left(P_M\big(v - \sum_{i=1}^{M-1} B_i u^i\big)\right) \end{aligned} \tag{7.25}$$

where $B_i P_i = I$, if possible, and $\operatorname{sat}_{u^i}(\cdot)$ saturates its argument with respect to the position and rate constraints of actuator group i, for $i = 1, \ldots, M$. Figure 7.6 illustrates the procedure for $M = 2$.

Example 7.4 Consider again Example 7.1 where $B = (2 \quad 1)$. Assume that it is preferable to use u_1 for control, and that u_2 should be used only if necessary. To enforce this we use daisy chain control allocation with the two actuator groups

$$\begin{aligned} u^1 &= u_1 \\ u^2 &= u_2 \end{aligned}$$

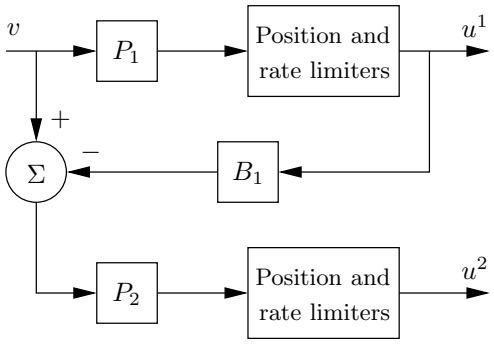


Figure 7.6: Daisy chain control allocation illustrated for the case of two control input groups. u^2 is nonzero only if the virtual control demand v cannot be produced using u^1 alone.

with control effectiveness $B_1 = 2$ and $B_2 = 1$, respectively, which gives $P_1 = 1/2$ and $P_2 = 1$. The procedure thus becomes

$$u_1 = u^1 = \mathrm{sat}_{u_1}\left(\frac{1}{2}v\right)$$
$$u_2 = u^2 = \mathrm{sat}_{u_2}(v - 2u_1)$$

where

$$\mathrm{sat}_{u_1}(x) = \mathrm{sat}_{u_2}(x) = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \leq x \leq 1 \\ 1 & \text{if } x > 1 \end{cases}$$

For $v = 1$ we get

$$u_1 = \mathrm{sat}_{u_1}(0.5) = 0.5$$
$$u_2 = \mathrm{sat}_{u_2}(1 - 2 \cdot 0.5) = 0$$

Comments

- The design choices consist of the actuator groupings and the P_i matrices. Note that if k actuators are placed in each group, and the resulting square submatrices B_i have full rank, each P_i is uniquely determined by $B_i P_i = I$.
- The method may fail to produce virtual control inputs which are indeed feasible (Bordignon 1996, chap. 6), (Bordignon and Durham 1995). In our example though, where v is a scalar, the daisy chain method is able to produce all attainable virtual control inputs, $-3 \leq v \leq 3$.

- In aircraft applications, daisy chain control allocation is often used when thrust vectoring (see Section 2.1.4) is available, see, e.g., Adams et al. (1994) and Enns et al. (1994). Conventional control surfaces, such as elevator, aileron, and rudder, are then primarily used for control, and the thrust vectoring vanes are used for auxiliary control.

7.4 Numerical Methods for Control Allocation

In the previous section, three different methods for control allocation were reviewed. For daisy chain control allocation, (7.25) defines the method, including how to actually compute the control input, u. For optimization based control allocation and direct control allocation, (7.18) and (7.20)–(7.21) define the methods. However, these expressions only specify which solutions are sought, and do not tell us how to find them. In this section, we survey the literature for numerical methods for optimization based and direct control allocation.

7.4.1 Optimization Based Control Allocation

Regardless of the choice of norm, a common trick is to approximately reformulate the sequential optimization problem (7.18) as a weighted optimization problem,

$$u_W = \arg \min_{\underline{u} \le u \le \overline{u}} \Big(\big\| W_u(u - u_d) \big\|_p^p + \gamma \big\| W_v(Bu - v) \big\|_p^p \Big) \tag{7.26}$$

To emphasize that primarily, $Bu - v$ should be minimized, a large value of the weighting factor γ can be used. In Björck (1996, p. 192) it is shown that for $p = 2$,

$$\lim_{\gamma \to \infty} u_W(\gamma) = u_S$$

where u_S is the solution to (7.18). This reformulation allows the optimization to be performed in one step, which may be more efficient than solving the origonal sequential optimization problem.

7.4.2 l_1-Optimal Control Allocation

First consider l_1-optimal control allocation. In this case, (7.18) and (7.26) can be rewritten as linear programming problems (Buffington et al. 1999, Bodson 2002), which enables for standard linear programming tools to be used, such as the simplex method (Luenberger 1984). Bodson (2002) considers the weighted formulation (7.26) and proposes the revised simplex method (Luenberger 1984, pp. 59-65) to exploit that $m \gg k$ in control allocation applications. Ikeda and Hood (2000) also use (7.26) and report the use of a Boeing in-house optimization scheme "*similar to simplex method*".

7.4.3 l_2-Optimal Control Allocation

For l_2-optimal control allocation, there is a much richer selection of numerical methods proposed.

Pseudoinverse Methods

Most existing methods can be classified as pseudoinverse methods. These methods exploit the fact that if we disregard the actuator constraints, (7.18) reduces to

$$\begin{aligned} \min_u \quad & \left\| W_u(u - u_d) \right\|_2 \\ \text{subject to} \quad & Bu = v \end{aligned}$$

which, using Lemma B.1, has the closed form solution

$$\begin{aligned} u &= (I - GB)u_d + Gv \\ G &= W_u^{-1}(BW_u^{-1})^\dagger \end{aligned} \tag{7.27}$$

where $\dagger$ is the pseudoinverse operator, see Appendix B. Different methods correspond to different ways of adjusting the nominal solution (7.27) to the actuator constraints.

Durham (1993) considers the case $u_d = 0$, and poses the question whether there is any G such that the pseudoinverse solution (7.27) is feasible for all attainable values of v. Durham refers this set as the attainable moment[3] subset (AMS). Hence, is there a G such that (7.27) solves the control allocation problem for the entire AMS? Durham shows that there is no such choice in general. Further, two methods are proposed for finding the "best" G considering two different objectives. One that allows the user to tailor G in different directions in v-space such that (7.27) becomes feasible for certain values of v on the boundary of the AMS. The second method determines the G that maximizes the part of the AMS in which feasible solutions are computed.

Snell et al. (1992) suggest a much simpler way to determine G to avoid infeasible solutions. Each control input is weighted with the inverse of its maximum value (assuming symmetric position limits and no rate limits), corresponding to

$$W_u = \text{diag}(1/\overline{u}_1, \ldots, 1/\overline{u}_m)$$

This way, actuators with large operating regions are used more than those with stricter position limits.

As stated above, the pseudoinverse solution (7.27) will not be feasible for all attainable virtual control inputs v, and certainly not for unattainable virtual control inputs. Various ways to accommodate to the constraints have been proposed. The simplest alternative is to truncate (7.27) by clipping those components that violate some constraint. However, since this typically causes only a few control inputs

[3]The word *moment* refers to the fact that Durham discusses the problem from an aircraft control perspective.

to saturate, is seems natural to use the remaining control inputs to make up the difference.

Virnig and Bodden (1994) propose a redistributed pseudoinverse (RPI) scheme, in which all control inputs that violate their bounds in the pseudoinverse solution (7.27) are saturated and removed from the optimization. Then, the control allocation problem is resolved with only the remaining control inputs as free variables.

Bordignon (1996) proposes an iterative variant of RPI. Instead of only redistributing the control effect once, the author proposes to keep redistributing the control effect until either, the pseudoinverse solution is feasible, or, all control inputs are saturated. This is known as the cascaded generalized inverse (CGI) approach. It is notable that the interpretation of RPI is ambiguous. In Eberhardt and Ward (1999b) and Bodson (2002), RPI is described as iterative, and then coincides with CGI. Enns (1998) also suggests to compute the pseudoinverse solution iteratively, as in CGI, but to only saturate one control input per iteration.

What is interesting, and important to point out, is that neither of these redistribution schemes are guaranteed to find the optimal solution to the original allocation problem (7.18), or even to achieve $Bu = v$ when possible (Bodson 2002). Let us illustrate this with a simple example.

Example 7.5 Let us apply CGI (Bordignon 1996) to the control allocation problem

$$2u_1 + u_2 = \underbrace{\begin{pmatrix} 2 & 1 \end{pmatrix}}_{B} u = v$$

constrained by

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \leq u \leq \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

with the design parameters in (7.18) set to $u_d = (0 \quad 0)^T$, $W_u = I$, $W_v = 1$. We will study the case of $v = 3.5$, illustrated in Figure 7.7(a).

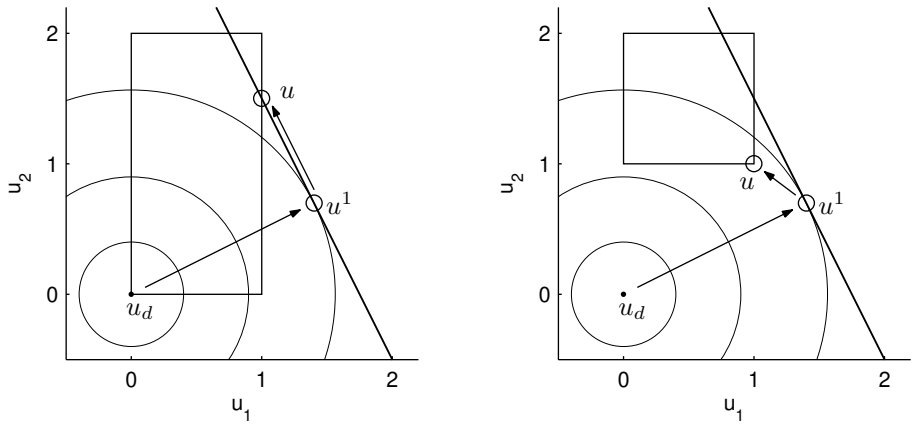
Computing the pseudoinverse solution (7.27) yields the first iterate

$$u^1 = B^\dagger v = \begin{pmatrix} 1.4 \\ 0.7 \end{pmatrix} \tag{7.28}$$

Here, u_1 is infeasible and saturates at $u_1 = 1$. Hence, the control allocation problem is resolved with only u_2 as a free variable. This corresponds to replacing the original B matrix by $\tilde{B} = 1$. Further, the virtual control input that should be produced by u_2 is given by $\tilde{v} = v - 2u_1 = 1.5$. The solution is given by

$$u_2 = \tilde{B}^\dagger \tilde{v} = 1.5$$

which is feasible, and the algorithm stops.



(a) Successful case

(b) Failing case

Figure 7.7: Illustration of the cascaded generalized inverse (CGI) method. The lines represents the virtual control demand $Bu = v$, the boxes represent the actuator constraints, and the circles are the level curves of $\|W_u(u - u_d)\|$. Although the optimal solution is the same in both cases, it is only found in the first case (Figure 7.7(a)).

In this case, CGI was successful since the output,

$$u = \begin{pmatrix} 1 \\ 1.5 \end{pmatrix} \tag{7.29}$$

is the true optimal solution. Let us now look at a case where CGI fails.

Example 7.6 Consider the same allocation problem as in the previous example, but let the constraints instead be

$$\begin{pmatrix} 0 \\ 1 \end{pmatrix} \leq u \leq \begin{pmatrix} 1 \\ 2 \end{pmatrix}$$

see Figure 7.7(b). Such tight position constraints could result from including rate constraints as in (7.7). Note that this is a subset of the constraints in Example 7.5. Hence, (7.29) is still the optimal solution.

By construction, CGI starts by neglecting the constraints. This means that the first iterate will again be given by (7.28). This time, both control inputs violate their position constraints. Saturating with respect to the constraints yields

$$u = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Since no free variables remain, the algorithm stops. Note that the CGI solution fails to meet the virtual control demand and only achieves $v = Bu = 3$.

In Chapter 8 we will return to this example and show how active set methods can be used to determine the optimal solution.

Fix-point Method

Burken et al. (2001) propose a fix-point iteration algorithm, based on Barnard (1976) and Lu (1996), for solving the weighted l_2-optimal control allocation problem (7.26). The authors only consider the case of $u_d = 0$ in which the algorithm becomes

$$u^k = \text{sat}\Big((1-\varepsilon)\omega B^T Q_1 v - (\omega H - I)u^{k-1}\Big), \quad k = 1, \ldots, N$$

where N is the maximum number of iterations and

$$\begin{aligned} \varepsilon &= \frac{1}{\gamma + 1} \\ Q_1 &= W_v^T W_v \\ Q_2 &= W_u^T W_u \\ H &= (1-\varepsilon)B^T Q_1 B + \varepsilon Q_2 \\ w &= \big\|H\big\|_F^{-1} = \left(\sum_{i=1}^{m}\sum_{j=1}^{m} h_{ij}^2\right)^{-\frac{1}{2}} \end{aligned}$$

where h_{ij} are the elements of H. $\big\|H\big\|_F$ is known as the Frobenius norm of H (Golub and Van Loan 1996). $\text{sat}(\cdot)$ is a vector saturator with components

$$\text{sat}_i(u) = \begin{cases} \underline{u}_i, & u_i < \underline{u}_i \\ u_i, & \underline{u}_i \le u_i \le \overline{u}_i \\ \overline{u}_i, & u_i > \overline{u}_i \end{cases}, \quad i = 1, \ldots, m$$

The algorithm can be interpreted as a gradient search method, where w decides the step length, and where the iterates are clipped to satisfy the constraints.

To improve the efficiency, the authors suggest to select the starting point u^0 as the control input from the previous sampling instant, i.e., $u^0 = u(t - T)$. As $N \to \infty$, the outcome u^N converges to the unique optimum of (7.26).

Ellipsoidal Constraints Method

Enns (1998) presents an approximate method for solving (7.18), where the $2m$ box constraints (7.8b) are replaced by one ellipsoidal constraint. The new problem is shown to be solved efficiently using a bisection method.

Active Set Methods

So far, only numerical methods from the control allocation literature have been listed. However, since (7.18) and (7.26) are regular least squares problems, methods from numerical optimization are also of interest.

In the standard optimization literature, so called active set methods are often recommended for small to medium-scale quadratic programming problems, see, e.g., (Björck 1996, Nocedal and Wright 1999). In most aircraft control allocation applications there are $k = 3$ virtual control inputs to be produced (in pitch, roll, and yaw), while the number of actuators, m, is typically about 10. Hence, active set methods could be a viable alternative for control allocation. Despite this, there are to the author's knowledge no papers published on active set methods for l_2-optimal control allocation. Note that in the l_1-case, the simplex method, which is an active set method, has been proposed for control allocation, see Section 7.4.2.

For an active set method to be efficient, it should make use of the special structure of the least squares problems at hand, in particular the fact that the constraints are so called box constraints. An example of this is Lötstedt (1984), who considers a special case of the sequential least squares problem (7.18), namely the minimal least squares problem

$$\begin{aligned} u &= \arg\min_{u\in\Omega} \|u\| \\ \Omega &= \arg\min_{\underline{u}\le u\le\overline{u}} \|W_v(Bu-v)\| \end{aligned} \tag{7.30}$$

where the minimal length solution is picked if $\|W_v(Bu-v)\|$ does not have a unique feasible minimizer. Lötstedt suggests the use of active set methods for solving the problem.

The sequential least squares problem (7.18) can not always be cast into a minimal least squares problem like (7.30) since in the new variables $\tilde{u} = W_u(u-u_d)$ the actuator constraints will in general no longer be simple box constraints. However, if W_u is a diagonal matrix, corresponding to pure scaling of the variables, the box constraint property is preserved.

Efficient active set methods for control allocation is the topic of Chapter 8.

7.4.4 Direct Control Allocation

Let us now consider direct control allocation, and review numerical methods for finding the solution to (7.20)-(7.21).

For $k = \dim v = 3$, Durham has proposed a number of methods (Durham 1994b,a, 1999, 2001), all of which are based on the notion of an attainable moment subset (AMS). The AMS is the k-dimensional image of the m-dimensional hyperbox (7.8b) mapped through the control effectiveness matrix B. Most of the methods solve the problem by associating the faces of the AMS with the faces of the constraints hyperbox. Extensions of these ideas to $k \geq 4$ can be found in Beck (2002).

Recently, Bodson (2002) proposed to use the simplex method to solve the direct allocation problem for arbitrary values of k.

7.5 Outline of Part II

In the remaining three chapters of this part we deal with different aspects of l_2-optimal control allocation for the linear control allocation problem (7.8). Flight control examples are used to illustrate the presented ideas and results. The chapters deal with three different subjects and can be read independently of eachother.

In Chapter 8 we investigate the use of active set methods to numerically compute the optimal solution. Efficient algorithms, that exploit the special structure of the control allocation problem, are developed. The algorithms are shown to be similar in complexity with the pseudoinverse methods in Section 7.4.3.

Chapter 9 is devoted to dynamic control allocation. The performance index in (7.18) is extended with a term to also penalize the actuator rates. This results in a frequency dependent distribution of control among the actuators which can be tuned using weighting matrices.

Finally, in Chapter 10 the relationship between control allocation and linear quadratic control design is investigated, in terms of resolving actuator redundancy. It is shown that for a certain class of linear systems, they are in fact equivalent in that they give the same design freedom.

Chapter 8

Active Set Methods for Optimization Based Control Allocation

The control allocation problem is often stated as a constrained least squares problem, see Section 7.3.1. Despite this, standard tools for constrained programming from numerical optimization are typically not used to find the solution. Instead, numerous approximate schemes have been proposed, such as the pseudoinverse methods and the fix-point method reviewed in Section 7.4.3, none of which are guaranteed to find the optimum in a finite number of iterations.

In this chapter we investigate the use of active set methods to solve the l_2-optimal control allocation problem (7.18):

$$\begin{aligned} u &= \arg\min_{u\in\Omega} \left\|W_u(u-u_d)\right\| \\ \Omega &= \arg\min_{\underline{u}\le u\le\overline{u}} \left\|W_v(Bu-v)\right\| \end{aligned} \tag{8.1}$$

where $\left\|\cdot\right\| = \left\|\cdot\right\|_2$ for notational convenience. Active set methods are used in many of today's commercial solvers for constrained quadratic programming, and can be shown to find the optimal solution in a finite number of iterations (Nocedal and Wright 1999, pp. 466–467). Active set methods resemble the iterative pseudoinverse methods in that inequality constraints are either disregarded, or regarded as equality constraints. What differs are the rules for which constraints to activate, i.e., which control variables to saturate. In the control allocation case, efficiency is achieved by exploiting that the optimization problem does not change much between two samples, and that the actuator constraints are simple box constraints.

Two active set methods are proposed, one for finding the optimal solution to the control allocation problem formulated as a sequential optimization problem, (8.1), and one for the weighted formulation (7.26). Open loop simulations show that

the methods are similar in computation time to the cascaded generalized inverse method of Bordignon (1996), and the fix-point method of Burken et al. (2001), and produce solutions with better accuracy.

The chapter is organized as follows: Section 8.1 contains an introduction to active set methods. In Section 8.2 the use of active set methods in control allocation is motivated. In Section 8.3, the active set methods tailored for control allocation are developed. The numerical details of the methods can be found in Section 8.4. In Section 8.5, the methods are evaluated numerically. Finally, conclusions are drawn in Section 8.6.

This chapter is based on

> O. Härkegård. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 1295–1300, Las Vegas, NV, Dec. 2002b.

8.1 Active Set Methods

Consider the bounded and equality constrained least squares problem

$$\min_u \|Au - b\| \tag{8.2a}$$

$$Bu = v \tag{8.2b}$$

$$Cu \geq U \tag{8.2c}$$

where $C = \begin{pmatrix} I \\ -I \end{pmatrix}$ and $U = \begin{pmatrix} \underline{u} \\ -\overline{u} \end{pmatrix}$ so that (8.2c) is equivalent to $\underline{u} \leq u \leq \overline{u}$. An active set method (Björck 1996, Nocedal and Wright 1999) solves this problem by solving a sequence of equality constrained problems. In each step some of the inequality constraints are regarded as equality constraints, and form the working set $\mathcal{W}$, while the remaining inequality constraints are disregarded. The working set at the optimum is known as the active set of the solution.

Note that this is much like the cascaded generalized inverse method of Bordignon (1996), see Section 7.4.3. The difference is that an active set method is more careful regarding which variables to saturate, and has the ability to free a variable that was saturated in a previous iteration. To check if an iterate is optimal, the Karush-Kuhn-Tucker conditions are used (Nocedal and Wright 1999, p. 328). Specifically, it is checked that all the Lagrange multipliers of the active inequalities are positive.

Algorithm 8.1 describes, in pesudocode, an active set method for solving (8.2), based on Björck (1996, alg. 5.2.1) and Nocedal and Wright (1999, alg. 16.1). The Lagrange multipliers used for optimality checking are determined by

$$A^T(Au - b) = \begin{pmatrix} B^T & C_0^T \end{pmatrix} \begin{pmatrix} \mu \\ \lambda \end{pmatrix} \tag{8.3}$$

Algorithm 8.1 (Active set method for problem (8.2))

Let u^0 be a feasible starting point. A point is feasible if it satisfies (8.2b) and (8.2c). Let the working set $\mathcal{W}$ contain (a subset of) the active inequality constraints at u^0.

for $i = 0, 1, 2, \ldots$

Given a suboptimal iterate u^i, find the optimal perturbation p, considering the inequality constraints in the working set as equality constraints and disregarding the remaining inequality constraints. Solve

$$\min_p \left\| A(u^i + p) - b \right\|$$
$$Bp = 0$$
$$p_i = 0, \ i \in \mathcal{W}$$

if $u^i + p$ is feasible

Set $u^{i+1} = u^i + p$ and compute the Lagrange multipliers in (8.3).

if all $\lambda \geq 0$

u^{i+1} is the optimal solution to (8.2). Stop with $u = u^{i+1}$.

else

Remove the constraint associated with the most negative λ from the working set.

else

Determine the maximum step length α such that $u^{i+1} = u^i + \alpha p$ is feasible. Add the bounding constraint at u^{i+1} to the working set.

end

where C_0 contains the rows of C that correspond to constraints in the working set (Björck 1996, p. 200). μ is associated with (8.2b) and λ with the active constraints in (8.2c).

8.2 Why Use Active Set Methods?

So why should active set methods be suited for control allocation?

Active set methods are the most efficient when a good estimate of the optimal active set is available. Then the number of changes in the working set, i.e., the number if iterations, will be small. In control allocation, a very good estimate is often given by the active set of the solution in the previous sampling instant. This holds as long as the virtual control trajectory, $v(t)$, varies slowly compared to the sampling time. Then, if it was optimal to saturate an actuator in position or in rate in the previous sampling instant, there is a good chance that it is still optimal

to let it saturate.

Another appealing property is that in each iteration, a feasible iterate u^{i+1} is produced that yields a lower value of the cost function than the previous iterate, u^i. Thus, the maximal number of iterations can be set to reflect the computation time available.

Furthermore, the calculations to be performed at each iteration are computationally rather cheap, consisting mainly of solving an equality constrained least squares problem, just as in the iterative pseudoinverse methods in Section 7.4.3. In our implementation, QR decompositions are used for this purpose, see Section 8.4.

8.3 Active Set Methods for Control Allocation

In this Section, the active set methods will be presented at an algorithmic level. The numerical details involved are covered in Section 8.4.

8.3.1 Preliminaries

Let us first list some preliminaries regarding the active set methods to be presented.

The only information forwarded from one optimization, i.e., from one sampling instant, to the next, is the resulting solution and the set of active constraints. All problem parameters, i.e., B, u_d, W_u, W_v, $\underline{u}$, $\overline{u}$, and, of course, v, can be changed to reflect the current control effectiveness of the actuators, the demand for virtual control input, etc. We will require though, that the previous solution is feasible with respect to the new position constraints, i.e., that $\underline{u}(t) \leq u(t-T) \leq \overline{u}(t)$ holds.

The initial working set $\mathcal{W}$ of each optimization is selected as the optimal active set in the previous sampling instant. The components of the starting point, u_i^0, $i = 1, \ldots, m$, are selected as as follows. If u_i did not saturate in the previous sampling instant, then $u_i^0 = u_i(t-T)$ is used. If $u_i(t-T)$ did saturate, then $u_i^0 = \underline{u}(t)$ or $u_i^0 = \overline{u}(t)$ is used, depending on whether u_i saturated at its upper or lower total position limit. Note that this total position limit includes the rate limit of the actuator.

When the control allocator is initiated, and there is no previous solution available, $u^0 = (\underline{u} + \overline{u})/2$ and $\mathcal{W} = \emptyset$ are selected.

The sequential least squares solver requires that no k control inputs produce linearly dependent control efforts, where $k = \dim v$. This is equivalent to that all $k \times k$ submatrices of B must have full rank. This restriction can be removed at the cost of a more complicated algorithm.

The weighting matrices W_u and W_v are assumed to be nonsingular. W_u being nonsingular ensures that the posed optimization problems both have a unique optimal solution, see Section 7.3.1.

8.3.2 Sequential Least Squares

First consider the sequential least squares (SLS) formulation (8.1) of the l_2-control allocation problem.

Algorithm 8.2 (Sequential least squares (SLS) control allocation)

Phase 1:

1. Let $\mathcal{W}$ be the resulting working set from the previous sampling instant, and assign u^0 as described in Section 8.3.1.

2. Solve

$$u_\Omega = \arg\min_u \left\| W_v(Bu - v) \right\|$$
$$\underline{u} \le u \le \overline{u}$$

using Algorithm 8.1 with the following modification. When the number of free variables exceeds $k = \dim v$, in which case the optimal perturbation p is not uniquely determined, pick the minimum perturbation.

3. If $Bu_\Omega = v$, move to phase 2. Otherwise stop with $u = u_\Omega$.

Phase 2:

1. Let $u^0 = u_\Omega$ and let $\mathcal{W}$ be the resulting solution and working set from phase 1.

2. Solve

$$u = \arg\min_u \left\| W_u(u - u_d) \right\|$$
$$Bu = v$$
$$\underline{u} \le u \le \overline{u}$$

using Algorithm 8.1.

Recall from Section 7.4.3 that the minimal least squares (MLS) method of Lötstedt (1984) is an active set method that can be used when W_u is diagonal, since then the change of variables $\tilde{u} = W_u(u - u_d)$ will preserve the box constraint property.

For the general case, Algorithm 8.2 solves this problem in two phases. In phase 1, some point $u_\Omega \in \Omega$ is determined by minimizing $\left\| W_v(Bu - v) \right\|$. If $Bu_\Omega = v$, then u_Ω is used as the starting point for phase 2, in which the secondary objective, $\left\| W_u(u - u_d) \right\|$, is minimized. If $Bu_\Omega \neq v$, the demanded virtual control input is infeasible, and the algorithm stops.

Example 8.1 Let us revisit Example 7.6, in which the cascaded generalized inverse method failed, and see how the SLS method solves the problem. The control allocation problem is defined by $B = (2 \quad 1)$, $v = 3.5$, $\underline{u} = (0 \quad 1)^T$, and $\overline{u} = (1 \quad 2)^T$, and the design parameters are given by $u_d = (0 \quad 0)^T$, $W_u = I$, and

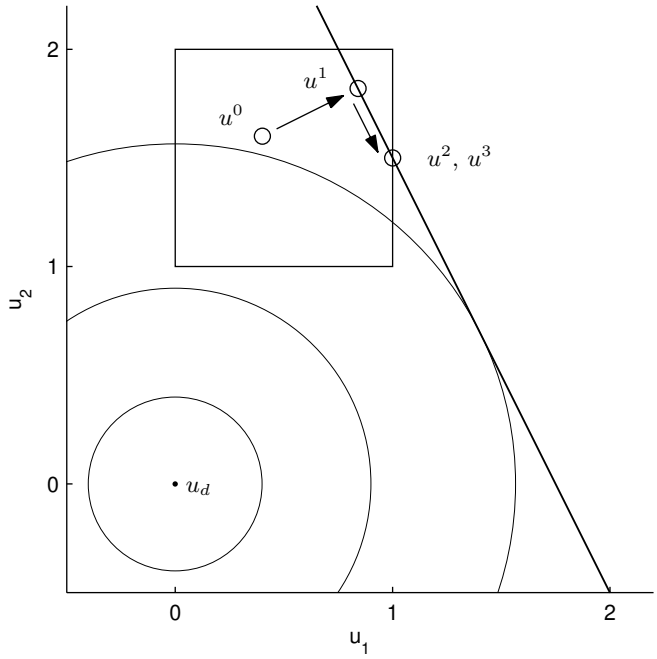


Figure 8.1: Illustration of the SLS method. The solid line represents the virtual control demand $Bu = v$, the box represents the actuator constraints, and the circles are the level curves of $\|W_u(u - u_d)\|$. In phase 1, u^1 is determined as the closest point to the starting point u^0 that satisfies the virtual control demand. In phase 2, $\|W_u(u-u_d)\|$ is minimized. The optimal solution is given by $u = u^3$.

$W_v = 1$. Figure 8.1 illustrates the situation.

The algorithm is initiated with $u^0 = (0.7 \quad 1.7)^T$, assuming this was the previous solution, and no active inequality constraints in $\mathcal{W}$. Solving for the minimum perturbation that minimizes $\|Bu - v\|$ yields $u^1 = u_\Omega = (0.84 \quad 1.82)^T$, see Figure 8.1. Since $Bu_\Omega = v$, phase 2 is entered.

In phase 2, since no box constraint is active, the algorithm starts by solving

$$\min_p \|u_\Omega + p\|$$
$$Bp = 0$$

This yields $u_\Omega + p = (1.4 \quad 0.7)^T$ which is infeasible with respect to $\overline{u}_1 = 1$. Hence, the maximum step length α is solved for, resulting in $u^2 = (1 \quad 1.5)^T$, and $u_1 = 1$ is added to the working set $\mathcal{W}$.

In the next iteration, the optimal perturbation is zero, implying $u^3 = u^2$. The

Algorithm 8.3 (Weighted least squares (WLS) control allocation)

1. Let $\mathcal{W}$ be the resulting working set from the previous sampling instant, and assign u^0 as described in Section 8.3.1.

2. Rewrite the cost function as

$$\left\|W_u(u-u_d)\right\|^2+\gamma\left\|W_v(Bu-v)\right\|^2=\left\|\underbrace{\begin{pmatrix}\gamma^{\frac{1}{2}}W_vB\\W_u\end{pmatrix}}_{A}u-\underbrace{\begin{pmatrix}\gamma^{\frac{1}{2}}W_vv\\W_uu_d\end{pmatrix}}_{b}\right\|^2$$

and solve

$$u=\arg\min_u\left\|Au-b\right\|$$
$$\underline{u}\le u\le\overline{u}$$

using Algorithm 8.1.

Lagrange multiplier of the active inequality constraint is computed using (8.3).

$$u^3=\begin{pmatrix}1\\1.5\end{pmatrix}=\begin{pmatrix}2&-1\\1&0\end{pmatrix}\begin{pmatrix}\mu\\\lambda\end{pmatrix}$$

gives $\lambda=2\ge 0$, which confirms that $u=u^3=(1\quad 1.5)^T$ is the optimal solution the the problem.

8.3.3 Weighted Least Squares

Consider now the approximate weighted least squares formulation (7.26) of the l_2-optimal control allocation problem:

$$u=\arg\min_{\underline{u}\le u\le\overline{u}}\left\|W_u(u-u_d)\right\|^2+\gamma\left\|W_v(Bu-v)\right\|^2 \tag{8.4}$$

where $\gamma\gg 1$ to emphasize that primarily, $Bu-v$ should be minimized. Recall from Section 7.4.1 that the solution of (8.4) converges to that of (8.1) as $\gamma\to\infty$.

Algorithm 8.3 describes the proposed active set based solution[1]. The gain from weighting the two optimization criteria into one is that the two phases of Algorithm 8.2 are now merged into one. In step 2, the weighted constraint rows are ordered first in A and b to avoid numerical problems, see Björck (1996, pp. 192-193).

[1]In Härkegård (2002b), γ was, incorrectly, used instead $\gamma^{\frac{1}{2}}$ in Algorithm 8.3.

Let us briefly discuss the choice of γ. Note that the "effective" penalty on the residuals in the second term of (8.4), relative to the penalty on the residuals in the first term, depends not only on γ but also on W_u, W_v, and B. For example, scaling γ by a factor 100 is equivalent to scaling W_v by 10, or scaling W_u by 0.1. It would be desirable to have an automatic procedure for determining γ, given W_u, W_v, and B, such that the WLS solution approximates the SLS solution well.

To this end, introduce the *effective weight*

$$\gamma_0 = \gamma \frac{\underline{\sigma}(W_v B)^2}{\overline{\sigma}(W_u)^2}$$

which incorporates the effects of all the variables affecting the "effective" scaling as discussed above. Given γ_0, selected by the user, we get

$$\gamma = \gamma_0 \frac{\overline{\sigma}(W_u)^2}{\underline{\sigma}(W_v B)^2} \tag{8.5}$$

to be used in the WLS method. Simulations indicate that $\gamma_0 \geq 100$ gives good results.

8.4 Computing the Solution

The main computational steps in the methods proposed in the previous section are to compute

- the optimal perturbation, p
- the Lagrange multipliers, λ

The Lagrange multipliers are determined by (8.3). To compute p we need to solve least squares problems of the form

$$\min_p \|Ap - b\|$$

A numerically stable and efficient way of solving least squares problems is to use QR decomposition, see Appendix B. If A has full column rank it can be decomposed into

$$A = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix}$$

where Q is orthogonal ($Q^T Q = I$) and R_1 is upper triangular and nonsingular. Since premultiplying with Q^T does not affect the norm we get

$$\|Ap - b\|^2 = \|Rp - Q^T b\|^2 = \|R_1 p - Q_1^T b\|^2 + \|Q_2^T b\|^2$$

The minimum norm is obtained for

$$p = R_1^{-1} Q_1^T b$$

Note that since R_1 is upper triangular we do not need to form R_1^{-1} explicitly, but p can be solved from $R_1 p = Q_1^T b$ by backsubstitution. For convenience, introduce the short hand notation

$$p = A \backslash b$$

for the least squares solution.

Let us now investigate the numerical computations involved in the two methods in some detail.

8.4.1 Sequential Least Squares

First consider Algorithm 8.2.

Phase 1: Optimization problem:

$$\min_u \|Au - b\|$$
$$Cu \geq U$$

where $A = W_v B$, $b = W_v v$, $C = \begin{pmatrix} I \\ -I \end{pmatrix}$ and $U = \begin{pmatrix} \underline{u} \\ -\overline{u} \end{pmatrix}$.

Solution Update:

$$\min_p \|A(u^i + p) - b\|$$
$$p_i = 0, \ i \in \mathcal{W}$$

For convenience, assume that p is partitioned as $p = \begin{pmatrix} p_f \\ 0 \end{pmatrix}$, where p_f are the free variables. Let $\dim p_f = m_f$ and $A = \begin{pmatrix} A_f & A_0 \end{pmatrix}$. This yields

$$\|A(u^i + p) - b\| = \|A_f p_f - d\|$$

where $d = b - Au^i$. For $m_f \leq k$, the unique minimizer given by

$$p_f = A_f \backslash d \tag{8.6}$$

For $m_f > k$, a parameterization of the minimizing solutions can be obtained from the QR decomposition of A_f^T. Note that A_f^T has full column rank if the requirements on W_v and B from Section 8.3.1 are fulfilled.

$$A_f^T = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1$$

along with the change of variables

$$p_f = Qq = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = Q_1 q_1 + Q_2 q_2$$

gives us

$$\left\|A_f p_f - d\right\| = \left\|R_1^T Q_1^T (Q_1 q_1 + Q_2 q_2) - d\right\| = \left\|R_1^T q_1 - d\right\| = 0$$

for

$$q_1 = R_1^{-T} d$$

This means that the set of minimizing solutions can be written as

$$p_f = Q_1 R_1^{-T} d + Q_2 q_2$$

where q_2 is a free variable. Since the columns of Q_1 and Q_2 are orthonormal, $\left\|p_f\right\|$, and thereby also $\left\|p\right\|$, is minimized for

$$q_2 = 0$$

Lagrange Multipliers: Comparing with (8.3) we get

$$A^T(Au - b) = C_0^T \lambda$$

Since $C_0 C_0^T = I$, multiplying by C_0 from the left yields

$$\lambda = C_0 A^T (Au - b) \tag{8.7}$$

Phase 2: Optimization problem:

$$\begin{aligned} \min_u \left\|Au - b\right\| \\ Bu = v \\ Cu \geq U \end{aligned}$$

where $A = W_u$, $b = W_u u_d$, and C and U as defined in phase 1.

Solution Update:

$$\begin{aligned} \min_p \left\|A(u^i + p) - b\right\| \\ Bp = 0 \\ p_i = 0,\ i \in \mathcal{W} \end{aligned} \quad \Longleftrightarrow \quad \begin{aligned} \min_p \left\|Ap - d\right\| \\ Ep = \begin{pmatrix} B \\ C_0 \end{pmatrix} p = 0 \end{aligned}$$

where $d = b - Au^i$. Using the QR decomposition

$$E^T = QR = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1 \tag{8.8}$$

and introducing

$$p = Qq = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} = Q_1 q_1 + Q_2 q_2$$

gives us

$$Ep = R_1^T Q_1^T (Q_1 q_1 + Q_2 q_2) = R_1^T q_1 = 0 \quad \Longleftrightarrow \quad q_1 = 0$$

since R_1 is nonsingular. Inserting $p = Q_2 q_2$ into the optimization criterion yields

$$\|Ap - d\| = \|AQ_2 q_2 - d\| = \min$$

for

$$q_2 = (AQ_2) \backslash d$$

Essentially, this is where our method differs from the minimal least squares method proposed by Lötstedt (1984). Since Lötstedt considers the minimum norm case, corresponding to $A = W_u = I$, the optimal solution is simply given by $q_2 = Q_2^T d$.

Lagrange Multipliers: Comparing with (8.3) we get

$$A^T(Au - b) = E^T \Lambda = Q_1 R_1 \Lambda$$

Premultiplying both sides with $R_1^{-1} Q_1^T$ yields the solution

$$\Lambda = \begin{pmatrix} \mu \\ \lambda \end{pmatrix} = R_1^{-1} Q_1^T A^T (Au - b)$$

from which the relevant Lagrange variables λ can be extracted.

8.4.2 Weighted Least Squares

In Algorithm 8.3 the optimization problem is given by

$$\min_u \|Au - b\|$$
$$Cu \geq U$$

where A and b were defined in Algorithm 8.3, $C = \left(\begin{smallmatrix} I \\ -I \end{smallmatrix}\right)$, and $U = \left(\begin{smallmatrix} \underline{u} \\ -\overline{u} \end{smallmatrix}\right)$. This problem coincides with the phase 1 problem of sequential least squares. Here, since A has full column rank, the unique optimal perturbation in each iteration is determined by (8.6) and the Lagrange multipliers by (8.7).

8.5 Simulations

Let us now evaluate the two proposed methods and compare them with some of the existing alternative methods discussed in Section 7.4.3. We will discuss the quality of the solutions produced, the computation times of the methods, and finally comment on some implementation issues.

8.5.1 Numerical Methods

The methods used in the evaluations are:

SLS	Sequential least squares (Algorithm 8.2)
MLS	Minimal least squares (Lötstedt 1984)
WLS	Weighted least squares (Algorithm 8.3)
WLS2	WLS with a maximum of 2 iterations
CGI	Cascaded generalized inverse method (Bordignon 1996)
FXP	Fixed-point method (Burken et al. 2001) (50 iterations)
LSQLIN	`lsqlin` (MATLAB Optimization Toolbox)

In MLS and FXP, which both solve the weighted problem (8.4), γ was selected as in (8.5) with $\gamma_0 = 100$.

All methods (except LSQLIN) have been implemented as MATLAB M-files which can be downloaded from

`http://www.control.isy.liu.se/~ola/`

The MLS implementation is based on the direct method described in Lötstedt (1984), which uses QR decompositions. QR decompositions are also used to compute the pseudoinverse solutions needed in the CGI method. This was found to be the faster than computing the pseudoinverse matrix explicitly as in (7.27). FXP is implemented according to the description in Section 7.4.3.

LSQLIN, based the standard least squares solver `lsqlin` from the MATLAB Optimization Toolbox, is used as a reference to confirm that the sequential and minimal least squares methods indeed find the optimal solution to (8.1). However, it is only applicable when the virtual control demand is known to be feasible, since `lsqlin` handles problems of the form (8.2).

8.5.2 Aircraft Simulation Data

Aircraft data are taken from the example used by Durham and Bordignon (1996), in which the control effectiveness matrix

$$B = 10^{-4} \times \begin{pmatrix} 253.8 & -253.8 & 274.3 & -274.3 & 19.22 & -250.0 & 250.0 & 4.5 \\ -3801 & -3801 & -349.5 & -349.5 & 0.1681 & 1125 & 1125 & 0 \\ -16.81 & 16.81 & -92.51 & 92.51 & -382.7 & 0 & 0 & -750.0 \end{pmatrix}$$

the position constraints

$$u_{\min} = \begin{pmatrix} -24 & -24 & -25 & -25 & -30 & -30 & -30 & -30 \end{pmatrix}^T \cdot \frac{\pi}{180} \text{ rad}$$

$$u_{\max} = \begin{pmatrix} 10.5 & 10.5 & 42 & 42 & 30 & 30 & 30 & 30 \end{pmatrix}^T \cdot \frac{\pi}{180} \text{ rad}$$

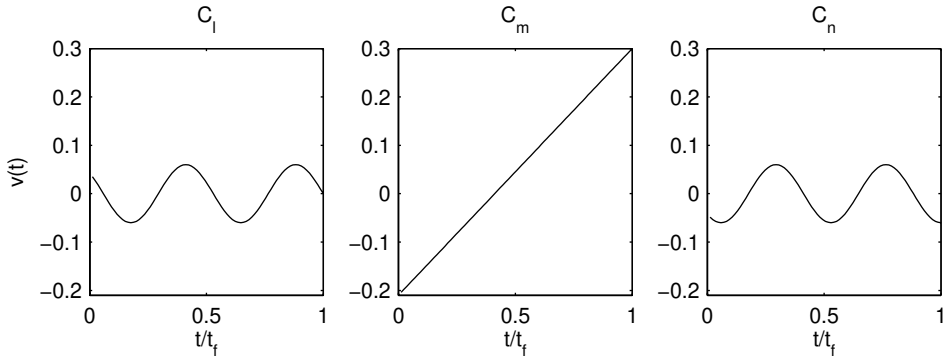


Figure 8.2: Virtual control trajectory $v(t) = \begin{pmatrix} C_l(t) & C_m(t) & C_n(t) \end{pmatrix}^T$ to be produced. For small values of the final time t_f, rate saturation occurs.

and the rate constraints

$$-\rho_{\min} = \rho_{\max} = \begin{pmatrix} 100 & 100 & 100 & 100 & 100 & 100 & 100 \end{pmatrix}^T \cdot \frac{\pi}{180} \text{ rad/s}$$

are loosely based on an F-18 airplane at 10.000 ft, Mach 0.23, and 30° angle of attack.

The 8-dimensional control input u consists of left and right horizontal tails (u_1, u_2), left and right ailerons (u_3, u_4), combined rudders (u_5), left and right thrust vectoring nozzles (u_6, u_7), operating in the same manner as horizontal tails, and yaw thrust vectoring (u_8).

The components of the 3-dimensional virtual control input v are the commanded aerodynamic moment coefficients C_l, C_m, and C_n. The virtual control trajectory, consisting of 85 samples, is shown in Figure 8.2, and corresponds to the helical path with radius 0.06 used in Durham and Bordignon (1996). By varying t_f, the final time of the trajectory, different "helical rates" are achieved. When the rate is too high, some parts of the trajectory become infeasible due to the actuator rate constraints.

8.5.3 Simulation Results

Three different cases were simulated:

Case 1: Feasible trajectory ($t_f = 25$ s, helical rate = 0.045) with W_u and W_v set to identity matrices and u_d set to the zero vector. See Table 8.2 and Figure 8.4 for simulation results.

Case 2: Same trajectory as in case 1 but with

$$W_u = \text{diag}(10, 10, 1, 1, 1, 1, 1, 1) \tag{8.9}$$

This corresponds to penalizing the use of the horizontal tails for control. See Table 8.3 and Figure 8.5.

Case 3: Partially infeasible trajectory ($t_f = 4$ s, helical rate = 0.28) with the same parameter settings as in case 1. See Table 8.4 and Figure 8.6.

In the figures, the SLS results are used to represent WLS also, since the discrepancy is very small.

The simulations were performed in MATLAB 6.1 running on a 800 MHz Pentium III computer. In Tables 8.2–8.4, the `tic` and `toc` commands were used to determine the computation times which were averaged over 1000 runs. The mean and max errors given correspond to the mean and max values of $\|Bu(t) - v(t)\|$ over time. Furthermore, the average number of saturated actuators in the solutions are given and also the average number of iterations performed by each method.

8.5.4 Comments

Solution Quality

All methods produce solutions which satisfy both actuator position and rate constraints, see Figures 8.4–8.6.

By construction, SLS and MLS both generate the exact solution to the control allocation problem formulated as the sequential least squares problem (8.1). WLS only solves an approximation of the original problem, see (8.4), but with the weight γ chosen large enough, as in the simulations, WLS comes very close to recovering the true optimal solution. Note that in case 3, the virtual control errors are actually smaller for WLS than for SLS and MLS. This unexpected result is due to that the actuator bounds at time t depends on $u(t-T)$, see (7.7). Since $u(t-T)$ differs between the methods, so do the bounds, which allows for WLS to perform slightly better on the average.

WLS2 uses at most two iterations, corresponding to at most two changes per sample in the set of active actuator constraints. For the feasible trajectory, WLS2 almost exactly recovers the optimal solution, while for the infeasible trajectory, the solution quality is somewhat degraded. The reason for WLS2 being so successful can be seen from Figure 8.3. In most sampling instants, in particular for the feasible trajectory, WLS (without any restriction on the number of iterations) finds the optimum in only one or two iterations. In those instants where WLS needs three or more iterations, WLS2 only finds a suboptimal solution, but can be thought of as retrieving the correct active set a few sampling instants later. Similar restrictions could also be introduced in SLS and MLS.

The CGI performance seems difficult to predict. In case 1, the CGI solution satisfies $Bu = v$ at each sampling instant although some control surface time histories differ slightly from the optimal ones produced by SLS and MLS. However, in case 2, the CGI solution is rather degraded. The same goes for case 3 where especially the pitching coefficient is poorly reproduced. The general flaw with CGI is the heuristic rule it is based on which claims that it is optimal to saturate all

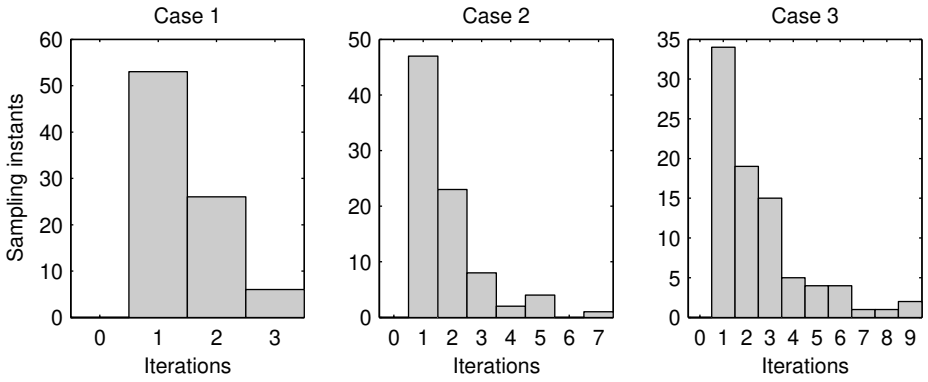


Figure 8.3: Histograms showing the number of iterations performed by WLS for the three simulated cases. In all cases, it holds that in less than half of the sampling instants more than two iterations are required to find the optimal solution.

control surfaces that violate their bounds in some iteration. Note that CGI yields the highest average number of actuator saturations in all cases.

In all of the simulated cases, FXP generates rather poor, although continuous, solutions to the control allocation problem. The FXP method relies on gradient search methods to update the solution. Gradient search methods are the most successful when the level curves of the cost function are circular so that the gradient points towards the minimum. Here, however, the level curves are stretched out to ellipsoids due to the scaling of the two objectives in (8.4). This also explains the degradation in quality in case 2 where further weighting is introduced. Changing γ ($10 \leq \gamma \leq 1000$) had no significant effect on the FXP results. Increasing the number of iterations yields better results, but even for $N = 1000$ iterations, FXP is outperformed by WLS in all test cases, in terms of the mean virtual control error.

Computation Time

Overall, the computation time for the different methods (except LSQLIN) are within the same order of magnitude. Thus, if CGI is considered a viable alternative for real-time applications, then so should SLS, MLS, and definitely WLS be. For comparison, the typical sampling frequency in modern aircraft is 50-100 Hz which corresponds to a sampling time of 10-20 ms.

WLS2 and FXP both have a maximal computation time which by construction is independent of the trajectory type. This is appealing since it makes the control allocation task easy to schedule in a real-time implementation.

Table 8.1 shows the benefits of warmstarting the active set methods. When $u(t-T)$ is ignored and $u^0 = \big(\underline{u}(t) + \overline{u}(t)\big)/2$ is used instead, the mean SLS computation time increases by 45-210 % (85-245 % for WLS). The increase in the maximum computation time is less dramatic, 20-40 % for SLS (50-60 % for WLS).

	SLS		WLS	
Case	Mean time (ms)	Max time (ms)	Mean time (ms)	Max time (ms)
1	1.12 (1.62)	2.44 (2.91)	0.52 (0.97)	1.03 (1.65)
2	1.30 (2.45)	3.41 (4.69)	0.60 (1.45)	1.83 (2.80)
3	0.80 (2.51)	3.28 (4.31)	0.71 (2.45)	2.30 (3.54)

Table 8.1: The benefits of warmstarting the SLS and WLS methods. The numbers correspond to reusing the solution from the previous sample, and ignoring it, respectively (the latter numbers in parentheses).

Implementation

FXP is the most straightforward method to implement, consisting mainly of matrix multiplications and saturations, see Section 7.4.3.

CGI is also easy to implement and relies heavily on computing pseudoinverse solutions to underdetermined least squares problems, which in the implementation is done using QR decomposition.

SLS, MLS, and WLS are active set based methods and as such they all require bookkeeping and maintenance of the working set of active constraints. Solving constrained least squares problems is also central to active set methods and in the implementation this is done using QR decomposition.

8.6 Conclusions

The main conclusion that can be drawn from our investigations is that classical active set methods seem well suited for the solving l_2-optimal control allocation problem (8.1).

First of all, the problem structure lends itself to an efficient active set implementation. The output from the previous sampling instant can be used as a starting point for the optimization, the constraints are simple box constraints, and the problem size is small. Also, feasible suboptimal solutions are produced in each iteration, which can be used if not enough time is available to compute the optimal solution.

Second, active set methods are closely related to the approximate pseudoinverse methods that are often used today in aircraft control allocation. Instead of using heuristics to decide which actuators should saturate, active set methods rely on the theory of constrained programming.

Further, our simulations show that the timing properties of the proposed methods proposed are similar to those of the comparison methods—the cascaded generalized inverse method of Bordignon (1996) and the fix-point method of Burken et al. (2001)—and that the solutions produced are in general of the same or better quality.

Algo-rithm	Mean time (ms)	Max time (ms)	Mean error	Max error	Mean satur.	Mean iter.
SLS	1.12	2.44	0	0	1.8	2.5
MLS	1.01	1.83	0	0	1.8	2.5
WLS	0.52	1.03	6.14e-4	1.08e-3	1.7	1.4
WLS2	0.51	0.84	7.54e-4	7.53e-3	1.7	1.4
CGI	0.91	1.50	0	0	2.0	2.4
FXP	1.97	2.01	1.38e-2	2.74e-2	1.6	50.0
LSQLIN	15.04	18.57	0	0	1.8	–

Table 8.2: Control allocation results for case 1.

Algo-rithm	Mean time (ms)	Max time (ms)	Mean error	Max error	Mean satur.	Mean iter.
SLS	1.30	3.41	0	0	2.9	2.9
MLS	1.12	2.75	0	0	2.9	2.8
WLS	0.60	1.83	5.38e-5	3.31e-4	2.9	1.8
WLS2	0.56	0.81	1.24e-3	2.13e-2	3.2	1.6
CGI	1.09	1.84	1.11e-2	9.03e-2	4.3	3.0
FXP	1.98	2.01	2.68e-2	6.10e-2	2.5	50.0
LSQLIN	16.84	22.00	0	0	2.8	–

Table 8.3: Control allocation results for case 2.

Algo-rithm	Mean time (ms)	Max time (ms)	Mean error	Max error	Mean satur.	Mean iter.
SLS	0.80	3.28	4.51e-3	1.25e-2	6.1	2.5
MLS	0.89	2.52	4.51e-3	1.25e-2	6.1	2.5
WLS	0.71	2.30	3.09e-3	1.15e-2	5.4	2.5
WLS2	0.56	0.81	1.24e-3	2.13e-2	3.2	1.6
CGI	0.94	1.46	3.00e-2	1.09e-1	7.6	2.6
FXP	1.97	2.01	1.35e-2	2.43e-2	3.1	50.0

Table 8.4: Control allocation results for case 3.

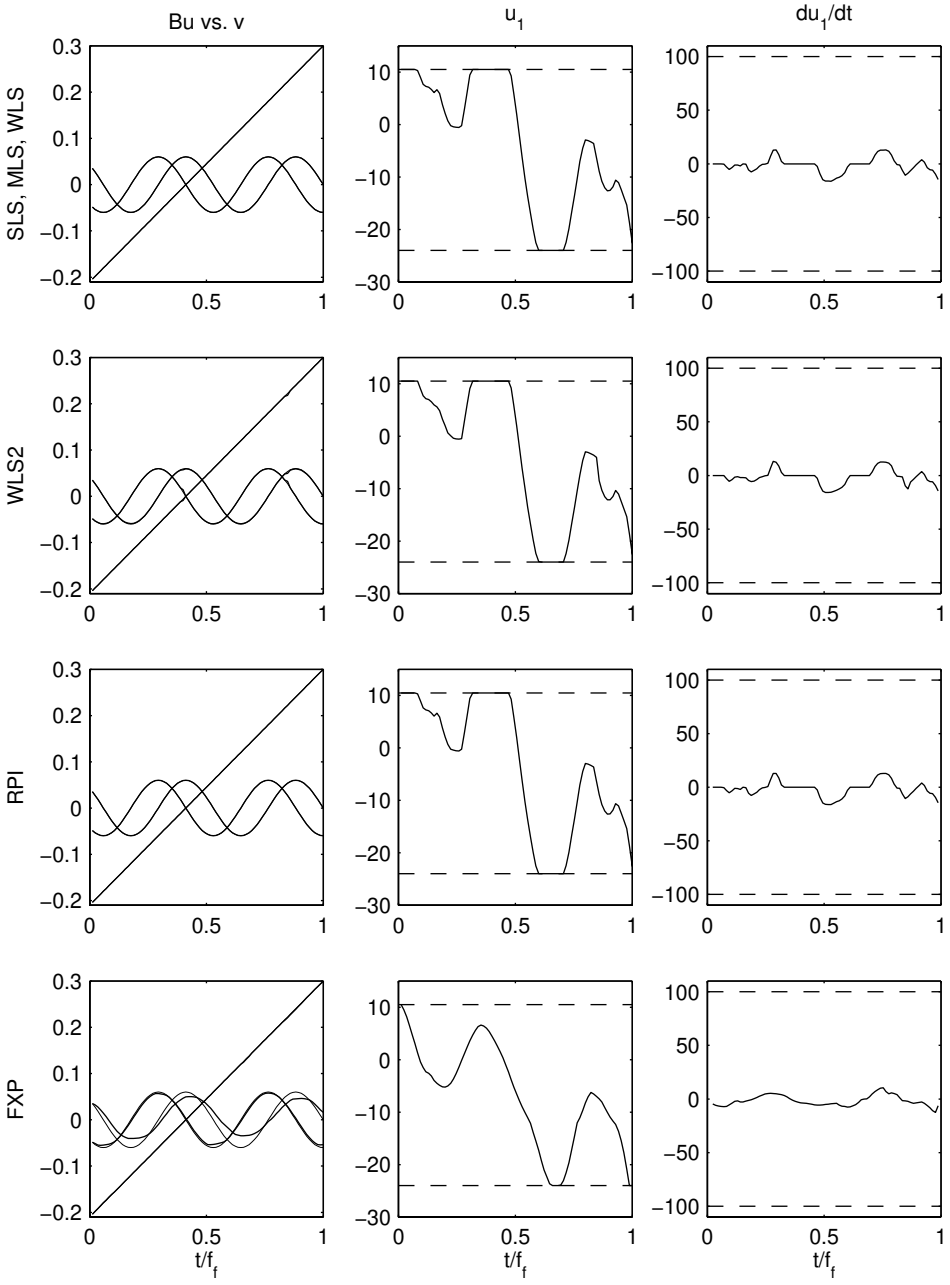


Figure 8.4: Simulation results for case 1. Left: Commanded (thin lines) vs. generated (thick lines) virtual control inputs. Middle and right: Left horizontal tail (u_1) position and rate. The dashed lines mark the actuator constraints.

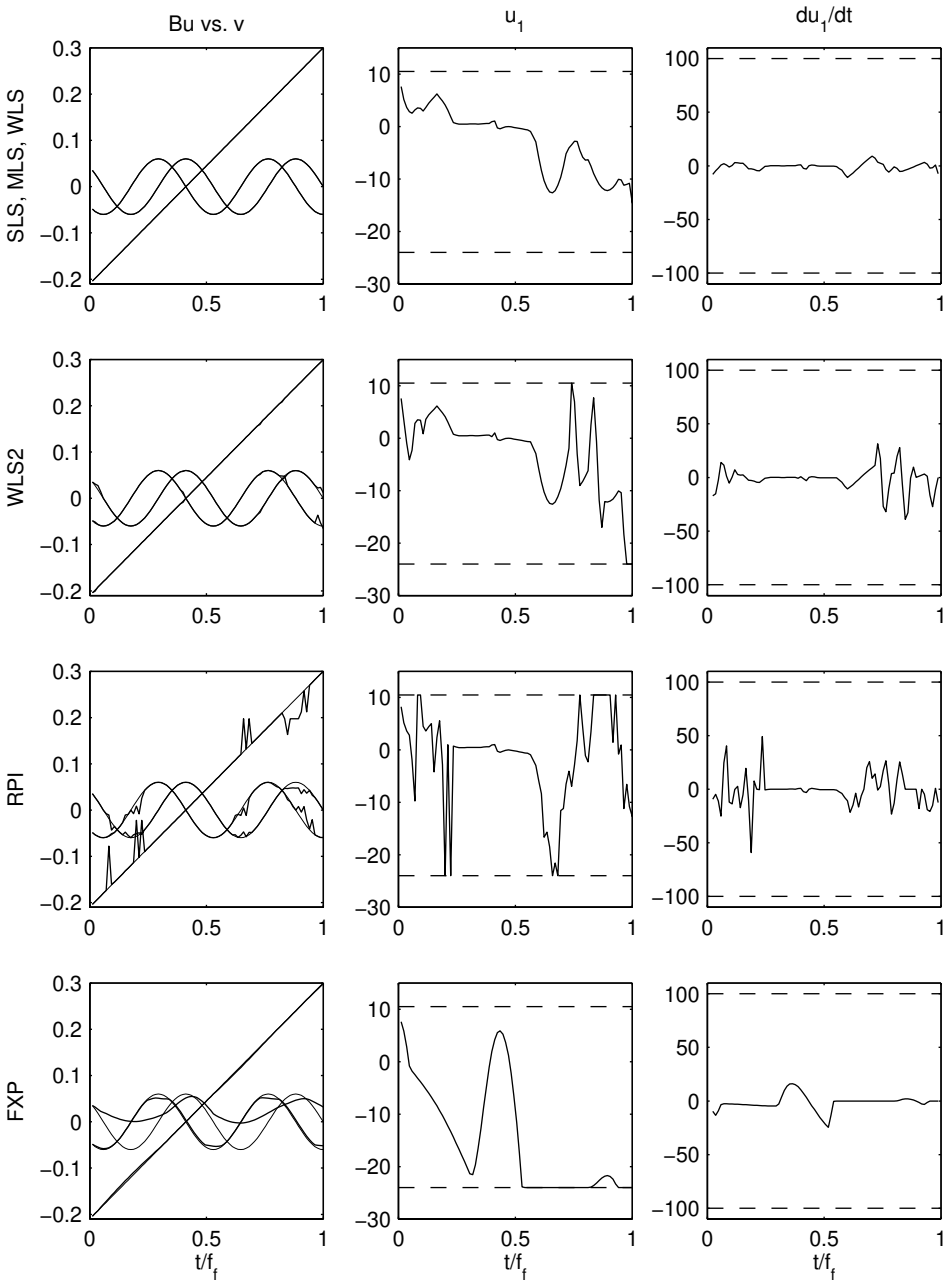


Figure 8.5: Simulation results for case 2. Left: Commanded (thin lines) vs. generated (thick lines) virtual control inputs. Middle and right: Left horizontal tail (u_1) position and rate. The dashed lines mark the actuator constraints.

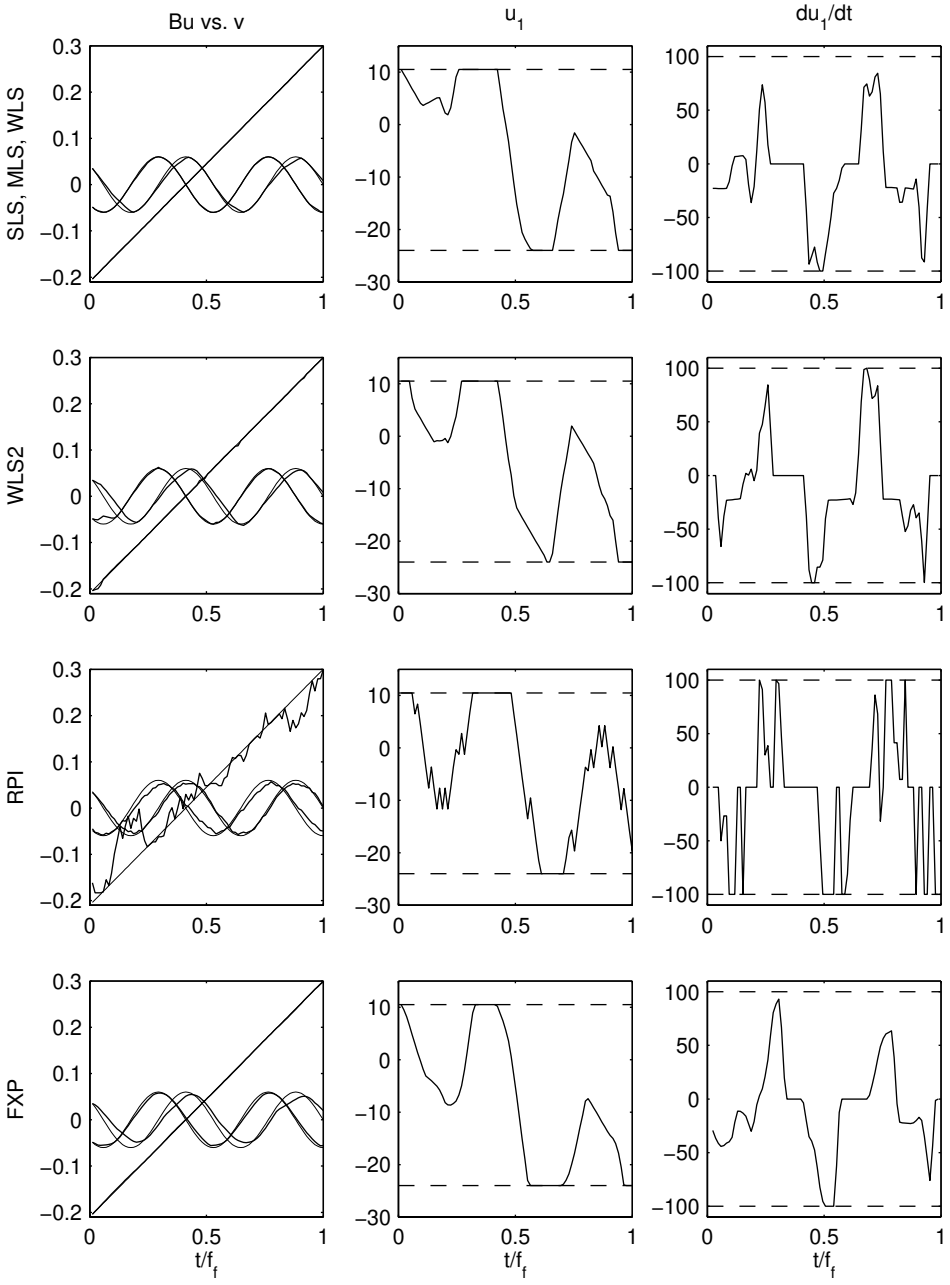


Figure 8.6: Simulation results for case 3. Left: Commanded (thin lines) vs. generated (thick lines) virtual control inputs. Middle and right: Left horizontal tail (u_1) position and rate. The dashed lines mark the actuator constraints.

Chapter 9

Dynamic Control Allocation

The foundation of control allocation is that there are several ways to apportion the control among the actuators, all of which make the system behave the same way. This design freedom is often used to optimize some static performance index, like minimum control, or to prioritize among the actuators. This can be thought of as affecting the distribution of control effect in magnitude among the actuators. Regardless of method (optimization based allocation, daisy chain allocation, direct allocation, etc.), the resulting mapping from virtual to true control input can be written as a static relationship

$$u(t) = f\big(v(t)\big)$$

A possibility that has been little explored is to also affect the distribution of the control effect in the frequency domain, and use the redundancy to have different actuators operate in different parts of the frequency spectrum. This requires the mapping from v to u to depend also on previous values of u and v, hence

$$u(t) = f\big(v(t), u(t-T), v(t-T), u(t-2T), v(t-2T), \dots\big)$$

where T is the sampling interval. We will refer to this as *dynamic control allocation*.

Some practical examples where filtering has been introduced in the control allocation can be found in the literature. Papageorgiou et al. (1997) consider a flight control case, where canards and tailerons are available for pitch control. To achieve a fast initial aircraft response, and to make use of the fast dynamics of the canards, the high frequency component of the required pitching moment is fed to the canards while the remaining low frequency component is fed to the tailerons, which are used solely at trimmed flight.

Another flight control example can be found in Reiner et al. (1996), in which thrust vectored control (TVC) is available. To prevent the TVC vanes from suffering thermal damage from the jet exhaust, the TVC deflection command is fed to a wash-out filter (static gain zero), so that the vanes do not remain deflected on the exhaust for long periods of time.

Berge and Fossen (1997) consider control of marine vessels, equipped with azimuth (rotatable) thrusters. Essentially, the authors use the low frequency component of the total thrust demand to decide the azimuth angles, which are then used to compute the force to be produced by each thruster.

Davidson et al. (2001) use rate saturation problems in flight control as a motivation for dynamic control allocation, or frequency-apportioned control allocation, as the authors call it. The high and low frequency components of the moment demand are each multiplied by a weighted pseudoinverse of the control effectiveness matrix, B, with the weights based on the rate and position bounds of the actuators, respectively. With this strategy, fast actuators are used for high frequency control, and the chances of rate saturation are reduced.

Hence, there are practical cases where dynamic control allocation is desirable. In this chapter, a new systematic method for dynamic control allocation is proposed. The method is an extension of regular l_2-optimal control allocation with an extra term to penalize actuator rates. When no saturations occur, the control allocation mapping becomes a linear filter of the form

$$u(t) = Fu(t-T) + Gv(t)$$

The frequency characteristics of this filter are decided by the weighting matrices used in the optimization criterion. Thus, unlike the previous methods, no filters are to be explicitly constructed by the control designer.

The chapter is organized as follows. In Section 9.1, the proposed method is presented. The static and dynamic properties of the resulting dynamic control allocator are investigated in Section 9.2. Section 9.3 contains a flight control design example illustrating the potential benefits of the method. Finally, conclusions are drawn in Section 9.4.

This chapter is based on

> O. Härkegård. Dynamic control allocation using constrained quadratic programming. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002a.

9.1 Dynamic Control Allocation Using QP

The dynamic control allocation method that we propose fits into the standard framework for optimization based control allocation presented in Section 7.3.1. The sequential optimization problem to be solved is given by

$$u(t) = \arg\min_{u(t)\in\Omega} \left\|W_1(u(t)-u_s(t))\right\|^2 + \left\|W_2(u(t)-u(t-T))\right\|^2 \tag{9.1a}$$

$$\Omega = \underset{\underline{u}(t)\le u(t)\le \overline{u}(t)}{\arg\min} \left\|W_v(Bu(t)-v(t))\right\| \tag{9.1b}$$

where $u \in \mathbb{R}^m$ is the true control input, $u_s \in \mathbb{R}^m$ is the desired steady state control input, $v \in \mathbb{R}^k$ is the virtual control input, $B \in \mathbb{R}^{k\times m}$ is the control effectiveness

matrix, and W_1, W_2, and W_v are square matrices of the proper dimensions. $\|\cdot\| = \|\cdot\|_2$ denotes the l_2-norm, see (7.17). Given Ω, the set of feasible control inputs (with respect to position and rate constraints as in (7.7)) that minimize the virtual control error (weighted by W_v), we pick the control input that minimizes the cost function in (9.1a).

What is new in this formulation, compared to (7.18), is that not only the position error $u(t)-u_s(t)$, but also the change in the control input, $u(t)-u(t-T)$, is penalized in the cost function in (9.1a). This corresponds to penalizing the actuator rates. Hodel and Callahan (2002) pose almost the exact same problem, but without motivating why the rate term should be included, and without analyzing the effects of including this term.

In (9.1a), $u_s(t)$ is the desired steady state distribution of control effort among the actuators. We will discuss the choice of u_s in Section 9.2.3. W_1 and W_2 are weighting matrices whose (i,i)-entries specify whether it is important for the i:th actuator, u_i, to quickly reach its desired stationary value, or to change its position as little as possible. With this interpretation, a natural choice is to use diagonal weighting matrices but in the analysis to follow we will allow arbitrary matrices with the following restriction.

Assumption 9.1 *Assume that the weighting matrices W_1 and W_2 are symmetric and such that*

$$W = \sqrt{W_1^2 + W_2^2}$$

is nonsingular.

See Appendix B for a definition of the matrix square root. This assumption certifies that there is a unique optimal solution to the control allocation problem (9.1), see Section 7.3.1.

For computational purposes, the two terms in (9.1a) can be merged into one term without affecting the solution, see Lemma 9.1. Thus, any solver suitable for real-time implementation, such as the CGI and FXP methods in Section 7.4.3, or the active set methods developed in Chapter 8, can be used to compute the solution to (9.1).

9.2 The Nonsaturated Case

How do the design variables, u_s, W_1, and W_2, affect the solution, $u(t)$? To answer this question, let us investigate the case where no actuators saturate in the solution to (9.1). Then the actuator constraints can be disregarded and (9.1) reduces to

$$\min_{u(t)} \quad \left\|W_1(u(t)-u_s(t))\right\|^2 + \left\|W_2(u(t)-u(t-T))\right\|^2 \tag{9.2a}$$

$$\text{subject to} \quad Bu(t) = v(t) \tag{9.2b}$$

9.2.1 Explicit Solution

Let us begin by stating the closed form solution to (9.2).

Theorem 9.1 *Let Assumption 9.1 hold. Then the control allocation problem* (9.2) *has the solution*

$$u(t) = Eu_s(t) + Fu(t-T) + Gv(t) \tag{9.3}$$

where

$$\begin{aligned} E &= (I - GB)W^{-2}W_1^2 \\ F &= (I - GB)W^{-2}W_2^2 \\ G &= W^{-1}(BW^{-1})^\dagger \end{aligned}$$

The † symbol denotes the pseudoinverse operator defined as

$$A^\dagger = A^T(AA^T)^{-1}$$

for a $k \times m$ matrix A with rank $k \leq m$, see Appendix B.

The theorem shows that the optimal solution to the control allocation problem (9.2) is given by the first order linear filter (9.3). The properties of this filter will be further investigated in Sections 9.2.2 and 9.2.3.

To prove the theorem, the following lemma is useful.

Lemma 9.1 *Let Assumption 9.1 hold. Then the cost function*

$$\left\|W_1(x - x_1)\right\|^2 + \left\|W_2(x - x_2)\right\|^2$$

has the same minimizing argument as

$$\left\|W(x - x_0)\right\|$$

where

$$\begin{aligned} W &= \sqrt{W_1^2 + W_2^2} \\ x_0 &= W^{-2}(W_1^2 x_1 + W_2^2 x_2) \end{aligned}$$

Proof:

$$\begin{aligned} &\left\|W_1(x - x_1)\right\|^2 + \left\|W_2(x - x_2)\right\|^2 = \\ &(x - x_1)^T W_1^2 (x - x_1) + (x - x_2)^T W_2^2 (x - x_2) = \\ &x^T(W_1^2 + W_2^2)x - 2x^T(W_1^2 x_1 + W_2^2 x_2) + \ldots = \\ &(x - x_0)^T W^2 (x - x_0) + \ldots = \left\|W(x - x_0)\right\|^2 + \ldots \end{aligned}$$

where

$$\begin{aligned} W &= \sqrt{W_1^2 + W_2^2} \\ x_0 &= W^{-2}(W_1^2 x_1 + W_2^2 x_2) \end{aligned}$$

The terms represented by dots do not depend on x, and hence they do not affect the minimization. □

We now state the proof of Theorem 9.1.

Proof: From Lemma 9.1 we know that the optimization criterion (9.2a) can be rewritten as

$$\min_{u(t)} \left\| W(u(t) - u_0(t)) \right\|$$

where

$$\begin{aligned} W &= \sqrt{W_1^2 + W_2^2} \\ u_0(t) &= W^{-2}(W_1^2 u_s(t) + W_2^2 u(t-T)) \end{aligned}$$

Applying Lemma B.1 to this criterion constrained by (9.2b) yields

$$\begin{aligned} u(t) &= \bar{F}u_0(t) + Gv(t) \\ \bar{F} &= I - GB \\ G &= W^{-1}(BW^{-1})^\dagger \end{aligned}$$

from which it follows that

$$u(t) = \underbrace{(I - GB)W^{-2}W_1^2}_{E} u_s(t) + \underbrace{(I - GB)W^{-2}W_2^2}_{F} u(t-T) + Gv(t)$$

which completes the proof. □

To analyze the filter (9.3), for example by plotting its frequency characteristics in a Bode diagram, it is convenient to describe it in state space form.

Corollary 9.1 *The linear filter (9.3) can be written in state space form as*

$$\begin{aligned} x(t+T) &= Fx(t) + F(Eu_s(t) + Gv(t)) \\ u(t) &= x(t) + Eu_s(t) + Gv(t) \end{aligned}$$

Proof: Introducing the shift operator q, such that $qu(t) = u(t+T)$, (9.3) becomes

$$u(t) = q^{-1}Fu(t) + \underbrace{Eu_s(t) + Gv(t)}_{w(t)}$$

Multiplying by q and solving for $u(t)$ yields

$$\begin{aligned} u(t) &= (qI - F)^{-1}qw(t) = (qI - F)^{-1}(qI - F + F)w(t) \\ &= (I + (qI - F)^{-1}F)w(t) \end{aligned}$$

Comparing with the standard state space form to transfer function relationship

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t) \end{aligned} \quad \leftrightarrow \quad y(t) = (C(qI - A)^{-1}B + D)u(t)$$

gives us the state space description

$$\begin{aligned} x(t+1) &= Fx(t) + Fw(t) = Fx(t) + F(Eu_s(t) + Gv(t)) \\ u(t) &= x(t) + w(t) = x(t) + Eu_s(t) + Gv(t) \end{aligned}$$ □

9.2.2 Dynamic Properties

Let us now study the dynamic properties of the filter (9.3). Note that the optimization criterion in (9.2) does not consider future values of $u(t)$. It is therefore not obvious that the resulting filter (9.3) is stable. The poles of the filter, which can be found as the eigenvalues of the feedback matrix F, are characterized by the following theorem.

Theorem 9.2 *Let F be defined as in Theorem 9.1 and let Assumption 9.1 hold. Then the eigenvalues of F, $\lambda(F)$, satisfy*

$$0 \leq \lambda(F) \leq 1$$

If W_1 is nonsingular, the upper eigenvalue limit becomes strict, i.e.,

$$0 \leq \lambda(F) < 1$$

Proof: We wish to characterize the eigenvalues of

$$\begin{aligned} F &= (I - GB)W^{-2}W_2^2 \\ &= (I - W^{-1}(BW^{-1})^\dagger B)W^{-2}W_2^2 \\ &= W^{-1}(I - (BW^{-1})^\dagger BW^{-1})W^{-1}W_2^2 \end{aligned} \tag{9.4}$$

Let the singular value decomposition (see Appendix B) of BW^{-1} be given by

$$BW^{-1} = U\Sigma V^T = U \begin{bmatrix} \Sigma_r & 0 \end{bmatrix} \begin{bmatrix} V_r^T \\ V_0^T \end{bmatrix} = U\Sigma_r V_r^T$$

where U and V are orthogonal matrices, and Σ_r is a $k \times k$ diagonal matrix with strictly positive diagonal entries (since BW^{-1} has rank k). This yields

$$I - (BW^{-1})^\dagger BW^{-1} = I - V_r\Sigma_r^{-1}U^T U\Sigma_r V_r^T = I - V_r V_r^T = V_0 V_0^T$$

The last step follows from the fact that $VV^T = V_r V_r^T + V_0 V_0^T = I$. Inserting this into (9.4) gives us

$$F = W^{-1}V_0 V_0^T W^{-1}W_2^2$$

Now use the fact that the nonzero eigenvalues of a matrix product AB, $\lambda_{nz}(AB)$, satisfy $\lambda_{nz}(AB) = \lambda_{nz}(BA)$, see Zhang (1999, p. 51), to get

$$\lambda_{nz}(F) = \lambda_{nz}(V_0^T W^{-1}W_2^2 W^{-1}V_0)$$

From the definition of singular values in Appendix B we get

$$\lambda(V_0^T W^{-1} W_2^2 W^{-1} V_0) = \sigma^2(W_2 W^{-1} V_0) \geq 0$$

This shows that the nonzero eigenvalues of F are real and positive and thus, $\lambda(F) \geq 0$ holds.

What remains to show is that the eigenvalues of F are bounded by 1. To do this we investigate the maximum eigenvalue, $\overline{\lambda}(F)$.

$$\overline{\lambda}(F) = \overline{\sigma}^2(W_2 W^{-1} V_0) = \left\| W_2 W^{-1} V_0 \right\|^2 \leq \left\| W_2 W^{-1} \right\|^2 \left\| V_0 \right\|^2$$

Since

$$\left\| V_0 \right\|^2 = \overline{\lambda}(\underbrace{V_0^T V_0}_{I}) = 1$$

we get

$$\overline{\lambda}(F) \leq \left\| W_2 W^{-1} \right\|^2 = \sup_{x \neq 0} \frac{x^T W^{-1} W_2^2 W^{-1} x}{x^T x}$$

Introducing $y = W^{-1}x$ yields

$$\overline{\lambda}(F) \leq \sup_{y \neq 0} \frac{y^T W_2^2 y}{y^T W^2 y} = \sup_{y \neq 0} \frac{y^T W_2^2 y}{y^T W_1^2 y + y^T W_2^2 y} \leq \sup_{y \neq 0} \frac{y^T W_2^2 y}{y^T W_2^2 y} = 1 \tag{9.5}$$

since $y^T W_1^2 y = \left\| W_1 y \right\|^2 \geq 0$ for any symmetric W_1. If W_1 is nonsingular, we get $y^T W_1^2 y = \left\| W_1 y \right\|^2 > 0$ for $y \neq 0$ and the last inequality in (9.5) becomes strict, i.e., $\overline{\lambda}(F) < 1$ in this case. □

The theorem states that the poles of the linear control allocation filter (9.3) lie between 0 and 1 on the real axis. This has two important practical implications:

- If W_1 is nonsingular the filter poles lie strictly inside the unit circle. This implies that the filter is asymptotically stable. W_1 being nonsingular means that all actuator positions except $u = u_s$ render a nonzero cost in (9.1a). If W_1 is singular, only marginal stability can be guaranteed (although asymptotic stability may hold).

- The fact that the poles lie on the positive real axis implies that the actuator responses to a step in the virtual control input are not oscillatory.

9.2.3 Steady State Properties

In the previous section we showed that the control allocation filter (9.3) is asymptotically stable under practically reasonable assumptions. Let us therefore investigate the steady state solution for a constant virtual control input.

Theorem 9.3 *Let u_s satisfy*

$$Bu_s = v_0$$

where $v(t) = v_0$ is the desired virtual control input. Then, if W_1 is nonsingular, the steady state control distribution of (9.3) is given by

$$\lim_{t\to\infty} u(t) = u_s$$

Proof: If W_1 is nonsingular, the linear filter (9.3) is asymptotically stable according to Theorem 9.2. This means that in the limit, $u(t) = u(t-T)$ holds. Then (9.2) reduces to

$$\begin{aligned} \min_{u} \quad & \left\|W_1(u-u_s)\right\|^2 \\ \text{subject to} \quad & Bu = v_0 \end{aligned} \tag{9.6}$$

If u_s satisfies $Bu_s = v_0$, then $u = u_s$ is obviously one optimal solution to (9.6). Further, if W_1 is nonsingular, $u = u_s$ is the unique optimal solution. □

The theorem essentially states that if $u = u_s$ can be achieved, it will be achieved, eventually. If we feed our dynamic control allocation scheme with a feasible, desirable control distribution, u_s, which solves $Bu_s = v$, the filter (9.3) will render this distribution at steady state. If $Bu_s \neq v$, the resulting steady state control distribution will depend not only on u_s, but also on W_1. This is undesirable since the role of the design parameter W_1 then becomes unclear.

So how do we find a good feasible steady state solution? In simple cases, we may be able to do it by hand but for larger cases the following method can be applied. Pick u_s as the solution to the static control allocation problem

$$\begin{aligned} \min_{u_s} \quad & \left\|W_s(u_s-u_d)\right\| \\ \text{subject to} \quad & Bu_s = v \end{aligned} \tag{9.7}$$

Here, u_d represents some fixed preferred, but typically infeasible control distribution, which, e.g., would give minimum drag. In the simplest case, with $W_s = I$ and $u_d = 0$, we get the pseudoinverse solution $u_s = B^\dagger v$.

In certain cases, some steady state actuator positions should be scheduled with, e.g., speed and altitude, rather than depend on v. This can be handled by introducing additional equality constraints

$$u_{s,i} = u_{p,i} \tag{9.8}$$

for those actuators i whose steady state positions have been predetermined. The optimal solution to (9.7)–(9.8) can be found using Lemma B.1.

9.3 Design Example

Let us now illustrate how to use the proposed design method for dynamic control allocation. The ADMIRE model (ADMIRE ver. 3.4h 2003), described in Section 2.4

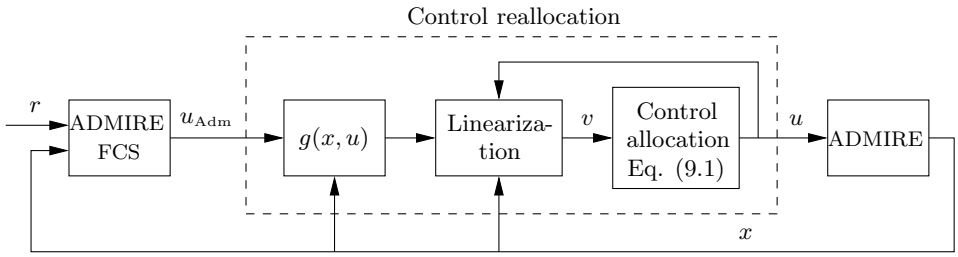


Figure 9.1: Overview of the closed loop system used for simulation. The control input produced by the ADMIRE flight control system is reallocated using dynamic control allocation.

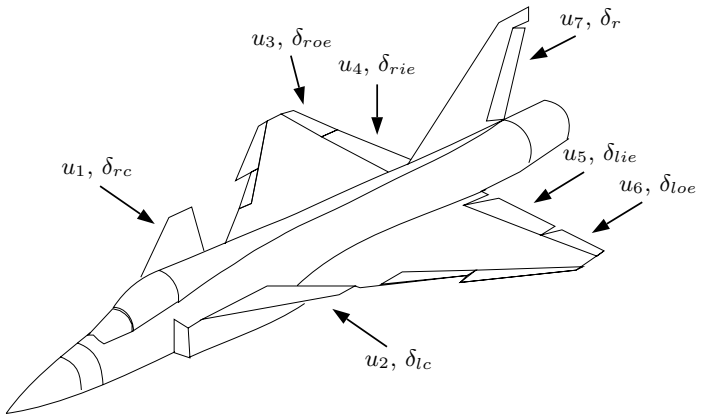


Figure 9.2: ADMIRE control surface configuration. u_i are the commanded deflections and δ_* are the actual deflections.

is used for simulation. ADMIRE is a Simulink based model of a fighter aircraft, and includes actuator dynamics and nonlinear aerodynamics. The existing ADMIRE flight control system is used to compute the aerodynamic moment coefficients, $g(x, u_{\text{Adm}})$, to be produced in roll, pitch, and yaw, see Figure 9.1. The control allocator then solves (9.1) for the commanded control surface deflections, u.

Since the $g(x, u)$ is nonlinear, the linearization techniques from Section 7.2.2 are used. The control effectiveness matrix B is recomputed at each sampling instant by linearizing $g(x, u)$ around the current measurement vector, $x(t)$, and the previous control input, $u(t - T)$. In the ADMIRE model, $T = 0.02$ s. The constrained least squares problem (9.1) is solved at each sampling instant using the sequential least squares solver from Chapter 8 (Algorithm 8.2).

The control input,

$$u = \begin{pmatrix} u_1 & \dots & u_7 \end{pmatrix}^T$$

consists of the commanded deflections for the canard wings (left and right), the elevons (inboard and outboard, left and right), and for the rudder, in radians, see Figure 9.2 where δ_* denote the actual actuator positions. The actuator position and rate constraints are given by

$$\begin{aligned} \delta_{\min} &= \begin{pmatrix} -55 & -55 & -30 & -30 & -30 & -30 & -30 \end{pmatrix}^T \text{deg} \\ \delta_{\max} &= \begin{pmatrix} 25 & 25 & 30 & 30 & 30 & 30 & 30 \end{pmatrix}^T \text{deg} \\ -\rho_{\min} = \rho_{\max} &= \begin{pmatrix} 50 & 50 & 150 & 150 & 150 & 150 & 100 \end{pmatrix}^T \text{deg/s} \end{aligned}$$

At trimmed flight at Mach 0.5, 1000 m, the control effectiveness matrix is given by

$$B = 10^{-2} \times \begin{pmatrix} 0.5 & -0.5 & -4.9 & -4.3 & 4.3 & 4.9 & 2.4 \\ 8.8 & 8.8 & -8.4 & -13.8 & -13.8 & -8.4 & 0 \\ -1.7 & 1.7 & -0.5 & -2.2 & 2.2 & 0.5 & -8.8 \end{pmatrix} \text{rad}^{-1}$$

from which it can be seen, e.g., that the inboard elevons are the most effective actuators for producing pitching moment while the rudder provides good yaw control, as expected. This is the B-matrix used in the design and analysis of the control allocation filter below.

Let us now consider the requirements regarding the control distribution. At trimmed flight, it is beneficial not to deflect the canards at all to achieve minimum drag. We therefore select the steady state distribution u_s as the solution to

$$\begin{aligned} \min_{u_s} \quad & \|u_s\| \\ \text{subject to} \quad & Bu_s = v \\ & u_{s,1} = u_{s,2} = 0 \end{aligned} \tag{9.9}$$

which yields

$$u_s(t) = Sv(t)$$

where

$$S = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -5.4 & -1.6 & -0.4 \\ -4.6 & -2.6 & -2.4 \\ 4.6 & -2.6 & 2.4 \\ 5.4 & -1.6 & 0.4 \\ 3.0 & 0 & -10.1 \end{pmatrix}$$

During the initial phase of a pitch maneuver, on the other hand, utilizing the canards counteracts the unwanted nonminimum phase tendencies that the pilot load factor, n_{zp}, typically displays. In the computation of n_{zp}, defined in (2.18), we will assmue that the pilot is positioned 1 m in front of the aircraft center of gravity. Thus, the canards should be used to realize parts of the high frequency content of the pitching moment. Selecting

$$W_1 = \text{diag}(2, 2, 2, 2, 2, 2, 2)$$
$$W_2 = \text{diag}(5, 5, 10, 10, 10, 10, 10)$$

with the lowest rate penalty on the canards, and using Theorem 9.1, yields the control allocation filter

$$u(t) = Fu(t - T) + G_{\text{tot}}v(t)$$

where

$$F = 10^{-1} \times \begin{pmatrix} 5.5 & -1.4 & 2.9 & 3.4 & 4.3 & 1.8 & -5.0 \\ -1.4 & 5.5 & 1.8 & 4.3 & 3.4 & 2.9 & 5.0 \\ 0.7 & 0.4 & 6.4 & -3.4 & 1.3 & 1.9 & 0.7 \\ 0.9 & 1.1 & -3.4 & 5.5 & 0.7 & 1.3 & -0.8 \\ 1.1 & 0.9 & 1.3 & 0.7 & 5.5 & -3.4 & 0.8 \\ 0.4 & 0.7 & 1.9 & 1.3 & -3.4 & 6.4 & -0.7 \\ -1.3 & 1.3 & 0.7 & -0.8 & 0.8 & -0.7 & 2.2 \end{pmatrix}$$

$$G_{\text{tot}} = G + ES = \begin{pmatrix} 1.7 & 2.8 & -5.3 \\ -1.7 & 2.8 & 5.4 \\ -5.3 & -0.8 & -0.6 \\ -4.7 & -1.3 & -2.2 \\ 4.7 & -1.3 & 2.2 \\ 5.3 & -0.8 & 0.6 \\ 2.4 & 0 & -8.2 \end{pmatrix}$$

in the nonsaturated case. Let us now evaluate the properties of this filter.

The eigenvalues of F are given by

$$\lambda(F) = 0,\ 0,\ 0,\ 0,\ 0.88,\ 0.91,\ 0.96,\ 0.96$$

which is in agreement with Theorem 9.2.

The frequency characteristics of the filter are illustrated in Figure 9.3, which shows a magnitude plot of the transfer functions from v to u, calculated using Corollary 9.1. Each transfer function has been weighted with its corresponding entry in B to show the proportion of v that the actuator produces. In roll, the elevons produce most of the control needed while in pitch, the canards contribute

substantially at high frequencies. Yaw control is produced almost exclusively by the rudder.

Figure 9.5 shows the simulation results from a pitch up command followed by a roll command. In accordance with the designed frequency distributions, the canard wings react quickly to the pitch command while at steady state only the elevons and the rudder are deflected.

Note that the discrepancies between p and p^{ref}, and between q and q^{ref}, are due to the design of the ADMIRE control system and not the choice of control allocation algorithm. Figure 9.6 shows the simulation results when the original ADMIRE flight control system is used, and no control reallocation is performed. The control signals differ in the two cases, but the output signals are almost the same.

It is interesting to compare the designed dynamic control allocator with two static alternatives. Figure 9.7 shows the simulation results when $W_2 = 0$. In this case, we get the distribution $u(t) = u_s(t)$ when no saturations occur. In Figure 9.7 the canard deflections are zero except around $t = 3$ when they are needed because the elevons saturate.

Figure 9.8 show the results when $W_2 = 0$ and $u_s = 0$. Then the control allocation objective (9.1a) becomes the often used min $\|u\|$ criterion, since $W_1 = 2I$. Figure 9.3 shows the frequency characteristics of the control allocator in this case. Since the actuator rates are not included in the control allocation objective, no dynamics are introduced and accordingly, the characteristics are the same for all frequencies.

What benefits does dynamic control allocation offer in this case? The overall behavior of the output signals is nearly the same regardless of the control allocation scheme, see Figure 9.5 to Figure 9.8. This is due to that they all generate the same aerodynamic moments, which is what mainly affects the aircraft behavior. Let us therefore focus on two details: the undershoot in the nonminimum phase load factor response, and the drag coefficient at trimmed flight.

Table 9.1 displays the value of the drag coefficient, C_D, at trimmed flight just before the maneuver starts ($t = 0.9$ s) for the different control allocation strategies. Only the minimum control strategy stands out, which gives a drag increase of 2% compared to the others.

Figure 9.4 shows a blowup of the load factor behavior at $t = 1$, which is when the pitch command is applied. The curve with the largest undershoot comes from when the canards are not used at all. The dynamic control allocator, which produces the largest canard deflections of all methods just after $t = 1$ s, yields the smallest undershoot.

Hence, the benefit in this case of using dynamic control allocation is that the high frequency control distribution, affecting the initial aircraft response to a pilot command, and the steady state control distribution, determining the drag at trimmed flight, can be selected different. In this way, good pitch response characteristics and minimum drag can be achieved in one design.

Note that the minimum norm control allocator could be modified to also achieve minimum drag by minimizing $\|u - u_d\|$ instead, where u_d is the particular constant

Method	C_D $(-)$
δ_c for high freq.	0.01452
ADMIRE FCS	0.01453
$\delta_c = 0$	0.01452
$\min \|u\|$	0.01481

Table 9.1: Drag coefficient, C_D, at trimmed flight for different control allocation methods.

distribution that yields minimum drag at trimmed flight. However, this would require for u_d to be determined explicitly for all flight cases of interest.

9.4 Conclusions

In this chapter a new method for dynamic control allocation has been developed. Dynamic control allocation offers an extra degree of freedom compared to static control allocation, in that the distribution of control effort among the actuators need not be the same for all frequencies.

This can be useful if the actuators have different bandwidths but this was not taken into account in the underlying control design. Then slow actuators can be utilized only to produce the low frequency contents of the virtual control input, while the fast actuators may operate over the whole frequency range.

Another case where dynamic allocation may be useful is when the actuators affect the system behavior somewhat differently. This was illustrated in the aircraft design example, where the canards were used to produce high frequency pitching moment to reduce the nonminimum phase behavior of the pilot load factor.

The proposed framework for dynamic control allocation has several advantages. Posing the problem as a constrained optimization problem provides automatic redistribution of control effort when one actuator saturates in position or in rate. Further, using the l_2-norm gives the problem nice analytical properties, since the solution is a linear filter when no saturations occur.

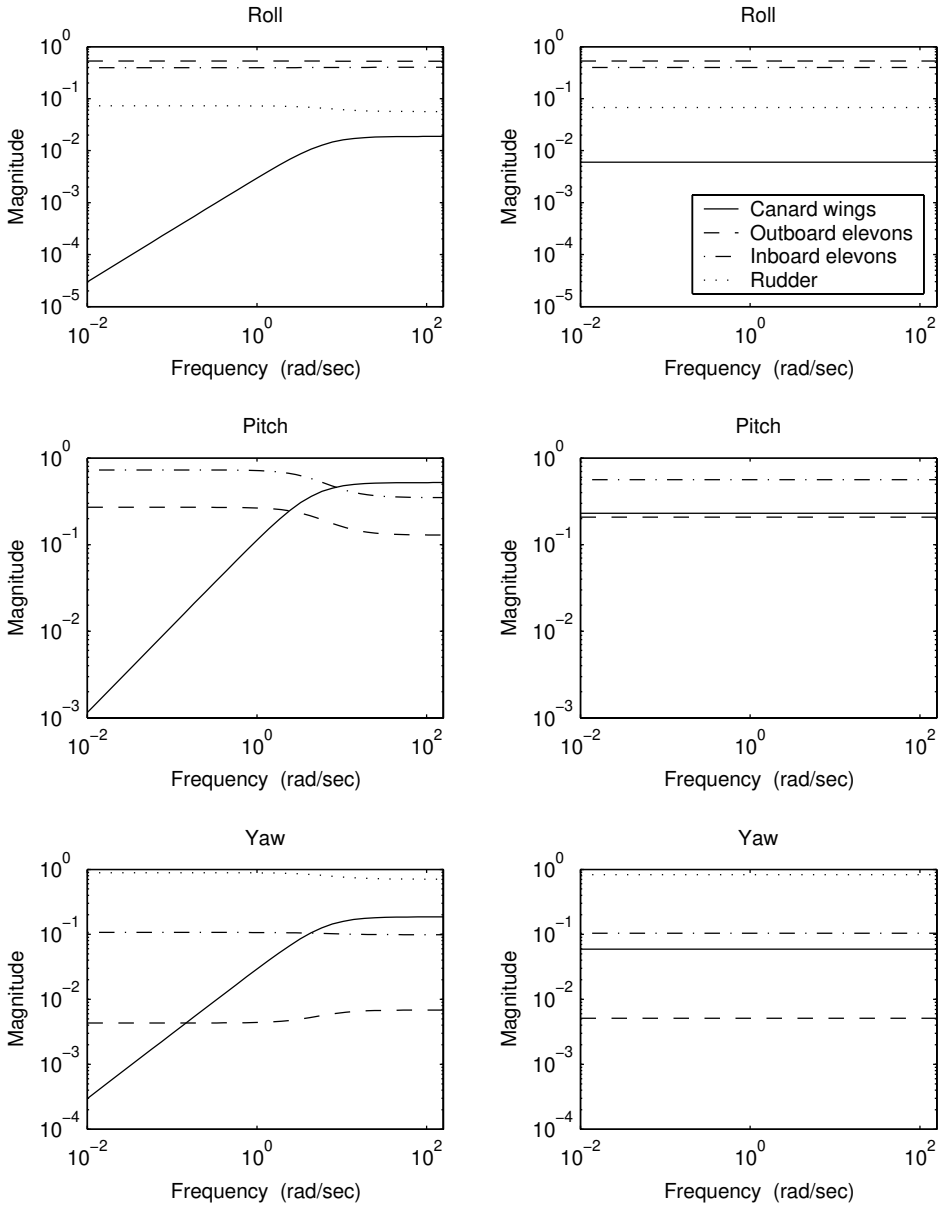


Figure 9.3: Dynamic vs. static control allocation. Left: Dynamic control allocation with canards operating only at high frequencies. Right: Static control allocation with a minimum control objective. Each transfer function has been weighted with its corresponding entry in B to show the proportion of v that the actuator produces.

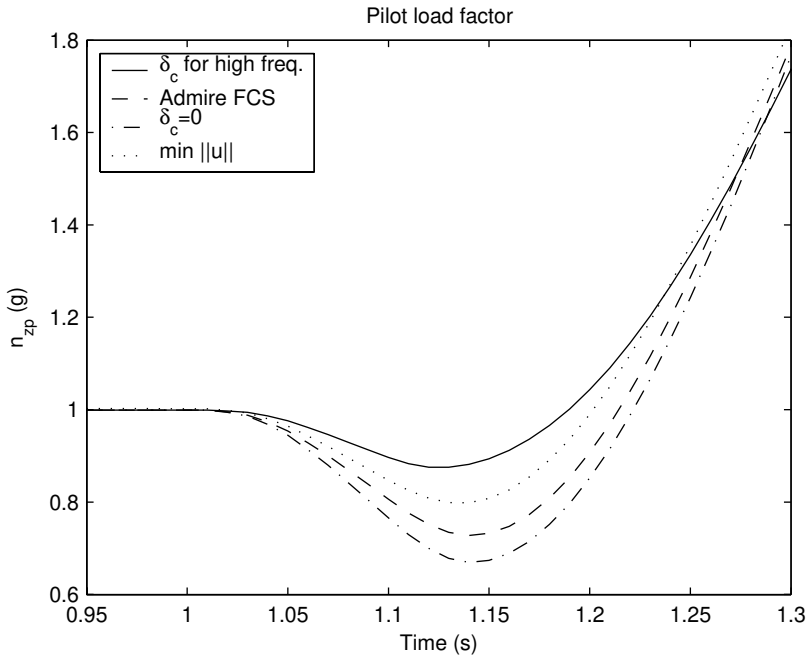


Figure 9.4: Comparison of the nonminimum phase behavior in n_{zp} for different control allocation strategies when a pitch command is applied. Using the canards to produce pitching moment reduces the undershoot in n_{zp}.

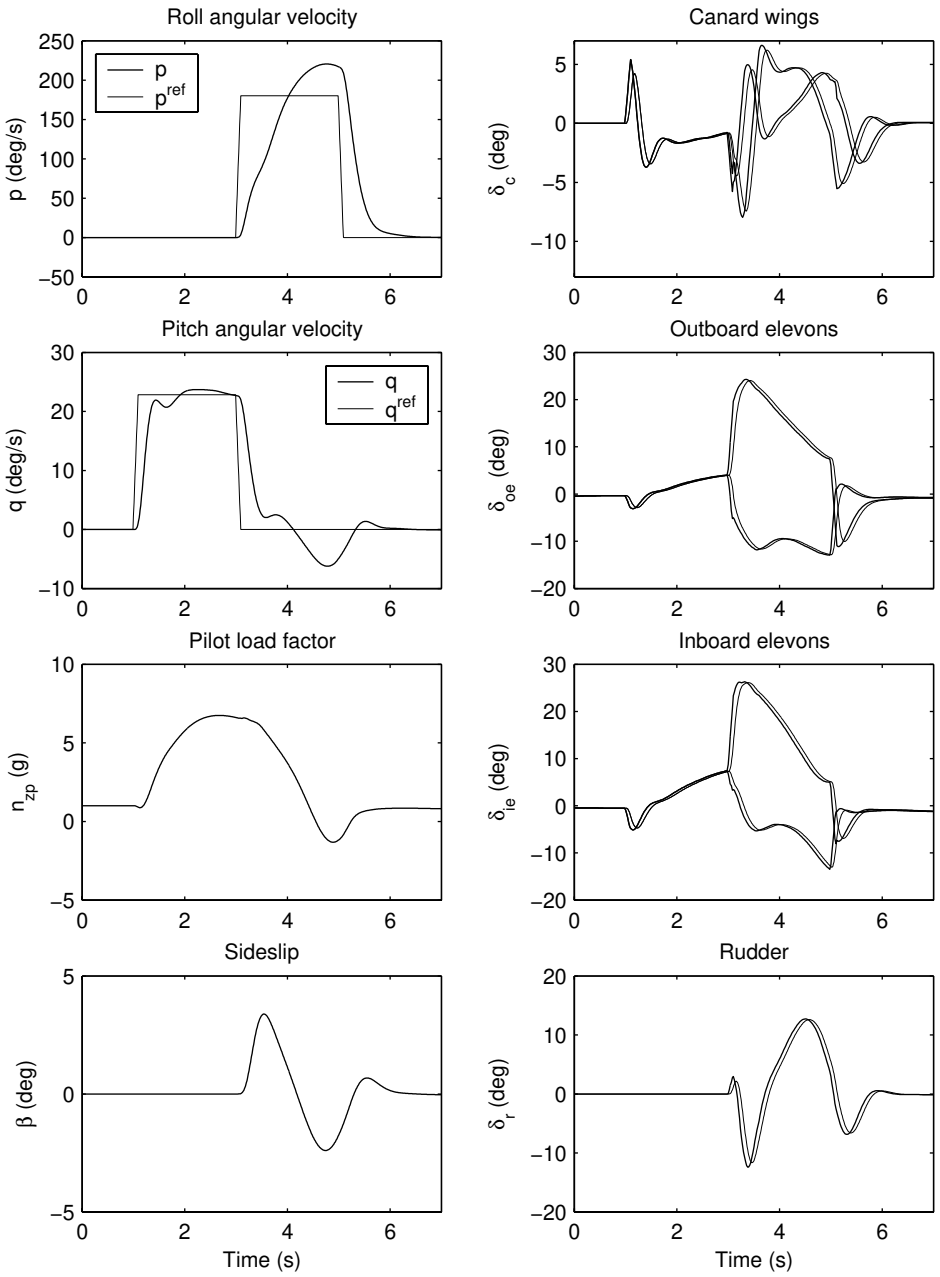


Figure 9.5: Simulation results when the dynamic control allocator is used, and the canards are used for high frequencies only. Left: aircraft trajectory. Right: commanded (thick lines) and actual (thin lines) control surface deflections.

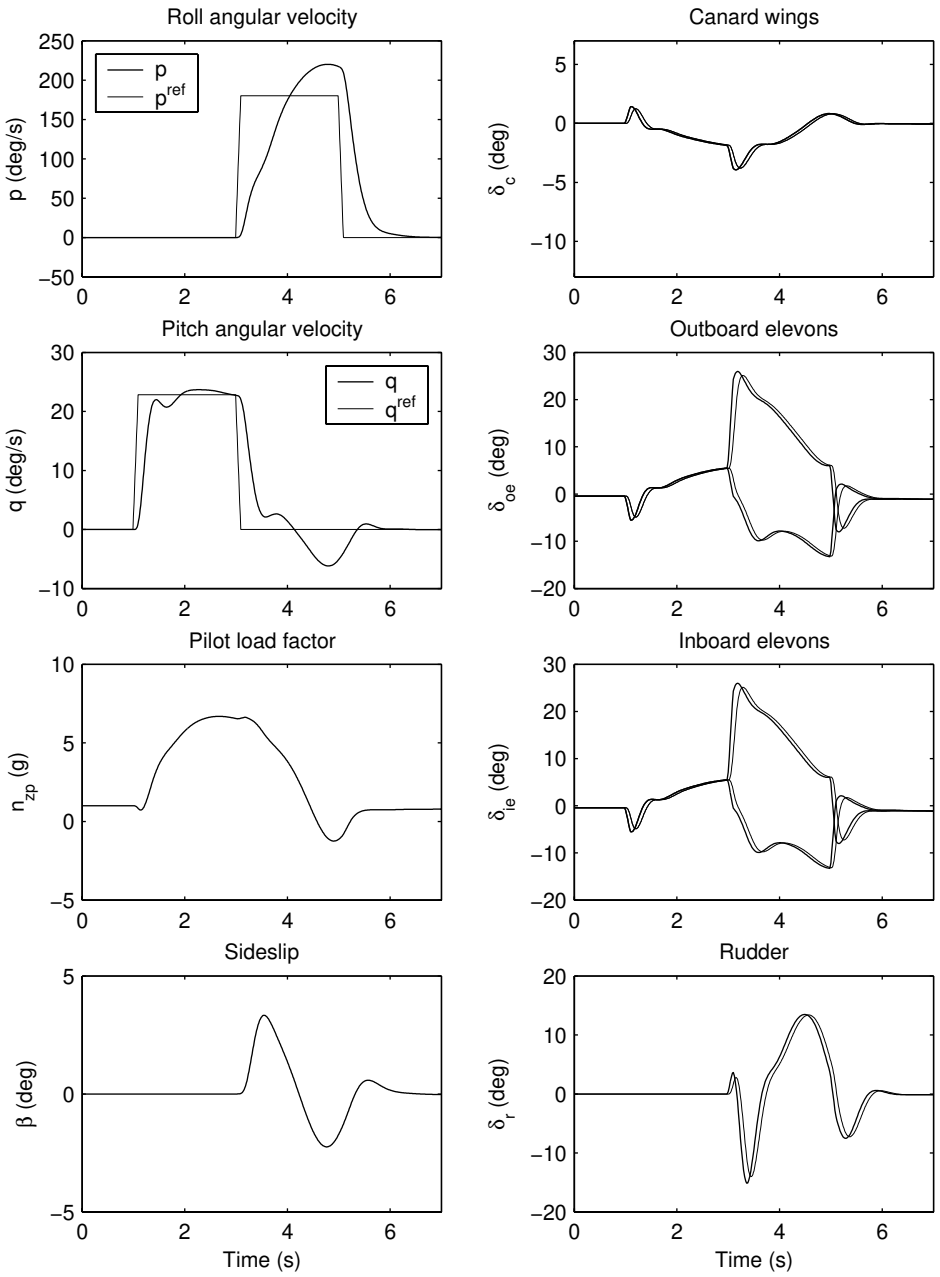


Figure 9.6: Simulation results when the original ADMIRE flight control system is used and no reallocation is performed. Left: aircraft trajectory. Right: commanded (thick lines) and actual (thin lines) control surface deflections.

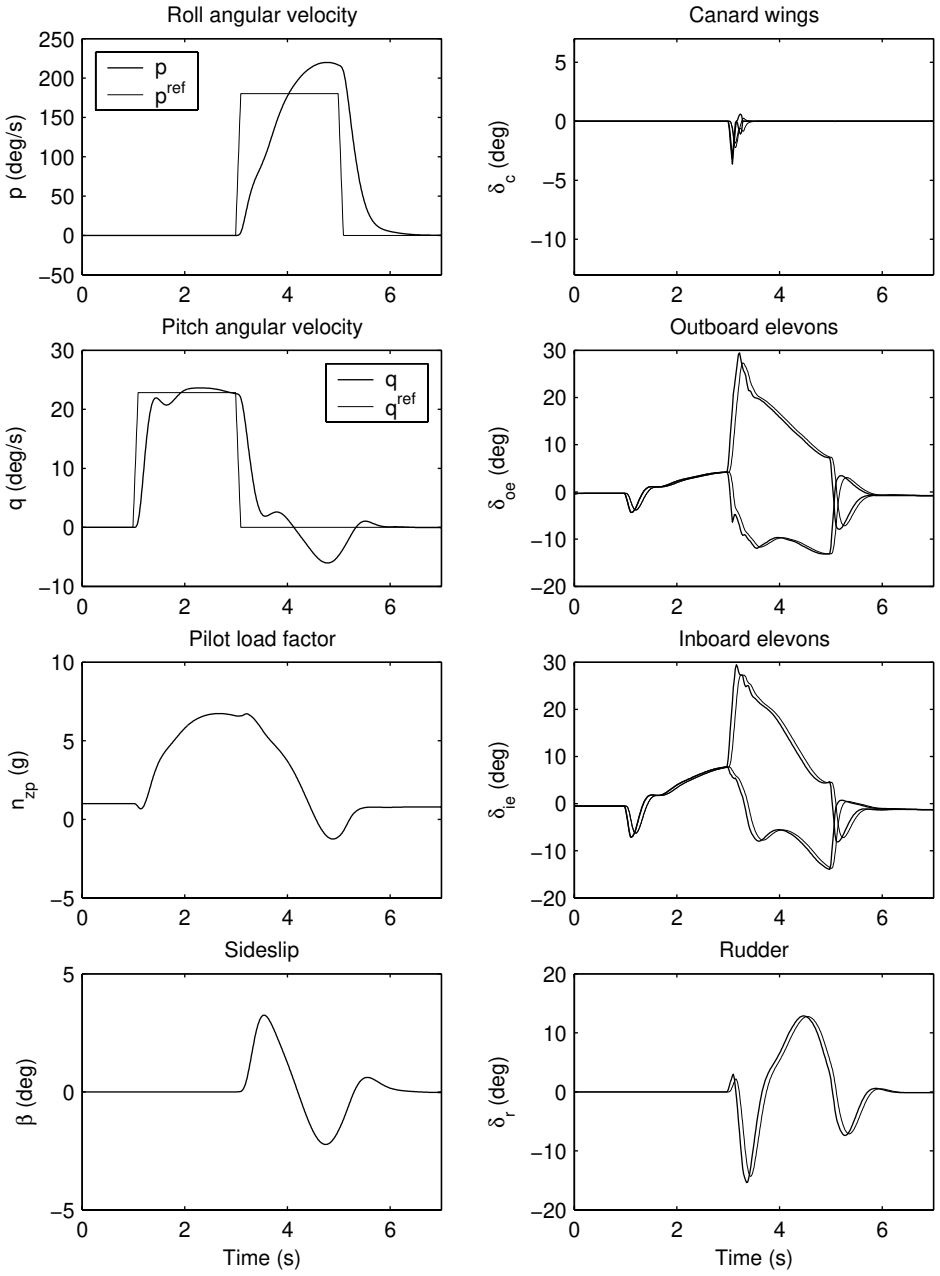


Figure 9.7: Simulation results when the canards are used only when the elevons saturate. Left: aircraft trajectory. Right: commanded (thick lines) and actual (thin lines) control surface deflections.

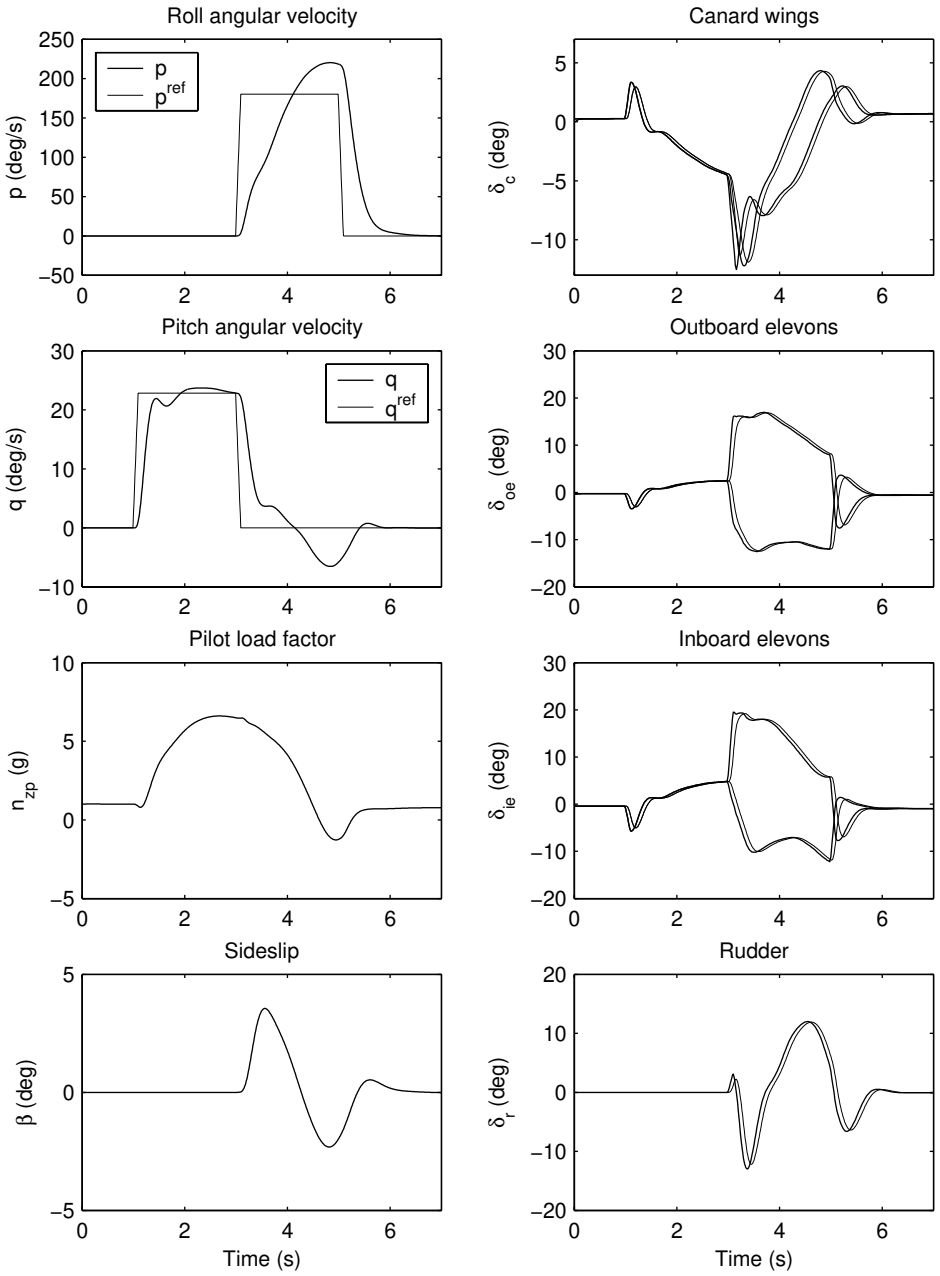


Figure 9.8: Simulation results when $\|u\|$ is minimized. Left: aircraft trajectory. Right: commanded (thick lines) and actual (thin lines) control surface deflections.

Chapter 10

Control Allocation vs Linear Quadratic Control

Actuator redundancy is one issue to be dealt with when designing controllers for systems with more inputs than outputs. A common approach is to use some optimal control design method, like linear quadratic (LQ) control or $\mathcal{H}_\infty$ control, to shape the closed loop dynamics as well as the actuator control distribution in one step (Zhou et al. 1996). Figure 10.1 illustrates the resulting controller structure.

An alternative, as we have seen previously in this thesis, is to separate the regulation task from the control distribution task. With this strategy, the control law specifies only which total control effect should be produced. The distribution of control among the actuators is decided by a separate control allocation module, see Figure 10.2.

In this chapter we derive some connections between these two strategies when quadratic performance indices are used both for control law design and for control allocation. Hence, LQ control and l_2-optimal control allocation will be used to design the control system building blocks in Figure 10.1 and Figure 10.2. This comparison is particularly interesting from a flight control perspective since LQ design today is a commonly used design method (Stevens and Lewis 1992, Amato et al. 1997), and control allocation is possibly becoming one.

The main result that we will show is that in the linear case, the two design strategies offer precisely the same design freedom. Given one design, we show how to select the parameters of the other design to obtain the same control law. We also motivate what benefits a modular design, with a separate control allocator, offers. In particular, actuator constraints can be handled in a potentially better way.

The chapter is organized as follows. In Section 10.1, linear quadratic regulation (LQR), the cornerstone of LQ control, is reviewed. In Section 10.2, the class of systems to which our results apply is presented, followed by the actual control designs that correspond to the two strategies outlined above. Section 10.3 contains

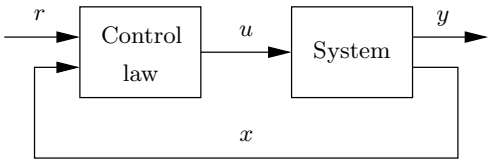


Figure 10.1: Standard control system structure.

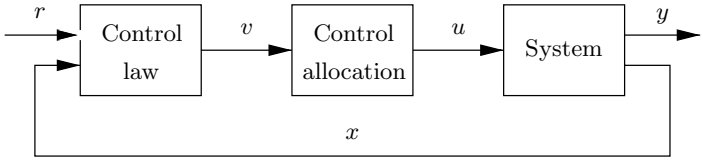


Figure 10.2: Control system structure when control allocation is performed separately.

two theorems connecting the strategies to each other, followed by a discussion on the benefits of using a separate control allocator. In Section 10.4, the results are illustrated using a realistic flight control example, and conclusions are drawn in Section 10.5.

This chapter is based on

> O. Härkegård. Resolving actuator redundancy—control allocation vs linear quadratic control. In *Proc. of the European Control Conference*, Cambridge, UK, Sept. 2003. Accepted for presentation.

10.1 Linear Quadratic Regulation

Consider a linear, controllable system

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx \end{aligned} \tag{10.1}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, and $y \in \mathbb{R}^p$ is the system output to be controlled. Let $\text{rank}(C) = p$ and $\text{rank}(B) = k \geq p$. Assume that x is measured, i.e., consider the full-state feedback case.

Let the control objective be to bring the state x to zero while minimizing the quadratic performance index

$$J = \int_0^\infty \big(x(t)^T Q x(t) + u(t)^T R u(t)\big) dt \tag{10.2}$$

where Q is a positive semidefinite matrix and R is a positive definite matrix. This is the standard infinite-time linear quadratic regulation (LQR) problem, see Anderson

and Moore (1989). Q and R are selected by the designer to achieve a trade-off between (a) quick convergence towards $x = 0$, and (b) moderate use of the actuators. The optimal control law is given by

$$u = -Lx$$

where

$$L = R^{-1}B^T S \tag{10.3}$$

and S is the unique positive semidefinite and symmetric solution to

$$A^T S + SA + Q - SBR^{-1}B^T S = 0 \tag{10.4}$$

known as the algebraic Riccati equation (ARE). The optimal performance index is given by $J_{\text{opt}} = x(0)^T S x(0)$.

10.2 LQ Designs for Overactuated Systems

As stated in the introduction, control design for overactuated dynamic systems can be performed in two fundamentally different ways; either a separate control allocator is used, or the control allocation is embedded in the control law design. In this section, the control laws resulting in the two cases are derived for a class of linear systems.

10.2.1 System Description

We will consider linear systems of the form

$$\begin{aligned} \dot{x} &= Ax + B_u u \\ y &= Cx \end{aligned} \tag{10.5}$$

where $x \in \mathbb{R}^n$ is the system state, $u \in \mathbb{R}^m$ is the control input, and $y \in \mathbb{R}^p$ is the system output to be controlled. Assume x to be measured so that full state information is available.

Let $\text{rank}(B_u) = k \leq m$ so that B_u can be factorized as

$$B_u = B_v B$$

where B_u is $n \times m$, B_v is $n \times k$, and B is $k \times m$. With this, an alternative system description is

$$\begin{aligned} \dot{x} &= Ax + B_v v \\ v &= Bu \\ y &= Cx \end{aligned} \tag{10.6}$$

where $v \in \mathbb{R}^k$ is the virtual control input.

We will restrict our discussion to the case $k = p$, i.e., when the number of virtual control inputs equals the number of variables to be controlled. The results can probably be extended to the general case $k \geq p$.

10.2.2 Control Design

Let the control objective be for the output y to track a constant reference signal r, so that $y = r$ is achieved asymptotically.

Standard LQ Design

In the first approach we consider (10.5) and use LQ design to achieve stability as well as to distribute the control effort among the control inputs.

Design 1: Determine the control input $u(t)$ by solving

$$\min_u \int_0^\infty \big((x - x^*)^T Q_1 (x - x^*) + (u - u^*)^T R_1 (u - u^*)\big) dt \tag{10.7}$$

where x^*, u^* solve

$$\begin{aligned} \min_{x,u} \quad & u^T R_1 u \\ \text{subject to} \quad & Ax + B_u u = 0 \\ & Cx = r \end{aligned} \tag{10.8}$$

Hence, if there are several choices of u that achieve $\dot{x} = 0$ and $y = r$, we pick u such that $u^T R_1 u$ is minimized. The optimal control law is given by the following theorem, based on Glad and Ljung (2000, Theorem 9.2, p. 244).

Theorem 10.1 *The optimal control law for Design 1 is given by*

$$\begin{aligned} u(t) &= L_r r - Lx(t) \\ L_r &= R_1^{-\frac{1}{2}} (G_0 R_1^{-\frac{1}{2}})^\dagger \\ L &= R_1^{-1} B_u^T S_1 \end{aligned} \tag{10.9}$$

Here,

$$G_0 = C(B_u L - A)^{-1} B_u$$

and S_1 is the unique positive semidefinite and symmetric solution to

$$A^T S_1 + S_1 A + Q_1 - S_1 B_u R_1^{-1} B_u^T S_1 = 0$$

Proof: Introduce the new variables $\tilde{x} = x - x^*$, $\tilde{u} = u - u^*$. The dynamics of $\tilde{x}$ become

$$\dot{\tilde{x}} = \dot{x} = Ax + B_u u = A\tilde{x} + B_u \tilde{u}$$

where the last step follows from that $Ax^* + B_u u^* = 0$. Using the LQR results from Section 10.1 gives the optimal control law

$$\tilde{u} = -L\tilde{x}$$

with L as in (10.9). In the original variables we get

$$u = \underbrace{u^* + Lx^*}_{u_r} - Lx$$

Inserting this into (10.8) gives us

$$\begin{aligned} \min_{x,u_r} \quad & (u_r - Lx)^T R_1 (u_r - Lx) \\ \text{subject to} \quad & Ax + B_u(u_r - Lx) = 0 \\ & Cx = r \end{aligned}$$

Solving for x, and using $B_u = B_v B$, the constraints become

$$\begin{aligned} x &= (B_u L - A)^{-1} B_v \underbrace{B u_r}_{v_r} \\ &\underbrace{C(B_u L - A)^{-1} B_v}_{D} v_r = r \end{aligned}$$

Assuming that D $(p \times p)$ is nonsingular (or the control problem would not be feasible), we see that v_r, and consequently x, are completely determined by r. Hence, x can be removed from the minimization, and the optimization problem becomes

$$\begin{aligned} \min_{u_r} \quad & (u_r - Lx)^T R_1 (u_r - Lx) \\ \text{subject to} \quad & B u_r = D^{-1} r \end{aligned}$$

which according to Lemma B.1 has the solution

$$u_r = (I - R_1^{-1} B^T (B^T R_1^{-1} B^T)^{-1} B) Lx + \underbrace{R_1^{-1} B^T (B R_1^{-1} B^T)^{-1} D^{-1}}_{L_r} r$$

Since $L = R_1^{-1} B^T B_v^T S_1$, the first term is zero. This gives $u_r = L_r r$. Introducing $G_0 = DB = C(B_u L - A)^{-1} B_u$, L_r can be rewritten as $L_r = R_1^{-1} G_0^T (G_0 R_1^{-1} G_0^T)^{-1}$, or, equivalently, $L_r = R_1^{-\frac{1}{2}} (G_0 R_1^{-\frac{1}{2}})^\dagger$. □

LQ Design and Control Allocation

Our second LQ based control design springs from the divided system description (10.6). We first design a stabilizing virtual control law using LQ design and then use l_2-optimal control allocation to distribute the control effect among the individual control inputs.

Design 2: Determine the virtual control input $v(t)$ by solving

$$\min_v \int_0^\infty \big((x-x^*)^T Q_2 (x-x^*) + (v-v^*)^T R_2 (v-v^*)\big) dt$$

where x^*, v^* solve

$$\begin{aligned} Ax + B_v v &= 0 \\ Cx &= r \end{aligned}$$

Then determine the control input $u(t)$ by solving the control allocation problem

$$\begin{aligned} \min_u \quad & \|Wv\| \\ \text{subject to} \quad & Bu = v \end{aligned}$$

Theorem 10.2 *The optimal control law for Design 2 is given by*

$$\begin{aligned} u(t) &= Pv(t) \\ P &= W^{-1}(BW^{-1})^\dagger \end{aligned}$$

where the optimal virtual control input is given by

$$\begin{aligned} v(t) &= L_r r - Lx(t) \\ L_r &= G_0^{-1} \\ L &= R_2^{-1} B_v^T S_2 \end{aligned}$$

Here,

$$G_0 = C(B_v L - A)^{-1} B_v$$

and S_2 is the unique positive semidefinite and symmetric solution to

$$A^T S_2 + S_2 A + Q_2 - S_2 B_v R_2^{-1} B_v^T S_2 = 0$$

Proof: The expression for L follows directly from Theorem 10.1. L_r also follows from Theorem 10.1 as a special case. Since G_0 is square we get $R_1^{-\frac{1}{2}}(G_0 R_1^{-\frac{1}{2}})^\dagger = G_0^{-1}$. $P = W^{-1}(BW^{-1})^\dagger$ follows from Lemma B.1. □

10.3 Main Results

We will now present two results which connect Design 1 and Design 2 in terms of the resulting virtual control input, $v(t)$, and true control input, $u(t)$. In the presentation, subscripts 1 and 2 are used to specify which design a certain entity (u, v, etc.) is related to.

Theorem 10.3 *Consider Design 1 and Design 2. Given Q_1 and R_1,*

$$\begin{aligned} Q_2 &= Q_1 \\ R_2 &= \left(BR_1^{-1}B^T\right)^{-1} \end{aligned} \tag{10.10}$$

achieve $v_2(t) = v_1(t)$. Conversely, given Q_2 and R_2,

$$\begin{aligned} Q_1 &= Q_2 \\ R_1 &= \left(B^\dagger R_2^{-1}(B^\dagger)^T\right)^{-1} \end{aligned} \tag{10.11}$$

achieve $v_1(t) = v_2(t)$.

Proof: From Theorem 10.1 and Theorem 10.2 we get

$$\begin{aligned} u_1(t) &= -R_1^{-1}B_u^T S_1 x(t) = -R_1^{-1}B^T B_v^T S_1 x(t) \\ v_2(t) &= -R_2^{-1}B_v^T S_2 x(t) \end{aligned}$$

where S_1 and S_2 solve

$$\begin{aligned} A^T S_1 + S_1 A + Q_1 - S_1 B_u R_1^{-1} B_u^T S_1 &= 0 \\ A^T S_2 + S_2 A + Q_2 - S_2 B_v R_2^{-1} B_v^T S_2 &= 0 \end{aligned}$$

For Design 1 we get the virtual control signal

$$v_1(t) = Bu_1(t) = -BR_1^{-1}B^T B_v^T S_1 x(t)$$

By inspection we see that

$$\begin{aligned} Q_2 &= Q_1 \\ R_2^{-1} &= BR_1^{-1}B^T \end{aligned}$$

give the same solutions to the ARE:s, $S_1 = S_2 = S$, and also the same virtual control signals, $v_1(t) = v_2(t)$, which proves (10.10). Solving for R_1^{-1} gives $R_1^{-1} = B^\dagger R_2^{-1}(B^\dagger)^T$ as one of many solutions, which proves (10.11). □

Theorem 10.4 *Consider Design 1 and Design 2. Given Q_1 and R_1,*

$$\begin{aligned} Q_2 &= Q_1 \\ R_2 &= \left(BR_1^{-1}B^T\right)^{-1} \\ W &= R_1^{\frac{1}{2}} \end{aligned} \tag{10.12}$$

achieve $u_2(t) = u_1(t)$. Conversely, given Q_2, R_2, and W,

$$\begin{aligned} Q_1 &= Q_2 \\ R_1 &= W^2 + B^T\left(R_2 - (BW^{-2}B^T)^{-1}\right)B \end{aligned} \tag{10.13}$$

achieve $u_1(t) = u_2(t)$.

Proof: For the control signals to be equal, the virtual control signals must be equal. From Theorem 10.3 we know that this holds if

$$\begin{aligned} Q_2 &= Q_1 \\ R_2^{-1} &= BR_1^{-1}B^T \end{aligned}$$

Further, according to Theorem 10.2,

$$u_2(t) = W^{-1}(BW^{-1})^\dagger v_2(t) = W^{-2}B^T(BW^{-2}B^T)^{-1}v_2(t)$$

Using this along with the expressions for $u_1(t)$ and $v_2(t)$ from the previous proof yields

$$\begin{aligned} u_1(t) &= -R_1^{-1}B^T B_v^T S x(t) \\ u_2(t) &= -W^{-2}B^T(BW^{-2}B^T)^{-1}BR_1^{-1}B^T B_v^T S x(t) \end{aligned}$$

The choice $W^2 = R_1$ yields

$$\begin{aligned} u_2(t) &= -R_1^{-1}B^T \underbrace{(BR_1^{-1}B^T)^{-1}BR_1^{-1}B^T}_{=I} B_v^T S x(t) \\ &= u_1(t) \end{aligned}$$

which proves (10.12). Note that the choice of W is not unique.

Deriving (10.13) is not quite as straightforward. To do this, consider Design 2 but with a different control allocation objective,

$$\min_u \left\|\tilde{W}u\right\| \quad \text{subject to} \quad Bu = v \tag{10.14}$$

From above we know that this gives the same control signal as Design 1 if

$$\begin{aligned} Q_2 &= Q_1 \\ R_2^{-1} &= BR_1^{-1}B^T \\ \tilde{W}^2 &= R_1 \end{aligned}$$

Further, (10.14) gives the same control law as Design 2 if

$$\tilde{W}^2 = W^2 + B^T X B$$

for any symmetric X such that $\tilde{W}^2$ is positive definite. This is true since under the constraint $Bu = v$ it holds that

$$\begin{aligned} \arg\min_u \left\|\tilde{W}u\right\| &= \arg\min_u u^T\tilde{W}^2 u = \arg\min_u u^T(W^2 + B^T X B)u \\ &= \arg\min_u u^T W^2 u + v^T X v = \arg\min_u u^T W^2 u = \arg\min_u \left\|Wu\right\| \end{aligned}$$

Thus, $u_1 = u_2$ is achieved for

$$\begin{aligned} Q_1 &= Q_2 \\ R_1 &= W^2 + B^T X B \end{aligned}$$

if there exists a symmetric matrix X that solves

$$R_2^{-1} = B R_1^{-1} B^T = B(W^2 + B^T X B)^{-1} B^T$$

and makes R_1 positive definite.

First solve for X. Using the matrix inversion formula

$$(A + BD)^{-1} = A^{-1} - A^{-1}B(I + DA^{-1}B)^{-1}DA^{-1}$$

gives us

$$\begin{aligned} R_2^{-1} &= B(W^2 + B^T X B)^{-1} B^T \\ &= B\Big(W^{-2} - W^{-2}B^T(I + X \underbrace{BW^{-2}B^T}_{M})^{-1} X B W^{-2}\Big) B^T \\ &= M - M(I + XM)^{-1} X M \end{aligned}$$

Rearranging gives

$$XM = (I + XM)M^{-1}(M - R_2^{-1}) = I - M^{-1}R_2^{-1} + XM - XR_2^{-1}$$

which has the solution

$$X = R_2 - M^{-1} = R_2 - (BW^{-2}B^T)^{-1}$$

This yields

$$R_1 = W^2 + B^T\big(R_2 - (BW^{-2}B^T)^{-1}\big)B$$

What remains is to check that R_1 is positive definite.

$$\begin{aligned} u^T R_1 u &= u^T W\Big(I + W^{-1}B^T\big(R_2 - (BW^{-2}B^T)^{-1}\big)\underbrace{BW^{-1}}_{N}\Big)\underbrace{Wu}_{\tilde{u}} \\ &= \tilde{u}^T\big(I + N^T(R_2 - (NN^T)^{-1})N\big)\tilde{u} \\ &= \tilde{u}^T\big(I + N^T R_2 N - N^T(NN^T)^{-1}N\big)\tilde{u} \end{aligned}$$

Using the singular value decomposition

$$N = U\begin{pmatrix}\Sigma_r & 0\end{pmatrix}\begin{pmatrix}V_r^T \\ V_0^T\end{pmatrix} = U\Sigma_r V_r^T$$

gives

$$N^T(NN^T)^{-1}N = V_r\Sigma_r U^T(U\Sigma_r^2 U^T)^{-1}U\Sigma_r V_r^T = V_r V_r^T$$

Parameterize $\tilde{u}$ as

$$\tilde{u} = V_r \tilde{u}_r + V_0 \tilde{u}_0$$

This yields

$$\begin{aligned} u^T R_1 u &= \tilde{u}_r^T \tilde{u}_r + \tilde{u}_0^T \tilde{u}_0 + \tilde{u}_r^T \Sigma_r U^T R_2 U \Sigma_r \tilde{u}_r - \tilde{u}_r^T \tilde{u}_r \\ &= \tilde{u}_0^T \tilde{u}_0 + \tilde{u}_r^T \underbrace{\Sigma_r U^T R_2 U \Sigma_r}_{\text{pos. def.}} \tilde{u}_r > 0, \quad u \neq 0 \end{aligned}$$

which shows that R_1 is indeed positive definite. □

Let us now discuss the implications of these two "conversion theorems", relating traditional LQ design to l_2-optimal control allocation.

The main message is that the two approaches give the designer the exact same freedom to shape the closed loop dynamics and to distribute the control effort among the actuators. Given the design parameters of one design, Theorem 10.4 states how the parameters of the other design should be selected to achieve precisely the same control law.

So why then bother to split the control design into two separate tasks? Let us list some benefits of using a modular control design.

- *Facilitates tuning.* In Design 1, modifying an element of the control input weighting matrix, R_1, will affect the control distribution as well as the closed loop behavior of the system. In Design 2, the tuning of the closed loop dynamics is separated from the design of the control distribution.

- *Easy to reconfigure.* An actuator failure can often be approximately modeled as a change in the B-matrix. In Design 2 this only affects the control allocation. Hence, if the failure is detected, the new B-matrix can be used for control allocation, while the original virtual control law can still be used, provided that damaged system is still controllable.

- *Arbitrary control allocation method.* From (10.6) we see that the system dynamics are completely determined by the virtual control input, v. Hence, Theorem 10.3 allows us to use any control allocation mapping $u = h(v)$ in Design 2, such that $Bh(v) = v$, without altering the close loop dynamics from Design 1.

- *Actuator constraints.* With a separate control allocator, actuator constraints can be handled to some extent. If at time t, the control input is bounded by $\underline{u}(t) \leq u(t) \leq \overline{u}(t)$, considering position and rate constraints (see (7.6)), the control allocation problem in Design 2 can be reformulated as (cf. Section 7.3.1)

$$\begin{aligned} u &= \arg\min_{u \in \Omega} \left\| Wu \right\| \\ \Omega &= \arg\min_{\underline{u} \leq u \leq \overline{u}} \left\| W_v (Bu - v) \right\| \end{aligned} \tag{10.15}$$

This way, the control capabilities of the actuator suite can be fully exploited before the closed loop performance is degraded. Also, when $Bu = v$ is not attainable, W_v allows the designer to prioritize between the components of the virtual control input.

Remark: It should be stressed that including the constraints in the control allocation is not equivalent to including the constraints in the original LQ problem in Design 1.

Apparently, a modular design has potential benefits. Unfortunately, it is not always possible to perform the division even when the number of actuators exceeds the number of controlled variables. Recall, for example, the discussion in Section 7.2.3 where it was shown that the rank assumption in Section 10.2.1 ($\text{rank}(B_u) = p$) breaks down in the presence of actuator dynamics. Hence, to achieve modularity, some model approximations may be necessary, as we will see in the design example in the following section.

10.4 Flight Control Example

To investigate the potential benefits of a modular LQ design, we use a flight control example, based on the ADMIRE model (ADMIRE ver. 3.4h 2003), see Section 2.4. To induce actuator saturations, we consider a low speed flight case, Mach 0.22, altitude 3000 m, where the control surface efficiency is poor.

System Description

The commanded variables are the angle of attack, α, and the roll rate, p. The sideslip angle, β, is to be regulated to zero. Hence,

$$y = \begin{pmatrix} \alpha & \beta & p \end{pmatrix}^T$$

Beside these, we also include the pitch rate, q, and the yaw rate, r, in the system state,

$$x = \begin{pmatrix} \alpha & \beta & p & q & r \end{pmatrix}^T$$

The control surface vector

$$\delta = \begin{pmatrix} \delta_c & \delta_{re} & \delta_{le} & \delta_r \end{pmatrix}^T$$

contains the positions of the canard wings, the right and left elevons, and the rudder, see Figure 2.6. The left and right canard wings, as well as the inner and outer elevons on each side, have been ganged. For the considered flight case, the

actuator position constraints are given by

$$\begin{aligned}\delta_{\min} &= \begin{pmatrix}-55^\circ & -30^\circ & -30^\circ & -30^\circ\end{pmatrix}^T \\ \delta_{\max} &= \begin{pmatrix}25^\circ & 30^\circ & 30^\circ & 30^\circ\end{pmatrix}^T\end{aligned} \tag{10.16}$$

The actuator rate constraints are not considered in this example.

At trimmed flight,

$$\begin{aligned}x = x_0 &= \begin{pmatrix}12.7^\circ & 0 & 0 & 0 & 0\end{pmatrix}^T \\ \delta = \delta_0 &= \begin{pmatrix}0 & 5.4^\circ & 5.4^\circ & 0\end{pmatrix}^T\end{aligned}$$

Introducing the deviations $\tilde{x} = x - x_0$ and $\tilde{\delta} = \delta - \delta_0$, the linearized ADMIRE dynamics become

$$\dot{\tilde{x}} = A\tilde{x} + B\tilde{\delta}$$

where

$$A = \begin{pmatrix}-0.5432 & 0.0137 & 0 & 0.9778 & 0 \\ 0 & -0.1179 & 0.2215 & 0 & -0.9661 \\ 0 & -10.5128 & -0.9967 & 0 & 0.6176 \\ 2.6221 & -0.0030 & 0 & -0.5057 & 0 \\ 0 & 0.7075 & -0.0939 & 0 & -0.2127\end{pmatrix}$$

$$B = \begin{pmatrix}0.0069 & -0.0866 & -0.0866 & 0.0004 \\ 0 & 0.0119 & -0.0119 & 0.0287 \\ 0 & -4.2423 & 4.2423 & 1.4871 \\ 1.6532 & -1.2735 & -1.2735 & 0.0024 \\ 0 & -0.2805 & 0.2805 & -0.8823\end{pmatrix}$$

The actuator dynamics are given by

$$\begin{aligned}\dot{\tilde{\delta}} &= B_\delta(\tilde{u} - \tilde{\delta}) \\ B_\delta &= 20I\end{aligned}$$

where $\tilde{u} = u - \delta_0$ and u is the control input consisting of the commanded control surface positions. This corresponds to first order dynamics with a time constant of 0.05 s. The total aircraft dynamics become

$$\begin{pmatrix}\dot{\tilde{x}} \\ \dot{\tilde{\delta}}\end{pmatrix} = \begin{pmatrix}A & B \\ 0 & B_\delta\end{pmatrix}\begin{pmatrix}\tilde{x} \\ \tilde{\delta}\end{pmatrix} + \begin{pmatrix}0 \\ B_\delta\end{pmatrix}\tilde{u} \tag{10.17}$$

Since $\text{rank}(B_\delta) = 4 \neq 3$, this model does not satisfy the requirements in Section 10.2.1. We therefore make the following two approximations:

- The actuator dynamics are neglected, and $\delta = u$ is used.
- The $\dot{\alpha}$ and $\dot{\beta}$ dependencies on δ, corresponding to the top two rows in B, are neglected.

This gives the approximate model

$$\dot{\tilde{x}} = A\tilde{x} + \tilde{B}_u\tilde{u} \tag{10.18}$$

where

$$\tilde{B}_u = \tilde{B}_v\tilde{B}$$

and where

$$\tilde{B}_v = \begin{pmatrix} 0_{2\times 3} \\ I_{3\times 3} \end{pmatrix}$$

$$\tilde{B} = \begin{pmatrix} 0 & -4.2423 & 4.2423 & 1.4871 \\ 1.6532 & -1.2735 & -1.2735 & 0.0024 \\ 0 & -0.2805 & 0.2805 & -0.8823 \end{pmatrix}$$

The resulting virtual control input, $v = \tilde{B}\tilde{u}$, contains the angular accelerations in roll, pitch, and yaw produced by the control surfaces.

Control Design

Three different control strategies are investigated.

1. Standard LQ design for the approximate model (10.18), see Design 1, with weighting matrices Q_1, R_1.

2. LQ design and l_2-optimal control allocation for the approximate model (10.18), see Design 2, with Q_2 and R_2 selected as in Theorem 10.4. To handle the actuator position constraints, the extended control allocation formulation (10.15) is used with two different choices of W_v.

3. Standard LQ design for the full model (10.17). L is selected as in Section 10.1, with weighting matrices

$$Q = \begin{pmatrix} Q_1 & 0 \\ 0 & 0 \end{pmatrix}$$
$$R = R_1$$

   and L_r is selected as in Theorem 10.1.

Q_1 and R_1 are selected by tuning design 3 to fulfill the requirements in Ståhl Gunnarsson (1999) regarding the damping (ζ) and natural frequency (ω_n) of the short period mode and the dutch roll mode, and the time constant (τ) of the roll mode.

$$Q_1 = \text{diag}(10, 10, 2, 1, 10)$$
$$R_1 = \text{diag}(10, 10, 10, 10)$$

yields

$$\begin{aligned} \zeta_{\text{sp}} &= 0.98 & \zeta_{\text{dr}} &= 0.58 \\ \omega_{n,\text{sp}} &= 1.84 & \omega_{n,\text{dr}} &= 1.72 \end{aligned} \qquad \tau_{\text{roll}} = 0.42$$

Simulation Results

The full model (10.17) is used for simulation. The reference trajectory is shown in Figure 10.3.

Figure 10.3 shows the simulation results when the actuator constraints (10.16) are not included in the aircraft model. Since no saturations occur, designs 1 and 2 give the exact same results. The difference in the results between these two and design 3 is that design 3 is based on the full model. This explains why only design 3 achieves the commanded angle of attack.

Figure 10.4 shows the results when the actuator constraints are included. Recall that this is a case where we expect control allocation to be useful. When the roll command is applied at $t = 3$ s, the left elevons saturate. In designs 1 and 3, this causes an overshoot in the pitch variables, α and q. In design 2, the control allocator copes with this saturation by redistributing as much of the lost control effect as possible to the right elevons and to the canards. The result is that the nominal trajectory from Figure 10.3 is almost completely recovered.

Finally, Figure 10.5 shows the effects of the axis prioritization matrix W_v. The choice $W_v = \text{diag}(1, 1, 1)$, corresponding to equal priorities for all three axes, gives a large overshoot in the sideslip angle at $t = 4$ s. To improve the yaw response, $W_v = \text{diag}(1, 1, 100)$ is selected (which was the choice in Figure 10.4). This puts a large penalty on the discrepancy between the demanded and the produced yaw angular acceleration in the control allocation, see (10.15). This forces the elevons and the rudder to be used primarily for yaw control. The result is a slightly slower initial roll response (barely visible), and an overall superior response in yaw as well as in roll.

10.5 Conclusions

Let us conclude this chapter on control allocation vs linear quadratic control.

For the considered class of linear systems, described in Section 10.2.1, standard LQ design, and LQ design in combination with l_2-optimal control allocation, both offer the exact same design freedom in shaping the closed loop response and

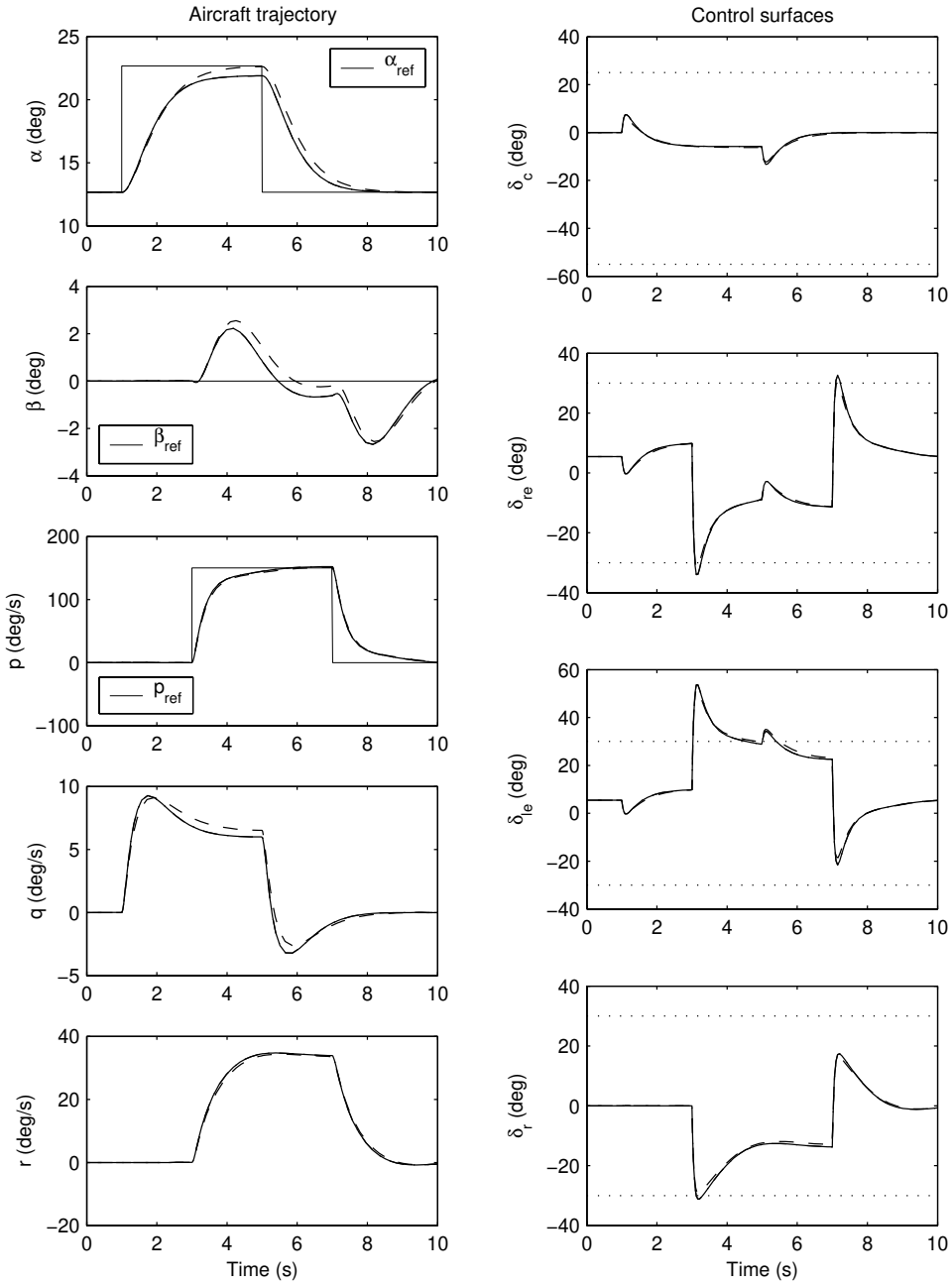


Figure 10.3: Aircraft trajectory, x (left), and control surface positions, δ (right), for control design 1, 2 (both solid), and 3 (dashed), when actuator constraints (dotted) are not included.

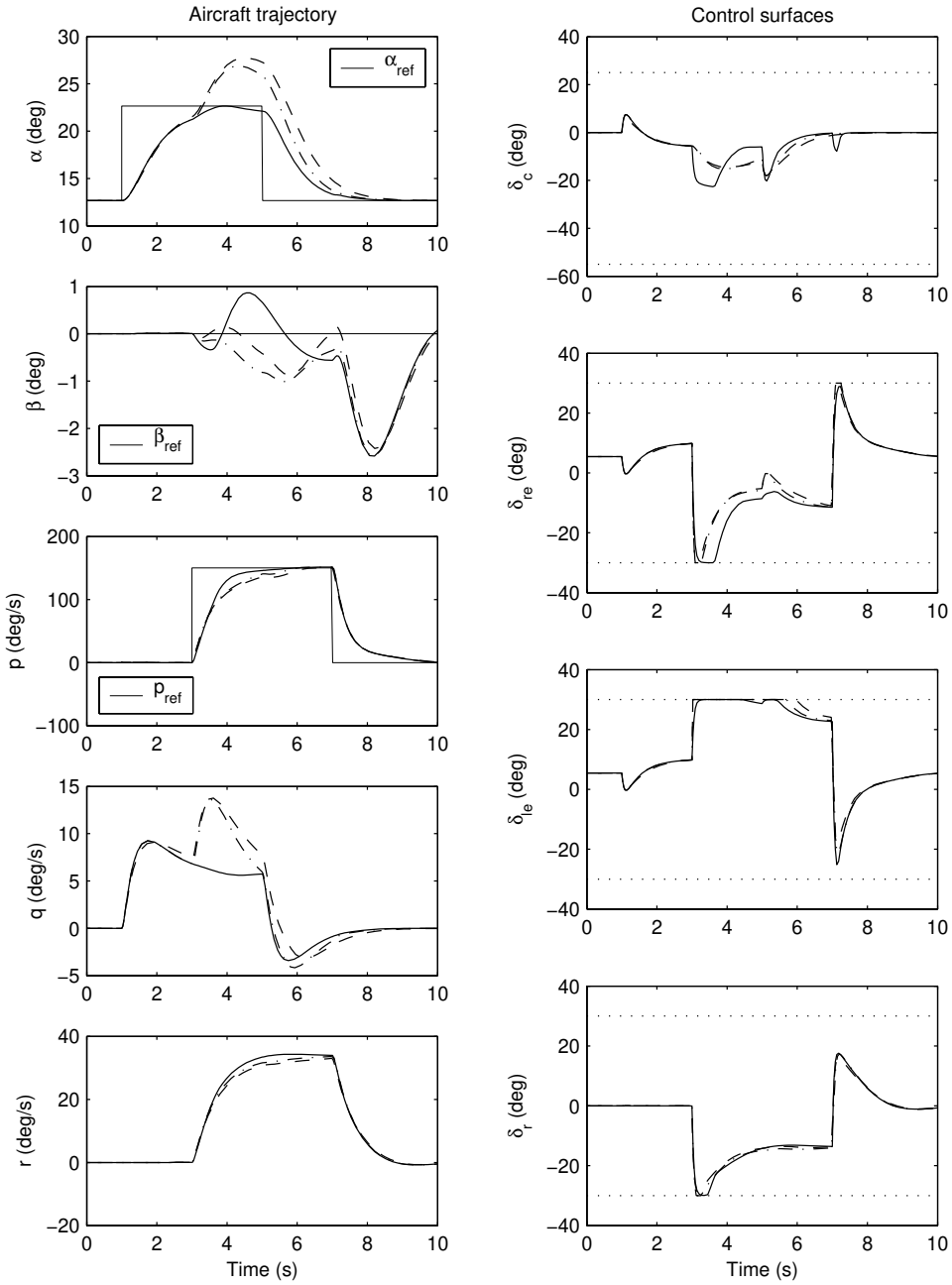


Figure 10.4: Aircraft trajectory, x (left), and control surface positions, δ (right), for control design 1 (dash-dotted), 2 (solid), and 3 (dashed), when actuator constraints (dotted) are included.

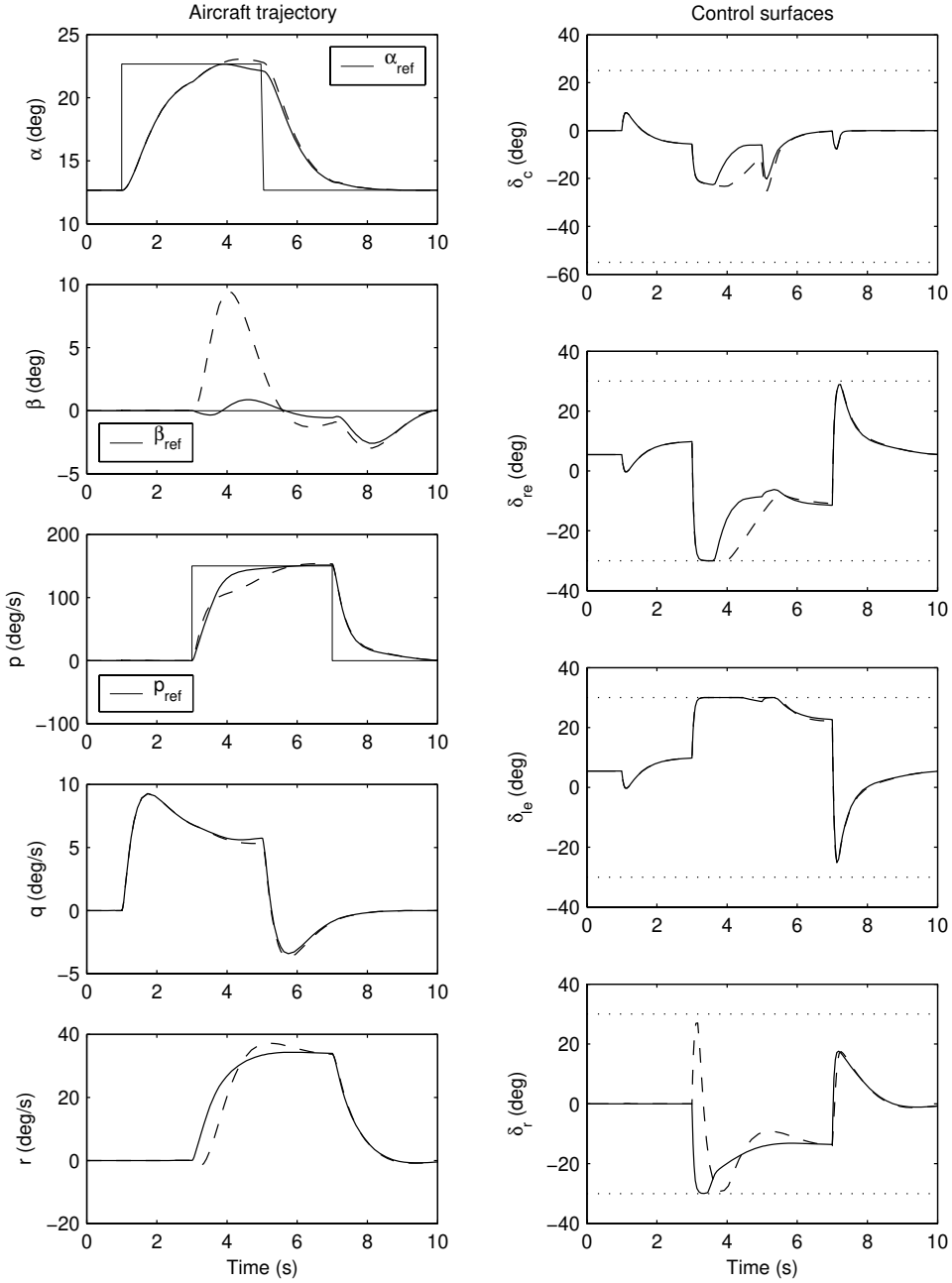


Figure 10.5: Aircraft trajectory, x (left), and control surface positions, δ (right), for control design 2 with $W_v = \mathrm{diag}(1, 1, 100)$ (solid) and $W_v = \mathrm{diag}(1, 1, 1)$ (dashed), respectively. In the first case, yaw control is prioritized resulting in a better sideslip response.

distributing the control effect among the actuators. Theoretically, this is an interesting result in itself since it ties together two useful tools for aircraft control design.

There are also practical implications. Given an existing LQ controller, we have shown how to split this into a new LQ controller, governing the closed loop dynamics, and a control allocator, distributing the control effect among the actuators. In the control allocator, actuator position and rate constraints can be considered, so that when one actuator saturates, the remaining actuators can be used to make up for the loss of control effect, if possible.

Unfortunately, not all systems with more actuators than controlled variables fit into the considered class of systems. In some cases, reasonable approximations may enable for the proposed modularity to be achieved, as in our flight control example. In such cases, the benefits of using a separate control allocator must be weighed against the disadvantages of using an approximate model for control design.

Part III

Evaluation

Chapter 11

Application to Flight Control

In the two preceding parts, backstepping and control allocation has been used to develop design tools applicable to flight control. In this chapter we use a selection of these tools to build a simplified flight control system.

In Section 11.1, the proposed control system is presented, and its different building blocks are described in detail. How to select the design variables of the control system is the topic of Section 11.2. The qualitative properties of the proposed controller are discussed in Section 11.3, and in Section 11.4 simulation results based on the ADMIRE model are presented. Conclusions are drawn in Section 11.5.

11.1 Building the Control System

The proposed control system configuration is shown in Figure 11.1. Given the (modified) reference signal $\bar{r}$, and the measured state vector x, the control laws compute the aerodynamic moment coefficients to be produced in pitch, roll, and yaw. In the control allocation module, this virtual control input is distributed among the available control surfaces so that the desired moments will be produced. The output of this block are the demanded control surface deflections, u. Both the control laws and the control allocator are designed based on a nominal aircraft model. To achieve robustness against model errors and external disturbances, integral control is used to modify the reference signal entering the control laws.

We will now present these three building blocks in detail and write down the computations involved for each of them.

11.1.1 Control Laws

The control laws are responsible for the regulation task of the control system. Here we will use the backstepping control laws derived in Chapter 4 for maneuvering flight control and for flight path angle control. For maneuvering flight control, the

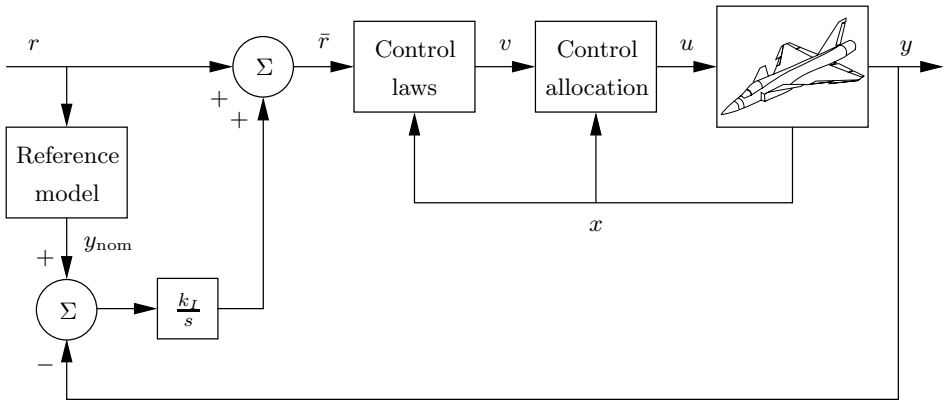


Figure 11.1: Control system configuration.

control objectives are

$$\begin{aligned} p_w &= p_w^{\text{ref}} \\ \alpha &= \alpha^{\text{ref}} \\ \beta &= 0 \end{aligned}$$

and for flight path angle control, the control objective is given by

$$\gamma = \gamma^{\text{ref}}$$

For maneuvering flight control, the rigid body control laws from Chapter 5 could also be used.

Speed control is handled separately in both cases. For maneuvering flight control the throttle setting is held constant at the trim value of the particular flight case. For flight path angle control, a simple proportional controller is used.

Let us now list the computations involved in evaluating the control laws in the two cases.

Maneuvering Flight Control

1. Compute the desired angular acceleration in the wind-axes frame,

$$\dot{\omega}_w^{\text{des}} = \begin{pmatrix} a_1 & a_2 & a_3 \end{pmatrix}^T$$

whose components are given by (4.41), (4.42), and (4.46):

$$\begin{aligned} a_1 &= k_p(p_w^{\text{ref}} - p_w) \\ a_2 &= -k_q(q_w + k_\alpha \cos\beta(\alpha - \alpha^{\text{ref}}) + \\ &\qquad \frac{1}{mV_T}(-L(\alpha^{\text{ref}}, \beta, p, q, r, \delta) + mg_3(\alpha^{\text{ref}}, \beta, \theta, \phi))) \\ a_3 &= k_r(-r_w + k_\beta \beta + \frac{1}{V_T} g \cos\theta \sin\phi) \end{aligned} \tag{11.1}$$

where k_p, k_q, k_r, k_α, and k_β are design parameters. In the evaluation of f_α in (4.42), the thrust force component is neglected since F_T is not measured. Further, the dependencies on the angular rates as well as on the control surface deflections are included in the computation of the lift force, although they were neglected in the control design. Including these effects reduces the workload of the integrator.

2. Compute the net torque, in the body-fixed coordinate frame, required to produce the desired angular acceleration, see (4.38):

$$T = IT_{bw}(\alpha, \beta)\dot{\omega}_w^{\text{des}} + \omega \times I\omega \tag{11.2}$$

Neglecting the torque contribution from the engine, T contains the aerodynamic moments to be produced in roll, pitch, and yaw.

3. Compute the aerodynamic moment coefficients to be produced, see (2.12):

$$v = \begin{pmatrix} C_l & C_m & C_n \end{pmatrix}^T = \frac{1}{S\bar{q}} \operatorname{diag}(b, \bar{c}, b)^{-1} T$$

Flight Path Angle Control

1. Compute the desired pitch angular acceleration, see (4.48),

$$a = -k_3(q + k_2(\alpha - \alpha_0 + (1 + k_1)(\gamma - \gamma^{\text{ref}})))$$

where k_1, k_2, and k_3 are design parameters.

2. Compute the pitching moment required to produce this angular acceleration, see (4.49):

$$M = I_y a$$

Again, the thrust force component has been neglected.

3. Compute the corresponding aerodynamic coefficient, see (2.12), and set the rolling and yawing moment coefficients to zero:

$$v = \frac{1}{S\bar{q}\bar{c}} \begin{pmatrix} 0 & M & 0 \end{pmatrix}^T$$

11.1.2 Control Allocation

The control allocator computes commanded control surface deflections, considering actuator position and rate limits, such that the desired aerodynamic moments coefficients are produced. Here we will use the dynamic control allocation method from Chapter 9 to achieve different control distributions at different frequencies.

1. Compute the control effectiveness matrix from the aerodynamic coefficients $C_M = (C_l \quad C_m \quad C_n)^T$, see (7.14):

$$B = \frac{\partial C_M}{\partial \delta}(\alpha, \beta, p, q, r, \delta_0)$$

   The choice of δ_0 is discussed in Section 11.2.2.

2. Compute the modified virtual control input, see (7.15):

$$\bar{v} = v - C_M(\alpha, \beta, p, q, r, \delta_0) + B\delta_0$$

3. Determine the set of feasible control surface deflections with respect to position and rate constraints, see (7.7),

$$\underline{u}(t) = \max\{u_{\min}, u(t-T) + T\rho_{\min}\}$$
$$\overline{u}(t) = \min\{u_{\max}, u(t-T) + T\rho_{\max}\}$$

   where T is the sampling time.

4. Perform l_2-optimal dynamic control allocation, see (9.1):

$$u(t) = \arg\min_{u(t)\in\Omega} \left\|W_1(u(t) - u_s(t))\right\|_2^2 + \left\|W_2(u(t) - u(t-T))\right\|_2^2$$
$$\Omega = \arg\min_{\underline{u}(t)\le u(t)\le \overline{u}(t)} \left\|W_v(Bu(t) - \bar{v}(t))\right\|_2$$

   Here, W_1, W_2, W_v, and u_s are design variables.

11.1.3 Integral Control

The integral control part of the control system is included to achieve output regulation in the presence of model errors and constant external disturbances acting on the system. In the proposed configuration, integral control is used to modify the reference signal, r, rather than the control input, u. This unconventional design choice was inspired by Su et al. (2001). Note that if the control laws are linear in $\bar{r}$, this is equivalent to adding the integral action to the virtual control input, v.

Theoretically, this strategy guarantees that constant disturbances appearing at the input of the control laws, thus contributing to $\bar{r}$, are suppressed. In practice, this strategy may also achieve output regulation when disturbances are introduced

elsewhere in the system, see Section 6.4.1, and when the true system differs from the nominal model used for control design, see Su et al. (2001).

The integrator input is given by the difference between the output of a reference model of the closed loop system, and the actual values of the controlled variables. This way, the integral controller is only active when the output signal y deviates from its predicted value. Thus, the following steps are to be performed in the integral control part of the control system:

1. Filter the reference signal r through a reference model of the nominal closed loop system. This gives the nominal output signal y_{nom}.
2. Integrate the output error and adjust the reference signal according to

$$\bar{r} = r + k_I \int_0^t (y_{\text{nom}}(\tau) - y(\tau))d\tau$$

where k_I contains the integrator gains for the different controlled variables.

11.2 Tuning the Control System

Each block in the control system in Figure 11.1 contains a number of design parameters to be tuned. In this section we discuss how to select these parameters to achieve a desirable total system behavior.

11.2.1 Control Laws

For maneuvering flight control, the design parameters are the feedback gains k_p, k_q, k_r, k_α, and k_β. If the flight envelope does not include high angles of attack, it was shown in Section 4.3.1 that the conditions $k_\alpha > 0$, $k_\beta > 0$, $k_p > 0$, $k_q > k_\alpha$, and $k_r > k_\beta$ guarantee closed loop stability.

A systematic way to select these parameters is to use reference models of the desired responses,

$$\begin{aligned} p_w &= \frac{1}{s\tau_{\text{roll}} + 1} p_w^{\text{ref}} \\ \alpha &= \frac{\omega_{n,\text{sp}}^2}{s^2 + 2\zeta_{\text{sp}}\omega_{n,\text{sp}} + \omega_{n,\text{sp}}^2} \alpha^{\text{ref}} \\ \beta &= \frac{\omega_{n,\text{dr}}^2}{s^2 + 2\zeta_{\text{dr}}\omega_{n,\text{dr}} + \omega_{n,\text{dr}}^2} \beta^{\text{ref}} \end{aligned} \tag{11.3}$$

where τ_{roll} is the roll time constant, ζ_{sp} and $\omega_{n,\text{sp}}$ are the damping and the natural frequency of the short period mode, and ζ_{dr} and $\omega_{n,\text{dr}}$ are the damping and the natural frequency of the dutch roll mode. Reference models of this kind are frequently used for flight control design, see, e.g., Ståhl Gunnarsson (1999).

Comparing the roll reference model with the true closed loop dynamics

$$\dot{p}_w = k_p(p_w^{\text{ref}} - p_w)$$

found by combining (4.37a) with (4.41), gives

$$k_p = \frac{1}{\tau_{\text{roll}}}$$

For the angle of attack, the linearized closed loop dynamics are found by combining (4.37b)–(4.37c) with (4.41), which gives

$$\begin{aligned}\dot{\alpha} &= \tilde{q}_w + a\alpha \\ \dot{\tilde{q}}_w &= -k_q(\tilde{q}_w + k_\alpha(\alpha - \alpha^{\text{ref}}))\end{aligned}$$

where $a = f'_\alpha(\alpha^{\text{ref}})$ and $\tilde{q}_w = q_w + f_\alpha(\alpha^{\text{ref}})$. Comparing with the reference model above gives

$$\begin{aligned}k_q &= 2\zeta_{\text{sp}}\omega_{n,\text{sp}} + a \\ k_\alpha &= \frac{\omega_{n,\text{sp}}^2}{k_q} + a\end{aligned}$$

Similarly, for sideslip regulation we get

$$\begin{aligned}k_r &= 2\zeta_{\text{dr}}\omega_{n,\text{dr}} + b \\ k_\beta &= \frac{\omega_{n,\text{dr}}^2}{k_r} + b\end{aligned}$$

where $b = f'_\beta(0)$ assuming that $\beta^{\text{ref}} = 0$.

Note that the parameters a and b depend on the flight case. This can be handled either by selecting fixed feedback gains based on a representative flight case in the middle of the flight envelope, or by scheduling the feedback gains with the flight case using the analytical expressions above.

Similar rules can be developed for the feedback gains defining the flight path angle control law.

11.2.2 Control Allocation

The design parameters for dynamic control allocation are given by the weighting matrices W_1, W_2, and W_v, and the control distribution at steady state, u_s. Another design choice to be made regards δ_0 which is used for linearizing the aerodynamic moment coefficients.

A systematic method to select u_s was presented in Section 9.2.3. W_v determines which aerodynamic moment to prioritize in case the demanded virtual control input is infeasible with respect to the actuator constraints. For example, selecting $W_v = \text{diag}(1, 10, 100)$ puts the highest priority on the yawing moment, and the lowest priority on the rolling moment.

A similar strategy can be used to select W_1 and W_2 whose entries penalize actuator position errors (relative to u_s) and actuator rates, respectively. For example, a large rate penalty for a certain actuator means that this actuator will be used to

produce low frequency virtual control input if there are other actuators that can be used to produce high frequency virtual control input. However, which types of transfer functions from v to u that can be achieved, and how to obtain a certain frequency characteristic for an actuator, remains to be investigated.

As discussed in Section 7.2.2, δ_0 can be selected as the previously commanded control input, $u(t-T)$. With this choice the local variations in control effectiveness are considered. However, if the control effectiveness of an aerodynamic surface does not vary monotonically with the surface deflection, the linearization will be zero at some point. This can cause the B matrix to be rank deficient which may lead to numerical problems in the control allocation. A more robust choice is to linearize around some fixed value of δ_0, for example $\delta = 0$, and let the integral controller handle the resulting model error.

11.2.3 Integral Control

For the integral controller, the design variables are the reference model of the nominal aircraft response and the integrator gains in k_I.

The reference models for the different controlled variables should be designed to reproduce the characteristics of the nominal closed loop system. These models may include nonlinearities as well as the previously neglected dynamics in the actuators and in the sensors. A straightforward choice is to pick the same linear reference models that were used for tuning the control law parameters, see (11.3).

As for the feedback gains in Section 11.2.1, the integrator gains in k_I could be selected using pole placement techniques. However, this issue remains to be investigated further. Su et al. (2001) derive an upper bound for k_I in order for the closed loop system to be stable. This bound relies on the characteristics of a Lyapunov function for the closed loop system without the integrator.

11.3 Control System Properties

Let us now evaluate the proposed control system structure from a qualitative perspective.

- *Which particular system properties are dealt with?*

The backstepping control laws are based on the nonlinear rigid body equations (2.6). Hence the couplings between longitudinal and lateral motion, including inertia coupling (see Section 2.3.1), are explicitly considered. Further, the design parameters can be selected to achieve stability also at high angles of attack, see Section 4.3.1.

The control allocation algorithm takes actuator position and rate limits into account. If an actuator saturates, the control effort is redistributed among the remaining actuators to make up the difference. Further, using dynamic control allocation allows for the bandwidths of the actuators to be considered in the sense that slow actuators can be fed with low frequency position command signals.

- *Which system properties are not considered?*

All actuator and sensor dynamics are neglected. The dependence of the aerodynamic forces on the control surface positions is also essentially neglected. Including this dependence in the computation of the lift force as in (11.1) means that the influence of the control surfaces at steady state is accounted for in the control laws, rather than leaving this task to the integral controller. However, no stability guarantees can be made since this effect was not considered in the control design.

- *What aerodynamic data does the control system rely on?*

In the angle of attack control laws, the value of lift coefficient needs to be computed, and for control allocation, the values and the first order derivatives of the aerodynamic moment coefficients are required. Further, the proposed method for tuning the feedback gains depends on the derivatives of C_L and C_Y with respect to α and β, respectively. Consequently, it is fairly straightforward to adjust the control system if an updated set of aerodynamic data for the aircraft is obtained.

- *What design requirements can be handled?*

The tuning method in Section 11.2.1 allows the designer to specify the characteristics of the roll mode, the short period mode, and the dutch roll mode. The control laws can also be tuned to achieve stability at high angles of attack, see Section 4.3.1.

The dynamic control allocator handles requirements regarding control surface activity, including frequency apportioning. Further, since the control allocation is performed at each sample, new information about the status of the control surfaces (jammed, flee-floating, etc.) can be directly incorporated. This can be viewed as handling certain safety requirements.

- *What is the computational complexity of the control system?*

The main computational burden consists of performing the control allocation, where at each sampling instant a constrained least squares problem must be solved. In Chapter 8 we developed active set methods for this purpose. Using active set methods, a trade-off can be made between the computation time and the optimality of the solution. Since these methods have not yet been implemented in any compiling language, no realistic timing data are available.

- *What is the implementational complexity of the control system?*

Implementing the control system requires a constrained least squares solver to be programmed. The active set methods in Chapter 8 are fairly complex to implement. The pseudoinverse methods reviewed in Section 7.4.3 are straightforward to implement but may not perform as well, as demonstrated in Section 8.5.

Further, if the aerodynamic data is stored in look-up tables, numerical differentiation is needed to perform the linearization of the aerodynamic moment coefficients for control allocation.

11.4 ADMIRE Simulations

The control system in Figure 11.1 has been implemented in MATLAB/Simulink and integrated with the ADMIRE model (see Section 2.4), replacing the existing flight control system. The properties of the angle of attack control law and the flight path angle control law at high angles of attack were investigated in Section 4.3.3. In this section we investigate other properties of the maneuvering flight control laws.

11.4.1 Design Parameters

The design parameters in the control system were selected as follows.

Control Laws

For maneuvering flight control, the feedback gains were selected as described in Section 11.2.1. The requirements

$$\tau_{\text{roll}} = 0.4 \qquad \begin{matrix} \zeta_{\text{sp}} = 0.9 & \zeta_{\text{dr}} = 0.7 \\ \omega_{n,\text{sp}} = 4.0 & \omega_{n,\text{dr}} = 3.0 \end{matrix}$$

at Mach 0.5, 1000 m, give

$$k_p = 2.5 \qquad \begin{matrix} k_q = 5.7 & k_r = 3.9 \\ k_\alpha = 1.3 & k_\beta = 2.0 \end{matrix}$$

For flight path angle control,

$$k_1 = 1.0 \qquad k_2 = 1.3 \qquad k_3 = 5.7$$

were selected, based on the angle of attack control parameters above.

Control Allocation

The parameters governing the dynamic properties of the of the control allocator were selected as in the design example in Section 9.3, i.e.,

$$\begin{aligned} W_1 &= \text{diag}(2, 2, 2, 2, 2, 2, 2) \\ W_2 &= \text{diag}(5, 5, 10, 10, 10, 10, 10) \\ u_s(t) &= S\bar{v}(t) \end{aligned}$$

with S determined by (9.9). This causes the canards to be used only for high frequencies so that only the elevons and the rudder are deflected at steady state, see Figure 9.3. The axis prioritization was selected as

$$W_v = \text{diag}(1, 10, 100)$$

The aerodynamic moment coefficients are linearized around $\delta_0 = \delta(t - T)$, which is computed by incorporating a model of the actuator dynamics in the control system. Algorithm 8.3 (weighted least squares) is used to solve the control allocation problem at each sample. This choice was motivated by the fact that WLS is numerically stable even when the control effectiveness matrix B becomes rank deficient.

To reduce the number of variables to be plotted, the two canards are ganged ($\delta_{rc} = \delta_{lc} = \delta_c$), as well as the two right elevons ($\delta_{roe} = \delta_{rie} = \delta_{re}$) and the two left elevons ($\delta_{loe} = \delta_{lie} = \delta_{le}$), in all simulations except the control surface failure simulation where only the canards were ganged.

Integral Control

For maneuvering flight control, the closed loop reference model was selected as in (11.3) with the coefficients given above. For flight path angle control, the reference model was selected as the linearization of the nominal closed loop system at Mach 0.3, 1000 m. In both cases the integrator gain was set to

$$k_I = 0.2$$

for each controlled variable.

11.4.2 Simulation Results

- *Variations with speed and altitude:* Figure 11.2 illustrates the closed loop behavior for three different initial conditions: Mach 0.5, 1000 m (which is the flight case the parameters were tuned for), Mach 0.22, 200 m (low speed and altitude), and Mach 0.8, 5000 m (high speed and altitude). To be able to compare the responses, the angle of attack at trimmed flight, α_0, has been subtracted from the α response. The controlled variables α, β, and p_w essentially behave the same in all cases.

- *Decoupling:* Figure 11.3 illustrates the decoupling properties of the controller. Recall that the backstepping control laws were derived considering the dynamics in roll, pitch, and yaw, as three separate systems. In the figure we see that the α and p_w responses are virtually the same regardless of the commanded value of the other variable.

- *Control surface hardover:* Figure 11.4 shows the results of a control surface hardover. At $t = 1$ s, a failure occurs in the right outer elevon which is driven to -30 degrees. This causes an undesired rolling, pitching and yawing motion to which the control system reacts rather slowly. At $t = 1.5$ s, the fault is assumed to be detected and the information is sent to the control allocator which is able to compensate for the hardover using the remaining control surfaces. Note that the roll maneuver commanded at $t = 4$ s is almost perfectly executed.

- *Inertia coupling:* Figure 11.5 shows the effects of not compensating for inertia coupling. The dashed line corresponds to neglecting the terms $(I_x - I_z)pr + I_{xz}(p^2 + r^2)$ in the pitch rate dynamics in (2.14) when the desired torque to be produced in computed in (11.2). Recall from Section 2.3.1 that these terms induce a pitching motion. When this effect is not accounted for, the angle of attack increases quickly as the roll command is applied at $t = 2$ s.

- *Robustness:* Figure 11.6 and Figure 11.7 illustrate the robustness of the system for two different types of model error. In both figures, the response of the perturbed system is compared with the nominal system response from Figure 11.2.

  Figure 11.6 shows the results of a 30% increase of the aircraft mass, m. This leads to an overshoot in α, but as the integral controller starts to work, the control error goes to zero.

  Figure 11.7 shows the effects of a 33% decrease, and a 50% increase, in the modeled control efficiency compared to the true efficiency. This model error is obtained by multiplying C_M used in Section 11.1.2 by 0.67 and 1.5, respectively. Form the figure we see that underestimating the control efficiency yields a system response that is faster than desired, but not oscillatory. Overestimating the control efficiency gives a slower system response. Due to the integral controller, this leads to an overshoot in the controlled variables α and p_w.

- *High angles of attack:* The stability properties at high angles of attack were investigated in Section 4.3.3.

11.5 Conclusions

In this chapter we have shown how to combine the backstepping control laws from Part I with the control allocation tools from Part II to construct a flight control system capable of dealing with the nonlinear dynamics of an aircraft as well as actuator redundancy and actuator position and rate constraints.

The control laws can be systematically tuned based on the requirements on the different modes of the aircraft (roll, short period, and dutch roll). This does not require any insight into the backstepping method used to construct the control laws. The control allocator can also be systematically tuned using weighting matrices to achieve the desired control distribution among the control surfaces.

Simulations indicate that the controller gives a similar response for different flight cases and possesses certain robustness properties. Further, considering the full nonlinear aircraft dynamics enables the controller to efficiently counteract the inherent couplings between lateral and longitudinal motion, including inertia coupling. A final important conclusion is that the results indicate that the approximations made during the control design were reasonable.

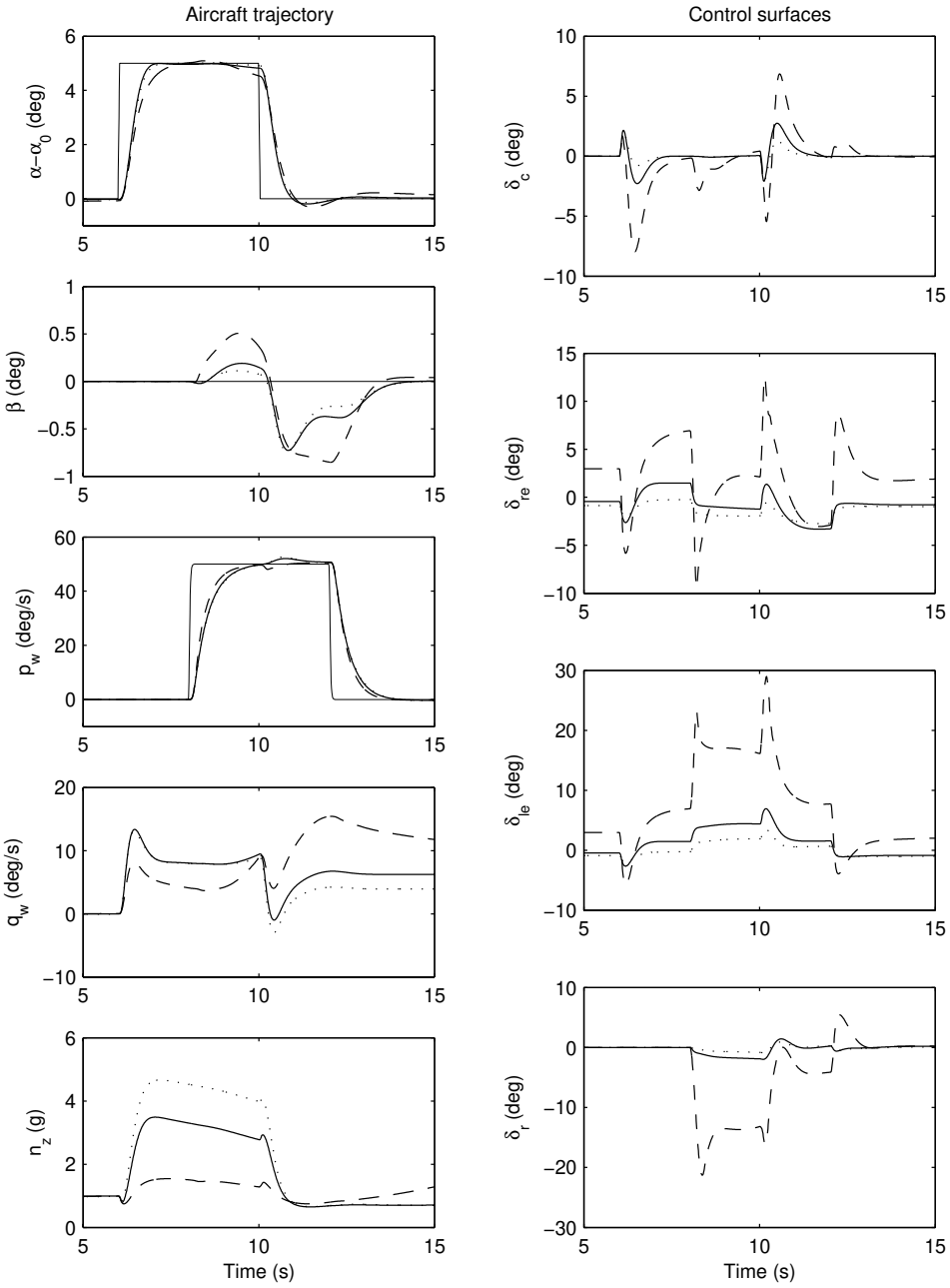


Figure 11.2: Simulation results when the same control commands (thin lines) are applied for three different flight cases: Mach 0.5, 1000 m (solid), Mach 0.22, 200 m (dashed), and Mach 0.8, 5000 m (dotted).

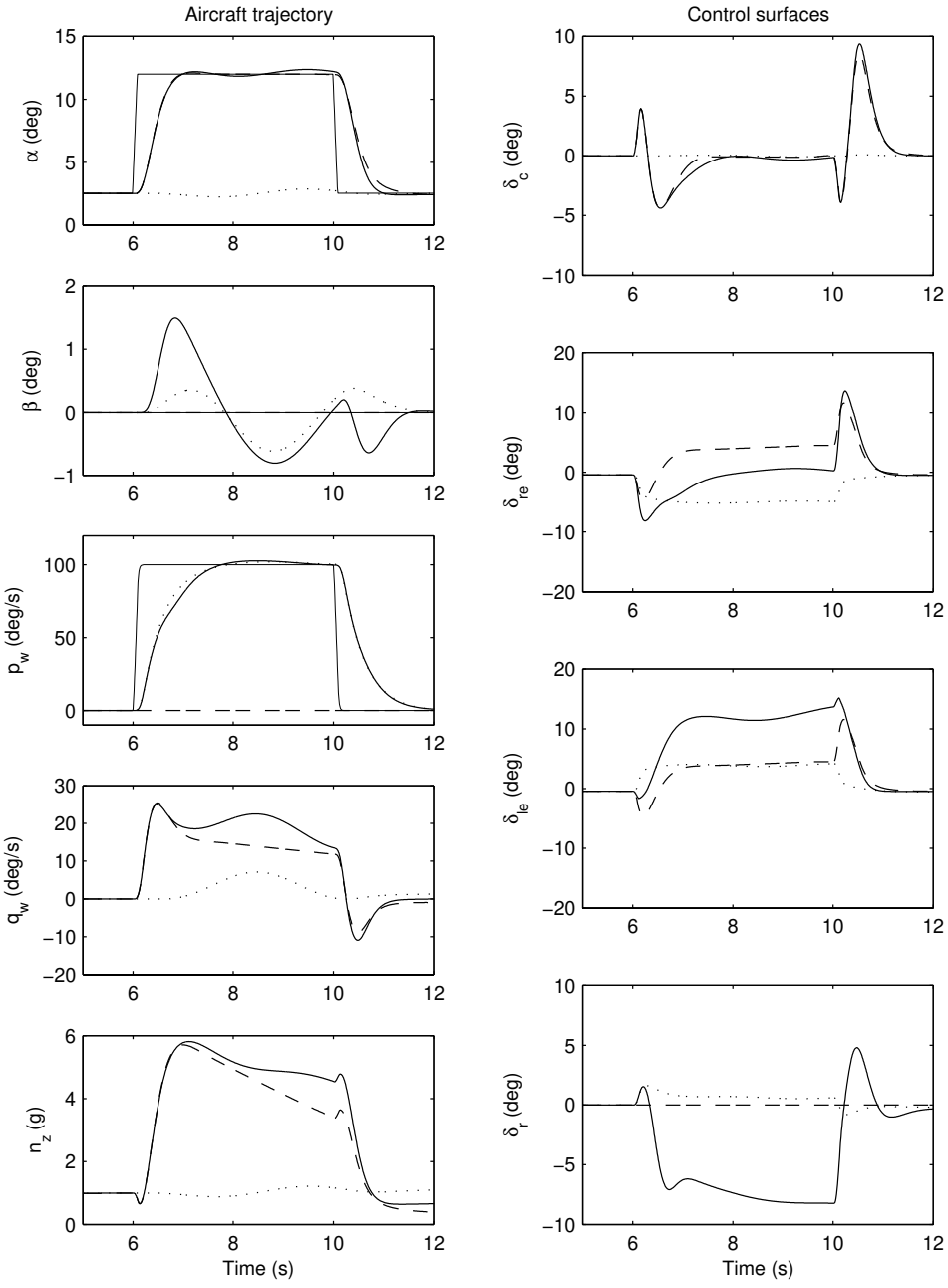


Figure 11.3: Illustration of the decoupling properties of the controller: commanded values (thin lines), aircraft response for a pure pitch command (dashed), a pure roll command (dotted), and both commands applied simultaneously (solid).

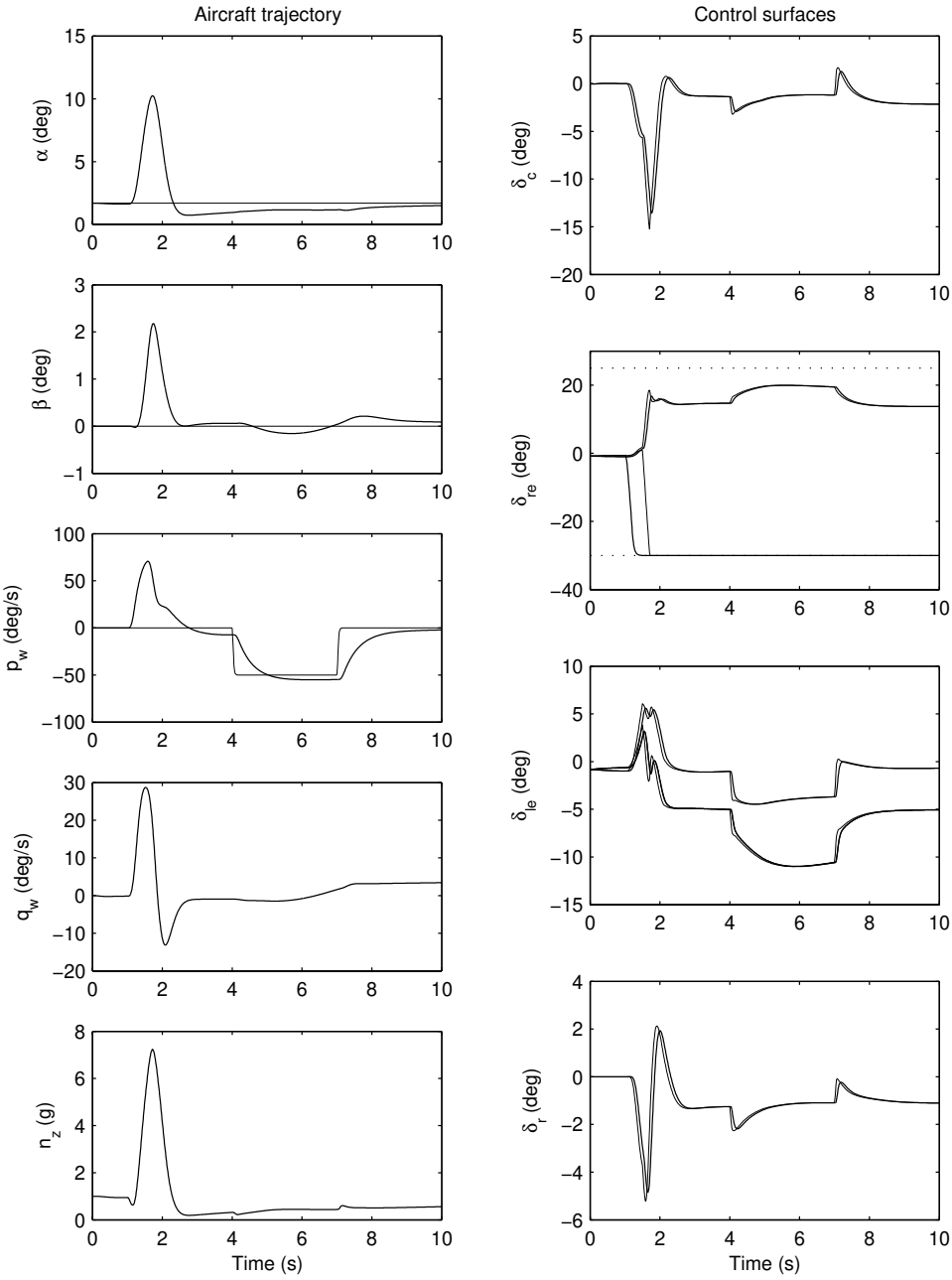


Figure 11.4: Results from a control surface hardover at $t = 1$ s. Left: commanded (thin lines) and actual (thick lines) aircraft response. Right: Commanded (thin lines) and actual (thick lines) control surface deflections.

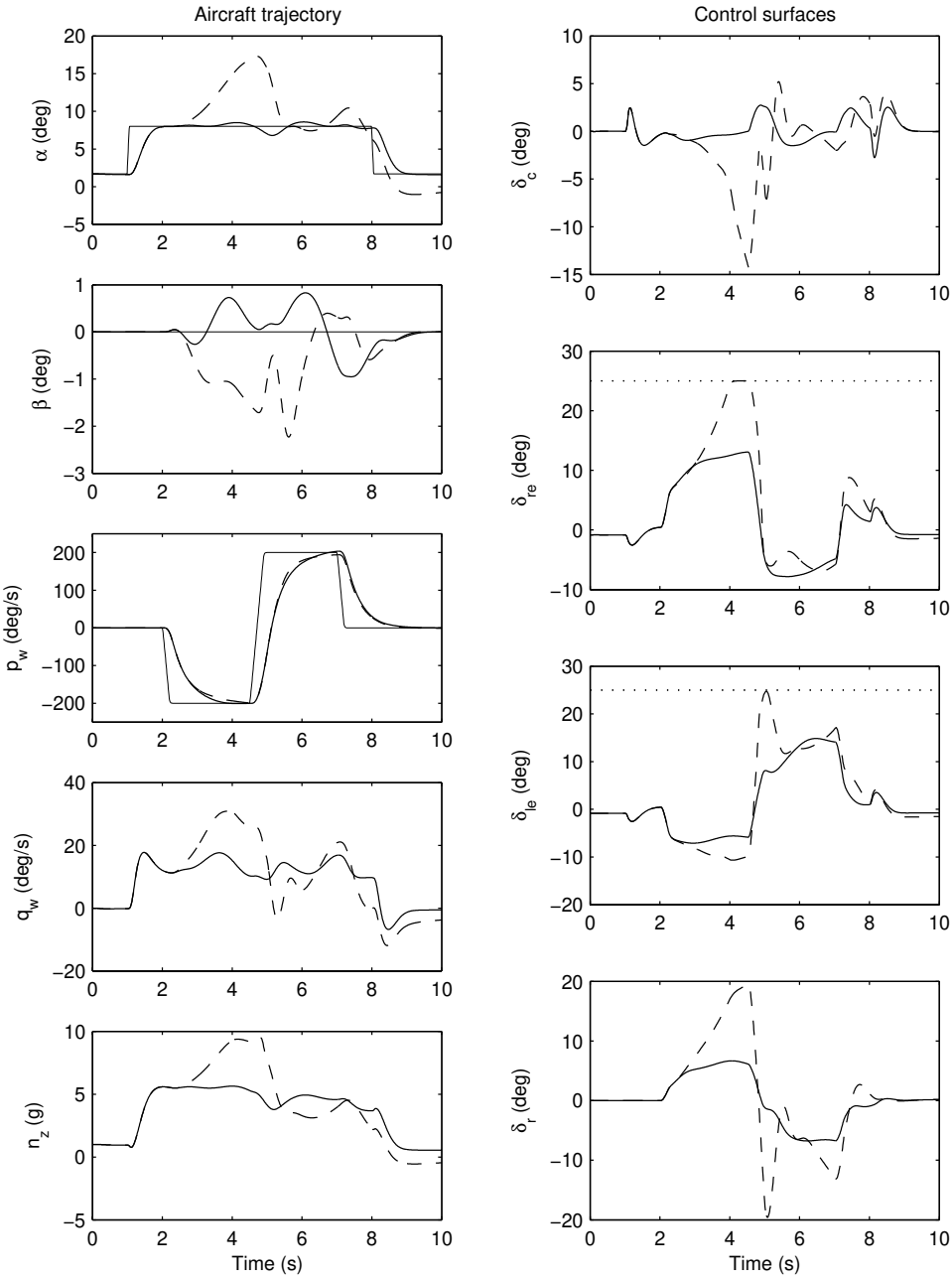


Figure 11.5: Illustration of the effects of inertia coupling: control commands (thin lines) and the resulting aircraft response with (solid) and without (dashed) compensation for inertia coupling, respectively.

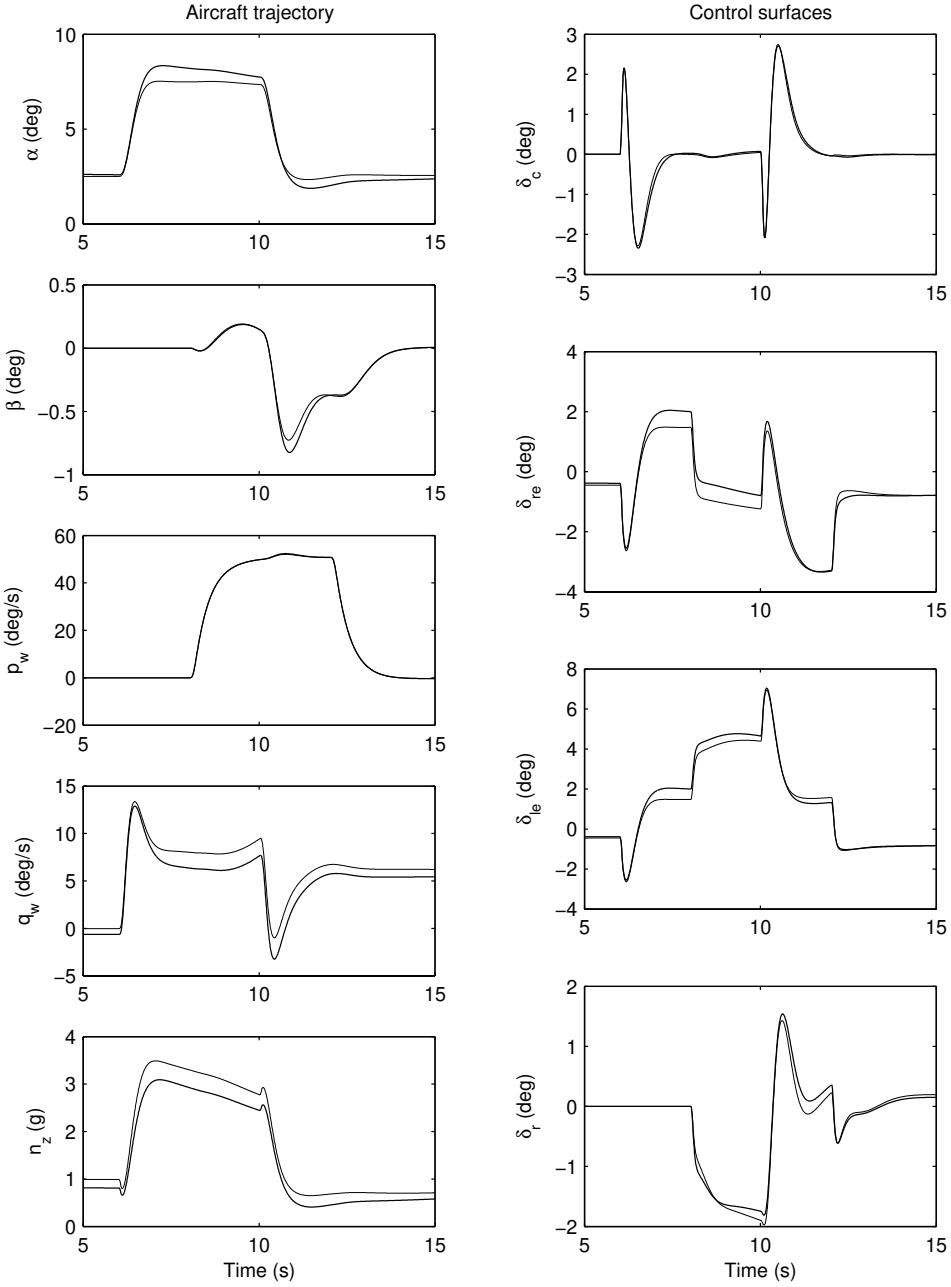


Figure 11.6: Robustness towards mass perturbations: nominal aircraft response (thin lines) and response for a 30% increase in the aircraft mass (thick lines).

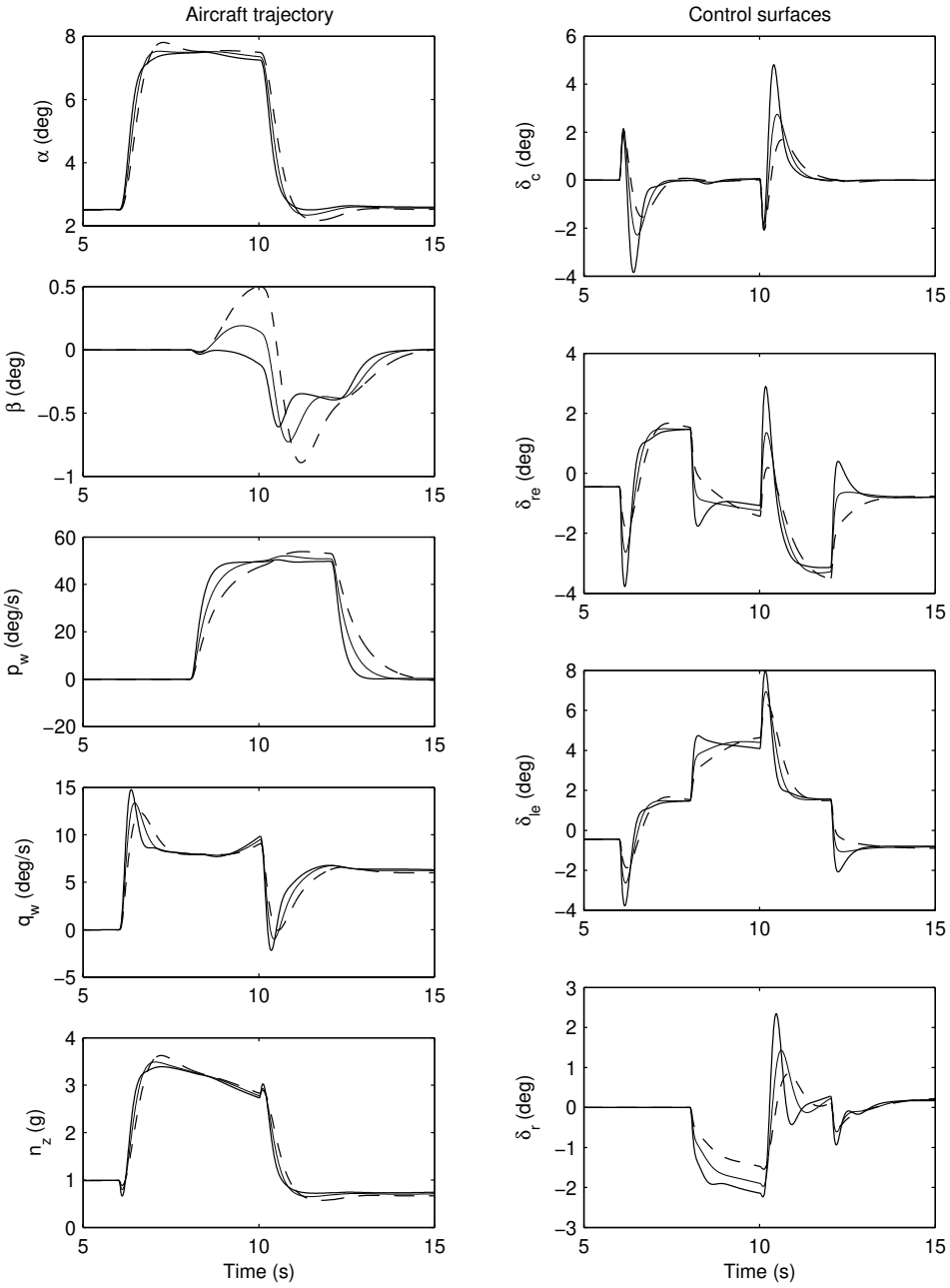


Figure 11.7: Robustness towards control efficiency perturbations: nominal aircraft response (thin lines) along with the response for a 33% decrease (solid) and a 50% increase (dashed) in the modeled control efficiency.

Chapter 12

Conclusions

Let us end this thesis with some overall conclusions.

The use of backstepping for flight control design seems promising. Compared with feedback linearization, backstepping requires the same model approximations to be made, while the resulting control laws can be designed to rely on less precise aerodynamic model information. An advantage over linear flight control is that a nonlinear framework allows for effects like inertia coupling to be explicitly accounted for. Under some assumptions, backstepping can also be applied to the control of a general rigid body, subject to external forces and moments.

A disadvantage of backstepping is that the design complexity grows rapidly with the system order which makes it difficult to include, e.g., actuator dynamics. Another drawback is that state feedback is required.

For control allocation, the least squares formulation used in this thesis constitutes a nice analytical framework since it provides a smooth transition from the nonsaturated case, where a linear control distribution is achieved, to the case where some of the actuators are saturated. It also allows for filtering to be incorporated to shape the frequency responses of the control signals.

To solve the control allocation problem in real-time, classical active set methods from numerical optimization provide a possible alternative to the pseudoinverse methods frequently used today. They are similar in terms of computational complexity, but unlike the pseudoinverse methods, active set methods are guaranteed to find the optimal solution. Further, a feasible suboptimal solution is produced in each iteration, which can be used if there is not enough time to compute the optimal solution.

A final conclusion is that control allocation may be useful also for control of linear systems. Compared with linear quadratic design, control allocation offers the same degree of freedom to shape the control distribution for those systems that both techniques can be applied to. The advantage of using control allocation is that actuator constraints can be considered which means that the control capabilities can be fully utilized.

Appendix A

Aircraft Nomenclature

State variables

Symbol	Unit	Definition
α	rad	angle of attack
β	rad	sideslip angle
γ	rad	flight path angle
$\mathbf{V} = \mathbf{e}_b(u \quad v \quad w)^T$		velocity vector
u	m/s	longitudinal velocity
v	m/s	lateral velocity
w	m/s	normal velocity
V_T	m/s	total velocity
$\boldsymbol{\omega} = \mathbf{e}_b\omega = \mathbf{e}_w\omega_w$		angular velocity vector
$\omega = (p \quad q \quad r)^T$		body-axes components
$\omega_w = (p_w \quad q_w \quad r_w)^T$		wind-axes components
p	rad/s	roll rate
q	rad/s	pitch rate
r	rad/s	yaw rate
$\mathbf{p} = \mathbf{e}_i(p_N \quad p_E \quad -h)^T$		position vector
p_N	m	position north
p_E	m	position east
h	m	altitude
$\Phi = (\phi \quad \theta \quad \psi)^T$		orientation in Euler angles
ϕ	rad	roll angle
θ	rad	pitch angle
ψ	rad	yaw angle
n_z	g	load factor, normal acceleration
n_{zp}	g	pilot load factor

Coordinate frames

Symbol	Definition
$\mathbf{e}_i = (\hat{\mathbf{x}}_i \;\; \hat{\mathbf{y}}_i \;\; \hat{\mathbf{z}}_i)$	inertial, Earth-fixed frame
$\mathbf{e}_b = (\hat{\mathbf{x}}_b \;\; \hat{\mathbf{y}}_b \;\; \hat{\mathbf{z}}_b)$	body-fixed frame
$\mathbf{e}_w = (\hat{\mathbf{x}}_w \;\; \hat{\mathbf{y}}_w \;\; \hat{\mathbf{z}}_w)$	wind-axes frame

Control surface deflections

Symbol	Unit	Definition
δ_{rc}	rad	right canard deflection
δ_{lc}	rad	left canard deflection
δ_{roe}	rad	right outer elevon deflection
δ_{rie}	rad	right inner elevon deflection
δ_{lie}	rad	left inner elevon deflection
δ_{loe}	rad	left outer elevon deflection
δ_r	rad	rudder deflection

Forces and moments

Symbol	Unit	Definition
g	$\mathrm{m/s^2}$	gravitational acceleration
F_T	N	engine thrust force
$D = \bar{q}SC_D$	N	drag force
$L = \bar{q}SC_L$	N	lift force
$Y = \bar{q}SC_Y$	N	side force
$\bar{L} = \bar{q}SbC_l$	Nm	rolling moment
$M = \bar{q}S\bar{c}C_m$	Nm	pitching moment
$N = \bar{q}SbC_n$	Nm	yawing moment

Aircraft data

Symbol	Unit	Definition
m	kg	aircraft mass
$I = \begin{pmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{pmatrix}$	$\mathrm{kg\,m^2}$	aircraft inertial matrix
S	$\mathrm{m^2}$	wing planform area
b	m	wing span
$\bar{c}$	m	mean aerodynamic chord
z_{TP}	m	z_b-position of engine thrust point
x_P	m	x_b-position of the pilot

Atmosphere

Symbol	Unit	Definition
ρ	$\mathrm{kg/m^3}$	air density
$\bar{q}$	$\mathrm{N/m^2}$	dynamic pressure

Appendix B

Some Results from Matrix Theory

In this appendix, some facts and results from matrix theory are presented. The material can be found in many textbooks, and for this review, Golub and Van Loan (1996), Björck (1996), Zhang (1999), Zhou et al. (1996), and Strang (1980) were consulted.

The two main topics are

- matrix decompositions: QR (used in Chapter 8) and SVD (Chapters 9 and 10)
- the pseudoinverse (used throughout Part II)

B.1 Norms and Singular Values

Let us first review the standard definitions of vector and matrix norms, and the singular values of a matrix.

Let x be an $n \times 1$ vector. The *p-norm* of x is defined as

$$\left\|u\right\|_p = \left(\sum_{i=1}^{m} \left|u_i\right|^p\right)^{1/p} \quad \text{for } 1 \leq p \leq \infty$$

For $p = 1, 2, \infty$ we get

$$\left\|x\right\|_1 = \sum_{i=1}^{n} \left|x_i\right|$$

$$\left\|x\right\|_2 = \sqrt{\sum_{i=1}^{n} x_i^2} = \sqrt{x^T x}$$

$$\left\|x\right\|_\infty = \max_{1 \leq i \leq n} \left|x_i\right|$$

Let A be an $m \times n$ matrix. The *induced p-norm* of A is defined as

$$\|A\|_p = \sup_{x \neq 0} \frac{\|Ax\|_p}{\|x\|_p}$$

The *singular values* of A are given by the positive square roots of the eigenvalues of $A^T A$,

$$\sigma_i(A) = \sqrt{\lambda_i(A^T A)}, \ i = 1, \ldots, n$$

ordered such that $\sigma_1 \geq \cdots \geq \sigma_r > \sigma_{r+1} = \cdots = \sigma_n = 0$ where $r = \text{rank}(A)$[1]. With this, the 2-norm of A satisfies

$$\|A\|_2^2 = \sup_{x \neq 0} \frac{\|Ax\|_2}{\|x\|_2} = \sup_{x \neq 0} \frac{x^T A^T A x}{x^T x} = \overline{\lambda}(A^T A) = \overline{\sigma}(A)^2$$

where $\overline{\lambda}$ and $\overline{\sigma}$ denote the largest eigenvalue and singular value, respectively.

In the remainder of this appendix, we will drop the index for the 2-norm and use the notation

$$\|\cdot\| = \|\cdot\|_2$$

B.2 Matrix Decompositions

We will now present three useful ways of decomposing, or factorizing, a matrix:

- the QR decomposition
- the singular value decomposition (SVD)
- the square root of a matrix

A frequently used matrix property is that $A \in \mathbb{R}^{n \times n}$ is said to be *orthogonal* if $A^T A = A A^T = I$.

The QR Decomposition

Let $A \in \mathbb{R}^{m \times n}$ where $m \geq n$. Then the *QR decomposition* of A is given by

$$A = QR$$

where $Q \in \mathbb{R}^{m \times m}$ is orthogonal and $R \in \mathbb{R}^{m \times n}$ is upper triangular. If $m \geq n$ and $\text{rank}(A) = n$ (full column rank), then

$$A = \begin{pmatrix} Q_1 & Q_2 \end{pmatrix} \begin{pmatrix} R_1 \\ 0 \end{pmatrix} = Q_1 R_1$$

[1] In the literature one can find several slightly different singular value definitions, and the resulting number of singular values may be r, $\min\{m, n\}$, m, or, as in our case, n. However, the nonzero eigenvalues remain the same.

where R_1 is nonsingular. Since Q is orthogonal, we have the relationships

$$QQ^T = Q_1Q_1^T + Q_2Q_2^T = I$$
$$Q^TQ = \begin{pmatrix} Q_1^TQ_1 & Q_1^TQ_2 \\ Q_2^TQ_1 & Q_2^TQ_2 \end{pmatrix} = \begin{pmatrix} I & 0 \\ 0 & I \end{pmatrix}$$

The QR decomposition is a useful tool for solving least squares problems.

The Singular Value Decomposition (SVD)

Let $A \in \mathbb{R}^{m\times n}$ with $\text{rank}(A) = r$. Then the *singular value decomposition (SVD)* of A is given by

$$A = U\Sigma V^T = U \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} V^T$$

where $U \in \mathbb{R}^{m\times m}$ and $V \in \mathbb{R}^{n\times n}$ are orthogonal matrices and

$$\Sigma_r = \text{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r)$$

contains the nonzero singular values of A.

The Square Root of a Matrix

Let $A \in \mathbb{R}^{n\times n}$ be symmetric and positive semidefinite. Then the SVD of A is given by $A = U\Sigma U^T$, and the we can define the *square root* of A, satisfying $A^{\frac{1}{2}}A^{\frac{1}{2}} = A$, as

$$A^{\frac{1}{2}} = U\Sigma^{\frac{1}{2}}U^T$$

where $\Sigma^{\frac{1}{2}} = \text{diag}(\sqrt{\sigma_1}, \ldots, \sqrt{\sigma_r}, 0, \ldots, 0)$. Note that $A^{\frac{1}{2}}$ also is symmetric and positive semidefinite.

B.3 The Pseudoinverse

Let $A \in \mathbb{R}^{m\times n}$ where $m \le n$ and $\text{rank}(A) = m$. Consider solving

$$Ax = y$$

for $x \in \mathbb{R}^n$, given $y \in \mathbb{R}^m$. If $m < n$, there is no unique solution. Any *generalized inverse* of A, G, satisfying $AG = I$, i.e., any right inverse of A, gives a solution

$$x = Gy$$

A common way to make the choice of x unique is to pick the minimum norm solution and solve

$$\begin{aligned} \min_x \quad & \|x\| \\ \text{subject to} \quad & Ax = y \end{aligned}$$

This problem has the unique solution

$$x = A^\dagger y \tag{B.1}$$

where

$$A^\dagger = A^T(AA^T)^{-1} \tag{B.2}$$

is the *pseudoinverse*, or the *Moore-Penrose inverse*, of A.

More generally, let A be any $m \times n$ matrix, without the restrictions above, with the SVD

$$A = U \begin{pmatrix} \Sigma_r & 0 \\ 0 & 0 \end{pmatrix} V^T$$

Then the problem

$$\begin{aligned} &\min_{x \in \Omega} \|x\| \\ &\Omega = \arg\min_x \|Ax - y\| \end{aligned}$$

where arg min gives the set of minimizing solutions, has the solution

$$x = A^\dagger y$$

where

$$A^\dagger = V^T \begin{pmatrix} \Sigma_r^{-1} & 0 \\ 0 & 0 \end{pmatrix} U$$

When $m \leq n$ and $\operatorname{rank}(A) = m$, this expression coincides with (B.2).

The pseudoinverse can also be applied when a weighted, shifted performance index is used.

Lemma B.1 *The least squares problem*

$$\begin{aligned} \min_x \quad & \|W(x - x_0)\| \\ \text{subject to} \quad & Ax = y \end{aligned}$$

where W is nonsingular, is solved by

$$\begin{aligned} x &= Fx_0 + Gy \\ F &= I - GA \\ G &= W^{-1}(AW^{-1})^\dagger \end{aligned}$$

Proof: The change of variables

$$e = W(x - x_0) \iff x = x_0 + W^{-1}e$$

gives the equivalent minimum norm problem

$$\begin{aligned} \min_{e} \quad & \|e\| \\ \text{subject to} \quad & A(x_0 + W^{-1}e) = y \iff AW^{-1}e = y - Ax_0 \end{aligned}$$

Using (B.1) we get

$$\begin{aligned} e &= (AW^{-1})^{\dagger}(y - Ax_0) \iff \\ x &= x_0 + W^{-1}(AW^{-1})^{\dagger}(y - Ax_0) \\ &= \underbrace{(I - W^{-1}(AW^{-1})^{\dagger}A)}_{F} x_0 + \underbrace{W^{-1}(AW^{-1})^{\dagger}}_{G} y \end{aligned}$$

□

Bibliography

R. J. Adams, J. M. Buffington, and S. S. Banda. Design of nonlinear control laws for high-angle-of-attack flight. *Journal of Guidance, Control, and Dynamics*, 17 (4):737–746, 1994.

ADMIRE ver. 3.4h. *Aerodata Model in Research Environment (ADMIRE), version 3.4h*. Swedish Defence Research Agency (FOI), 2003.

M. T. Alrifai, J. H. Chow, and D. A. Torrey. A backstepping nonlinear control approach to switched reluctance motors. In *Proc. of the 37th IEEE Conference on Decision and Control*, pages 4652–4657, Dec. 1998.

F. Amato, M. Mattei, and S. Scala. Linear quadratic optimal control. In J.-F. Magni, S. Bennani, and J. Terlouw, editors, *Robust Flight Control: A Design Challenge*, chapter 4, pages 33–41. Springer, 1997.

B. D. O. Anderson and J. B. Moore. *Optimal Control: Linear Quadratic Methods*. Prentice-Hall International, Inc., 1989.

D. Anderson and S. Eberhardt. How airplanes fly: A physical description of lift. *Sport Aviation*, Feb. 1999. http://www.allstar.fiu.edu/aero/airflylvL3.htm.

Z. Artstein. Stabilization with relaxed controls. *Nonlinear Analysis, Theory, Methods & Applications*, 7(11):1163–1173, 1983.

D. Axehill and J. Sjöberg. Adaptive cruise control for heavy vehicles – hybrid control and MPC. Master's thesis no. 3416, Linköpings universitet, 2003.

H. Backström. Report on the usage of the Generic Aerodata Model. Technical report, Saab Aircraft AB, May 1997.

R. D. Barnard. Continuous-time implementation of optimal-aim controls. *IEEE Transactions on Automatic Control*, 21:432–434, 1976.

R. E. Beck. *Application of Control Allocation Methods to Linear Systems with Four or More Objectives*. PhD thesis, Virginia Polytechnic Institute and State University, 2002.

S. P. Berge and T. I. Fossen. Robust control allocation of overactuated ships: Experiments with a model ship. In *Proc. of the 4th IFAC Conference on Manoeuvring and Control of Marine Craft*, Brijuni, Croatia, Sept. 1997.

D. P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1995.

Å. Björck. *Numerical Methods for Least Squares Problems*. SIAM, 1996.

M. Bodson. Evaluation of optimization methods for control allocation. *Journal of Guidance, Control, and Dynamics*, 25(4):703–711, July–Aug. 2002.

M. Bodson and W. A. Pohlchuk. Command limiting in reconfigurable flight control. *Journal of Guidance, Control, and Dynamics*, 21(4):639–646, July–Aug. 1998.

J.-L. Boiffier. *The Dynamics of Flight: The Equations*. John Wiley & Sons, 1998.

K. A. Bordignon. *Constrained Control Allocation for Systems with Redundant Control Effectors*. PhD thesis, Virginia Polytechnic Institute and State University, 1996.

K. A. Bordignon and W. C. Durham. Closed-form solutions to the constrained control allocation problem. *Journal of Guidance, Control, and Dynamics*, 18(5): 1000–1007, Sept.–Oct. 1995.

M. M. Bridges, D. M. Dawson, and C. T. Abdallah. Control of rigid-link, flexible-joint robots: A survey of backstepping approaches. *Journal of Robotic Systems*, 12(3):199–216, 1995.

J. Buffington, P. Chandler, and M. Pachter. On-line system identification for aircraft with distributed control effectors. *International Journal of Robust and Nonlinear Control*, 9(14):1033–1049, 1999.

J. M. Buffington. Tailless aircraft control allocation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 737–747, New Orleans, LA, 1997.

J. M. Buffington and D. F. Enns. Lyapunov stability analysis of daisy chain control allocation. *Journal of Guidance, Control, and Dynamics*, 19(6):1226–1230, Nov.–Dec. 1996.

D. J. Bugajski and D. F. Enns. Nonlinear control law with application to high angle-of attack flight. *Journal of Guidance, Control, and Dynamics*, 15(3):761–767, May–June 1992.

J. J. Burken, P. Lu, Z. Wu, and C. Bahm. Two reconfigurable flight-control design methods: Robust servomechanism and control allocation. *Journal of Guidance, Control, and Dynamics*, 24(3):482–493, May–June 2001.

C. I. Byrnes and A. Isidori. Output regulation for nonlinear systems: an overview. *International Journal of Robust and Nonlinear Control*, 10(5):323–337, 2000.

J. J. Carroll, A. J. Geoghan, D. M. Dawson, and P. Vedagarbha. A backstepping based computed torque controller for switched reluctance motors driving inertial loads. In *Proc. of the 4th IEEE Conference on Control Applications*, pages 779–786, Sept. 1995.

J. J. Carroll, M. Schneider, and D. M. Dawson. Integrator backstepping techniques for the tracking control of permanent magnet brush DC motors. In *Conference Record of the 1993 IEEE Industry Applications Society Annual Meeting*, pages 663–671, Oct. 1993.

U. Claréus. Aerodynamic highlights of a fourth generation delta canard fighter aircraft. In *6th Annual International Symposium*, Zhukovsky, Russia, Aug. 2001.

J. Dahlgren. Robust nonlinear control for a missile using backstepping. Master's thesis no. 3300, Linköpings universitet, 2002.

J. B. Davidson, F. J. Lallman, and W. T. Bundick. Integrated reconfigurable control allocation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, Aug. 2001.

D. B. Doman and M. W. Oppenheimer. Improving control allocation accuracy for nonlinear aircraft dynamics. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002.

W. C. Durham. Constrained control allocation. *Journal of Guidance, Control, and Dynamics*, 16(4):717–725, July–Aug. 1993.

W. C. Durham. Attainable moments for the constrained control allocation problem. *Journal of Guidance, Control, and Dynamics*, 17(6):1371–1372, Nov.–Dec. 1994a.

W. C. Durham. Constrained control allocation: Three moment problem. *Journal of Guidance, Control, and Dynamics*, 17(2):330–336, Mar.–Apr. 1994b.

W. C. Durham. Efficient, near-optimal control allocation. *Journal of Guidance, Control, and Dynamics*, 22(2):369–372, Mar.–Apr. 1999.

W. C. Durham. Computationally efficient control allocation. *Journal of Guidance, Control, and Dynamics*, 24(3):519–524, May–June 2001.

W. C. Durham and J. G. Bolling. Minimum drag control allocation. *Journal of Guidance, Control, and Dynamics*, 20(1):190–193, Jan.–Feb. 1997.

W. C. Durham and K. A. Bordignon. Multiple control effector rate limiting. *Journal of Guidance, Control, and Dynamics*, 19(1):30–37, Jan.–Feb. 1996.

W. C. Durham, F. H. Lutze, and W. Mason. Kinematics and aerodynamics of the velocity vector roll. *Journal of Guidance, Control, and Dynamics*, 17(6): 1228–1233, Nov.–Dec. 1994.

R. L. Eberhardt and D. G. Ward. Indirect adaptive flight control of a tailless fighter aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 466–476, Portland, OR, 1999a.

R. L. Eberhardt and D. G. Ward. Indirect adaptive flight control system interactions. *International Journal of Robust and Nonlinear Control*, 9(14):1013–1031, 1999b.

D. Enns. Control allocation approaches. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 98–108, Boston, MA, 1998.

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, Jan. 1994.

B. Etkin and L. D. Reid. *Dynamics of Flight: Stability and Control*. John Wiley & Sons, third edition, 1996.

K. Ezal, Z. Pan, and P. V. Kokotović. Locally optimal and robust backstepping design. *IEEE Transactions on Automatic Control*, 45(2):260–270, Feb. 2000.

D. Fontaine and P. Kokotović. Approaches to global stabilization of a nonlinear system not affine in control. In *Proc. of the American Control Conference*, pages 1645–1647, June 1998.

T. I. Fossen and Å. Grøvlen. Nonlinear output feedback control of dynamically positioned ships using vectorial observer backstepping. *IEEE Transactions on Control Systems Technology*, 6(1):121–128, Jan. 1998.

J. Fredriksson. *Nonlinear Model-based Control of Automotive Powertrains*. PhD thesis no. 427, Control and Automation Laboratory, Department of Signals and Systems, Chalmers University of Technology, 2002.

R. A. Freeman and P. V. Kokotović. *Robust Nonlinear Control Design: State-Space and Lyapunov Techniques*. Birkhäuser, 1996.

R. A. Freeman and J. A. Primbs. Control Lyapunov functions: New ideas from an old source. In *Proc. of the 35th Conference on Decision and Control*, pages 3926–3931, Dec. 1996.

L. Gentili and L. Marconi. Robust nonlinear regulation for a magnetic levitation system. In *Proc. of the 5th IFAC Symposium on Nonlinear Control Systems*, St. Petersburg, Russia, July 2001.

S. T. Glad. Robustness of nonlinear state feedback – a survey. *Automatica*, 23 (4):425–435, 1987.

S. T. Glad and O. Härkegård. Backstepping control of a rigid body. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 3944–3945, Las Vegas, NV, Dec. 2002.

T. Glad and L. Ljung. *Control Theory: Multivariable and Nonlinear Methods*. Taylor & Francis, 2000.

G. H. Golub and C. F. Van Loan. *Matrix Computations*. John Hopkins University Press, third edition, 1996.

Å. Grøvlen and T. I. Fossen. Nonlinear control of dynamic positioned ships using only position feedback: An observer backstepping approach. In *Proc. of the 35th Conference on Decision and Control*, pages 3388–3393, Dec. 1996.

J. K. Hall and M. Pachter. Formation maneuvers in three dimensions. In *Proc. of the 39th Conference on Decision and Control*, Dec. 2000.

O. Härkegård. Dynamic control allocation using constrained quadratic programming. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002a.

O. Härkegård. Efficient active set algorithms for solving constrained least squares problems in aircraft control allocation. In *Proc. of the 41st IEEE Conference on Decision and Control*, pages 1295–1300, Las Vegas, NV, Dec. 2002b.

O. Härkegård. Resolving actuator redundancy—control allocation vs linear quadratic control. In *Proc. of the European Control Conference*, Cambridge, UK, Sept. 2003. Accepted for presentation.

O. Härkegård and S. T. Glad. A backstepping design for flight path angle control. In *Proc. of the 39th Conference on Decision and Control*, pages 3570–3575, Sydney, Australia, Dec. 2000.

O. Härkegård and S. T. Glad. Control of systems with input nonlinearities and uncertainties: An adaptive approach. In *Proc. of the European Control Conference*, pages 1912–1917, Porto, Portugal, Sept. 2001a.

O. Härkegård and S. T. Glad. Flight control design using backstepping. In *Proc. of the 5th IFAC Symposium on Nonlinear Control Systems*, pages 259–264, St. Petersburg, Russia, July 2001b.

W. B. Herbst. Future flight technologies. *Journal of Aircraft*, 17(8):561–566, Aug. 1980.

A. S. Hodel and R. Callahan. Autonomous reconfigurable control allocation (ARCA) for reusable launch vehicles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002.

J. Hu, D. M. Dawson, and K. Anderson. Position control of a brushless DC motor without velocity measurements. *IEE Proceedings of Electric Power Applications*, 142(2):113–122, June 1995.

J. Hu, D. M. Dawson, and Y. Qian. Position tracking control for robot manipulators driven by induction motors without flux measurements. *IEEE Transactions on Robotics and Automation*, 12(3):419–438, June 1996.

Y. Ikeda and M. Hood. An application of L1 optimization to control allocation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, Aug. 2000.

A. Isidori. *Nonlinear Control Systems*. Springer, third edition, 1995.

M. Jankovic, M. Jankovic, and I. Kolmanovsky. Constructive Lyapunov control design for turbocharged diesel engines. *IEEE Transactions on Control Systems Technology*, 8(2):288–299, Mar. 2000.

Z.-P. Jiang and H. Nijmeijer. Tracking control of mobile robots: A case study in backstepping. *Automatica*, 33(7):1393–1399, 1997.

I. Kanellakopoulos, P. V. Kokotovic, and A. S. Morse. A toolkit for nonlinear feedback design. *Systems & Control Letters*, 18(2):83–92, Feb. 1992.

H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, third edition, 2002.

U. Kiencke and L. Nielsen. *Automotive Control Systems*. Springer, 2000.

P. Kokotović. Constructive nonlinear control: Progress in the 90's. In *IFAC 1999 Proceedings*, pages 49–77, 1999.

P. Kokotović, H. K. Khalil, and J. O'Reilly. *Singular Perturbation Methods in Control: Analysis and Design*. Academic Press, 1986.

P. V. Kokotović. The joy of feedback: Nonlinear and adaptive. *IEEE Control Systems Magazine*, 12(3):7–17, June 1992.

A. J. Krener and A. Isidori. Linearization by output injection and nonlinear observers. *Systems & Control Letters*, 3:47–52, June 1983.

M. Krstić, D. Fontaine, P. V. Kokotović, and J. D. Paduano. Useful nonlinearities and global stabilization of bifurcations in a model of jet engine surge and stall. *IEEE Transactions on Automatic Control*, 43(12):1739–1745, Dec. 1998.

M. Krstić, I. Kanellakopoulos, and P. Kokotović. *Nonlinear and Adaptive Control Design*. John Wiley & Sons, 1995.

S. H. Lane and R. F. Stengel. Flight control design using non-linear inverse dynamics. *Automatica*, 24(4):471–483, 1988.

I. Lindfors. Thrust allocation method for the dynamic positioning system. In *10th International Ship Control Systems Symposium (SCSS'93)*, pages 3.93–3.106, Ottawa, Canada, 1993.

J. Löfberg. Backstepping with local LQ performance and global approximation of quadratic performance. In *Proc. of the American Control Conference*, pages 3898–3902, June 2000.

P. Lötstedt. Solving the minimum least squares problem subject to bounds on the variables. *BIT*, 24:206–224, 1984.

P. Lu. Constrained tracking control of nonlinear systems. *Systems & Control Letters*, 27(5):305–314, 1996.

P. Lu and Z. Shen. Unifying treatment to control of nonlinear systems with two timescales. *Journal of Guidance, Control, and Dynamics*, 25(5):975–979, Sept.–Oct. 2002.

D. G. Luenberger. *Linear and Nonlinear Programming*. Addison-Wesley, second edition, 1984.

A. Lundán and O. Mattila. A system for the control of the homogenisation of the cement raw meal. In *4th IFAC Symposium on Digital Computer Applications to Process Control*, Zürich, Switzerland, 1974.

A. M. Lyapunov. *The General Problem of the Stability of Motion*. Taylor & Francis, 1992. English translation of the original publication in Russian from 1892.

J.-F. Magni, S. Bennani, and J. Terlouw, editors. *Robust Flight Control: A Design Challenge*. Springer, 1997.

MATLAB Optimization Toolbox. *Optimization Toolbox User's Guide. Version 2.2*. The MathWorks, Inc., www.mathworks.com, 2002.

G. Meyer, R. Su, and L. R. Hunt. Application of nonlinear transformations to automatic flight control. *Automatica*, 20(1):103–107, 1984.

I. Moir and A. Seabridge. *Aircraft Systems*. Longman Scientific & Technical, 1992.

R. C. Nelson. *Flight Stability and Automatic Control*. McGraw-Hill, second edition, 1998.

J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer, 1999.

A. B. Page and M. L. Steinberg. High-fidelity simulation testing of control allocation methods. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002.

G. Papageorgiou, K. Glover, and R. A. Hyde. The $\mathcal{H}_\infty$ loop-shaping approach. In J.-F. Magni, S. Bennani, and J. Terlouw, editors, *Robust Flight Control: A Design Challenge*, chapter 29, pages 464–483. Springer, 1997.

J. A. M. Petersen and M. Bodson. Control allocation for systems with coplanar controls. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, pages 1776–1786, Denver, CO, Aug. 2000.

J. Reiner, G. J. Balas, and W. L. Garrard. Robust dynamic inversion for control of highly maneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 18 (1):18–24, Jan.–Feb. 1995.

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.

J. Roskam. Things the pros should know about high angle of attack. *Professional Pilot*, 23(8):72–76, Aug. 1989.

W. J. Rugh. *Linear System Theory*. Prentice Hall, second edition, 1996.

W. J. Rugh and J. S. Shamma. Research on gain scheduling. *Automatica*, 36(10): 1401–1425, Oct. 2000.

R. Sepulchre, M. Janković, and P. V. Kokotović. *Constructive Nonlinear Control*. Springer, 1997a.

R. Sepulchre, M. Jankovic, and P. V. Kokotovic. Integrator forwarding: A new recursive nonlinear robust design. *Automatica*, 33(5):979–984, May 1997b.

R. Sepulchre, M. Janković, and P. V. Kokotović. Interlaced systems and recursive designs for global stabilization. In *Proc. of the 1997 European Control Conference*, 1997c.

M. Sharma and D. G. Ward. Flight-path angle control via neuro-adaptive backstepping. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002.

R. H. Shertzer, D. J. Zimpfer, and P. D. Brown. Control allocation for the next generation of entry vehicles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Monterey, CA, Aug. 2002.

S. N. Singh, P. Chandler, C. Schumacher, S. Banda, and M. Pachter. Nonlinear adaptive close formation control of unmanned aerial vehicles. *Dynamics and Control*, 10(2):179–194, 2000.

S. N. Singh and M. Steinberg. Adaptive control of feedback linearizable nonlinear systems with application to flight control. *Journal of Guidance, Control, and Dynamics*, 19(4):871–877, July–Aug. 1996.

J.-J. E. Slotine and W. Li. *Applied Nonlinear Contol*. Prentice Hall, 1991.

S. A. Snell, D. F. Enns, and W. L. Garrard Jr. Nonlinear inversion flight control for a supermaneuverable aircraft. *Journal of Guidance, Control, and Dynamics*, 15(4):976–984, July–Aug. 1992.

E. D. Sontag. A 'universal' construction of Artstein's theorem on nonlinear stabilization. *Systems & Control Letters*, 13:117–123, 1989.

O. J. Sørdalen. Optimal thrust allocation for marine vessels. *Control Engineering Practice*, 5(9):1223–1231, 1997.

K. Ståhl Gunnarsson. Kravdokument för styrlagsdesign. Technical Report FAM-99.030, Saab AB, Aug. 1999. In Swedish.

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

B. L. Stevens and F. L. Lewis. *Aircraft Control and Simulation*. John Wiley & Sons, 1992.

J. P. Strand, K. Ezal, T. I. Fossen, and P. V. Kokotović. Nonlinear control of ships: A locally optimal design. In *Proc. of the 4th IFAC Symposium on Nonlinear Control Systems*, pages 732–737, July 1998.

G. Strang. *Linear Algebra and Its Applications*. Academic Press, second edition, 1980.

S. W. Su, B. D. O. Anderson, and T. S. Brinsmead. Constant disturbance suppression for nonlinear systems design using singular perturbation theory. In *Proc. of the 40th Conference on Decision and Control*, pages 3272–3277, Orlando, FL, Dec. 2001.

R. Venkataraman and D. B. Doman. Control allocation and compensation for over-actuated systems with non-linear effectors. In *Proc. of the American Control Conference*, pages 1812–1814, Arlington, VA, June 2001.

J. C. Virnig and D. S. Bodden. Multivariable control allocation and control law conditioning when control effectors limit. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Scottsdale, AZ, Aug. 1994.

E. Wänström. Control and analysis of a nonlinear missile using backstepping. Master's thesis no. 3166, Linköpings universitet, 2001.

K. H. Well, B. Faber, and E. Berger. Optimization of tactical aircraft maneuvers utilizing high angles of attack. *Journal of Guidance, Control, and Dynamics*, 5 (2):131–137, Mar.–Apr. 1982.

T. Westerlund, H. Toivonen, and K.-E. Nyman. Stochastic modelling and self-tuning control of a continuous cement raw material mixing system. *Modeling, Identification and Control*, 1(1):17–37, 1980.

K. A. Wise, J. S. Brinker, A. J. Calise, D. F. Enns, M. R. Elgersma, and P. Voulgaris. Direct adaptive reconfigurable flight control for a tailless advanced fighter aircraft. *International Journal of Robust and Nonlinear Control*, 9(14):999–1012, 1999.

C. A. Woolsey, A. M. Bloch, N. E. Leonard, and J. E. Marsden. Dissipation and controlled euler-poincare systems. In *Proc. of the 40th IEEE Conference on Decision and Control*, pages 3378–3383, Orlando, FL, Dec. 2001.

C. A. Woolsey and N. E. Leonard. Global asymptotic stabilization of an underwater vehicle using internal rotors. In *Proc. of the 38th IEEE Conference on Decision and Control*, pages 2527–2532, Phoenix, AZ, Dec. 1999.

Z.-J. Yang and D. Miyazaki. Adaptive robust nonlinear control of a voltage-controlled magnetic levitation system. In *Proc. of the 5th IFAC Symposium on Nonlinear Control Systems*, St. Petersburg, Russia, July 2001.

F. Zhang. *Matrix Theory: Basic results and techniques*. Springer, 1999.

K. Zhou, J. C. Doyle, and K. Glover. *Robust and Optimal Control*. Prentice-Hall, 1996.

Index

PhD Dissertations, Division of Automatic Control, Linköping University

M. Millnert: Identification and control of systems subject to abrupt changes. Thesis no. 82, 1982. ISBN 91-7372-542-0.

A.J.M. van Overbeek: On-line structure selection for the identification of multivariable systems. Thesis no. 86, 1982. ISBN 91-7372-586-2.

B. Bengtsson: On some control problems for queues. Thesis no. 87, 1982. ISBN 91-7372-593-5.

S. Ljung: Fast algorithms for integral equations and least squares identification problems. Thesis no. 93, 1983. ISBN 91-7372-641-9.

H. Jonson: A Newton method for solving non-linear optimal control problems with general constraints. Thesis no. 104, 1983. ISBN 91-7372-718-0.

E. Trulsson: Adaptive control based on explicit criterion minimization. Thesis no. 106, 1983. ISBN 91-7372-728-8.

K. Nordström: Uncertainty, robustness and sensitivity reduction in the design of single input control systems. Thesis no. 162, 1987. ISBN 91-7870-170-8.

B. Wahlberg: On the identification and approximation of linear systems. Thesis no. 163, 1987. ISBN 91-7870-175-9.

S. Gunnarsson: Frequency domain aspects of modeling and control in adaptive systems. Thesis no. 194, 1988. ISBN 91-7870-380-8.

A. Isaksson: On system identification in one and two dimensions with signal processing applications. Thesis no. 196, 1988. ISBN 91-7870-383-2.

M. Viberg: Subspace fitting concepts in sensor array processing. Thesis no. 217, 1989. ISBN 91-7870-529-0.

K. Forsman: Constructive commutative algebra in nonlinear control theory. Thesis no. 261, 1991. ISBN 91-7870-827-3.

F. Gustafsson: Estimation of discrete parameters in linear systems. Thesis no. 271, 1992. ISBN 91-7870-876-1.

P. Nagy: Tools for knowledge-based signal processing with applications to system identification. Thesis no. 280, 1992. ISBN 91-7870-962-8.

T. Svensson: Mathematical tools and software for analysis and design of nonlinear control systems. Thesis no. 285, 1992. ISBN 91-7870-989-X.

S. Andersson: On dimension reduction in sensor array signal processing. Thesis no. 290, 1992. ISBN 91-7871-015-4.

H. Hjalmarsson: Aspects on incomplete modeling in system identification. Thesis no. 298, 1993. ISBN 91-7871-070-7.

I. Klein: Automatic synthesis of sequential control schemes. Thesis no. 305, 1993. ISBN 91-7871-090-1.

J.-E. Strömberg: A mode switching modelling philosophy. Thesis no. 353, 1994. ISBN 91-7871-430-3.

K. Wang Chen: Transformation and symbolic calculations in filtering and control. Thesis no. 361, 1994. ISBN 91-7871-467-2.

T. McKelvey: Identification of state-space models from time and frequency data. Thesis no. 380, 1995. ISBN 91-7871-531-8.

J. Sjöberg: Non-linear system identification with neural networks. Thesis no. 381, 1995. ISBN 91-7871-534-2.

R. Germundsson: Symbolic systems – theory, computation and applications. Thesis no. 389, 1995. ISBN 91-7871-578-4.

P. Pucar: Modeling and segmentation using multiple models. Thesis no. 405, 1995. ISBN 91-7871-627-6.

H. Fortell: Algebraic approaches to normal forms and zero dynamics. Thesis no. 407, 1995. ISBN 91-7871-629-2.

A. Helmersson: Methods for robust gain scheduling. Thesis no. 406, 1995. ISBN 91-7871-628-4.

P. Lindskog: Methods, algorithms and tools for system identification based on prior knowledge. Thesis no. 436, 1996. ISBN 91-7871-424-8.

J. Gunnarsson: Symbolic methods and tools for discrete event dynamic systems. Thesis no. 477, 1997. ISBN 91-7871-917-8.

M. Jirstrand: Constructive methods for inequality constraints in control. Thesis no. 527, 1998. ISBN 91-7219-187-2.

U. Forssell: Closed-loop identification: Methods, theory, and applications. Thesis no. 566, 1999. ISBN 91-7219-432-4.

A. Stenman: Model on demand: Algorithms, analysis and applications. Thesis no. 571, 1999. ISBN 91-7219-450-2.

N. Bergman: Recursive Bayesian estimation: Navigation and tracking applications. Thesis no. 579, 1999. ISBN 91-7219-473-1.

K. Edström: Switched bond graphs: Simulation and analysis. Thesis no. 586, 1999. ISBN 91-7219-493-6.

M. Larsson: Behavioral and structural model based approaches to discrete diagnosis. Thesis no. 608, 1999. ISBN 91-7219-615-5.

F. Gunnarsson: Power control in cellular radio systems: Analysis, design and estimation. Thesis no. 623, 2000. ISBN 91-7219-689-0.

V. Einarsson: Model checking methods for mode switching systems. Thesis no. 652, 2000. ISBN 91-7219-836-2.

M. Norrlöf: Iterative learning control: Analysis, design, and experiments. Thesis no. 653, 2000. ISBN 91-7219-837-0.

F. Tjärnström: Variance expressions and model reduction in system identification. Thesis no. 730, 2002. ISBN 91-7373-253-2.

J. Löfberg: Minimax approaches to robust model predictive control. Thesis no. 812, 2003. ISBN 91-7373-622-8.

J. Roll: Local and piecewise affine approaches to system identification. Thesis no. 802, 2003. ISBN 91-7373-608-2.

J. Elbornsson: Analysis, estimation and compensation of mismatch effects in A/D converters. Thesis no. 811, 2003. ISBN 91-7373-621-X.