Guillaume J.J. Ducard

Advances in Industrial Control

Fault-tolerant Flight Control and Guidance Systems

Practical Methods for Small Unmanned Aerial Vehicles

Springer

Advances in Industrial Control

Other titles published in this series:

Digital Controller Implementation and Fragility
Robert S.H. Istepanian and James F. Whidborne (Eds.)

Optimisation of Industrial Processes at Supervisory Level
Doris Sáez, Aldo Cipriano and Andrzej W. Ordys

Robust Control of Diesel Ship Propulsion
Nikolaos Xiros

Hydraulic Servo-systems
Mohieddine Jelali and Andreas Kroll

Model-based Fault Diagnosis in Dynamic Systems Using Identification Techniques
Silvio Simani, Cesare Fantuzzi and Ron J. Patton

Strategies for Feedback Linearisation
Freddy Garces, Victor M. Becerra, Chandrasekhar Kambhampati and Kevin Warwick

Robust Autonomous Guidance
Alberto Isidori, Lorenzo Marconi and Andrea Serrani

Dynamic Modelling of Gas Turbines
Gennady G. Kulikov and Haydn A. Thompson (Eds.)

Control of Fuel Cell Power Systems
Jay T. Pukrushpan, Anna G. Stefanopoulou and Huei Peng

Fuzzy Logic, Identification and Predictive Control
Jairo Espinosa, Joos Vandewalle and Vincent Wertz

Optimal Real-time Control of Sewer Networks
Magdalene Marinaki and Markos Papageorgiou

Process Modelling for Control
Benoît Codrons

Computational Intelligence in Time Series Forecasting
Ajoy K. Palit and Dobrivoje Popovic

Modelling and Control of Mini-Flying Machines
Pedro Castillo, Rogelio Lozano and Alejandro Dzul

Ship Motion Control
Tristan Perez

Hard Disk Drive Servo Systems (2nd Ed.)
Ben M. Chen, Tong H. Lee, Kemao Peng and Venkatakrishnan Venkataramanan

Measurement, Control, and Communication Using IEEE 1588
John C. Eidson

Piezoelectric Transducers for Vibration Control and Damping
S.O. Reza Moheimani and Andrew J. Fleming

Manufacturing Systems Control Design
Stjepan Bogdan, Frank L. Lewis, Zdenko Kovačić and José Mireles Jr.

Windup in Control
Peter Hippe

Nonlinear H_2/H_∞ Constrained Feedback Control
Murad Abu-Khalaf, Jie Huang and Frank L. Lewis

Practical Grey-box Process Identification
Torsten Bohlin

Control of Traffic Systems in Buildings
Sandor Markon, Hajime Kita, Hiroshi Kise and Thomas Bartz-Beielstein

Wind Turbine Control Systems
Fernando D. Bianchi, Hernán De Battista and Ricardo J. Mantz

Advanced Fuzzy Logic Technologies in Industrial Applications
Ying Bai, Hanqi Zhuang and Dali Wang (Eds.)

Practical PID Control
Antonio Visioli

(continued after Index)

Guillaume J.J. Ducard

Fault-tolerant Flight Control and Guidance Systems

Practical Methods for Small Unmanned Aerial Vehicles

Springer

Guillaume J.J. Ducard, PhD
Measurement and Control Laboratory
Department of Mechanical and Process Engineering
ETH Zurich
Sonneggstrasse 3
8092 Zurich
Switzerland
guillaume.ducard@imrt.mavt.ethz.ch
guillaumeducard@ieee.org

ISSN 1430-9491
ISBN 978-1-84882-560-4 e-ISBN 978-1-84882-561-1
DOI 10.1007/978-1-84882-561-1
Springer Dordrecht Heidelberg London New York

British Library Cataloguing in Publication Data
A catalogue record for this book is available from the British Library

Library of Congress Control Number: 2009926142

Cover design: eStudioCalamar, Figueres/Berlin

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Advances in Industrial Control

Professor (Emeritus) O.P. Malik
Department of Electrical and Computer Engineering
University of Calgary
2500, University Drive, NW
Calgary, Alberta
T2N 1N4
Canada

Professor K.-F. Man
Electronic Engineering Department
City University of Hong Kong
Tat Chee Avenue
Kowloon
Hong Kong

Professor G. Olsson
Department of Industrial Electrical Engineering and Automation
Lund Institute of Technology
Box 118
S-221 00 Lund
Sweden

Professor A. Ray
Department of Mechanical Engineering
Pennsylvania State University
0329 Reber Building
University Park
PA 16802
USA

Professor D.E. Seborg
Chemical Engineering
3335 Engineering II
University of California Santa Barbara
Santa Barbara
CA 93106
USA

Doctor K.K. Tan
Department of Electrical and Computer Engineering
National University of Singapore
4 Engineering Drive 3
Singapore 117576

Professor I. Yamamoto
Department of Mechanical Systems and Environmental Engineering
The University of Kitakyushu
Faculty of Environmental Engineering
1-1, Hibikino,Wakamatsu-ku, Kitakyushu, Fukuoka, 808-0135
Japan

To my parents

Series Editors' Foreword

The series *Advances in Industrial Control* aims to report and encourage technology transfer in control engineering. The rapid development of control technology has an impact on all areas of the control discipline. New theory, new controllers, actuators, sensors, new industrial processes, computer methods, new applications, new philosophies , new challenges. Much of this development work resides in industrial reports, feasibility study papers and the reports of advanced collaborative projects. The series offers an opportunity for researchers to present an extended exposition of such new work in all aspects of industrial control for wider and rapid dissemination.

Autonomy for aerial, land, and marine (surface and underwater) vehicles is an ever-expanding field of industrial control engineering in which there is significant international interest. Currently, there are many prototypes and working autonomous vehicles in all the fields of application; however, some areas are better developed than others. Meanwhile in the control conference literature it is possible to see that frontier research has reached the problems of working with groups, convoys or swarms of cooperating autonomous vehicles.

The tasks that autonomous mobiles can tackle are very often either hazardous, or, conversely, routine, where the use of an *insitu* human operator is to be avoided, or simply technically (and economically) unnecessary. Typically, such tasks involve inspection, monitoring, and detection. For example, unmanned aerial vehicles (UAVs) can be used to perform airborne sea searches, inspect long-distance power lines or oil and gas pipelines (particularly those traversing hostile, or hazardous terrain), monitor environmental or meteorological variables and survey crop production and forestry resources. This list is by no means exhaustive and UAVs can perform many other valuable tasks.

In the technological field of UAVs, designing, and testing, a complete fault-tolerant control and guidance system is a demanding and challenging task. This is the objective of the research reported by Dr. Guillaume Ducard in this *Advances in Industrial Control* monograph. The practical problem faced by the control engineer is that once aircraft actuators fail, or the aircraft

suffers some structural airframe damage then the aircraft flying characteristics change, the flying performance deteriorates and immediate action for airborne survival is required. Identifying the flying status of the aircraft and reconfiguring the control system to maintain airborne performance are just two of the actions that may be needed. In Dr Ducard's book, there is a careful and thorough development of a system and its component modules to deal with fault situations like these. The reader will find the analysis, the development, the testing, and the presentation to be comprehensive. As the monograph subtitle, *Practical Methods for Small Unmanned Aerial Vehicles*, suggests, careful consideration is given to the development of designs and algorithms that satisfy the tight constraints of real-time capability, modularity, and limited computing power. Simulations are presented with realistic and targeted scenarios to test the separate components and then finally the complete system.

Fault-tolerant Control and Guidance Systems will be of interest to a wide range of readers from the industrial and the control communities. Academic researchers and graduate students working in the mobile autonomous vehicle field may find transferable knowledge in this well-structured monograph and it will obviously appeal to aerospace researchers and engineers. It is therefore a very welcome addition to the *Advances in Industrial Control* series.

Industrial Control Centre
Glasgow
Scotland, UK
2009

M.J. Grimble
M.A. Johnson

Preface

This book is based on part of the research work I carried out at the Measurement and Control Laboratory of the Swiss Federal Institute of Technology (ETH Zurich) over the past few years in the unmanned aerial vehicle (UAV) group founded by Professor H. P. Geering in 1986. At that time, research focused on developing navigation algorithms and robust control techniques applied to unmanned helicopters. In 2006, our colleague Markus Möckli successfully completed flight tests of an aircraft autonomously flying aerobatic maneuvers. In 2007, Marco Gerig flight tested guidance and control algorithms for aerobatic maneuvers with a small autonomous helicopter.

Concurrently, in the context of my PhD work completed in 2007, a new research activity was initiated dealing with the design of fault-tolerant flight control and guidance systems for a small unmanned aircraft. This work focused on designing techniques to detect and isolate faults among sensors and actuators and on developing methods to adaptively reconfigure the flight control laws and the vehicle trajectory. Reasonable complexity, real-time capability and modularity were the main requirements for the algorithms designed.

My first thanks go to Professor Hans Peter Geering for offering me exceptional working conditions over the few years under his supervision and for opportunities to do research on the fascinating topics covered in this book. I gratefully acknowledge his constant support in many different aspects.

I owe special thanks to Brigitte Rohrbach, our former secretary, for her support in copy-editing part of this manuscript, for her invaluable help in many aspects, and for her kindness in general.

I am also very grateful for the precious assistance I received from Oliver Jackson, Sorina Moosdorf, Aislinn Bunning, Professor Michael Johnson and Professor Mike Grimble.

Last but not least, I thank my parents and my brother for their continuous and loving support in every possible way in life in general.

ETH Zurich
Spring 2009

Guillaume Ducard

Contents

Abbreviations

List of abbreviations that are used in the book.

EKF	extended Kalman filter
EMMAE	extended multiple model adaptive estimation
FDI	fault detection and isolation
KF	Kalman filter
LPV	linear parameter varying
MMAE	multiple model adaptive estimation
MMST	multiple model switching and tuning
MPC	model predictive control
MRAC	model reference adaptive control
MSLS	modified sequential least squares
NASA	National Aeronautics and Space Administration
NDI	nonlinear dynamic inversion
NFZ	no-fly zone
PD	proportional derivative
PI	proportional integral
PID	proportional, integral and derivative
PRM	probabilistic road map
RLS	recursive least squares
SLS	sequential least squares
SPF	sigma point filter
SVs	singular values
UAV	unmanned aerial vehicle
UKF	unscented Kalman filter

Chapter 1
Introduction

This book deals with the design of fault-tolerant control and guidance systems for a small unmanned aircraft. This book focuses on designing techniques to detect and isolate faults among sensors and actuators and on developing methods to appropriately reconfigure the flight control laws and the vehicle trajectory. Real-time capability and modularity are two main requirements for the algorithms designed.

1.1 Motivations for Fault-tolerant Control Systems for Unmanned Aerial Vehicles

Unmanned aerial vehicles have been around and in service since the 1990s and are going to be routinely used for a wide range of tasks such as:

- Sea rescue searches from the air
- Border patrols, homeland security, law enforcement, monitoring of drug trafficking
- Monitoring and control of road traffic and transportation
- Crop yield prediction, drought monitoring, spraying of pesticides
- Inspections of power lines, bridges and barrages
- Observation of oil and gas pipelines
- Forest monitoring, fire detection, firefighting: operation and management
- Relaying and broadcasting of mobile telecommunication, TV or radio programs, Internet connections
- Tactical reconnaissance and operational support
- Landmine detection, operation in hazard or disaster zones
- Digital charting and mapping, mineral and archaeological prospecting
- Environmental and climate research: monitoring of air quality, meteorological studies and predictions

- Automated unmanned airshipping, postal delivery
- Filming and movie applications

New generations of UAVs will be designed to achieve their mission not only with increased efficiency, but also with more safety and security. Future UAVs will be operated with algorithms capable of monitoring the aircraft's health and of taking action if needed. Fault-tolerant control systems for small and low-cost UAVs should not increase significantly the number of actuators or sensors needed to achieve the safer operation. Safe and reliable operation of the UAV relies on the following key points:

- The flight control system must be robust against the aircraft model's uncertainties and external disturbances.
- An efficient fault detection and isolation (FDI) system should be capable of monitoring the health status of the aircraft.
- The guidance system should be reconfigurable depending on actuator fault occurrence or aircraft damage. It should generate an appropriate flight trajectory that avoids obstacles despite flight performance degradation.

1.2 Book Outline

The book starts with a literature review about the fault-tolerant flight control systems already developed for aircraft. In Chap. 3, the nonlinear model used for simulation and control design is presented.

Chapter 4 presents a new system for the detection and isolation of actuator faults in an aircraft. Chapter 5 details the design and the implementation of a reconfigurable control allocation module.

Chapter 6 describes the technique known as *nonlinear dynamic inversion* (NDI) and presents the architecture and the design procedure of the controllers used in the aircraft autopilot. Chapter 7 is dedicated to the analysis and control of the longitudinal axis of the aircraft and presents an architecture for the altitude controller that uses robust NDI in all the control loops. Chapter 8 is dedicated to the analysis and control of the lateral-directional motion of the aircraft. It describes the controllers for the roll and yaw rates, for the roll angle, and for the sideslip angle.

Chapter 9 presents a new adaptive path-planning algorithm, which reconfigures the aircraft's trajectory based on its flying performance after the occurrence of an actuator failure. Chapter 10 focuses on an aileron failure and shows how the degraded flying performance can be evaluated and used to reconfigure the guidance system.

Finally, the book concludes with an outlook and summarizes the methods presented.

Chapter 2
Review

This chapter reviews some of the most relevant fault-tolerant flight control systems that can be found in the literature. Since the terminology used in this field is not unique and differs among authors, the chapter starts with a brief definition of some terms and expressions frequently used throughout this book.

2.1 Definition of Fault-tolerant Systems

Since the systems of interest are said to be fault-tolerant, let us first clarify the terminological distinction between a *fault* and a *failure* [1].

2.1.1 Fault

"A fault is an unpermitted deviation of at least one characteristic property (feature) of the system from the acceptable, usual, standard condition." [1]

Based on this definition, a fault corresponds to an abnormal behavior of the system, which may not affect the overall functioning of the system but may eventually lead to a failure (defined below). Finally, a fault may be small or hidden, and therefore difficult to detect and estimate.

For example, consider the temperature of an engine. If this temperature exceeds a certain accepted limit, say 100°C, there is a fault in the system. Although this excessive temperature does not prevent the engine from working properly for a while, it may eventually damage components of the engine and possibly lead to its breaking down.

In this book, an actuator fault corresponds to any abnormal behavior. This includes bias or loss of effectiveness as shown in Fig. 2.1d.

A sensor fault occurs as soon as the measurement data deviate from the real physical measured process by more than the noise uncertainty. Bias, excessive noise, or wrong scaling factors are also classified as sensor faults, as shown in Fig. 2.2.

2.1.2 Failure

"A failure is a permanent interruption of a system's ability to perform a required function under specified operating conditions." [1]

Resulting from one or more faults, a failure is therefore an event that terminates the functioning of a unit in the system. On an aircraft, actuators are used to deflect control surfaces such as ailerons, elevators, and rudders, and also to actuate the engine throttle or the landing-gear mechanism. An actuator is declared failed when it can no longer be used in a controlled manner.

For a control surface, there are two major types of failures [2]. As shown in Fig. 2.1a, the control surface may become ineffective and float at the zero-moment position. The control surface can also be locked at any arbitrary intermediate position (Fig. 2.1b) or reach and stay at the saturation position as shown in Fig. 2.1c.

Mechanical failures may also happen. This is the case when the mechanical link between the control surface and its corresponding actuator or servo breaks. The engine may also fail.

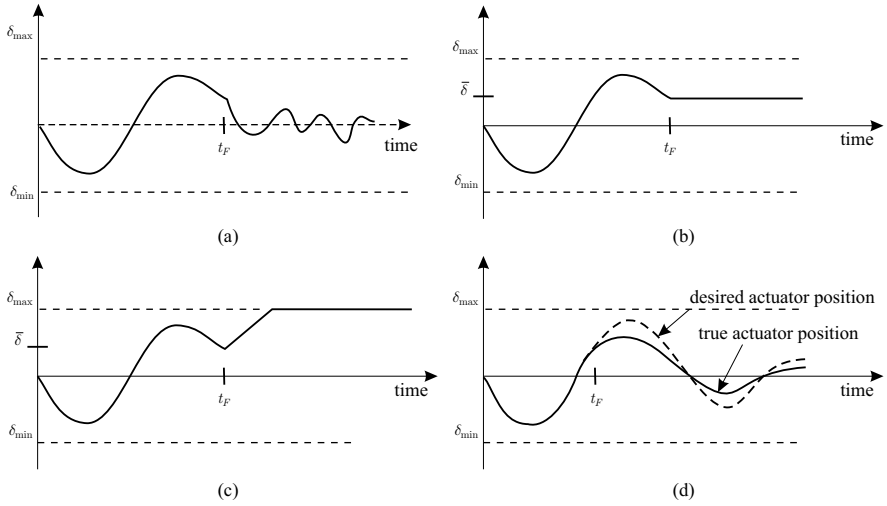


Fig. 2.1 Several types of actuator failures: (a) floating around trim; (b) locked-in-place; (c) hard-over; and (d) loss of effectiveness (actuator fault occurring after t_F)

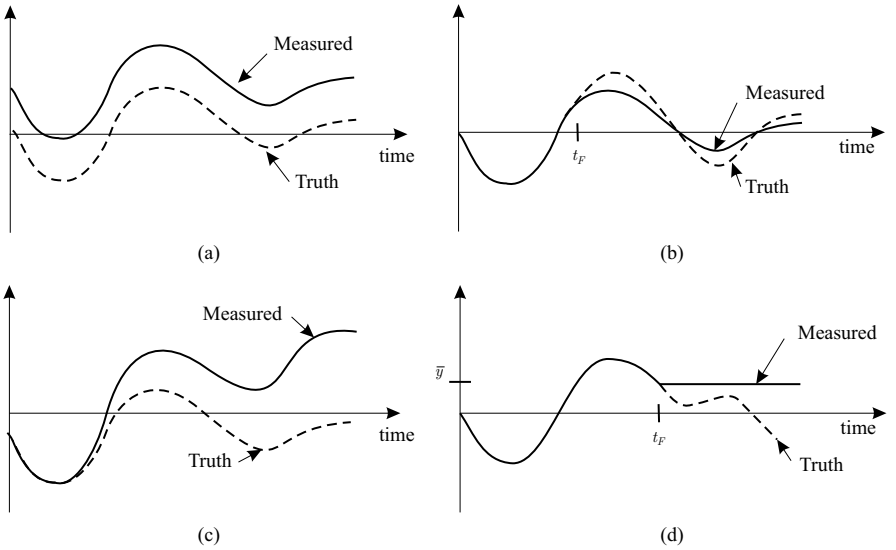


Fig. 2.2 Several types of sensor faults: (a) sensor bias; (b) loss of accuracy or calibration error; (c) sensor drift; and (d) frozen sensor (occurring after t_F)

Finally, there are many sources of possible irreversible damage to the aircraft that may be classified as structural failures. They correspond to the scenarios where a piece of the aircraft is missing, such as an aileron, a tail rudder, an elevator, or part of a wing.

The reconfigurable flight control system of this book is capable of detecting faults in the system (which are more difficult to detect than failures) and is able to adequately compensate for failures (which is more difficult than to only accommodate faults).

2.1.3 Fault-tolerant Control System

A fault-tolerant control system is capable of controlling the system with satisfactory performance even if one or several faults, or more critically, one or several failures occur in this system. Fault-tolerant control systems may be regrouped into two main families: passive fault-tolerant controllers and active fault-tolerant controllers.

2.1.3.1 Passive Fault-tolerant Controllers

In a passive fault-tolerant controller, deviations of the plant parameters from their true values or deviations of the actuators from their expected position

may be efficiently compensated by a fixed robust feedback controller [3–5]. However, if these deviations become excessively large and exceed the robustness properties, some actions need to be taken. Also, if deviations occur at the sensor side, inevitable deviations from the reference command signals will happen. Therefore, an active fault-tolerant control architecture is needed in order to achieve extended fault-tolerance capability.

2.1.3.2 Active Fault-tolerant Controllers

An active fault-tolerant controller usually contains a separate module: an FDI system that monitors the health of the aircraft. The FDI system informs a supervision module of the seriousness of the fault/failure or damage. Based thereon, the supervision module may decide to reconfigure the flight controllers, the guidance system, and the navigation system.

There are also two families of FDI systems, namely passive FDI and active FDI systems. Passive FDI systems "wait" until a fault or failure occurs [6], whereas active FDI systems will artificially excite the aircraft, either by flying health-check maneuvers [7, 8] or by injecting test signals in the actuator commands and then assessing the individual health status of actuators and sensors [8–15].

In this book, an active fault-tolerant control system has been developed, which contains an active nonlinear FDI system and robust nonlinear controllers in all of the control loops of the autopilot. Furthermore, a supervision module has been designed that is capable of reconfiguring the control allocation process described in Chap. 5, the controllers presented in Chaps. 7 and 8, and the guidance system explained in Chaps. 9 and 10.

2.1.4 Dealing with Faults and Failures in Practice

In this book, there will be an emphasis on actuator faults and failures. Indeed, a sensor failure does not modify the flying performance of the aircraft. The sensor failure can be handled either by using a redundant sensor if available, or by reconstructing the missing measurement data with the knowledge of the plant and the measurement data furnished by the remaining sensors [16]. The FDI method presented in this book is also capable of reconstructing the data of a failed attitude sensor [17]. However, as soon as there is an actuator failure or any damage to the airframe, the flying qualities of the aircraft inevitably degrade, and immediate action must be taken to preserve the aircraft's integrity. This is the focus of this book.

2.2 Challenges of Designing Reconfigurable Control Systems

There are many challenges when designing a reconfigurable flight control system and the difficulties may be categorized as follows:

2.2.1 Difficulties of Designing Reliable FDI Systems

A reliable FDI system provides accurate information about the health status of the aircraft. In order to achieve such a result, the FDI system needs to be robust against external disturbances, model uncertainties and sensor noise. In addition, the FDI system should not trigger false alarms and should still be sufficiently sensitive to detect the faults.

Robustness is a fundamental issue in the performance of FDI systems and reconfigurable flight controllers [18, 19]. FDI systems may experience significant performance reduction if model uncertainties are not properly considered. A robustness analysis framework for failure detection and accommodation systems is provided in [18] and [20].

2.2.2 Interaction Between Flight Controllers and FDI Systems

It is often the case that a reconfigurable flight control system incorporates an FDI system and a flight controller. The FDI system monitors the aircraft's behavior and identifies relevant parameters that are usually used by the flight controller to synthesize the control commands. Therefore, the performance of the flight controller is dependent on the results provided by the FDI system and *vice versa*. Thus, the interactions between these two systems should be rigorously investigated.

The following observation is made in [19]: "it is fairly common for integration of failure detection and accommodation systems to be problematic if they are designed separately".

Challenges exist when some aircraft parameters need to be identified during the flight in real time and under feedback control [21]. This task is even more difficult and delicate when an actuator or a sensor fault happens. Moreover, the robustness of the flight controller can mask some aircraft faults and failures and make the detection problem more difficult.

There exist already many examples of integrated fault-tolerant control, sometimes referred to as IFTC [6, 19, 22–25].

New generations of reconfigurable flight control systems will not only rely on a fault-tolerant controller, but will include complete and integrated systems that reconfigure the flight controllers, adapt the guidance system, and reshape on-line the vehicle trajectories [25]. This book also provides an example of such a complete reconfigurable system applied to a small UAV.

2.2.3 Other Practical Challenges

Usually, the flight control system relies on some nominal values for the mass, the moments of inertia and the aerodynamic coefficients to generate the control signals. When the aircraft experiences an actuator failure or airframe damage, the aircraft becomes asymmetric. It is thus not trivial to determine which of these parameters need to be (on-line) re-estimated to keep good flying performance.

It is often the case that the available on-board processing power is limited, in particular for small or micro UAVs. The design of a reconfigurable flight controller is therefore a tradeoff between performance, complexity and available processing power.

Finally, challenges are many when the fault diagnostic, the reconfiguration of the control system, the reconfiguration of the path plan or of the mission is to be done autonomously under limited or no human supervision.

2.3 Different Approaches for FDI Systems

Table 2.1 provides a list of common and recent techniques that are encountered in the literature for the design of FDI systems.

2.3.1 Trends in Filter Design for FDI System

More than a decade ago in the mid-1990s, several implementations of recursive least squares (RLS) algorithms were used in FDI systems and successfully flight tested. For example, the work by Ward *et al.* in [21] describes a computationally efficient real-time parameter identification and reconfigurable control algorithm. The identification algorithm is based on a modified sequential least-squares (MSLS) found in [61], the recursive version of which is found in [27]. The MSLS parameter identification algorithm is based on RLS techniques and incorporates additional constraints to take into account *a priori* information and to adjust the size of the data window used in the regressor of the filter [26].

Table 2.1 List of some recent and popular techniques used to design FDI systems for flight applications

Technique	Example of recent books/papers using this technique (ordered chronologically)
(Modified -) RLS	[21, 26–28]
KF (bank of -)	[29–35]
EKF (bank of -)	[11, 13, 34, 36, 37]
UKF	[38–41]
LPV filters	[42]
Interaction matrix	[43]
Particle filters	[44]
Neural networks	[7, 45–48]
Statistical methods	[1, 49]
Wavelet analysis	[3, 7]
H_∞	[6, 50–52]
Robust model-based system	[18, 53, 54]
Parity space approach	[1, 18, 55–57]
Unknown input observer	[58–60]

Many FDI filters have also been designed using mathematical models of the system being monitored. Model-based FDI methods have been enhanced using robust FDI techniques as defined by Chen and Patton in [18]. It consists of incorporating during the design of FDI systems the effects of disturbance signals, model uncertainties and measurement noise [53]. It is often the case that several model-based filters are organized in a bank in which one filter is sensitive to a specified failure but the other filters remain insensitive to that failure. A very recent example of this technique can be found in [54] where a robust fault diagnosis for a spacecraft attitude control system is designed.

Many different variants of Kalman filters (KFs) have been constructed for detecting and isolating faults or for state estimation and state reconstruction. The use of extended Kalman filters (EKFs) applied to nonlinear systems for FDI purposes has also gained recent interest in [11, 13, 34] and in this book. A recent paper presented a method that uses EKFs to estimate on-line the aircraft's aerodynamic parameters and the components of the wind velocity. These estimates are used to update the parameters of the flight controller [36].

The unscented Kalman filter (UKF) is among the latest extensions of Kalman-type filters and seems to provide remarkable results for systems that are particularly nonlinear. The paper by Campbell [40] discusses the implementation of a sigma point filter (SPF) which was originally introduced as the UKF [38], where the distributions are approximated by a finite set of points. It is used to estimate aircraft states and aerodynamic derivatives in

real time. This is a nonlinear estimation algorithm that can be performed online, which possesses robustness properties against parameter uncertainties, against filter tuning and initial conditions.

The discussion in [38] explains that the SPF has similar performance to a truncated second-order EKF but without the need to calculate the Jacobian matrices. A comparison between EKF and SPF can also be found in [39]. The main results of this paper indicate that the SPF filter has equal or better performance than an EKF for real-time estimation application for the following reasons: the SPF is more robust against initial uncertainties and against jumps in the data, is less sensitive to tuning of the process noise, is less susceptible to divergence, is more accurate from one time step to the next and, finally, requires equivalent computational load. Very recent contributions in [41] focused on a new formulation for the state update equation of the filter for improved accuracy.

Recently, linear parameter-varying (LPV) filters gained the attention of some researchers in the fault-tolerant control community. For example, a design of LPV-based FDI filters is found in [42]. An example of an H_∞ control law that minimizes command tracking errors under actuator fault occurrence combined with an FDI filter based on an affine LPV model of a Boeing 747 is found in [19].

2.3.2 Trends in Active Fault Detection

Better, faster, and more reliable actuator and sensor fault-tolerant systems are being designed by exploiting the concept of active fault supervision. This consists of injecting artificial signals in actuators in such a way that a robust and reliable fault diagnosis can be made even if the system is excited very slightly.

Very few papers have discussed this technique so far. The work published by Honeywell in 1998 [8] and 2001 [9] is among the first occurrences of using artificial exciting signals for FDI purposes. Test signals are injected into the null space of the inputs using redundant control surfaces such that these signals (ideally) cancel one another and thereby do not excite aircraft motion [9], but contribute to better fault diagnosis; see also [62].

In 2006, the use of artificial signals was demonstrated to improve significantly the performance of an FDI system based on the extended multiple model adaptive estimation (EMMAE) method [11, 13]. This method is described in Chap. 4 of this book and in [13]. In 2007, the authors of [12] suggested an adaptive fault-tolerant controller with self fault-diagnosis actuators. This is done by generating high-frequency signals for actuators with suspected failures, and minimizing the effects of those signals on the system state using the remaining healthy actuators.

Recent contributions in active fault-diagnosis utilizing artificial excitation signals can also be found in [14] and [15].

2.4 Different Approaches for Flight Control Systems

Table 2.2 provides a list of the most common techniques that are encountered in the literature for the design of flight control systems.

Table 2.2 List of some recent and popular techniques used to design flight control systems

Technique	Example of recent books/papers using this technique (ordered chronologically)
P, PD, PI, PID	[25, 37, 63]
H_∞, LQ, LQG, LTR	[3–5, 64–67]
Dynamic inversion	[37, 47, 48, 62, 63, 68–74]
Quantitative feedback theory	[75, 76]
LPV	[77–79]
Model predictive control, Receding horizon	[4, 40]
Backstepping	[80–82]
Neural networks	[45–48, 74, 83]
Adaptive control	[4, 21, 29, 35, 84–86]
Model following	[4, 37, 47, 48]
Sliding mode control	[86–92]
Fuzzy logic	[4, 93]
Eigenstructure assignment	[4, 22, 94]

2.5 Techniques to Design Fault-tolerant Flight Control Systems

Table 2.3 provides a list of the most common techniques that are encountered in the literature for the design of fault-tolerant and reconfigurable flight control systems. Some of these techniques as described below.

Table 2.3 List of some recent and popular techniques used to design reconfigurable flight control systems

Technique	Example of recent books/papers using this technique (ordered chronologically)
Multiple model switching and tuning	[95, 96]
Multiple model adaptive control	[30, 32, 33, 35, 97]
Interacting multiple models	[44, 98]
Control allocation	[37, 47, 62, 82, 99–105]
Sliding mode control	[86, 90–92, 106–109]
Model predictive control	[40, 110]
Eigenstructure assignment	[22, 94, 98]
Model reference adaptive control	[28, 74, 84, 111]
Model reference + Dynamic inversion	[12, 25, 36, 37, 47, 48, 74, 84, 112]
Neural networks	[45–48, 74, 83]
Other recent fault compensation strategies	[113]

2.5.1 Multiple Model Techniques

2.5.1.1 Multiple Model Switching and Tuning

In the multiple model switching and tuning (MMST) technique shown in Fig. 2.3, the dynamics of each fault scenario are described by a dedicated model. Each model is paired with its respective controller. The control system is reconfigured by choosing the model/controller pair that is the most appropriate at each time step.

A switching logic module computes for each model i a performance index J_i, which is a function of the error e_i between the model M_i and the measurement data vector y. The performance index J_i is of the following form:

$$\begin{gathered} J_i(t) = \alpha e_i^2(t) + \beta \int_0^t \exp\left[-\lambda(t-\tau)\right] e_i^2(\tau) d\tau \\ \alpha \geq 0,\ \beta > 0,\ \lambda > 0\ . \end{gathered} \tag{2.1}$$

The coefficients α and β are responsible for the tradeoff between instantaneous and long-term contributions of the error e_i in the calculation of the index J_i. The coefficient λ is used as a forgetting factor.

The model M_i that produces the smallest performance index J is the closest to the current system, and therefore the controller K_i becomes active.

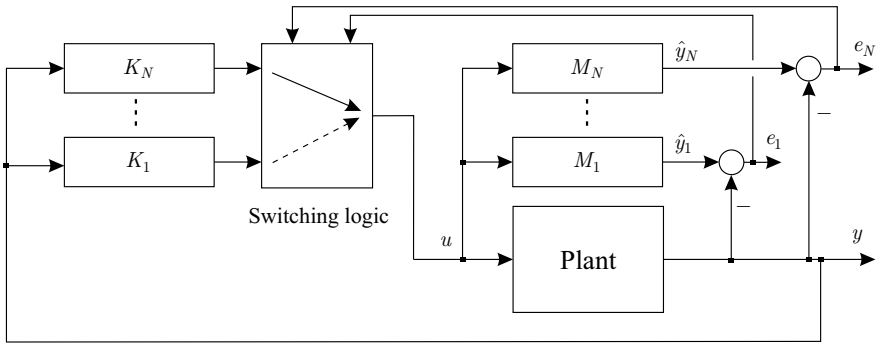


Fig. 2.3 MMST scheme

Most of the MMST reconfigurable schemes also include a tuning part, which is a separate identification algorithm that updates the parameters of the model M_i while the controller K_i is active.

The MMST technique has the advantages of being fast and usually stable if the actually occurring failures match the predefined fault scenarios. However, severe limitations of the method appear in practice as soon as an unmodeled failure is encountered or if multiple or structural failures occur. Moreover, the number of individual pairs of model/controller to be designed may become excessively large if the system is to successfully operate over a wide range of failure scenarios [95, 96].

2.5.1.2 Multiple Model Adaptive Estimation

Another approach to detect and isolate actuator or sensor faults is the multiple model adaptive estimation (MMAE) method [114] as depicted in Fig. 2.4. It is based on a bank of KFs running in parallel, each of which is matching a particular fault status of the system. A hypothesis testing algorithm uses the residuals from each KF to assign a conditional probability to each fault hypothesis. As one may expect, the computational load is quite intense. Therefore, the on-line use of this method was impractical for a long time. However, with the more powerful processors now available this method has regained appeal in many applications.

Several papers have demonstrated how the MMAE method can be used in the context of FDI systems and control reconfiguration for aircraft [30–32] and underwater vehicles [33]. The advantages and limitations of the MMAE method are discussed in detail in Chap. 4, where an extended and nonlinear FDI method is designed based on the MMAE technique.

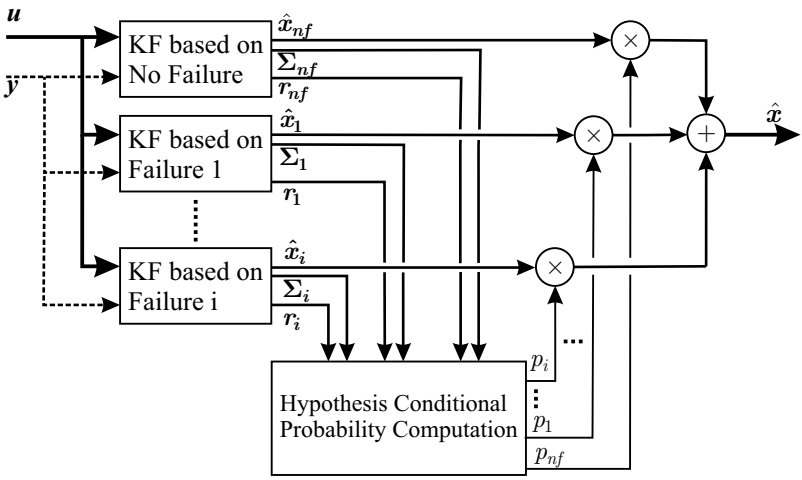


Fig. 2.4 Classical MMAE scheme

2.5.2 Control Allocation Techniques

Control allocation techniques are described in detail in Chap. 5. Briefly stated, the flight control system generates a virtual control command $\mathbf{C_v} = [C_L \quad C_M \quad C_N]^T$ in terms of the desired roll, pitch, and yaw torques. This virtual command $\mathbf{C_v}$ is passed to the control allocator, which is provided with each actuator's position limits and effectiveness to produce any torque component of the $\mathbf{C_v}$ vector. An algorithm is computed on-line to optimally generate the control signals for the actuators [99, 100, 103].

The biggest advantage of using a control allocation technique is that actuator failures can be compensated without the need for modifying the flight control laws [82]. Moreover, actuator constraints, such as deflection limits and motion rates, can be taken into account by the control allocator when the virtual command $\mathbf{C_v}$ is "distributed" over the actuators. Finally, the deflection of each actuator can be chosen by the control allocator to optimize some criteria, such as total drag, total deflections, or to prioritize some actuators.

However, as explained in [96], control allocation techniques may have the following disadvantage: "the dynamics and limitations of the actuators after a failure are not taken into account in the control laws. This means that the controller will still attempt to achieve the original system performance even though the actuators are not capable of achieving it". Chapters 8 and 10 explain how these disadvantages have been overcome in this book.

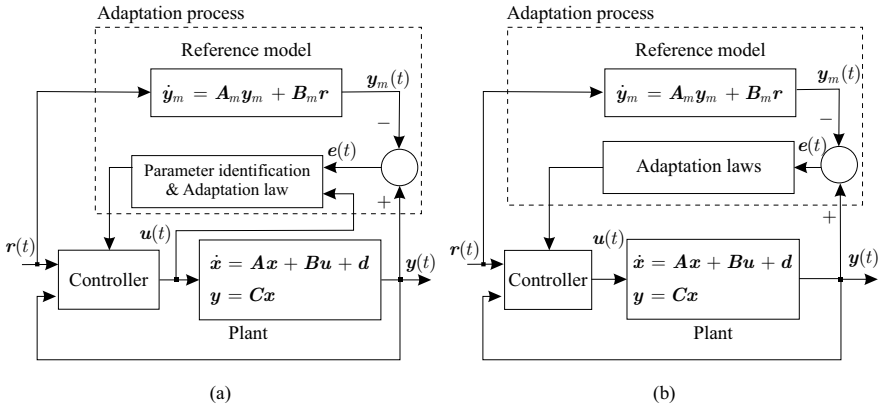


Fig. 2.5 MRAC scheme: (a) indirect controller parameter adaptation; and (b) direct controller parameter adaptation

2.5.3 Model Reference Adaptive Control

Model reference adaptive control (MRAC) [111, 115] is a method that can be utilized when tolerance to damage or structural failures is required. This technique is also often used as a final stage of a complex control system combining several algorithms. The goal is to have the output of the plant under consideration follow the output of a reference model.

The linearized plant under consideration is of the form

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} + \mathbf{d} \,, \\ \mathbf{y} &= \mathbf{C}\mathbf{x} \,, \end{aligned} \tag{2.2}$$

where the system state vector is $\mathbf{x} \in \mathbf{R}^n$, the control input vector is $\mathbf{u} \in \mathbf{R}^p$, and the measurement vector is $\mathbf{y} \in \mathbf{R}^k$. The reference model is of the form

$$\dot{\mathbf{y}}_m = \mathbf{A}_m \mathbf{y}_m + \mathbf{B}_m \mathbf{r} \,, \tag{2.3}$$

where the output vector of the reference model is $\mathbf{y}_m \in \mathbf{R}^k$ and the reference signal vector is $\mathbf{r} \in \mathbf{R}^l$. The dynamics matrix $\mathbf{A}_m \in \mathbf{R}^{k \times k}$ and $\mathbf{B}_m \in \mathbf{R}^{k \times l}$ can be chosen arbitrarily, but $\mathbf{A}_m$ must be stable [115].

In order to have the output of the plant follow the output of the reference model, the parameters of the controller are adapted by some adaptation laws as shown in Fig. 2.5. There are two types of adaptation, namely the indirect and the direct adaptation.

In the indirect adaptation, the matrices $\mathbf{A}$, $\mathbf{B}$, and the vector $\mathbf{d}$ are first estimated, for example using a least squares algorithm. In a second step, based on the estimates $\hat{\mathbf{A}}$, $\hat{\mathbf{B}}$ and $\hat{\mathbf{d}}$, the controller parameters are computed

by the adaptation laws such that the closed-loop system matches the desired dynamics of the reference model.

In the direct adaptation, the controller parameters are directly adapted such that the plant tracks the reference model.

However, the MRAC technique has some limitations. The adaptation laws require an estimation algorithm to track certain parameters of the system. It is therefore necessary that these system parameters evolve slowly enough in order that the estimation routine can track them properly. Faults or failures, however, may cause abrupt changes in the values of the system parameters. During the transient phase, in which the adaptive algorithm identifies the new faulty plant, it is not guaranteed that the controller can stabilize the system. Therefore, the MRAC technique is usually not used on its own but in combination with other algorithms in a more complicated fault-tolerant control architecture [25, 84].

2.5.4 Other Reconfigurable Control Methods

As shown in Table 2.3, there are other methods to design a reconfigurable flight control system. For instance, the *eigenstructure assignment* is used to reconfigure the feedback control laws in [98] and [94]. In *model predictive control* (MPC), the constraints on actuators or on any other state variable are systematically taken into account during the generation of the control signals [110]. A recent robust nonlinear MPC for low level aircraft control is presented in [40].

Sliding mode control has been investigated in [86, 90, 106–109]. The sliding mode control technique possesses some insensitivity and robustness properties to certain types of disturbance and uncertainty, which is an appealing feature for fault-tolerant flight control [87, 88]. The ability of sliding mode control to maintain the aircraft desired performance in case of faults without requiring an explicit FDI system makes it an other example of a passive approach to fault-tolerant control. Recent developments that employ sliding mode control for fault-tolerant control of a civil aircraft for both sensor and actuator faults are described in [89] and [91].

Other popular reconfigurable flight control systems use adaptive feedback linearization via artificial neural networks [45] or via on-line parameter identification methods [84]. New combinations of model reference and inverse dynamics controllers have been discussed in [84] and [25], and very recently in [12, 36, 74, 112, 116–118] and in this book.

2.6 Reconfigurable Guidance Systems

Over the last two decades, many path-planning algorithms have been investigated, especially for ground robots, single UAV, and more recently for a formation of UAVs. Among the methods used in path planning, we can mention the *probabilistic road maps* (PRM) method [119], which explores all the possible paths within the space surrounding the vehicle and finally selects the lowest cost route. However, the computational load makes the PRM method impractical for real-time path planning in small UAVs. An extension to the PRM method has recently been presented in [120]. It is called modified *rapidly-exploring random trees*, which is capable of efficiently searching for feasible paths in the space while taking into account constraints from the vehicle performance. However, efforts are still going on to implement an on-the-fly path-replanning system as pop-up obstacles are discovered or when the performance of the vehicle degrades.

There are other methods based on *potential field functions*. However, the primitive forms of potential field functions present some difficulties when choosing an appropriate potential function, and the algorithm may be stuck at some local minimum [121]. Since then, a whole family of potential field methods with superior performance has been developed. They are known as *navigation functions* [122, 123]. Other path-planning techniques are based on optimization methods, such as *mixed integer linear programming* or MPC techniques [124], which still involve intensive computations.

In this book, a reconfigurable guidance algorithm for a UAV is presented, which newly combines the lateral guidance control law from [125] and [126], originally designed for UAVs tracking circles for mid-air rendez vous, with a new, simple adaptive path-planning algorithm, which takes advantage of the curve path-following property of the above-mentioned lateral guidance law [127–130].

2.7 Real Flight Tests

One of the biggest challenge in UAV research is to fly test the algorithms developed. Several successful flight tests occurred over the past few years.

At the end of the 1990s, a significant milestone in the development of reconfigurable control systems was the flight testing with a NASA F15 aircraft of the so-called Self Repairing Flight Control System, which achieved failure and damage tolerance through an indirect adaptive reconfigurable flight control architecture that used an explicit FDI system to perform on-line damage and fault detection and estimation using hypothesis testing techniques associated with a bank of KFs [29].

Using an alternative approach based on an indirect adaptive control architecture, during the summer 1996, a series of flight tests demonstrated an

adaptive approach to reconfigurable flight control called Self Designing Controller. The results of the flight tests are reported in [21]. The fault scenario corresponded to the landing of an F-16 with a simulated missing elevon.

Another successful flight test was conducted as part of the Reconfigurable Systems for Tailless Fighter Aircraft also known as the RESTORE program. The technology used in this program combined a dynamic inversion control law in an explicit model-following framework. A neural network was also used for on-line learning of selected aircraft parameters and used in the feedback linearization loops [47, 48].

Flight tests of the US Air Force's Integrated Adaptive Guidance and Control program developed for the Boeing X-40A are reported in [25].

Flight testing of a simple reconfigurable control system on an autonomous model aircraft is reported in [28].

A number of flight tests of reconfigurable control systems were carried out as part of the Defense Advanced Research Projects Agency (DARPA) Software Enabled Control program on two unmanned combat aerial vehicles (UCAVs), namely the Boeing T-33/UCAV and Boeing X-45 UCAV [52].

NASA Langley Research Center is also actively involved in developing and testing reconfigurable systems for the new generation of re-entry vehicles, commercial aircraft and small UAVs [111, 131]. Several recovery systems and technologies were developed as part of the Aviation Safety program. High risk flight tests were conducted utilizing a dynamically scaled transport aircraft that has been developed at the NASA Langley Research Center as part of the Airborne Subscale Transport Aircraft Research testbed [132, 133].

The NASA/Boeing X-36 Tailless Fighter Agility Research Aircraft program successfully demonstrated the tailless fighter design using advanced technologies to improve the maneuverability and survivability of possible future fighter aircraft.

At the end of 2004 and beginning of 2005, the US Air Force and Boeing Company conducted a flight test of a modified MK-82 weapon at Eglin Air Force Base, which was controlled with a direct adaptive model reference flight control system that is capable of learning on-line some aerodynamics parameters with a neural-network algorithm similar to the one used in the RESTORE program [85].

More recently, the long endurance SeaScan UAV has been developed by the Insitu Group for weather reconnaissance and has recently been deployed in Iraq [40].

Finally, Honeywell Research Laboratories, Barron Associates, Inc., Scientific Systems Company, Inc., *etc.*, are among the companies actively involved in designing and flight testing fault-tolerant control systems.

References

1. R. Isermann. *Fault-Diagnosis Systems, An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag, Berlin Heidelberg, 2006.
2. J. D. Boskovic and R. K. Mehra. A Hybrid Fault-Tolerant Scheme for Flight Control Applications. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, August 2001. AIAA paper 2001-4400.
3. R. J. Adams. *Robust Multivariable Flight Control*. Springer-Verlag, 1994.
4. J.-F. Magni, S. Bennani, and J. Terlouw. *Robust Flight Control: A Design Challenge*. Springer-Verlag, London, 1997. In *Lecture Notes in Control and Information Sciences, 224*.
5. S. I. AlSwailem. *Application of Robust Control in Unmanned Vehicle Flight Control System Design*. PhD thesis, Cranfield University, 2004.
6. G. Hearns, M. J. Grimble, and M. A. Johnson. Integrated Fault Monitoring and Reliable Control. In *UKACC International Conference on CONTROL'98*, pages 1175–1179, 1−4 September 1998.
7. M. Azam, K. Pattipati, J. Allanach, S. Poll, and A. Petterson-Hine. In-Flight Fault Detection and Isolation in Aircraft Flight Control Systems. In *Proceedings of IEEE Aerospace Conference*, 2005. Paper 1429.
8. M. Elgersma, D. Enns, S. Shald, and P. Voulgaris. Parameter Identification for Systems with Redundant Actuators. In *Proceedings of the AIAA Guidance, Navigation and Control Conference and Exhibit*, Boston, MA, August 1998.
9. M. Elgersma and S. Glavaski. Reconfigurable Control for Active Managemement of Aircraft System Failures. In *Proceedings of IEEE American Control Conference*, pages 2627–2639, Arlington, VA, June 2001.
10. S. L. Campbell and R. Nikoukhah. *Auxiliary Signal Design for Failure Detection*. Princeton University Press, New Jersey, 2004.
11. G. Ducard and H. P. Geering. A Reconfigurable Flight Control System based on the EMMAE Method. In *Proceedings of the IEEE American Control Conference*, pages 5499–5504, Minneapolis, MN, June 2006.
12. J. D. Boskovic, J. Redding, and R. K. Mehra. Stable Adaptive Reconfigurable Flight Control with Self-Diagnostics. In *Proceedings of the IEEE American Control Conference*, pages 5765–5770, NY, July 2007.
13. G. Ducard and H. P. Geering. Efficient Nonlinear Actuator Fault Detection and Isolation System for Unmanned Aerial Vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 31(1):225–237, January-February 2008.
14. F. Bateman, H. Noura, and M. Ouladsine. Actuators Fault Diagnosis and Tolerant Control for an Unmanned Aerial Vehicle. In *Proceedings of the 16th IEEE International Conference on Control Applications, Part of IEEE Multiconference on Systems and Control*, pages 1061–1066, Singapore, October 2007.
15. F. Bateman, H. Noura, and M. Ouladsine. Active Fault Detection and Isolation Strategy for an Unmanned Aerial Vehicle with Redundant Flight Control Surfaces. In *Proceedings of the 16th Mediterranean Conference on Control and Automation*, pages 1246–1251, June 2008.
16. R. H. Chen and J. L. Speyer. Sensor and Actuator Fault Reconstruction. *AIAA Journal of Guidance, Control, and Dynamics*, 27(2):186–196, March-April 2004.
17. D. Rupp, G. Ducard, H. P. Geering, and E. Shafai. Extended Multiple Model Adaptive Estimation for the Detection of Sensor and Actuator Faults. In *Proceedings of IEEE Control and Decision Conference, and European Control Conference*, pages 3079–3084, Seville, Spain, 2005.
18. J. Chen and R. J. Patton. *Robust Model Based Diagnosis for Dynamic Systems*. Kluwer Academic, The Netherlands, 1999.

19. J-Y Shin, C. Belcastro, and T. Khong. Closed-loop Evaluation of an Integrated Failure Identification and Fault Tolerant Control System for a Transport Aircraft. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, 2006. AIAA 2006-6310.
20. C. Belcastro and B-C. Chang. Uncertainty Modeling for Robustness Analysis of Failure Detection and Accomodation Systems. In *Proceedings of the IEEE American Control Conference*, pages 4776–4782, 2002.
21. D. G. Ward, J. F. Monaco, and M. Bodson. Development and Flight Testing of a Parameter Identification Algorithm for Reconfigurable Control. *AIAA Journal of Guidance, Control, and Dynamics*, 21(6):948–956, Nov-Dec 1998.
22. Y. Zhang and J. Jiang. Integrated Design of Reconfigurable Fault-Tolerant Control Systems. *AIAA Journal of Guidance, Control, and Dynamics*, 24(1):133–136, 2000.
23. J. D. Boskovic, S. E. Bergstrom, and R. K. Mehra. Robust Integrated Flight Control Design Under Failures, Damage, and State-Dependant Disturbances. *AIAA Journal of Guidance, Control, and Dynamics*, 28(5):902–917, September-October 2005.
24. Y. M. Zhang and J. Jiang. Bibliographical Review on Reconfigurable Fault-Tolerant Control Systems. In *Proceedings of the IFAC Symposium*, pages 265–276, Washington, DC, 2003.
25. J. D. Schierman, D. G Ward, J. R. Hull, N. Gandhi, M. W. Oppenheimer, and D. B. Doman. Integrated Adaptive Guidance and Control for Re-Entry Vehicles with Flight-Test Results. *AIAA Journal of Guidance, Control, and Dynamics*, 27(6):975–988, November-December 2004.
26. D. Ward, R. L. Barron, M. P. Carley, and T. J. Curtis. Real-time Parameter Identification for Self-designing Flight Control. In *Proceedings of the National Aerospace and Electronics Conference (NAECON)*, Dayton, OH, 1994.
27. M. Bodson. An Adaptive Algorithm with Information-Dependant Data Forgetting. In *Proceedings of the IEEE American Control Conference*, pages 3485–3489, Seattle, WA, 1995.
28. D. Shore and M. Bodson. Flight Testing of a Reconfigurable Control System on an Unmanned Aircraft. *AIAA Journal of Guidance, Control, and Dynamics*, 28(4):698–707, July-August 2005.
29. J. Urnes, R. Yeager, and J. Stewart. Flight Demonstration of the Self-repairing Flight Control System in a NASA F-15 Aircraft. In *National Aerospace Electronics Conference*, Dayton, OH, May 1990. Rept. 90CH2881-1.
30. P. S. Maybeck and R. D. Stevens. Reconfigurable Flight Control Via Multiple Model Adaptive Control Methods. *IEEE Transactions on Aerospace and Electronic Systems*, 27(3):470–479, May 1991.
31. P. Eide and P. S. Maybeck. An MMAE Failure Detection System for the F-16. *IEEE Transactions on Aerospace and Electronic Systems*, 32(3):1125–1136, July 1996.
32. P. S. Maybeck. Multiple Model Adaptive Algorithms for Detecting and Compensating Sensor and Actuator/Surface Failures in Aircraft Flight Control Systems. *International Journal of Robust and Nonlinear Control*, 9(14):1051–1070, 1999.
33. L. Ni. *Fault-Tolerant Control of Unmanned Underwater Vehicles*. PhD thesis, VA Tech. Univ., Blacksburg, VA, 2001.
34. C. Hajiyev and F. Caliskan. *Fault-Diagnosis and Reconfiguration in Flight Control Systems*. Springer, 2003. ISBN: 978-1-4020-7605-3.
35. S. Fekri, M. Athans, and A. Pascoal. Issues, Progress and New Results in Robust Adaptive Control. *International Journal of Adaptive Control and Signal Processing*, 20(10):519–579, 2006.
36. N. Tanaka, S. Suzuki, K. Masui, and H. Tomita. Restructurable Guidance and Control for Aircraft with Failures Considering Gusts Effects. *AIAA Journal of Guidance, Control, and Dynamics*, 29(3):635–642, May-June 2006.

37. G. J. J. Ducard. *Fault-tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle*. PhD thesis, ETH Zürich, 2007. Diss. No. 17505.
38. S. Julier, J. Uhlmann, and H. F. Durrant-Whyte. A New Method for the Nonlinear Transformation of Means and Covariances in Filters and Estimators. *IEEE Transactions on Automatic Control*, 45(3):477–482, 2000.
39. S. Brunke and M. Campbell. Estimation Architecture for Future Autonomous Vehicles. In *Proceedings of the IEEE American Control Conference*, pages 1108–1114, 2002.
40. M. E. Campbell, J. W. Lee, E. Scholte, and D. RathBun. Simulation and Flight Test of Autonomous Aircraft Estimation, Planning, and Control Algorithms. *AIAA Journal of Guidance, Control, and Dynamics*, 30(6):1597–1609, November-December 2007.
41. L. Perea and P. Elosegui. New State Update Equation for the Unscented Kalman Filter. *AIAA Journal of Guidance, Control, and Dynamics*, 31(5):1500–1503, September-October 2008.
42. I. Szaszi, A. Marcos, G. Balas, and J. Bokor. Linear Parameter-varying Detection Filter Design for a Boeing 747-100/200 Aircraft. *AIAA Journal of Guidance, Control, and Dynamics*, 28(3):461–470, 2005.
43. B. H. Koh, Z. Li, P. Dharap, S. Nagarajaiah, and M. Q. Phan. Actuator Failure Detection Through Interaction Matrix Formulation. *AIAA Journal of Guidance, Control, and Dynamics*, 28(5):895–901, 2005.
44. I. Rapoport and Y. Oshman. Fault-Tolerant Particle Filtering by Using Interacting Multiple Model-Based Rao-Blackwellisation. *AIAA Journal of Guidance, Control, and Dynamics*, 28(6):1171–1177, 2005.
45. A. J. Calise, S. Lee, and M. Sharma. Direct Adaptive Reconfigurable Control of a Tailless Fighter Aircraft. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Boston, MA, August 1998.
46. A. Younghwan. *A Design Of Fault Tolerant Flight Control Systems For Sensor And Actuator Failures Using On-Line Learning Neural Networks*. PhD thesis, West Virginia University, 1998.
47. K. Wise, J. Brinker, A. Calise, D. Enns, M. Elgersma, and P. Voulgaris. Direct Adaptive Reconfigurable Flight Control For A Tailless Advanced Fighter Aircraft. *International Journal of Robust Nonlinear Control*, 9:999–1012, 1999.
48. J. Brinker and K. A. Wise. Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August 2000.
49. P. A. Samara, G. N. Fouskitakis, J. S. Sakellariou, and S. D. Fassois. A Statistical Method for the Detection of Sensor Abrupt Faults in Aircraft Control Systems. *IEEE Transactions on Control Systems Technology*, 16(4):789–798, July 2008.
50. E. G. Collins and T. Song. Robust H_∞ Estimation and Fault Detection of Uncertain Dynamic Systems. *AIAA Journal of Guidance, Control, and Dynamics*, 23(5):857–864, September-October 2000.
51. A. Marcos, S. Ganguli, and G. J. Balas. An Application of H_∞ Fault Detection and Isolation to a Transport Aircraft. *Control Engineering Practice*, 13:105–119, 2005.
52. H. P. Rotstein, R. Ingvalson, T. Keviczky, and G. J. Balas. Fault-Detection Design for Unhinhabited Aerial Vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 29(5):1051–1060, September-October 2006.
53. R. J. Patton, P. M. Frank, and R. N. Clark. *Issues of Fault Diagnosis for Dynamic Systems*. Springer-Verlag, London, 2000.
54. R. J. Patton, F. J. Uppal, S. Simani, and B. Polle. Reliable Fault Diagnosis Scheme for a Spacecraft Control System. *Journal of Risk and Reliability*, 222:139–152, 2008. DOI: 10.1243/1748006XJRR98.

55. E. Y. Chow and A. S. Willsky. Analytical Redundancy and the Design of Robust Detection Systems. *IEEE Transactions on Automatic Control*, 29(7):603–614, 1984.
56. P. Frank. Enhancement of Robustness in Observer-Based Fault Detection. *International Journal of Control*, 59(4):955–984, 1994.
57. J. J. Gertler. Fault Detection and Isolation Using Parity Relations. *Control Engineering Practice*, 5(5):653–661, 1997.
58. R. J. Patton, P. M. Frank, and R. N. Clark. *Fault Diagnosis in Dynamic Systems: Theory and Applications*. Prentice-Hall, Englewood Cliffs, NJ, 1989.
59. J. Chen, R. J. Patton, and H. Zhang. Design of Unknown Input Observers and Robust Fault Detection Filters. *International Journal of Control*, 63(1):85–105, 1996.
60. R. J. Patton and J. Chen. Observer-Based Fault Detection and Isolation: Robustness and Applications. *Control Engineering Practice*, 5(5):671–682, 1997.
61. D. Ward and R. Barron. A Self-Designing Receding Horizon Optimal Flight Controller. In *Proceedings of the IEEE American Control Conference*, pages 3490–3494, Seattle, WA, 1995.
62. J. Buffington, P. Chandler, and M. Pachter. On-Line Identification For Aircraft With Distributed Control Effectors. *AIAA Journal of Guidance, Control, and Dynamics*, 9:1033–1049, 1999.
63. D. Enns and T. Keviczky. Dynamic Inversion Based Flight Control for Autonomous RMAX Helicopter. In *Proceedings of the IEEE American Control Conference*, pages 3916–3923, Minneapolis, MN, June 2006.
64. Z. M. Smit and I. K. Craig. Robust Flight Controller Design using H_∞ Loop-Shaping and Dynamic Inversion Techniques. Paper AIAA-98-4132, pp. 305−313, 1998.
65. R. D. Colgren. *Applications of Robust Control to Nonlinear Systems*. Volume 205, Progress in Astronautics and Aeronautics, AIAA Inc., Reston, VA, 2004.
66. H. P. Geering. *Robuste Regelung*. IMRT Press, ETH Zurich, 2004. 3rd Edition.
67. M. Sadraey and R. Colgren. Robust Nonlinear Controller Design for a Complete UAV Mission. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, 2006.
68. D. Enns. Robustness of Dynamic Inversion vs. μ Synthesis: Lateral Directional Flight Control Example. AIAA paper 90-3338-CP, 1990.
69. M. G. Goman and E. N. Kolesnikov. Robust Nonlinear Dynamic Inversion Method for an Aircraft Motion Control. Paper AIAA-98-4208, pp. 511−521, 1998.
70. M. B. McFarland and S. M. Hoque. Robustness of a Nonlinear Missile Autopilot Designed Using Dynamic Inversion. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, 14−17 August 2000. Paper AIAA-2000-3970.
71. D. Itô, D. T. Ward, and J. Valasek. Robust Dynamic Inversion Controller Design and Analysis for the X38. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, 6−9 August 2001. Paper 2001-4380.
72. D. B. Doman and A. D. Ngo. Dynamic Inversion-Based Adaptive/Reconfigurable Control of the X-33 on Ascent. *AIAA Journal of Guidance, Control, and Dynamics*, 25(2):275–284, 2002.
73. D. Enns and G. Papageorgiou. Dynamic Inversion Tutorial - Theory and Examples of Applications. In *Workshop at the IEEE American Control Conference*, Minneapolis, MN, June 2006.
74. Y. Shin, A. J. Calise, and M. D. Johnson. Adaptive Control of Advanced Fighter Aircraft in Nonlinear Flight Regimes. *AIAA Journal of Guidance, Control, and Dynamics*, 31(5):1464–1477, September-October 2008.

75. P. L. Fontenrose and C. E. Hall. Development and Flight Testing of Quantitative Feedback Theory Pitch Rate Stability Augmentation System. *AIAA Journal of Guidance, Control, and Dynamics*, 19:1109–1115, 1996.
76. S. J. Rasmussen and C. H. Houpis. Development, Implementation and Flight Test of a MIMO Digital Flight Control System for an Unmanned Research Vehicle Designed Using Quantitative Feedback Theory. *International Journal of Robust and Nonlinear Control*, 7:629–642, 1997.
77. S. Bennani, R. van der Sluis, G. Schram, and J. A. Mulder. Control Law Reconfiguration Using Robust Linear Parameter Varying Control. 1999. Paper AIAA-99-4137.
78. G. Papageorgiou, K. Glover, G. D'Mello, and Y. Patel. Taking Robust LPV Control into Flight on the VAAC Harrier. In *Proceedings of IEEE Control and Decision Conference*, pages 4558–4564, Sydney, Australia, 2000.
79. G. Papageorgiou and R. A. Hyde. Analyzing the Stability of NDI-based Flight Controllers with LPV Methods. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001.
80. M. L. Steinberg and A. B. Page. Nonlinear Adaptive Flight Control with Genetic Algorithm Design Optimization. *International Journal of Robust and Nonlinear Control*, 9:1097–1115, 1999.
81. J. Farrell, M. Polycarpou, and Manu Sharma. Adaptive Backstepping with Magnitude, Rate, and Bandwidth Constraints: Aircraft Longitude Control. In *Proceedings of IEEE American Control Conference*, pages 3898–3903, Denver, CO, June 2003.
82. O. Härkegård. *Backstepping and Control Allocation with Applications to Flight Control*. PhD thesis, Linköping University, Sweden, 2003.
83. S. Ferrari and R. F. Stengel. Online Adaptive Critic Flight Control. *AIAA Journal of Guidance, Control, and Dynamics*, 27(5):777–786, September-October 2004.
84. M. Bodson. Reconfigurable Nonlinear Autopilot. *AIAA Journal of Guidance, Control, and Dynamics*, 26(5):719–727, 2003.
85. K. A. Wise, E. Lavretsky, and N. Hovakimyan. Adaptive Control of Flight: Theory, Applications, and Open Problems. In *Proceedings of the IEEE American Control Conference*, pages 5966–5971, Minneapolis, MN, June 2006.
86. C. I. Ahn, Y. Kim, and H. J. Kim. Adaptive Sliding Mode Controller Design for Fault Tolerant Flight Control System. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 2006. AIAA paper 2006-6089.
87. V. I. Utkin. *Sliding Modes in Control Optimization*. Springer-Verlag, Berlin, Germany, 1992.
88. S. N. Singh, M. L. Steinberg, and A. B. Page. Nonlinear Adaptive and Sliding Mode Flight Path Control of F/A-18 Model. *IEEE Transactions on Aerospace and Electronic Systems*, 39(4):1250–1262, October 2003.
89. H. Alwi and C. Edwards. Fault Tolerant Control of a Civil Aircraft Using a Sliding Mode Based Scheme. In *Proceedings of IEEE Control and Decision Conference, and European Control Conference*, pages 1011–1016, Seville, Spain, 2005.
90. M. L. Corradini, G. Orlando, and G. Parlangeli. A Fault Tolerant Sliding Mode Controller for Accomodating Actuator Failures. In *Proceedings of IEEE Control and Decision Conference, and European Control Conference*, pages 3091–3096, Seville, Spain, 2005.
91. H. Alwi and C. Edwards. Fault Detection and Fault-tolerant Control of a Civil Aircraft Using a Sliding-Mode-Based Scheme. *IEEE Transactions on Control Systems Technology*, 16(3):499–510, May 2008.

92. H. Alwi, C. Edwards, O. Stroosma, and J. A. Mulder. Fault Tolerant Sliding Mode Control Design with Piloted Simulator Evaluation. *AIAA Journal of Guidance, Control, and Dynamics*, 31(5):1186–1201, September-October 2008.
93. G. Schram, G. J. C. Copinga, P. M. Bruijn, and H. B. Verbruggen. Failure Tolerant Control of Aircraft: a Fuzzy Logic Approach. In *Proceedings of the IEEE American Control Conference*, pages 2281–2285, Philadelphia, PA, 1998.
94. Y. M. Zhang and J. Jiang. Active Fault-Tolerant Control System Against Partial Actuator Failures. *IEE Proceedings in Control Theory Application*, 149(1):95–104, January 2001.
95. J. D. Boskovic and R. K. Mehra. A Multiple Model-based Reconfigurable Flight Control System Design. In *Proceedings of the 37th IEEE Conference on Decision and Control*, pages 4503–4508, Tampa, FL, December 1998.
96. C. N. Jones. Reconfigurable Flight Control, First Year Report. Available at http://www-control.eng.cam.ac.uk/ cnj22/docs/yearone.pdf, September 2002.
97. M. Athans. The Stochastic Control of the F-8C Aircraft Using a Multiple Model Adaptive Control (MMAC) Method - Part I: Equilibrium Flight. *IEEE Transactions on Automatic Control*, 22:768–780, 1977.
98. Y. Zhang and J. Jiang. Integrated Active Fault-Tolerant Control Using IMM Approach. *IEEE Transactions on Aerospace and Electronic Systems*, 37(4):1221–1235, October 2001.
99. W. Durham. Constrained Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 16(4):717–725, 1993.
100. J. Buffington and D. Enns. Lyapunov Stability Analysis of Daisy Chain Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 19(6):1226–1230, 1996. AIAA paper 1998-4487.
101. J. Buffington, P. Chandler, and M. Pachter. Integration of On-line System Identification and Optimization-based Control Allocation. 1998. AIAA paper 1998-4487.
102. Y. Ikeda and M. Hood. An Approach of L1 Optimization to Control Allocation. August 2000. AIAA paper 2000-4566.
103. M. Bodson. Evaluation of Optimization Methods for Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 25(4):703–711, 2002.
104. J. Jin. Modified Pseudoinverse Redistribution Methods for Redundant Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 28(5):1076–1079, 2005.
105. G. Ducard, H. P. Geering, and E. Dumitrescu. Efficient Control Allocation for Fault Tolerant Embedded Systems on Small Autonomous Aircrafts. In *Proceedings of the 1st IEEE Symposium on Industrial Embedded Systems*, pages 1–10, Antibes, Juan les pins, France, October 2006.
106. C. Edwards and S. K. Spurgeon. *Sliding Mode Control: Theory and Applications*. Taylor & Francis, New York, 1998.
107. Y. B. Shtessel and J. Buffington. Multiple Time Scale Flight Control using Reconfigurable Sliding Modes. *AIAA Journal of Guidance, Control, and Dynamics*, 22(6):873–883, November-December 1999.
108. Y. Shtessel, J. Buffington, and S. Banda. Tailless Aircraft Flight Control Using Multiple Time Scale Reconfigurable Sliding Modes. *IEEE Transactions on Control Systems Technology*, 10(3):288–296, May 2002.
109. R. A. Hess and S. R. Wells. Sliding Mode Control Applied To Reconfigurable Flight Control Design. *AIAA Journal of Guidance, Control, and Dynamics*, 26:452–462, 2003.
110. J. M. Maciejowski and C. N. Jones. MPC Fault-Tolerant Flight Control Case Study: Flight 1862. In *Proceedings of the IFAC SAFEPROCESS Conference*, Washington DC, June 2003.

111. G. Tao, S. Chen, X. Tang, and S. M. Joshi. *Adaptive Control of Systems with Actuator Failures*. Springer-Verlag, London Berlin Heidelberg, 2004.
112. J. D. Boskovic, J. Redding, and R. K. Mehra. Robust Fault Tolerant Flight Control using a New Failure Parameterization. In *Proceedings of the IEEE American Control Conference*, pages 5753–5758, NY, July 2007.
113. J. Cieslak, D. Henry, A. Zolghadri, and P. Goupil. Development of an Active Fault-Tolerant Flight Control Strategy. *AIAA Journal of Guidance, Control, and Dynamics*, 31(1):135–147, January-February 2008.
114. D. T. Magill. Optimal Adaptive Estimation of Sampled Stochastic Processes. *IEEE Transactions on Automatic Control*, 10(4):434–439, October 1965.
115. E. Shafai. *Einführung in die Adaptive Regelung*. Lecture Notes, IMRT, ETH Zurich, 2003.
116. M. D. Tandale and J. Valasek. Fault-Tolerant Structured Adaptive Model Inversion Control. *AIAA Journal of Guidance, Control, and Dynamics*, 29(3):635–642, May-June 2006.
117. G. Ducard and H. P. Geering. Airspeed Control for Unmanned Aerial Vehicles: a Nonlinear Dynamic Inversion Approach. In *Proceedings of the IEEE 16th Mediterranean Conference on Control and Automation*, pages 676–681, Ajaccio, Corsica, France, June 2008.
118. G. Ducard and H. P. Geering. Stability Analysis of a Dynamic Inversion Based Pitch Rate Controller for an Unmanned Aircraft. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 360–366, Nice, France, September 2008.
119. L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, August 1996.
120. J. N. Amin, J. D. Boskovic, and R. K. Mehra. A Fast and Efficient Approach to Path Planning for Unmanned Vehicles. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 21−24 2006.
121. Y. Koren and J. Borenstein. Potential Fields Methods and their Inherent Limitations for Mobile Robot Navigation. In *Proceedings of IEEE Conference on Robotics and Automation*, Sacramento, CA, April 1991.
122. S. G. Loizou and K. J. Kyriakopoulos. Closed-loop Navigation for Multiple Holonomic Vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2861–2866, Minneapolis, Minnesota, 2002.
123. E. Rimon and D. Koditschek. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.
124. Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How. Decentralized Robust Receding Horizon Control for Multi-vehicle Guidance. In *Proceedings of IEEE American Control Conference*, pages 2047–2052, Minneapolis, MN, June 2006.
125. S. Park. *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles*. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Available at http://hdl.handle.net/1721.1/16662, Cambridge, MA, 2004.
126. S. Park, J. Deyst, and J. P. How. A New Nonlinear Guidance Logic for Trajectory Tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, 2004.
127. G. Ducard and H. P. Geering. A Computationally Efficient Guidance System for a Small UAV. In *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics*, Angers, France, May 2007.

128. G. Ducard and E. Dumitrescu. Discrete Controller Synthesis for Dependable Trajectory Generation. In *Proceedings of the Safety and Reliability Conference*, Stavanger, Norway, June 2007.
129. G. Ducard, K. C. Kulling, and H. P. Geering. A Simple and Adaptive On-Line Path Planning System for a UAV. In *Proceedings of the IEEE 15th Mediterranean Conference on Control and Automation*, pages 1–6, Athens, Greece, June 2007. T34-009.
130. G. Ducard, K. C. Kulling, and H. P. Geering. Evaluation of Reduction in the Performance of a Small UAV after an Aileron Failure for an Adaptive Guidance System. In *Proceedings of the IEEE American Control Conference*, pages 1793–1798, New York, NY, July 2007.
131. M. A. Motter. Autonomous Flying Controls Testbed. Technical report, NASA Langley Research Center, 2005.
132. R. Bailey, R. Hostetler, K. Barnes, Celeste Belcastro, and Christine Belcastro. Experimental Validation: Subscale Aircraft Ground Facilities And Integrated Test Capability. In *AIAA Guidance, Navigation, and Control Conference*, 2005. Paper AIAA 2005-6433.
133. T. Jordan, W. Langford, and J. Hill. Airborne Subscale Transport Aircraft Research Testbed-Aircraft Model Development. In *AIAA Guidance, Navigation, and Control Conference*, 2005. Paper AIAA 2005-6432.

Chapter 3
Nonlinear Aircraft Model

This chapter presents the axes, the frames, and the nonlinear model of the aircraft used in this book [1–4].

3.1 Definitions of the Frames

3.1.1 Navigation Frame

The orientation of the navigation frame is North, East, Down (x_n y_n z_n). This frame does not move and is attached to the earth's local tangent plane. When the plane is on the ground before taking off, the origin O_n of the navigation frame is initialized by the position of the airplane's center of mass. In the rest of this book, the navigation frame is considered as a local inertial frame where Newton's laws apply.

3.1.2 Body Frame

The positive x axis of the body frame points forward along the aircraft's longitudinal axis, the positive y axis is directed along the right wing, the positive z axis is normal to the x and y axes, pointing downward. The origin O_b is located at the aircraft's center of mass. This defines a right-handed orthogonal body coordinate frame (x_b y_b z_b) attached to the aircraft as shown in Fig. 3.1.

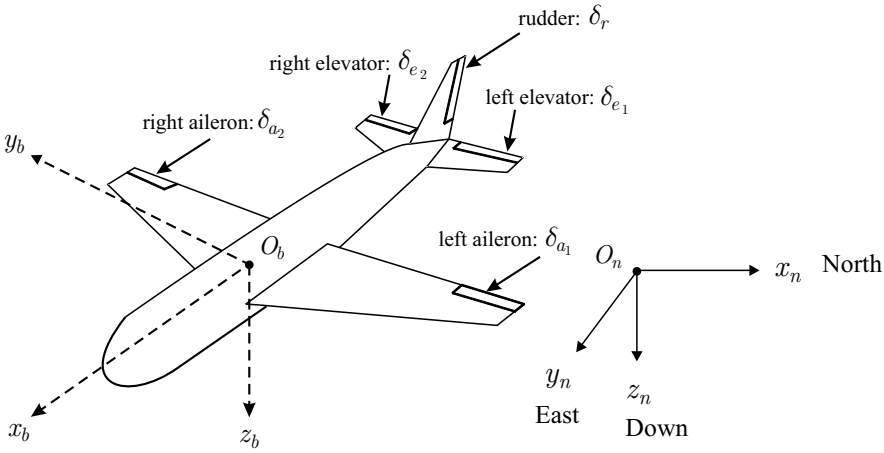


Fig. 3.1 Aircraft configuration

3.1.3 Euler Angles

Three Euler angle rotations continuously relate the orientation of the aircraft's body-fixed frame to the navigation frame.

As shown in Figs. 3.2 and 3.3, the navigation coordinate frame is first transformed into the intermediate frame 1 via a rotation about the z_n axis by the angle ψ, which defines the aircraft's heading. This is followed by a rotation about the new y_1 axis by an angle θ, which defines the aircraft's elevation. Finally, the aircraft bank angle, ϕ, defines the rotation about the new x_2 axis. Figure 3.4 shows a 3D representation of the Euler angles describing the orientation of the body frame with respect to the navigation frame.

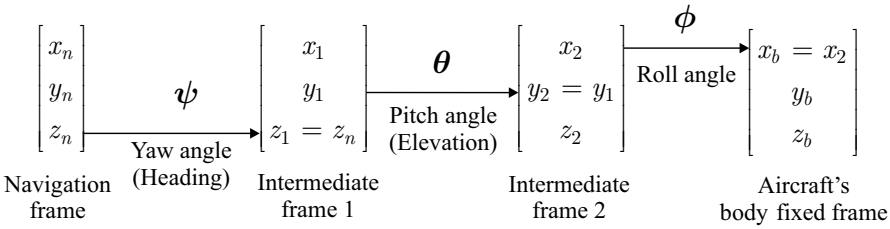


Fig. 3.2 Euler angles

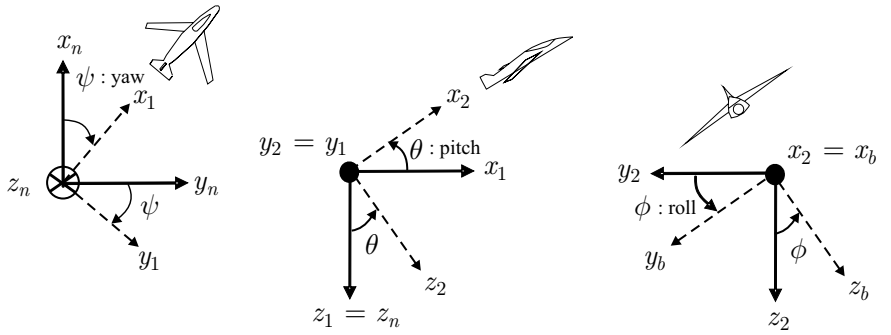


Fig. 3.3 Euler angles and frame transformation

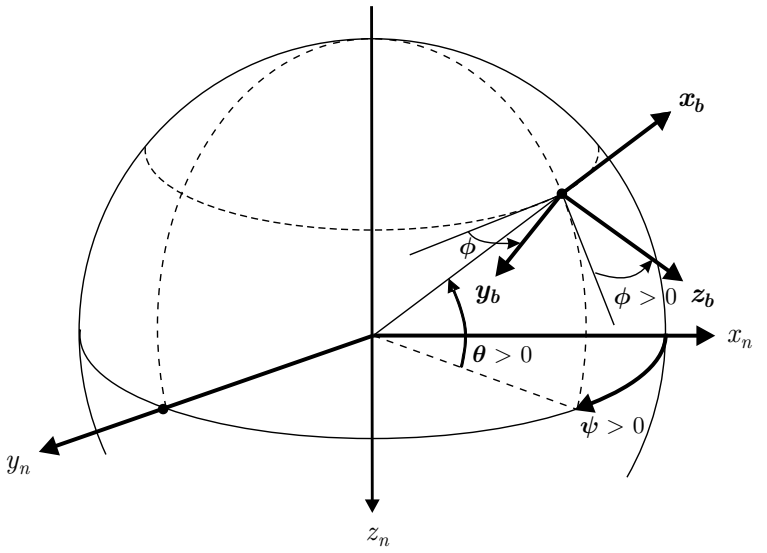


Fig. 3.4 3D representation of the Euler angles

3.1.4 Direction Cosine Matrix

The attitude transformation matrix (also called direction cosine matrix) is necessary to transform vectors and point coordinates from the aircraft's body-fixed frame (b) to the navigation frame (n) and *vice versa*.

The direction cosine matrix $\mathbf{C_n^b}$ transforms the vector $\mathbf{A}$ expressed in the navigation frame $\mathbf{A}^n$ into a vector expressed in the aircraft's body-fixed frame $\mathbf{A}^b$ as follows:

$$\mathbf{A}^b = \mathbf{C_n^b}\mathbf{A}^n, \tag{3.1}$$

with

$$\mathbf{C_n^b} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{pmatrix} \begin{pmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{pmatrix} \begin{pmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{pmatrix}, \tag{3.2}$$

yielding

$$\mathbf{C_n^b} = \begin{pmatrix} \cos\theta\cos\psi & \cos\theta\sin\psi & -\sin\theta \\ \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & \sin\phi\cos\theta \\ \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & \cos\phi\cos\theta \end{pmatrix}. \tag{3.3}$$

The direction cosine matrix $\mathbf{C_b^n}$, which transforms the vector $\mathbf{A}$ expressed in the aircraft's body-fixed frame $\mathbf{A}^b$ into a vector expressed in the navigation frame $\mathbf{A}^n$ is $\mathbf{C_b^n} = \left(\mathbf{C_n^b}\right)^{-1} = \left(\mathbf{C_n^b}\right)^T$.

3.1.5 Quaternion Representation

The orientation of the aircraft body-fixed frame (b) with respect to the navigation frame (n) can also be expressed with a quaternion representation as follows:

$$\mathbf{C_n^b} = \begin{bmatrix} 1-2({q_2}^2+{q_3}^2) & 2(q_1q_2+q_0q_3) & 2(q_1q_3-q_0q_2) \\ 2(q_1q_2-q_0q_3) & 1-2({q_1}^2+{q_3}^2) & 2(q_2q_3+q_0q_1) \\ 2(q_1q_3+q_0q_2) & 2(q_2q_3-q_0q_1) & 1-2({q_1}^2+{q_2}^2) \end{bmatrix},$$

where the quaternion $q^n = [q_0 \quad q_1 \quad q_2 \quad q_3]^T = \begin{bmatrix} \cos(\theta/2) \\ \sin(\theta/2)\mathbf{u^n} \end{bmatrix}$ transforms the navigation frame (n) into the body-fixed frame (b) by a rotation of an angle θ around the axis $\mathbf{u^n}$ as shown in Fig. 3.5. In [5], it is shown how elements of the quaternion can be expressed in terms of the Euler angles and *vice versa*.

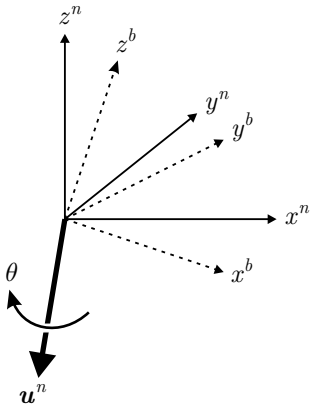


Fig. 3.5 Quaternion rotation from the navigation frame (n) to the body frame (b)

3.1.6 Wind Frame

The air flow acting on the airframe is responsible for the aerodynamic forces. The air flow is described by the airspeed vector $\mathbf{V_T}$. Its norm is V_T and its direction relative to the airframe is defined by two angles, namely the angle of attack α and the sideslip angle β.

As shown in Fig. 3.6, the angle of attack α is the angle between the projection of the airspeed vector $\mathbf{V}_T$ onto the (x_b, z_b) plane and the x_b axis. The sideslip angle β is the angle between the projection of the airspeed vector $\mathbf{V}_T$ onto the (x_b, z_b) plane and the airspeed vector itself. The wind axes coordinate system is such that the x_w axis points along the airspeed vector $\mathbf{V}_T$.

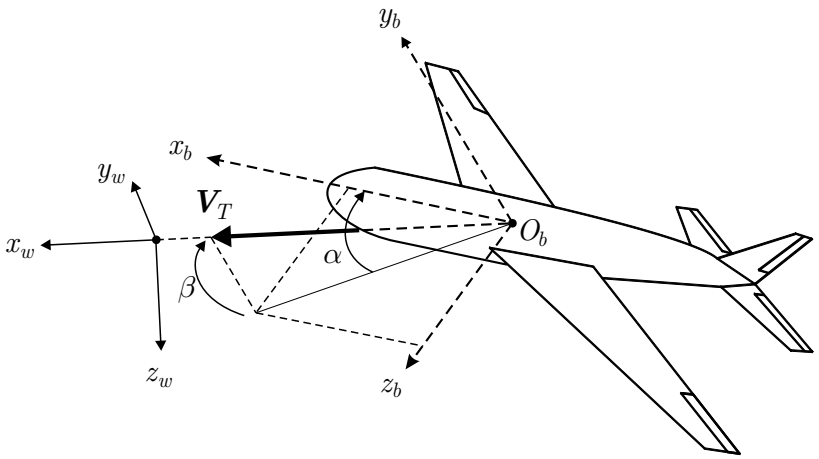


Fig. 3.6 Angle of attack and sideslip angle definition, $\alpha > 0$ and $\beta > 0$

The rotation matrix $\mathbf{C_b^w}$ is necessary to transform vectors and point coordinates from the aircraft body-fixed frame (b) to the wind frame (w) and *vice versa* according to the following formulae:

$$\mathbf{A}^w = \mathbf{C_b^w}\mathbf{A}^b \qquad \text{or} \qquad \mathbf{A}^b = (\mathbf{C_b^w})^{\mathbf{T}}\mathbf{A}^w = \mathbf{C_w^b}\mathbf{A}^w \,, \tag{3.4}$$

with

$$\begin{aligned} \mathbf{C_b^w} &= \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} , \\ &= \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\sin\beta\cos\alpha & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} . \end{aligned} \tag{3.5}$$

As an example, the airspeed vector is expressed in the body-fixed frame as follows:

$$\begin{aligned} \mathbf{V_T}^b &= \mathbf{C_w^b} \mathbf{V_T}^w \\ \begin{bmatrix} u_T \\ v_T \\ w_T \end{bmatrix} &= \mathbf{C_w^b} \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} . \end{aligned} \tag{3.6}$$

The subscript T is used to distinguish the coordinates of the airspeed vector $\mathbf{V_T}^b$ from the coordinates of the aircraft's inertial velocity vector $\mathbf{v}^b = [u \quad v \quad w]^T$.

3.2 Wind Disturbance

Several types of wind disturbances $\mathbf{W}$ will be introduced in the simulations of the following chapters to test the robustness of the control, guidance, and fault detection algorithms.

As found in [3], the aircraft's inertial velocity $\mathbf{v}$ is the sum of the airspeed $\mathbf{V_T}$ and the wind velocity $\mathbf{W}$ (see Fig. 3.7),

$$\mathbf{v} = \mathbf{V_T} + \mathbf{W} . \tag{3.7}$$

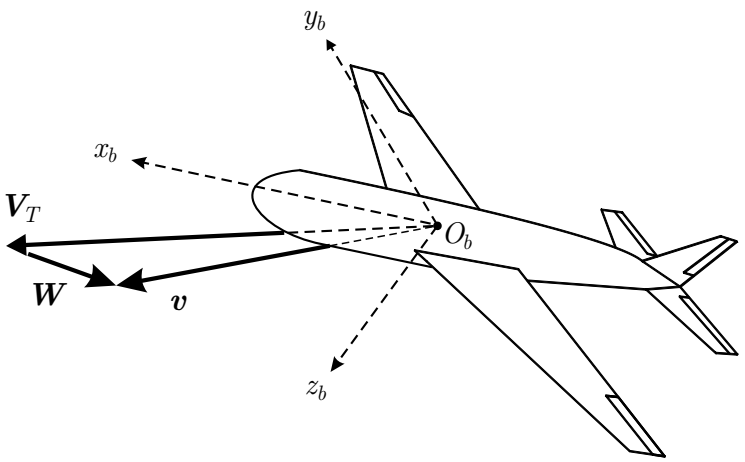


Fig. 3.7 Geometry of the airspeed vector $\mathbf{V_T}$, the wind disturbance $\mathbf{W}$, and the inertial velocity vector $\mathbf{v}$

For a wind disturbance given by its coordinates in the navigation frame $\mathbf{W}^n$, the vectors in (3.7) are projected in the body-fixed frame as follows:

$$\begin{aligned} \mathbf{v}^b &= \mathbf{V_T}^b + \mathbf{C_n^b}\mathbf{W}^n \,, \\ \iff \mathbf{V_T}^b &= \mathbf{v}^b - \mathbf{C_n^b}\mathbf{W}^n \,, \\ \begin{bmatrix} u_T \\ v_T \\ w_T \end{bmatrix} &= \begin{bmatrix} u \\ v \\ w \end{bmatrix} - \mathbf{C_n^b} \begin{bmatrix} W_N \\ W_E \\ W_D \end{bmatrix} \,. \end{aligned} \tag{3.8}$$

For the nonlinear simulations of the aircraft, the aerodynamic forces and torques are functions of the angle of attack α, the sideslip angle β, the airspeed V_T, and the dynamic pressure $\bar{q}$. They are defined as follows:

$$V_T = \sqrt{u_T^2 + v_T^2 + w_T^2} \,,\ \alpha = \arctan\left(\frac{w_T}{u_T}\right) \,,\ \beta = \arcsin\left(\frac{v_T}{V_T}\right) \,,\ \bar{q} = \frac{\rho V_T^2}{2} \,. \tag{3.9}$$

3.3 Model of the Low Altitude Atmosphere

The dynamic pressure $\bar{q} = \frac{\rho V_T^2}{2}$ is a key parameter involved in the aerodynamic forces, as described in Sect. 3.6. The air density ρ needs to be computed using the International Standard Atmosphere model as follows:

$$\begin{aligned} T &= T_0 \left[1 + ah/T_0\right] \,, \\ \rho &= \frac{p_0 \left[1 + ah/T_0\right]^{5.2561}}{RT} \,, \end{aligned} \tag{3.10}$$

where the temperature $T_0 = 288.15\,\mathrm{K}$, and the coefficients $a = -6.5 \times 10^{-3}\,\mathrm{K/m}$, $R = 287.3\,\mathrm{m^2 K^{-1} s^{-2}}$, and $p_0 = 1013 \times 10^2\,\mathrm{N m^{-2}}$. For example, the value of the air density calculated for an altitude $h = 500\,\mathrm{m}$ is $\rho = 1.166\,\mathrm{kg\,m^{-3}}$. This model of the low altitude atmosphere is valid up to an altitude of 11000 m.

3.4 Equations of Rigid-body Motion

In the following, we will consider the aircraft flying over a small region of the earth. We will assume that the earth is locally flat ($R_{earth} \to \infty$), and thus we neglect centripetal acceleration due to the earth curvature. We also neglect effects due to the coriolis acceleration, which consequently assumes the earth to be an inertial (or Galilean) frame, where Newton's laws are applicable.

3.4.1 Equations of Forces

Newton's second law relates the mass m of a system, the velocity vector $\mathbf{v}$ of its center of mass, and the force vectors $\mathbf{F_j}$ acting on the system. Subscript i indicates that time derivation is computed in the inertial frame as follows:

$$\sum_j \mathbf{F_j} = \Big[\frac{d}{dt}\Big(m\mathbf{v}\Big)\Big]_i \,. \tag{3.11}$$

Equation 3.12 recalls the relationship of the time derivative of an arbitrary time-dependent vector $\mathbf{U}(t)$ with respect to two different frames F_0 and F_1. The subscript next to the brackets in (3.12) indicates in which frame time derivation occurs. The vector describing the relative rotation of the frame F_1 with respect to the frame F_0 is $\boldsymbol{\omega}_{F_1/F_0}$ yielding

$$\Big[\frac{d\mathbf{U}(t)}{dt}\Big]_{F_0} = \Big[\frac{d\mathbf{U}(t)}{dt}\Big]_{F_1} + \boldsymbol{\omega}_{F_1/F_0} \times \mathbf{U}(t) \,. \tag{3.12}$$

The subscript b refers to the body-fixed frame attached to the aircraft. Equation 3.11 is rewritten as

$$\sum_j \mathbf{F_j} = \Big[\frac{d}{dt}\Big(m\mathbf{v}\Big)\Big]_i = \Big[\frac{d(m\mathbf{v})}{dt}\Big]_b + \boldsymbol{\omega}_{b/i} \times (m\mathbf{v}) \,. \tag{3.13}$$

Projecting the vectors of the previous equation in the aircraft body-fixed frame (b), and assuming the mass to be constant, yields

$$\frac{1}{m}\left[\sum_j \mathbf{F_j}^b\right] = \Big[\frac{d(\mathbf{v}^b)}{dt}\Big]_b + \boldsymbol{\omega}_{b/i}{}^b \times (\mathbf{v}^b) \,, \tag{3.14}$$

$$\frac{1}{m}\left[m\mathbf{g}^b + \mathbf{F}^b_{engine} + \mathbf{F}^b_{aerodynamic}\right] = \begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} + \begin{pmatrix}p\\ q\\ r\end{pmatrix} \times \begin{pmatrix}u\\ v\\ w\end{pmatrix} \,. \tag{3.15}$$

The equations for the motion of the aircraft are finally obtained as follows:

$$\begin{pmatrix}-g\sin\theta\\ g\sin\phi\cos\theta\\ g\cos\phi\cos\theta\end{pmatrix} + \frac{1}{m}\left[\begin{pmatrix}F_{thrust}\\ 0\\ 0\end{pmatrix} + \begin{pmatrix}X^b\\ Y^b\\ Z^b\end{pmatrix}\right] - \begin{pmatrix}qw - rv\\ ru - pw\\ pv - qu\end{pmatrix} = \begin{pmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\end{pmatrix} \,. \tag{3.16}$$

3.4.1.1 Rotational Velocities and Propagation of Aircraft's Angular Rates

The rotational velocity vector $\boldsymbol{\omega_{b/n}}$ describes the angular motion of the body frame (b) with respect to the navigation frame (n). The vector $\boldsymbol{\omega_{b/n}}$ can be expressed in either frame. We recall that the relative rotational velocities between three coordinate systems (a), (b), and (c) are related to each other as follows:

$$\boldsymbol{\omega_{a/c}} = \boldsymbol{\omega_{a/b}} + \boldsymbol{\omega_{b/c}} \ . \tag{3.17}$$

Figure 3.8 shows that the rotation velocity vector between:

- The intermediate frame 1 and the navigation frame is $\boldsymbol{\omega_{1/n}} = \dot{\psi}\mathbf{z_n} = \dot{\psi}\mathbf{z_1}$
- The intermediate frame 2 and 1 is $\boldsymbol{\omega_{2/1}} = \dot{\theta}\mathbf{y_1} = \dot{\theta}\mathbf{y_2}$
- The aircraft body-fixed frame and the intermediate frame 2 is $\boldsymbol{\omega_{b/2}} = \dot{\phi}\mathbf{x_2} = \dot{\phi}\mathbf{x_b}$.

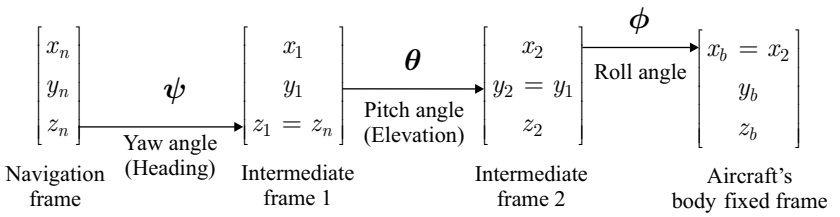


Fig. 3.8 Euler angles

The rotational velocity vector $\boldsymbol{\omega_{b/n}}$ can be constructed as follows:

$$\begin{aligned} \boldsymbol{\omega_{b/n}} &= \boldsymbol{\omega_{b/2}} + \boldsymbol{\omega_{2/1}} + \boldsymbol{\omega_{1/n}} \ , \\ &= \dot{\phi}\mathbf{x_b} + \dot{\theta}\mathbf{y_2} + \dot{\psi}\mathbf{z_1} \ . \end{aligned} \tag{3.18}$$

Projecting each rotational velocity vector in the body-fixed coordinate system, we obtain the following relationships:

$$\begin{aligned} {\boldsymbol{\omega_{b/n}}}^b &= \dot{\phi}\mathbf{x_b} + \dot{\theta}\left(\cos\phi\mathbf{y_b} - \sin\phi\mathbf{z_b}\right) + \dot{\psi}\left(\cos\theta\cos\phi\mathbf{z_b} + \cos\theta\sin\phi\mathbf{y_b} - \sin\theta\mathbf{x_b}\right) \ , \\ &= \left(\dot{\phi} - \dot{\psi}\sin\theta\right)\mathbf{x_b} + \left(\dot{\psi}\cos\theta\sin\phi + \dot{\theta}\cos\phi\right)\mathbf{y_b} + \left(\dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi\right)\mathbf{z_b} \ . \end{aligned}$$

By definition ${\boldsymbol{\omega_{b/n}}}^b = [p \quad q \quad r]^T$, where the aircraft's roll rate is p, its pitch rate is q, and its yaw rate is r. The relationship between the airplane turn rates p, q, r expressed in the body-fixed frame and the Euler attitude rates $\dot{\phi}$, $\dot{\theta}$, and $\dot{\psi}$ is

$$
\begin{aligned}
p &= \dot{\phi} - \dot{\psi}\sin\theta \,, \\
q &= \dot{\psi}\cos\theta\sin\phi + \dot{\theta}\cos\phi \,, \\
r &= \dot{\psi}\cos\theta\cos\phi - \dot{\theta}\sin\phi \,.
\end{aligned} \tag{3.19}
$$

In matrix form, the propagation of the Euler angles with time is as follows:

$$
\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \cos\theta\sin\phi \\ 0 & -\sin\phi & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \,, \tag{3.20}
$$

$$
\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \tan\theta\sin\phi & \tan\theta\cos\phi \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi/\cos\theta & \cos\phi/\cos\theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \,. \tag{3.21}
$$

The propagation of the aircraft's angular rates can also be computed with the quaternion representation. Indeed, the attitude of the aircraft defined in terms of the quaternion $\mathbf{q}^n$ can be obtained after integration of the following formula:

$$
\begin{aligned}
\dot{\mathbf{q}}^n &= \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} {\boldsymbol{\omega}_{\mathbf{b/n}}}^b \,, \\
\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} &= \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \,.
\end{aligned} \tag{3.22}
$$

The nonlinear aircraft model used for simulations in this book only uses the quaternion formulation, because some computational and singularity issues occur when Euler angles are used in the vicinity of a pitch angle of 90°.

3.4.1.2 Aircraft Position and Propagation of Longitudinal Velocities

The position of the aircraft $\mathbf{p}^n$ expressed in the navigation frame (n) is obtained by integration of the ground speed vector $[\dot{x}_N \; \dot{x}_E \; \dot{x}_D]^T$, which is computed as follows:

$$\begin{aligned}\dot{\mathbf{p}}^n = \frac{d}{dt}\{\mathbf{p}^n\} = \frac{d}{dt}\{\mathbf{C}_b^n \mathbf{p}^b\} &= \dot{\mathbf{C}}_b^n \mathbf{p}^b + \mathbf{C}_b^n \dot{\mathbf{p}}^b, \qquad (\text{since } \mathbf{p}^b = 0) \\ &= \mathbf{C}_b^n \mathbf{v}^b, \\ \begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} &= \mathbf{C}_b^n \begin{bmatrix} u \\ v \\ w \end{bmatrix} . \end{aligned} \tag{3.23}$$

3.4.2 Equations of Moments

Newton's law for momentum in the local navigation frame considered as an inertial frame is [3]

$$\begin{aligned} \mathbf{M}^n &= \left.\frac{d}{dt}\right|_n \{\mathbf{I}^n \boldsymbol{\omega}_{\mathbf{b/n}}{}^n\} , \\ &= \frac{d}{dt}\{\mathbf{C}_b^n \mathbf{I}^b \boldsymbol{\omega}_{\mathbf{b/n}}{}^b\} , \\ &= \dot{\mathbf{C}}_b^n \mathbf{I}^b \boldsymbol{\omega}_{\mathbf{b/n}}{}^b + \mathbf{C}_b^n \dot{\mathbf{I}}^b \boldsymbol{\omega}_{\mathbf{b/n}}{}^b + \mathbf{C}_b^n \mathbf{I}^b \dot{\boldsymbol{\omega}}_{\mathbf{b/n}}{}^b , \quad {}^{1} \\ &= \mathbf{C}_b^n \boldsymbol{\omega}_{\mathbf{b/n}}{}^b \times \left(\mathbf{I}^b \boldsymbol{\omega}_{\mathbf{b/n}}{}^b\right) + \mathbf{C}_b^n \mathbf{I}^b \dot{\boldsymbol{\omega}}_{\mathbf{b/n}}{}^b , \\ \iff \mathbf{M}^b &= \boldsymbol{\omega}_{\mathbf{b/n}}{}^b \times \left(\mathbf{I}^b \boldsymbol{\omega}_{\mathbf{b/n}}{}^b\right) + \mathbf{I}^b \dot{\boldsymbol{\omega}}_{\mathbf{b/n}}{}^b , \\ \iff \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= \left(\mathbf{I}^b\right)^{-1} \left(\begin{bmatrix} L \\ M \\ N \end{bmatrix}^b - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I}^b \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) . \end{aligned} \tag{3.24}$$

The body-fixed inertia matrix of the aircraft is $\mathbf{I}^b = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix}$.

3.5 Engine

3.5.1 Engine Rate

The dynamics for the engine speed n are modeled by a first-order linear system with the time constant τ_n and the engine speed reference signal n_c as follows:

$$\dot{n} = -\frac{1}{\tau_n} n + \frac{1}{\tau_n} n_c . \tag{3.25}$$

[1] $\dot{\mathbf{I}}^b \approx 0$.

3.5.2 Thrust Force

The thrust force is generated by the propeller and can be expressed with dimensionless coefficients [6]. The dimensionless thrust coefficient is

$$C_{F_T}(J) = C_{F_{T1}} + C_{F_{T2}} J + C_{F_{T3}} J^2 \ , \tag{3.26}$$

with the ratio $J = \frac{V_T}{D\pi n}$, where the diameter of the propeller is D, the engine speed is n, and the airspeed is V_T. The thrust force is computed as follows:

$$F_T = \rho n^2 D^4 C_{F_T}(J) \ . \tag{3.27}$$

3.6 Model of the Aerodynamic Forces

3.6.1 Lift Force

The dimensionless lift coefficient is modeled as a linear function of the angle of attack α as

$$C_Z(\alpha) = C_{Z1} + C_{Z\alpha}\alpha \ . \tag{3.28}$$

The lift force is calculated by multiplying the lift coefficient by the wing surface S and the dynamic pressure $\bar{q}$ as follows:

$$Z^w = \bar{q} S C_Z(\alpha) \ .^2 \tag{3.29}$$

3.6.2 Lateral Force

The lateral force acting on the aircraft is mainly due to the fuselage, which is considered to be an inefficient wing with zero offset due to the symmetry of the airplane in the (x_b, z_b) plane, yielding

$$\begin{aligned} C_Y(\beta) &= C_{Y1}\beta \ , \\ Y^w &= \bar{q} S C_Y(\beta) \ . \end{aligned} \tag{3.30}$$

3.6.3 Drag Force

Due to the symmetry of the fuselage, minimum drag is obtained when the sideslip angle β is zero. The wing is not symmetric, therefore, minimum drag

[2] The superscript w indicates that the vector is expressed in the wind frame.

is obtained for an angle of attack different from zero. The dimensionless drag coefficient is approximated by a quadratic function in α and β according to

$$C_X(\alpha, \beta) = C_{X1} + C_{X\alpha}\alpha + C_{X\alpha 2}\alpha^2 + C_{X\beta 2}\beta^2 \ . \tag{3.31}$$

The drag force is obtained by multiplying the drag coefficient by the dynamic pressure and the wing surface resulting in

$$X^w = \bar{q}SC_X(\alpha, \beta) \ . \tag{3.32}$$

3.7 Model of the Aerodynamic Torques

In order to change the attitude of the aircraft, torques are applied to the airframe. They are generated by control surfaces such as ailerons, elevators, and rudders. The control surface deflections are scaled such that the range of δ_a, δ_e, and δ_r are the same:

$$\delta_a, \delta_e, \delta_r \ \in \ [-1, 1] \ . \tag{3.33}$$

The total torque $\mathbf{M}$ applied to the airframe contains only aerodynamics effects and is expressed in the body-fixed frame as follows:

$$\mathbf{M}^b = \begin{bmatrix} {M_x}^b \\ {M_y}^b \\ {M_z}^b \end{bmatrix} = \begin{bmatrix} L^b \\ M^b \\ N^b \end{bmatrix} \ . \tag{3.34}$$

3.7.1 Roll Torque L^b

The generation of the roll torque is modeled by a linear function of the aileron deflection δ_a, the sideslip angle β, and the dimensionless angular rates $\tilde{p}$ and $\tilde{r}$. The dimensionless angular rates are introduced as follows:

$$\tilde{p} = \frac{bp}{2V_T} \ , \qquad \tilde{q} = \frac{\bar{c}q}{2V_T} \ , \qquad \tilde{r} = \frac{br}{2V_T} \ , \tag{3.35}$$

where the wingspan is b and the mean aerodynamic chord is $\bar{c}$.

The dimensionless roll torque is

$$C_L(\delta_a, \beta, \tilde{p}, \tilde{r}) = C_{La}\delta_a + C_{L\beta}\beta + C_{L\tilde{p}}\tilde{p} + C_{L\tilde{r}}\tilde{r} \ . \tag{3.36}$$

The roll torque is then obtained by multiplying $C_L(\delta_a, \beta, \tilde{p}, \tilde{r})$ by the dynamic pressure $\bar{q}$ and the wing surface S as follows:

$$L^b = \bar{q}SbC_L(\delta_a, \beta, \tilde{p}, \tilde{r}) \ . \tag{3.37}$$

The effectiveness of the ailerons to produce some roll torque is modeled through the coefficient C_{La} and depends mainly on the size of the ailerons. The term $C_{L\tilde{p}}$ is the damping factor related to the dimensionless roll rate $\tilde{p}$. The term $C_{L\tilde{r}}$ is useful to model the effect of unequal left and right wing speeds due to the rotation around the z_b axis, resulting in a difference in the lift forces, and thus modifying the roll torque generation.

3.7.2 Pitch Torque M^b

The generation of the pitch torque M^b expressed in the aircraft body-fixed frame (b) is modeled by a linear function of the elevator deflection δ_e, of the angle of attack α, and of the dimensionless pitch rate $\tilde{q}$. The dimensionless pitch torque is modeled as

$$C_M(\delta_e, \alpha, \tilde{q}) = C_{M1} + C_{M_e}\delta_e + C_{M\tilde{q}}\tilde{q} + C_{M\alpha}\alpha \ . \tag{3.38}$$

The effectiveness of the elevator to produce some pitch torque is accounted for through the coefficient C_{Me} and is dependent mainly on the size of the elevator. The derivative term $C_{M\alpha}$ is negative if the airplane is stable on its longitudinal axis. The damping factor $C_{M\tilde{q}}$ depends mainly on the length of the fuselage and the surface of the horizontal tail.

The pitch torque is finally computed as follows:

$$M^b = \bar{q}S\bar{c}C_M(\delta_e, \alpha, \tilde{q}) \ . \tag{3.39}$$

3.7.3 Yaw Torque N^b

The generation of the yaw torque N^b is modeled by a linear function of the rudder deflection δ_r, of the sideslip angle β, and of the dimensionless yaw rate $\tilde{r}$ as follows:

$$\begin{aligned} C_N(\delta_r, \tilde{r}, \beta) &= C_{N_{\delta_r}}\delta_r + C_{N\tilde{r}}\tilde{r} + C_{N\beta}\beta \ , \\ N^b &= \bar{q}SbC_N(\delta_r, \tilde{r}, \beta) \ . \end{aligned} \tag{3.40}$$

The effectiveness of the rudder to produce some yaw torque is modeled through the coefficient $C_{N_{\delta_r}}$ and is dependent mainly on the size of the rudder. The damping factor $C_{N\tilde{r}}$ is affected by primarily the lever arm and the size of the vertical tail.

3.8 Summary of the Nonlinear Aircraft Model

The differential equations of the nonlinear six degree-of-freedom aircraft model are recalled below:

$$\begin{bmatrix} \dot{x}_N \\ \dot{x}_E \\ \dot{x}_D \end{bmatrix} = \mathbf{C}_b^n \begin{bmatrix} u \\ v \\ w \end{bmatrix} , \tag{3.41}$$

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \begin{bmatrix} -g \sin\theta \\ g \sin\phi \cos\theta \\ g \cos\phi \cos\theta \end{bmatrix} + \frac{1}{m} \left[\begin{pmatrix} F_T \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} X^b \\ Y^b \\ Z^b \end{pmatrix} \right] - \begin{bmatrix} qw - rv \\ ru - pw \\ pv - qu \end{bmatrix} , \tag{3.42}$$

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} -q_1 & -q_2 & -q_3 \\ q_0 & -q_3 & q_2 \\ q_3 & q_0 & -q_1 \\ -q_2 & q_1 & q_0 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} , \tag{3.43}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = \left(\mathbf{I}^b\right)^{-1} \left(\begin{bmatrix} L \\ M \\ N \end{bmatrix}^b - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I}^b \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) , \tag{3.44}$$

$$\dot{n} = -\frac{1}{\tau_n} n + \frac{1}{\tau_n} n_c \, . \tag{3.45}$$

Expressions for the thrust force F_T and torques $[L \quad M \quad N]^T$ are given in previous sections of this chapter. The numerical values of all the parameters involved in the model are given in Appendix F.

References

1. B. Stevens and F. Lewis. *Aircraft Control and Simulation, Second Edition*. Wiley, New York, NY, 2003.
2. R. F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.
3. M. Möckli. *Guidance and Control for Aerobatic Maneuvers of an Unmanned Airplane*. PhD thesis, ETH Zurich, 2006. Diss No. 16586.
4. G. J. J. Ducard. *Fault-tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle*. PhD thesis, ETH Zürich, 2007. Diss. No. 17505.
5. C. Eck. *Navigation Algorithms with Applications to Unmanned Helicopters*. PhD thesis, ETH Zürich, 2001. Diss. No. 14402.
6. R. Möller. Modelling an Acrobatic Aircraft. Term paper, ETH Zürich, 2003.

Chapter 4
Nonlinear Fault Detection and Isolation System

In this chapter, three main limitations of the classical implementation of the MMAE method to isolate faults based on predefined fault hypotheses are highlighted. The first limitation concerns the number of filters that must be designed in order to span the range of possible fault scenarios, which must be limited due to computational load. The second limitation appears when an actuator is locked at an arbitrary non-zero position that biases the residuals of the KFs, leading to inaccurate fault detection and state estimation. Third, most of the implementations of an MMAE method only work efficiently around predefined operating conditions. This chapter presents a nonlinear actuator FDI system, which works over the entire operating envelope of an aircraft. Locked-in-place and floating actuator faults can be handled. The robustness of the FDI system is enhanced by the use of auxiliary excitation signals. The FDI system is also capable of handling two simultaneous actuator failures with no increase of the computational load. The complete system has been demonstrated in simulation with a nonlinear model of a model aircraft in moderate to severe wind conditions.

4.1 Introduction

New generations of UAVs will be designed to achieve their mission not only with increased efficiency, but also with more safety and security. Future UAVs will be operated with algorithms capable of monitoring the aircraft health and of taking action if needed. Fault-tolerant control systems for small and low-cost UAVs should not increase significantly the number of actuators or sensors to achieve the safer operation. This chapter describes a computationally efficient on-line nonlinear FDI system that monitors the actuators' health without requiring any sensors to measure the deflection of the control surfaces.

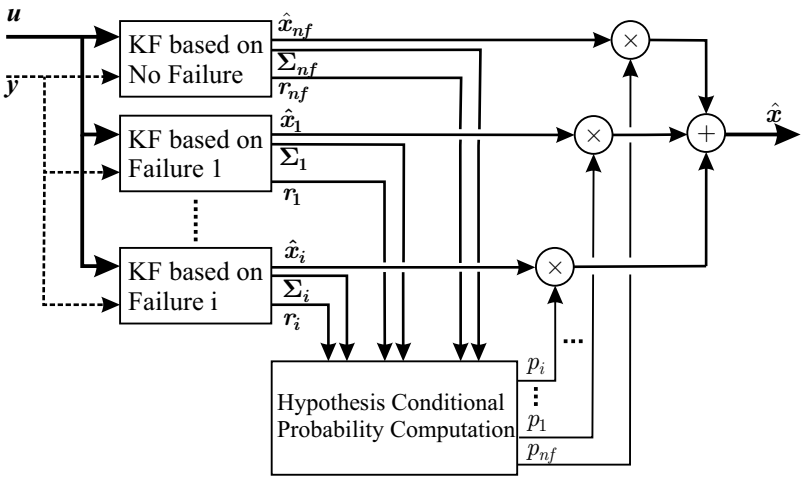


Fig. 4.1 Classical MMAE scheme

4.2 FDI Using MMAE Schemes

One approach to detect and isolate actuator or sensor faults is the MMAE method [1] as depicted in Fig. 4.1. It is based on a bank of KFs running in parallel, each of which is matching a particular fault status of the system. A hypothesis testing algorithm uses the residuals from each KF to assign a conditional probability to each fault hypothesis. As one may expect, the computational load is quite intense. Therefore, the on-line use of this method was impractical for a long time. However, with the more powerful processors now available this method has regained appeal in many applications.

Several papers have demonstrated how the MMAE method can be used in the context of FDI systems for aircraft [2–4] and underwater vehicles [5].

4.2.1 Advantage of the MMAE Method

The main advantage of the MMAE method lies in its responsiveness to parameter variations, leading to faster fault isolation than that attained by other methods without a multiple model structure.

The method also enables the reconstruction of a correct state estimate even when an actuator or sensor fault occurs, since the estimated state vector is the sum of each KF estimate weighted by its corresponding probability.

4.2.2 Limitations of the MMAE Method

The MMAE method can be applied in practice as long as the expected faults can be hypothesized by a reasonable number of KFs. However, the number of addressable faults is limited due to the computational load required for each filter.

The method also does not provide satisfactory results when an occurring fault does not closely match a predefined fault hypothesis. This may occur when an actuator is locked-in-place at an arbitrary position that affects the dynamics of the system. Since lock-in-place faults cannot be predicted, they can have detrimental effects on the filter performance. Due to the biased residuals, the KF provides inaccurate estimates of the state variables, which causes severe problems with the probability calculation. Therefore, neither the fault detection nor the fault isolation works properly, and the state estimation is useless for control purposes.

Moreover, in most of the fault-tolerant applications that use the MMAE method, the KFs are designed based on a linear model of the unmanned vehicle operated at certain nominal conditions. Very few papers describe the MMAE method being used in the nonlinear case, when the system operates over the entire range of possible operating conditions. The authors of [6] used a multiple model approach for the sensor fault detection of nonlinear systems. However, the assumption is made that the nonlinear system can be approximated by a finite number of interpolated linear time-invariant models, which constitute the banks of KFs whose residuals are used to determine the effective operating regime and isolate the faulty sensor. In this chapter, a unique bank of filters is designed, which can operate over the whole span of the aircraft's flying conditions.

4.2.3 New Extensions to the MMAE Method: The EMMAE Method

In order to make the MMAE method applicable for any flight conditions and capable of isolating lock-in-place or floating actuator faults, the MMAE algorithm is combined with EKFs used for the nonlinear estimation of some (unknown) fault parameter: the deflection of a faulty control surface (or actuator). The resulting method is called EMMAE; see [7–9].

This chapter explains why the on-line estimation of the deflection of a faulty actuator enables the EMMAE method to cope with lock-in-place or floating actuator fault scenarios and drastically reduces the number of filters needed.

Moreover, this method takes advantage of the estimated faulty-actuator deflection to reconfigure the settings of the control allocator described in Chap. 5 in order to efficiently compensate for the fault.

Furthermore, some techniques were added to enhance the robustness and the performance of the EMMAE-FDI system when there is very low excitation of the aircraft during steady flight and to improve the speed and accuracy of the fault isolation.

Finally, the results of simulations are presented and demonstrate the complete system on a nonlinear model of an aircraft experiencing consecutive actuator faults under severe wind disturbance.

4.3 A New FDI Scheme Based on the EMMAE Method

4.3.1 Modeling Actuator Faults

A lock-in-place or floating actuator fault in the system can be seen as if the desired control input δ_j was disconnected and replaced by a faulty control signal $\bar{\delta}_j$ that takes control over the plant, as shown in Fig. 4.2. In a concise manner [10], the true input of the plant can be written as

$$u_i(t) = \delta_i(t) + \sigma_{A_i}(\bar{\delta}_i(t) - \delta_i(t)) \; . \tag{4.1}$$

In the case of actuator failure(s), the vector of the (unknown) inputs is

$$\bar{\boldsymbol{\delta}}(t) = [\bar{\delta}_1(t) \; \bar{\delta}_2(t) \quad ... \quad \bar{\delta}_m(t)]^T \; , \tag{4.2}$$

with $\boldsymbol{\sigma} = \text{diag}\{\sigma_{A1} \quad \sigma_{A2} \quad ... \quad \sigma_{Am}\}$, where

$$\sigma_{Aj} = \left\{ \begin{array}{ll} 1, & \text{if the } j^{\text{th}} \text{ actuator fails} \\ 0, & \text{otherwise} \end{array} \right\} \; . \tag{4.3}$$

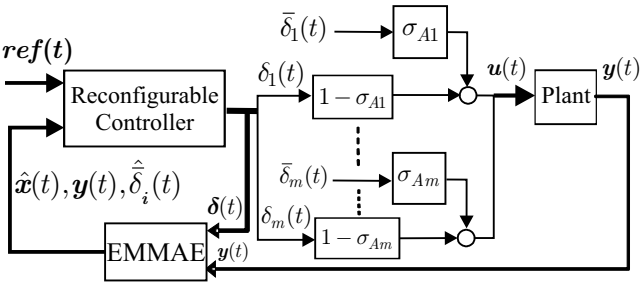


Fig. 4.2 Modeling of actuator faults

In the method presented below, the unknown parameters $\bar{\delta}_j$ are constantly estimated by their respective EKF. The conditional fault-hypothesis probabilities p_j assign the value for σ_{Aj}.

4.3.2 The EMMAE Method

The MMAE method is to be made applicable for any arbitrary lock-in-place faults or uncontrolled varying faults and at all flying conditions. Therefore, the original MMAE algorithm is modified by replacing the linear KFs by EKFs used as nonlinear estimators of the state vector and a fault parameter, namely the deflection of a faulty control surface (or actuator). The implementation of the EMMAE is depicted in Fig. 4.3. Contrary to the FDI designs with the classical MMAE method where several KF are designed for several faulty deflections for one actuator, in the EMMAE method only one EKF is responsible for completely monitoring one actuator's health. Therefore, the EMMAE method drastically reduces the number of filters required for actuator health monitoring.

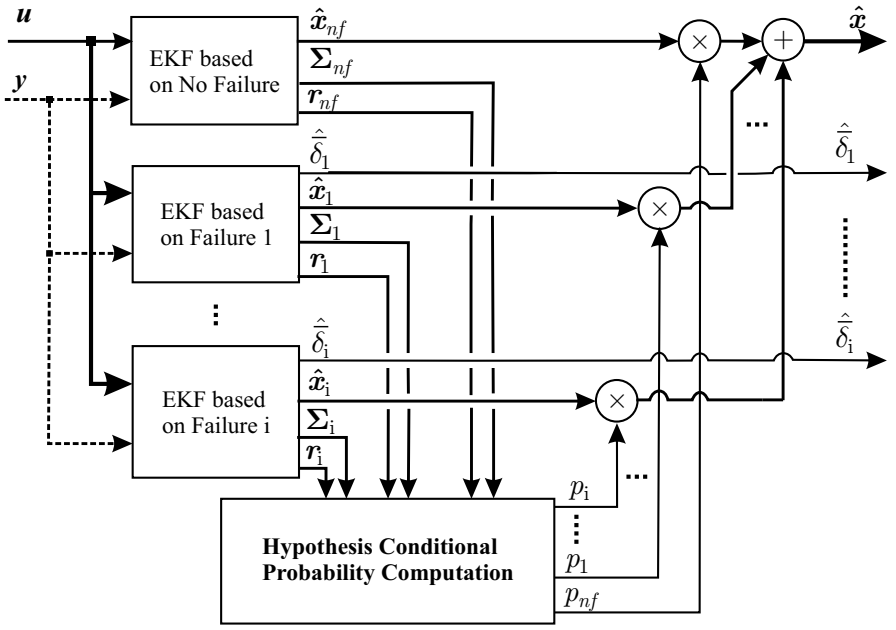


Fig. 4.3 EMMAE-FDI scheme: each EKF monitors its assigned actuator

The addition of the actuator deflection estimate in the system state vector enables the EMMAE method to work for all the possible positions where an

actuator can be locked or floating. In order to better illustrate the difference between the EMMAE and MMAE methods, we recall how the models are defined in the regular MMAE method [2–5] for an actuator or a sensor failure. The MMAE scheme considers a bank of linear models of the form

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B}\mathbf{u} \,, \\ \mathbf{y} &= \mathbf{C}\mathbf{x} + \mathbf{D}\mathbf{u} \,, \end{aligned} \tag{4.4}$$

where each model matches a fault scenario.

For example, the model that describes a failure of the j^{th} actuator will have its $\mathbf{B}$ matrix modified such that the j^{th} column of the $\mathbf{B}$ matrix is replaced by the very same column times a factor λ_j that varies from zero (complete loss of the actuator) to one (fully functioning actuator), see (4.5). Any intermediate value of λ_j indicates a reduction in the effectiveness of the j^{th} actuator to modify the dynamics of the aircraft as shown in (4.5):

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \mathbf{B_j}\mathbf{u} \,, \\ \dot{\mathbf{x}} &= \mathbf{A}\mathbf{x} + \begin{bmatrix} b_{11} & \cdots & b_{1j}\lambda_j & \cdots & b_{1N} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ b_{l1} & \cdots & b_{lj}\lambda_j & \cdots & b_{lN} \\ \vdots & \cdots & \vdots & \cdots & \vdots \\ b_{p1} & \cdots & b_{pj}\lambda_j & \cdots & b_{pN} \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_j \\ \vdots \\ u_N \end{bmatrix} . \end{aligned} \tag{4.5}$$

If we are to design a filter to detect the failure of the i^{th} sensor, the i^{th} row of the $\mathbf{C}$ matrix will be replaced by the very same row times a factor that varies from zero (total sensor failure such as sensor disconnection) to one (no sensor failure) as shown in (4.6):

$$\begin{aligned} \mathbf{y} &= \mathbf{C_i}\mathbf{x} \,, \\ \mathbf{y} &= \begin{bmatrix} c_{11} & c_{12} & \cdots & c_{1k} & \cdots & c_{1p} \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ c_{i1}\lambda_i & c_{i2}\lambda_i & \cdots & c_{ik}\lambda_i & \cdots & c_{ip}\lambda_i \\ \vdots & \vdots & \cdots & \vdots & \cdots & \vdots \\ c_{m1} & c_{m2} & \cdots & c_{mk} & \cdots & c_{mp} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_k \\ \vdots \\ x_p \end{bmatrix} . \end{aligned} \tag{4.6}$$

However, this kind of approach for the modeling of actuator and sensor failure is very restrictive. Indeed, in the case of a total loss of the j^{th} actuator, the factor λ_j equals zero. This means that whatever control input the controller generates for the j^{th} actuator, it has no influence on the dynamics of the aircraft, and the faulty actuator deflection is considered to be zero. Note that if the j^{th} actuator is actually locked at a non-zero deflection angle, the control signal to the j^{th} actuator has no influence on the dynamics of

the aircraft; however, the faulty-actuator deflection does have an influence on the dynamics of the aircraft. This condition results in an unknown bias term that will prevent the j^{th} KF in the MMAE method from working properly. Therefore, the residuals will be biased, and the state estimation and the computation of the probabilities will be incorrect as well.

In the EMMAE method, we not only modify the control input matrix, but also the dynamics matrix. Indeed, in order to define a model that describes a failure of the j^{th} actuator, the j^{th} column of the control input matrix is zeroed and the state vector is augmented with the j^{th} actuator deflection $\bar{\delta}_j$. The dynamics matrix is also augmented with the original j^{th} column of the control input matrix. In this way, the control inputs from the controller to the j^{th} actuator are totally ignored, but the faulty deflection $\bar{\delta}_j$ that is constantly estimated ($\hat{\bar{\delta}}_j$) in the state vector contributes to modifying the dynamics of the aircraft model of the j^{th} filter, yielding residuals that are the smallest for the filter matching the occurring fault. Let us illustrate how the filters in the EMMAE are constructed in practice.

4.4 Aircraft Actuator Configuration and Nonlinear Dynamics

4.4.1 The Aircraft Configuration

The five control surfaces of the aircraft under consideration are one left aileron, one right aileron, one left elevator, one right elevator, and one rudder, as shown in Fig. 4.4. All actuators are fully independent, which means that ailerons (or elevators) can individually move up, down, or together in the same direction. This configuration permits some pitch torque to be produced with ailerons or some roll torque to be produced with elevators.

The state vector of the FDI filters is chosen to take only the most relevant state variables of the aircraft in order to reduce the computational load when running the EKFs. The state vector is $\mathbf{x} = [p \quad q \quad r \quad \alpha \quad \beta]^T$. The control vector for the aircraft is $\mathbf{u} = [\delta_{a_1} \quad \delta_{a_2} \quad \delta_{e_1} \quad \delta_{e_2} \quad \delta_r \quad F_T]^T$, and the vector involving only actuator deflections is $\boldsymbol{\delta} = [\delta_{a_1} \quad \delta_{a_2} \quad \delta_{e_1} \quad \delta_{e_2} \quad \delta_r]$.

4.4.2 Aircraft Nonlinear Dynamics

Among the nonlinear equations which describe the dynamics of the aircraft, those involving the turn rates are of high interest. They show the explicit relationship between turn rates and the torques applied to the aircraft, *i.e.*, $[L\,M\,N]^T$, expressed in the body-axes frame (x_b, y_b, z_b) of the aircraft:

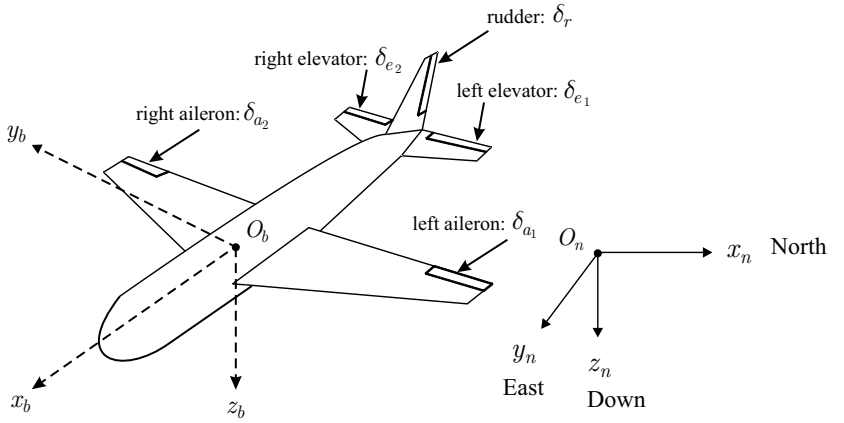


Fig. 4.4 Aircraft configuration

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = (\mathbf{I}^b)^{-1} \left(\begin{bmatrix} L \\ M \\ N \end{bmatrix}^b - \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \mathbf{I}^b \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right) . \tag{4.7}$$

In the context of this work, the aircraft is a small UAV for which the aerodynamic moments have been modeled as follows (see [11–14]):

$$\begin{aligned} L &= \bar{q}SbC_L(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, p, r, \beta) , \\ M &= \bar{q}S\bar{c}C_M(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \alpha, q) , \\ N &= \bar{q}SbC_N(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \delta_r, r, \beta) , \end{aligned} \tag{4.8}$$

where the dynamic pressure is $\bar{q} = \frac{\rho V_T^2}{2}$, the total airspeed of the aircraft is V_T, the air density is ρ, the wing total surface is S, the wing span is b, and the mean aerodynamic wing chord is $\bar{c}$.

The aerodynamic derivatives are expressed as a linear combination of the state elements and control inputs as

$$\begin{aligned} C_L &= C_{L_{a1}}\delta_{a1} + C_{L_{a2}}\delta_{a2} + C_{L_{e1}}\delta_{e1} + C_{L_{e2}}\delta_{e2} + C_{L\tilde{p}}\tilde{p} + C_{L\tilde{r}}\tilde{r} + C_{L\beta}\beta , \\ C_M &= C_{M1} + C_{M_{a1}}\delta_{a1} + C_{M_{a2}}\delta_{a2} + C_{M_{e1}}\delta_{e1} + C_{M_{e2}}\delta_{e2} + C_{M\tilde{q}}\tilde{q} + C_{M\alpha}\alpha , \\ C_N &= C_{N_{\delta r}}\delta_r + C_{N\tilde{r}}\tilde{r} + C_{N\beta}\beta , \end{aligned} \tag{4.9}$$

with

$$\tilde{p} = \frac{bp}{2V_T} , \qquad \tilde{q} = \frac{\bar{c}q}{2V_T} , \qquad \tilde{r} = \frac{br}{2V_T} . \tag{4.10}$$

The last two nonlinear differential equations concern the angle of attack α and the sideslip angle β as follows (see Appendices A and C for the derivation

of the following two formulae):

$$\dot{\alpha} \approx q + \frac{g}{V_T}\left\{1 + \frac{\bar{q}S}{mg}([C_{X1} + C_{Z\alpha}]\alpha + C_{Z1})\right\} ,$$
$$\dot{\beta} \approx -r + \frac{\bar{q}SC_{Y1}}{mV_T}\beta , \tag{4.11}$$

with the drag derivative C_{X1}, the side force derivative C_{Y1}, and the lift derivatives C_{Z1}, $C_{Z\alpha}$ being constant terms (see Appendix F). The inertia matrix is $\mathbf{I}^b = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix}$, and the measurement vector is $\mathbf{y} = [p \quad q \quad r \quad \alpha \quad \beta]^T$.

4.5 Design of the EKFs

In 1960, Kalman introduced his approach to linear filtering based on the method of minimum variance [15]. Compared with existing filtering techniques at that time, the KF, though usually more computationally intense, offered performance improvements and ease of implementation on a digital computer due to its recursive formulation. Moreover, the KF processes all available measurement data or information that can be provided to it, regardless of their precision, on the basis of their stochastic descriptions, in order to generate an overall best estimate of the parameter considered [16].

A mathematical model has been developed to describe the behavior of the aircraft. We know that this model will never be perfect, leaving many effects unmodeled. Also, this model will only be approximated by the computer implementation. Furthermore, several parameters of the model will not be known exactly, and the sensor measurement data will be corrupted by noise and biases. For all these reasons, we decided to use Kalman filtering to take into account such system dynamics and measurement noise, errors, and uncertainties.

KFs are well suited when the real world can be described by linear differential equations expressed in state space form and when the measurements are linear functions of the states. However, in most realistic problems, the real world is described by nonlinear differential equations. In this work, we will therefore consider the implementation of EKFs to take into account these nonlinearities.

4.5.1 EKF Equations

The EKFs are designed based on a set of continuous nonlinear differential equations that describe the plant under consideration as follows (see [17]):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) + \mathbf{w} \ , \tag{4.12}$$

where the state vector is $\mathbf{x}$, the control input vector is $\mathbf{u}$, the set of nonlinear functions of the state and control vectors is $\mathbf{f}(\mathbf{x}, \mathbf{u})$, and the random zero-mean process noise vector is $\mathbf{w}$.

The continuous process noise covariance matrix describing the random process $\mathbf{w}$ is given by

$$\mathbf{R}_w = E\{\mathbf{w}\mathbf{w}^T\} \ . \tag{4.13}$$

Equation 4.12 is first linearized around the current operating point and then discretized using the Euler integration method. Note that the Euler integration method is also used for the simulations presented in the following sections. The discretized form of (4.12) is expressed in state space form as

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k \mathbf{x}_k + \mathbf{G}_k \mathbf{u}_k + \mathbf{w}_k \ , \tag{4.14}$$

where the state vector is evaluated at the discrete time instant $t_k = kT_s$, with T_s being the sampling period of the system. The control input vector at time step k is $\mathbf{u}_k$, and the discrete random zero-mean process noise $\mathbf{w}_k$ is used to describe uncertainties in our model.

Finally, the discrete form of the measurement equation, either a linear or a nonlinear function of the states, is

$$\mathbf{y}_k = \mathbf{h}(\mathbf{x}_k) + \mathbf{v}_k \ , \tag{4.15}$$

where the discrete zero-mean random noise $\mathbf{v}_k$ is described by the measurement noise covariance matrix $\mathbf{R}_{v,k} = E\{\mathbf{v}_k \mathbf{v}_k^T\}$ and consists of the variances of each of the measurement noise sources.

The discrete transition matrix $\boldsymbol{\phi}_k$ is approximated by (see Appendix B.2.2)

$$\boldsymbol{\phi}_k \approx \mathbf{I} + \mathbf{F}(k) T_s \ , \tag{4.16}$$

where the continuous system dynamics matrix $\mathbf{F}(k)$ is obtained by linearizing the continuous nonlinear equations and is evaluated at the latest available state estimate $\hat{\mathbf{x}}_{k|k}$ according to

$$\mathbf{F}(k) = \left. \frac{\partial \mathbf{f}(\mathbf{x}, \mathbf{u})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}, \, \mathbf{u}=\mathbf{u}_k} \ . \tag{4.17}$$

Similarly, the continuous measurement matrix $\mathbf{H}(k)$ is computed by linearizing the (possibly nonlinear) measurement equation $\mathbf{h}(\mathbf{x})$ and is successively

evaluated at the latest available state estimate $\hat{\mathbf{x}}_{k|k-1}$ according to

$$\mathbf{H}(k) = \mathbf{H}_k = \left.\frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}}\right|_{\mathbf{x}=\hat{\mathbf{x}}_{k|k}} . \tag{4.18}$$

The equations used in the EKF [17, 18] are described below. The schematic overview of the computation steps is shown in Fig. 4.5:

1. The Kalman gain matrix $\mathbf{L}_k$ is computed as

$$\mathbf{L}_k = \mathbf{\Sigma}_{k|k-1}\mathbf{H}_k^T \left[\mathbf{H}_k \mathbf{\Sigma}_{k|k-1}\mathbf{H}_k^T + \mathbf{R}_{v,k}\right]^{-1} \tag{4.19}$$

and is a function of the last propagated state error covariance matrix $\mathbf{\Sigma}_{k|k-1}$, and of the measurement noise covariance matrix $\mathbf{R}_{v,k}$.

2. The measurement update of the state estimate is as follows:

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{L}_k[\mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_{k|k-1})] , \tag{4.20}$$

where the last extrapolated state estimate is $\hat{\mathbf{x}}_{k|k-1}$, the Kalman gain is $\mathbf{L}_k$, the measurement vector is $\mathbf{y}_k$ and the estimated measurement vector is $\mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$. The set of continuous nonlinear measurement equations is $\mathbf{h}(.)$.

3. The third step concerns the update of the state error covariance matrix $\mathbf{\Sigma}_{k|k} = \mathbf{E}\{\mathbf{e}_{k|k}\mathbf{e}_{k|k}^T\}$ with $\mathbf{e}_{k|k} = \mathbf{x}(k) - \hat{\mathbf{x}}_{k|k}$, where $\mathbf{x}(k)$ is the true (unknown) value of the state vector at the discrete instant k. The matrix $\mathbf{\Sigma}_{k|k}$ is recursively computed as a function of the last predicted state error covariance matrix $\mathbf{\Sigma}_{k|k-1}$ and the last computed Kalman gain matrix $\mathbf{L}_k$ as follows:

$$\mathbf{\Sigma}_{k|k} = [\mathbf{I} - \mathbf{L}_k\mathbf{H}_k]\,\mathbf{\Sigma}_{k|k-1} . \tag{4.21}$$

4. The forward propagation of the state error covariance matrix is

$$\mathbf{\Sigma}_{k+1|k} = \boldsymbol{\phi}_k \mathbf{\Sigma}_{k|k} \boldsymbol{\phi}_k^T + \mathbf{R}_{w,k} , \tag{4.22}$$

where the matrix $\mathbf{R}_{w,k}$ represents the covariance of the discrete process noise acting on the elements of the state vector. The value of $\mathbf{R}_{w,k}$ is found from the continuous process noise covariance matrix $\mathbf{R}_w$ and the continuous transition matrix $\boldsymbol{\phi}$ according to (see also Appendix B.1)

$$\mathbf{R}_{w,k} = \int_0^{T_s} \boldsymbol{\phi}(\tau)\mathbf{R}_w\boldsymbol{\phi}^T(\tau)d\tau . \tag{4.23}$$

There is an alternative solution of introducing process noise into the system through the control input matrix. Indeed, if the plant is described by the following equation:

$$\mathbf{x}_{k+1} = \boldsymbol{\phi}_k\mathbf{x}_k + \mathbf{G}_k\mathbf{u}_k + \mathbf{G}_k\mathbf{w}_k , \tag{4.24}$$

then the process noise covariance matrix is defined by

$$\mathbf{R}_{w,k} = \mathbf{G}_k \mathbf{R}_w \mathbf{G}_k^T \ , \tag{4.25}$$

where the covariance matrix of the discrete process noise acting on the elements of the state vector is $\mathbf{R}_{w,k}$, and the discrete control input matrix is $\mathbf{G}_k = T_s \frac{\partial \mathbf{f}}{\partial \mathbf{u}}\Big|_{\mathbf{V_T}(k)}$.

5. The propagation forward of the state estimate does not have to be done with the discrete transition matrix $\boldsymbol{\phi}_k$ but rather it is done directly by integrating the actual nonlinear differential equations forward at each sampling interval. If the Euler integration technique is used, the extrapolated state estimate is computed with

$$\hat{\mathbf{x}}_{k+1|k} = \hat{\mathbf{x}}_{k|k} + \dot{\hat{\mathbf{x}}}_{k|k} T_s \ , \tag{4.26}$$

where the derivative is obtained from

$$\dot{\hat{\mathbf{x}}}_{k|k} = \mathbf{f}\left(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k\right) \ . \tag{4.27}$$

Remarks:

- Note that the EKF presented in this section keeps track of the total estimates and not of the incremental ones (deviation from a nominal trajectory) as would be the case in a linearized KF. Indeed, the residuals are built from the difference between the true measurement vector and the predicted measurement vector using the set of nonlinear measurement equations acting on the total state estimate $\mathbf{h}(\hat{\mathbf{x}}_{k|k-1})$.
- Moreover, the measurement update of the state estimate is done using the total estimate $\hat{\mathbf{x}}(k|k-1)$, and the propagation forward of the total state estimate is achieved using the set of nonlinear differential equations acting on the total state estimate $\mathbf{f}(\hat{\mathbf{x}}_{k|k}, \mathbf{u}_k)$ instead of using the discrete transition matrix acting on incremental state quantities.
- Nevertheless, the computation of the Kalman gains and state error covariance matrices makes use of the recursively updated linear model through $\boldsymbol{\phi}_k$, $\mathbf{H}_k$, and $\mathbf{R}_{w,k}$.
- The schematic overview shown in Fig. 4.5 clarifies the mechanization of the EKF implemented in this work.

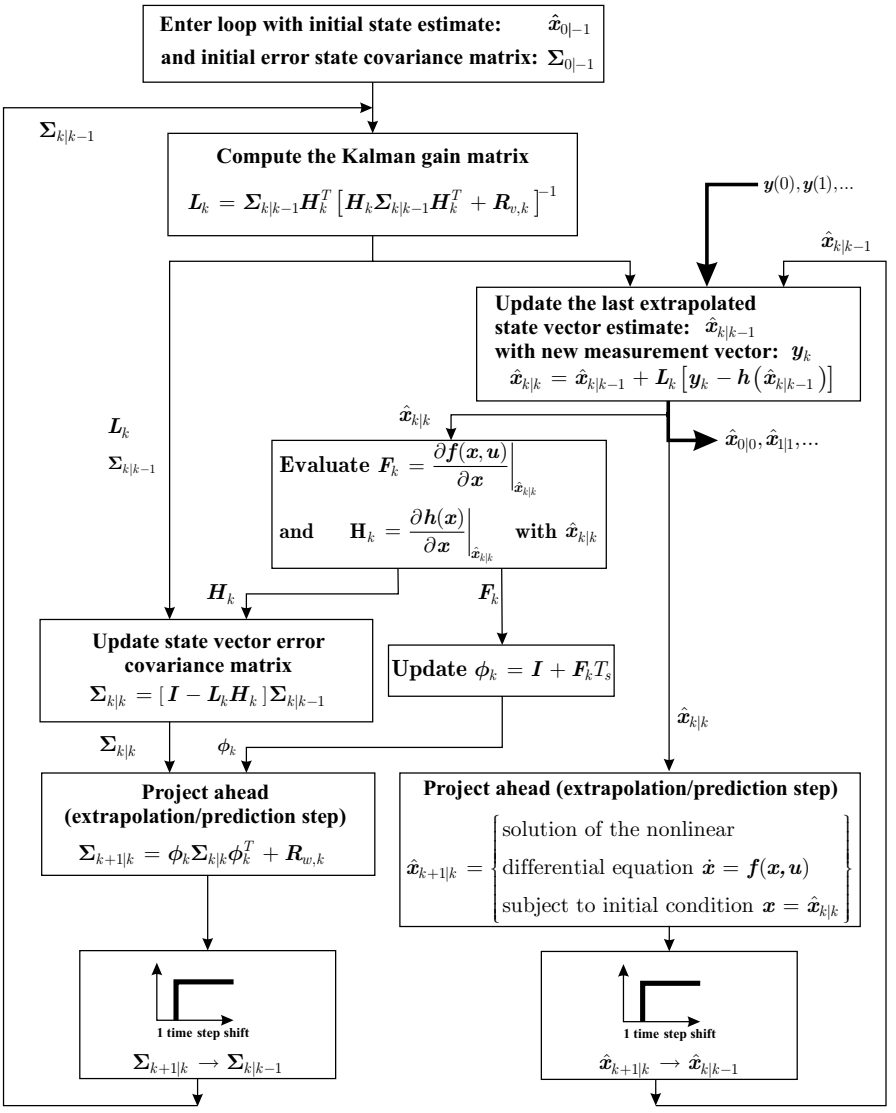


Fig. 4.5 EKF mechanization

4.5.2 Designing the EKF for the No-fault Scenario

The continuous system dynamics matrix for the no-fault filter $\mathbf{F}_{nf}(k)$ evaluated at time step k can be explicitly derived from the nonlinear model as

$$\mathbf{F}_{nf}(k) = \begin{bmatrix} & & & 0 & \frac{Sb[I_{zz}C_{L\beta} - I_{xz}C_{N\beta}]}{D_1}\bar{q} \\ & \mathbf{F_1} & & \frac{S\bar{c}C_{M\alpha}}{I_{yy}}\bar{q} & 0 \\ & & & 0 & \frac{Sb[I_{xx}C_{N\beta} - IxzC_{L\beta}]}{D_1}\bar{q} \\ 0 & 1 & 0 & \frac{\rho V_T S C_{Z\alpha}}{2m} & 0 \\ 0 & 0 & -1 & 0 & \frac{\rho V_T S C_{Y1}}{2m} \end{bmatrix}_{\hat{\mathbf{x}}_{nf}(k|k)}, \quad (4.28)$$

with the submatrix $\mathbf{F_1}$ defined as

$$\mathbf{F_1} = \begin{bmatrix} \frac{I_{zz}Sb^2C_{L\tilde{p}}}{2D_1V_T}\bar{q} - \frac{N_1}{D_1}q & \frac{-N_1}{D_1}p + \frac{N_2}{D_1}r & \frac{(I_{zz}C_{L\tilde{r}} - I_{xz}C_{N\tilde{r}})Sb^2}{2D_1V_T}\bar{q} + \frac{N_2}{D_1}q \\ \frac{I_{xx}-I_{zz}}{I_{yy}}\, r - 2\frac{Izx}{I_{yy}}p & \frac{S\bar{c}^2C_{M\tilde{q}}}{2V_TI_{yy}}\bar{q} & -\frac{I_{xx}-I_{zz}}{I_{yy}}p - \frac{2I_{xz}}{I_{yy}}r \\ -\frac{Sb^2C_{L\tilde{p}}I_{xz}}{2D_1V_T}\bar{q} + \frac{N_3}{D_1}q & \frac{N_3}{D_1}p + \frac{N_1}{D_1}r & \frac{Sb^2[-I_{xz}C_{L\tilde{r}} + I_{xx}C_{N\tilde{r}}]}{2D_1V_T}\bar{q} + \frac{N_1}{D_1}q \end{bmatrix}, \quad (4.29)$$

where $N_1 = I_{xz}(I_{xx} - I_{yy} + I_{zz})$, $N_2 = I_{yy}I_{zz} - I_{xz}^2 - I_{zz}^2$, $N_3 = I_{xz}^2 - I_{xx}I_{yy} + I_{xx}^2$, and $D_1 = I_{xx}I_{zz} - I_{xz}^2$.

The discrete transition matrix for the no-fault filter is calculated with $\boldsymbol{\phi}_{nf,k} = \mathbf{I} + \mathbf{F}_{nf}(k)T_s$.

We also compute the control input matrix of the no-fault filter $\mathbf{G}_{nf}(k)$ as follows:

$$\mathbf{G}_{nf}(k) = \bar{q}_k \begin{bmatrix} \frac{SbI_{zz}C_{La_1}}{D_1} & \frac{SbI_{zz}C_{La_2}}{D_1} & \frac{SbI_{zz}C_{Le_1}}{D_1} & \frac{SbI_{zz}C_{Le_2}}{D_1} & \frac{-SbI_{xz}C_{N_{\delta r}}}{D_1} \\ \frac{S\bar{c}C_{Ma_1}}{I_{yy}} & \frac{S\bar{c}C_{Ma_2}}{I_{yy}} & \frac{S\bar{c}C_{Me_1}}{I_{yy}} & \frac{S\bar{c}C_{Me_2}}{I_{yy}} & 0 \\ \frac{-SbI_{xz}C_{La_1}}{D_1} & \frac{-SbI_{xz}C_{La_2}}{D_1} & \frac{-SbI_{xz}C_{Le_1}}{D_1} & \frac{-SbI_{xz}C_{Le_2}}{D_1} & \frac{SbI_{xx}C_{N_{\delta r}}}{D_1} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (4.30)$$

The discrete control input matrix for the no-fault filter is $\mathbf{G}_{nf,k} = \mathbf{G}_{nf}(k)T_s$.

4.5.3 Augmenting the State Vector with the Faulty Actuator Parameter $\bar{\delta}_i$

The state vector of the i^{th} filter is augmented to monitor the occurrence of the i^{th} actuator fault. The deflection of the failed actuator is included in the state vector in a way to be estimated by the EKF. Therefore the state vector for each filter i is

$$\mathbf{z}_i = \begin{bmatrix} \mathbf{x} \\ \bar{\delta}_i \end{bmatrix} . \tag{4.31}$$

The augmented state vector leads to the following state space equations for each filter i:

$$\begin{aligned} \mathbf{z}_i(k+1) &= \mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i(k), \boldsymbol{\delta}(k)) + \mathbf{w}_k \ , \\ \mathbf{y}_i(k) &= \mathbf{h}(\mathbf{z}_i(k)) + \mathbf{v}_k \ , \end{aligned} \tag{4.32}$$

where

$$\mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i(k), \boldsymbol{\delta}(k)) = \begin{bmatrix} \mathbf{f}(\mathbf{z}_i(k), \boldsymbol{\delta}(k)) \\ \bar{\delta}_i(k) \end{bmatrix} . \tag{4.33}$$

The linearization of the dynamics matrix yields

$$\begin{aligned} \mathbf{F}_{z_i}(k) &= \frac{\partial}{\partial \mathbf{z}_i} \mathbf{f}_{\mathbf{z}_i}(\mathbf{z}_i(k), \boldsymbol{\delta}(k)) \bigg|_{\mathbf{z}_i = \hat{\mathbf{z}}_i(k|k)} \ , \\ &= \begin{bmatrix} \mathbf{F}(k) & \mathbf{G}^{(i)}(k) \\ 0 & 1 \end{bmatrix} , \end{aligned} \tag{4.34}$$

where $\mathbf{G}^{(i)}$ represents the i^{th} column of $\mathbf{G}$. The input matrix becomes

$$\mathbf{G}_{z_i}(k) = \frac{\partial}{\partial \bar{\delta}_i} \mathbf{f}_{z_i}(\mathbf{z}_i(k), \boldsymbol{\delta}(k)) \bigg|_{\mathbf{z}_i = \hat{\mathbf{z}}_i(k|k)} = \begin{bmatrix} \mathbf{G^{(0,i)}}(k) \\ 0 \end{bmatrix} , \tag{4.35}$$

with $\mathbf{G^{(0,i)}}$ representing the matrix $\mathbf{G}$ with its i^{th} column set to zero.

The linearization of the measurement matrix is

$$\mathbf{H}_{z_i}(k) = \frac{\partial}{\partial \mathbf{z}_i} \mathbf{h}(\mathbf{z}_i(k), \boldsymbol{\delta}(k)) \bigg|_{\mathbf{z}_i = \hat{\mathbf{z}}_i(k|k)} = [\mathbf{C}_x(k) \quad \mathbf{C}_{\bar{\delta}_i}(k)] , \tag{4.36}$$

with

$$\begin{aligned} \mathbf{C}_x(k) &= \frac{\partial}{\partial \mathbf{x}} \mathbf{h}(\mathbf{z}_i(k)) \bigg|_{\mathbf{z}_i = \hat{\mathbf{z}}_i(k|k)} = \mathbf{H} \ , \\ \mathbf{C}_{\bar{\delta}_i}(k) &= \frac{\partial}{\partial \bar{\delta}_i} \mathbf{h}(\mathbf{z}_i(k)) \bigg|_{\mathbf{z}_i = \hat{\mathbf{z}}_i(k|k)} = 0 \ . \end{aligned} \tag{4.37}$$

Using the above equations, the linearized system evaluated at each sampling time can be written as

$$\begin{bmatrix} \mathbf{x}(k+1) \\ \bar{\delta}_i(k+1) \end{bmatrix} = \begin{bmatrix} \mathbf{F}(k) & \mathbf{G}^i(k) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x(k) \\ \bar{\delta}_i(k) \end{bmatrix} + \begin{bmatrix} \mathbf{G}^{(0,i)}(k) \\ 0 \end{bmatrix} \boldsymbol{\delta}(k) \,,$$
$$\mathbf{y}(k) = [\mathbf{H} \quad 0] \begin{bmatrix} \mathbf{x}(k) \\ \bar{\delta}_i(k) \end{bmatrix} \,. \tag{4.38}$$

4.5.4 Designing the EKF for the Case of a Failure on Aileron 1

In order to provide an example of how to derive the matrices for the EKF corresponding to the scenario of a lock-in-place or floating actuator failure, we will consider the filter that monitors the functioning of aileron 1.

The system dynamics matrix for the aileron1-fault filter $\mathbf{F}_{\delta a1}(k)$ is explicitly derived from the nonlinear model as

$$\mathbf{F}_{\delta a1}(k) = \begin{bmatrix} & \mathbf{F_1} & & 0 & \frac{Sb[I_{zz}C_{L\beta} - I_{xz}C_{N\beta}]}{D_1}\bar{q} & \frac{SbI_{zz}C_{La_1}}{D_1}\bar{q} \\ & & & \frac{S\bar{c}C_{M\alpha}}{I_{yy}}\bar{q} & 0 & \frac{S\bar{c}C_{Ma_1}}{I_{yy}}\bar{q} \\ & & & 0 & \frac{Sb[I_{xx}C_{N\beta} - IxzC_{L\beta}]}{D_1}\bar{q} & \frac{-SbI_{xz}C_{La_1}}{D_1}\bar{q} \\ 0 & 1 & 0 & \frac{\rho V_T S C_{Z\alpha}}{2m} & 0 & 0 \\ 0 & 0 & -1 & 0 & \frac{\rho V_T S C_{Y1}}{2m} & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}_{\hat{\mathbf{z}}_1(k|k)} \,, \tag{4.39}$$

with

$$\mathbf{F_1} = \begin{bmatrix} \frac{I_{zz}Sb^2C_{L\tilde{p}}}{2D_1V_T}\bar{q} - \frac{N_1}{D_1}q & \frac{-N_1}{D_1}p + \frac{N_2}{D_1}r & \frac{(I_{zz}C_{L\tilde{r}} - I_{xz}C_{N\tilde{r}})Sb^2}{2D_1V_T}\bar{q} + \frac{N_2}{D_1}q \\ \frac{I_{xx}-I_{zz}}{I_{yy}}\, r - 2\frac{Izx}{I_{yy}}p & \frac{S\bar{c}^2C_{M\tilde{q}}}{2V_TI_{yy}}\bar{q} & -\frac{I_{xx}-I_{zz}}{I_{yy}}p - \frac{2I_{xz}}{I_{yy}}r \\ -\frac{Sb^2C_{L\tilde{p}}I_{xz}}{2D_1V_T}\bar{q} + \frac{N_3}{D_1}q & \frac{N_3}{D_1}p + \frac{N_1}{D_1}r & \frac{Sb^2[-I_{xz}C_{L\tilde{r}} + I_{xx}C_{N\tilde{r}}]}{2D_1V_T}\bar{q} + \frac{N_1}{D_1}q \end{bmatrix} \,, \tag{4.40}$$

with $N_1 = I_{xz}(I_{xx} - I_{yy} + I_{zz})$, $N_2 = I_{yy}I_{zz} - I_{xz}^2 - I_{zz}^2$, $N_3 = I_{xz}^2 - I_{xx}I_{yy} + I_{xx}^2$, and $D_1 = I_{xx}I_{zz} - I_{xz}^2$.

The discrete transition matrix for the aileron1-fault filter is calculated with $\boldsymbol{\phi}_{k,\delta_{a1}} = \mathbf{I} + \mathbf{F}_{\delta_{a1}}(k)T_s$.

The control input matrix of the no-fault filter $\mathbf{G}_{\delta_{a1}}(k)$ is computed as follows:

$$\mathbf{G}_{\delta_{a1}}(k) = \begin{bmatrix} 0 & \frac{SbI_{zz}C_{La_2}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le_1}}{D_1}\bar{q} & \frac{SbI_{zz}C_{Le_2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{N_{\delta r}}}{D_1}\bar{q} \\ 0 & \frac{S\bar{c}C_{Ma_2}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me_1}}{I_{yy}}\bar{q} & \frac{S\bar{c}C_{Me_2}}{I_{yy}}\bar{q} & 0 \\ 0 & \frac{-SbI_{xz}C_{La_2}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le_1}}{D_1}\bar{q} & \frac{-SbI_{xz}C_{Le_2}}{D_1}\bar{q} & \frac{SbI_{xx}C_{N_{\delta r}}}{D_1}\bar{q} \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}_{\bar{q}(k)} . \tag{4.41}$$

The discrete control input matrix for the aileron1-fault filter is $\mathbf{G}_{\delta_{a1},k} = \mathbf{G}_{\delta_{a1}}(k)T_s$. All the other filters monitoring the other actuators are designed in a similar way.

4.6 Actuator Fault Isolation

4.6.1 Hypothesis Testing

A hypothesis testing algorithm uses the residuals and the state error covariance matrix from each EKF to assign a conditional probability to each fault scenario. The estimated state vector of the system is the sum of the state vector of each EKF weighted by its corresponding probability

$$\hat{\mathbf{x}}[k] = \sum_i \hat{\mathbf{x}}_i[k] \cdot p_i[k], \tag{4.42}$$

where $\hat{\mathbf{x}}_i[k]$ is the state estimate computed by the EKF that assumes the fault scenario θ_i. The index i covers all the fault scenarios implemented, including the no-fault case. By $p_i[k]$ we denote the probability that the i^{th} fault scenario is occurring. Now, the main difficulty lies in the on-line computation of the probability $p_i[k]$. In order to determine which fault scenario the actual plant is the closest to, we must consider the measurement data from the sensors. The last available measurement vector is $\mathbf{y}[k]$, sometimes also written $\mathbf{y}_k$ in the following. We also define the sequence of the last measurement vectors as $\mathbf{Y}_k = \{\mathbf{y}_k, \mathbf{y}_{k-1}, \mathbf{y}_{k-2}, ..., \mathbf{y}_0\}$. The fault probability $p_i[k]$ can be expressed as the *a posteriori* conditional probability $p_i[k] = p(\theta = \theta_i|\mathbf{Y}_k)$, *i.e.*, the probability that the actual plant can be categorized in the scenario θ_i given the sequence of the last measurements $\mathbf{Y}_k$.

Bayes' law states that

$$p_i[k] = p[\theta = \theta_i|\mathbf{Y}_k] = \frac{p[\mathbf{Y}_k|\theta = \theta_i]p[\theta = \theta_i]}{p[\mathbf{Y}_k]} \ , \tag{4.43}$$

where the probability $p[\mathbf{Y}_k]$ can be decomposed as

$$\begin{aligned} p[\mathbf{Y}_k] &= p[\mathbf{Y}_k|\theta = \theta_1] \cdot p[\theta = \theta_1] + ... + p[\mathbf{Y}_k|\theta = \theta_N] \cdot p[\theta = \theta_N] \ , \\ &= \sum_{j=0}^{N} p[\mathbf{Y}_k|\theta = \theta_j] \cdot p[\theta = \theta_j] \ . \end{aligned} \tag{4.44}$$

Combining (4.43) and (4.44) yields

$$p_i[k] = p[\theta = \theta_i|\mathbf{Y}_k] = \frac{p[\mathbf{Y}_k|\theta = \theta_i]p[\theta = \theta_i]}{\sum_{j=0}^{N} p[\mathbf{Y}_k|\theta = \theta_j] \cdot p[\theta = \theta_j]} \ , \tag{4.45}$$

with N being the number of different scenarios under consideration. In order to make a recursive form appear in the probabilities, the measurement data sequence $\mathbf{Y}_k$ is rewritten as the sequence $\{\mathbf{y}_k, \mathbf{Y}_{k-1}\}$:

$$\begin{aligned} p[\mathbf{Y}_k|(\theta = \theta_j)] &= p[\mathbf{y}_k, \mathbf{Y}_{k-1}|(\theta = \theta_j)] \\ &= p\left[\mathbf{y}_k|\left(\mathbf{Y}_{k-1}, \theta = \theta_j\right)\right] \cdot p[\mathbf{Y}_{k-1}|(\theta = \theta_j)] \\ &= p\left[\mathbf{y}_k|(\theta = \theta_j, \mathbf{Y}_{k-1})\right] \cdot p[(\theta = \theta_j)|\mathbf{Y}_{k-1}] \\ &= p\left[\mathbf{y}_k|(\theta = \theta_j, \mathbf{Y}_{k-1})\right] \cdot p_j[k-1] \ . \end{aligned} \tag{4.46}$$

Using the result (4.46) in (4.45) yields

$$p_i[k] = p[(\theta = \theta_i)|\mathbf{Y}_k] = \frac{p\left[\mathbf{y}_k|(\theta = \theta_i, \mathbf{Y}_{k-1})\right] \cdot p_i[k-1] \cdot p[\theta = \theta_i]}{\sum_{j=0}^{N} p\left[\mathbf{y}_k|(\theta = \theta_j, \mathbf{Y}_{k-1})\right] \cdot p_j[k-1] \cdot p[\theta = \theta_j]} \ . \tag{4.47}$$

Since a fault may occur at any time, regardless of which actuator may fail, we decide to assign the same probability to all the scenarios, *i.e.*, $p[\theta = \theta_j] = 1/N$ for $j = 1, ..., N$. Therefore, the equation above simplifies to the following recursive expression:

$$p_i[k] = p[\theta = \theta_i|\mathbf{Y}_k] = \frac{p\left[\mathbf{y} = \mathbf{y}_k|(\theta = \theta_i, \mathbf{Y}_{k-1})\right] \cdot p_i[k-1]}{\sum_{j=0}^{N} p\left[\mathbf{y} = \mathbf{y}_k|(\theta = \theta_j, \mathbf{Y}_{k-1})\right] \cdot p_j[k-1]} \ . \tag{4.48}$$

Remarks:

- By examining the probabilities, we can determine the "health status" of the system. It is either the no-fault case or a case where an actuator is locked-in-place or floating. An actuator fault is declared valid if the

corresponding fault probability exceeds 90% for a certain amount of time. A fault is declared removed when the corresponding fault probability is below 5% for a certain amount of time.

- One may notice that the denominator of (4.48) corresponds to the sum of each scenario probability's numerator, such that the fault probabilities add up to one.
- In practice, in order to prevent the possibility that the recursive computation of the fault probability in (4.48) stays at zero forever as soon as the probability reaches zero, the lower bound of each probability is set to 0.001.
- This method that uses probabilities for fault isolation is sometimes called a Bayes classifier (see [19], Chap. 16).

4.6.2 Gaussian Conditional Probability Density

We now derive an explicit formula for the term $p\left[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})\right]$, which corresponds to the probability of obtaining the measurement data $\mathbf{y}[k]$ at time $t_k = kT_s$, assuming the scenario θ_i exists and given the sequence of the last measurements $\mathbf{Y}_{k-1}$.

The probability density is chosen to be a Gaussian function [16] with its characteristic bell-shaped curve according to the following formula:

$$p\left[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})\right] = \lambda_i[k] e^{-\mathbf{r}_i[k]^T \mathbf{\Sigma}_i^{-1}[k] \mathbf{r}_i[k]/2} , \tag{4.49}$$

with $\lambda_i[k] = \frac{1}{(2\pi)^{m/2} |\mathbf{\Sigma}_i[k]|^{1/2}}$, where $|...|$ denotes the determinant of the matrix, m represents the measurement dimension, and $\mathbf{\Sigma}_i[k]$ is the residual covariance matrix calculated at time step k by the i^{th} EKF. The term $\mathbf{r}_i[k]$ corresponds to the residuals of the i^{th} EKF, when the measurement update step occurs according to the equation

$$\mathbf{r}_i[k] = \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_i(k|k-1)) . \tag{4.50}$$

Intuitive Explanation of the Probability Density

In the case of a single-input single-output problem, in which the state and measurement vectors reduce to scalars, the measurement data $h(\hat{x}_i(k|k-1))$ that we expect according to our model can be seen as the mean value of the measurement data computed by the i^{th} EKF, *i.e.*, $\hat{y}_i[k]$ in Fig. 4.6.

The width of the conditional Gaussian density is governed only by the covariance matrix $\Sigma_i[k]$. Figure 4.7 shows the shapes of the Gaussian function for several standard deviations $\sigma^2 = \Sigma(jj)$. The residual $r_i[k]$ determines

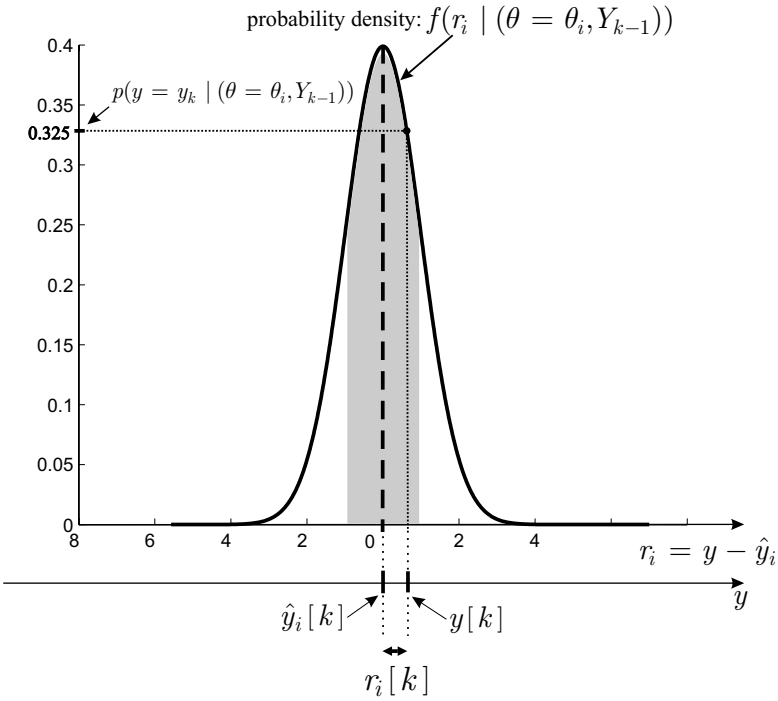


Fig. 4.6 Conditional probability density in the scalar case

the relative position of the peak of the probability density with the actual measurement y_k.

In the multivariable case, which applies in this FDI system, the fault probability $p\,[\mathbf{y} = \mathbf{y}_k|(\theta = \theta_i, \mathbf{Y}_{k-1})]$ is given by $\mathbf{f}(\mathbf{r}_i = \mathbf{r}_i[k]|(\theta = \theta_i, \mathbf{Y}_{k-1}))$, with the probability density defined as a function of the residual $\mathbf{r}_i$ with

$$f(\mathbf{r}_i|(\theta = \theta_i, \mathbf{Y}_{k-1})) = \frac{1}{(2\pi)^{m/2}|\mathbf{\Sigma}_i[k]|^{1/2}} \cdot e^{-\mathbf{r}_i^T \mathbf{\Sigma}_i^{-1}[k]\mathbf{r}_i/2} \,. \tag{4.51}$$

Therefore, the filter that corresponds to the fault scenario produces an estimate for the measurement vector $\hat{\mathbf{y}}_i[k] = \mathbf{h}(\hat{\mathbf{x}}_i(k|k-1))$ very close (apart from noise) to the actual value of the measurement data vector $\mathbf{y}[k]$. The residual $\mathbf{r}_i[k] = \mathbf{y}_k - \mathbf{h}(\hat{\mathbf{x}}_i(k|k-1))$ will be small and close to zero. This means that the corresponding probability $p\,[\mathbf{y} = \mathbf{y}_k|(\theta = \theta_i, \mathbf{Y}_{k-1})]$ is the highest for the filter matching the fault scenario.

By examining the probabilities computed with (4.52) we can determine the health status of the system, either in the no-fault case or in an actuator/sensor failure case:

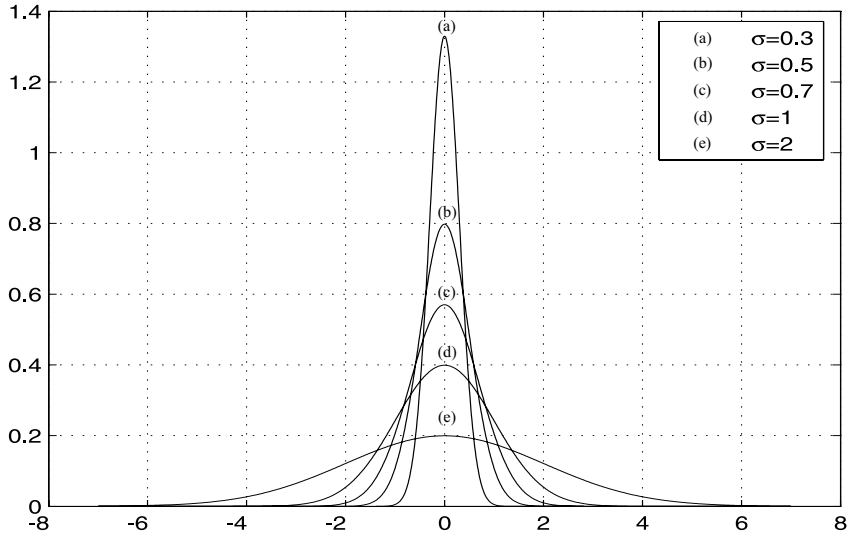


Fig. 4.7 Gaussian functions for zero-mean value and several standard deviations σ

$$p_i[k] = p[\theta = \theta_i | \mathbf{Y}_k] = \frac{p\left[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_i, \mathbf{Y}_{k-1})\right] \cdot p_i[k-1]}{\sum_{j=0}^{N} p\left[\mathbf{y} = \mathbf{y}_k | (\theta = \theta_j, \mathbf{Y}_{k-1})\right] \cdot p_j[k-1]} \,. \quad (4.52)$$

Remarks:

- The hypothesis testing uses a Gaussian density function, which assumes that the residuals from the EKFs are Gaussian distributed. When this is not the case, there is a little inconsistency with the application of the theory. However, the assumption that these residuals are Gaussian distributed is still reasonable, especially when the aircraft dynamics are slow.
- The reason why we use a Gaussian distribution to describe the probability density of the current measurement to take on the value y_k based on the fault hypothesis θ_i and the previous measurements Y_{k-1}, is to make the mathematics tractable. As mentioned in the book by Maybeck [16], "the Kalman filter, which propagates the first and second-order statistics [mean and variance of a process], includes all information contained in the [Gaussian] conditional probability density, rather than only some of it, as would be the case with a different form of density". If another probability density function was known, then (4.51) could be changed accordingly.

4.7 Simulation Results of the EMMAE-FDI with no Supervision System

4.7.1 Simulation Conditions

In order to obtain realistic simulations, the sensor measurement data are corrupted on purpose with zero-mean white Gaussian noise corresponding to typical specifications of low-cost sensors. For the turn rate sensors, the standard deviation is $\sigma_{p,q,r} = 5\,\mathrm{deg/s} = 0.0873\,\mathrm{rad/s}$, which corresponds to a noise covariance of $\mathbf{\Sigma}_{p,q,r} = 0.0076 \times I_3$ [rad^2/s^2]. For the air-flow angle sensors, the noise standard deviation is $\sigma_{\alpha,\beta} = 2\,\mathrm{deg} = 0.0349\,\mathrm{rad}$ ($\mathbf{\Sigma}_{\alpha,\beta} = 0.0012 \times I_2$ [rad^2]). The airspeed sensor noise has a standard deviation of $\sigma_{V_T} = 1\,\mathrm{m/s}$ ($\Sigma_{V_T} = 1\,\mathrm{m^2/s^2}$). Poor sensor quality adversely affects the FDI reliability.

When there is little excitation and when the faulty actuator deflection is close to the trim conditions, an actuator failure becomes even more difficult to detect. Indeed, due to large sensor noise, the control signals (see Fig. 4.8) become noisy as well, which reduces the difference between the actual faulty actuator deflection and its corresponding control signal. The larger this difference, the easier it is to detect the fault.

The EKF process noise covariance matrix and the sensor noise covariance matrix are selected as follows: $\mathbf{R}_w = 0.002 \times \mathbf{I}_5$ and $\mathbf{R}_v = \mathrm{diag}[0.1 \times \mathbf{I}_3 \quad 0.02 \times \mathbf{I}_2]$.

4.7.2 Scenario

The scenario to test the fault detection method is chosen to put the FDI in the most difficult conditions, which are those of minimum excitation of the system. This is achieved when the aircraft is flying straight and level (no maneuvers, no wind) at a constant speed of 30 m/s.

The actuator faults are simulated by blocking the control surfaces close to the trim deflections corresponding to straight level flight conditions, because those faults are harder to detect and to estimate. For example, the ailerons and the rudder are intentionally made to fail close to the neutral deflection (0°), and the elevators are made to fail close to −2°. Fault detection with the EMMAE method is tested on a six degrees-of-freedom nonlinear aircraft model of a UAV currently in use at the laboratory.

Simulations were performed in MATLAB®/Simulink® on closed-loop control architecture, with a nonlinear autopilot which regulates the speed, altitude, and the attitude of the aircraft. The actuator configuration of the aircraft is depicted in Fig. 4.4. The EMMAE-FDI system is therefore composed of six EKFs, one for monitoring the no-fault case, two for monitoring

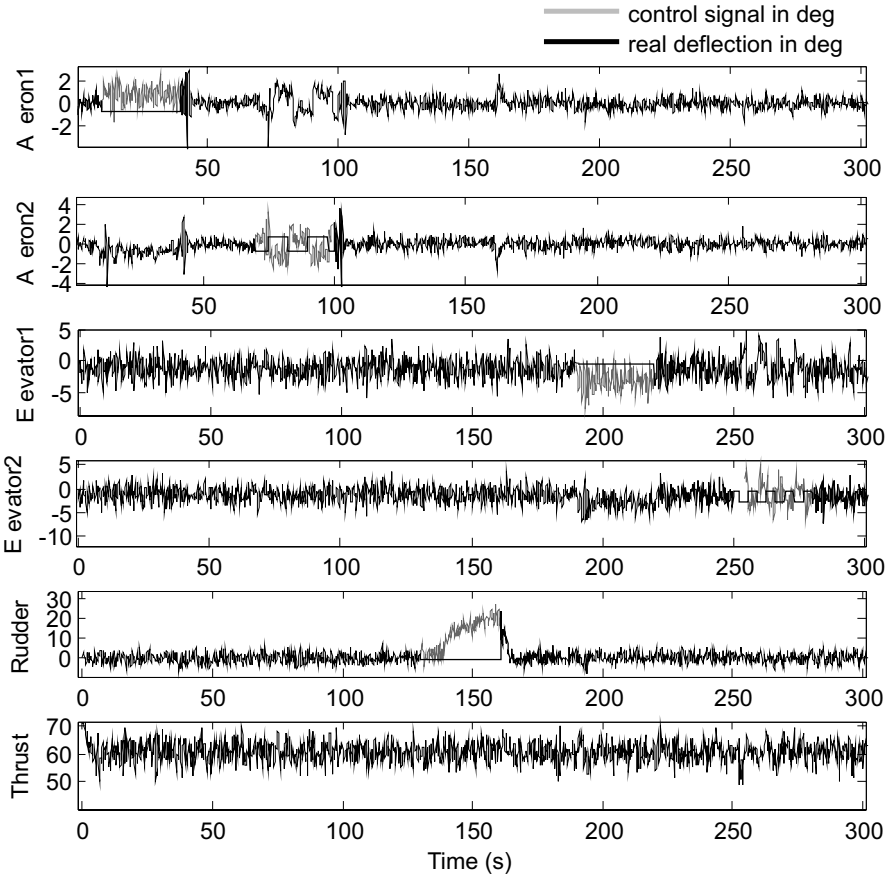


Fig. 4.8 Control signals and actual actuator deflections

each aileron (one on each wing), two EKFs for monitoring each of the two independent elevators, and one EKF for the rudder.

As depicted in Fig. 4.8, a sequence of consecutive faults is generated. From t = 10...40 s aileron 1 fails and is locked at $-1°$ deflection, for t = 70...100 s aileron 2 fails and is "floating" between the two positions $-1°$ and $1°$ in a square-wave fashion. For t = 130...160 s the rudder fails and is locked at $-1°$. For t = 190...220 s elevator 1 gets locked at $-0.5°$, and finally, for t = 250...280 s elevator 2 is floating between two uncontrolled positions $-1°$ and $-3°$ in a square wave fashion. After this sequence of faults, the aircraft continues to fly straight and level, and no more faults are introduced.

4.7.3 Comments on the Simulation Results

Figure 4.9 shows the results obtained by the FDI system after the sequence of faults. The top plot labeled "No-Fault" has a probability of 1 when the EMMAE-FDI system does not detect any fault in the aircraft. An actuator fault is declared valid if the corresponding fault probability exceeds 90% for a certain amount of time. A fault is declared removed when the corresponding fault probability is below 5% for a certain amount of time. When aileron 1 fails at t = 10 s, the "No-Fault" filter needs about 6 s for its probability to go down to almost 0, which means that a failure occurred somewhere. After the aileron 1 fault is introduced, both probabilities for aileron 1 and 2 to fail (Pbaf1, Pbaf2) start rising at t = 11.5 s. At t = 17 s, the FDI begins to distinguish between the two ailerons which one has failed, and Pbaf2 returns to zero while Pbaf1 rises up to 90% at t = 34 s. Therefore, it took 24 s for the FDI to indicate that aileron 1 experiences a failure. At t = 40 s, the fault of aileron 1 is removed and the actuator behaves normally again. Figure 4.9

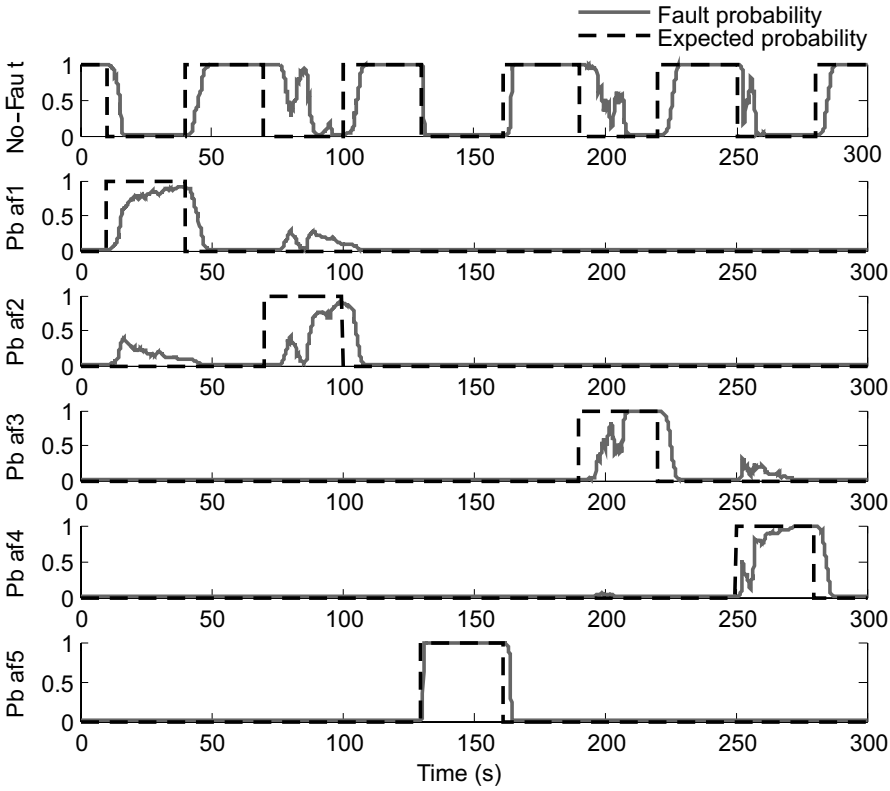


Fig. 4.9 Probabilities from each filter of the EMMAE-FDI after a sequence of faults (no supervision system)

shows that the probability Pbaf1, indicating whether aileron 1 fails, decreases slowly and reaches 0 again after 10 s, while the "No-Fault" probability rises accordingly. It thus takes 10 s for the FDI to indicate that the fault has been removed.

As Fig. 4.8 shows, at t = 70 s aileron 2 fails and has an uncontrolled square wave motion between $-1°$ and $1°$. Figure 4.9 shows that the FDI takes 20 s to detect such a failure. We observe an ambiguity between the two ailerons for a few seconds before the probability Pbaf2 finally reaches 90% at t = 90 s. At t = 100 s the fault of aileron 2 is removed, and this actuator behaves normally again. However, the FDI is not capable of detecting quickly that the fault has been removed. It takes about 8 s to do so.

For the rudder, the fault is introduced at t = 130 s and is isolated by the FDI system at t = 131 s when Pbaf5 exceeds 90%. The fault removal is detected in less than 5 s. It takes less time to isolate a rudder fault and to detect its removal because there is only one rudder, unlike the other actuators which are redundant (two ailerons, two elevators). Therefore, a malfunctioning rudder cannot be compensated by a redundant rudder, thus resulting in no actuator-fault ambiguity.

The introduction of the elevator 1 fault at t = 190 s is isolated by its corresponding filter (probability signal Pbaf3 exceeding 90%) at t = 208 s. After the fault removal, the FDI system takes 8 s to indicate the fault removed. We observe the same kind of behavior for elevator 2 as for elevator 1. Finally, after the last fault has been removed, 5 s are needed by the FDI system to slowly build up probability in the "No-Fault" filter to indicate that no more faults are detected in the system.

4.7.4 Remarks on the First Attempt to Use the EMMAE-FDI System

- The results plotted in Fig. 4.9 indicate that the current implementation of the method is able to detect the fact that a failure occurred, even in a very low-excitation case, but it could not tell quickly and reliably which actuator experienced the fault. In cases of redundant actuators having the same influence on the aircraft aerodynamics, the EMMAE method has difficulty quickly resolving ambiguities between redundant actuators when they cannot be properly excited.
- It appears that failures of actuators near trim deflection are more difficult to detect and isolate.
- Moreover, whenever a fault is removed, the EMMAE method alone requires a long time to detect that fact. It is critical, however, that the FDI quickly detects the removal of a failure, or quickly recognizes that a false alarm has been triggered due to possible external perturbations, such as strong wind gusts.

- Finally, it is critical that the probabilities quickly reach the "expected values" which correctly describe the fault scenario. Indeed, the estimated state vector of the system, which is the sum of the state vector of each EKF weighted by its corresponding probability, must be sufficiently correct and accurate if this state estimate is fed back to the controller.

4.8 Improvements to the EMMAE-FDI System

This section describes the techniques that have been added to enhance the performance of the FDI system when there is very low excitation of the system, particularly during steady level flight. In order to improve the speed and the accuracy of the fault isolation, a supervision module is designed whose tasks are detailed below.

4.8.1 Design of an Active Supervision Module (Supervisor)

The supervision module shown in Fig. 4.10 is designed to monitor probability signals from the FDI. If an actuator-failure probability exceeds a certain threshold for some time, then the supervisor is designed to superimpose an artificial control signal on the corresponding actuator. If an actuator fails, the additional signal will have no effect on the aircraft dynamics, but it will help the FDI confirm more quickly the failure of this actuator. On the other hand, if the actuator has not actually failed, the aircraft will respond according to the additional signal, and the FDI will then remove the fault assigned to this

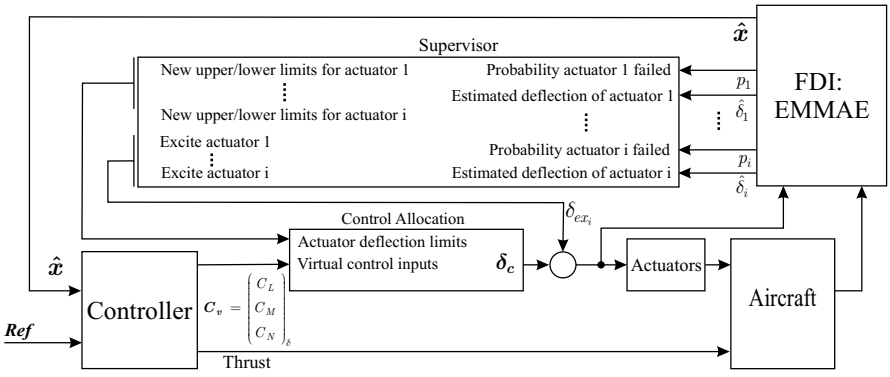


Fig. 4.10 EMMAE-FDI system in a reconfigurable flight control system

actuator. If the corresponding fault probability falls below a certain threshold for a defined period of time, the supervisor removes the superimposed signal.

It has been shown that significant identification improvements can be obtained by optimizing the signals applied to the control surfaces [20]. Actuator excitation methods have already been developed for system identification purposes. Null-space injection methods have been designed that work directly in optimization-based control allocation approaches, which generate actuator excitation while still providing the desired pseudo-commands [21, 22]. In this work, the excitation signals are adaptively controlled within the limits from 1 to 4 degrees, with the function $\delta_{ex\_i}(t) = [1 + 3(1 - p_i(t))] \cos(2\pi f_i t)$.

Note that an actuator is only excited when its corresponding fault probability p_i exceeds 5%. Most of the time, only one actuator is excited. Although better performance may be expected if the excitation signals to each actuator were independent and uncorrelated, we used the same excitation signal for all actuators. The frequency of the signal is to be chosen within the range of the aircraft bandwidth, in our case f_i=1 Hz.

The excitation signal has an adaptive amplitude dependent on the probability p_i of actuator i to have failed. In this way, when the probability p_i is low, the excitation amplitude is large and *vice versa*. Simulation results show that this adaptive amplitude for the excitation signal efficiently improves the accuracy and speed for fault isolation compared with a fixed amplitude excitation signal. This adaptive amplitude ensures that the actuator is excited as little as possible, but still enough to isolate the fault or to remove a false alarm. Figure 4.11 shows the practical implementation of the excitation signal generator and provides an example for aileron 1.

This method is therefore a systematic way of testing each triggered failure alarm, to confirm it or to remove it, hence making the FDI more robust. Whereas other proposed schemes [5, 23] suggest having the aircraft perform a "health-check maneuver" or "diagnostic maneuver" as soon as a failure is detected, in the method introduced in this work the actuator is directly excited by the supervision module rather than by the aircraft autopilot, yielding much faster and more accurate fault isolation.

4.8.2 Performance of the EMMAE-FDI with the Supervision System

4.8.2.1 Detection Performance

Figure 4.12 shows how the aileron faults are accurately detected and isolated in less than 5 s. The rudder fault is isolated after 1 s. The elevator faults take longer (about 9 s) to be isolated. However, the removal of all the faults is detected in less than 5 s. Furthermore, there is no more ambiguity or false detection among the actuator faults. Comparing the results shown in Fig.

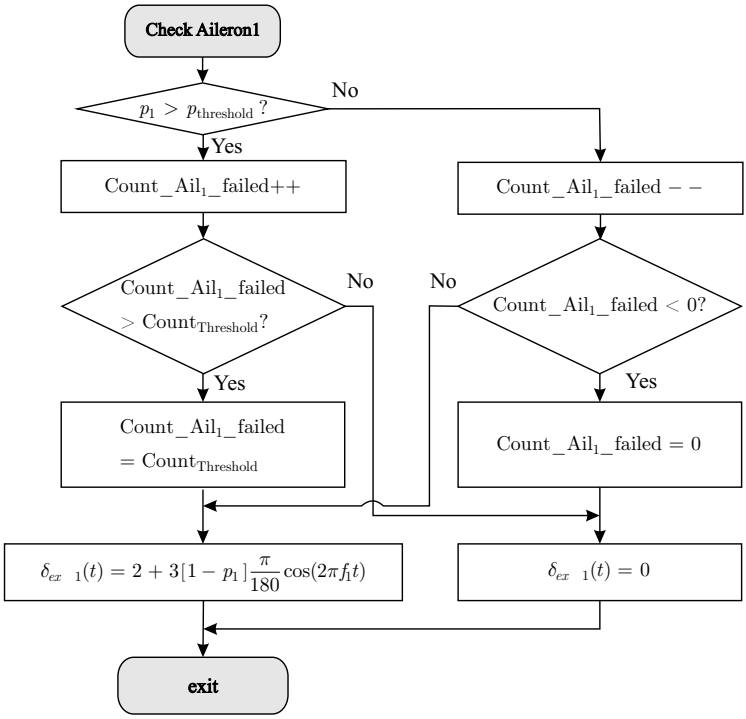


Fig. 4.11 Generation of the actuator excitation signal (example for aileron 1)

4.12 with those of Fig. 4.9, we see that the performance and the robustness of the EMMAE-FDI system have greatly improved due to the supervision module.

We recall that these results are obtained in the most difficult conditions for the FDI system. Indeed there is no external disturbance such as wind gusts, the aircraft flies straight and level, and the actuators fail close to their trim deflection.

4.8.2.2 State Estimation Performance

The left plot in Fig. 4.13 shows a comparison between noisy measurements and the state estimate from the EMMAE method in the implementation of Fig. 4.10. The right plot in Fig. 4.13 shows a comparison between noiseless measurements and the state estimate from the EMMAE method. Clearly, despite a large amount of noise in the sensor measurements, the state estimate accurately tracks the true (noiseless) measurements, and the filtered data enable the controller to generate control signals with less noise, which facil-

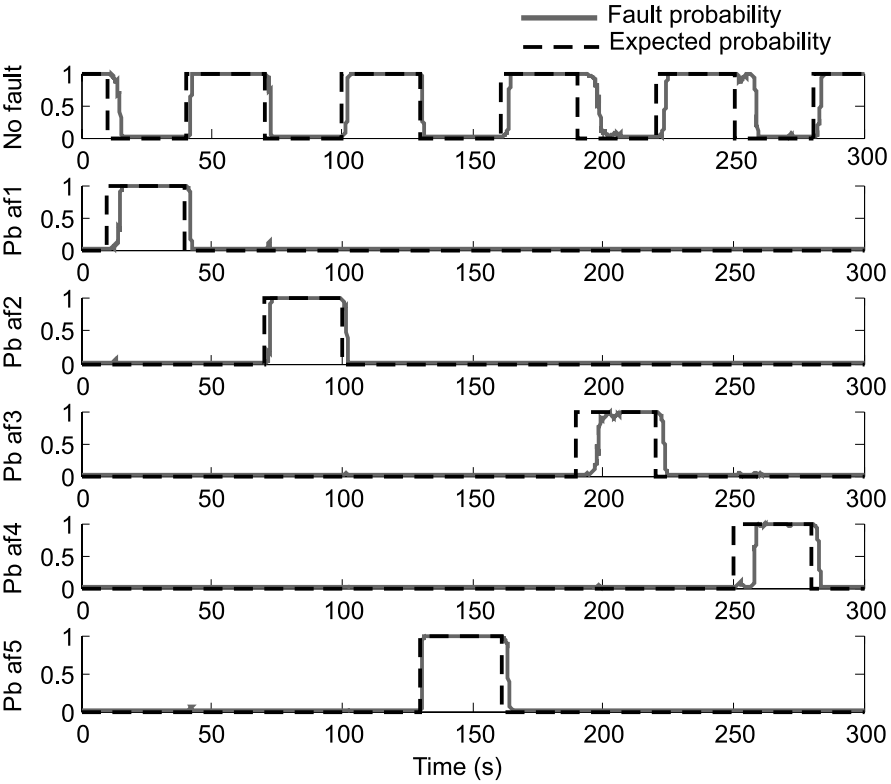


Fig. 4.12 Probabilities from each filter of the EMMAE-FDI after a sequence of faults (with the supervisor)

itates fault detection and isolation as well as state estimation from correct probability computations.

Furthermore, as shown in Fig. 4.14, the faulty control surface deflection is well estimated in the case of a frozen actuator, but it is slowly estimated in the case of an uncontrolled square wave motion of the control surface. An engineering fix to have the EKF track the faulty actuator deflection faster is to increase the process noise on the desired control input; it is, however, to the detriment of estimate accuracy. In practice, it is more likely that an actuator simply locks at a certain deflection or floats around the local and slowly varying angle of attack, thereby reducing the challenge of estimating the faulty actuator deflection.

The elements of the state vector differ from their respective estimate whenever the probabilities are not correct. As mentioned, the simulations that are presented here are for straight and level flight, *i.e.*, the worst condition for any FDI system. In practice, a small UAV is maneuvering almost all the time, which further improves the results and the performance of the FDI system.

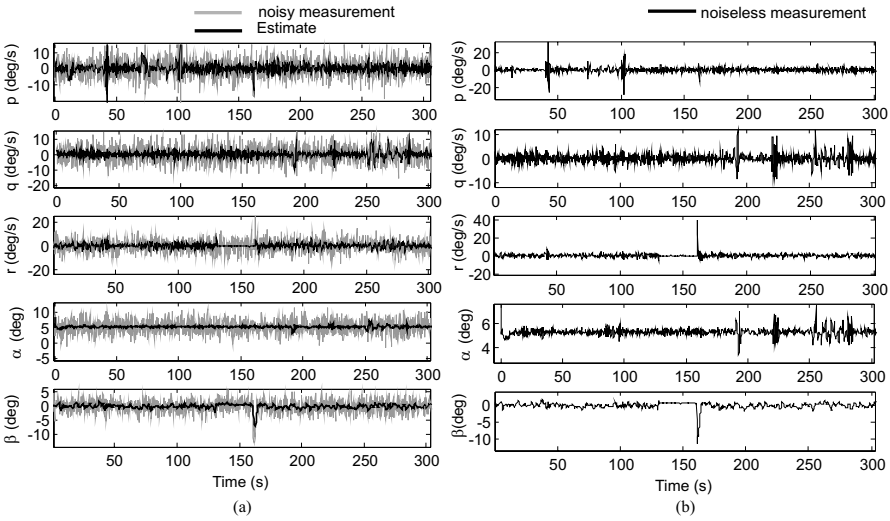


Fig. 4.13 Comparison between: (a) noisy; and (b) noiseless measurements and the probability-weighted state estimate from the EMMAE method

4.9 A Realistic Flight Scenario

In this section, the FDI system is tested in a realistic flight scenario, in which the UAV takes off, tracks a predefined trajectory, and follows an altitude and a speed reference profile. First, the simulation is done without wind, and in a second phase, Dryden Wind turbulences are included in the simulation to test the robustness of the FDI system in the presence of external disturbances.

4.9.1 No-wind and No-actuator-fault Conditions

Figure 4.15 shows the flight path of the aircraft when there is no wind and no actuator failure. The left plot shows that the aircraft takes off at point 1 and flies in the direction of the points 2, 3, ..., 6 and then flies again the whole sequence of points. The overshoot during the turns after the points 3 and 6 are due to the guidance system and the excessive speed of the aircraft when approaching these points. Lower speeds result in less overshoot.

The speed profile shown in the top right plot of Fig. 4.15 is chosen to cover a significant range of the aircraft speed in order to test the performance of the fault detection system at different operating conditions. Clearly, the measured airspeed V_T is corrupted by noise, like all the other measurement signals defined in Sect. 4.7.1 that are used by the KFs in the FDI system.

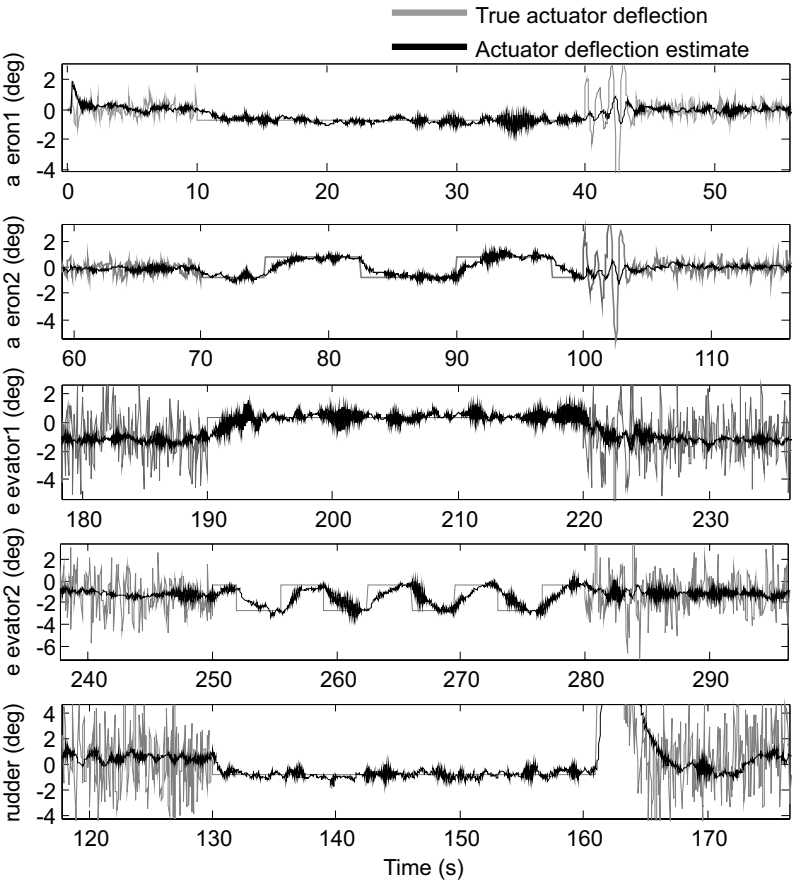


Fig. 4.14 True actuator positions and associated estimates (a position estimate is only valid during the occurrence of a fault)

Also, the altitude reference signal is chosen such that the aircraft has the vertical motion depicted in the bottom right plot of Fig. 4.15.

Figure 4.16 shows the results of the FDI system, namely the fault-probability signals. Even though no actuator fault has been introduced yet in the simulation of Fig. 4.15, Fig. 4.16 shows that the FDI system without the active fault supervision system indicates that some actuator faults appear and disappear between $t = 80\,s$ and $t = 180\,s$, which corresponds to the time interval in which the aircraft speed exceeds $45\,m/s$.

In Fig. 4.16, the ailerons are assigned a fault probability of around 50% between the time intervals [80 ... 105 s], [110 ... 117 s], [120 ... 127 s] and [138 ... 162 s], which correspond to the straight flight path segments [4,5], [5,6], [6,1] and [1,2], respectively. Clearly, the system suffers from a lack of persistent excitation in these time intervals, in particular for the roll and yaw

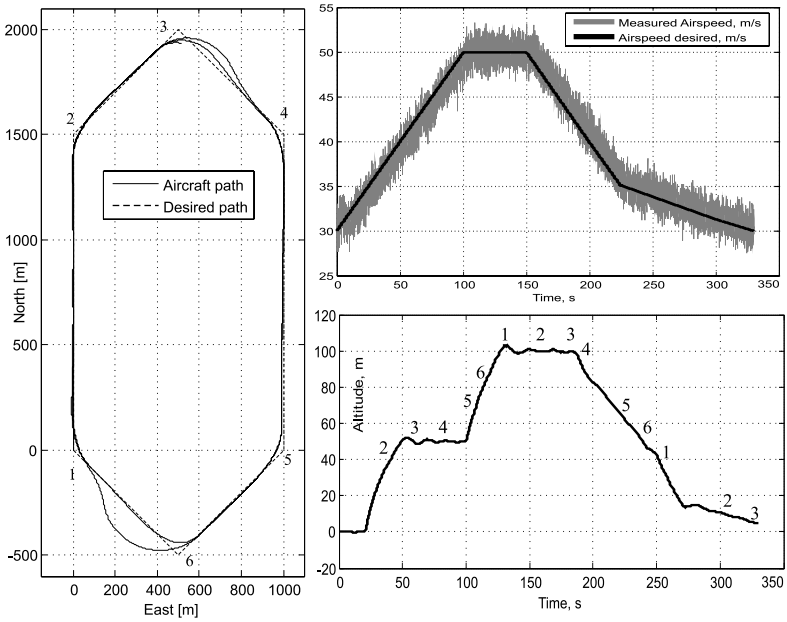


Fig. 4.15 Flight simulation with no wind and no actuator fault

axes. This issue is efficiently resolved as soon as the active fault supervision system of Sect. 4.8.1 is enabled.

The artificial excitation signals that are superimposed on actuator commands by the active fault supervision module are shown in Fig. 4.17. As expected, the roll axis is excited by aileron commands during the phases of straight flight and when the speed exceeds 45 m/s.

Figure 4.18 shows that the false actuator fault alarms are eliminated. Therefore, the active fault supervision module contributes significantly to increasing the robustness of the FDI system by providing the system with enough excitation. Actuator faults are thus accurately detected.

Apparently, the higher the speed of the aircraft, the greater the need for exciting the aircraft around its rotational axes to achieve proper fault diagnosis. This may be explained by the fact that the value of the terms in the dynamic matrix in (4.39) increases with the square of the aircraft speed through the dynamic pressure. Therefore the KFs become more sensitive to the noise of the turn rates when they are not excited (close to zero). In turn, residuals build up and false alarms appear. The artificial excitation of the turn rates by the supervision module can be seen as a means of increasing the signal-to-noise ratio of the turn rates during steady flight regimes.

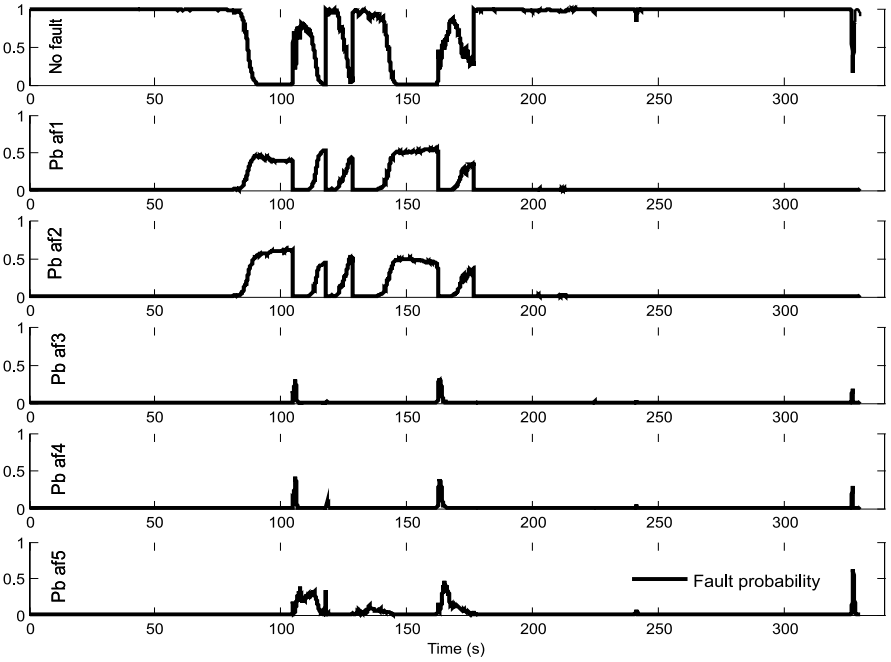


Fig. 4.16 Fault probabilities in the case of no wind, no actuator fault, and no active fault supervision system

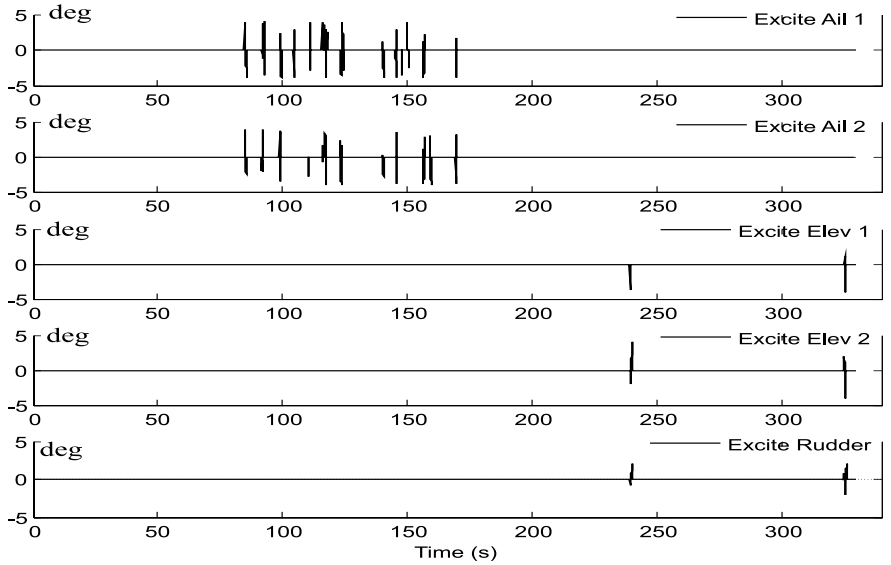


Fig. 4.17 Artificial excitation signals superimposed on the actuator control signals by the active fault supervision system of Sect. 4.8.1, yielding the FDI results in Fig. 4.18

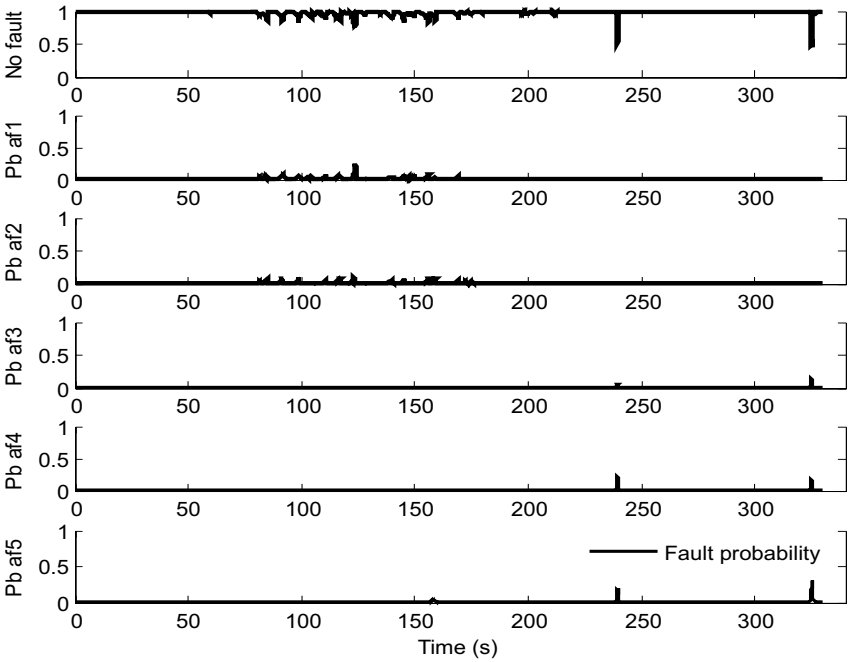


Fig. 4.18 Fault probabilities in the case of no wind, no fault, and with active fault supervision

4.9.2 Wind Conditions and No Actuator Faults

In this section, Dryden Wind turbulences are included in the simulation to test the robustness of the FDI system in the presence of external disturbances.

Figure 4.19 shows the wind speeds in the North, East and Down directions. They correspond to severe wind conditions for the type of aircraft simulated, which is a small aerobatic model aircraft.

The guidance system used in this simulation is described in Chap. 9 and successfully compensates for wind disturbances since the flight path is almost identical to the one flown without wind. The left plot in Fig. 4.20 shows the flight path of the aircraft. The top right plot shows how the speed measurement signal is disturbed by the wind gusts. The bottom right plot shows how the altitude tracking is perturbed by the wind as well.

Figure 4.21 shows that the wind gusts contribute to exciting the aircraft naturally, such that the fault alarms that appeared on Fig. 4.16 no longer exist. Moreover, Fig. 4.22 shows how the use of the active actuator fault supervision system helps in removing some false alarms even in the case of severe wind conditions. For example, for $110\,\mathrm{s} < \mathrm{t} < 115\,\mathrm{s}$ or $195\,\mathrm{s} < \mathrm{t} < 200\,\mathrm{s}$, the FDI system with no supervision module indicates that the ailerons are

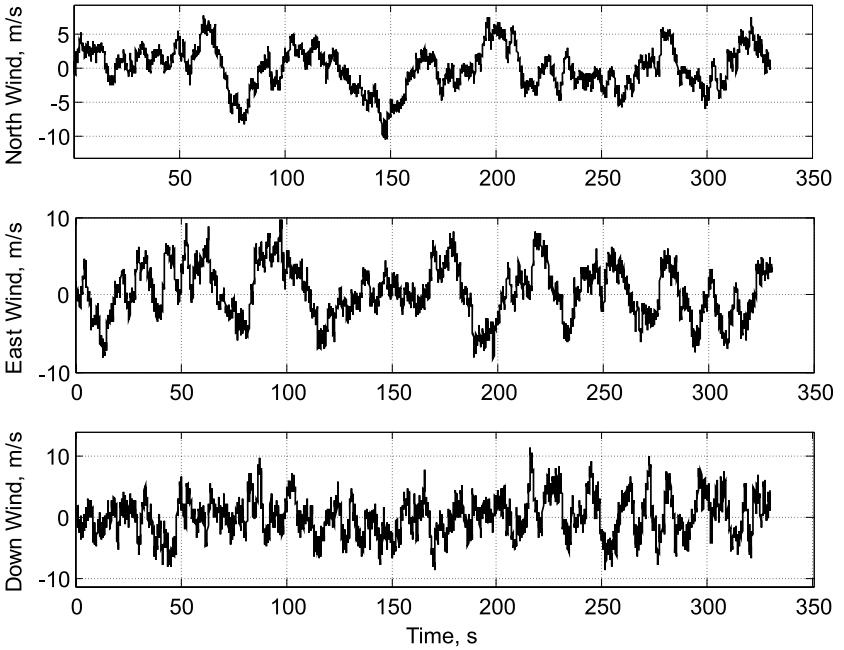


Fig. 4.19 Wind speed in m/s

failing, whereas with the supervision module, these two false fault alarms do not appear. In such wind conditions, the active supervision system would be more effective if the amplitude of the excitation signals was higher. Therefore, the wind speed or a level of wind disturbance might be an additional criterion to design the adaptive excitation amplitude.

4.9.3 Strong Winds, Actuator Faults and Active Supervision Module

In this section, the previous flight scenario is simulated again with the sequence of faults presented in Sect. 4.7.1. The fault-probability signals for each actuator are generated by the EMMAE-FDI system and are shown in Fig. 4.23. The FDI system detects and isolates the actuator faults in less than 5 s for aileron 1 (Pbaf1), 3 s for aileron 2 (Pbaf2), 2 s for elevator 1 (Pbaf3), 3 s for elevator 2 (Pbaf4), and 1 s for the rudder (Pbaf5). In all cases, the removal of the fault is accurately detected in less than 1 s.

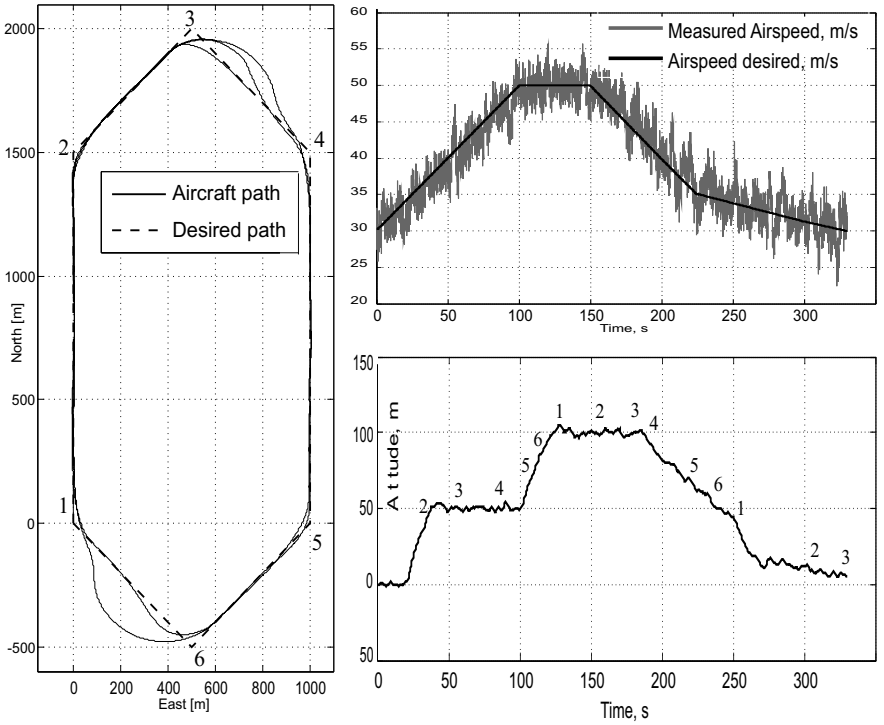


Fig. 4.20 Flight simulation in wind conditions

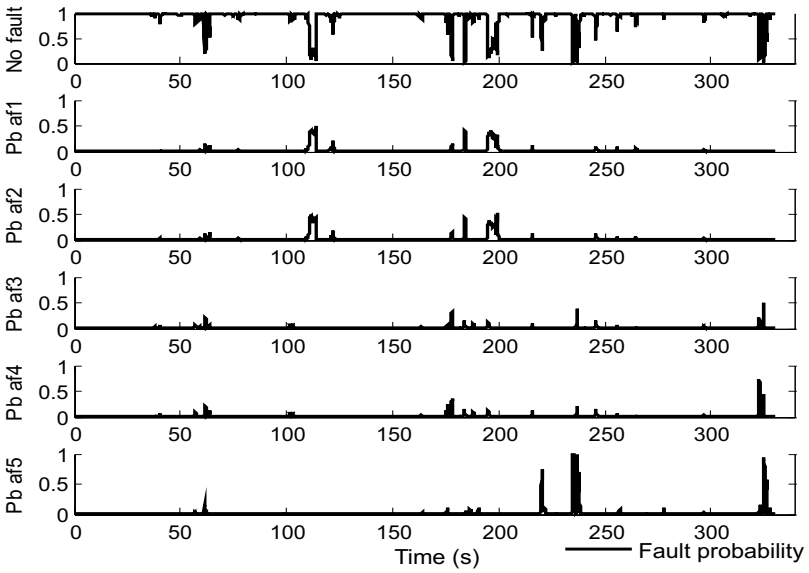


Fig. 4.21 Fault probabilities in the case of strong winds, no actuator fault, no active fault supervision system

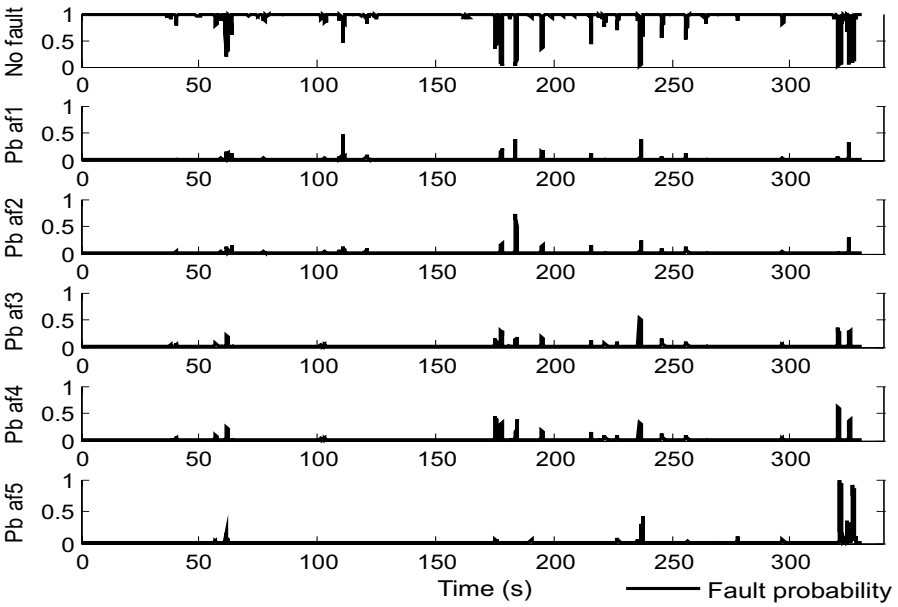


Fig. 4.22 Fault probabilities in the case of strong winds, no actuator fault, with active fault supervision system

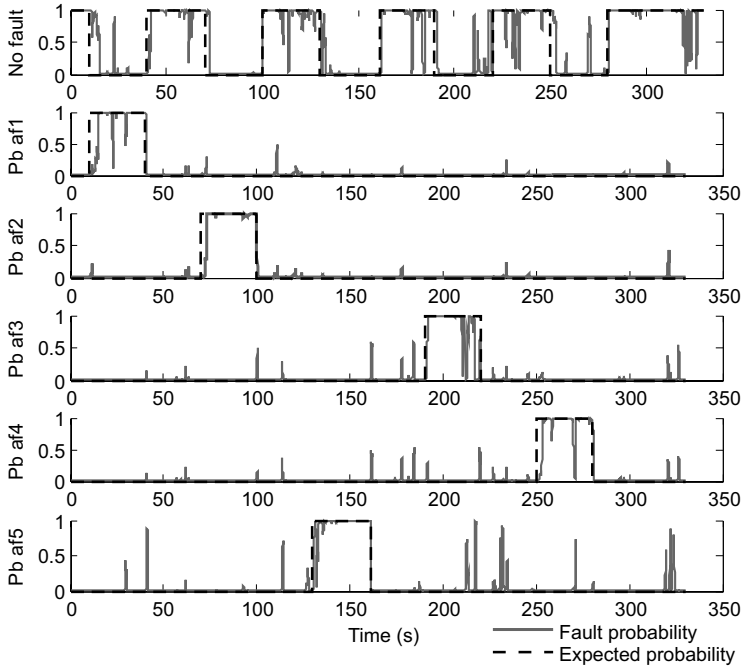


Fig. 4.23 Fault probabilities in the case of severe wind, with a sequence of actuator faults, with the active fault supervision system

4.10 An Additional Filtering Stage for the EMMAE-FDI System

The results shown in Fig. 4.23 are still not completely satisfactory since the fault-probability signals are noisy. In order to improve the quality of the final fault diagnosis of the aircraft, an additional filtering stage is added to the EMMAE-FDI system. As shown in Fig. 4.24, each fault-probability signal $p_1, p_2, \ldots, p_i$ passes through a low-pass filter whose cut-off frequency is 0.2 Hz, and a hysteresis block that sets its output to one if its input exceeds 0.6, and sets its output back to zero when the input goes below 0.4 again. After passing through this additional filtering stage, the signals of Fig. 4.23 are transformed into the filtered actuator-fault probabilities $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$ shown in Fig. 4.25.

Although it may take longer to finally isolate the fault in the system, the results shown in Fig. 4.25 are in accordance with the expected fault probabilities. We recall that this simulation is made in severe wind conditions, flying the flight path shown in Fig. 4.20 at different speeds and altitudes.

Figure 4.25 shows that the noise in the probability signals $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$ is almost completely removed. These filtered probabilities can thus be used to select alternative flight modes in a reconfigurable flight controller, for example. Note that the excitation signals are still triggered by the unfiltered

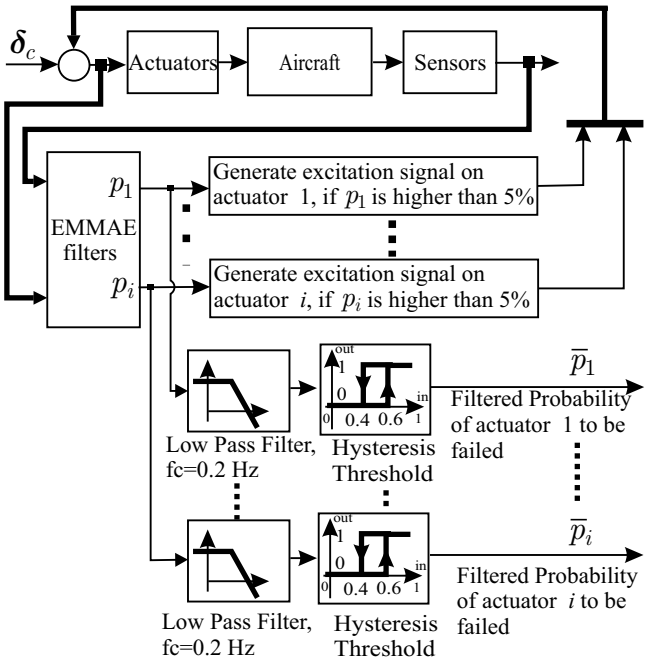


Fig. 4.24 Complete architecture of the active actuator-fault detection, isolation and supervision system

probability signals that come directly from the EMMAE module. In this way, we ensure that the system is excited after any variation of the probabilities $p_1, p_2, \ldots, p_i$.

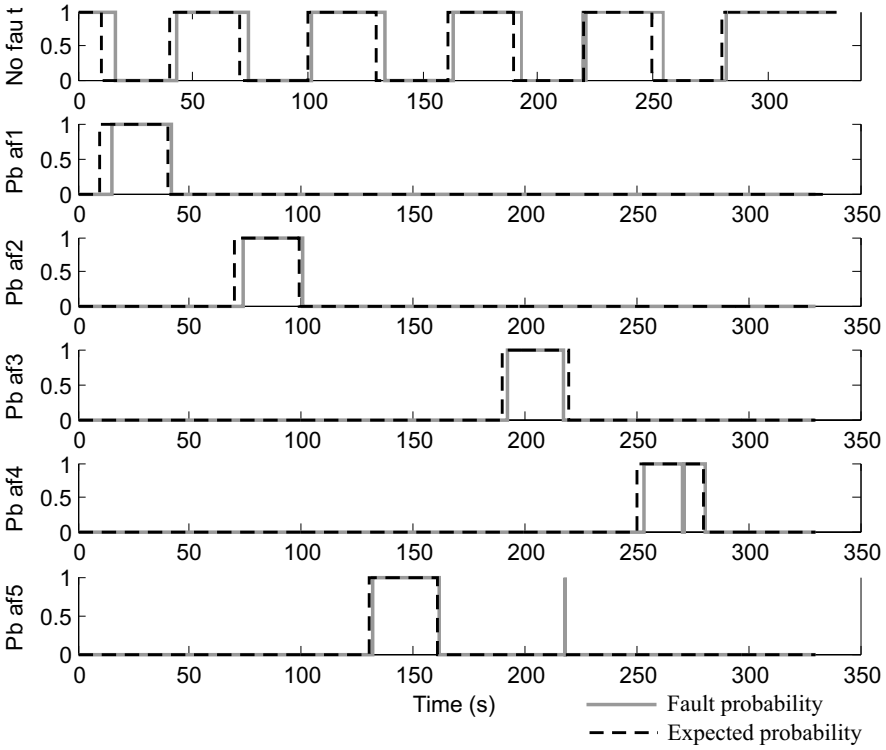


Fig. 4.25 Filtered actuator-fault probabilities $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$ after the second stage of filtering in the active FDI supervision module

4.11 Detection and Isolation of Simultaneous Failures

This section addresses the case of a second fault occurring simultaneously in the system. The EMMAE-FDI is slightly modified to detect and isolate the occurrence of a second fault and to continue the monitoring of the fault that occurred first. The system described here works properly for the detection and isolation of the second fault as long as the first fault does not change, *i.e.*, the faulty control surface remains locked. In such a case, the filter i that isolated the first actuator fault provides an estimation of the deflection of the faulty control surface. This estimation $\hat{\bar{\delta}}_i$ is used as a new input instead

of the input δ_i for all the other filters in the bank of EKFs in the EMMAE-FDI. Note that the value of $\hat{\bar{\delta}}_i$ used as input for the other filters must be the value before the second fault occurred. Indeed, when the second fault k occurs, then the i^{th} filter has an input δ_k that is false, leading to the wrong estimation of $\hat{\bar{\delta}}_i$. Figure 4.26 shows the modifications that are to be made at the input of the EKFs in the case of the first failure occurring on actuator 1.

Figure 4.27 shows the probability signals $\bar{p}_1, \bar{p}_2, \ldots, \bar{p}_i$ generated at the output of the FDI-system shown in Fig. 4.24. Aileron 1 fails between t = 30 ... 110 s and elevator 1 fails between t = 50 ... 90 s. Pbaf11 indicates the probability of aileron 1 failing only, Pbaf13 indicates the probably of both actuators (aileron 1 and elevator 1) failing at the same time. The system can properly detect and isolate the fault situations. The flight scenario that yields the results of Fig. 4.27 and Fig. 4.28 corresponds to straight level flight, with no wind. The maximum time delay needed to isolate the first fault is 3.8 s and 2 s for the second failure.

Figure 4.28 shows the estimation of the deflection of the faulty control surfaces. As expected, the estimation of the position of the failing aileron is valid as long as the second fault is not introduced, *i.e.*, between the time intervals t = 30 ... 50 s and t = 90 ... 110 s. The deflection of the failing elevator is properly estimated during the occurrence of the corresponding fault between t = 50 ... 90 s.

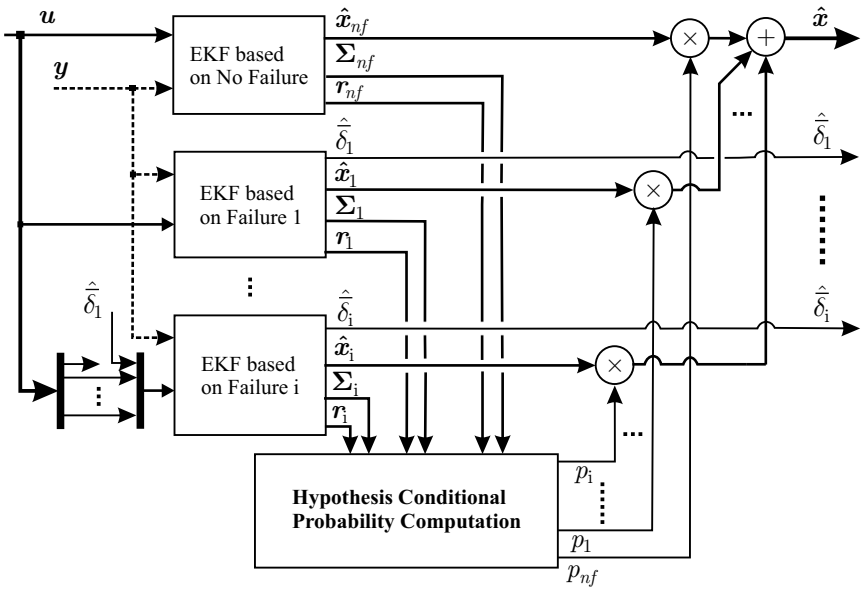


Fig. 4.26 Modified EMMAE-FDI for the isolation of a second simultaneous fault

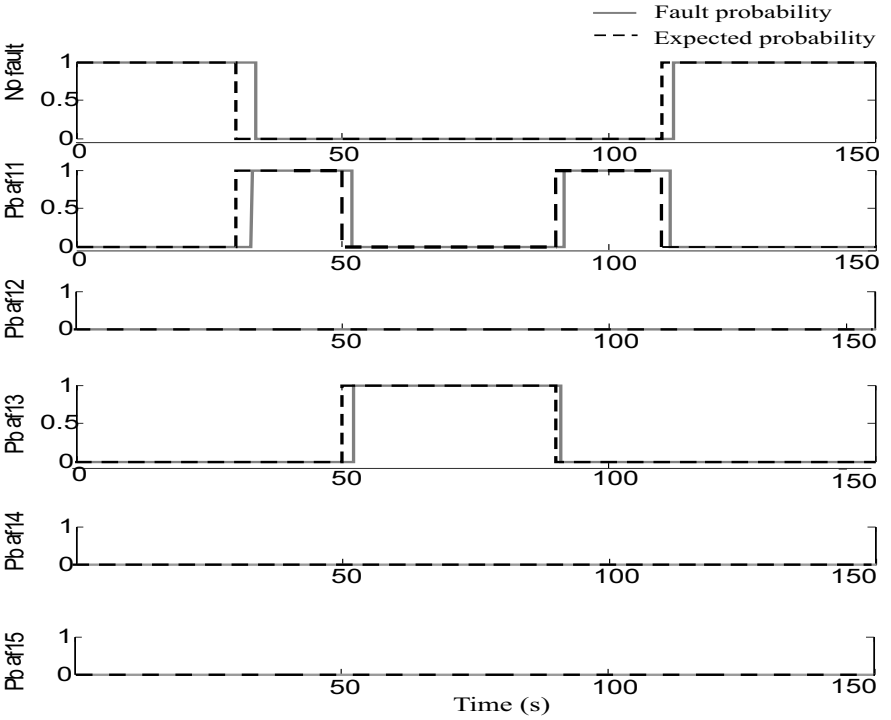


Fig. 4.27 Fault isolation in the case of two simultaneous failures

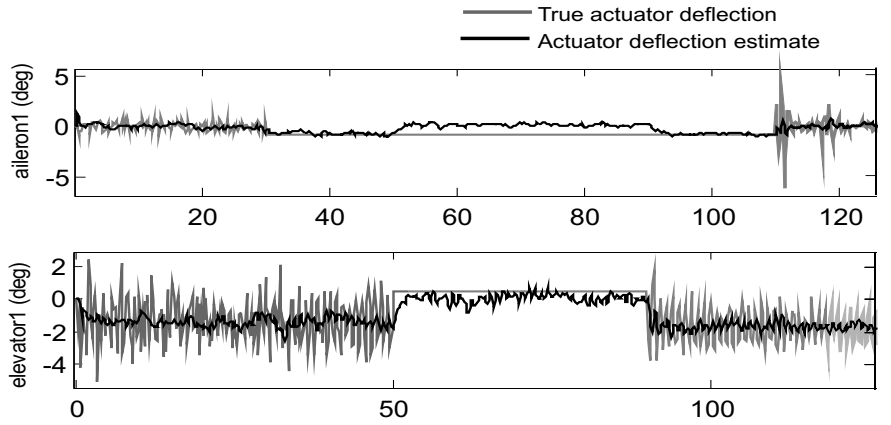


Fig. 4.28 Estimation of the deflection of the faulty control surfaces

4.12 Use of the EMMAE-FDI for a Reconfigurable Flight Control System

The design of a reconfiguring control system in the event of a major failure or damage is a challenging task [24]. Many papers show how to build an adaptive controller using a multiple-model structure. An overview of these architectures and their robustness properties can be found in [25]. This section shows a new feature of the EMMAE-FDI, which consists of using the estimation of the faulty control surface to reconfigure part of the control system depicted in Fig. 4.10, using control allocation techniques; see Chap. 5.

4.12.1 Control Allocation

The controller generates a virtual control command $\mathbf{C_v} = [C_L \quad C_M \quad C_N]^T$ in terms of desired roll, pitch, and yaw torque. This virtual command $\mathbf{C_v}$ is passed to the control allocator, which is provided with each actuator's position limits and effectiveness to produce any torque component of the $\mathbf{C_v}$ signal. A constrained optimization problem is then solved on-line to optimally generate control signals for actuators.

The control allocator solves the following (possibly underdetermined) constrained system of equations, which may be regarded as a mapping in the controlled system $\mathbf{g}(\boldsymbol{\delta}(t)) = \mathbf{C_v}(t)$, with the true actuator control signals $\boldsymbol{\delta}(t) \in \mathbf{R}^N$ and N being the number of control surfaces. After linearization, the mapping equation may be rewritten in the standard formulation of the constrained linear control allocation problem:

$$\begin{aligned} \mathbf{B}\boldsymbol{\delta}(t) &= \mathbf{C_v}(t), \\ \underline{\delta}_i(t) \le \delta_i(t) &\le \overline{\delta}_i(t), \end{aligned} \tag{4.53}$$

with the constraints

$$\begin{aligned} \underline{\delta}_i(t) &= \max\{\delta_{i,\,min},\ \rho_{i,\,down}\ \Delta T + \delta_i(t - \Delta T)\}, \\ \overline{\delta}_i(t) &= \min\{\delta_{i,\,max},\ \rho_{i,\,up}\ \Delta T + \delta_i(t - \Delta T)\}, \end{aligned} \tag{4.54}$$

where $\delta_{i,max}$, $\delta_{i,min}$ are actuator i position limits, $\rho_{i,up}$ and $\rho_{i,down}$ are the actuator i rate limits, and T_s is the sampling time of the digital control system.

4.12.2 Benefits of the Supervision Module for Control Allocation

The control allocator is responsible for distributing the desired moments among a set of actuators. Therefore, if the control allocator is provided with the information that a faulty control surface is deflected at a certain angle, then the control allocator can generate appropriate commands to the unfailed actuators in an optimal manner, *i.e.*, by taking advantage of the torque already generated by the faulty actuator(s). A method to indicate to the control allocator which actuator has failed and at what position is to have the supervision module merge the upper and lower deflection limits $\delta_{i,max}$, $\delta_{i,min}$ of the failed actuator i and make these limits equal to the estimated deflection $\hat{\bar{\delta}}_i$ of actuator i computed by the corresponding EKF.

This method allows the faulty actuator to be frozen or floating at any position. Moreover, since $\hat{\bar{\delta}}_i$ is actually the estimate for the deflection angle of a control surface, this method is still valid for mechanical-link failure between an actuator (servo) and its corresponding control surface.

A complete description of this reconfigurable control allocator can be found in Chap. 5 of this book [26].

4.13 Computational Complexity of the EMMAE-FDI

Let us denote by N the number of actuators that are monitored by an FDI system. In the classical implementation of an MMAE-FDI, for each actuator k filters are to be designed for k different possible positions of the failed actuator. Therefore, $kN+1$ KFs are required ($+1$ refers to the no-fault filter). If the appearance of a second fault is to be checked as well, a new bank of KFs has to be reloaded, based on the knowledge of the first fault that occurred. In total $Nk + 1 + (N-1)kN = N^2k + 1$ KFs must be designed.

One major advantage of the EMMAE-FDI method presented in this chapter over classical MMAE schemes is that it requires only one filter to completely monitor one actuator. Any possible actuator-fault scenario is taken into account by only one filter. Therefore, for the monitoring of a single actuator fault, only $N+1$ filters are required with the EMMAE-FDI method. For the monitoring of a second actuator fault with the EMMAE-FDI method, no other bank of filters has to be loaded. Indeed, if actuator i fails, it suffices to feed all the other filters with the estimate of the faulty control surface deflection $\hat{\bar{\delta}}_i$ instead of the input δ_i. Thus, again only $N+1$ filters are needed with the EMMAE-FDI system to detect and isolate a second fault after a first actuator fault has been introduced.

As a simple example, our UAV is equipped with five actuators. A classical MMAE scheme designed for three possible faulty deflections per actuator

requires sixteen filters, whereas the EMMAE method needs only six filters for any lock-in-place and floating actuator fault scenarios.

Finally, the choice of the state vector $\mathbf{x} = [p \quad q \quad r \quad \alpha \quad \beta \quad \hat{\delta}]^T$ minimizes the number of relevant state elements for the satisfactory operation of the filters, thus limiting the number of computations required for the EKFs. Indeed, in previous work by the author [8, 9], a first version of this EMMAE method needed up to nine states in the state vector to achieve satisfactory results.

4.14 Conclusions

The EMMAE-FDI algorithm, combined with an active supervision module, offers fast and accurate fault detection and isolation. Moreover, the addition of the estimation of the faulty control surface deflection in the state vector makes the method applicable for actuator failures such as frozen or floating at an arbitrary position. Only one filter is needed to monitor the health of a single actuator. The filters used in the EMMAE-FDI are EKFs, which provide nonlinear state estimations at any flight operating condition. An active FDI technique is developed, which generates appropriate artificial excitation of the aircraft when needed. An additional filtering stage for the fault-probability signals has been designed to enhance the robustness of the diagnosis, even in the event of severe wind turbulence. The whole system has been demonstrated using nonlinear simulations of a realistic flight scenario. The FDI system was shown to be capable of handling two simultaneous actuator failures without increasing the computational load. Finally, when a fault is clearly isolated, the faulty actuator deflection estimate can be advantageously used to modify on-line the settings of a control allocator, making the whole system suitable for flight control reconfiguration without any change in the initial controller and any additional actuator position sensor [7].

References

1. D. T. Magill. Optimal Adaptive Estimation of Sampled Stochastic Processes. *IEEE Transactions on Automatic Control*, 10(4):434–439, October 1965.
2. P. Eide and P. S. Maybeck. An MMAE Failure Detection System for the F-16. *IEEE Transactions on Aerospace and Electronic Systems*, 32(3):1125–1136, July 1996.
3. P. S. Maybeck. Multiple Model Adaptive Algorithms for Detecting and Compensating Sensor and Actuator/Surface Failures in Aircraft Flight Control Systems. *International Journal of Robust and Nonlinear Control*, 9(14):1051–1070, 1999.
4. P. S. Maybeck and R. D. Stevens. Reconfigurable Flight Control Via Multiple Model Adaptive Control Methods. *IEEE Transactions on Aerospace and Electronic Systems*, 27(3):470–479, May 1991.

5. L. Ni. *Fault-Tolerant Control of Unmanned Underwater Vehicles*. PhD thesis, VA Tech. Univ., Blacksburg, VA, 2001.
6. D. Theilliol, J. C. Ponsart, H. Noura, and L. G. Vela Valdes. A Multiple Model Based Approach for Sensor Fault-Tolerant Control of Nonlinear Systems. In *Proceedings of the Congreso Latinoamericano de Control Automatico*, October 2004.
7. G. Ducard and H. P. Geering. Efficient Nonlinear Actuator Fault Detection and Isolation System for Unmanned Aerial Vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 31(1):225–237, January-February 2008.
8. G. Ducard and H. P. Geering. A Reconfigurable Flight Control System based on the EMMAE Method. In *Proceedings of the IEEE American Control Conference*, pages 5499–5504, Minneapolis, MN, June 2006.
9. D. Rupp, G. Ducard, H. P. Geering, and E. Shafai. Extended Multiple Model Adaptive Estimation for the Detection of Sensor and Actuator Faults. In *Proceedings of IEEE Control and Decision Conference, and European Control Conference*, pages 3079–3084, Seville, Spain, 2005.
10. G. Tao, S. Chen, X. Tang, and S. M. Joshi. *Adaptive Control of Systems with Actuator Failures*. Springer-Verlag, London Berlin Heidelberg, 2004.
11. M. Möckli. *Guidance and Control for Aerobatic Maneuvers of an Unmanned Airplane*. PhD thesis, ETH Zurich, 2006. Diss No. 16586.
12. B. Stevens and F. Lewis. *Aircraft Control and Simulation, Second Edition*. Wiley, New York, NY, 2003.
13. R. F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.
14. G. J. J. Ducard. *Fault-tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle*. PhD thesis, ETH Zürich, 2007. Diss. No. 17505.
15. R. E. Kalman. A New Approach to Linear Filtering and Prediction Problems. *Journal of Basic Engineering*, 82(1):35–46, March 1960.
16. P. S. Maybeck. *Stochastic Models, Estimation, and Control, Volume 1*. Academic Press, New York, Inc, 1979; republished by Navtech, Arlington, VA, 1994.
17. P. Zarchan and H. Musoff. *Fundamentals of Kalman Filtering: A Practical Approach, Second Edition* . Volume 208, Progress in Astronautics and Aeronautics, AIAA Inc., Reston, VA, 2005.
18. R. G. Brown and P. Y. C. Hwang. *Introduction to Random Signals and Applied Kalman Filtering*. John Wiley & Sons, New York, 1997.
19. R. Isermann. *Fault-Diagnosis Systems, An Introduction from Fault Detection to Fault Tolerance*. Springer-Verlag, Berlin Heidelberg, 2006.
20. E. Morelli. Flight Test Validation of Optimal Input Design and Comparison to Conventional Inputs. In *Proceedings of the Atmospheric Flight Mechanics Conference*, New Orleans, LA, 1997.
21. J. Buffington, P. Chandler, and M. Pachter. Integration of On-line System Identification and Optimization-based Control Allocation. 1998. AIAA paper 1998-4487.
22. D. B. Doman and A. D. Ngo. Dynamic Inversion-Based Adaptive/Reconfigurable Control of the X-33 on Ascent. *AIAA Journal of Guidance, Control, and Dynamics*, 25(2):275–284, 2002.
23. M. Azam, K. Pattipati, J. Allanach, S. Poll, and A. Petterson-Hine. In-Flight Fault Detection and Isolation in Aircraft Flight Control Systems. In *Proceedings of IEEE Aerospace Conference*, 2005. Paper 1429.
24. J. M. Maciejowski. Reconfiguring Control Systems by Optimization. In *Proceedings of the European Control Conference*, July 1997.
25. S. Fekri, M. Athans, and A. Pascoal. Issues, Progress and New Results in Robust Adaptive Control. *International Journal of Adaptive Control and Signal Processing*, 20(10):519–579, 2006.

26. G. Ducard, H. P. Geering, and E. Dumitrescu. Efficient Control Allocation for Fault Tolerant Embedded Systems on Small Autonomous Aircrafts. In *Proceedings of the 1st IEEE Symposium on Industrial Embedded Systems*, pages 1–10, Antibes, Juan les pins, France, October 2006.

Chapter 5
Control Allocation

This chapter describes the design of a control allocation module with explicit laws for fast operation and low computational load, such that this algorithm can run in a small processor or microcontroller with limited floating-point operation capability. The control allocation method is capable of compensating for actuator faults. Given the appropriate fault detection system, there is no need to redesign the controller when such faults occur, since the control allocator compensates for the fault. The allocation method is also designed to be reconfigurable based on the results obtained from the EMMAE-FDI system presented in Chap. 4. Finally, this chapter terminates with a comparison, which shows that this method yields satisfactory results, provides optimal solutions in some cases, and is simpler and faster than conventional methods.

5.1 Introduction to Control Allocation

In flight control, the proposed control allocation is used to compute control surface deflections in order to produce certain desired aerodynamic moments in roll, pitch, and yaw. The greatest benefit of this approach is achieved in over-actuated systems. Using control allocation, the design of the control system can be separated into the derivation of the control laws and the design of a control allocator. This approach offers the following three advantages:

- The actuator constraints, such as position and rate limits, can be taken into account. If one actuator is saturated, the remaining actuators can be used to produce the desired control effort.
- Control allocation takes advantage of the system's redundancy and allows the system to be optimized for certain objectives. For instance in a plane, these objectives may be the minimization of the drag, wing load, total control surface deflections, *etc.* [1, 2].

- In cases of actuator failures, a supervision controller can reconfigure the behavior of the control allocator in order to compensate for those failures, without the need for redesigning the control laws.

Several methods have been described in the literature: direct control allocation [3], daisy chaining [4], and the linear programming method [5]. The main technique these methods have in common is solving a constrained optimization problem. The pseudoinverse redistribution method [6, 7] is another technique, which makes use of a pseudoinverse computation of the control input $\boldsymbol{B}$ matrix. Although it does not always provide an optimal solution, it is usually faster than the other methods.

This chapter presents a method which does not have to solve any optimization problem and does not need to compute any pseudoinverse matrix. It is based on a set of explicit laws to drive the actuators depending on the fault situation. The appropriate law is selected by a supervision module, which uses data from a FDI module; see Chap. 4.

The control allocation algorithm presented in this chapter is designed for a small unmanned airplane, with five control surfaces producing the torques around the roll, pitch, and yaw axes. The method has been designed for fast operation and very low computational load, suitable for a small processor or microcontroller with limited floating-point operation capability [8]. Finally, the performance of this method is compared with that of a sequential least squares (SLS) control allocation method [9].

5.2 Reconfigurable Flight Control System

As described in Chap. 3 and shown in Fig. 5.1, the aircraft is equipped with five control surfaces, which are one left aileron, one right aileron, one left elevator, one right elevator, and one rudder. All actuators are fully independent, which means that ailerons (or elevators) can individually move up, down, or together in the same direction. This configuration permits some pitch torque to be produced with ailerons or some roll torque to be produced with elevators. The flight control system shown in Fig. 5.2 is a modified version of the one shown in Fig. 4.10. In Chap. 4, the supervision module (Supervisor) is responsible for the generation of appropriate excitation signals for fault detection purposes [10].

In this chapter, the supervisor is additionally made responsible for the reconfiguration of the control allocation module. Based on the results from the EMMAE-FDI system, the supervisor is to decide what is the health status of the aircraft, and in particular, what fault situation occurs. Based thereon, the supervisor selects an appropriate actuator behavior mode that the control allocator should use for the control of ailerons and elevators. These modes are detailed in the next section. Additionally, the supervisor passes to the control allocator the estimate(s) of the position of the failed actuator(s).

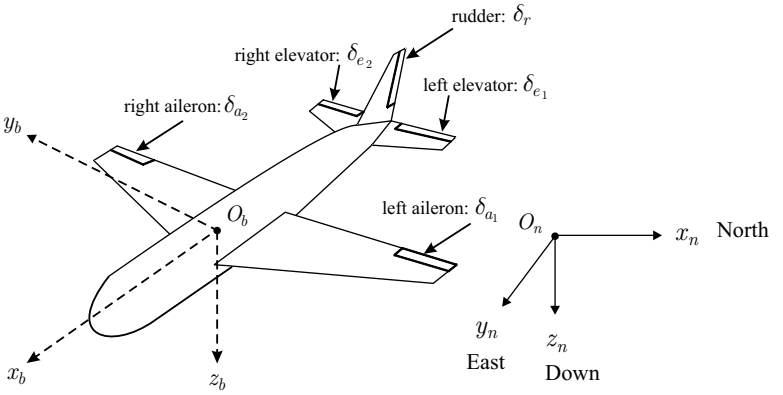


Fig. 5.1 Aircraft configuration

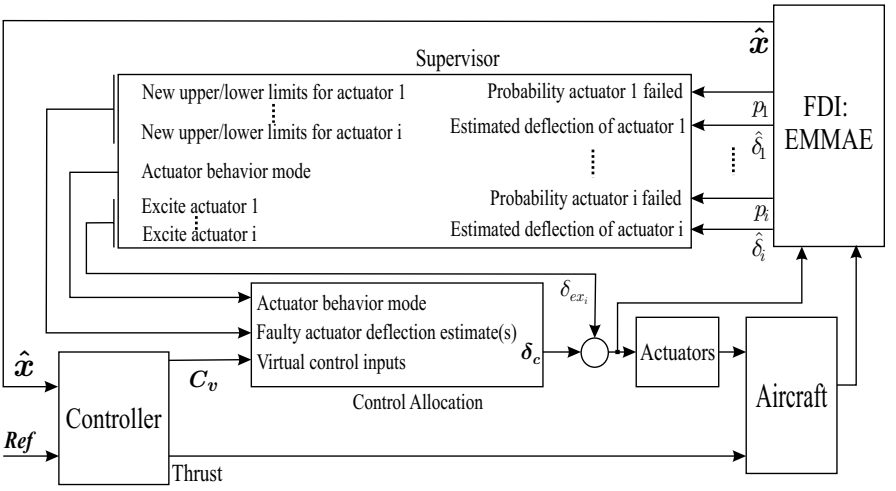


Fig. 5.2 Control allocation module in a reconfigurable flight control system

The flight controller generates a virtual control command vector $\boldsymbol{C_v} = [C_L\ C_M\ C_N]^T$ in terms of the desired dimensionless aerodynamic coefficients for roll, pitch, and yaw torques. This virtual command $\boldsymbol{C_v}$ is passed to the control allocator, which knows each actuator's position limits and effectiveness to produce torque components of the $\boldsymbol{C_v}$ signal.

In conventional methods [6, 9], the control allocator solves the following (possibly underdetermined) constrained system of equations, which may be regarded as a mapping in the controlled system $\boldsymbol{g}(\boldsymbol{\delta}(t)) = \boldsymbol{C_v}(t)$, with the true actuator control signals $\boldsymbol{\delta}(t) \in \mathbf{R}^N$ and N being the number of control surfaces. After linearization, the mapping equation may be rewritten in the standard formulation of the constrained linear control allocation problem:

$$\begin{aligned} \boldsymbol{B}\boldsymbol{\delta}(t) &= \boldsymbol{C}_{\boldsymbol{v}}(t), \\ \underline{\delta}_i(t) \leq \delta_i(t) &\leq \overline{\delta}_i(t), \end{aligned} \tag{5.1}$$

with the constraints

$$\begin{aligned} \underline{\delta}_i(t) &= \max\{\delta_{i,\,min},\ \rho_{i,\,down}\ \Delta T + \delta_i(t - \Delta T)\}, \\ \overline{\delta}_i(t) &= \min\{\delta_{i,\,max},\ \rho_{i,\,up}\ \Delta T + \delta_i(t - \Delta T)\}, \end{aligned} \tag{5.2}$$

where $\delta_{i,max}$, $\delta_{i,min}$ are the i^{th} actuator position limits, $\rho_{i,up}$ and $\rho_{i,down}$ are the i^{th} actuator rate limits (see Fig. 5.3), and T_s is the sampling time of the digital control system. Note that in the context of this work, actuators' motion rates are not considered.

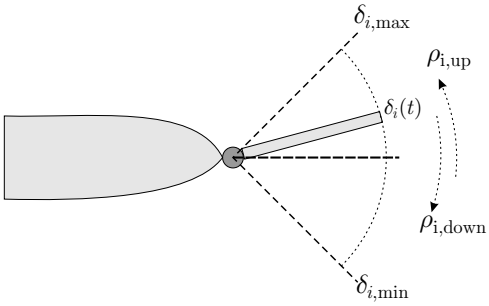


Fig. 5.3 Control surface deflection, motion, and rate limits

The convention used here for ailerons and elevators is positive deflection when the control surface is up and negative deflection when the control surface is down. For the rudder, positive deflection occurs when the rudder deflects to the right, when looking at the aircraft from the back, Table 5.1.

Table 5.1 Control surface sign conventions

Actuator	Deflection	Sense	Primary effect
Ailerons	Right wing trailing edge up	+	Positive roll moment
Elevators	Trailing edge up	+	Positive pitch moment
Rudder	Trailing edge right	+	Positive yaw moment

For convenience the relevant equations of aerodynamic moments needed for control allocation are recalled below (see Chap. 3):

$$
\begin{aligned}
L &= \bar{q}SbC_L(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \tilde{p}, \tilde{r}, \beta) \ , \\
M &= \bar{q}S\bar{c}C_M(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \tilde{q}, \alpha) \ , \\
N &= \bar{q}SbC_N(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \delta_r, \tilde{r}, \beta) \ ,
\end{aligned} \tag{5.3}
$$

where the control-surface deflection vector is $\boldsymbol{\delta} = [\delta_{a1} \ \delta_{a2} \ \delta_{e1} \ \delta_{e2} \ \delta_r]^T$ and $\bar{q} = \frac{\rho V_T^2}{2}$ is the dynamic pressure, V_T is the total airspeed of the aircraft, ρ is the air density, S is the wing total surface, b is the wing span, and $\bar{c}$ is the mean aerodynamic wing chord.

The aerodynamic derivatives are expressed as a linear combination of the state elements and control inputs as

$$
\begin{aligned}
C_L &= C_{L_{a1}}\delta_{a1} + C_{L_{a2}}\delta_{a2} + C_{L_{e1}}\delta_{e1} + C_{L_{e2}}\delta_{e2} + C_{L\tilde{p}}\tilde{p} + C_{L\tilde{r}}\tilde{r} + C_{L\beta}\beta \ , \\
C_M &= C_{M_{a1}}\delta_{a1} + C_{M_{a2}}\delta_{a2} + C_{M_{e1}}\delta_{e1} + C_{M_{e2}}\delta_{e2} + C_{L\tilde{q}}\tilde{q} + C_{L\alpha}\alpha \ , \\
C_N &= C_{N_{\delta_r}}\delta_r + C_{N\tilde{r}}\tilde{r} + C_{N\beta}\beta \ ,
\end{aligned} \tag{5.4}
$$

with

$$
\tilde{p} = \frac{bp}{2V_T} \ , \qquad \tilde{q} = \frac{\bar{c}q}{2V_T} \ , \qquad \tilde{r} = \frac{br}{2V_T} \ . \tag{5.5}
$$

The inertia matrix is expressed in the aircraft's body axes as follows:

$$
\boldsymbol{I}^b = \begin{bmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{zx} & 0 & I_{zz} \end{bmatrix} . \tag{5.6}
$$

Equation 5.4 shows that the dimensionless torques (C_L, C_M, C_N) linearly depend on elements of the state vector $[p \ q \ r \ \alpha \ \beta]^T$ and on elements of the control input vector $\boldsymbol{\delta}$ as follows:

$$
[C_L \quad C_M \quad C_N]^T = [C_{L,M,N}]_x \ \boldsymbol{x} + [C_{L,M,N}]_\delta \ \boldsymbol{\delta}. \tag{5.7}
$$

The rudder produces torque along the aircraft z-body axis, together with the ailerons when they do not generate the same aerodynamic drag. This effect is known as adverse yaw [11, 12] and is modeled by the term $C_{Ndrag}(\delta_{a2}) - C_{Ndrag}(\delta_{a1})$ in (5.8). We will not consider the adverse yaw effect from elevators when they are used with opposite deflections.

The virtual command issued by the controller for torque generation is

$$
\boldsymbol{C_v} = \begin{pmatrix} C_L \\ C_M \\ C_N \end{pmatrix} = \begin{pmatrix} C_{L_{a1}}\delta_{a1} + C_{L_{a2}}\delta_{a2} + C_{L_{e1}}\delta_{e1} + C_{L_{e2}}\delta_{e2} \\ C_{M_{a1}}\delta_{a1} + C_{M_{a2}}\delta_{a2} + C_{M_{e1}}\delta_{e1} + C_{M_{e2}}\delta_{e2} \\ C_{N_{\delta_r}}\delta_r + C_{Ndrag}(\delta_{a2}) - C_{Ndrag}(\delta_{a1}) \end{pmatrix} . \tag{5.8}
$$

The virtual command is rewritten as

$$
\boldsymbol{C_v} = [C_{L,M,N}]_\delta \ \boldsymbol{\delta}, \tag{5.9}
$$

with

$$[C_{L,M,N}]_\delta = \begin{bmatrix} C_{L_{a1}} & C_{L_{a2}} & C_{L_{e1}} & C_{L_{e2}} & 0 & 0 \\ C_{M_{a1}} & C_{M_{a2}} & C_{M_{e1}} & C_{M_{e2}} & 0 & 0 \\ C_{N_{drag,\delta_a}} & C_{N_{drag,\delta_a}} & 0 & 0 & C_{N_{\delta_r}} & 0 \end{bmatrix},$$
$$\boldsymbol{\delta} = [\delta_{a1} \quad \delta_{a2} \quad \delta_{e1} \quad \delta_{e2} \quad \delta_r \quad \delta_{\text{Thrust}}]^T. \tag{5.10}$$

Given $\boldsymbol{C_v}$ and $[C_{L,M,N}]_\delta$, the problem of control allocation is to find the control input vector $\boldsymbol{\delta}$. The system in (5.9) has three equations and five unknowns. In order to solve this system for a unique value of a control input vector $\boldsymbol{\delta}$, modes for the behavior of the actuators are set according to the fault situation. The fault configurations are shown in Table 5.2.

Table 5.2 Fault configurations

Aileron 1	Aileron 2	Elevator 1	Elevator 2	Actuator Behavior Mode	Equation
				0	see Fig. 5.5
X				1	see (5.16)
	X			2	see (5.16)[a]
		X		3	see (5.18)
			X	4	see (5.18)[a]
X	X			5	see (5.18)
		X	X	5	see (5.18)
X		X		5	see (5.18)
X			X	5	see (5.18)
	X	X		5	see (5.18)
	X		X	5	see (5.18)
X	X	X		6	see (5.18)[b]
	X	X	X	6	see (5.18)[b]
X	X		X	6	see (5.18)[b]
X		X	X	6	see (5.18)[b]
X	X	X	X	7	

[a] modified accordingly
[b] or (5.18)

5.3 Behavior Mode of Ailerons and Elevators

5.3.1 Nominal Mode: Mode 0

In the nominal mode, the system does not experience any fault. When the ailerons are deflected, they not only alter the lift being produced at each wing, but the drag as well. The aileron that deflects up ($\delta_{aileron} > 0$) produces less drag than the one deflecting down; see Fig. 5.4. The difference in the drag between the two ailerons will generate some yaw torque. The most common way to overcome adverse yaw is to use differential ailerons. This consists of deflecting the up-going aileron to a larger angle than the down-going one. Figure 5.4 shows a case where aileron 2 deflects to a larger angle than aileron 1, so that they both generate the same drag.

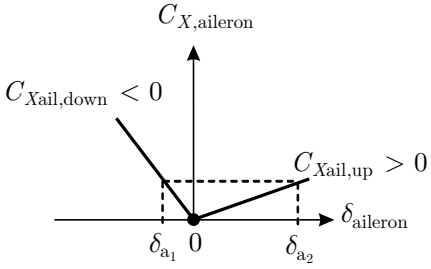


Fig. 5.4 Drag coefficients for an aileron deflecting up or down

In the nominal mode the ailerons are driven differentially according to (5.11):

$$\delta_{a_1} = -\gamma \cdot \delta_{a_2}, \tag{5.11}$$

with

$$\gamma = \left(\frac{C_{X_{\text{ail,up}}}}{|C_{X_{\text{ail,down}}}|} \right)^{\text{sign}(\delta_{\text{a}_2})}, \tag{5.12}$$

and $\text{sign}(\delta_{\text{a}_2})$ defined in Fig. 5.5. Given the conventions shown in Table 5.1, the sign of each aerodynamic coefficient in (5.8) can be explicitly written down as follows:

$$\begin{aligned}
&C_{L_a} = C_{L_{a2}} = -C_{L_{a1}} > 0\,,\\
&C_{L_e} = C_{L_{e2}} = -C_{L_{e1}} > 0\,,\\
&C_{M_a} = C_{M_{a2}} = C_{M_{a1}} > 0\,,\\
&C_{M_e} = C_{M_{e2}} = C_{M_{e1}} > 0\,,\\
&C_{N_{\delta_r}} > 0\,.
\end{aligned} \tag{5.13}$$

In this nominal mode, given (5.11) and (5.12), no adverse yaw torque is generated and the system for torque generation may be reduced as follows:

$$\begin{aligned} C_L &= C_{L_a}(\delta_{a_2} - \delta_{a_1}) + C_{L_e}(\delta_{e_2} - \delta_{e_1}) \,, \\ C_M &= C_{M_a}(\delta_{a_2} + \delta_{a_1}) + C_{M_e}(\delta_{e_2} + \delta_{e_1}) \,, \\ C_N &= C_{N_{\delta_r}} \delta_r \,. \end{aligned} \tag{5.14}$$

The system in (5.14) has three equations with five unknowns. In order to obtain a unique solution for this system, the number of unknowns must be reduced to three. In the nominal case, we thus set the following two behavior modes for actuators:

$$\begin{aligned} \delta_{a_1} &= -\gamma \cdot \delta_{a_2}, \\ \delta_{e_1} &= \delta_{e_2} \,. \end{aligned} \tag{5.15}$$

Using the conditions of (5.15), Eq. 5.14 now has three unknowns and can be solved uniquely. Figure 5.5 shows the finite state algorithm that computes control allocation in the nominal mode.

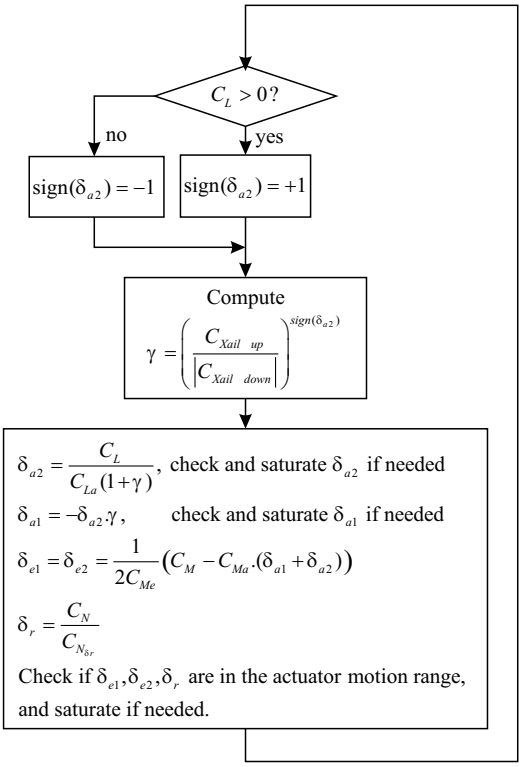


Fig. 5.5 Control allocation in the nominal mode (no fault), mode 0

5.3.2 Single Actuator Fault Modes: Modes 1 to 4

The FDI system is either based on the EMMAE method [10] which returns the estimated position of a faulty actuator (control surface), or on a direct measurement of the deflections of the control surfaces. Therefore, the deflection of the failed control surface is known. This reduces the number of unknowns in (5.14) by one.

A predefined rule for compensating this failure is then used to further reduce the system to three unknowns for three equations. In this way, we obtain a unique solution of the system of equations in (5.14). The goal is always to provide the control allocator with an analytic and explicit solution to properly drive the healthy actuators, depending on the fault configuration.

5.3.2.1 Aileron Fault: Modes 1 and 2

If aileron 1 fails (mode 1), the FDI system will estimate or measure its corresponding faulty deflection $\delta_{\mathrm{a1,fault}}$. Aileron 2 is used to produce the roll torque required to stabilize the aircraft. The elevators are used to correct undesired pitch torque due to the ailerons, and they produce additional needed pitch torque according to the following equations:

$$
\begin{aligned}
&\delta_{a1,\mathrm{fault}} \text{ is given by the FDI system,}\\
&\delta_{a2} = \delta_{a1,\mathrm{fault}} - C_L/C_{La}\ ,\\
&\delta_{e1} = \delta_{e2} = \left(C_M - C_{M_a}(\delta_{a1,\mathrm{fault}} + \delta_{a2})\right)/(2C_{M_e})\ ,\\
&\delta_r = \left(C_N - \left[C_{Ndrag(\delta_{a2})} - C_{Ndrag(\delta_{a1})}\right]\right)/C_{N_{\delta_r}}\ .\\
&\text{Check if } \delta_{\mathrm{a2}},\ \delta_{\mathrm{e1}},\ \delta_{\mathrm{e2}},\ \delta_{\mathrm{r}} \text{ are in the actuator motion range, otherwise}\\
&\text{saturate.}
\end{aligned}
\tag{5.16}
$$

If δ_{a2} saturates then $\delta_{a2} = \delta_{a2,min}$ or $\delta_{a2,max}$, if δ_{a2} fails (mode 5) then $\delta_{a2} = \delta_{a2,\mathrm{fault}}$ given by the FDI system. In both cases another unknown in the system is removed. Only the two deflections for the elevators remain to be computed. Note that they can also generate additional roll motion until they themselves saturate, according to the following equations:

$$
\begin{aligned}
&\delta_{a1,\mathrm{fault}} \text{ is given by the FDI system,}\\
&\delta_{a2} = \delta_{a2,\mathrm{sat}} \text{ if } \delta_{\mathrm{a2}} \text{ saturates, or } \delta_{\mathrm{a2}} = \delta_{\mathrm{a2,fault}} \text{ if } \delta_{\mathrm{a2}} \text{ fails,}\\
&\theta_1 = \left(C_M - C_{M_a}(\delta_{a1} + \delta_{a2})\right)/C_{M_e}\ , \quad \theta_2 = \left(C_L - C_{L_a}(\delta_{a1} - \delta_{a2})\right)/C_{L_e}\ ,\\
&\delta_{e1} = (\theta_1 + \theta_2)/2\ , \qquad \delta_{e2} = (\theta_1 - \theta_2)/2\ ,\\
&\delta_r = \left(C_N - \left[C_{Ndrag(\delta_{a2})} - C_{Ndrag(\delta_{a1})}\right]\right)/C_{N_{\delta_r}}\ .\\
&\text{Check if } \delta_{\mathrm{e1}},\ \delta_{\mathrm{e2}},\ \delta_{\mathrm{r}} \text{ are in the actuator motion range, otherwise}\\
&\text{saturate.}
\end{aligned}
\tag{5.17}
$$

Similar results can be derived for the case when aileron 2 fails on its own (mode 2).

Remark:

- In practice, it is also possible to simplify the control law for the rudder, simply by driving the rudder with $\delta_r = C_N/C_{N_{\delta_r}}$.
- We will use this formulation in the rest of the chapter. Indeed, if adverse yaw effects occur due to aileron deflection, then the controller itself will compensate by modifying the torque command C_N accordingly.

5.3.2.2 Elevator Fault: Modes 3 and 4

Let us consider the case when elevator 1 is locked (mode 3). The locked elevator will generate some roll motion if the other elevator does not deflect with the same angle. If this undesired roll motion is compensated by having the other elevator deflect the same angle as the locked one, then some undesired pitch torque will be generated in turn, and it will be very hard to compensate for the undesired pitch motion by only acting on the ailerons. One solution is to compute first the deflection of the remaining elevator to produce the desired pitch moment. In a second step, we use differential ailerons to correct for the roll torque.

$$
\begin{aligned}
&\delta_{e1,\text{fault}} \text{ is given by the FDI system,}\\
&\delta_{e2} = C_M/C_{M_e} - \delta_{e1,\text{fault}}. \text{ Check and saturate } \delta_{e2} \text{ if needed.}\\
&\text{In the case } \delta_{e2} \text{ also fails (mode 5) } \delta_{e2} = \delta_{e2,\text{fault}}, \text{ given by the FDI system.}\\
&\delta_{a1} = -\delta_{a2} = (C_L - C_{L_e}(\delta_{e1} - \delta_{e2})),\\
&\delta_r = C_N/C_{N_{rudd}},\\
&\text{Check if } \delta_{a1},\ \delta_{a2},\ \delta_r \text{ are in the actuator motion range, otherwise}\\
&\text{saturate.}
\end{aligned}
\tag{5.18}
$$

Once again equivalent results can be derived for the case when elevator 2 fails (mode 4). If the actuators exceed their respective physical limitation, they are simply saturated.

5.3.2.3 Rudder Fault

If the aircraft is equipped with a redundant rudder, then the method is also applicable to compensate for rudder failure. The equation for C_N in (5.14) has to be modified to take into account both rudders. In the case of failure, the

supervision module would compensate a rudder fault, and thus an undesired yaw moment, by deflecting the remaining rudder.

5.4 Multiple Failures

5.4.1 Case of Two Simultaneous Failures: Mode 5

If the FDI system is capable of handling two simultaneous failures, and if the position of these two locked actuators is known, then two unknowns in (5.14) are removed and replaced by the faulty values. The remaining two actuator deflections can then be computed. However, we may face problems that cannot be solved any more. For example, if the two elevators are both locked up or down, then the induced pitch motion will never be compensated by the ailerons. In such a case, an emergency procedure similar to that of the following modes 6 and 7 has to be initiated.

5.4.2 More Than Two Simultaneous Failures: Modes 6 and 7

In the case of three simultaneous faults (mode 6), there is one unknown for two different equations. Therefore, the two equations cannot be solved at the same time, and it is very likely that the aircraft cannot be controlled anymore. In mode 7, all the control surfaces of the aircraft are failed. In these two modes, an emergency procedure has to be engaged to prevent the UAV from crashing. One possibility is to shut down the engine and deploy a parafoil. If the UAV is small and low cost and does not necessarily need to be recuperated, then a another possibility is to blow up the UAV when it is high enough to avoid a ground or obstacle collision.

5.5 Extensions of the Method

The method described in this chapter could be extended to a larger number of actuators. As an example, consider an aircraft that is equipped with two ailerons on the right wing and two ailerons on the left wing. One strategy is to only use the outer ailerons during normal operation. This situation corresponds to the one already described in Sect. 5.3.1.

If one of the outer ailerons fails, then the other outer aileron on the opposite wing is used to compensate for this failure, as described in Sect. 5.3.2. If the

latter actuator saturates or also fails, then the two inner ailerons (flaps) will be used. In that case, (5.14) will be augmented with two new terms in the equation for C_L. It then is easy to find explicitly the required deflections of the inner ailerons to generate the desired value of C_L.

5.6 Computational Load of the Method

In order to have an estimation of how many floating-point operations are required in the control allocator, Table 5.3 summarizes the number of operations involved for the cases nominal mode, single, and double faults. These cases are all simulated and presented in the next section. Clearly, the control allocator presented requires very few floating-point operations.

Table 5.3 Number of floating-point operations required for control allocation

Fault situation	... > ... or ... < ...	+, −	×	÷
Nominal mode, see Fig. 5.5	11	3	4	4
Aileron 1 fault, see (5.16)	8	5	2	3
Aileron 1 fault and saturation of aileron 2, see (5.18)	6	8	4	2
Aileron 1 + aileron 2 faults, see (5.18)	6	8	4	3
Elevator fault, see (5.18)	8	5	3	3
Elevator 1 + aileron 2 faults, see (5.18)	6	6	4	2

5.7 Simulation Results

The method described has been tested in MATLAB®/Simulink® in a closed-loop control architecture. The aircraft is in steady conditions flying straight and level. The input of the controller is a reference signal for roll, pitch, and yaw rates of $p_{ref} = q_{ref} = r_{ref} = 0\,[\text{deg/s}]$. The actuators are modeled as a first-order low-pass filter with the time constant $\tau = 0.05\,\text{s}$. In order to introduce some noise into the system, some white noise with a variance of $0.32\,\text{deg}^2$ has been added on the actuators. Actuators are saturated at $\pm 45\,\text{deg}$.

The control allocator implemented for comparison is based on the work by Härkegård [9]. That method solves a constrained optimization problem online using an SLS method to optimally generate control signals for actuators by minimizing overall actuator deflections. That method was also used for comparison purposes in the paper by Jin [7].

The tic toc command in MATLAB®/Simulink® has been used to compare execution times of both methods. Simulations have been run on an AMD Turion 64, 1.6 GHz processor.

As shown in Fig. 5.6, a sequence of consecutive faults is generated. During t = 2...6 s aileron 1 fails and is locked at −20 deg, for t = 10...15 s aileron 2 fails and is locked at +40 deg, for t = 20...25 s elevator 1 fails and is locked at −5 deg, and finally for t = 30...35 s elevator 2 fails and is locked at +15 deg. Finally, two cases of simultaneous failures are tested. For t = 40...45 s both ailerons are locked at +10 deg and −1 deg, respectively. For t = 50...55 s both elevators are locked at +5 deg and −10 deg, respectively.

In the plots presented, fastCA refers to the method introduced in this chapter, whereas SLS refers to the sequential least square method [9]. Fig. 5.6 shows the respective behaviors of the two methods. In the non-fault case, both methods generate exactly the same control signals, even if the reference signal for the turn rates is different from zero (a sine wave, for example). This means that fastCA generates an optimal solution in the non-fault case. Slightly different control signals are generated in fault cases. For example, during an aileron failure, fastCA only compensates for roll motion by acting on the remaining aileron, whereas SLS also makes use of elevators. In the case of an elevator fault, it is interesting to note that both methods generate the same control signals for the remaining elevator, but generate slightly different control signals for ailerons.

The second fault on aileron 2 that is introduced at t = 10 s is rather severe. Aileron 1 thus tries to compensate this fault to maintain the roll rate at zero. We see that aileron 1 briefly saturates, and both methods use differential elevators to help correcting for the roll motion. Finally, SLS allocation sometimes produces large peaks in the control signals when a fault is introduced or removed, especially when the reference signal is not zero, whereas such peaks never appeared with fastCA. Figure 5.7 shows the signals for roll, pitch, and yaw rates. They all remain close to zero, with the exception of the short peak that appears when the severe fault on aileron 2 is introduced. But for all of the other faults, they are almost unnoticeable in the turn rate measurement signals.

5.7.1 Impact of the Control Allocator on the Controller

Figure 5.8 shows the virtual control signal $\boldsymbol{C_v} = [C_L \; C_M \; C_N]^T$ produced by the controller and the influence of the control allocator on the controller. Here the FDI system is ideal, *i.e.*, it is infinitely fast and knows instantaneously which fault occurred, and the control allocator is also immediately reconfigured in the proper mode.

Aside from the severe fault introduced at t = 10 s causing the controller to generate a high command value for C_L as aileron 1 saturates, the faults

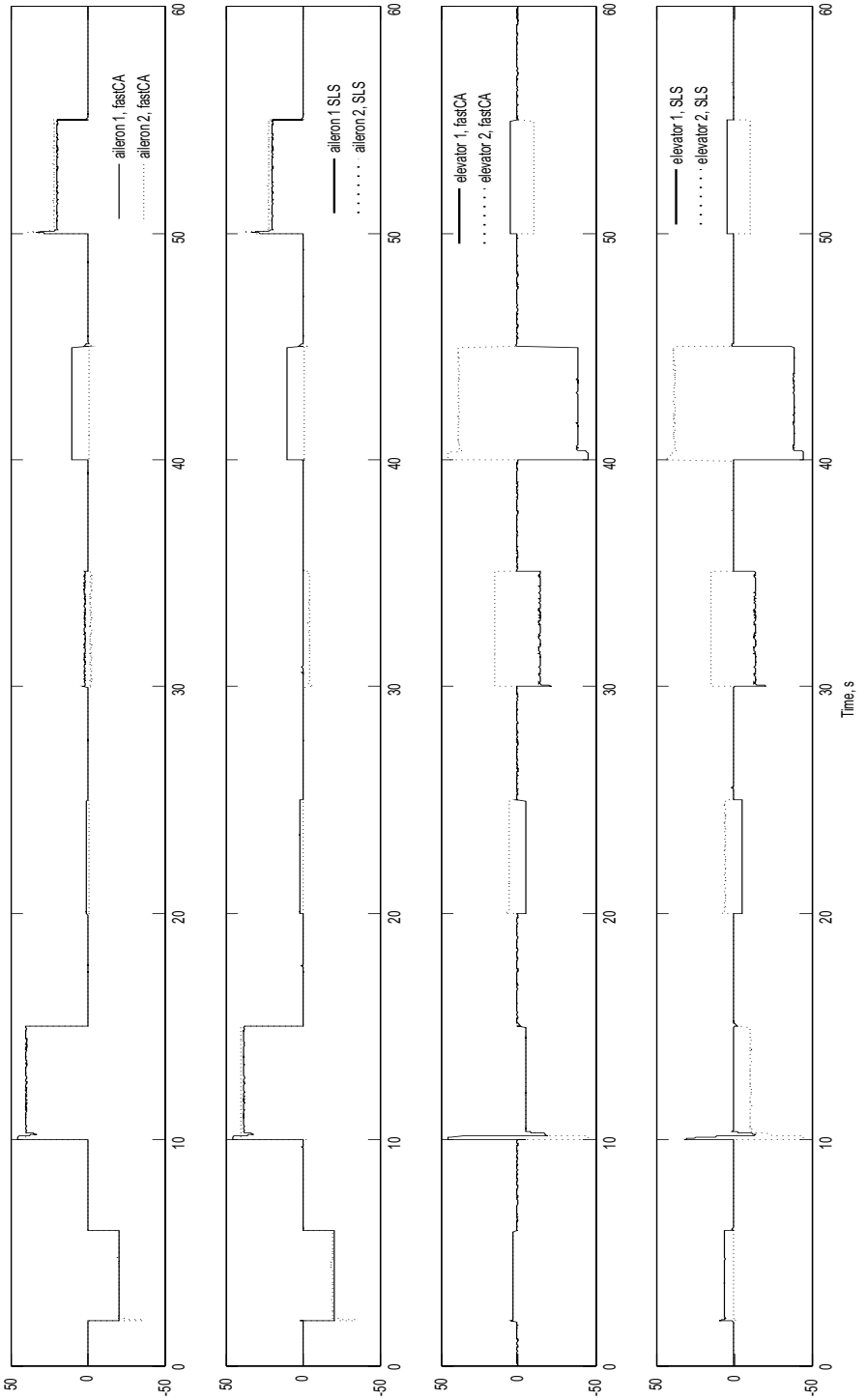


Fig. 5.6 Fault succession, comparison of behaviors between SLS and fastCA

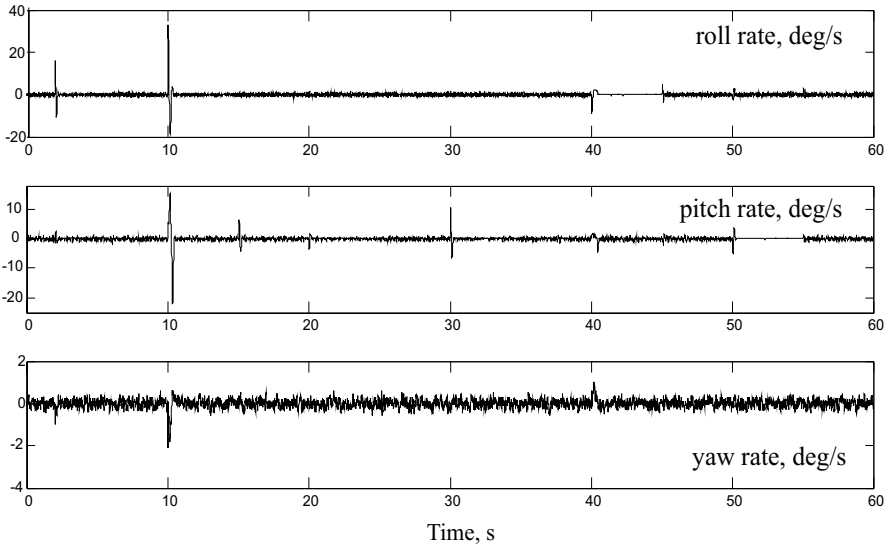


Fig. 5.7 Roll, pitch, and yaw rate during the succession of faults

have almost no influence on the controller. The controller thus acts as if there were no faults, due to the reconfigurable control allocator that autonomously compensates for the faults.

In the case of a realistic FDI system, the fault is isolated after some delay. During the fault isolation process, the control allocator is still in its nominal mode, and the aircraft flight controller itself tries to compensate for the fault by modifying the values of C_L, C_M, and C_N such that the reference signals for the turn rates are best satisfied. When the fault is isolated, the control allocator can be reconfigured in the most appropriate mode. Note that the flight controller on its own cannot drive, if needed, the elevators differentially or the ailerons both up or down. The flight controller only produces the values for C_L, C_M, and C_N, and does not modify the behavior of the actuators. Rather, this is done by the supervision module shown in Fig. 5.2, when the FDI results are available.

5.7.2 Comparison of Computational Effort for Control Allocation

The upper plot in Fig. 5.9 shows the time needed to compute the control allocation algorithm. The upper gray curve represents the SLS method, which needs about 1.3 ms, whereas the black line represents the fastCA method, which takes about 45 μs. The bottom plot in Fig. 5.9 shows the relative speed

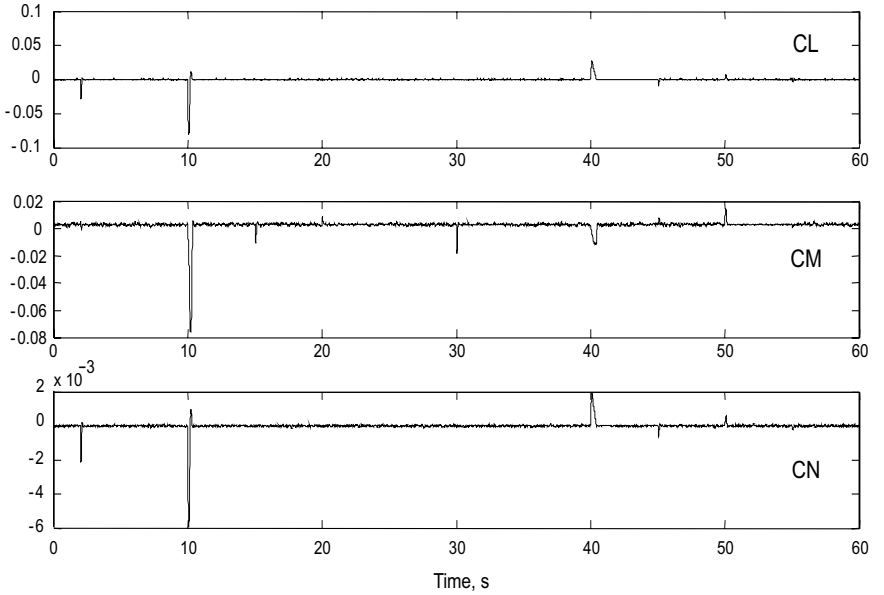


Fig. 5.8 Virtual control signal, $\boldsymbol{C_v} = [C_L\ C_M\ C_N]^T$

of the fastCA compared with the SLS method. During normal operation, on average, fastCA is 30 times faster. During the single-fault mode, fastCA is 25 times faster than SLS. The noise that appears in Fig. 5.9 seems to be due to the operating system running in the background of MATLAB®, which from time to time executes other tasks (outside MATLAB®) in-between the tic and toc commands.

5.8 Conclusions

The control allocation technique described in this chapter is an explicit set of allocation laws. Depending on the fault configuration, a supervision module has to select the appropriate law to distribute the commands. The results show that the behavior of this method is very similar to that of an optimization-based method, but it is much faster. Indeed, no optimization solver has to be run on-line, and no pseudoinverse matrix has to be computed. This method also requires very few floating-point operations and is deterministic, *i.e.*, we know precisely how much time the control allocation requires to compute its solution. The technique is therefore suitable to be implemented in a small processor or a microcontroller, where the computational power is rather limited and where the program has to run in a real-time environment.

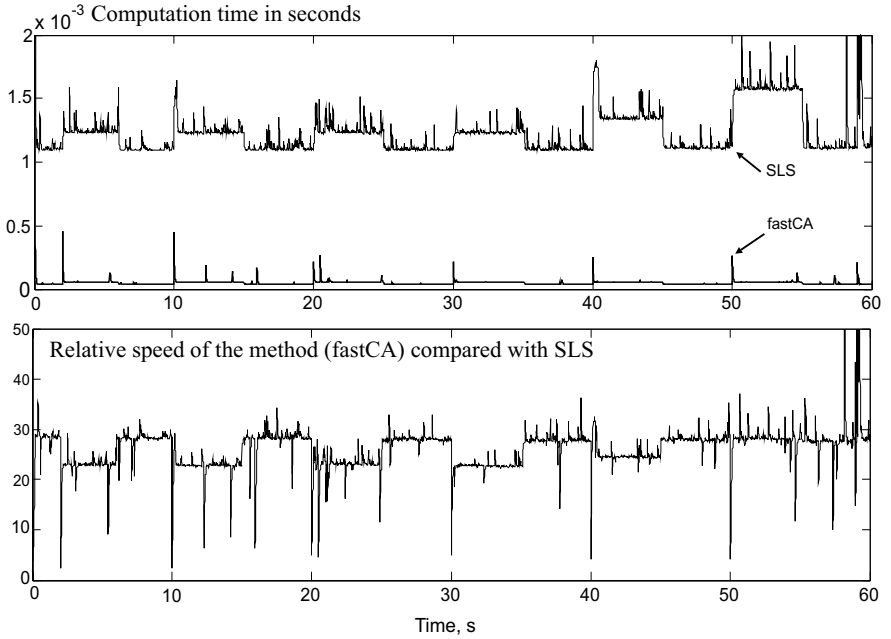


Fig. 5.9 Relative speed of SLS and fastCA

References

1. J. Buffington, P. Chandler, and M. Pachter. Integration of On-line System Identification and Optimization-based Control Allocation. 1998. AIAA paper 1998-4487.
2. J. Buffington, P. Chandler, and M. Pachter. On-Line Identification For Aircraft With Distributed Control Effectors. *AIAA Journal of Guidance, Control, and Dynamics*, 9:1033–1049, 1999.
3. W. Durham. Constrained Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 16(4):717–725, 1993.
4. J. Buffington and D. Enns. Lyapunov Stability Analysis of Daisy Chain Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 19(6):1226–1230, 1996. AIAA paper 1998-4487.
5. Y. Ikeda and M. Hood. An Approach of L1 Optimization to Control Allocation. August 2000. AIAA paper 2000-4566.
6. M. Bodson. Evaluation of Optimization Methods for Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 25(4):703–711, 2002.
7. J. Jin. Modified Pseudoinverse Redistribution Methods for Redundant Control Allocation. *AIAA Journal of Guidance, Control, and Dynamics*, 28(5):1076–1079, 2005.
8. G. Ducard, H. P. Geering, and E. Dumitrescu. Efficient Control Allocation for Fault Tolerant Embedded Systems on Small Autonomous Aircrafts. In *Proceedings of the 1st IEEE Symposium on Industrial Embedded Systems*, pages 1–10, Antibes, Juan les pins, France, October 2006.
9. O. Härkegård. *Backstepping and Control Allocation with Applications to Flight Control*. PhD thesis, Linköping University, Sweden, 2003.

10. G. Ducard and H. P. Geering. Efficient Nonlinear Actuator Fault Detection and Isolation System for Unmanned Aerial Vehicles. *AIAA Journal of Guidance, Control, and Dynamics*, 31(1):225–237, January-February 2008.
11. H. Sollman and S. Harris. *Mastering Instrument Flying, 3rd edition*. MacGraw-Hill Professional, 1999.
12. R. F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.

Chapter 6
Nonlinear Control Design

An aircraft is intrinsically a nonlinear system. Therefore, if linear controllers are to be used in the aircraft flight control system, several linear controllers have to be designed and then gain-scheduled over the operating regime of the aircraft. However, recent nonlinear control techniques have made it possible to deal directly with the known nonlinearities of the aircraft dynamics, which yields a unique controller suitable for a wide range of operating conditions. This chapter describes the technique known as NDI and presents the architecture and the design procedure of the controllers used in the aircraft autopilot.

6.1 Concept of Dynamic Inversion

6.1.1 Derivation of a Dynamic Inversion Controller

The usage of dynamic inversion control has been very popular over the past few years for the design of flight control systems [1–15].

Let the plant be described by a set of nonlinear differential equations of the form

$$\dot{\boldsymbol{x}}(t) = f(\boldsymbol{x}) + g(\boldsymbol{x})\boldsymbol{u}\,, \tag{6.1}$$

$$\boldsymbol{y}(t) = h(\boldsymbol{x})\,, \tag{6.2}$$

where the state vector is $\boldsymbol{x}(t) \in \mathbf{R}^n$ and the measurement vector is $\boldsymbol{y}(t) \in \mathbf{R}^p$. The system is assumed to be linear in the control input vector $\boldsymbol{u}(t) \in \mathbf{R}^m$. This assumption holds for an aircraft, the model of which is described in Chap. 3.

From (6.2), we obtain the following total derivative of the output $\boldsymbol{y}$ with respect to the time t:

$$\dot{\boldsymbol{y}} = \frac{\partial h}{\partial \boldsymbol{x}} \frac{d\boldsymbol{x}}{dt} = \frac{\partial h}{\partial \boldsymbol{x}} f(\boldsymbol{x}) + \frac{\partial h}{\partial \boldsymbol{x}} g(\boldsymbol{x})\boldsymbol{u} = \boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u} \; . \tag{6.3}$$

6.1.2 General Case

In order to have the output of the plant $\boldsymbol{y}(t)$ follow a desired trajectory, the signal for the desired output dynamics $\dot{\boldsymbol{y}}_{des}(t)$ is constructed and (6.3) is inverted in order to obtain the appropriate control signal $\boldsymbol{u}_c(t)$ as follows:

$$\boldsymbol{u}_c(t) = \boldsymbol{G}^{-1}(\boldsymbol{x})\Big(\dot{\boldsymbol{y}}_{des}(t) - \boldsymbol{F}(\boldsymbol{x})\Big) \; . \tag{6.4}$$

Now, the control design task is to build a suitable control signal for the desired output dynamics $\dot{\boldsymbol{y}}_{des}(t)$ as follows:

$$\dot{\boldsymbol{y}}_{des}(t) = \boldsymbol{K}\Big(\boldsymbol{y}_c(t), \boldsymbol{y}_{meas}(t)\Big) \; , \tag{6.5}$$

where the symbol $\boldsymbol{K}$ represents the controller, which has two inputs, namely the command signal $\boldsymbol{y}_c$ and the output measurement vector $\boldsymbol{y}_{meas}$, as shown in Fig. 6.1.

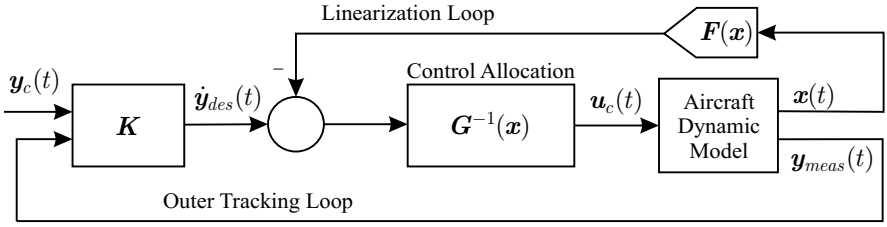


Fig. 6.1 NDI scheme

6.1.3 Formulation of the Signal for the Desired Output Dynamics $\dot{\boldsymbol{y}}_{des}(t)$

The signal for the desired dynamics $\dot{\boldsymbol{y}}_{des}(t)$ can be constructed with the following design steps [16].

First, the error signal is defined as

$$\boldsymbol{e}(t) = \boldsymbol{y}_c(t) - \boldsymbol{y}_{meas}(t) \; . \tag{6.6}$$

Second, the terms in (6.6) are differentiated with respect to time yielding

$$\begin{aligned} & \dot{\boldsymbol{e}}(t) = \dot{\boldsymbol{y}}_c(t) - \dot{\boldsymbol{y}}_{meas}(t) \ , \\ \Longleftrightarrow \ & \dot{\boldsymbol{y}}_{meas}(t) = \dot{\boldsymbol{y}}_c(t) - \dot{\boldsymbol{e}}(t) \ . \end{aligned} \tag{6.7}$$

Third, a controller $\boldsymbol{K}$ is designed to drive the error $\boldsymbol{e}(t)$ to zero, and to guarantee that the dynamic system in (6.8) is asymptotically stable:

$$\dot{\boldsymbol{e}}(t) = -\boldsymbol{K}\boldsymbol{e}(t) \ , \tag{6.8}$$

where all of the eigenvalues of the p by p matrix $\boldsymbol{K}$ have strictly positive real parts.

Fourth, if perfect plant dynamic inversion occurs, then the plant output follows the desired dynamics, *i.e.*, $\dot{\boldsymbol{y}}_{meas}(t) = \dot{\boldsymbol{y}}_{des}(t)$. Therefore, (6.7) is rewritten as follows:

$$\dot{\boldsymbol{y}}_{des}(t) = \dot{\boldsymbol{y}}_c(t) - \dot{\boldsymbol{e}}(t) \ . \tag{6.9}$$

The complete expression for the signal $\dot{\boldsymbol{y}}_{des}(t)$ is finally obtained by using (6.8) in (6.9) yielding

$$\dot{\boldsymbol{y}}_{des}(t) = \dot{\boldsymbol{y}}_c(t) + \boldsymbol{K} \cdot \boldsymbol{e}(t) \ . \tag{6.10}$$

6.2 Ideal or Perfect Dynamic Inversion

If there is no uncertainty in the plant model, *i.e.*, $\boldsymbol{F}(\boldsymbol{x})$ and $\boldsymbol{G}(\boldsymbol{x})$ are perfectly known and if actuators and sensors are also perfect, the dynamic inversion process is perfect or ideal. In such a case, dynamic inversion transforms the controlled system from $\dot{\boldsymbol{y}}_{des}(t)$ to $\boldsymbol{y}_{meas}(t)$ into a system of p parallel and uncoupled integrators.

In order to better illustrate how the pure integrators are actually created by perfect dynamic inversion, let us consider Fig. 6.2, which shows the effect of the additive inverse path and multiplicative inverse path on the dynamics of the plant.

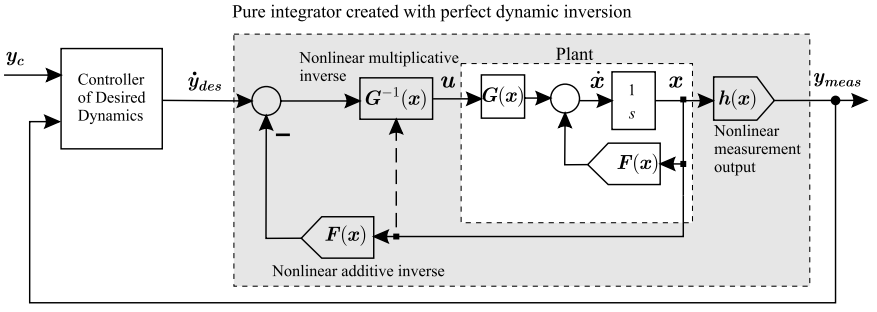


Fig. 6.2 Dynamic inversion with desired dynamics for a nonlinear plant

The plant output signal $\boldsymbol{y}_{meas} = \boldsymbol{h}(\boldsymbol{x})$ can be written as follows:

$$\begin{aligned}
\boldsymbol{y}_{meas} &= \frac{1}{s}\, \dot{\boldsymbol{y}}_{meas} \ , \\
&= \frac{1}{s}\, \left[\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\boldsymbol{u}_c\right] \ , \text{ see (6.3), with } \boldsymbol{u}_c \text{ defined in (6.4)} \ , \\
&= \frac{1}{s}\, \left[\boldsymbol{F}(\boldsymbol{x}) + \boldsymbol{G}(\boldsymbol{x})\Big(\boldsymbol{G}^{-1}(\boldsymbol{x})\Big(\dot{\boldsymbol{y}}_{des}(t) - \boldsymbol{F}(\boldsymbol{x})\Big)\Big)\right] \ , \\
&= \frac{1}{s}\boldsymbol{I}_p\, \dot{\boldsymbol{y}}_{des}(t) \ . \qquad (6.11)
\end{aligned}$$

Equation 6.11 clearly indicates that perfect dynamic inversion transforms the controlled system from $\dot{\boldsymbol{y}}_{des}(t)$ to $\boldsymbol{y}_{meas}(t)$ into p parallel integrators as shown in Fig. 6.3.

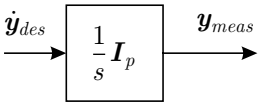


Fig. 6.3 Transfer function from $\dot{\boldsymbol{y}}_{des}$ to $\boldsymbol{y}_{meas}$ in the case of perfect dynamic inversion

Figure 6.4 shows the architecture of the dynamic inversion process in the linear case, where the linearized plant is written in the standard form

$$\begin{aligned}
\dot{\boldsymbol{x}}(t) &= \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) \ , \\
\boldsymbol{y}(t) &= \boldsymbol{C}\boldsymbol{x}(t) \ . \qquad (6.12)
\end{aligned}$$

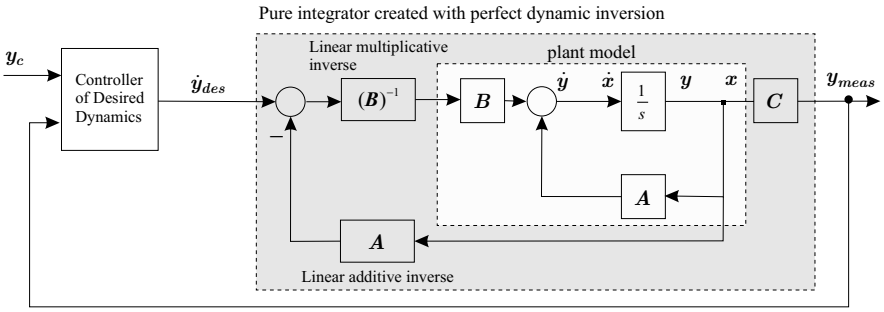


Fig. 6.4 Dynamic inversion with desired dynamics for a linear plant

6.3 Architecture of the Controller of Desired Dynamics

The architecture of the controller that parameterizes the desired dynamics $\dot{\boldsymbol{y}}_{des}(t)$ is based on a proportional and integral (PI) controller.

6.3.1 Selection of a PI Controller

There are several ways to construct PI or proportional, integral and derivative (PID) controllers for a plant with the transfer function $G_p(s)$. A classical configuration is the "parallel" PI controller shown in Fig. 6.5.

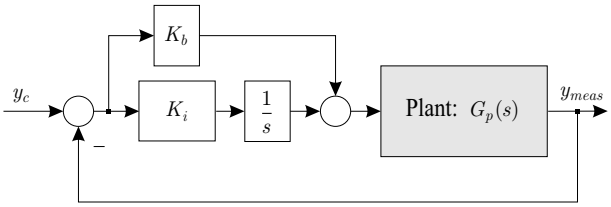


Fig. 6.5 Configuration of a classical parallel PI controller

The closed-loop transfer function of the PI controller and the plant G_p of Fig. 6.5 is

$$\frac{y_{meas}}{y_c} = \frac{(sK_b + K_i)G_p}{s + (sK_b + K_i)G_p} \,. \tag{6.13}$$

The PI controller has introduced a zero at $s = -K_i/K_b$, which is dependent on the controller parameters K_i and K_b. It is preferable to have a PI controller structure without a zero or with a zero that can be placed arbitrarily and can be used to cancel a slow pole in the closed-loop transfer function. Therefore, the previous PI controller is modified in order to obtain the PI-type controller shown in Fig. 6.6.

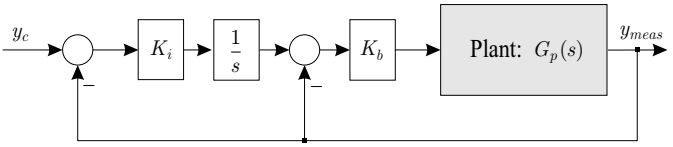


Fig. 6.6 Configuration of the modified PI controller

The closed-loop transfer function of the modified PI controller and the plant G_p is

$$\frac{y_{meas}}{y_c} = \frac{K_b K_i G_p}{s + K_b(s + K_i)G_p} \,. \tag{6.14}$$

Clearly, the transfer function in (6.14) has the same denominator as (6.13) but the closed-loop zero of the controller has been removed. The modified PI controller of Fig. 6.6 contains a pure integral controller of the error signal and an inner-loop proportional feedback, which can be considered as a differentiator-free rate-feedback control loop.

6.3.2 Feedforward of the Command Signal y_c

The control structure in Fig. 6.5 is called "one degree-of-freedom" because the controller is acting on a single signal, *i.e.*, the error signal $y_c - y_{meas}$. However, in order to improve the command-tracking performance of the controller, we treat the two signals y_c and y_{meas} independently by introducing a feedforward controller on the command signal y_c.

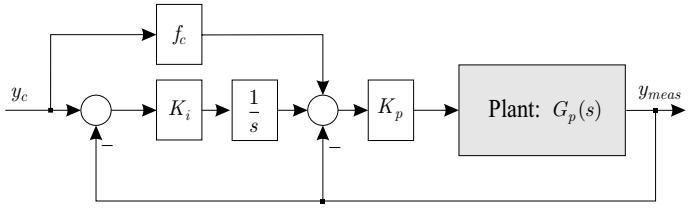


Fig. 6.7 Architecture of the modified PI controller with feedforward of the command signal

The resulting control system is depicted in Fig. 6.7 and has a so-called "two degree-of-freedom" control scheme, since the plant input is the sum of the contributions of the feedback and the feedforward controllers.

Adding the feedforward path of the command signal improves the performance at frequencies where feedback control is not effective (*i.e.*, at higher frequencies than the feedback controller corner frequency). The transfer function of the system shown in Fig. 6.7 is as follows:

$$\frac{y_{meas}(s)}{y_c(s)} = \frac{G_p K_b(f_c s + K_i)}{s + G_p K_b(s + K_i)} \,. \tag{6.15}$$

Equation 6.16 shows that the closed-loop transfer function has a zero, which can be placed at a desired location by modifying the value of the coefficient f_c. In this way, the zero can be chosen in order to cancel the smallest real pole of the closed-loop transfer function and hence improve the time response of the system to the command signals.

The controller that will be used in this project features:

- A proportional action with the gain K_b, which sets the bandwidth of the controller.
- An integral action with the gain $K_i = f_i K_b$, with the auxiliary integrator gain f_i as shown in Fig. 6.8. The reason why the coefficient f_i is introduced here will be clarified later in the chapter. The coefficient f_i is selected once and K_b becomes the single parameter determining the bandwidth of the closed-loop system.
- A command-feedforward gain f_c, which enhances the time response of the system to command signals.

The transfer function of the system shown in Fig. 6.7 is as follows:

$$\frac{y_{meas}(s)}{y_c(s)} = \frac{G_p K_b (f_c\, s + f_i\, K_b)}{s\,(1 + G_p\, K_b) + G_p\, f_i\, K_b^2}\,. \tag{6.16}$$

If we consider the plant $G_p(s)$ to be a pure integrator created by perfect dynamic inversion, then the closed-loop transfer function of the system shown in Fig. 6.8 is

$$\frac{y_{meas}(s)}{y_c(s)} = \frac{K_b(f_c s + f_i K_b)}{s^2 + K_b s + f_i K_b^2}\,. \tag{6.17}$$

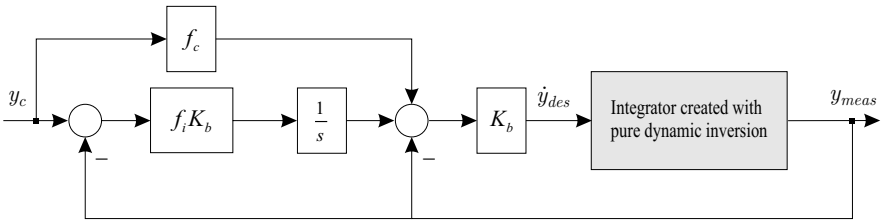


Fig. 6.8 Architecture of the modified PI controller with command feedforward

6.3.3 Open-loop Gain

In order to compute the open-loop gain, the inner-rate feedback loop is first closed as shown in Fig. 6.9, and the open-loop gain is computed as

$$L_e(s) = L_y(s) = \frac{G_p(s)}{1 + K_b\, G_p(s)} \frac{K_b^2 f_i}{s}\,. \tag{6.18}$$

Clearly, the open-loop transfer function is not dependent on the gain f_c.

If the plant $G_p(s)$ is a pure integrator, the open-loop gain is

$$L_e(s) = L_y(s) = \frac{1}{s + K_b} \frac{K_b^2 f_i}{s} , \tag{6.19}$$

where the gain K_b is used to set the bandwidth of the controller as shown in Fig. 6.10.

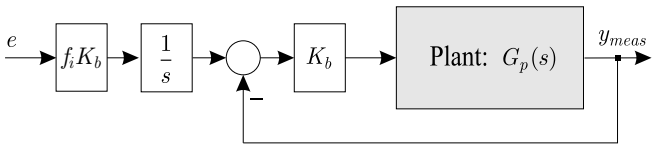


Fig. 6.9 Open-loop gain

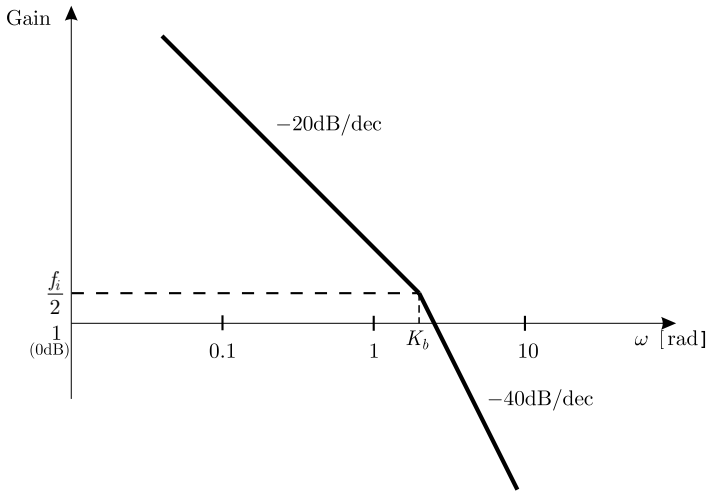


Fig. 6.10 Asymptotic Bode plot of the loop-transfer function $L_y(s)$ of (6.18)

6.3.4 Design Rules for the Command-feedforward Gain f_c

Equation 6.17 is of the form $\frac{K_{gain}(s+zero)}{(s+pole_1)(s+pole_2)}$. The gain f_c is to be selected in order to make the real zero of the transfer function cancel the smallest real pole. In this way, the closed-loop time constant is as small as possible.

The poles of (6.17) are the solutions of the equation $s^2 + K_b s + f_i K_b^2 = 0$. Let us define the auxiliary variable $\Delta = K_b^2(1 - 4f_i)$.

- If the integral gain $f_i < 0.25$, then $\Delta > 0$ and the two poles of (6.17) are real and equal to $pole_{1,2} = (-K_b \pm \sqrt{\Delta})/2$. In order to cancel the slowest pole by the zero, the condition $\frac{-f_i K_b}{f_c} = \frac{-K_b+\sqrt{\Delta}}{2}$ must be satisfied, which results in the choice for the command-feedforward gain f_c as follows [12, 13]:

$$f_c = (1 + \sqrt{1 - 4f_i})/2 \; . \tag{6.20}$$

- If the integral gain f_i is chosen larger than 0.25, then two complex conjugate poles $pole_{1,2} = (-K_b \pm j\sqrt{-\Delta})/2$ appear and f_c can be selected to make the real zero cancel the magnitude of the complex poles, *i.e.*, $\Big(K_b^2 + K_b^2(4f_i - 1)\Big)/4 = (f_i^2 K_b^2)/f_c^2$, which results in the following choice for f_c [12, 13]:

$$f_c = \sqrt{f_i} \; . \tag{6.21}$$

The denominator of (6.17) can be written in the form $s^2 + 2\omega_0\zeta s + \omega_0^2$ with $2\omega_0\zeta = K_b$ and $\omega_0^2 = f_i K_b^2$. An analytical value for the damping ratio is also found to be $\zeta = 1/(2\sqrt{f_i})$.

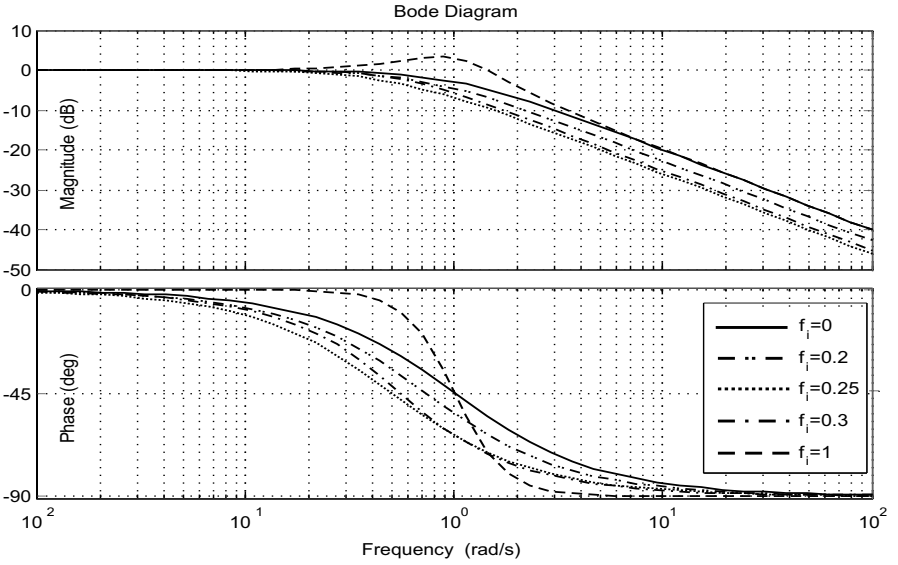


Fig. 6.11 Bode plots of the closed-loop transfer function y_{meas}/y_c for $K_b = 1$ and different values of f_i according to Table 6.1

Table 6.1 shows that for $f_i > 0.25$ complex conjugate poles appear as expected, and they can no longer be perfectly canceled by the zero. Therefore, the values for f_i will be chosen up to 0.25. In the flight control loops designed in the following chapters, the integral gain is chosen to be $f_i = 0.25$. This choice yields a value $f_c = 0.5$ according to (6.20), which leads to a "critically

damped" (closed-loop damping coefficient $\zeta_{cl} = 1$) system, whose transfer function is

$$\frac{y_{meas}(s)}{y_c(s)} = \frac{K_b/2}{s + K_b/2}\,. \tag{6.22}$$

Clearly, (6.22) indicates that the bandwidth of the closed-loop transfer function y_{meas}/y_c is given by the term $K_b/2$, which is a key parameter for the tuning of the controllers described in the following chapters.

Table 6.1 Summary of the control design parameters shaping the signal of desired dynamics $\dot{y}_{des}$

	$f_i = 0$	$f_i = 0.01$	$f_i = 0.2$	$f_i = 0.25$	$f_i = 0.3$	$f_i = 1$
K_b	1	1	1	1	1	1
f_c	1	0.99	0.724	0.5	0.5477	1
zero	0	-0.01	-0.276	-0.5	-0.5477	-1
$pole_1$	0	-0.01	-0.276	-0.5	$-0.5 + j0.2236$	$-0.5 + j0.8660$
$pole_2$	-1	-0.99	-0.724	-0.5	$-0.5 - j0.2236$	$-0.5 - j0.8660$
$\omega_{cl}{}^a$	1	0.99	0.72	0.5	0.55	1
ζ	∞	5	1.12	1	0.9129	0.5

[a] closed-loop bandwidth

6.3.5 Feedforward of the Rate of Change of the Command Signal $\dot{y}_c$

In Chap. 7, some controllers will be designed with a feedforward of the rate of change of the command signal $\dot{y}_c$ in order to increase the command-tracking performance of the system. Figure 6.12 shows the complete architecture of such a controller.

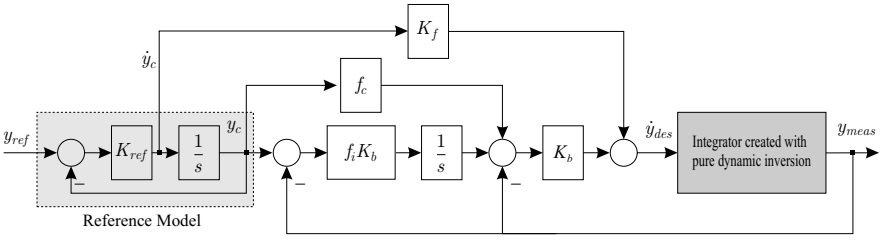


Fig. 6.12 Architecture of the PI controller with feedforward and reference model

One may notice that for $K_f = 1$, the signal $\dot{y}_{des}(t)$ is exactly constructed as shown in (6.10), *i.e.*, $\dot{y}_{des}(t) = \dot{y}_c(t) + K \cdot e(t)$.

The transfer function of the system in Fig. 6.12 from y_c to y_{meas} is as follows:

$$\frac{y_{meas}(s)}{y_c(s)} = \frac{G_p\left[K_f s^2 + K_b(f_c s + f_i K_b)\right]}{s + G_p(K_b s + f_i K_b^2)} \,. \tag{6.23}$$

If the plant $G_p(s)$ is a pure integrator obtained by the perfect dynamic inversion process, the transfer function $\frac{y_{meas}(s)}{y_c(s)}$ is as follows:

$$\frac{y_{meas}(s)}{y_c(s)} = \frac{K_f s^2 + K_b(f_c s + f_i K_b)}{s^2 + K_b s + f_i K_b^2} \,. \tag{6.24}$$

It is very interesting to note that for $K_f = 1$ and $f_c = 1$, the transfer function in (6.24) simplifies to $\frac{y_{meas}(s)}{y_c(s)} = 1$, and the open-loop gain $L_y(s)$ remains exactly the same as in (6.18) [10, 12]. This architecture will be particularly useful in the aircraft velocity-control loops.

6.3.6 Reference Model and Explicit Model Following

In Fig. 6.12, the reference signal y_{ref} passes through a *reference model* filter that generates the commanded control signal $y_c(t)$ and its time derivative $\dot{y}_c(t)$ used in the feedforward path. Inserting a reference model in the control system is also called *explicit model following* and makes the aircraft behave with certain "flying qualities" imposed by the reference model [3, 5, 9, 16, 17]. In this project, the explicit model-following technique is employed such that the closed-loop system under consideration exhibits first-order responses. The reference model has the following transfer function

$$\frac{y_c(s)}{y_{ref}(s)} = \frac{K_{ref}}{s + K_{ref}} \,. \tag{6.25}$$

For $K_f = 1$ and $f_c = 1$ in (6.24), we obtain

$$\frac{y_{meas}(s)}{y_{ref}(s)} = \frac{y_{meas}}{y_c} \cdot \frac{y_c}{y_{ref}} = \frac{K_{ref}}{s + K_{ref}} \,. \tag{6.26}$$

The latter equation means that if dynamic inversion operates properly, the system responds to the reference signal y_{ref} like a first-order system, with the coefficient K_{ref} defining its cut-off frequency.

6.3.7 Integrator Anti-windup

A common problem encountered with integral control of real systems is *integrator windup*. Saturation may occur on actuators or on internal states like a limited angle of attack, a limited rate of climb, *etc.* When saturation occurs, the integrator may begin to integrate a large error signal that takes its output far beyond the plant's control-input saturation levels. When the plant comes out of saturation, or when the command reverses, it may take a significant time before this excessive control output is removed and linear control is regained [16]. Figure 6.13 shows the implementation of the anti-windup loop.

For practical application, a suitable value for the anti-windup gain is to be selected. When the control signal exceeds the upper actuator limit $u > u_{max}$, the actuator saturates and its output stays at a constant value u_{sat}, and therefore $\Delta u = u_{sat} - u$ is a negative number, which is fed back to decrease the input of the integrator until Δu reaches zero again, thus yielding $u = u_{sat}$.

The control objective of the anti-windup loop is to drive the signal Δu in Fig. 6.14 to zero. Therefore, the loop should be stabilizing, which imposes the condition $K_a K_b > 0$. Moreover, Δu is the solution of the following differential equation:

$$\tau \dot{\Delta u} + \Delta u = 0\,. \tag{6.27}$$

Equation 6.27 corresponds to a first-order system with the time constant $\tau = 1/(K_b K_a)$. The larger the product $K_b K_a$, the faster the term Δu returns to zero. A reasonable number adopted in this work is $K_a = 3$.

Remark:

- A very interesting framework for the design of anti-windup system for NDI controllers can be found in the work by G. Herrmann *et al.* [18].

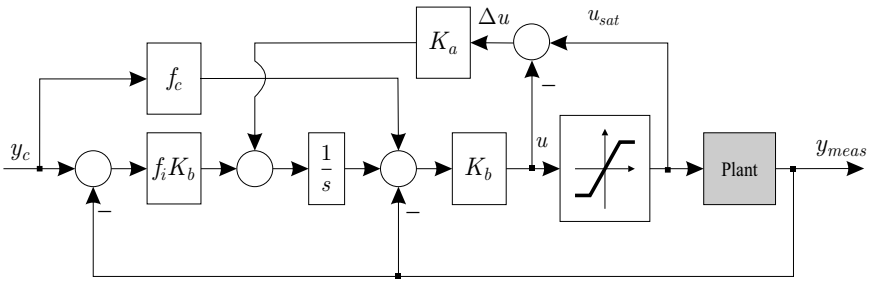


Fig. 6.13 Architecture of the integrator anti-windup system

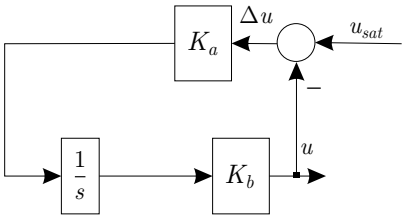


Fig. 6.14 Stability study of the integrator anti-windup loop

References

1. D. Enns. Robustness of Dynamic Inversion vs. μ Synthesis: Lateral Directional Flight Control Example. AIAA paper 90-3338-CP, 1990.
2. M. G. Goman and E. N. Kolesnikov. Robust Nonlinear Dynamic Inversion Method for an Aircraft Motion Control. Paper AIAA-98-4208, pp. 511−521, 1998.
3. K. Wise, J. Brinker, A. Calise, D. Enns, M. Elgersma, and P. Voulgaris. Direct Adaptive Reconfigurable Flight Control For A Tailless Advanced Fighter Aircraft. *International Journal of Robust Nonlinear Control*, 9:999–1012, 1999.
4. J. Buffington, P. Chandler, and M. Pachter. On-Line Identification For Aircraft With Distributed Control Effectors. *AIAA Journal of Guidance, Control, and Dynamics*, 9:1033–1049, 1999.
5. J. Brinker and K. A. Wise. Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August 2000.
6. M. B. McFarland and S. M. Hoque. Robustness of a Nonlinear Missile Autopilot Designed Using Dynamic Inversion. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, 14−17 August 2000. Paper AIAA-2000-3970.
7. D. Itô, D. T. Ward, and J. Valasek. Robust Dynamic Inversion Controller Design and Analysis for the X38. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, 6−9 August 2001. Paper 2001-4380.
8. D. B. Doman and A. D. Ngo. Dynamic Inversion-Based Adaptive/Reconfigurable Control of the X-33 on Ascent. *AIAA Journal of Guidance, Control, and Dynamics*, 25(2):275–284, 2002.
9. M. Bodson. Reconfigurable Nonlinear Autopilot. *AIAA Journal of Guidance, Control, and Dynamics*, 26(5):719–727, 2003.
10. J. D. Schierman, D. G Ward, J. R. Hull, N. Gandhi, M. W. Oppenheimer, and D. B. Doman. Integrated Adaptive Guidance and Control for Re-Entry Vehicles with Flight-Test Results. *AIAA Journal of Guidance, Control, and Dynamics*, 27(6):975–988, November-December 2004.
11. N. Tanaka, S. Suzuki, K. Masui, and H. Tomita. Restructurable Guidance and Control for Aircraft with Failures Considering Gusts Effects. *AIAA Journal of Guidance, Control, and Dynamics*, 29(3):635–642, May-June 2006.
12. D. Enns and T. Keviczky. Dynamic Inversion Based Flight Control for Autonomous RMAX Helicopter. In *Proceedings of the IEEE American Control Conference*, pages 3916–3923, Minneapolis, MN, June 2006.
13. D. Enns and G. Papageorgiou. Dynamic Inversion Tutorial - Theory and Examples of Applications. In *Workshop at the IEEE American Control Conference*, Minneapolis, MN, June 2006.

14. G. J. J. Ducard. *Fault-tolerant Flight Control and Guidance Systems for a Small Unmanned Aerial Vehicle*. PhD thesis, ETH Zürich, 2007. Diss. No. 17505.
15. Y. Shin, A. J. Calise, and M. D. Johnson. Adaptive Control of Advanced Fighter Aircraft in Nonlinear Flight Regimes. *AIAA Journal of Guidance, Control, and Dynamics*, 31(5):1464–1477, September-October 2008.
16. B. Stevens and F. Lewis. *Aircraft Control and Simulation, Second Edition*. Wiley, New York, NY, 2003.
17. J.-F. Magni, S. Bennani, and J. Terlouw. *Robust Flight Control: A Design Challenge*. Springer-Verlag, London, 1997. In *Lecture Notes in Control and Information Sciences, 224*.
18. G. Herrmann, M. C. Matthew, C. Turner, P. Menon, D. G. Bates, and I. Postlethwaite. Anti-windup Synthesis For Nonlinear Dynamic Inversion Controllers. In *Proceedings of the 5th IFAC Symposium on Robust Control Design*, Toulouse, France, 2006.

Chapter 7
Autopilot for the Longitudinal Motion

The nonlinear differential equations governing the motion of an aircraft are described in Chap. 3. For the plant analysis and control design, these equations are linearized around a certain operating point. Two sets of state variables appear to be clearly decoupled, each defining a specific mode of aircraft motion. The state variables involved in the longitudinal mode are the pitch rate q, the airspeed V_T, the angle of attack α, and the pitch angle θ. The lateral-directional mode involves the state variables for the roll rate p, the yaw rate r, the sideslip angle β, and the roll angle ϕ.

This chapter is dedicated to the analysis and control of the longitudinal motion of the aircraft and presents an architecture for the altitude controller, which uses robust NDI in all of the control loops. This chapter brings an innovative and practical approach for stability and robustness analyses of the plant undergoing the dynamic inversion process. Moreover, this chapter provides a systematic procedure for the selection of some uncertain model parameters involved in the controllers. Finally, a new nonlinear airspeed controller is also designed and presented.

7.1 Equations for Longitudinal Mode Analysis

A linear model of the longitudinal motion of the aircraft is constructed as follows:

$$\boldsymbol{\Delta \dot{x}}_{long} = \boldsymbol{A}_{long}\boldsymbol{\Delta x}_{long} + \boldsymbol{B}_{long}\boldsymbol{\Delta u}_{long}\,, \tag{7.1}$$

where the state vector is defined as $\boldsymbol{x}_{long} = [q \quad V_T \quad \alpha \quad \theta]^T$ and the control input vector is $\boldsymbol{u}_{long} = [\delta_{elevator} \quad \delta_{Thrust}]$ as shown in Fig. 7.1.

In order to build the linear longitudinal model and therefore obtain the matrices $\boldsymbol{A}_{long}$, $\boldsymbol{B}_{long}$, $\boldsymbol{C}_{long}$, and $\boldsymbol{D}_{long}$, a nonlinear differential equation must first be formulated for each of the state variables as follows:

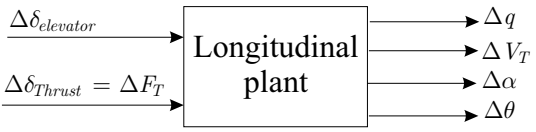


Fig. 7.1 Linear model of the aircraft longitudinal mode

$$\dot{q} = f_q(\boldsymbol{x}, \boldsymbol{u}) \; , \qquad \dot{V}_T = f_{V_T}(\boldsymbol{x}, \boldsymbol{u}) \; ,$$
$$\dot{\alpha} = f_\alpha(\boldsymbol{x}, \boldsymbol{u}) \; , \qquad \dot{\theta} = f_\theta(\boldsymbol{x}, \boldsymbol{u}) \; ,$$

where the state vector of the aircraft is $\boldsymbol{x}$, and the control vector is $\boldsymbol{u}$.

7.1.1 Pitch Rate Differential Equation

The nonlinear differential equation for the pitch rate is found from (3.44) as

$$\dot{q} = \frac{1}{I_{yy}} \left[M^b - (I_{xx} - I_{zz})pr + I_{xz}(p^2 - r^2) \right] \; , \tag{7.2}$$

where the pitch torque M^b is expressed as a linear combination of the control surface deflections, the angle of attack α, and the dimensionless pitch rate $\tilde{q} = \bar{c}q/2V_T$ as follows:

$$M^b = \bar{q}S\bar{c}\,(C_{M1} + C_{M_e}\delta_e + C_{M\alpha}\alpha + C_{M\tilde{q}}\,\tilde{q}) \; . \tag{7.3}$$

The definition of the other variables can be found in Chap. 3.

7.1.2 Airspeed Differential Equation

The nonlinear differential equation of the aircraft's airspeed is derived in Appendix A and is formulated as follows:

$$\dot{V}_T = \frac{1}{m}(X^w + F_T \cos\alpha \cos\beta + m g_x^w) \; , \tag{7.4}$$

where the drag force is $X^w = \frac{\rho V^2}{2} S(C_{X1} + C_{X\alpha}\,\alpha + C_{X\alpha 2}\,\alpha^2 + C_{X\beta 2}\,\beta^2)$ and the gravity term is

$$g_x^w = g(-\sin\theta\cos\alpha\cos\beta + \cos\theta\sin\phi\sin\beta + \cos\theta\cos\phi\sin\alpha\cos\beta) \; . \tag{7.5}$$

For the analysis of the longitudinal motion, the aircraft is assumed to fly straight and level with no sideslipping and no wind, yielding

$$\beta_{nom} = 0, \quad \theta_{nom} = \alpha_{nom}, \quad p_{nom} = q_{nom} = r_{nom} = 0, \quad \phi_{nom} = 0\ . \quad (7.6)$$

The nonlinear differential equation for the airspeed is simplified as follows:

$$\dot{V}_T = \frac{1}{m}\left(\frac{\rho V^2 S}{2}(C_{X1} + C_{X\alpha}\,\alpha + C_{X\alpha 2}\,\alpha^2) + F_T \cos\alpha\right). \quad (7.7)$$

7.1.3 Differential Equation for the Angle of Attack

The nonlinear differential equation of the aircraft's angle of attack α is derived in Appendix A and is formulated as follows:

$$\dot{\alpha} = \frac{1}{\cos\beta}\left(\frac{1}{mV_T}[Z^w - F_T \sin\alpha + mg_z^w] + q^w\right), \quad (7.8)$$

with the lift force $Z^w = \frac{\rho V^2}{2} S(C_{Z1} + C_{Z\alpha}\alpha)$, the gravity term $g_z^w = g(\sin\alpha\sin\theta + \cos\alpha\cos\theta\cos\phi)$, and the pitch rate in wind axes $q^w = q\cos\beta - p\sin\beta\cos\alpha - r\sin\alpha\sin\beta$. Under the assumptions in (7.6) the dynamics of the angle of attack can be simplified as

$$\dot{\alpha} = \frac{1}{mV_T}\left[\frac{\rho V^2}{2} S(C_{Z1} + C_{Z\alpha}\,\alpha) - F_T \sin\alpha + mg\right] + q\ . \quad (7.9)$$

7.1.4 Differential Equation for the Pitch Angle

The differential equation for the pitch angle θ is

$$\dot{\theta} = q\ . \quad (7.10)$$

7.1.5 Matrices for the Longitudinal Mode

The matrices for the linear longitudinal mode are obtained as follows:

$$\begin{bmatrix} \dot{\Delta q} \\ \dot{\Delta V_T} \\ \dot{\Delta\alpha} \\ \dot{\Delta\theta} \end{bmatrix} = \boldsymbol{A_{Long}} \begin{bmatrix} \Delta q \\ \Delta V_T \\ \Delta\alpha \\ \Delta\theta \end{bmatrix} + \boldsymbol{B_{Long}} \begin{bmatrix} \Delta\delta_{elevator} \\ \Delta\delta_{Thrust} \end{bmatrix}, \quad (7.11)$$

with the dynamic matrix

$$\boldsymbol{A_{Long}} = \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{x}}\right)_{\boldsymbol{x}_{nom},\,\boldsymbol{u}_{nom}}$$

$$= \begin{bmatrix} \frac{V_T\,\rho\,S\,\bar{c}^2\,C_{M\dot{q}}}{4I_{yy}} & A_{Long(1,2)} & \frac{V_T^2\,\rho\,S\,\bar{c}\,C_{M\alpha}}{2I_{yy}} & 0 \\ 0 & \frac{\rho V S\left[C_{X1}+C_{X\alpha}\,\alpha+C_{X\alpha 2}\,\alpha^2\right]}{m} & A_{Long(2,3)} & 0 \\ 1 & \frac{Z^w+F_T\sin\alpha-mg}{mV_T^2} & \frac{V_T\,\rho\,S\,C_{Z\alpha}}{2m} & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}_{\substack{\boldsymbol{x}_{nom},\\ \boldsymbol{u}_{nom}}} \tag{7.12}$$

$$A_{Long(1,2)} = \frac{\rho\,V_T\,S\,\bar{c}\,\left[C_{M1}\;+\;C_{Me}\,\delta_e\;+\;C_{M\alpha}\,\alpha\right]}{I_{yy}}\,,$$

$$A_{Long(2,3)} = \frac{\frac{\rho V_T^2}{2}S(C_{X\alpha}\;+\;2C_{X\alpha 2}\,\alpha)\;-\;F_T\sin\alpha}{m}\,,$$

and the control input matrix

$$\boldsymbol{B_{Long}} = \left(\frac{\partial \boldsymbol{f}(\boldsymbol{x},\boldsymbol{u})}{\partial \boldsymbol{u}}\right)_{\substack{\boldsymbol{x}_{nom},\\ \boldsymbol{u}_{nom}}} = \begin{bmatrix} \frac{\bar{q}\,S\,\bar{c}\,C_{Me}}{I_{yy}} & 0 \\ 0 & \frac{\cos\alpha\cos\beta}{m} \\ 0 & \frac{-\sin\alpha}{mV_T\cos\beta} \\ 0 & 0 \end{bmatrix}_{\substack{\boldsymbol{x}_{nom},\\ \boldsymbol{u}_{nom}}}\,, \tag{7.13}$$

where the input $\delta_{Thrust} = F_T$ is expressed in N. The numerical values of $\boldsymbol{A_{Long}}$ and $\boldsymbol{B_{Long}}$ are provided in Appendix E.

7.2 Dynamic Modes of the Longitudinal Plant

The eigenvalues of the dynamic matrix $\boldsymbol{A_{Long}}$ are computed and two pairs of complex eigenvalues appear corresponding to two oscillatory modes called the short-period mode and the phugoid mode as shown in Fig. 7.2.

7.2.1 Short-period Mode

This mode has a frequency of $\omega_{sp} = 4.72\,\text{rad/s}$ and a damping coefficient of $\zeta_{sp} = 0.9$. The short-period mode corresponds to oscillations in the angle of attack and the pitch angle at approximately constant airspeed. In order to efficiently compensate for the undesired motion due to this mode, the pitch rate and the angle-of-attack controllers are to be designed with a bandwidth larger than ω_{sp}.

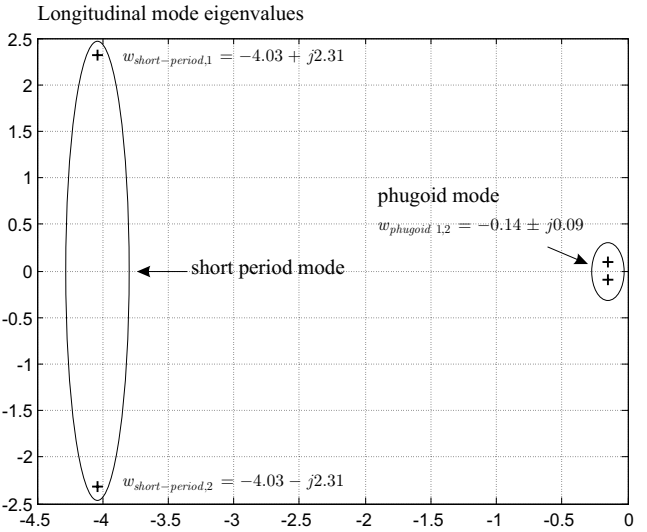


Fig. 7.2 Aircraft longitudinal mode eigenvalues

7.2.2 Phugoid Mode

The phugoid mode involves the coupled effects of potential and kinetic energy. Indeed, as the aircraft pitches up and climbs, its speed decreases causing the aircraft to pitch down again and therefore descend. As the aircraft descends, its speed increases and the aircraft starts to pitch up again, and so on. This oscillation is slow with a frequency of $\omega_{ph} = 0.2\,\text{rad/s}$ and a damping coefficient of $\zeta_{ph} = 0.64$.

7.3 Validation of the Linear Longitudinal Model

In order to validate the analytical terms found for $\boldsymbol{A_{long}}$ in (7.12) and $\boldsymbol{B_{long}}$ in (7.13), the linear longitudinal model is simulated for some input perturbation and compared with a nonlinear simulation, which has the same input perturbation.

7.3.1 Perturbation on the Elevator Command

The top plot in Fig. 7.3 shows a perturbation of about 10 % on the elevator command. The other plots in Fig. 7.3 show that the nonlinear and linear simulations are in good agreement.

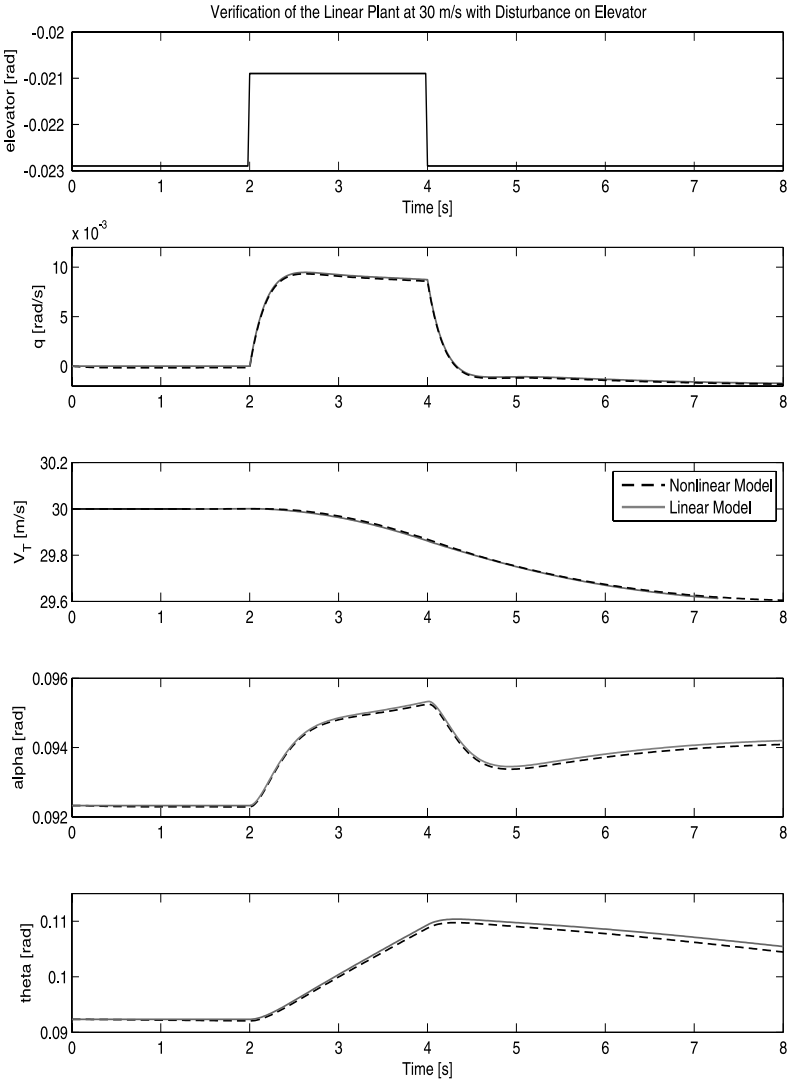


Fig. 7.3 Validation of the linear longitudinal plant with a small perturbation on the elevator

7.3.2 Perturbation on the Engine Speed n_{mot}

The top plot in Fig. 7.4 shows a perturbation of about 10 % on the engine rotational speed n_{mot}. The other plots in Fig. 7.4 show that the nonlinear and linear simulations are in good agreement, thus validating the linear model of the longitudinal axis.

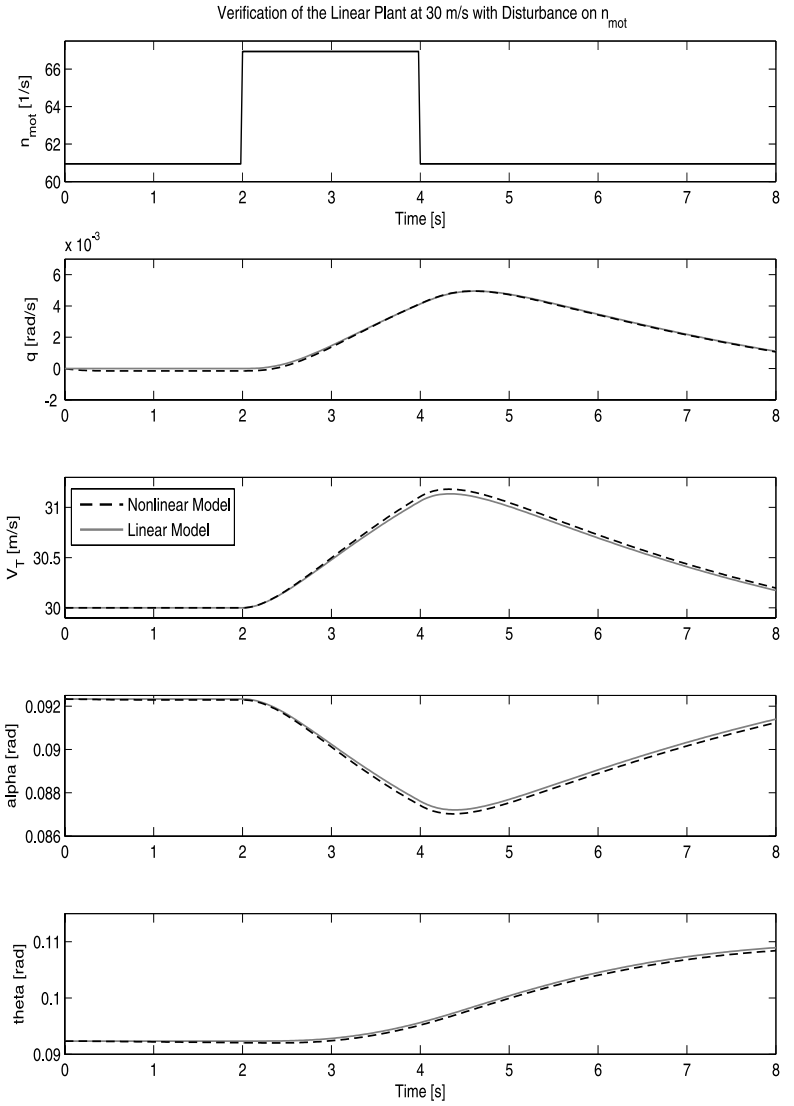


Fig. 7.4 Validation of the linear longitudinal plant with a small perturbation on the engine rate n_{mot}

7.4 Stability Analysis of the Uncertain Dynamic Inversion

Perfect NDI control assumes that the plant dynamics are perfectly known, such that perfect cancelation of the nonlinearities in the plant dynamics can be achieved. This assumption is obviously not satisfied in practice where all the parameters of the aircraft model are subject to uncertainties and where all the measurement data are corrupted by noise and bias.

Many papers have already discussed methods to design stable and robust control systems based on the dynamic inversion method [1–8]. In general the NDI is used in an inner control loop to cancel certain nonlinearities of the system and an outer control loop is designed to guarantee the performance and the stability requirements of the whole system; see Fig. 6.1. For example, in [4] the outer loop is designed using the linear quadratic gaussian (LQG) control method. The stability analysis is made using the methodology developed by [9], which gives the largest stability domain around a steady-state operating point.

Other papers show how H_∞ controllers can be designed for the outer contol loop [7, 10, 11]. A μ-analysis is presented, for example, in [1, 12] and the stability of the system can be measured by the value of μ, whereas the robustness of the system is guaranteed if $\mu < 1$.

In [7], NDI is used in the inner loop controller in order to equalize the plant dynamics over the flight envelope. The stability of the closed-loop system (together with the outer-loop controller) over all operating points is assessed using the *relative error method* presented in [13].

Another approach is the stochastic stability and robustness analysis and design [8, 14]. This technique estimates the probability of system instability and performance requirements violation when stochastic variations of the system parameters are introduced.

In [3], an alternative approach for the design and analysis of a robust NDI method for aircraft motion control is presented. That approach is based upon Lyapunov stability theory and multiple time-scale dynamic inversion.

In [6], linear stability and robustness analyses are conducted on linearized models computed locally throughout the flight envelope and the classical gain and phase margins are determined.

More recently, the analysis of NDI flight control laws has also been investigated using LPV or quasi-LPV models. Using this framework, it is possible to model nonlinear effects and parametric uncertainties in the system. In order to analyse stability and robustness properties of LPV-based models, quadratic Lyapunov functions and parameter-dependent Lyapunov functions have been used in [15] and [16], respectively. Some additional techniques based on Lyapunov quadratic functions are developed in [17] for assessment of the robust performance and robust stability of a quasi-LPV model with respect to time-varying, parametric uncertainties.

Although many advanced analysis techniques have been presented to quantify the stability of a control system using NDI, very few papers show the "physical" effects of uncertainties on the behavior of the system. This section considers the longitudinal motion of an aircraft and shows the effects of parameter and measurement uncertainties on the inversion processes. The contributions of this work to the analyses of stability and robustness of the NDI controller for the longitudinal motion are [18]:

- To show that an explicit transfer function can be derived from the plant input to the plant output, which includes the dynamic inversion process and all the relevant model parameter and measurement uncertainties.
- To suggest a framework in which the stability and robustness properties can be assessed using the notion of phase and gain margins from classical control theory.
- To show that uncertain parameters selected in the dynamic inversion paths can either stabilize or destabilize the system to be controlled.
- To provide a systematic procedure for the selection of uncertain model parameters involved in the dynamic inversion paths.

7.4.1 Uncertain Model Parameters and Measurement Data

The uncertain model parameters involved in the longitudinal axis are C_{M1}, C_{Me}, $C_{M\tilde{q}}$, $C_{M\alpha}$, I_{yy}. The uncertain measured state elements are α, q, and V_T. The uncertainty levels of these parameters are summarized in Table 7.1.

Table 7.1 Summary of the parameter and measurement uncertainties for the pitch-axis control loop

Parameter	Nominal value	Level of uncertainty	Unit
C_{M1}	2.08×10^{-2}	$\pm 10\,\%$	-
C_{Me}	5.45×10^{-1}	$\pm 20\,\%$	-
$C_{M\tilde{q}}$	-9.03×10^{-2}	$\pm 20\,\%$	rad^{-1}
$C_{M\alpha}$	-9.83	$\pm 20\,\%$	rad^{-1}
I_{yy}	10.9	$\pm 5\,\%$	kg m^2
α	9.23×10^{-2} (5.29)	$\pm 5\,\%$	rad (deg)
q	0	$\pm 5\,\%$	rad/s
V_T	30	$\pm 5\,\%$	m/s

Furthermore, it is assumed that there is no uncertainty on the aircraft mass m, the wing surface S, the wing span b, the air density ρ, and the mean aerodynamic chord $\bar{c}$.

7.4.2 Linear Modeling of the Uncertain Dynamic Inversion

Figure 7.5 shows the NDI process involved in the innermost control loop of the longitudinal axis, namely the pitch rate control loop. If dynamic inversion operates perfectly, the plant behaves as a pure integrator. However, as soon as some uncertainty is introduced in this process, this pure integrator transforms into either a stable or unstable system depending on the uncertainty. This is discussed in the rest of this chapter.

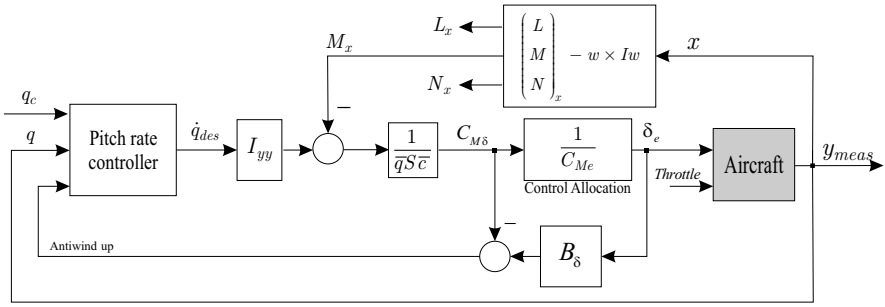


Fig. 7.5 NDI for pitch rate control

In order to build a linear transfer function from the input $\dot{q}_{des}$ to the output y_{meas}, the additive inverse term and the multiplicative inverse term of the dynamic inversion path are linearized as shown in Fig. 7.6. The uncertain plant matrices are called $\boldsymbol{A}^u$, $\boldsymbol{B}^u$ and $\boldsymbol{C}^u$, where the superscript u stands for "uncertain".

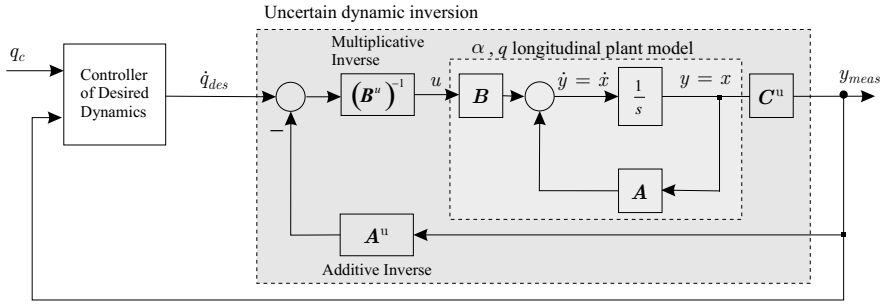


Fig. 7.6 Linear dynamic inversion with uncertainties for pitch rate control

The state space representation of the system in Fig. 7.6 is [18]

$$\begin{aligned}\dot{\boldsymbol{\Delta x}} &= \left[\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{CB}^u)^{-1}\boldsymbol{A}^u\boldsymbol{C}^u\right]\boldsymbol{\Delta x} + \boldsymbol{B}(\boldsymbol{CB}^u)^{-1}\boldsymbol{\Delta u}\,, \\ &= \boldsymbol{A_{DI}}\,\boldsymbol{\Delta x} + \boldsymbol{B_{DI}}\,\boldsymbol{\Delta u}\,. \end{aligned} \tag{7.14}$$

7.4.3 Model Simplification for the Longitudinal Motion

The variables describing the aircraft longitudinal axis can be divided based on a time-scale-separation argumentation between the fast time-scale variables α and q, and the slow time-scale variable V_T and flight-path angle $\gamma = \theta - \alpha$.

A linear model of the pitch axis is built which only involves the fast time-scale variables as follows:

$$\begin{aligned}\begin{bmatrix}\dot{q}\\ \dot{\alpha}\end{bmatrix} &= \begin{bmatrix}a_{11} & a_{12}\\ a_{21} & a_{22}\end{bmatrix}\begin{bmatrix}q\\ \alpha\end{bmatrix} + \begin{bmatrix}b_{11} & 0\\ 0 & b_{22}\end{bmatrix}\begin{bmatrix}\delta_e\\ F_T\end{bmatrix}\,, \\ &= \boldsymbol{A_{q,\alpha}}\begin{bmatrix}q\\ \alpha\end{bmatrix} + \boldsymbol{B_{q,\alpha}}\begin{bmatrix}\delta_e\\ F_T\end{bmatrix}\,. \end{aligned} \tag{7.15}$$

The dynamic inversion process on the longitudinal axis is employed to directly affect the dynamics of the pitch rate q as shown in Fig. 7.5. Therefore, the state variable α is not concerned with the inversion process and thus the second line of the matrix $\boldsymbol{A_{DI}}$ is equal to the second line of the matrix $\boldsymbol{A_{q,\alpha}}$ of (7.15). Uncertainties in the terms of the dynamic matrix of the system in (7.15) are introduced as follows:

$$\boldsymbol{A^u_{q,\alpha}} = \begin{bmatrix}a_{11} + \Delta a_{11} & a_{12} + \Delta a_{12}\\ a_{21} + \Delta a_{21} & a_{22} + \Delta a_{22}\end{bmatrix}\,; \tag{7.16}$$

uncertainties in the terms of the the control input matrix are introduced as follows:

$$\boldsymbol{B^u_{q,\alpha}} = \begin{bmatrix}b_{11} + \Delta b_{11} & 0\\ 0 & b_{22} + \Delta b_{22}\end{bmatrix}\,; \tag{7.17}$$

and uncertainties in the terms of the measurement matrix are introduced as follows:

$$\boldsymbol{C^u_{q,\alpha}} = \begin{bmatrix}c_{11} + \Delta c_{11} & 0\\ 0 & c_{22} + \Delta c_{22}\end{bmatrix}\,. \tag{7.18}$$

7.4.4 Linear Model of the Pitch Axis and Dynamic Inversion Process

7.4.4.1 Perfect Dynamic Inversion

If there is no model uncertainty and no measurement error, the dynamic inversion process is perfect, yielding

$$\boldsymbol{A_{DI}}_{(1,1)} = \boldsymbol{A_{DI}}_{(1,2)} = 0, \quad \text{and} \quad \boldsymbol{B_{DI}}_{(1,1)} = 1 \, . \tag{7.19}$$

The matrices $\boldsymbol{A_{DI}} \in \mathbf{R}^{2\times 2}$ and $\boldsymbol{B_{DI}} \in \mathbf{R}^{2\times 2}$ of the system in (7.15) are of the form

$$\boldsymbol{A_{DI}} = \begin{bmatrix} 0 & 0 \\ a_{21} & a_{22} \end{bmatrix}, \quad \boldsymbol{B_{DI}} = \begin{bmatrix} 1 & 0 \\ 0 & b_{22} \end{bmatrix}, \quad \boldsymbol{C_{DI}} = \begin{bmatrix} C_q & 0 \\ 0 & C_\alpha \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{7.20}$$

A graphical representation of the perfect dynamic inversion process is shown in the right-hand plot in Fig. 7.7, where we clearly see that $q_{meas} = \dot{q}_{des}/s$.

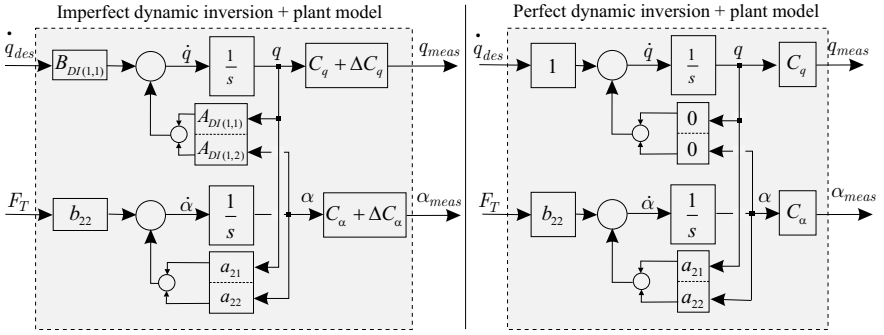


Fig. 7.7 Imperfect vs. perfect dynamic inversion

7.4.4.2 Imperfect Dynamic Inversion

In the case of imperfect dynamic inversion, the terms in the matrices $\boldsymbol{A_{DI}}$ and $\boldsymbol{B_{DI}}$ are obtained as follows:

$$\begin{aligned} \dot{q} &= a_{11}c_{11}q + a_{12}c_{22}\alpha + b_{11}\left(b_{11}^u\right)^{-1}\left(\dot{q}_{des} - a_{11}^u c_{11}^u q - a_{12}^u c_{22}^u \alpha\right) \, , \\ &= \left\{a_{11}c_{11} - b_{11}\left(b_{11}^u\right)^{-1} a_{11}^u c_{11}^u\right\} q + \left\{a_{12}c_{22} - b_{11}\left(b_{11}^u\right)^{-1} a_{12}^u c_{22}^u\right\} \alpha \\ &\quad + b_{11}\left(b_{11}^u\right)^{-1} \dot{q}_{des} \, , \\ &= A_{DI(1,1)} q + A_{DI(1,2)} \alpha + B_{DI(1,1)} \dot{q}_{des} \, . \end{aligned} \tag{7.21}$$

The dynamics of the angle of attack are obtained as follows:

$$\begin{aligned}\dot{\alpha} &= a_{21}q + a_{22}\alpha + b_{22}F_T \ , \\ &= A_{DI(2,1)}q + A_{DI(2,2)}\alpha + B_{DI(2,2)}F_T \ . \end{aligned} \tag{7.22}$$

The terms c_{11} and c_{22} are both equal to one, whereas the uncertain measurement coefficients are $c_{11}^u = c_{11} + \Delta c_{11}$ and $c_{22}^u = c_{22} + \Delta c_{22}$.

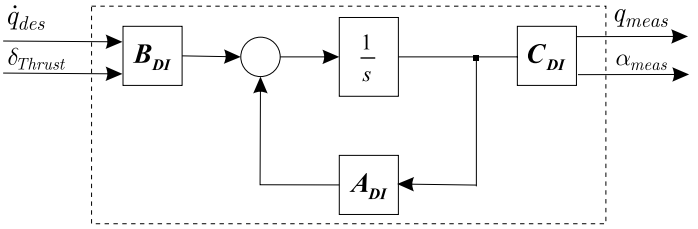


Fig. 7.8 Uncertain dynamic inversion and simplified longitudinal plant

The dynamic matrix, the control input matrix, and the measurement matrix of the simplified plant, together with the uncertain dynamic inversion process as shown in Fig. 7.8, are finally obtained as follows:

$$\boldsymbol{A_{DI}} = \begin{bmatrix} a_{11} - \frac{b_{11}(a_{11}+\Delta a_{11})(c_{11}+\Delta c_{11})}{b_{11}+\Delta b_{11}} & a_{12} - \frac{b_{11}(a_{12}+\Delta a_{12})(c_{22}+\Delta c_{22})}{b_{11}+\Delta b_{11}} \\ a_{21} & a_{22} \end{bmatrix}, \tag{7.23}$$

$$\boldsymbol{B_{DI}} = \begin{bmatrix} \frac{b_{11}}{b_{11}+\Delta b_{11}} & 0 \\ 0 & b_{22} \end{bmatrix}, \tag{7.24}$$

$$\boldsymbol{C_{DI}} = \begin{bmatrix} c_{11} + \Delta c_{11} & 0 \\ 0 & c_{22} + \Delta c_{22} \end{bmatrix}. \tag{7.25}$$

7.4.4.3 Computing the Plant's Transfer Functions Including Model Uncertainties

Explicit transfer functions can be obtained using the following formula:

$$\boldsymbol{G}(s) = \boldsymbol{C_{DI}}(s\boldsymbol{I} - \boldsymbol{A_{DI}})^{-1}\boldsymbol{B_{DI}} = \frac{\boldsymbol{C_{DI}}\mathrm{adj}(s\boldsymbol{I} - \boldsymbol{A_{DI}})\boldsymbol{B_{DI}}}{|s\boldsymbol{I} - \boldsymbol{A_{DI}}|} \ . \tag{7.26}$$

The transfer function for the channel from $\dot{q}_{des}$ to q_{meas} is obtained as follows:

$$\frac{q_{meas}}{\dot{q}_{des}} = \frac{\boldsymbol{C_{DI}}_{(1,1)}\boldsymbol{B_{DI}}_{(1,1)}(s - \boldsymbol{A_{DI}}_{(2,2)})}{\mathcal{D}}\,, \tag{7.27}$$

with $\mathcal{D} = s^2 - s(\boldsymbol{A_{DI}}_{(1,1)} + \boldsymbol{A_{DI}}_{(2,2)}) + \boldsymbol{A_{DI}}_{(1,1)}\boldsymbol{A_{DI}}_{(2,2)} - \boldsymbol{A_{DI}}_{(2,1)}\boldsymbol{A_{DI}}_{(1,2)}$.

7.4.4.4 MATLAB® Validation

In order to validate the results derived in the above paragraphs, three different methods are used to calculate the transfer function $q_{meas}/\dot{q}_{des}$.

First, the Simulink® diagram shown in Fig. 7.9 is constructed, where the matrices A1_alq_long, B1_alq_long, and C1_alq_long represent the uncertain matrices $\boldsymbol{A}^{\boldsymbol{u}}_{\boldsymbol{q},\boldsymbol{\alpha}}$, $\boldsymbol{B}^{\boldsymbol{u}}_{\boldsymbol{q},\boldsymbol{\alpha}}$, and $\boldsymbol{C}^{\boldsymbol{u}}_{\boldsymbol{q},\boldsymbol{\alpha}}$, respectively. The state space block contains the original matrices $\boldsymbol{A}_{\boldsymbol{q},\boldsymbol{\alpha}}$, $\boldsymbol{B}_{\boldsymbol{q},\boldsymbol{\alpha}}$, and $\boldsymbol{C}_{\boldsymbol{q},\boldsymbol{\alpha}}$. The output of the state space block is the vector $[q \quad \alpha]^T$. The MATLAB® script shown in Table 7.2 is used to compute the transfer function from the input $\dot{q}_{des}$ to the plant output q_{meas} using the MATLAB® command "linmod".

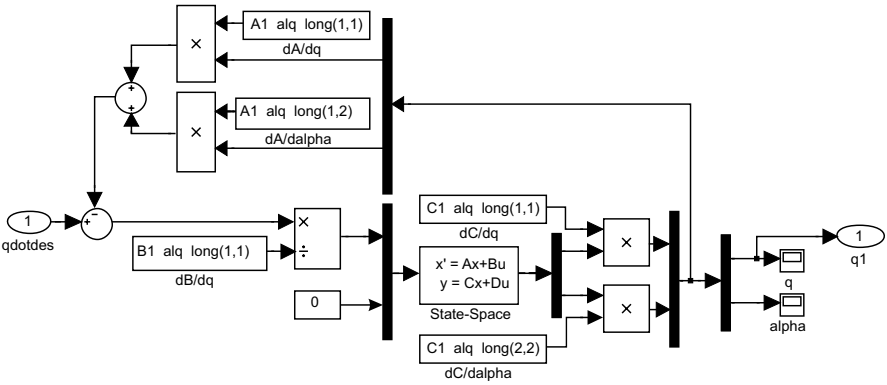


Fig. 7.9 Simulink® diagram called "alpha_qlinearDITest" used to compute the transfer function with the command "linmod" from $\dot{q}_{des}$ to q as shown in Table 7.2

Table 7.2 MATLAB® commands used with the Simulink® diagram in Fig. 7.9 to compute the transfer function from $\dot{q}_{des}$ to q_{meas}

```
[aa_alq, bb_alq, cc_alq, dd_alq] = linmod('alpha_qlinearDITest');
sys_plant_alq_linmod = ss(aa_alq, bb_alq, cc_alq, dd_alq);
tf_qdotdes_to_q_alq_linmod = tf(sys_plant_alq_linmod);
```

A second method is investigated, where the matrices $\boldsymbol{A_{DI}}$, $\boldsymbol{B_{DI}}$, and $\boldsymbol{C_{DI}}$ are first computed by MATLAB®, and the command "ss" is utilized to create a state space system out of these matrices. Finally, the transfer function is computed with the command "tf" as shown in Table 7.3.

Table 7.3 Second method to compute the transfer function from $\dot{q}_{des}$ to q_{meas}

```
sys_DI = ss(A_DI, B_DI, C_DI, D_DI);
DI_tf = tf(sys_DI);
DI_tf(1, 1)
```

Third, the transfer function is explicitly evaluated given (7.27), and the MATLAB® code is shown in Table 7.4.

Table 7.4 Third method to compute the transfer function from $\dot{q}_{des}$ to q_{meas}

```
s = tf('s');
hand_qrateDI_tf = C_DI(1, 1) * B_DI(1, 1) * (s − A_DI(2, 2))/(s^2 − ...
...(A_DI(1, 1) + A_DI(2, 2)) * s + A_DI(1, 1) * A_DI(2, 2) − A_DI(2, 1) * A_DI(1, 2));
```

The top plot in Fig. 7.10 shows that if no uncertainty is introduced in the system, the transfer function obtained by each of the three methods is that of a pure integrator, *i.e*, $q_{meas}/\dot{q}_{des} = 1/s$.

The bottom plot in Fig. 7.10 shows the case where all the terms of the matrices $\boldsymbol{A^u_{q,\alpha}}$, $\boldsymbol{B^u_{q,\alpha}}$, and $\boldsymbol{C^u_{q,\alpha}}$ are 30 % lower than the original values. Using the three different methods the transfer function $q_{meas}/\dot{q}_{des}$ is computed. The results are identical and show that uncertainties in the system transform the pure integrator obtained by perfect dynamic inversion into $q_{meas}/\dot{q}_{des} = (s + 3.657)/(s^2 + 5.091s + 6.607)$ in this case.

Section 7.4.6 presents in more detail how the parameter uncertainties affect the dynamic inversion process.

7.4.5 Evaluation of the Uncertainty Terms in the Matrix $\boldsymbol{A_{DI}}$

Suppose the variable y is a function of the parameters x, z, t, with the associated uncertainties Δx, Δz, Δt. The maximum possible error on $y = f(x, z, t)$ is computed with the following relationship:

$$\Delta y = \left|\frac{\partial f}{\partial x}\right| \Delta x + \left|\frac{\partial f}{\partial z}\right| \Delta z + \left|\frac{\partial f}{\partial t}\right| \Delta t \ . \tag{7.28}$$

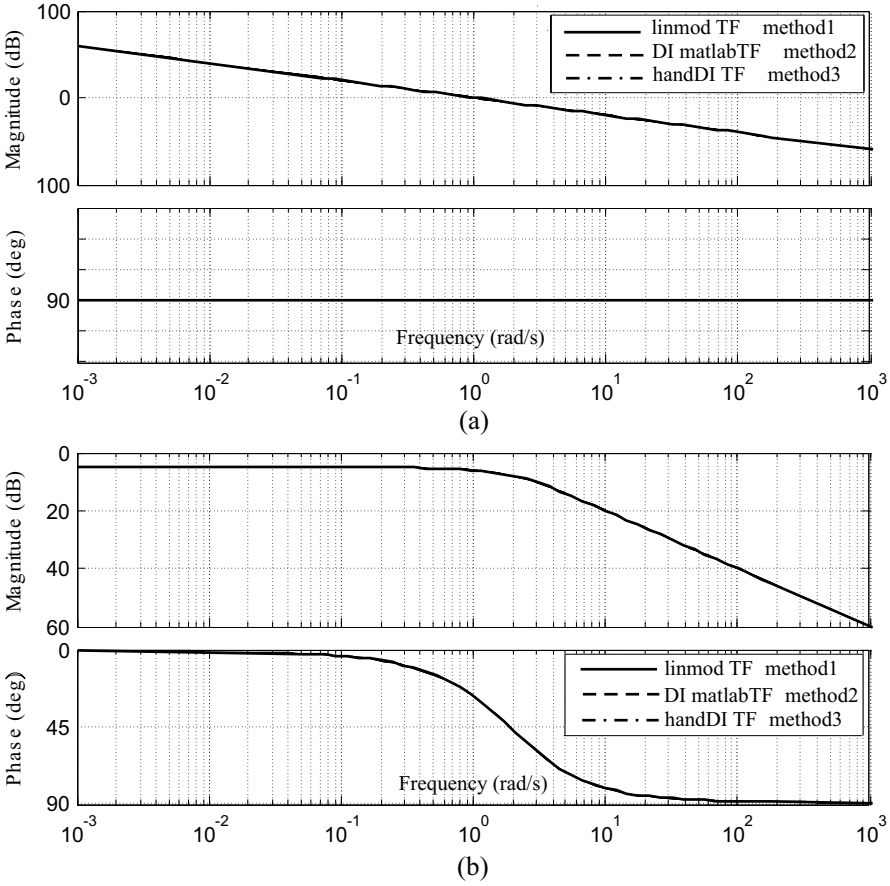


Fig. 7.10 Comparisons between the three methods to obtain the transfer function $q_{meas}/\dot{q}_{des}$: (a) Bode diagram: no uncertainty in the system; and (b) Bode diagram: with uncertainties in the system

If the variable y is a ratio of the variables x and z, $y = \frac{x}{z}$, the maximum relative error on y is computed as follows [19]:

$$\frac{\Delta y}{y} = \frac{\Delta x}{x} + \frac{\Delta z}{z} \ . \tag{7.29}$$

The uncertainty of the term a_{11} of the matrix $\boldsymbol{A_{q,\alpha}}$ defined in (7.15) is computed as follows:

$$a_{11} = \frac{\partial \dot{q}}{\partial q} = f(V_T, C_{M\tilde{q}}, I_{yy}) = \frac{V_T \rho S \bar{c}^2 C_{M\tilde{q}}}{4 I_{yy}} \ ,$$

$$\begin{aligned}
\Delta a_{11} &= \left|\frac{\partial f}{\partial V_T}\right| \Delta V_T + \left|\frac{\partial f}{\partial C_{M\tilde{q}}}\right| \Delta C_{M\tilde{q}} + \left|\frac{\partial f}{\partial I_{yy}}\right| \Delta I_{yy} \ , \\
\left|\frac{\Delta a_{11}}{a_{11}}\right| &= \left|\frac{\Delta V_T}{V_T}\right| + \left|\frac{\Delta C_{M\tilde{q}}}{C_{M\tilde{q}}}\right| + \left|\frac{\Delta I_{yy}}{I_{yy}}\right| \ , \\
&\approx 5\,\% + 20\,\% + 5\,\% = 30\,\% \ . \qquad (7.30)
\end{aligned}$$

The uncertainty of the term a_{12} of the matrix $\boldsymbol{A_{q,\alpha}}$ defined in (7.15) is computed as follows:

$$\begin{aligned}
a_{12} &= \frac{\partial \dot{q}}{\partial \alpha} = f(V_T, C_{M\alpha}, I_{yy}) = \frac{V_T^2 \rho S \bar{c} C_{M\alpha}}{2 I_{yy}} \ , \\
\left|\frac{\Delta a_{12}}{a_{12}}\right| &= \left|\frac{\Delta V_T}{V_T}\right| + \left|\frac{\Delta C_{M\alpha}}{C_{M\alpha}}\right| + \left|\frac{\Delta I_{yy}}{I_{yy}}\right| \ , \\
&\approx 5\,\% + 20\,\% + 5\,\% = 30\,\% \ . \qquad (7.31)
\end{aligned}$$

The uncertainty of the term b_{11} of the matrix $\boldsymbol{B_{q,\alpha}}$ in (7.15) is computed as follows:

$$\begin{aligned}
b_{11} &= \frac{\partial \dot{q}}{\partial \delta_e} = f(V_T, C_{Me}, I_{yy}) \ , \\
&= \frac{\rho V_T^2 S \bar{c} C_{Me}}{2 I_{yy}} \ , \\
\left|\frac{\Delta b_{11}}{b_{11}}\right| &= \left|\frac{\Delta V_T}{V_T}\right| + \left|\frac{\Delta C_{Me}}{C_{M\delta e}}\right| + \left|\frac{\Delta I_{yy}}{I_{yy}}\right| \ , \\
&\approx 5\,\% + 20\,\% + 5\,\% = 30\,\% \ . \qquad (7.32)
\end{aligned}$$

The uncertainty of the terms a_{21}, a_{22}, b_{21} and b_{22} is chosen to be zero, since these terms are not used in the dynamic inversion of the dynamics of the pitch rate q.

7.4.6 Effect of Uncertainties on Dynamic Inversion

In this section, the matrices $\boldsymbol{A_{DI}}$ and $\boldsymbol{B_{DI}}$ are computed with uncertainties according to (7.23) and (7.24).

The transfer function $q_{meas}/\dot{q}_{des}$ in (7.27) is computed, where the coefficients $\frac{\Delta a_{11}}{a_{11}}$, $\frac{\Delta b_{11}}{b_{11}}$, $\frac{\Delta a_{12}}{a_{12}}$, $\frac{\Delta c_{11}}{c_{11}}$ are successively made negative and positive in a nested for-loop algorithm in MATLAB®. The results are shown in Fig. 7.12 and Fig. 7.13.

It appears that the transfer function $q_{meas}/\dot{q}_{des}$ has at least one unstable pole if:

- $\frac{\Delta a_{11}}{a_{11}}$ and $\frac{\Delta b_{11}}{b_{11}}$ have the same sign, and $\frac{\Delta q}{q} > 0$
- $\frac{\Delta a_{11}}{a_{11}} > 0$ and $\frac{\Delta b_{11}}{b_{11}} < 0$.

The transfer function $q_{meas}/\dot{q}_{des}$ only has stable poles if:

- $\frac{\Delta a_{11}}{a_{11}}$ and $\frac{\Delta b_{11}}{b_{11}}$ have the same sign, and $\frac{\Delta q}{q} < 0$
- $\frac{\Delta a_{11}}{a_{11}} < 0$ and $\frac{\Delta b_{11}}{b_{11}} > 0$.

Therefore, for the actual implementation of the pitch rate controller in the UAV control unit, a safe choice for the plant parameters would be

$$a_{11} = a_{11,\text{nom}} \left(1 - \left|\frac{\Delta a_{11}}{a_{11}}\right|\right) ,$$
$$b_{11} = b_{11,\text{nom}} \left(1 + \left|\frac{\Delta b_{11}}{b_{11}}\right|\right) , \tag{7.33}$$

where the terms $a_{11,\text{nom}}$ and $b_{11,\text{nom}}$ may be obtained from wind-tunnel experiments or real flight tests. For the measurement of the pitch rate, a safe choice would be

$$q = q_{meas} \left(1 - \left|\frac{\Delta q}{q}\right|\right) . \tag{7.34}$$

By using the above suggestions in (7.33) and (7.34), the plant from $\dot{q}_{des}$ to q_{meas} remains stable, even if one of the three above conditions is violated. Other simulation results show that measurement uncertainties of the angle of attack α have no influence on the stability of the transfer function $q_{meas}/\dot{q}_{des}$ [18].

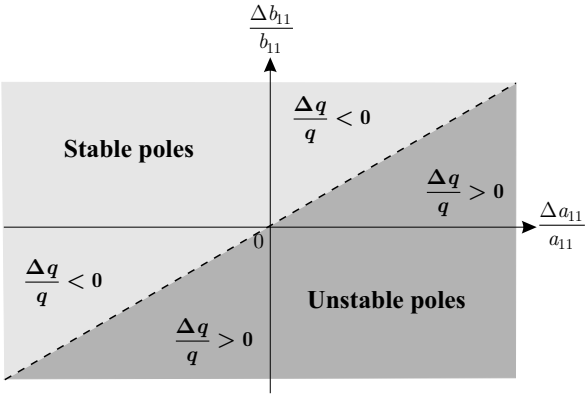


Fig. 7.11 Unstable vs. stable poles due to parameter uncertainties

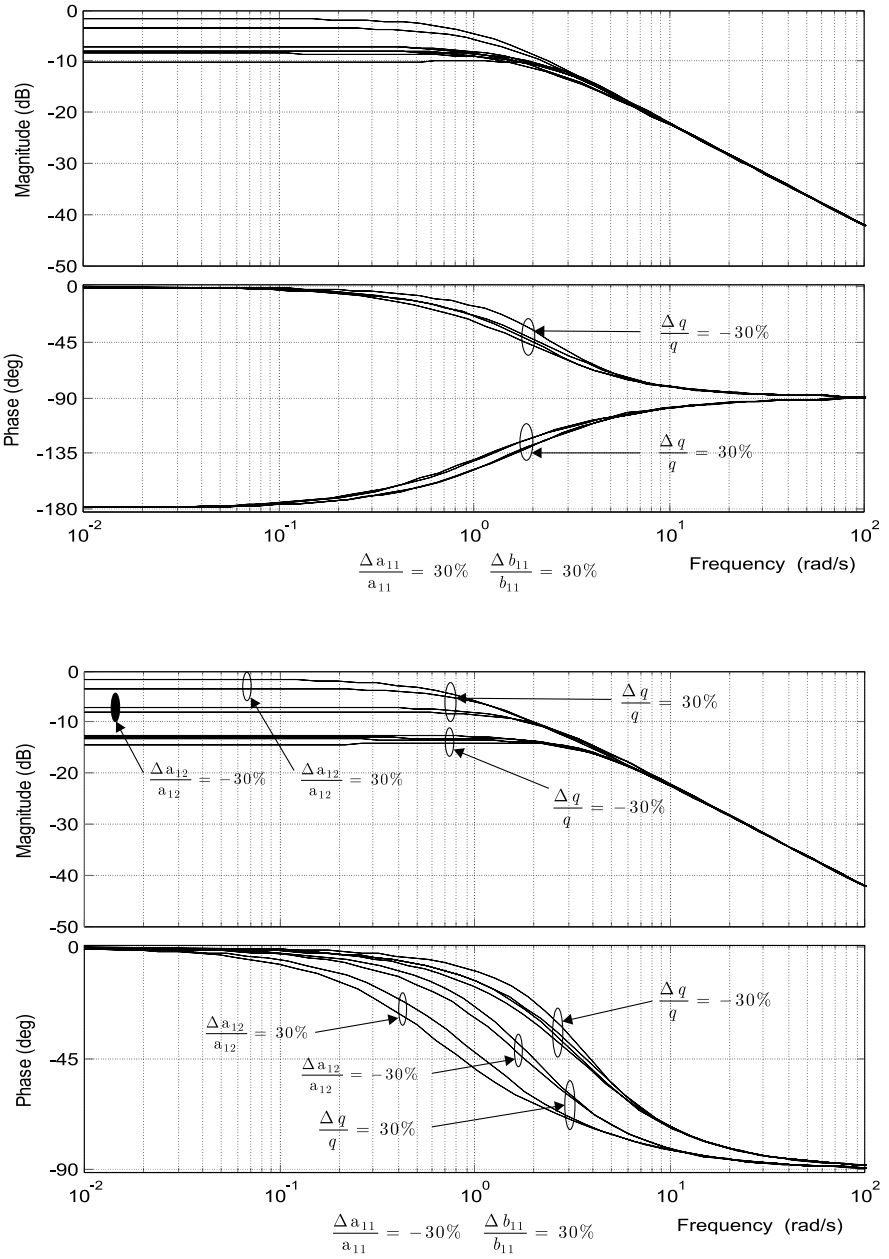


Fig. 7.12 Transfer function of the longitudinal plant including uncertain dynamic inversion from $\dot{q}_{des}$ to q_{meas}

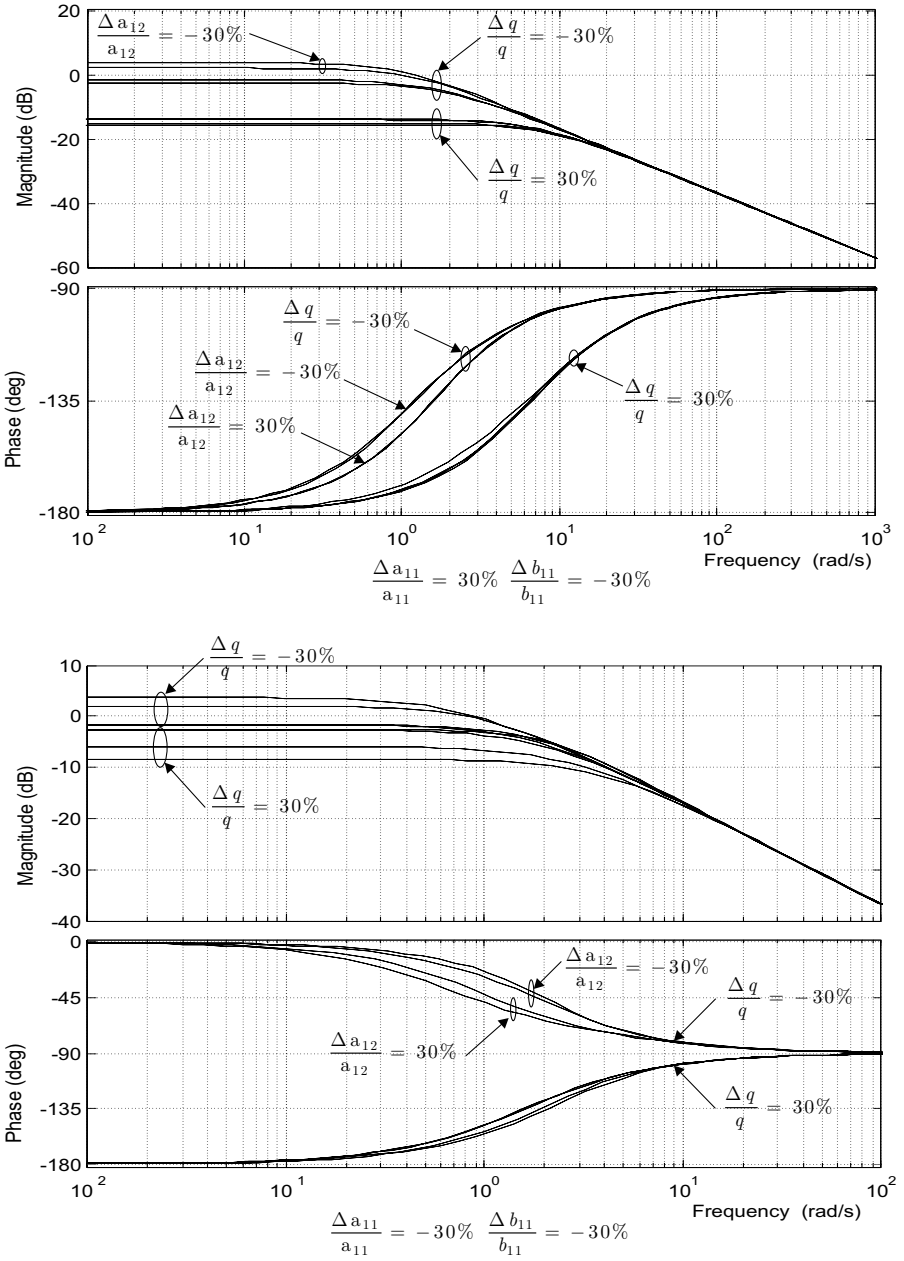


Fig. 7.13 Transfer function of the longitudinal plant including uncertain dynamic inversion from $\dot{q}_{des}$ to q_{meas}

7.4.7 Mathematical Selection of the Uncertain Model Parameters

Equation 7.27 can be written explicitly as follows:

$$\frac{q_{meas}}{\dot{q}_{des}} = \frac{\frac{b_{11}}{b_{11}+\Delta b_{11}}(s-a_{22})}{s^2+\Gamma s+\gamma}\,, \tag{7.35}$$

where the factor Γ is

$$\Gamma = -\left[a_{11} - \frac{b_{11}}{b_{11}+\Delta b_{11}}(a_{11}+\Delta a_{11})(c_{11}+\Delta c_{11}) + a_{22}\right]\,, \tag{7.36}$$

and the factor γ is

$$\begin{aligned}\gamma = \quad & \left[a_{11} - \frac{b_{11}}{b_{11}+\Delta b_{11}}(a_{11}+\Delta a_{11})(c_{11}+\Delta c_{11})\right] a_{22} \\ & - \left[a_{12} - \frac{b_{11}}{b_{11}+\Delta b_{11}}(a_{12}+\Delta a_{12})(c_{11}+\Delta c_{11})\right] a_{21}, \quad (\text{with } a_{21}=1).\end{aligned} \tag{7.37}$$

The term Γ is rearranged in order to make relative uncertainty terms appear of the form $\Delta a_{..}/a_{..}$ and is approximated by neglecting products of uncertainty terms:

$$\Gamma = -\left[a_{11} - \frac{b_{11}}{b_{11}+\Delta b_{11}}(a_{11}+\Delta a_{11})(c_{11}+\Delta c_{11}) + a_{22}\right]\,, \tag{7.38}$$

$$\Gamma = \frac{-a_{11}}{1+\frac{\Delta b_{11}}{b_{11}}}\left[1+\frac{\Delta b_{11}}{b_{11}} - \left(1+\frac{\Delta a_{11}}{a_{11}}\right)c_{11}\left(1+\frac{\Delta c_{11}}{c_{11}}\right)\right] - a_{22} \quad (c_{11}=1)\,.$$

Let us define $\Gamma' = \Gamma\left(1+\frac{\Delta b_{11}}{b_{11}}\right)$, the previous equation is rewritten as follows:

$$\begin{aligned}\Gamma' &= -a_{11}\left[1+\frac{\Delta b_{11}}{b_{11}} - \left(1+\frac{\Delta a_{11}}{a_{11}}+\frac{\Delta c_{11}}{c_{11}}\right)\right] - a_{22}\left(1+\frac{\Delta b_{11}}{b_{11}}\right)\,, \\ \Gamma' &= a_{11}\left[\frac{\Delta a_{11}}{a_{11}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right] - a_{22}\left(1+\frac{\Delta b_{11}}{b_{11}}\right)\,, \\ \Gamma' &= \alpha_1 - \alpha_2\,,\end{aligned} \tag{7.39}$$

with $\alpha_1 = a_{11}\left[\frac{\Delta a_{11}}{a_{11}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right]$, and $\alpha_2 = a_{22}\left(1+\frac{\Delta b_{11}}{b_{11}}\right)$.

The term γ is rearranged as follows:

$$
\begin{aligned}
\gamma = \quad & \frac{a_{11}a_{22}}{1+\frac{\Delta b_{11}}{b_{11}}}\left[1+\frac{\Delta b_{11}}{b_{11}}-\left(1+\frac{\Delta a_{11}}{a_{11}}\right)\left(1+\frac{\Delta c_{11}}{c_{11}}\right)\right] \\
& - \frac{a_{12}}{1+\frac{\Delta b_{11}}{b_{11}}}\left[1+\frac{\Delta b_{11}}{b_{11}}-\left(1+\frac{\Delta a_{12}}{a_{12}}\right)\left(1+\frac{\Delta c_{11}}{c_{11}}\right)\right] , \\
\gamma\left(1+\frac{\Delta b_{11}}{b_{11}}\right) = & - a_{11}a_{22}\left[\frac{\Delta a_{11}}{a_{11}}-\frac{\Delta b_{11}}{b_{11}}+\frac{\Delta c_{11}}{c_{11}}\right] \\
& + a_{12}\left[\frac{\Delta a_{12}}{a_{12}}-\frac{\Delta b_{11}}{b_{11}}+\frac{\Delta c_{11}}{c_{11}}\right] , \\
\gamma^{'} = & - a_{22}\alpha_1 + \alpha_3 , \qquad (7.40)
\end{aligned}
$$

with $\gamma^{'} = \gamma\left(1+\frac{\Delta b_{11}}{b_{11}}\right)$, and $\alpha_3 = a_{12}\left[\frac{\Delta a_{12}}{a_{12}}-\frac{\Delta b_{11}}{b_{11}}+\frac{\Delta c_{11}}{c_{11}}\right]$.

The poles of the transfer function in (7.35) are the roots of

$$s^2 + \Gamma s + \gamma = 0 . \qquad (7.41)$$

These roots are given by $s_{1,2} = \frac{-\Gamma \pm \sqrt{\Delta}}{2}$, where $\Delta = \Gamma^2 - 4\gamma$. In order that there are only stable poles in the system, the following two inequalities must be satisfied

$$-\Gamma + \sqrt{\Delta} < 0 \quad \text{and} \quad -\Gamma - \sqrt{\Delta} < 0 . \qquad (7.42)$$

The poles will lie in the left half plane if $\Gamma > 0$. Suppose that $\Delta > 0$, in which case the poles are real, then the conditions above are equivalent to

$$-\Gamma < \sqrt{\Delta} < \Gamma \quad \text{and} \quad \Gamma > 0 , \qquad (7.43)$$

which is again equivalent to

$$
\begin{aligned}
& \Delta < \Gamma^2 \quad \text{and} \quad \Gamma > 0 , \\
\Longleftrightarrow \; & \Gamma^2 - 4\gamma < \Gamma^2 \quad \text{and} \quad \Gamma > 0 , \\
\Longleftrightarrow \; & \Gamma > 0 \quad \text{and} \quad \gamma > 0 . \qquad (7.44)
\end{aligned}
$$

The first inequality in (7.44), namely $\Gamma > 0$, is satisfied if

$$
\begin{aligned}
& \Gamma^{'} > 0, \left(1+\frac{\Delta b_{11}}{b_{11}}\right) \text{ is always positive} , \\
\Longleftrightarrow \; & \alpha_1 - \alpha_2 > 0 , \\
\Longleftrightarrow \; & a_{11}\left[\frac{\Delta a_{11}}{a_{11}}-\frac{\Delta b_{11}}{b_{11}}+\frac{\Delta c_{11}}{c_{11}}\right] > a_{22}\left(1+\frac{\Delta b_{11}}{b_{11}}\right) , \\
\Longleftrightarrow \; & \left[\frac{\Delta a_{11}}{a_{11}}-\frac{\Delta b_{11}}{b_{11}}+\frac{\Delta c_{11}}{c_{11}}\right] < \frac{a_{22}}{a_{11}}\left(1+\frac{\Delta b_{11}}{b_{11}}\right), \quad (\mathrm{a}_{11} < 0) . \qquad (7.45)
\end{aligned}
$$

Since both of the coefficients a_{11} and a_{22} are negative, the inequality in (7.45) holds if

$$\left[\frac{\Delta a_{11}}{a_{11}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right] < 0\,, \tag{7.46}$$

which is always guaranteed if we choose

$$\frac{\Delta a_{11}}{a_{11}} < 0, \quad \frac{\Delta b_{11}}{b_{11}} > 0, \quad \text{and } \frac{\Delta \mathrm{c}_{11}}{\mathrm{c}_{11}} < 0\,. \tag{7.47}$$

The conditions in (7.47) for the selection of the uncertainty levels, which guarantee the existence of stables poles only are indeed the same as the conditions found from the simulation results (see (7.33) and (7.34)).

The second inequality in (7.44), namely $\gamma > 0$, is fulfilled if

$$\begin{aligned}
&\gamma^{'} > 0 \\
&\iff -a_{12}\left[\frac{\Delta a_{12}}{a_{12}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right] < -a_{11}a_{22}\left[\frac{\Delta a_{11}}{a_{11}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right], \\
&\iff \left[\frac{\Delta a_{12}}{a_{12}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right] < \frac{a_{11}a_{22}}{a_{12}}\left[\frac{\Delta a_{11}}{a_{11}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right], \ (a_{12} < 0) \\
&\iff \left[\frac{\Delta a_{12}}{a_{12}} - \frac{\Delta b_{11}}{b_{11}} + \frac{\Delta c_{11}}{c_{11}}\right] < 0\,.
\end{aligned} \tag{7.48}$$

The condition in (7.48) holds if in addition to the conditions in (7.47), we choose

$$\frac{\Delta a_{12}}{a_{12}} < 0\,. \tag{7.49}$$

If the term $\Delta = \Gamma^2 - 4\gamma$ is negative, then the poles will be complex conjugate. They will lie in the left half plane under the condition that $\Gamma > 0$, which is fulfilled by the conditions of (7.47).

The numerator of the transfer function in (7.35) has a negative zero if the term $a_{22} = \frac{\partial \dot{\alpha}}{\partial \alpha} < 0$. This is the case for the aircraft under consideration, whatever the level of uncertainty for this term.

7.5 General Control Architecture for the Longitudinal Motion

The longitudinal motion controller is inspired by [20] and consists of several loops, where every loop is designed assuming that the next inner loop achieves perfect tracking of the command signals. This condition is not satisfied if the next inner loop bandwidth is not sufficiently larger than the bandwidth of the outer loop. Therefore, the controllers will be designed to ensure reasonable time-scale separation.

The differences between the autopilot for the longitudinal motion of this chapter and the one in [20] are the following:

- The controllers in [20] for the tracking of roll-, pitch-, and yaw-rate commands are based on MRAC laws, where the matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ of the linear aircraft model are constantly estimated by an on-line stabilized RLS algorithm, which provides an adaptation of the matrix terms with the current operating point of the aircraft and/or the possible modification of the aircraft aerodynamics due to some damage or failures.

  In this book, the controllers for the turn rates use NDI, and therefore rely on a good knowledge of the plant parameters. Uncertainties on the plant parameters and on sensor data are taken into account during the design of the controllers of the turn-rate desired dynamics. These controllers are to be sufficiently robust to guarantee satisfactory stability and performance properties, as discussed in Sect. 7.6.
- In [20], the controllers for the angle of attack, altitude rate, and altitude commands, all consist of proportional compensators, whose gains are chosen for a satisfactory closed-loop pole placement in the ideal case.

  In this chapter, all the controllers used have the architecture presented in Chap. 6 either in Fig. 6.8 or Fig. 6.12. These controllers include PI gains together with feedforward paths of the command signal.
- In [20], although some simulation results are provided that demonstrate the robustness of the algorithm in the presence of external disturbance or sensor noise, no robustness analysis is presented. The rest of this chapter is dedicated to the stability and robustness analyses of the control systems in the presence of large parameter and measurement uncertainties.

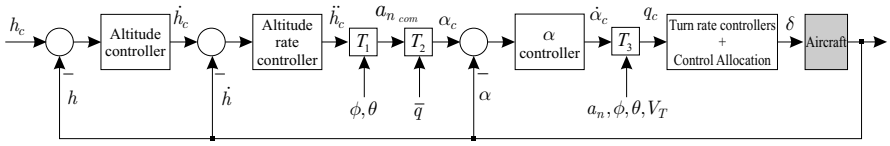


Fig. 7.14 General architecture for the control of the longitudinal motion

In Fig. 7.14, the blocks T_1, T_2, and T_3 represent transformations that are linear in the main input variable, but nonlinear in the other variables. These nonlinear transformations are derived in the Appendix C, [20].

7.5.1 Nonlinear Transformation T_3

The nonlinear transformation T_3 transforms the commanded rate of change for the angle of attack $\dot{\alpha}_c$ into the pitch rate command q_c as follows:

$$q_c = \dot{\alpha}_c - \frac{g}{V_T}(\cos\theta\cos\phi + a_n) \ . \tag{7.50}$$

7.5.2 Nonlinear Transformation T_2

The nonlinear transformation T_2 transforms the aircraft normal acceleration command $a_{n,com}$ into a command for the angle of attack α_c as follows:

$$\alpha_c = \frac{a_{n,com} - a_2\bar{q}}{a_1\bar{q}} \,, \tag{7.51}$$

where a_1 and a_2 are constant and defined in Appendix C.

7.5.3 Nonlinear Transformation T_1

The nonlinear transformation T_1 transforms the vertical acceleration command $\ddot{h}_c$ into the aircraft normal acceleration command $a_{n,com}$ as follows:

$$a_{n,com} = \frac{\ddot{h}_c/g - 1}{\cos\phi\cos\theta} \,. \tag{7.52}$$

7.6 Pitch Rate Control

Figure 7.15 shows the longitudinal plant linearized by the uncertain dynamic inversion process together with the controller of the pitch rate dynamics, which generates the control signal $\dot{q}_{des}$.

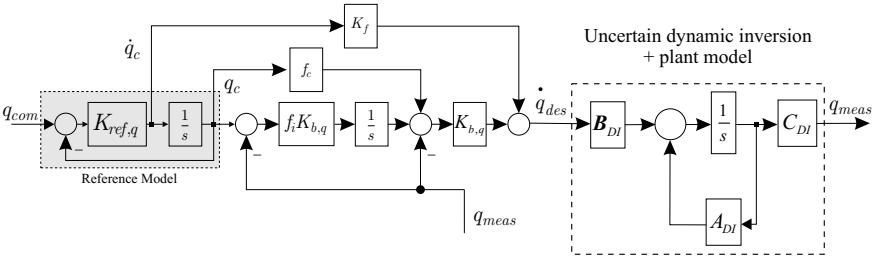


Fig. 7.15 Pitch rate controller with the plant that includes parameter and measurement uncertainties

In order to study the stability and the robustness of the system in the presence of model and measurement uncertainties, the open-loop gain of the plant and the pitch rate controller is computed with (6.18), which is recalled here for convenience:

$$L_e(s) = L_y(s) = \frac{G_q(s)}{1 + K_{b,q}\, G_q(s)} \frac{K_{b,q}^2 f_{i,q}}{s} \,, \tag{7.53}$$

$$\text{with } \mathrm{G_q(s)} = \frac{\mathrm{q_{meas}(s)}}{\mathrm{\dot{q}_{des}(s)}}$$
$$= \frac{\boldsymbol{C_{DI}}_{(1,1)} \boldsymbol{B_{DI}}_{(1,1)} (s - \boldsymbol{A_{DI}}_{(2,2)})}{s^2 - s(\boldsymbol{A_{DI}}_{(1,1)} + \boldsymbol{A_{DI}}_{(2,2)}) + \boldsymbol{A_{DI}}_{(1,1)} \boldsymbol{A_{DI}}_{(2,2)} - \boldsymbol{A_{DI}}_{(2,1)} \boldsymbol{A_{DI}}_{(1,2)}} \,. \tag{7.54}$$

A series of simulations are run, where the coefficients $\frac{\Delta a_{11}}{a_{11}}$, $\frac{\Delta a_{12}}{a_{12}}$, $\frac{\Delta b_{11}}{b_{11}}$, $\frac{\Delta c_{11}}{c_{11}}$, and $\frac{\Delta c_{22}}{c_{22}}$ are successively equal to $+30\,\%$ and $-30\,\%$ in a nested for-loops algorithm. The gains chosen for the pitch controller are $K_{b,\,q} = 12\,[\mathrm{s}^{-1}]$ and $f_{i,\,q} = 0.25$. The results are shown in Figs. 7.17 to 7.20.

7.6.1 Stability/Robustness Requirements

The stability of the system is assessed by analyzing the root locus and Nyquist plots provided for each example. Additional requirements are formulated for the shape of the Bode plot of the open-loop gain as shown in Fig. 7.16a. Phase and gain margins are defined by the diamond region shown in Fig. 7.16b. In the Nichols chart, if the curve of the loop gain avoids the diamond region, the phase and gain margins are guaranteed [16]. The two most important uncertainty terms for the pitch rate control loop are $\frac{\Delta a_{11}}{a_{11}}$ and $\frac{\Delta b_{11}}{b_{11}}$. Therefore, four series of plots are provided in Fig. 7.17 to Fig. 7.20. For each series, one value of these two uncertainty terms is modified at a time, whereas all the other uncertainty terms are varied.

Figure 7.17 shows the results obtained when $\frac{\Delta a_{11}}{a_{11}} = +30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = +30\,\%$. Clearly, the requirements of the loop gain Bode plot are fulfilled. The loop gain also avoids the diamond region in the Nichols chart, therefore 45° phase and 5 dB gain margins are guaranteed. The root locus plot shows that right half-plane poles and zeros cancel each other and that the loop gain can be safely increased without destabilizing the system. There is no unstable pole. Finally, the Nyquist plot shows that the critical -1 point is safely avoided (no right encirclement [21]) by the loop gain in all cases.

Depending on the uncertainty configuration, the bandwidth of the controller may be significantly affected. The requirement on the bandwidth is that the closed-loop bandwidth of the system should be higher than the frequency of the short-period mode $\omega_{sp} = 4.72\,\mathrm{rad/s}$. The top-left plot in Fig. 7.21 shows that a gain of $K_{b,q} = 12\,[\mathrm{s}^{-1}]$ is sufficient in the case $\frac{\Delta a_{11}}{a_{11}} = \frac{\Delta b_{11}}{b_{11}} = +30\,\%$.

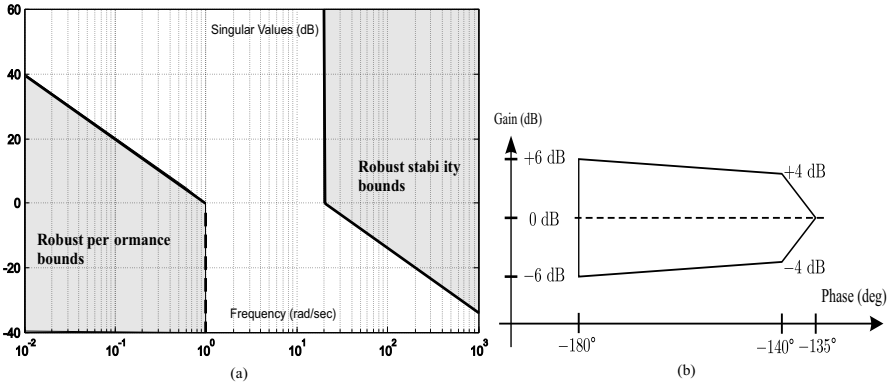


Fig. 7.16 (a) Requirements for the shape of the loop gain Bode plot; and (b) diamond region

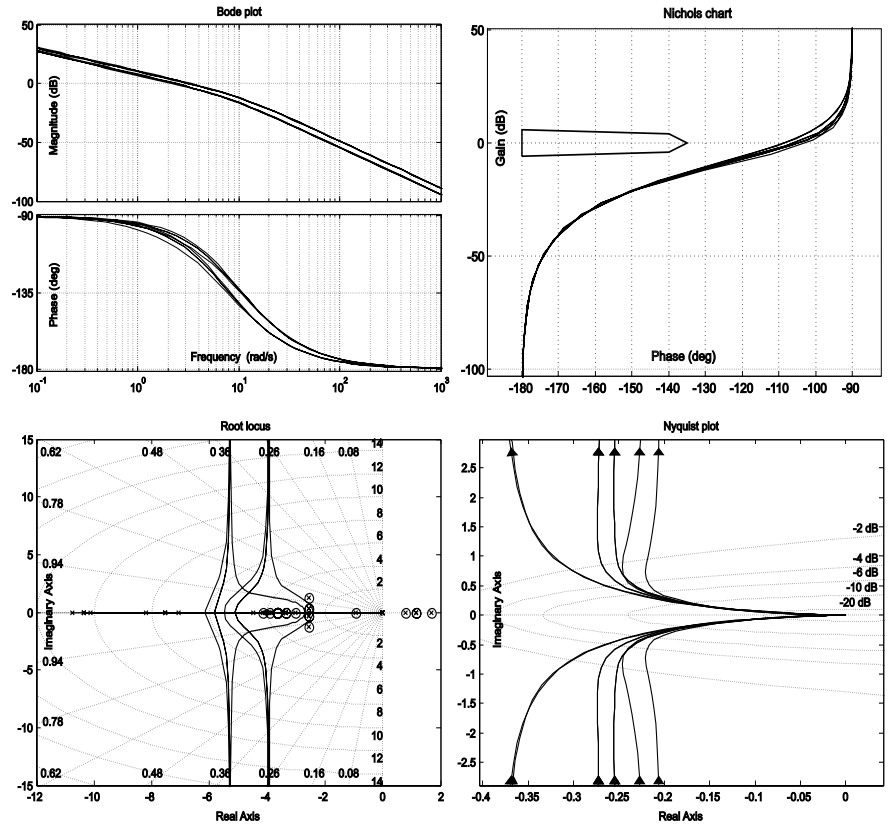


Fig. 7.17 Open-loop gain of the pitch controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = +30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = +30\,\%$

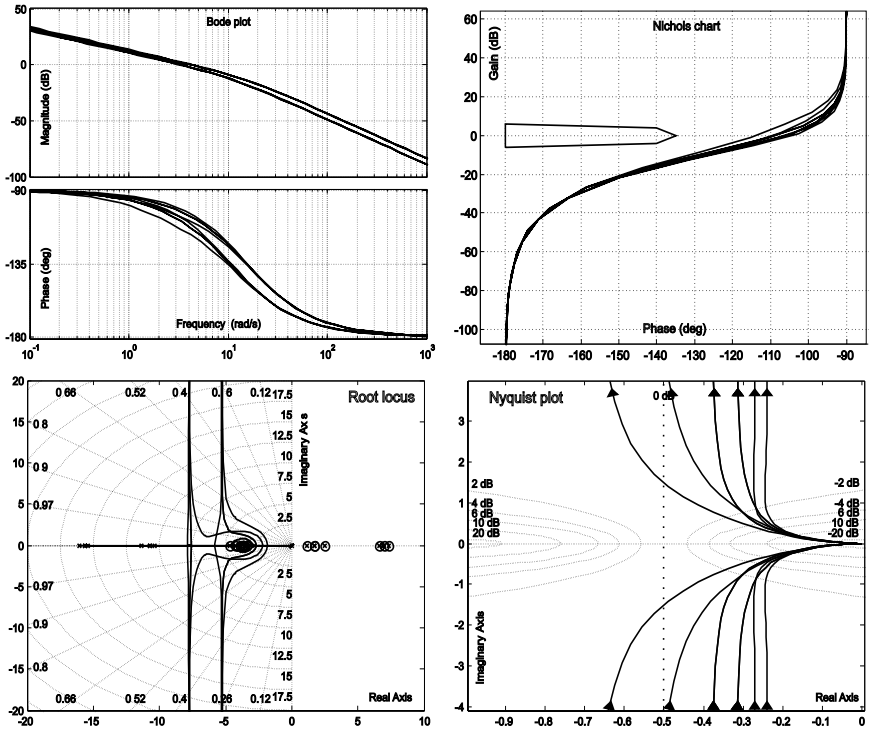


Fig. 7.18 Open-loop gain of the pitch controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = +30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = -30\,\%$

Figure 7.18 shows the results obtained in the case $\frac{\Delta a_{11}}{a_{11}} = +30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = -30\,\%$. The Bode plot and Nichols chart requirements are fulfilled. The root locus plot shows the existence of several right half-plane pairs of poles and zeros cancelling each other. There is no unstable pole. The Nyquist plot shows that no right encirclement of the -1 point occurs. The closed-loop Bode plot is shown in Fig. 7.21.

Figure 7.19 shows the results obtained in the case $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = +30\,\%$. The Bode plot and Nichols chart requirements are fulfilled. The root locus plot shows that there is no right half-plane pole or zero. The Nyquist plot shows that no right encirclement of the -1 point occurs.

The closed-loop Bode plot in Fig. 7.21 shows that the bandwidth ω_c of the controlled system may be just enough ($\omega_c \approx \omega_{sp}$) for some uncertainty configuration on the parameters $a_{12},\ c_{11},\ c_{22}$. Therefore, a slightly larger value of $K_{b,q}$ may be required in order to ensure $\omega_c > \omega_{sp}$. This uncertainty configuration ($\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = +30\,\%$) is the one suggested in Sect. 7.4.6 for the selection of the plant parameters used in the dynamic inversion process. It clearly gives the best results in terms of stability/robustess of the system compared with the other plant uncertainty configurations.

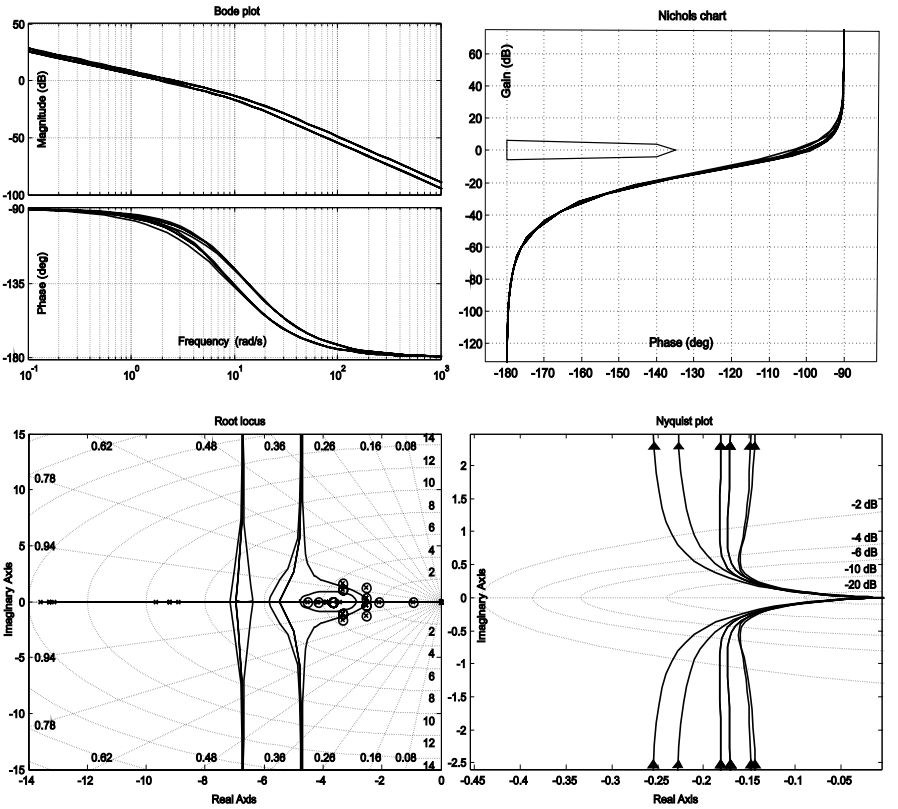


Fig. 7.19 Open-loop gain of the pitch controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = +30\,\%$

Figure 7.20 shows the results obtained in the case $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = -30\,\%$. The Bode plot and Nichols chart requirements are fulfilled. The root locus plot shows the existence of several right half-plane pairs of poles and zeros cancelling each other. There is no unstable pole. The Nyquist plot shows that no right encirclement of the -1 point occurs. The closed-loop Bode plot is shown in Fig. 7.21.

Simulation results showed that the controller for the pitch rate desired dynamics is able to meet the requirements for stability and robustness of the system for all the different uncertainty configurations presented. The configuration where dynamic inversion operates with $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = 30\,\%$ is clearly the safest case. Therefore, in practice the parameter a_{11} is to be intentionally estimated too low, whereas b_{11} will be intentionally estimated too high.

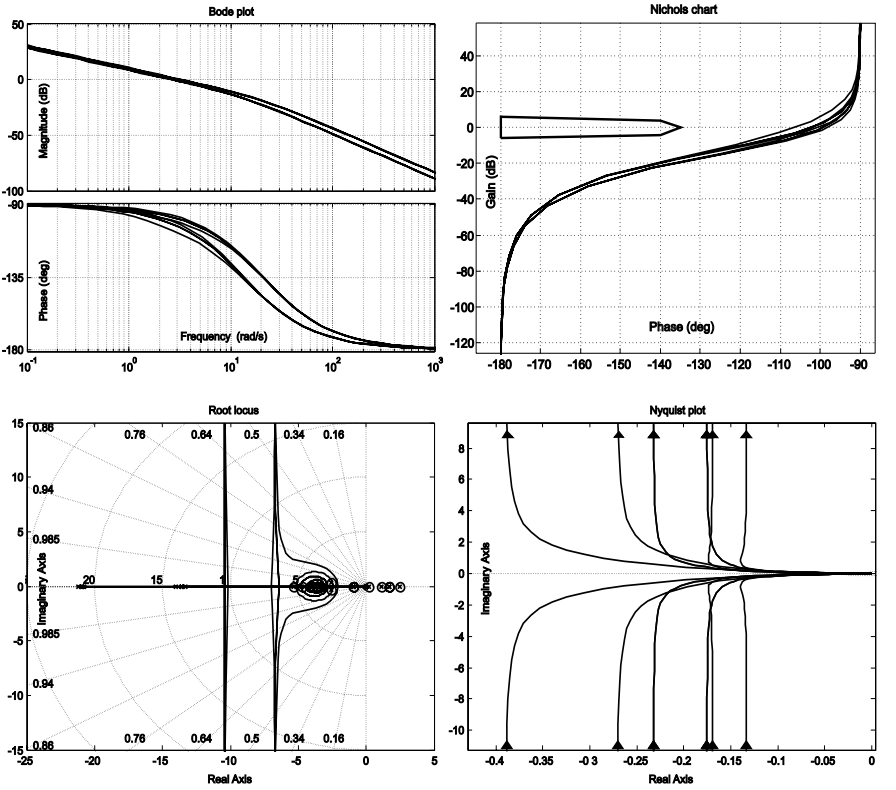


Fig. 7.20 Open-loop gain of the pitch controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = -30\,\%$

7.6.2 Pitch Rate Closed-loop Transfer Function

The closed-loop transfer function q_{meas}/q_{com} is obtained as follows:

$$\begin{aligned}\frac{q_{meas}}{q_{com}} &= \frac{q_{meas}}{q_c} \cdot \frac{q_c}{q_{com}} \,, \\ &= \frac{G_q \left[K_f s^2 + K_{b,q}(f_c s + f_i K_{b,q})\right]}{s + G_q\, K_{b,q}(s + f_i K_{b,q})} \cdot \frac{K_{ref,q}}{s + K_{ref,q}} \,, \end{aligned} \tag{7.55}$$

where

$$\begin{aligned} G_q &= \frac{q_{meas}}{\dot{q}_{des}} \,, \\ &= \frac{\boldsymbol{C_{DI}}_{(1,1)} \boldsymbol{B_{DI}}_{(1,1)} (s - \boldsymbol{A_{DI}}_{(2,2)})}{s^2 - s(\boldsymbol{A_{DI}}_{(1,1)} + \boldsymbol{A_{DI}}_{(2,2)}) + \boldsymbol{A_{DI}}_{(1,1)} \boldsymbol{A_{DI}}_{(2,2)} - \boldsymbol{A_{DI}}_{(2,1)} \boldsymbol{A_{DI}}_{(1,2)}} \,. \end{aligned} \tag{7.56}$$

The Bode plots of the closed-loop transfer function in (7.56) are shown in Fig. 7.21 for several uncertainty configurations. Clearly, the closed-loop bandwidth is affected by the uncertainty of the pitch rate measurement $\Delta q/q$. For negative values of the uncertainty $\Delta q/q$ the nominal bandwidth is reduced, whereas for positive values of the uncertainty $\Delta q/q$ the nominal bandwidth is increased. Section 7.4.7 demonstrated that the pitch rate measurement should be intentionally estimated too low to guarantee stability of the control system, which corresponds to the lower-bandwidth plots shown in Fig. 7.21. The gain $K_{b,q}$ should be chosen such that the closed-loop bandwidth ω_{cl} is larger than the short-period mode frequency $\omega_{sp} = 4.72\,\mathrm{rad/s}$. Figure 7.21 shows that ω_{cl} is larger than or equal to ω_{sp} in all cases, which confirms that the value of $K_{b,q} = 12\,[\mathrm{s}^{-1}]$ is sufficiently large.

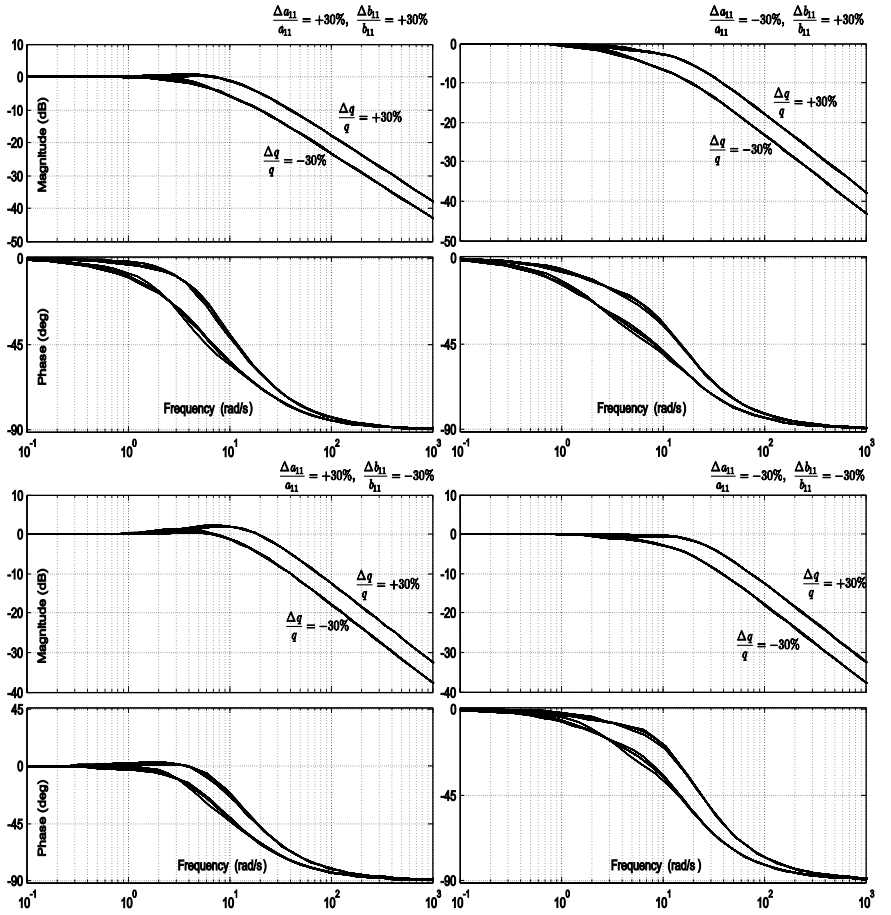


Fig. 7.21 Bode plot of the closed-loop transfer function from q_{com} to q_{meas} for several uncertainty configurations

7.7 Angle-of-attack Control Loop

The controller for the angle of attack is shown in Fig. 7.22. The dynamics of the angle of attack are described by the linear model in (7.57), where the numerical values of the matrices are shown.

$$\begin{bmatrix} \dot{q} \\ \dot{\alpha} \end{bmatrix} = \begin{bmatrix} -4.779 & -4.542 \\ 1 & -3.698 \end{bmatrix} \begin{bmatrix} q \\ \alpha \end{bmatrix} + \begin{bmatrix} 27.41 & 0 \\ 0 & -1.1 \times 10^{-4} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_{Thrust} \end{bmatrix} . \quad (7.57)$$

It is assumed that the variations of δ_{Thrust} are small. Moreover, the thrust force δ_{Thrust} modifies the dynamics of the angle of attack through the term $b_{22} = -1.1 \times 10^{-4}$, which is negligible compared to a_{21}, a_{22}, or b_{11}. Consequently, the dynamics of the angle of attack $\dot{\alpha}$ are considered to be only dependent on the pitch rate q and on the angle of attack α as shown in Fig. 7.22.

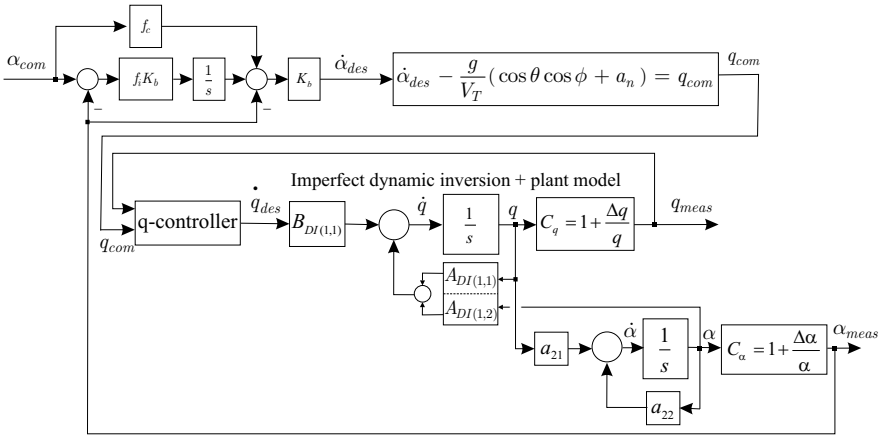


Fig. 7.22 Dynamics of the angle of attack

The control loop of the pitch rate is closed, and the transfer function α_{meas}/q_{com} is computed as follows:

$$G_\alpha = \frac{\alpha_{meas}}{q_{com}} = \frac{\alpha_{meas}}{q_{meas}} \cdot \frac{q_{meas}}{q_{com}} = \frac{C_\alpha a_{21} C_q^{-1}}{s - a_{22}} \cdot \frac{q_{meas}}{q_{com}} , \quad (7.58)$$

where q_{meas}/q_{com} is defined in (7.55), and where the measurement coefficient of the angle of attack C_α is defined as follows:

$$C_\alpha = 1 + \frac{\Delta\alpha}{\alpha} . \quad (7.59)$$

A simplified version of Fig. 7.22 is shown in Fig. 7.23.

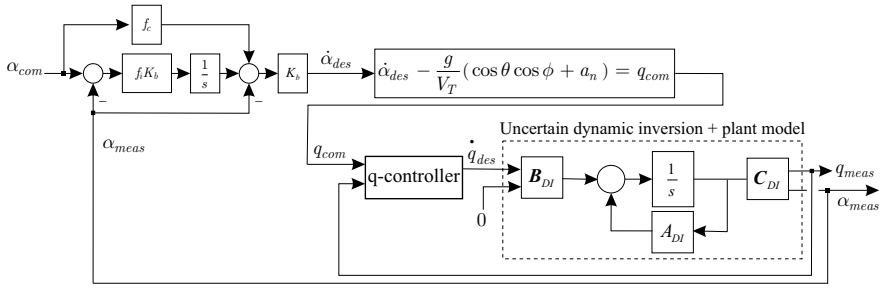


Fig. 7.23 Angle-of-attack controller

7.7.1 Open-loop and Closed-loop Gains

The open-loop gain is

$$L_{e,\alpha} = \frac{G_\alpha}{1 + G_\alpha\, K_{b,\alpha}} \cdot \frac{K_{b,\alpha}^2\; f_{i,\alpha}}{s}\;. \tag{7.60}$$

The gains are selected to be $f_{i,\alpha} = 0.25$, $f_c = 0.5$, and $K_{b,\alpha} = 6\,[\mathrm{s}^{-1}]$. Simulation results are shown in Fig. 7.24 to Fig. 7.27 for several uncertainty configurations of the terms $a_{11},\ a_{12},\ b_{11}\ , C_q$, and C_α. The closed-loop transfer function is:

$$\frac{\alpha_{meas}}{\alpha_{com}} = \frac{G_\alpha \left(K_{b,\alpha}\; f_c s + f_{i,\alpha}\; K_{b,\alpha}^2\right)}{s(1 + G_\alpha\; K_{b,\alpha}) + G_\alpha\; f_{i,\alpha}\; K_{b,\alpha}^2}\;. \tag{7.61}$$

7.7.2 Comments on the Results

An excessive value of $K_{b,\alpha}$ can destabilize the system in all cases (time-scale separation issue with the pitch rate loop). In all cases, there is no unstable closed-loop pole, and the phase and gain margins defined by the diamond region in the Nichols chart are guaranteed. The Bode plots of the closed-loop gain show that the closed-loop bandwidth is very sensitive to the uncertainty of the pitch rate measurement $\Delta q/q$ and to the uncertainty of the angle-of-attack measurement $\Delta\alpha/\alpha$. The largest bandwidth is obtained for $\frac{\Delta q}{q} < 0$ and $\frac{\Delta\alpha}{\alpha} > 0$ simultaneously. Conversely, the smallest bandwidth is obtained for $\frac{\Delta q}{q} > 0$ and $\frac{\Delta\alpha}{\alpha} < 0$ simultaneously. In Sect. 7.4.6, it was already suggested to underevaluate the value of the pitch rate measurement q_{meas}. For the dynamic inversion process, it would be wiser to select an underestimated value for the measurement of α_{meas}, *i.e.*, $\alpha = \alpha_{meas}(1 - |\Delta\alpha/\alpha|)$. These two suggestions used in the dynamic inversion process result in an intermediate

and reasonable closed-loop bandwidth. The configuration where NDI operates with $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = 30\,\%$ is still clearly the safest case.

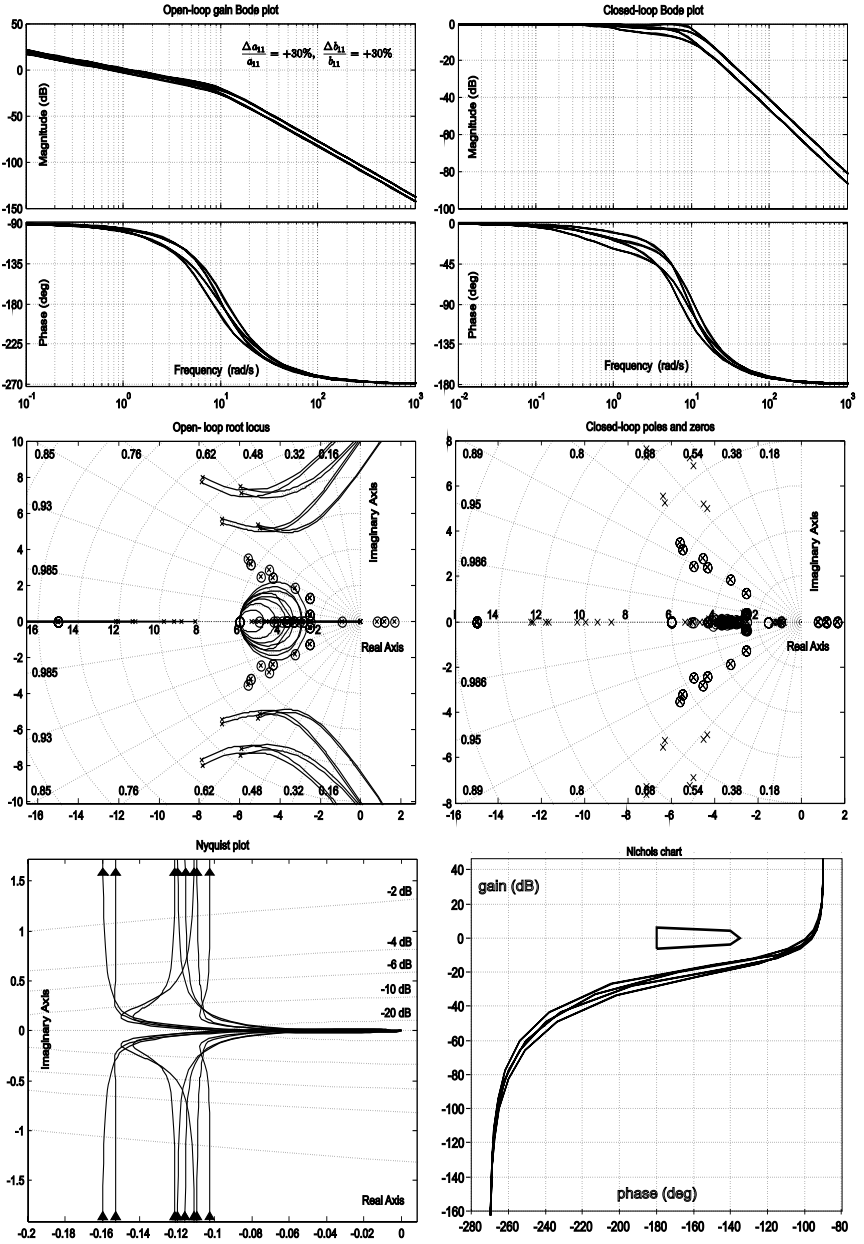


Fig. 7.24 Angle-of-attack controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = +30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = +30\,\%$

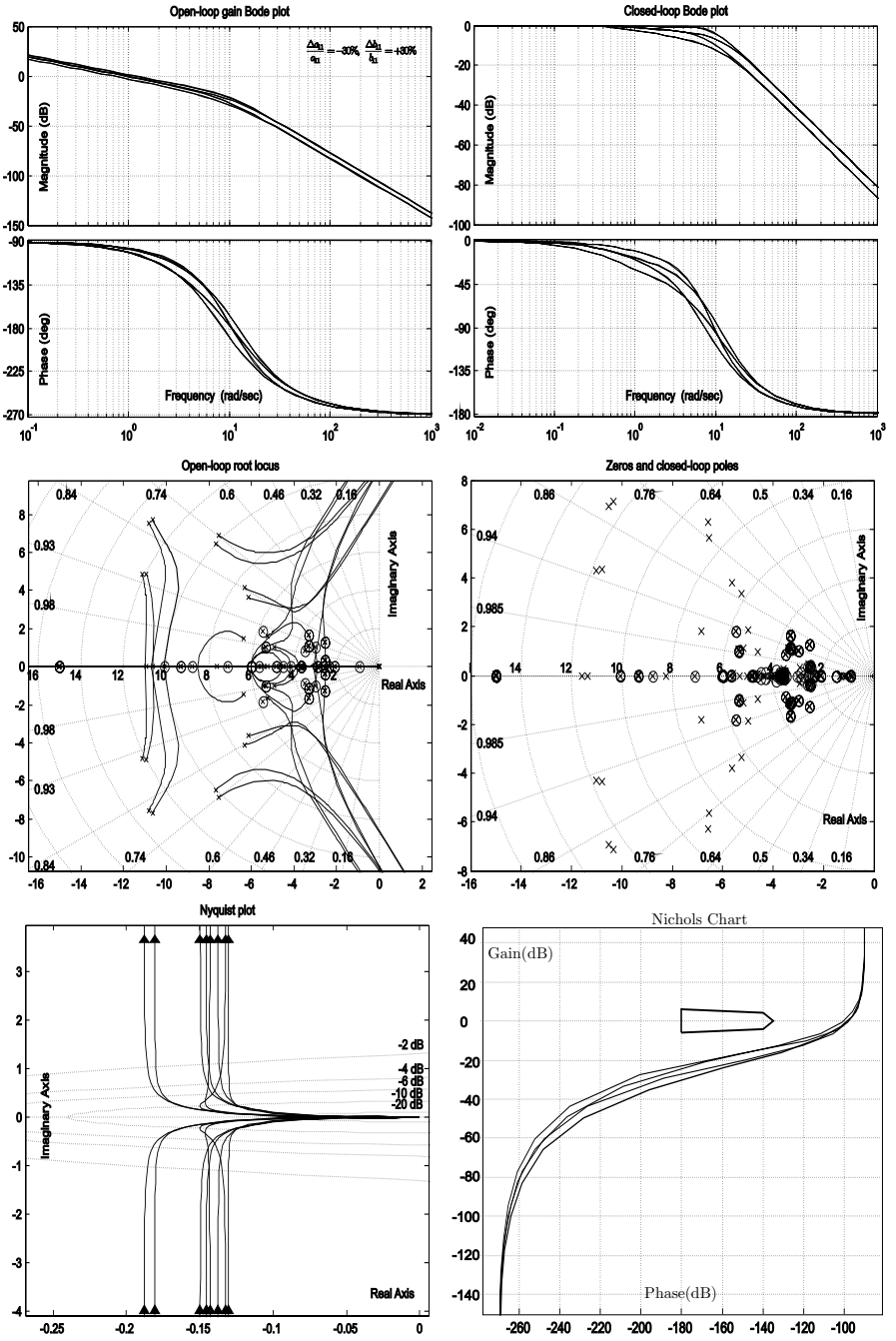


Fig. 7.25 Angle-of-attack controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = +30\,\%$

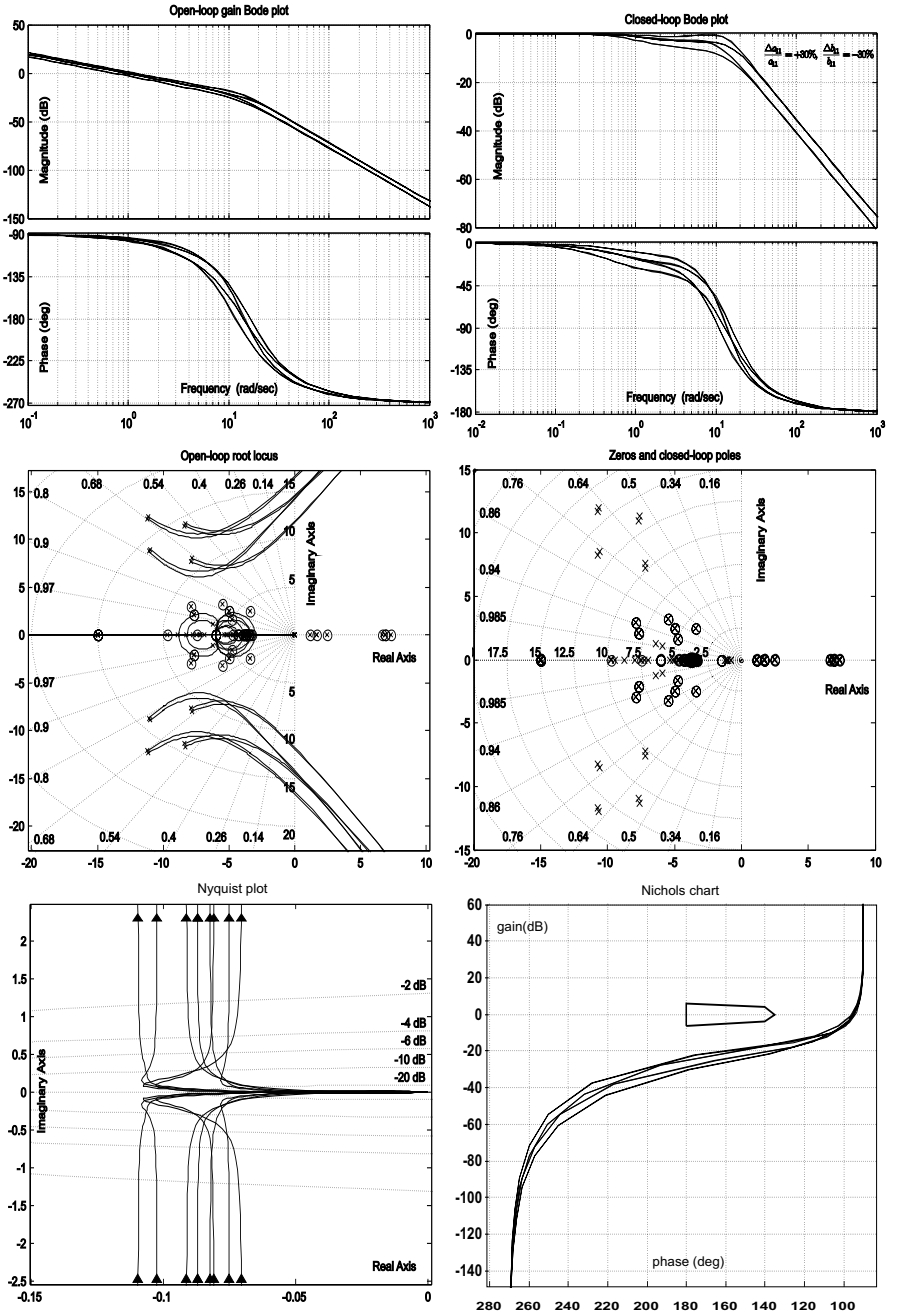


Fig. 7.26 Angle-of-attack controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = +30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = -30\,\%$

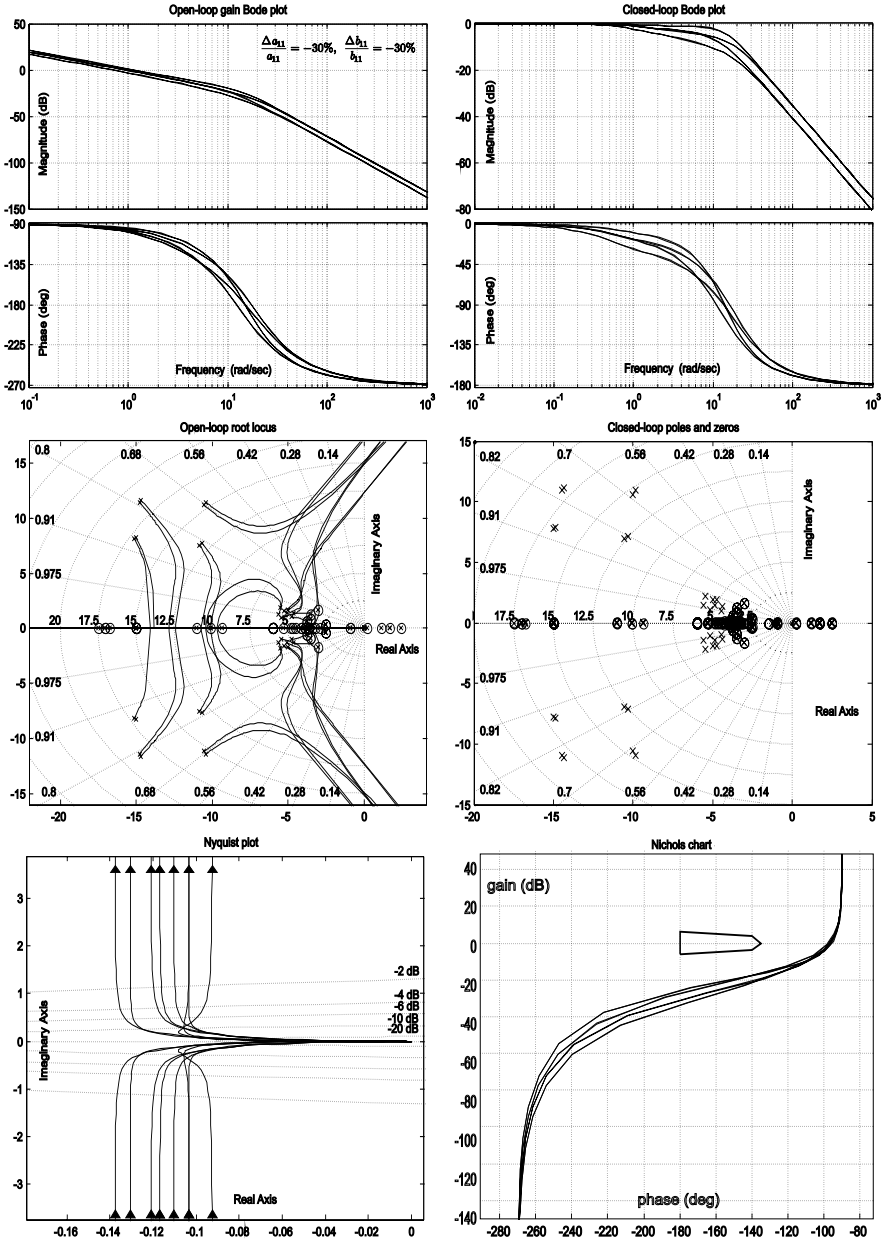


Fig. 7.27 Angle-of-attack controller and the plant including model parameter and measurement uncertainties, $\frac{\Delta a_{11}}{a_{11}} = -30\,\%$ and $\frac{\Delta b_{11}}{b_{11}} = -30\,\%$

7.8 Rate-of-climb Controller

The flight path angle γ, the pitch angle θ, and the angle of attack α are related as follows:

$$\theta = \alpha + \gamma \,. \tag{7.62}$$

The rate of climb $\dot{h}$ is defined as follows:

$$\dot{h} = V_T \sin\gamma \,. \tag{7.63}$$

The linear system has been computed around nominal flying conditions which correspond to straight and level flight. This means that the flight path angle is $\gamma \approx 0$. Therefore, the difference between the pitch angle and the angle of attack is small, *i.e.*, $\theta - \alpha \approx 0$. The rate of climb can therefore be approximated by

$$\dot{h} \approx V_T \cdot (\theta - \alpha) \,. \tag{7.64}$$

The pitch angle θ is obtained by time integration of the pitch rate q as follows:

$$\theta = \frac{q}{s} \,. \tag{7.65}$$

The angle of attack α may be expressed in terms of the pitch rate q with

$$q = \frac{s - a_{22}}{a_{21}}\,\alpha \,, \tag{7.66}$$

where the terms a_{21} and a_{22} are those defined in (7.15) and therefore

$$\dot{h} \approx V_T \cdot (\theta - \alpha) = V_T \left(\frac{s - a_{22}}{a_{21} s}\,\alpha - \alpha \right) = V_T \frac{s(1 - a_{21}) - a_{22}}{a_{21} s}\,\alpha \,. \tag{7.67}$$

Finally, a model of the altitude rate is obtained as follows:

$$\dot{h}_{meas} \approx C_{\dot{h}} V_T \frac{s(1 - a_{21}) - a_{22}}{a_{21} s}\,\alpha \,, \tag{7.68}$$

where the term $C_{\dot{h}}$ contains the uncertainty in the altitude rate measurement as follows (also see Fig. 7.28):

$$C_{\dot{h}} = 1 + \frac{\Delta\dot{h}}{\dot{h}} \,. \tag{7.69}$$

The nonlinear transformation between the desired vertical acceleration $\ddot{h}_{des}$ and the commanded angle of attack α_{com} as shown in Fig. 7.28 is (see Appendix C)

$$\alpha_{com} = \ddot{h}_{des} \cdot \frac{1}{\xi_1\,\bar{q}\,g\,\cos\phi\cos\theta} - \left(\frac{1}{\xi_1\,\bar{q}\,\cos\phi\cos\theta} + \frac{\xi_2}{\xi_1} \right) \,, \tag{7.70}$$

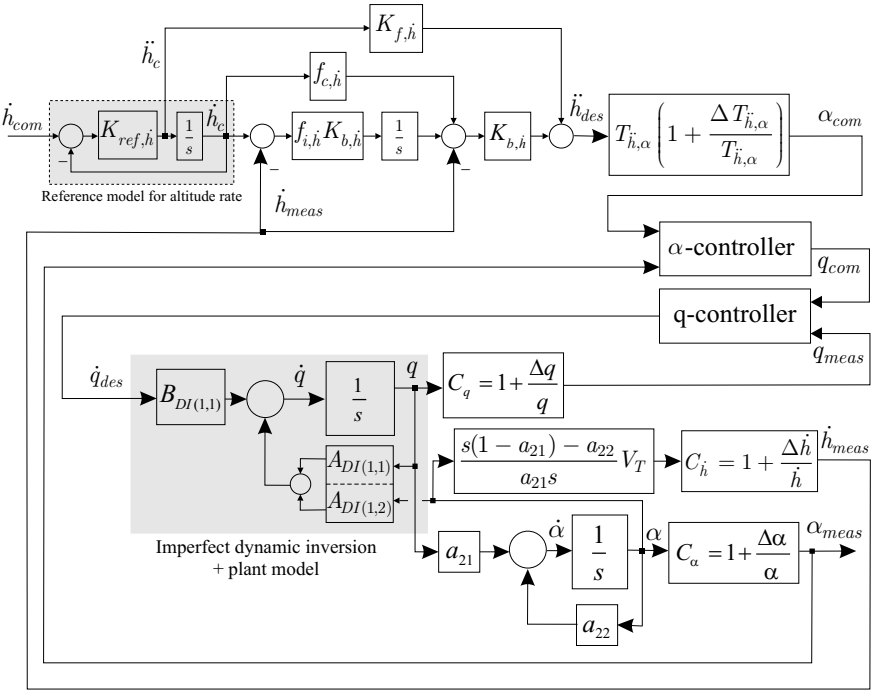


Fig. 7.28 Rate-of-climb controller

with

$$\xi_1 = \frac{S(C_{X1} + C_{Z\alpha})}{mg} \approx \frac{SC_{Z\alpha}}{mg}, \quad (C_{X1} \ll C_{Z\alpha})$$
$$\xi_2 = \frac{SC_{Z1}}{mg}\,. \tag{7.71}$$

Therefore, the term $T_{\ddot{h},\alpha} = 1/\left(\xi_1\,\bar{q}\,g\,\cos\phi\cos\theta\right)$ is acting as a gain and should not be forgotten in the computation of the loop gain.

The uncertainty associated with the gain $T_{\ddot{h},\alpha}$ is computed as follows:

$$\left|\frac{\Delta T_{\ddot{h},\alpha}}{T_{\ddot{h},\alpha}}\right| \approx \left|\frac{\Delta\xi_1}{\xi_1}\right| + \left|\frac{\Delta\bar{q}}{\bar{q}}\right| + \left|\frac{\Delta\phi}{\phi}\right| + \left|\frac{\Delta\theta}{\theta}\right|$$
$$\approx 30\,\% + 10\,\% + 5\,\% + 5\,\% = 50\,\%\,. \tag{7.72}$$

The value of the controller gains shown in Fig. 7.28 are $K_{b,\dot{h}} = 1\,[\mathrm{s}^{-1}]$, $f_{i,\dot{h}} = 0.25$, $f_{c,\dot{h}} = 1$, and $K_{f,\dot{h}} = 1$. Although not represented in the figures, the angle-of-attack command α_{com} is saturated between 0° and 13°, and an anti-windup system is implemented in the altitude rate controller.

7.8.1 Open-loop Gain

The open-loop gain is computed as follows:

$$L_e = \frac{G_{\dot{h}}}{1 + G_{\dot{h}} K_{b,\dot{h}}} \cdot \frac{K_{b,\dot{h}}^2 f_{i,\dot{h}}}{s} \,, \tag{7.73}$$

where

$$G_{\dot{h}} = \frac{\dot{h}_{meas}}{\ddot{h}_{des}} = T_{\ddot{h},\alpha} \left(1 + \frac{\Delta T_{\ddot{h},\alpha}}{T_{\ddot{h},\alpha}} \right) \cdot \frac{\alpha_{meas}}{\alpha_{com}} \cdot \frac{\alpha}{\alpha_{meas}} \cdot \frac{\dot{h}_{meas}}{\alpha} \,, \tag{7.74}$$

where the transfer function $\alpha_{meas}/\alpha_{com}$ is defined in (7.61), and the ratio $\alpha/\alpha_{meas} = C_\alpha^{-1}$ is defined in (7.59) and shown in Fig. 7.28. The ratio $\dot{h}_{meas}/\alpha$ is defined in (7.68).

Figure 7.29 shows the open-loop gains obtained when the parameters $\frac{\Delta a_{11}}{a_{11}}$, $\frac{\Delta a_{12}}{a_{12}}$, $\frac{\Delta b_{11}}{b_{11}}$, $\frac{\Delta q}{q}$, and $\frac{\Delta \alpha}{\alpha}$ take on alternatively the values $+30\,\%$ and $-30\,\%$ in a nested for-loop algorithm. The term $\frac{\Delta T_{\ddot{h},\alpha}}{T_{\ddot{h},\alpha}}$ takes on alternatively the values $+50\,\%$ and $-50\,\%$, whereas the term $\frac{\Delta \dot{h}}{\dot{h}}$ takes on alternatively the values $+5\,\%$ and $-5\,\%$.

In all cases, the system is asymptotically stable and the bandwidth of the open loop is very weakly affected by the uncertainties and is around 0.8 rad/s. The lower the gain $K_{b,\dot{h}}$, the weaker is the influence of the plant and measurement uncertainties on the bandwidth of the altitude rate control loop.

7.8.2 Closed-loop Gain

The closed-loop gain is computed as follows:

$$\begin{aligned} \frac{\dot{h}_{meas}}{\dot{h}_{com}} &= \frac{\dot{h}_{meas}}{\dot{h}_c} \cdot \frac{\dot{h}_c}{\dot{h}_{com}} \,, \\ &= \frac{G_{\dot{h}} \left[K_{f,\dot{h}} s^2 + K_{b,\dot{h}} (f_{c,\dot{h}} s + f_{i,\dot{h}} K_{b,\dot{h}}) \right]}{s + G_{\dot{h}} K_{b,\dot{h}} (s + f_{i,\dot{h}} K_{b,\dot{h}})} \cdot \frac{K_{ref,\dot{h}}}{s + K_{ref,\dot{h}}} \,, \end{aligned} \tag{7.75}$$

with

$$G_{\dot{h}} = T_{\ddot{h},\alpha} \left(1 + \frac{\Delta T_{\ddot{h},\alpha}}{T_{\ddot{h},\alpha}} \right) \cdot \frac{\alpha_{meas}}{\alpha_{com}} \cdot \frac{\alpha}{\alpha_{meas}} \cdot \frac{\dot{h}_{meas}}{\alpha} \,. \tag{7.76}$$

The gains used in the altitude rate reference model are $K_{ref,\dot{h}} = 1\,[\mathrm{s}^{-1}]$ and $K_{f,\dot{h}} = 1$. The choice of $K_{b,\dot{h}} = 1\,[\mathrm{s}^{-1}]$ yields a closed-loop bandwidth

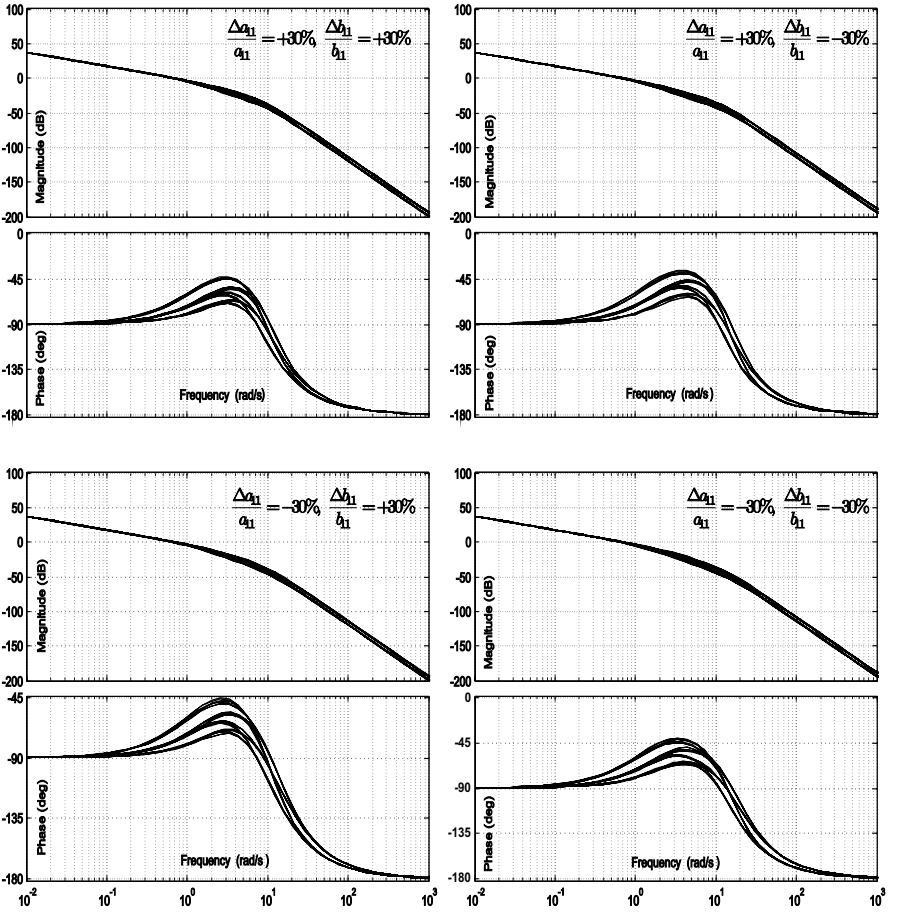


Fig. 7.29 Bode plots of the open-loop gains of the altitude rate controller for several uncertainty configurations

in the range of 0.6 to 1 rad/s depending on the uncertainty configuration as shown in Fig. 7.30.

7.9 Altitude Controller

The altitude controller is shown in Fig. 7.31. The measured altitude is obtained as follows:

$$h_{meas} \approx C_h \frac{\dot{h}}{s} \,, \tag{7.77}$$

where the uncertain coefficient for the altitude measurement is defined as

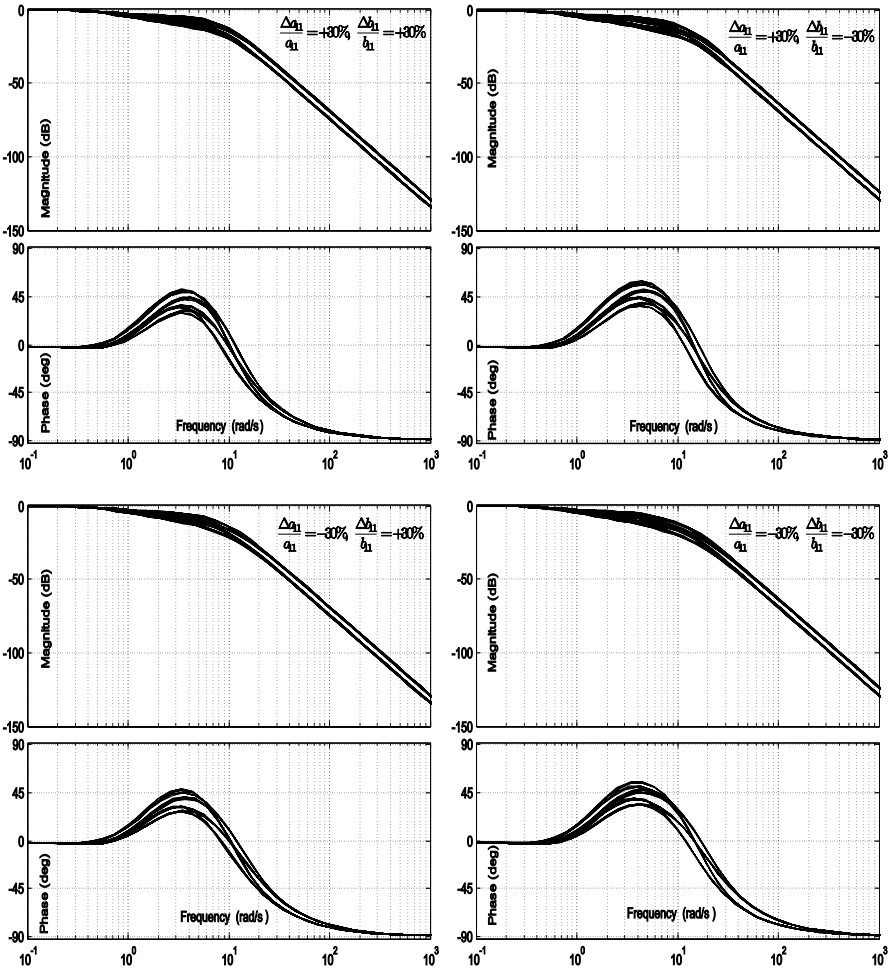


Fig. 7.30 Bode plots of the closed-loop gains of the altitude rate controller for several uncertainty configurations

$$C_h = 1 + \frac{\Delta h}{h} \ . \tag{7.78}$$

The gains of the altitude controller are $K_{ref,h} = 0.3\,[\mathrm{s}^{-1}]$, $f_{c,h} = 1$, $K_{f,h} = 1$, $K_{b,h} = 0.2\,[\mathrm{s}^{-1}]$, and $f_{i,h} = 0.25$.

The altitude rate command $\dot{h}_{com}$ is saturated to stay within the range $[-V_T/4,\ +V_T/4]$, and an anti-windup system is implemented in the altitude controller.

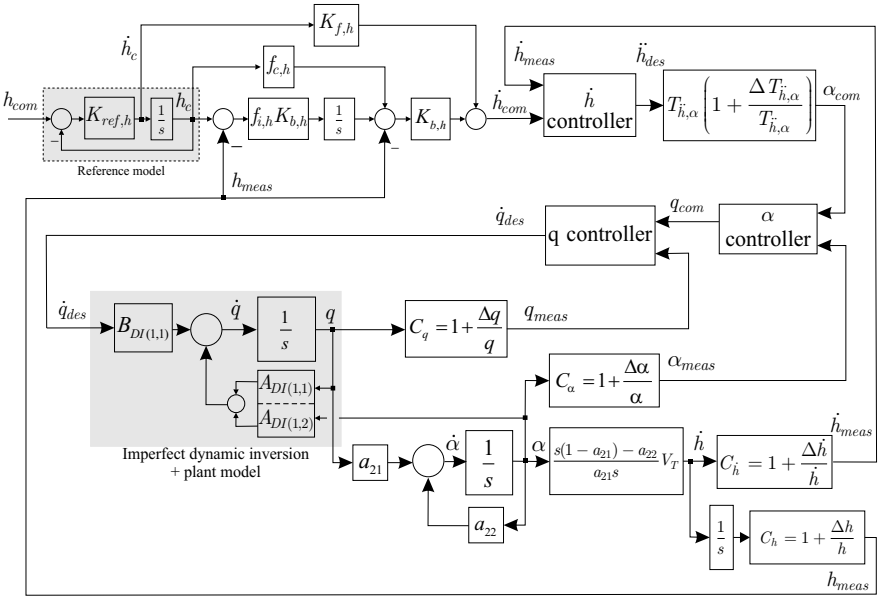


Fig. 7.31 Altitude controller

7.9.1 Open-loop Gain

The open-loop gain is computed as follows:

$$L_{e,h} = \frac{G_h}{1 + G_h K_{b,h}} \cdot \frac{K_{b,h}^2 \, f_{i,h}}{s} \,, \tag{7.79}$$

where

$$G_h = \frac{h_{meas}}{\dot{h}_{com}} = \frac{h_{meas}}{\dot{h}_{meas}} \cdot \frac{\dot{h}_{meas}}{\dot{h}_{com}} = \frac{C_h C_{\dot{h}}{}^{-1}}{s} \cdot \frac{\dot{h}_{meas}}{\dot{h}_{com}} \,. \tag{7.80}$$

The transfer function $\dot{h}_{meas}/\dot{h}_{com}$ is defined in (7.75), the coefficient C_h is defined in (7.78), and the coefficient $C_{\dot{h}}$ is defined in (7.69).

Figure 7.32 shows the open-loop gains obtained when the parameters $\frac{\Delta a_{11}}{a_{11}}$, $\frac{\Delta a_{12}}{a_{12}}$, $\frac{\Delta b_{11}}{b_{11}}$, $\frac{\Delta q}{q}$, and $\frac{\Delta\alpha}{\alpha}$ take on alternatively the values $+30\,\%$ and $-30\,\%$ in a nested for-loops algorithm. The term $\frac{\Delta T_{\ddot{h},\alpha}}{T_{\ddot{h},\alpha}}$ takes on alternatively the values $+50\,\%$ and $-50\,\%$ and $\frac{\Delta h}{h}$ the values $+1\,\%$ and $-1\,\%$.

In all cases, the system is asymptotically stable, the bandwidth of the open-loop gain is almost unaffected by the uncertainties and is in the range 0.3 to 0.4 rad/s.

7.9.2 Closed-loop Gain

The closed-loop gain is computed as follows:

$$\begin{aligned}\frac{h_{meas}}{h_{com}} &= \frac{h_{meas}}{h_c} \cdot \frac{h_c}{h_{com}} \\ &= \frac{G_h \left[K_{f,h} s^2 + K_{b,h}(f_{c,h} s + f_{i,h} K_{b,h})\right]}{s + G_h K_{b,h}(s + f_{i,h} K_{b,h})} \cdot \frac{K_{ref,h}}{s + K_{ref,h}} \,. \end{aligned} \quad (7.81)$$

The choice of $K_{b,h} = 0.2\,[\mathrm{s}^{-1}]$ yields a closed-loop bandwidth in the range of 0.2 to 0.3 rad/s as shown in Fig. 7.33. This guarantees sufficient time-scale separation with the rate-of-climb controller.

Figure 7.33 shows that the plant and measurement uncertainties have almost no influence on the bandwidth of the altitude control loop.

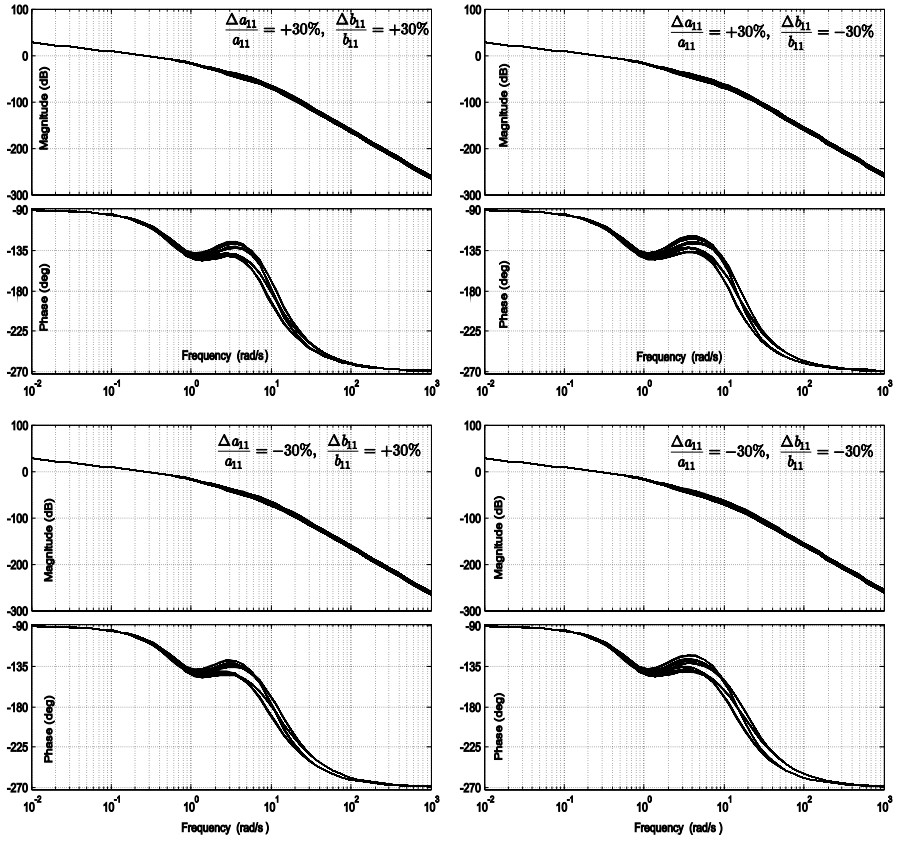


Fig. 7.32 Open-loop gains of the altitude controller for several uncertainty configurations

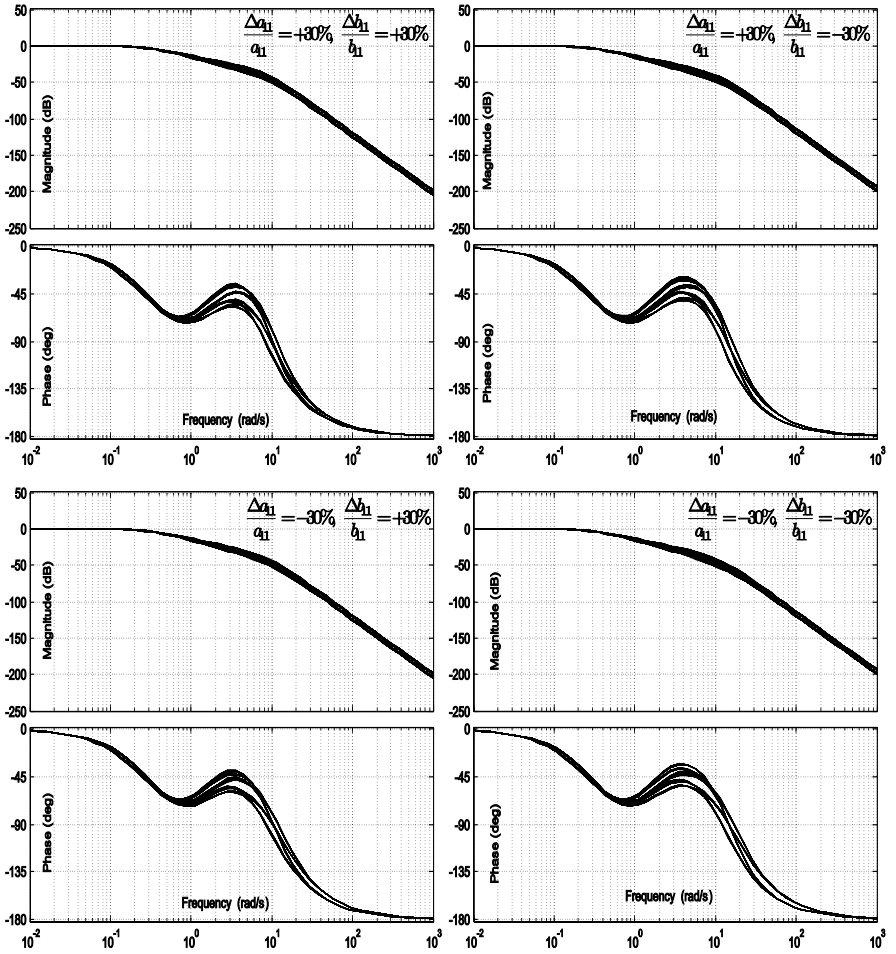


Fig. 7.33 Closed-loop gains of the altitude controller for several uncertainty configurations

7.9.3 Performance of the Altitude Controller

Figure 7.34 shows the performance of the altitude controller in the nominal case (no plant uncertainty) for a constant airspeed of 30 m/s. The aircraft is capable of tracking the reference vertical trajectory without steady-state error for a step or a ramp input. There is also no overshoot for a step input. The aircraft reaches the reference altitude quickly.

The overall performance of the longitudinal motion controllers presented in this chapter are very satisfactory.

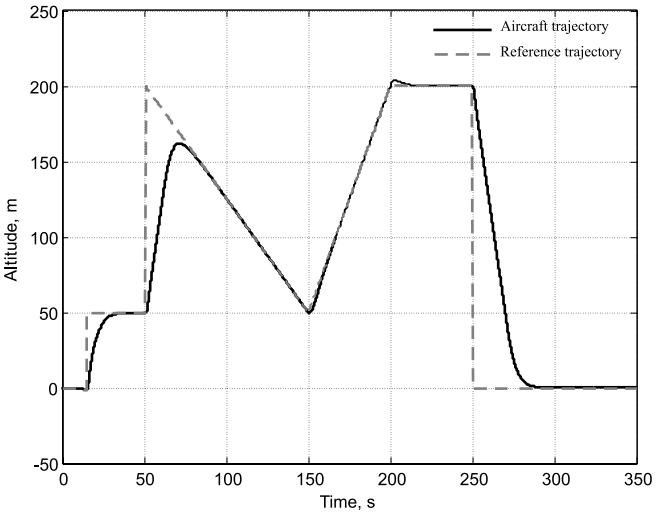


Fig. 7.34 Performance of the altitude controller (nominal conditions)

Remarks:

- In this chapter, an anti-windup system is implemented for all the control loops that employ an integrator. Moreover, axis control saturation can be detected with an analysis of the control inputs and outputs of the control allocation module. The possible occurrence of an actuator failure should also be taken into account for anti-windup compensation.
- The paper by Schierman *et al.* [22] contains a very interesting scheme for the longitudinal control of the Boeing X-40A reusable launch vehicle. The inner-control loops for attitude utilize a model-following and dynamic inversion approach. The architecture contains five control loops for the control of altitude, altitude rate, flight path angle, angle of attack and pitch rate, respectively. Each loop employs a model-following scheme, the bandwidth of which is adjusted in a backstepping manner to maintain robustness stability margins in the presence of actuator saturation or failure.

7.10 Airspeed Controller

7.10.1 Content of this Section

This section presents a new nonlinear airspeed controller for a UAV [23]. The thrust force of the UAV is generated by a propeller. The nonlinear dynamics of the airspeed are appropriately expressed as a function of the thrust force, which is a nonlinear function of the engine speed. The latter variable is the only physical control input of the system. The airspeed control system is constituted of a linear controller responsible for the generation of the desired airspeed dynamics, which are successively converted into a desired thrust command and an engine speed command through two consecutive nonlinear transformations. The proposed architecture is modular, easy to implement, and computationally efficient. Nonlinear simulation results demonstrate the effectiveness of the method.

7.10.2 Motivation

An essential variable involved in the flight dynamics of an aircraft is the *airspeed* or the speed of the airframe relative to the surrounding air. The airspeed V_T is indeed part of all of the aerodynamic equations of forces and moments acting on the airplane through the dynamic pressure $\bar{q}$ [24, 25]. Human pilots also know well that the airspeed is a critical flight parameter. It is to be carefully monitored in order to avoid any undesired stall of the airplane. In [26], it is explained that "precise airspeed control" is required in order to achieve "precise altitude and heading for longer and longer periods in straight and level flight." Precise airspeed control is also necessary during climbs or descents at specified rates, or during the safety critical maneuvers at low airspeed such as the glide and flare paths before landing.

Human pilots are usually taught that the airspeed is to be adjusted by acting on the aircraft's propulsion power, and that altitude is to be controlled by modifying the pitch attitude of the aircraft. However, there are situations in which the role of pitch and power are reversed, *i.e.*, the pitch attitude is used to control the speed and the power to control the altitude. This is the case, for example, during take off, landing, and any maneuver to reach a certain gliding slope.

In this work, the first method is employed, *i.e.*, the airspeed is adjusted by the propulsion power, and the altitude of the aircraft is controlled by the pitch attitude at any given airspeed.

Figure 7.35 shows the speed autopilot found in [20], which assumes that the transfer function from the throttle command $\delta_{th,com}$ to the aircraft's airspeed V_T is of the form $V_T(s)/\delta_{th,com}(s) = k_v/s(s + a_v)$, where the constant

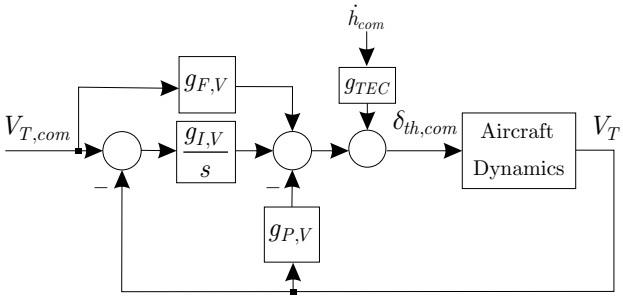


Fig. 7.35 Airspeed controller [20]

term k_v/a_v is the amount of steady-state longitudinal acceleration per unit of throttle command and $1/a_v$ is the time constant of the engine thrust response. The gains $g_{F,V}$, $g_{I,V}$, $g_{P,V}$ and g_{TEC} are used as feedforward, integral, proportional, and total energy compensation gains, respectively [20, 27].

Although the controller shown in Fig. 7.35 provides satisfactory results over a limited altitude range as found in [20], this type of modeling does not take into account that the airspeed response due to throttle commands also depends on the flight conditions, such as altitude and current flying speed. Indeed, the aerodynamic forces acting on the airframe and the thrust produced by the engine are strongly dependent on the air density ρ, which varies nonlinearly with the altitude h and with the airspeed V_T. Therefore, the dynamics of the airspeed are a nonlinear function of the airspeed, the altitude, the air density, the engine speed, and the throttle command as follows:

$$\dot{V}_T = f(V_T, h, \rho, n, \delta_{th}) \; . \tag{7.82}$$

Some papers suggest the design of robust nonlinear airspeed controllers using a combination of H_∞ control and feedback linearization [10, 11]. However, in [11] for example, the thrust is linearly modeled in terms of a throttle setting and the control design does not take into account detailed nonlinear dynamics of the thrust generated by the engine through the propeller.

In this section, a nonlinear airspeed controller is designed taking into account a nonlinear modeling of the thrust force generation. This airspeed controller is to adapt itself over the flight envelope and be of low computational load.

This section starts with a quick review of the engine and thrust models. Second, the nonlinear transformations between the desired dynamics of the airspeed and the engine speed are established. Third, a linear controller is designed to shape the dynamics of the desired airspeed signal. Finally, the section includes simulation results that demonstrate the effectiveness of the airspeed control system [23].

7.10.3 Engine Speed

The dynamics of the engine speed n are modeled by a first-order linear system with the time constant τ_n and the engine speed reference signal n_c as follows:

$$\dot{n} = -\frac{1}{\tau_n} n + \frac{1}{\tau_n} n_c \, . \tag{7.83}$$

7.10.4 Thrust Force

The thrust force F_T generated by a propeller with the diameter D, and an engine speed n is modeled as follows [10, 28]:

$$F_T = \rho n^2 D^4 C_{F_T}(J) \, , \tag{7.84}$$

where the dimensionless thrust coefficient $C_{F_T}(J)$ is defined as

$$C_{F_T}(J) = C_{F_{T1}} + C_{F_{T2}} J + C_{F_{T3}} J^2 \, , \tag{7.85}$$

with the advanced ratio J being

$$J = \frac{V_T}{D \pi n} \, , \tag{7.86}$$

and the constant coefficients $C_{F_{T1}}$, $C_{F_{T2}}$, and $C_{F_{T3}}$ as defined in Appendix F.

7.10.5 Nonlinear Transformations

The nonlinear differential equation of the aircraft's forward velocity u about the x-axis in the aircraft's body frame is found for example in Chap. 3 as follows:

$$\begin{aligned} \dot{u} &= -g \sin\theta + \frac{1}{m} \left(F_T + \boldsymbol{C}^{\boldsymbol{b}}_{\boldsymbol{w}(1,:)} [X^w \quad Y^w \quad Z^w]^T \right) \\ &= -g \sin\theta + \frac{1}{m} \left(F_T + \cos\alpha \cos\beta X^w - \cos\alpha \cos\beta Y^w - \sin\alpha Z^w \right) \, . \end{aligned} \tag{7.87}$$

The aerodynamic forces X^w, Y^w, and Z^w are defined as follows:

$$\begin{aligned} X^w &= \bar{q}S\left(C_{X1} + C_{X\alpha}\alpha + C_{X\alpha 2}\alpha^2 + C_{X\beta 2}\beta^2\right) , \\ Y^w &= \bar{q}SC_{Y\beta}\beta , \\ Z^w &= \bar{q}S\left(C_{Z1} + C_{Z\alpha}\alpha\right) . \end{aligned} \tag{7.88}$$

For values of α in the range of $[0, ..., 0.2\,\text{rad}]$ and β close to zero, and taking $\dot{u}$ as an approximation of $\dot{V}_T$, Eqs. 7.87 and 7.88 are simplified as follows:

$$\begin{aligned} \dot{V}_T &\approx -g\sin\theta + \frac{1}{m}\left(F_T + X^w - Y^w + \alpha Z^w\right) , \\ &\approx -g\sin\theta + \frac{1}{m}\left[F_T + \bar{q}S\left(C_{X1} + C_{Z1}\right)\right] + \frac{\bar{q}S}{m}\left(C_{X\alpha} + C_{Z\alpha}\right)\alpha . \end{aligned} \tag{7.89}$$

The airspeed controller has to track the reference airspeed signal $V_{T,com}$. Therefore, the first stage of the controller is designed to shape the desired dynamics of the airspeed as follows:

$$\dot{V}_{T,des} = K_V(V_{T,com}, V_T) , \tag{7.90}$$

where the symbol K_V designates a stabilizing control law for the desired airspeed dynamics as a function of the commanded and measured airspeed.

The desired dynamics for the airspeed $\dot{V}_{T,des}$ are converted into a thrust command $F_{T,com}$ as follows:

$$F_{T,com} = m[\dot{V}_{T,des} + g\sin\theta] - F^b_{aero,x} , \tag{7.91}$$

where the aerodynamic force $F^b_{aero,x}$ is expressed as

$$F^b_{aero,x} = \frac{\bar{q}S}{m}\left[\, C_{X1} + C_{Z1} + \left(C_{X\alpha} + C_{Z\alpha}\right)\alpha \,\right] . \tag{7.92}$$

The thrust command $F_{T,com}$ is to be converted into a command for the engine speed n_{com}. Equations $7.84 - 7.86$ are rearranged in order to solve for n_{com} in the following equation:

$$n^2_{com}\left(C_{F_{T1}}\rho D^4\right) + n_{com}\left(C_{F_{T2}}\rho D^3\frac{V_T}{\pi}\right) + C_{F_{T3}}\rho D^2\frac{V_T^2}{\pi^2} - F_{T,com} = 0 . \tag{7.93}$$

In order to solve the above equation at each time step, the airspeed is assumed to be constant over the sampling time yielding

$$n_{com} = \frac{\Delta_1 V_T + \sqrt{\Delta_2 V_T^2 + \Delta_3 F_{T,com}}}{\Delta_4} , \tag{7.94}$$

with

$$
\begin{aligned}
\Delta_1 &= \frac{-C_{F_{T2}}\rho D^3}{\pi}\,, \\
\Delta_2 &= \frac{\left(C_{F_{T2}}^2 - 4C_{F_{T1}}C_{F_{T3}}\right)\rho D^6}{\pi^2}\,, \\
\Delta_3 &= 4C_{F_{T1}}\rho D^4\,, \\
\Delta_4 &= 2C_{F_{T1}}\rho D^4\,.
\end{aligned} \tag{7.95}
$$

The architecture of the nonlinear airspeed controller is shown in Fig. 7.36.

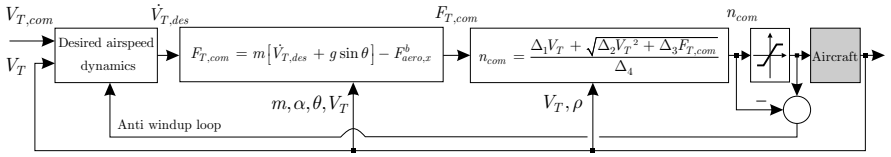


Fig. 7.36 Nonlinear airspeed controller

7.10.6 Controller of the Desired Airspeed Dynamics

The controller for the desired airspeed dynamics is the same as in Fig. 6.13 with the following gains $f_c = 1$, $f_i = 0.25$ and $K_b = 3\,[\mathrm{s}^{-1}]$.

In order to avoid conflicts between the airspeed controller and the angle-of-attack controller which cause undesired pitch oscillations, the bandwidth of the airspeed controller is chosen to be smaller than that of the angle-of-attack controller.

7.10.7 Simulation Results

Figure 7.37 shows the airspeed and the angle of attack during the aggressive vertical maneuvers of Fig. 7.34. The airspeed reference signal is 30 m/s and must remain constant despite aggressive altitude changes. Figure 7.37 shows the good performance of the airspeed controller and of the angle-of-attack controller. Moreover, they do not conflict with each other, since there is no mutual oscillatory influence from one to the other.

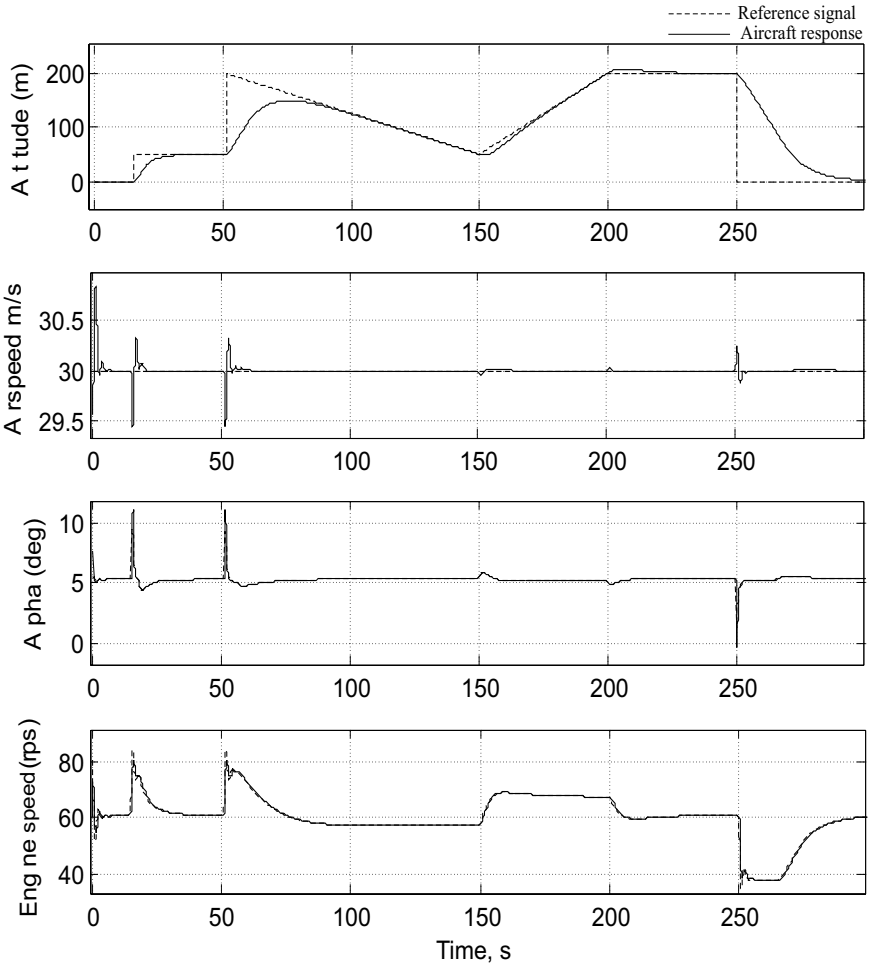


Fig. 7.37 Altitude, airspeed, angle of attack, and engine speed

References

1. D. Enns. Robustness of Dynamic Inversion vs. μ Synthesis: Lateral Directional Flight Control Example. AIAA paper 90-3338-CP, 1990.
2. D. Enns and T. Keviczky. Dynamic Inversion Based Flight Control for Autonomous RMAX Helicopter. In *Proceedings of the IEEE American Control Conference*, pages 3916–3923, Minneapolis, MN, June 2006.
3. M. G. Goman and E. N. Kolesnikov. Robust Nonlinear Dynamic Inversion Method for an Aircraft Motion Control. Paper AIAA-98-4208, pp. 511−521, 1998.
4. D. Itô, D. T. Ward, and J. Valasek. Robust Dynamic Inversion Controller Design and Analysis for the X38. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Montreal, Canada, 6−9 August 2001. Paper 2001-4380.

5. J. Kaloust, W. Schroeder, and C. Cable. Stall and Recovery Control System (SARCS): A Nonlinear Autopilot Controller for Airborne Vehicles. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, August 2000. paper 2000-4564.
6. M. B. McFarland and S. M. Hoque. Robustness of a Nonlinear Missile Autopilot Designed Using Dynamic Inversion. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Denver, CO, 14−17 August 2000. Paper AIAA-2000-3970.
7. Z. M. Smit and I. K. Craig. Robust Flight Controller Design using H_∞ Loop-Shaping and Dynamic Inversion Techniques. Paper AIAA-98-4132, pp. 305−313, 1998.
8. Q. Wang and R. F. Stengel. Robust Nonlinear Control of a Hypersonic Aircraft. pages 413–423, 1999. AIAA paper-99-4000.
9. S. Tarbouriech, G. Garcia, and D. Henrion. Local Stabilization of Linear Systems with Position and Rate Bounded Actuators. In *Proceedings of the 14th Triennal World Congress*, pages 459–464, Beijing, P.R. China, 1999.
10. M. Möckli. *Guidance and Control for Aerobatic Maneuvers of an Unmanned Airplane*. PhD thesis, ETH Zurich, 2006. Diss No. 16586.
11. M. Sadraey and R. Colgren. Robust Nonlinear Controller Design for a Complete UAV Mission. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, 2006.
12. J. Brinker and K. A. Wise. Flight Testing of a Reconfigurable Flight Control Law on the X-36 Tailless Fighter Aircraft. In *Proceedings of the AIAA Guidance, Navigation, and Control Conference*, Denver, CO, August 2000.
13. M. G. Safonov and R. Y. Chiang. Model Reduction for Robust Control: a Schur Relative Error Method. *International Journal of Adaptive Control and Signal Processing*, 2:259–272, 1988.
14. Q. Wang and R. F. Stengel. Robust Nonlinear Flight Control of a High-Performance Aircraft. *IEEE Trans. Control Systems Technology*, 13(1):15–26, January 2005.
15. A. Tsourdos, D. J. Leith, W. E. Leithead, and B. A. White. A Velocity Based Framework for the Robust Stability Analysis of Dynamic Inversion Flight Controllers. In *Proceedings of the IEEE American Control Conference*, pages 3341–3345, Minneapolis, MN, June 2001.
16. G. Papageorgiou and R. A. Hyde. Analyzing the Stability of NDI-based Flight Controllers with LPV Methods. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2001.
17. C. Papageorgiou and K. Glover. Robustness Analysis of Nonlinear Flight Controllers. *AIAA Journal of Guidance, Control, and Dynamics*, 28(4):639–648, September-October 2005.
18. G. Ducard and H. P. Geering. Stability Analysis of a Dynamic Inversion Based Pitch Rate Controller for an Unmanned Aircraft. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 360–366, Nice, France, September 2008.
19. Introduction au calcul d'incertitudes. EPFL Lecture Notes available at http://ipn.epfl.ch/webdav/site/ipn/shared/intro%20incertitudes.pdf.
20. M. Bodson. Reconfigurable Nonlinear Autopilot. *AIAA Journal of Guidance, Control, and Dynamics*, 26(5):719–727, 2003.
21. H. P. Geering. *Regelungstechnik: Mathematische Grundlagen, Entwurfsmethoden, Beispiele, 6th Edition*. Springer, Berlin, 2003.
22. J. D. Schierman, D. G Ward, J. R. Hull, N. Gandhi, M. W. Oppenheimer, and D. B. Doman. Integrated Adaptive Guidance and Control for Re-Entry Vehicles with Flight-Test Results. *AIAA Journal of Guidance, Control, and Dynamics*, 27(6):975–988, November-December 2004.

23. G. Ducard and H. P. Geering. Airspeed Control for Unmanned Aerial Vehicles: a Nonlinear Dynamic Inversion Approach. In *Proceedings of the IEEE 16th Mediterranean Conference on Control and Automation*, pages 676–681, Ajaccio, Corsica, France, June 2008.
24. R. F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.
25. B. Stevens and F. Lewis. *Aircraft Control and Simulation, Second Edition*. Wiley, New York, NY, 2003.
26. H. Sollman and S. Harris. *Mastering Instrument Flying, 3rd edition*. MacGraw-Hill Professional, 1999.
27. C. Voth and U.-L. Ly. Design of a Total Energy Compensation Autopilot Using Constrained Parameter Optimization. *AIAA Journal of Guidance, Control, and Dynamics*, 14(5):927–935, Sept-Oct 1991.
28. R. Möller. Modelling an Acrobatic Aircraft. Term paper, ETH Zürich, 2003.

Chapter 8
Autopilot for the Lateral Motion

This chapter is dedicated to the analysis and control of the lateral motion of the aircraft and presents an architecture for the lateral-directional controllers which use robust NDI in all of the control loops. This chapter brings an innovative and practical approach for stability and robustness analyses of the plant undergoing the dynamic inversion process. Finally, this chapter provides practical suggestions for the selection of the uncertain model parameters involved in the controllers.

8.1 Equations for Lateral Motion Analysis

A linear model of the longitudinal motion of the aircraft is constructed as follows:

$$\dot{\boldsymbol{\Delta x}}_{\boldsymbol{lat}} = \boldsymbol{A}_{\boldsymbol{lat}}\, \boldsymbol{\Delta x}_{\boldsymbol{lat}} + \boldsymbol{B}_{\boldsymbol{lat}}\, \boldsymbol{\Delta u}_{\boldsymbol{lat}} \,, \tag{8.1}$$

where the state vector is defined as $\boldsymbol{x}_{\boldsymbol{lat}} = [p \quad r \quad \beta \quad \phi]^T$ and the control input vector is $\boldsymbol{u}_{\boldsymbol{lat}} = [\delta_{aileron} \quad \delta_{rudder}]^T$ as shown in Fig. 8.1.

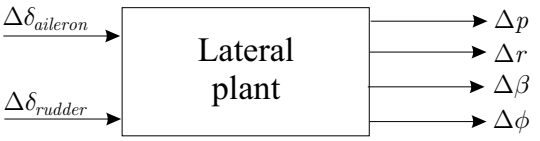


Fig. 8.1 Linear model of the aircraft lateral-directional motion

In order to build the linear longitudinal model and therefore obtain the matrices $\boldsymbol{A}_{\boldsymbol{lat}}$, $\boldsymbol{B}_{\boldsymbol{lat}}$, $\boldsymbol{C}_{\boldsymbol{lat}}$, and $\boldsymbol{D}_{\boldsymbol{lat}}$, a nonlinear differential equation must first be formulated for each of the state variables as follows:

$$\dot{p} = f_p(\boldsymbol{x}, \boldsymbol{u}) ,$$
$$\dot{r} = f_r(\boldsymbol{x}, \boldsymbol{u}) ,$$
$$\dot{\beta} = f_\beta(\boldsymbol{x}, \boldsymbol{u}) ,$$
$$\dot{\phi} = f_\phi(\boldsymbol{x}, \boldsymbol{u}) ,$$

where the state vector of the aircraft is $\boldsymbol{x}$, and the control vector is $\boldsymbol{u}$.

8.1.1 Differential Equation for the Roll Rate p

The nonlinear differential equation for the roll rate is found from (3.44) as

$$\dot{p} = I^{-1}_{(1,1)}(L^b - T^b_p) + I^{-1}_{(1,3)}(N^b - T^b_r) , \tag{8.2}$$

where

$$T^b_p = I_{zx}qp + I_{zz}qr - I_{yy}qr , \tag{8.3}$$
$$T^b_r = I_{yy}qp - I_{xx}qp - I_{xz}qr , \tag{8.4}$$

and where the roll torque L^b and the yaw torque N^b are expressed as a linear combination of the control surface deflections, the sideslip angle β, and the dimensionless roll rate $\tilde{p} = bp/2V_T$ or the dimensionless yaw rate $\tilde{r} = br/2V_T$, respectively, as follows:

$$L^b = \bar{q}SbC_L(\delta_{a1}, \delta_{a2}, \delta_{e1}, \delta_{e2}, \beta, \tilde{p}, \tilde{r}) , \tag{8.5}$$
$$N^b = \bar{q}SbC_N(\delta_r, \beta, \tilde{r}) . \tag{8.6}$$

Chapter 3 provides definition of the other variables.

8.1.2 Differential Equation for the Yaw Rate r

The nonlinear differential equation for the yaw rate is found from (3.44) as

$$\dot{r} = I^{-1}_{(3,1)}(L^b - T^b_p) + I^{-1}_{(3,3)}(N^b - T^b_r) . \tag{8.7}$$

8.1.3 Differential Equation for the Sideslip Angle β

The nonlinear differential equation of the aircraft's sideslip angle β is derived in Appendix A and is formulated as follows:

$$\dot{\beta} = -r^w + \frac{Y^w - F_T \cos\alpha \sin\beta + m g_y^w}{m V_T} \,, \tag{8.8}$$

where

$$\begin{aligned} r^w &= C^w_{b\ (3,:)} \begin{bmatrix} p & q & r \end{bmatrix}^T = -p \sin\alpha + r \cos\alpha \,, \\ Y^w &= \bar{q} S C_{Y1} \beta \,, \\ g_y^w &= g(\cos\alpha \sin\beta \sin\theta + \cos\beta \cos\theta \sin\phi - \sin\alpha \sin\beta \cos\theta \cos\phi) \,. \end{aligned} \tag{8.9}$$

8.1.4 Differential Equation for the Roll Angle ϕ

The differential equation for the roll angle is obtained as follows:

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \,. \tag{8.10}$$

8.1.5 Matrices for the Lateral Mode

The linear model for the lateral motion is obtained around the following nominal conditions $p_{nom} = q_{nom} = r_{nom} = 0, \quad \theta_{nom} = \alpha_{nom}, \quad \beta_{nom} = 0, \quad \phi_{nom} = 0$:

$$\begin{bmatrix} \dot{\Delta p} \\ \dot{\Delta r} \\ \dot{\Delta \beta} \\ \dot{\Delta \phi} \end{bmatrix} = \boldsymbol{A_{Lat}} \begin{bmatrix} \Delta p \\ \Delta r \\ \Delta \beta \\ \Delta \phi \end{bmatrix} + \boldsymbol{B_{Lat}} \begin{bmatrix} \Delta \delta_{aileron} \\ \Delta \delta_{rudder} \end{bmatrix} \,, \tag{8.11}$$

where the dynamic matrix is

$$\begin{aligned} \boldsymbol{A_{Lat}} &= \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{x}} \right)_{\boldsymbol{x}_{nom}} \\ &= \begin{bmatrix} \frac{\bar{q} S b^2 I_{zz} C_{L\tilde{p}}}{2 V_T D_1} & \frac{\bar{q} S b^2 (I_{zz} C_{L\tilde{r}} - I_{xz} C_{N\tilde{r}})}{2 V_T D_1} & \frac{\bar{q} S b (I_{zz} C_{L\beta} - I_{xz} C_{N\beta})}{D_1} & 0 \\ \frac{-\bar{q} S b^2 I_{xz} C_{L\tilde{p}}}{2 V_T D_1} & \frac{\bar{q} S b^2 (I_{xx} C_{N\tilde{r}} - I_{xz} C_{L\tilde{r}})}{2 V_T D_1} & \frac{\bar{q} S b (I_{xx} C_{N\beta} - I_{xz} C_{L\beta})}{D_1} & 0 \\ \sin\alpha & -\cos\alpha & \frac{\bar{q} S C_{Y1} - F_T \cos\alpha}{m V_T} & \frac{g \cos\theta}{V_T} \\ 1 & \tan\theta & 0 & 0 \end{bmatrix}_{\boldsymbol{x}_{nom}} \,, \end{aligned} \tag{8.12}$$

and the control input matrix is

$$\boldsymbol{B_{Lat}} = \left(\frac{\partial \boldsymbol{f}}{\partial \boldsymbol{u}}\right)_{\boldsymbol{x}_{nom}} = \begin{bmatrix} \frac{I_{zz}\,\rho\,V_T^2\,S\,b\,C_{La}}{2D_1} & -\frac{I_{xz}\,\rho\,V_T^2\,S\,b\,C_{N_{\delta r}}}{2D_1} \\ -\frac{I_{xz}\,\rho\,V_T^2\,S\,b\,C_{La}}{2D_1} & \frac{I_{xx}\,\rho\,V_T^2\,S\,b\,C_{N_{\delta r}}}{2D_1} \\ 0 & 0 \\ 0 & 0 \end{bmatrix}_{\boldsymbol{x}_{nom}} , \tag{8.13}$$

with

$$D_1 = I_{zz}I_{xx} - I_{xz}^2 \, . \tag{8.14}$$

The numerical values of $\boldsymbol{A_{Lat}}$ and $\boldsymbol{B_{Lat}}$ are listed in Appendix E.

8.2 Dynamic Modes of the Lateral Plant

The eigenvalues of the dynamic matrix $\boldsymbol{A_{Lat}}$ are shown in Fig. 8.2.

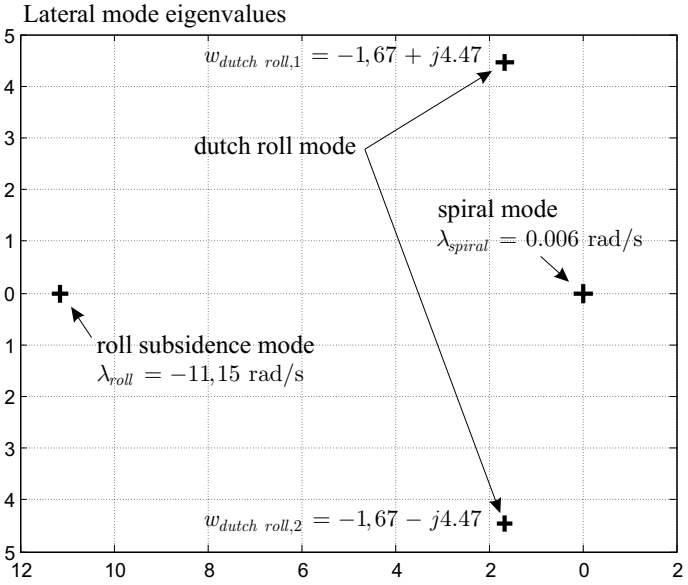


Fig. 8.2 Aircraft lateral motion eigenvalues

8.2.1 Dutch Roll Mode

Figure 8.2 shows the existence of an oscillatory mode defined by the pair of complex eigenvalues $\omega_{Dutch,roll} = -1.67 \pm j4.47$. This mode is called the *Dutch roll mode* and has a frequency around $\omega = 4.95\,\text{rad/s}$ and a damping factor of $\zeta = 0.353$. In this oscillatory mode, all of the state variables of the lateral motion are involved, and the behavior of the aircraft at this mode is a combination of roll and yawing motion with some sideslipping [1, 2]. This mode can be excited by pulses on the rudder, which cause the aircraft to roll and yaw at the same time. This mode is to be carefully controlled especially during the landing phase in windy conditions.

8.2.2 Roll Subsidence Mode

This mode describes the response of the roll angle to some lateral control inputs. It is a stable exponential mode which gives an idea of how fast the aircraft will start to roll [1, 2]. This mode gets faster as the speed increases. Figure 8.2 shows that the frequency of this mode is $\omega_{roll} = 11.37$ rad/s, and therefore the aircraft has a quick roll response, which is to be expected since the model corresponds to an aerobatic aircraft.

8.2.3 Spiral Mode

The *spiral mode* can be stable or unstable. In the case shown in Fig. 8.2, the mode is unstable with a very low frequency $\omega_{spiral} = 0.034$ rad/s. For instance, if the aircraft is given a small initial roll angle on the right, a sideslip to the right is produced, which produces a yawing moment to the right. The aircraft keeps turning while the roll angle keeps increasing. This causes the aircraft to fly a steeper and steeper spiral.

8.3 Validation of the Linear Lateral Model

In order to validate the analytical terms found for $\boldsymbol{A_{lat}}$ in (8.12) and $\boldsymbol{B_{lat}}$ in (8.13), the linear lateral model is simulated for some input perturbations and compared with a nonlinear simulation for the input perturbations.

8.3.1 Perturbation on the Aileron Command

The top plot in Fig. 8.3 shows a perturbation on the aileron of about 10 % of the maximum aileron command. The other plots in Fig. 8.3 show that the nonlinear and linear simulations are in good agreement for the roll rate response p. For the other state variables, namely r, β, and ϕ, the linear and the nonlinear simulations start to slowly separate after 4 s. However, this is sufficient to validate the linear model response to an aileron perturbation.

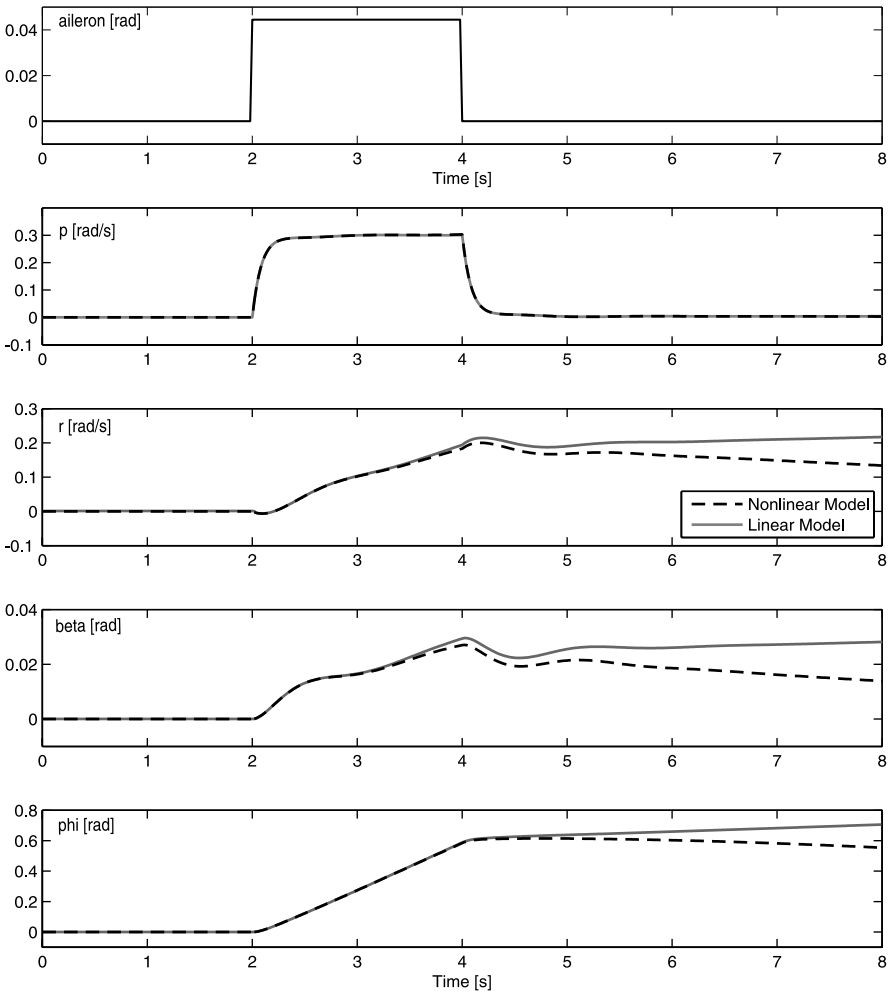


Fig. 8.3 Validation of the lateral plant with a perturbation on the aileron

8.3.2 Perturbation on the Rudder Command

The top plot in Fig. 8.4 shows a perturbation of about 40 % of the maximum rudder deflection. This is a significantly strong perturbation, and yet the deviations of the state variables between the linear and nonlinear simulations are rather small. This validates the behavior of the linear lateral model after a rudder perturbation.

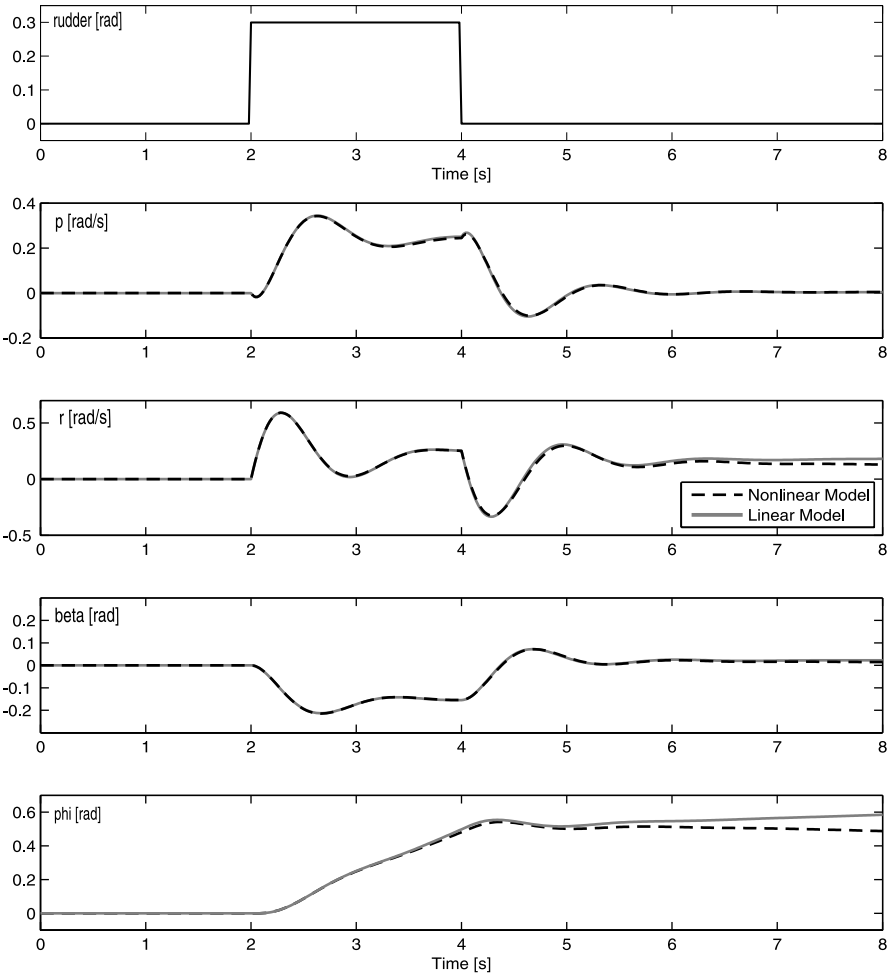


Fig. 8.4 Validation of the lateral plant with a perturbation on the rudder

8.3.3 Linearization at Different Operating Points

Figure 8.5 shows how the eigenvalues of the lateral motion evolve for different airspeeds. All the modes become faster as the airspeed increases, except for the spiral mode. Consequently, the corner frequency of the lateral-directional controller has to become larger as the airspeed increases and be at least larger than the frequency of the Dutch roll mode.

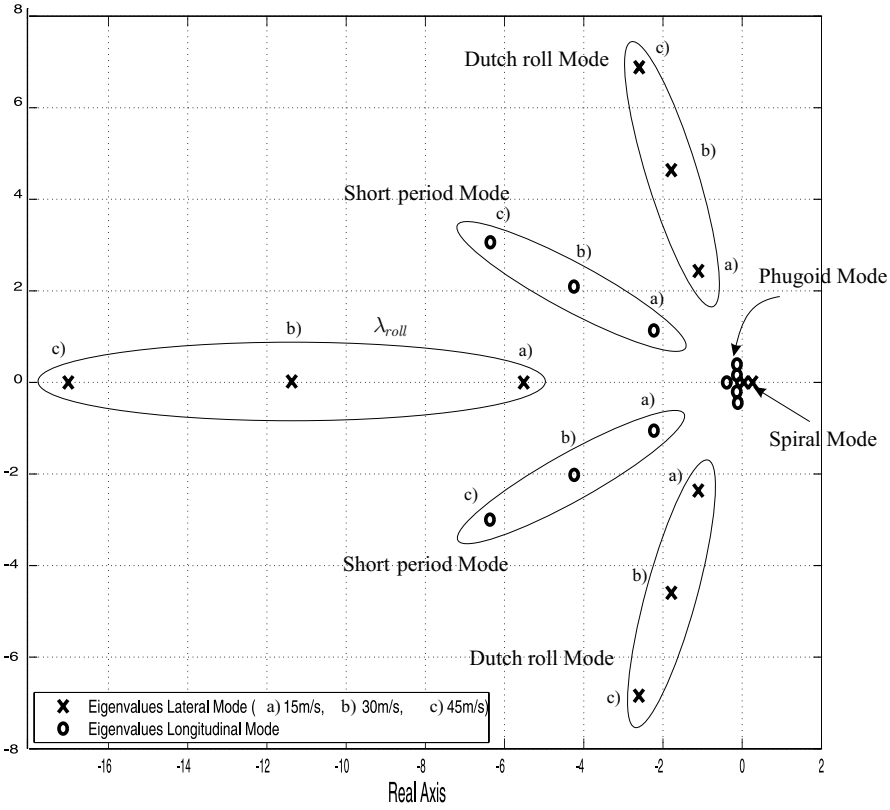


Fig. 8.5 Eigenvalues of the plant for different aircraft speeds

8.4 Stability Analysis of the Uncertain Dynamic Inversion

In order to control the lateral-directional motion of the aircraft, the controllers must deal with the roll and yaw axes simultaneously. Because the plant dynamics are not perfectly known and the measurements are noisy, perfect cancelation of the nonlinearities in the plant dynamics cannot be achieved by NDI.

The contributions of this section to the analyses of stability and robustness of the plant together with dynamic inversion for the lateral motion are:

- To show that transfer functions can be derived from the plant inputs to the plant outputs, which include the dynamic inversion process and all the relevant model parameter uncertainties and measurement uncertainties.
- To validate the control design and assess the stability and robustness properties using the notion of phase and gain margins from classical control theory.
- To provide a procedure for the selection of uncertain model parameters involved in the dynamic inversion paths.

8.4.1 Uncertain Model Parameters and Measurement Data

The uncertain model parameters involved in the lateral motion are $C_{L_{a1}}$, $C_{L_{a2}}$, $C_{L_{e1}}$, $C_{L_{e2}}$, $C_{L\beta}$, $C_{L\tilde{p}}$, $C_{L\tilde{r}}$, I_{xx}, I_{xz}, I_{zx}, I_{zz}, $C_{N\delta r}$, $C_{N\beta}$, $C_{N\tilde{r}}$ and the measurement of p, r, β, V_T.[1] The uncertainty levels of these parameters are summarized in Table 8.1.

Furthermore, it is assumed that there is no uncertainty on the aircraft mass m, the wing surface S, the wingspan b, and the air density ρ.

8.4.2 Modeling of the Uncertain Dynamic Inversion

Figure 8.6 shows the NDI process involved in the innermost control loops of the lateral-directional motion, namely the roll and yaw rate control loops.

If dynamic inversion operates perfectly, the channels $\dot{p}_{des}$ to p and $\dot{r}_{des}$ to r are totally independent. Perfect dynamic inversion cancels the cross-coupling between the roll and yaw axes, and each channel behaves as a pure integrator.

However, as soon as some uncertainty is introduced in the inversion process, these pure integrators transform into either stable or unstable first-

[1] Please refer to Chap. 3 or the Nomenclature for the definition of these parameters.

Table 8.1 Summary of the parameter and measurement uncertainties for the lateral-directional controllers

Parameter	Nominal value	Level of uncertainty	Unit
$C_{L_{a1}}$	-3.395×10^{-2}	$\pm 10\,\%$	-
$C_{L_{a2}}$	3.395×10^{-2}	$\pm 10\,\%$	-
$C_{L_{e1}}$	-0.485×10^{-2}	$\pm 10\,\%$	-
$C_{L_{e2}}$	0.485×10^{-2}	$\pm 10\,\%$	-
$C_{L\beta}$	-1.30×10^{-2}	$\pm 30\,\%$	-
$C_{N\delta r}$	5.34×10^{-2}	$\pm 10\,\%$	-
$C_{N\beta}$	8.67×10^{-2}	$\pm 10\,\%$	-
$C_{N\tilde{r}}$	-2.14×10^{-1}	$\pm 10\,\%$	-
I_{xx}	2.56	$\pm 5\,\%$	kg m^2
$I_{xz} = I_{zx}$	0.5	$\pm 5\,\%$	kg m^2
I_{zz}	11.3	$\pm 5\,\%$	kg m^2
p	0	$\pm 5\,\%$	rad/s
r	0	$\pm 5\,\%$	rad/s
β	0	$\pm 5\,\%$	rad
V_T	30	$\pm 5\,\%$	m/s

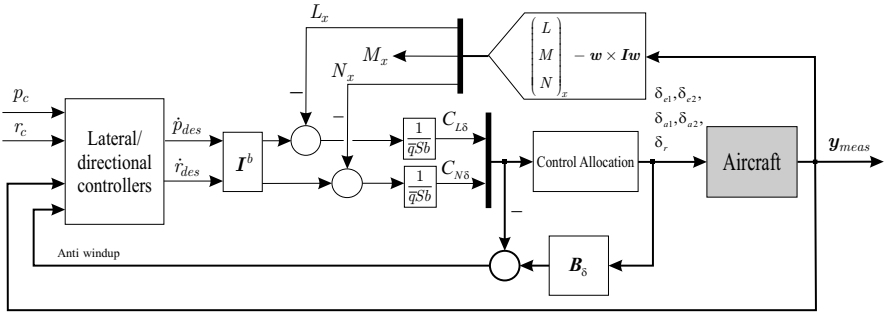


Fig. 8.6 NDI for lateral-directional motion control

order systems depending on the uncertainty. The rest of the chapter shows that when the dynamic inversion path is not perfect anymore due to model uncertainties and measurement errors, cross-couplings appear again between the roll and yaw axes, and the transfer function of each channel is no longer a pure integrator. Under certain uncertainty configuration, the dynamic inversion process destabilizes the plant, which is manifest by the appearance of unstable poles in the transfer functions. The following section is dedicated to the analysis of the influence of the parameter and measurement uncertainties on the stability of the plant after dynamic inversion.

8.4.3 Linear Representation of the Lateral-directional Motion

The plant and the dynamic inversion processes shown in Fig. 8.6 are linearized and written in a state space representation as follows:

$$\begin{aligned}\dot{\boldsymbol{\Delta x}} &= \left[\boldsymbol{A} - \boldsymbol{B}(\boldsymbol{CB^u})^{-1}\boldsymbol{A^u C^u}\right]\boldsymbol{\Delta x} + \boldsymbol{B}(\boldsymbol{CB^u})^{-1}\boldsymbol{\Delta u}\ , \\ &= \boldsymbol{A_{DI}}\,\boldsymbol{\Delta x} + \boldsymbol{B_{DI}}\,\boldsymbol{\Delta u}\ ,\end{aligned} \tag{8.15}$$

where the state vector is $\boldsymbol{x} = [p \quad r \quad \beta]^T$, and the control input vector is $\boldsymbol{u} = [\dot{p}_{des} \quad \dot{r}_{des}]^T$ as shown in Fig. 8.7. The superscript u is used to indicate an uncertain matrix.

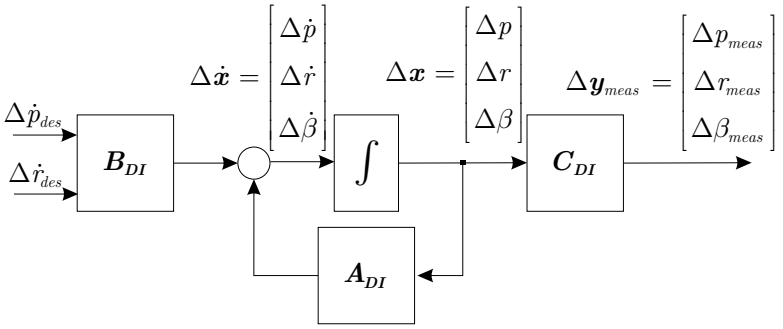


Fig. 8.7 Linear representation of the lateral-directional plant and the uncertain dynamic inversion process

8.4.4 Definition of the Matrices A_{DI}, B_{DI}, and C_{DI} for the Lateral Mode

Since the only relevant state variables of the lateral-directional motion that are involved in the dynamic inversion process are p, r, and β, the aircraft lateral-directional linear model is simplified as follows:

$$\begin{bmatrix}\dot{\Delta}p\\ \dot{\Delta}r\\ \dot{\Delta}\beta\end{bmatrix} = \begin{bmatrix}a_{11} & a_{12} & a_{13}\\ a_{21} & a_{22} & a_{23}\\ a_{31} & a_{32} & a_{33}\end{bmatrix}\begin{bmatrix}\Delta p\\ \Delta r\\ \Delta\beta\end{bmatrix} + \begin{bmatrix}b_{11} & b_{12}\\ b_{21} & b_{22}\\ b_{31} & b_{32}\end{bmatrix}\begin{bmatrix}\Delta\delta_a\\ \Delta\delta_r\end{bmatrix}\ ,$$

$$\begin{bmatrix}\dot{\Delta}p\\ \dot{\Delta}r\\ \dot{\Delta}\beta\end{bmatrix} = \boldsymbol{A_{p,r,\beta}}\begin{bmatrix}\Delta p\\ \Delta r\\ \Delta\beta\end{bmatrix} + \boldsymbol{B_{p,r,\beta}}\begin{bmatrix}\Delta\delta_a\\ \Delta\delta_r\end{bmatrix}\ . \tag{8.16}$$

The matrices $\boldsymbol{A_{DI}}$, $\boldsymbol{B_{DI}}$, and $\boldsymbol{C_{DI}}$ found in (8.15) are obtained as follows:

$$\boldsymbol{A_{DI}} = \boldsymbol{A_{p,r,\beta}} - \begin{bmatrix} \boldsymbol{B_{p,r,\beta}}_{(1:2,1:2)} \left(\boldsymbol{B^{u}_{p,r,\beta}}\right)^{-1} \boldsymbol{A^{u}_{p,r,\beta}}_{(1:2,:)} \boldsymbol{C^{u}} \\ \boldsymbol{0}_{1\times 3} \end{bmatrix} \in \Re^{3\times 3} . \tag{8.17}$$

The reduced and uncertain dynamic matrix $\boldsymbol{A^{u}_{p,r,\beta}}_{(1:2,:)}$ used in (8.17) is defined as

$$\begin{aligned} \boldsymbol{A^{u}_{p,r,\beta}}_{(1:2,:)} &= \begin{bmatrix} a_{11}^{'} & a_{12}^{'} & a_{13}^{'} \\ a_{21}^{'} & a_{22}^{'} & a_{23}^{'} \end{bmatrix} \in \Re^{2\times 3}, \\ &= \begin{bmatrix} a_{11} + \Delta a_{11} & a_{12} + \Delta a_{12} & a_{13} + \Delta a_{13} \\ a_{21} + \Delta a_{21} & a_{22} + \Delta a_{22} & a_{23} + \Delta a_{23} \end{bmatrix} , \end{aligned} \tag{8.18}$$

with the terms a_{11}, a_{12}, ..., a_{23} defined in (8.16). The reduced control input matrix $\boldsymbol{B_{p,r,\beta}}_{(1:2,1:2)}$ is defined as follows:

$$\boldsymbol{B_{p,r,\beta}}_{(1:2,1:2)} = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \in \Re^{2\times 2} , \tag{8.19}$$

with the terms b_{11}, b_{12}, ..., b_{22} defined in (8.16). Uncertain terms are added to the reduced control input matrix $\boldsymbol{B^{u}_{p,r,\beta}}$ as follows:

$$\begin{aligned} \left(\boldsymbol{B^{u}_{p,r,\beta}}\right)^{-1} &= \begin{bmatrix} b_{11}^{'} & b_{12}^{'} \\ b_{21}^{'} & b_{22}^{'} \end{bmatrix}^{-1} \in \Re^{2\times 2} \\ &= \begin{bmatrix} b_{11} + \Delta b_{11} & b_{12} + \Delta b_{12} \\ b_{21} + \Delta b_{21} & b_{22} + \Delta b_{22} \end{bmatrix}^{-1} . \end{aligned} \tag{8.20}$$

The control input matrix of the system of (8.15) is expressed as follows:

$$\boldsymbol{B_{DI}} = \begin{bmatrix} \boldsymbol{B_{p,r,\beta}}_{(1:2,1:2)} \left(\boldsymbol{B^{u}_{p,r,\beta}}\right)^{-1} \\ \boldsymbol{0}_{1\times 2} \end{bmatrix} \in \Re^{3\times 2} . \tag{8.21}$$

The uncertain output or measurement matrix of the system in Fig. 8.7 is

$$\boldsymbol{C_{DI}} = \boldsymbol{C^{u}} = \begin{bmatrix} C_{p}^{'} & 0 & 0 \\ 0 & C_{r}^{'} & 0 \\ 0 & 0 & C_{\beta}^{'} \end{bmatrix} \in \Re^{3\times 3} , \tag{8.22}$$

with $C_{p}^{'} = 1 + \frac{\Delta p}{p}$, $C_{r}^{'} = 1 + \frac{\Delta r}{r}$, and $C_{\beta}^{'} = 1 + \frac{\Delta \beta}{\beta}$. A graphical representation of the dynamic inversion process in the linear case and applied to the lateral-directional axes is shown in Fig. 8.8.

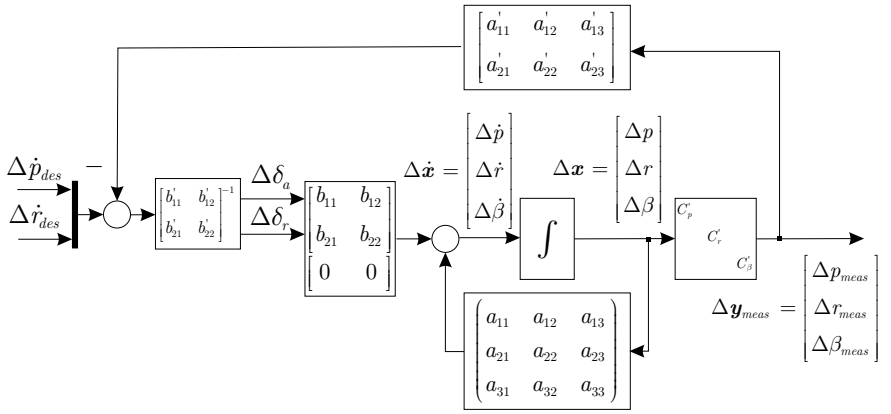


Fig. 8.8 Linear representation of dynamic inversion applied to the lateral-directional axes

The explicit transfer functions can be obtained using the following formula

$$\boldsymbol{G}(s) = \boldsymbol{C_{DI}}(s\boldsymbol{I} - \boldsymbol{A_{DI}})^{-1}\boldsymbol{B_{DI}} = \frac{\boldsymbol{C_{DI}}\ \mathrm{adj}(s\boldsymbol{I} - \boldsymbol{A_{DI}})\boldsymbol{B_{DI}}}{|s\boldsymbol{I} - \boldsymbol{A_{DI}}|}\ . \qquad (8.23)$$

It is known that for multi-input and multi-output (MIMO) systems, checking stability margins "one-loop-at-a-time" is inappropriate and that the robust stability can be assessed by analyzing the shape of the singular value (SV) $\bar{\sigma}$ [3]. However, the stability of each separate channel $\dot{p}_{des} \rightarrow p$ and $\dot{r}_{des} \rightarrow r$ will be still qualitatively discussed in the next two sections. They give practical insight on how the uncertain dynamic inversion affects stability of each channel and give suggestions on how to choose the values of the model coefficients involved in the dynamic inversion paths. Robust stability and robust performance of the lateral-directional control system is discussed later in this chapter.

8.4.5 Stability of the Channel $\dot{p}_{des}$ to p_{meas}

In this section, the stability of the plant channel from $\dot{p}_{des}$ to p_{meas} is analyzed when the most relevant model parameters are uncertain.

If the dynamic inversion and the measurement of p are perfect, the transfer function $p_{meas}/\dot{p}_{des}$ is one of a pure integrator. However, when the model parameters used in the inversion process contain uncertainties, the pure integrator transforms into a system whose transfer function contains poles and zeros. The explicit transfer function $p_{meas}/\dot{p}_{des}$ containing the uncertain parameters is obtained with (8.23).

Simulation results show that this transfer function is most sensitive to the variations of the model parameters $a_{11} = \partial\dot{p}/\partial p$ and $b_{11} = \partial\dot{p}/\partial\delta_a$ as defined in (8.16).

Figure 8.10 shows that some unstable poles appear as soon as $\Delta b_{11}/b_{11} < 0$. Uncertainties on the parameter $a_{12} = \partial\dot{p}/\partial r$ do not affect the stability of the channel (see numerical values of the lateral-directional matrix in Appendix E). The third most influential parameter is $a_{13} = \partial\dot{p}/\partial\beta$. The effects of the uncertainties on a_{13} are shown in Fig. 8.9.

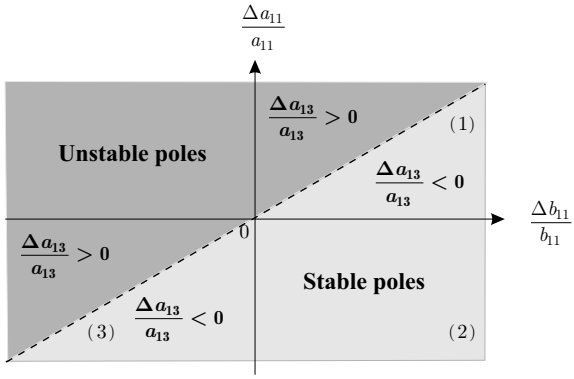


Fig. 8.9 Chart of stable and unstable poles appearing in the transfer function $p/\dot{p}_{des}$ due to uncertainties on the parameters a_{11}, b_{11} and a_{13}

The light gray area in Fig. 8.9 indicates the combination in the parameter uncertainties which results in stable poles only. Therefore, during the design of the roll rate controller, we suggest taking as plant parameters running on the onboard computer the following coefficients:

$$
\begin{aligned}
a_{11} &= \frac{\partial\dot{p}}{\partial p} = a_{11,\text{nom}}\left(1 - \left|\frac{\Delta a_{11}}{a_{11}}\right|\right) , \\
a_{13} &= \frac{\partial\dot{p}}{\partial\beta} = a_{13,\text{nom}}\left(1 - \left|\frac{\Delta a_{13}}{a_{13}}\right|\right) , \\
b_{11} &= \frac{\partial\dot{p}}{\partial\delta_a} = b_{11,\text{nom}}\left(1 + \left|\frac{\Delta b_{11}}{b_{11}}\right|\right) .
\end{aligned} \tag{8.24}
$$

Another parameter whose uncertainty has a significant influence on the stability of the channel from $\dot{p}_{des}$ to p is the measurement data of the roll rate. Simulation results show that any uncertainty in the roll rate measurement such that $\Delta p/p > 0$ causes the appearance of large unstable poles in the regions (1) and (3) in Fig. 8.9, and a small right half-plane pole also appears in region (2).

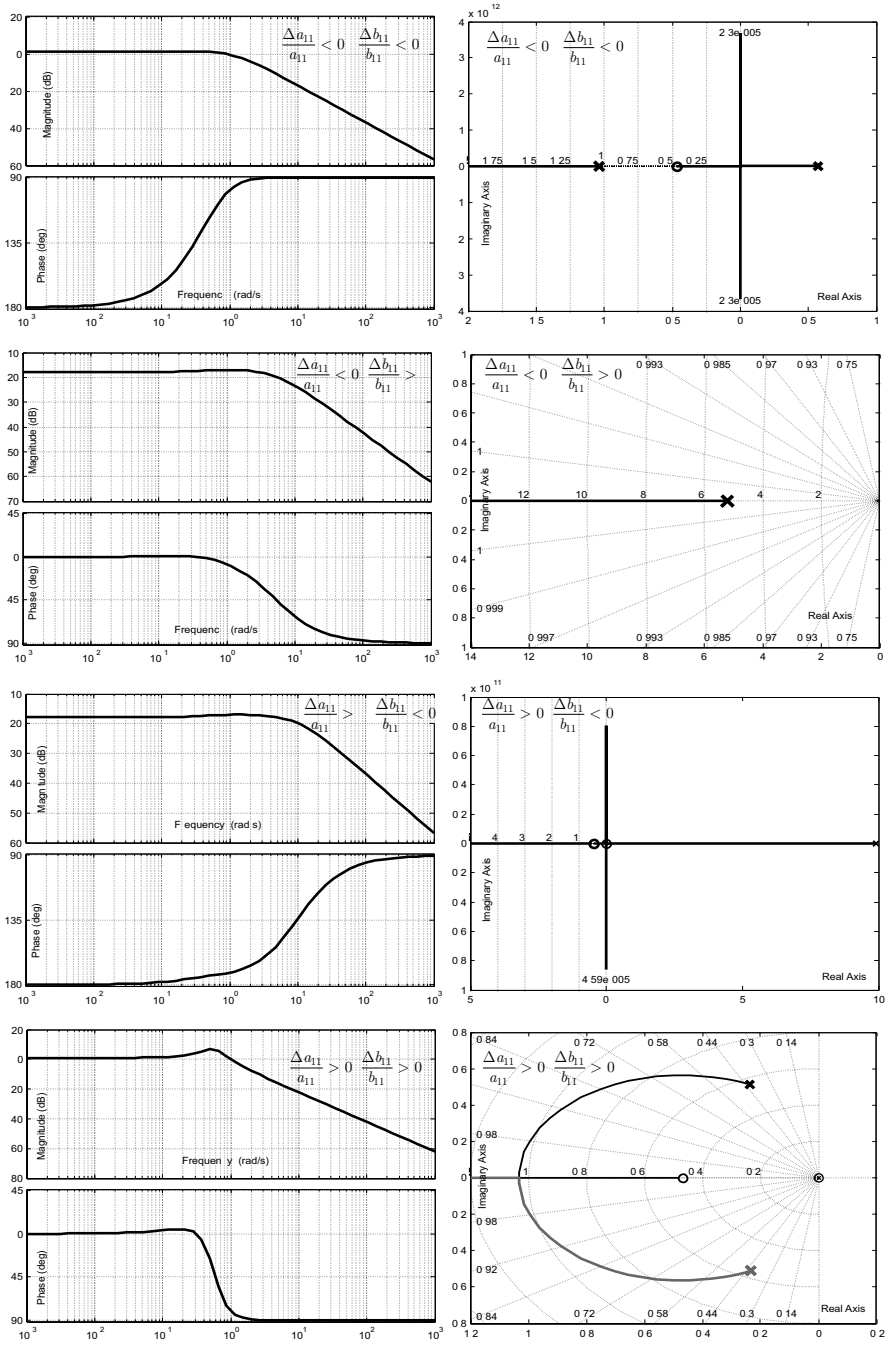


Fig. 8.10 Influence of the uncertainties $\Delta a_{11}/a_{11}$ and $\Delta b_{11}/b_{11}$ on the stability of the channel $\dot{p}_{des}$ to p_{meas}. Bode plots on the *left*, root locus plots on the *right*

Therefore, the roll rate measurement data used for the dynamic inversion process should be selected as follows:

$$p = p_{meas}\left(1 - \left|\frac{\Delta p}{p}\right|\right). \tag{8.25}$$

Other parameters such as $\Delta r/r$, $\Delta\beta/\beta$, b_{12} and b_{22} have almost no influence on the stability of the channel $\dot{p}_{des} \rightarrow p_{meas}$. Additional simulations show that it is best for stability to select the parameters as follows:

$$\begin{aligned}
a_{21} &= \frac{\partial \dot{r}}{\partial p} = a_{21,\mathrm{nom}}\left(1 - \left|\frac{\Delta a_{21}}{a_{21}}\right|\right), \\
a_{22} &= \frac{\partial \dot{r}}{\partial r} = a_{22,\mathrm{nom}}\left(1 - \left|\frac{\Delta a_{22}}{a_{22}}\right|\right), \\
a_{23} &= \frac{\partial \dot{r}}{\partial \beta} = a_{23,\mathrm{nom}}\left(1 - \left|\frac{\Delta a_{23}}{a_{23}}\right|\right).
\end{aligned} \tag{8.26}$$

This choice for the selection of the parameters used in the dynamic inversion path will be confirmed in the next section when the stability of the channel $\dot{r}_{des}$ to r_{meas} is discussed.

8.4.6 Stability of the Channel $\dot{r}_{des}$ to r_{meas}

For the analysis of the stability of the channel $\dot{r}_{des}$ to r_{meas}, all of the terms discussed in the previous section are selected according to (8.24) and (8.25). For all other coefficients involved in dynamic inversion, extensive simulations show that it is best to choose

$$\begin{aligned}
a_{21} &= \frac{\partial \dot{r}}{\partial p} = a_{21,\mathrm{nom}}\left(1 - \left|\frac{\Delta a_{21}}{a_{21}}\right|\right), & p &= p_{meas}\left(1 - \left|\frac{\Delta p}{p}\right|\right), \\
a_{22} &= \frac{\partial \dot{r}}{\partial r} = a_{22,\mathrm{nom}}\left(1 - \left|\frac{\Delta a_{22}}{a_{22}}\right|\right), & r &= r_{meas}\left(1 - \left|\frac{\Delta r}{r}\right|\right), \\
a_{23} &= \frac{\partial \dot{r}}{\partial \beta} = a_{23,\mathrm{nom}}\left(1 - \left|\frac{\Delta a_{23}}{a_{23}}\right|\right), & \beta &= \beta_{meas}\left(1 - \left|\frac{\Delta \beta}{\beta}\right|\right), \\
b_{21} &= \frac{\partial \dot{r}}{\partial \delta_a} = b_{21,\mathrm{nom}}\left(1 - \left|\frac{\Delta b_{21}}{b_{21}}\right|\right), \\
b_{22} &= \frac{\partial \dot{r}}{\partial \delta_r} = b_{22,\mathrm{nom}}\left(1 + \left|\frac{\Delta b_{22}}{b_{22}}\right|\right).
\end{aligned} \tag{8.27}$$

Simulation results show that the stability of the channel $\dot{r}_{des}$ to r_{meas} is influenced mostly by the terms b_{21}, $\Delta r/r$, and $\Delta\beta/\beta$. A heuristic rule that seems to work remarkably well is to select uncertainty levels in order to:

- Estimate the measurement data intentionally too low
- Estimate negative model parameters intentionally too low
- Estimate positive model parameters intentionally too high.

Based on this statement and on the parameter selections of (8.24) to (8.27), the channel $\dot{r}_{des}$ to r_{meas} only contains stable poles and zeros as shown in Fig. 8.11. Almost any other configuration of uncertain parameters yields unstable poles.

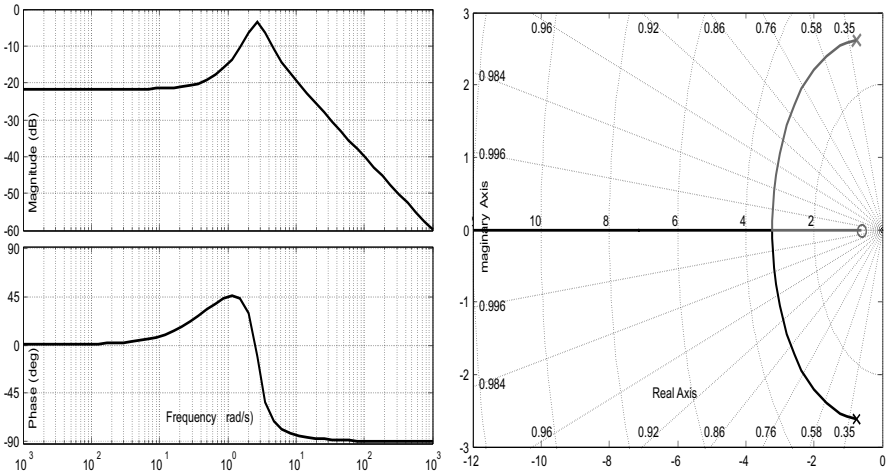


Fig. 8.11 Bode plot and root locus of the channel $\dot{r}_{des}$ to r_{meas} with uncertainty levels indicated in (8.24) to (8.27)

8.5 Roll and Yaw Rate Controllers

Figure 8.13 shows the SVs of the plant $\boldsymbol{G}$ represented in Fig. 8.12. Clearly, the loop gain has no integrator behavior, and the SVs of $\boldsymbol{G}$ are small at 0 Hz. Therefore, in order to avoid a large steady-state error in the closed loop, the roll and yaw rate controllers shown in Fig. 8.14 are designed with an integrator behavior.

The SVs $\underline{\sigma}(\boldsymbol{G})$ and $\bar{\sigma}(\boldsymbol{G})$ are also largely separated, they are said to be unbalanced. Therefore, the controller $\boldsymbol{K}$ is designed to balance the SVs as much as possible, *i.e.*, in order that the SVs $\underline{\sigma}(\boldsymbol{GK})$ and $\bar{\sigma}(\boldsymbol{GK})$ are close to

the same values. This ensures that the speed of the responses will be nearly the same in all channels of the system.

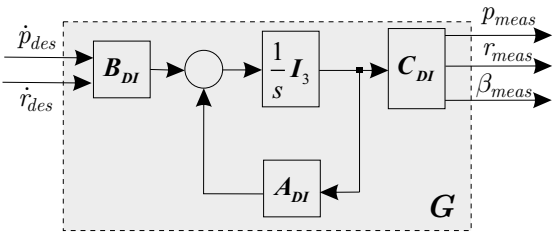


Fig. 8.12 The lateral/directional plant after uncertain dynamic inversion

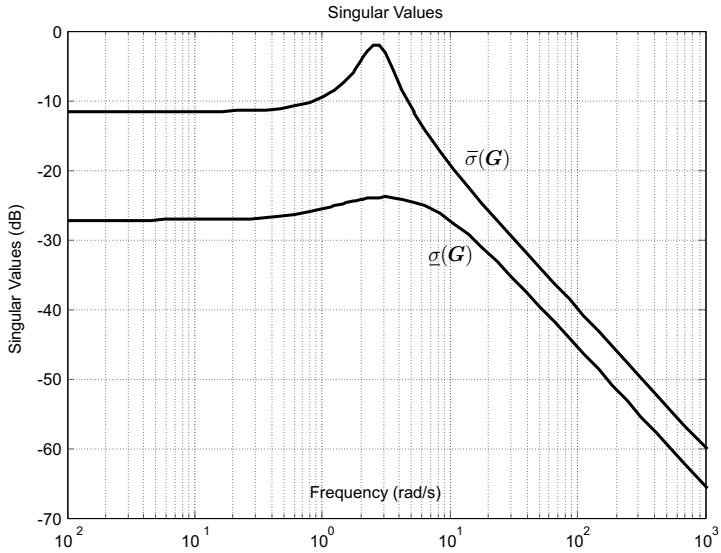


Fig. 8.13 SVs of the plant shown in Fig. 8.12 after uncertain dynamic inversion with 30 % uncertainty on the plant parameters selected according to (8.27)

8.5.1 Architecture of the Controllers

Figure 8.14 shows the complete architecture of the roll and yaw rate controller $\boldsymbol{K}$.

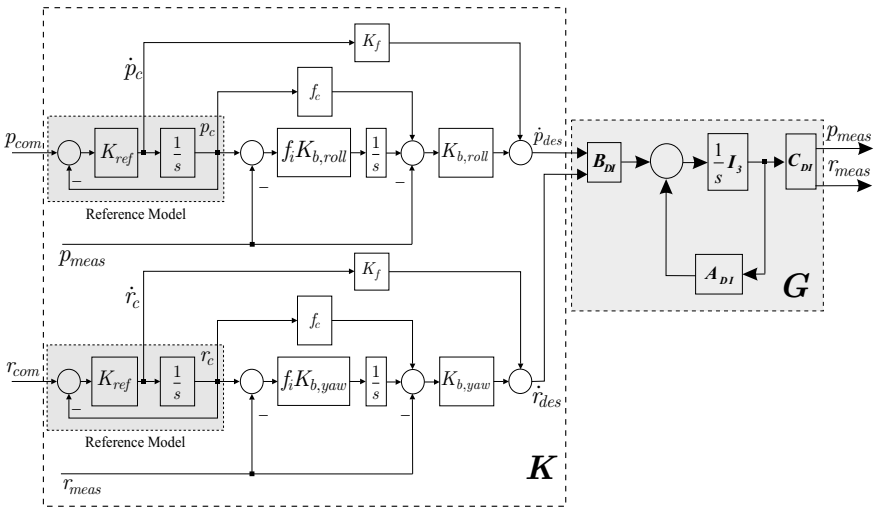


Fig. 8.14 Complete architecture of the roll and yaw rate controllers

8.5.2 Open-loop Analysis of the Roll and Yaw Rate Controllers

For the analysis of the loop gain (roll, yaw rate controllers and plant), Fig. 8.14 is modified as shown in Fig. 8.15, where the inner-rate feedback of the turn rates p_{meas} and r_{meas} are included as part of the new plant $\boldsymbol{G_1}$. The state space representation of the plant $\boldsymbol{G_1}$ is obtained as follows:

$$\dot{\boldsymbol{x}} = \left[\boldsymbol{A_{DI}} - \boldsymbol{B_{DI}} \begin{bmatrix} K_{b,roll} & 0 & 0 \\ 0 & K_{b,yaw} & 0 \end{bmatrix} \boldsymbol{C_{DI}} \right] \boldsymbol{x} + \boldsymbol{B_{DI}} \begin{bmatrix} K_{b,roll} & 0 \\ 0 & K_{b,yaw} \end{bmatrix} \begin{pmatrix} u_p \\ u_r \end{pmatrix} , \tag{8.28}$$

$$\boldsymbol{y}_{meas} = \boldsymbol{C_{DI}}_{(1:2,:)} \, \boldsymbol{x} \, . \tag{8.29}$$

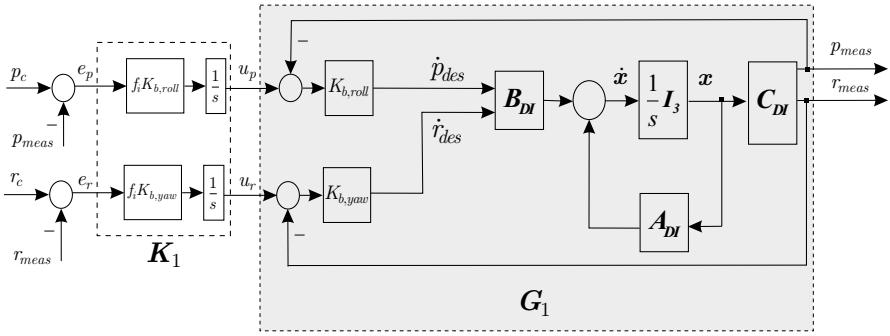


Fig. 8.15 Open-loop analysis of the roll and yaw rate controllers

The controller $\boldsymbol{K_1}$ is a block diagonal controller as follows:

$$\boldsymbol{K_1} = \begin{bmatrix} \frac{f_i \ K_{b,roll}}{s} & 0 \\ 0 & \frac{f_i \ K_{b,yaw}}{s} \end{bmatrix} . \tag{8.30}$$

The controller gains are $k_{b,roll}$=22 [s^{-1}] and $K_{b,yaw}$=23 [s^{-1}], and in both controllers the integrator coefficient is $f_i = 0.25$.

Figure 8.16 shows that the SVs of the open-loop gain $\boldsymbol{G_1 K_1}$ have been balanced such that they have approximately the same gain at low frequencies. Therefore, there is about the same control bandwidth for all the channels of the system. At low frequencies, the SVs have a slope of −20 dB/decade with a large gain in order to avoid steady-state error in tracking the commanded signal p_{com} and r_{com}.

Figure 8.17 shows the SVs of the open-loop gain $\boldsymbol{G_1 K_1}$, where the plant $\boldsymbol{G_1}$ is obtained with the parameter selection indicated in (8.24) to (8.27) with 30 % uncertainty on all the dynamic and control input matrix terms and 30 % uncertainty on all the measurement data.

8.5.3 Frequency-domain Stability and Robustness Bounds

8.5.3.1 Low Frequencies

The robust performance bounds are specified in Figs. 8.16 and 8.17. They are given in terms of the minimum SV $\underline{\sigma}(\boldsymbol{G_1 K_1})$ being sufficiently large at low frequencies.

To ensure that external disturbances such as wind or wind gusts are attenuated by at least a factor of 0.1 up to the frequency $\omega = 0.5$ rad/s (see

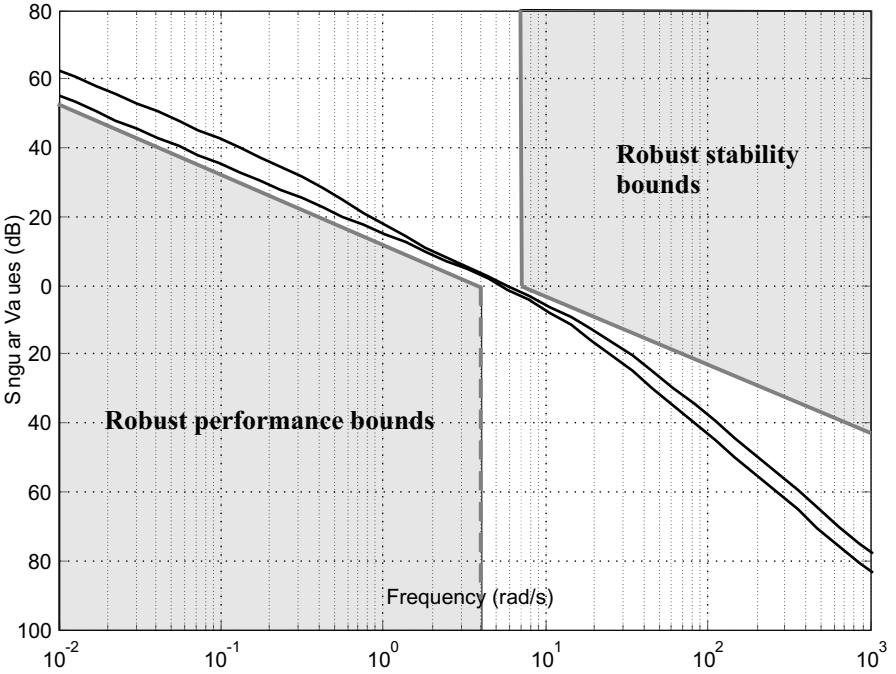


Fig. 8.16 SVs of the loop gain $\boldsymbol{G_1 K_1}$, with perfect dynamic inversion of the plant. $k_{b,roll}$=22 [s^{-1}] and $K_{b,yaw}$=23 [s^{-1}]

Military Specifications 1797, [1]), $\underline{\sigma}(\boldsymbol{G_1 K_1})$ should be larger than 20 dB up to the frequency $\omega = 0.5$ rad/s.

Moreover, the corner frequency ω_c of the loop gain should be larger than the frequency of the oscillatory Dutch roll mode $\omega_{Dutch,roll} = 4.95$ rad/s. The results shown in Fig. 8.16 satisfy all of the low-frequencies requirements.

The results of Fig. 8.17 are obtained with uncertainties in the dynamic inversion. The controlled system slightly violates the robust performance bounds at low frequencies, but this is not critical. The bandwidth is slightly too low, and the controller gains $K_{b,roll}$ and $K_{b,yaw}$ should be slightly increased.

8.5.3.2 High Frequencies

At high frequencies, the maximal SVs $\bar{\sigma}(\boldsymbol{G_1 K_1})$ are to be upper-bounded to guarantee stability robustness of the closed-loop system against uncertainties due to actuator modeling inaccuracies, neglected aircraft flexible modes, high-frequency unmodeled dynamics, plant parameter variations, and sensor noise and errors.

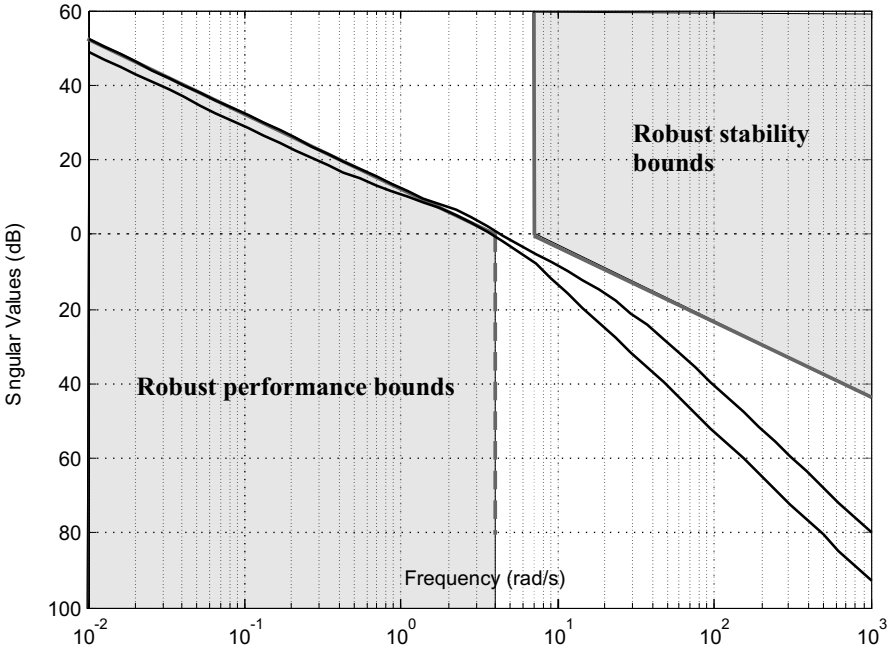


Fig. 8.17 SVs of the loop gain $\boldsymbol{G_1 K_1}$, with uncertain dynamic inversion of the plant by 30% uncertainty level on each variable according to (8.24) to (8.27). $k_{b,roll}$=22 [s^{-1}] and $K_{b,yaw}$=23 [s^{-1}]

The uncertainty model $m(\omega)$ is inspired by [1] and modified to assume that the rigid-body model of the aircraft is accurate to within 30 % up to a frequency of 3 rad/s, after which the uncertainty grows at the rate of 20 dB/decade. This uncertainty model has a transfer function as follows:

$$m(\omega) = \left| \frac{j\omega + 3}{10} \right| . \tag{8.31}$$

In order to guarantee stability robustness against modeling errors, the condition in (8.32) must be satisfied at the frequencies where plant uncertainties become significant, *i.e.*, $m(\omega) \gg 1$ or $1/m(\omega) \ll 1$ ($\omega > 7$ rad/s). This bound is shown in Figs. 8.16 and 8.17.

$$\bar{\sigma}\left(\boldsymbol{G_1 K_1}(j\omega)\right) < \frac{1}{m(\omega)} = \left| \frac{10}{j\omega + 3} \right| . \tag{8.32}$$

In the case of perfect dynamic inversion (Fig. 8.16) or imperfect dynamic inversion with 30 % uncertainty on all the plant coefficients (Fig. 8.17), the high-frequencies requirements are well satisfied.

8.6 Coordinated-turn Controllers

The lateral-directional control system is designed such that the aircraft makes coordinated turns by having the bank angle $\phi(t)$ follow some desired command $\phi_c(t)$ and maintaining at the same time the sideslip angle $\beta(t)$ at zero.

8.6.1 Sideslip Angle Controllers

The nonlinear differential equation of the sideslip angle is derived in Appendix D and is rewritten in a more convenient form in (8.33).

$$\dot{\beta} = \frac{1}{V_T}\left[g \sin\phi \cos\theta + a_y\right] + p \sin\alpha - r \cos\alpha \ . \tag{8.33}$$

By inverting (8.33), the commanded yaw rate is expressed as a nonlinear function of the desired rate of change of the sideslip angle as follows:

$$r_{com} = -\frac{\dot{\beta}_{des}}{\cos\alpha} + \frac{1}{V_T \cos\alpha}\left[g \sin\phi \cos\theta + a_y\right] + p \tan\alpha \ . \tag{8.34}$$

The complete architecture of the lateral-directional controller is shown in Fig. 8.18.

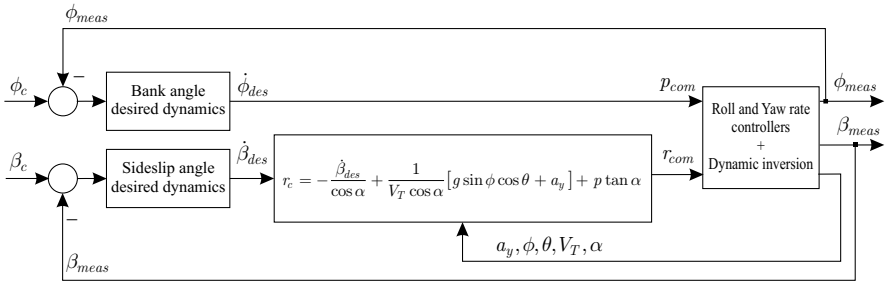


Fig. 8.18 Complete lateral-directional controller

8.6.2 Desired Dynamics of the Bank Angle

The controller for the desired dynamics of the bank angle ϕ is shown in Fig. 8.19. The controller gains are $K_{ref,\phi} = 2.7\,[\mathrm{s}^{-1}]$, $K_{f,\phi} = 1$, $f_{c,\phi} = 1$, $f_{i,\phi} = 0.25$, and $K_{b,\phi} = 3\,[\mathrm{s}^{-1}]$.

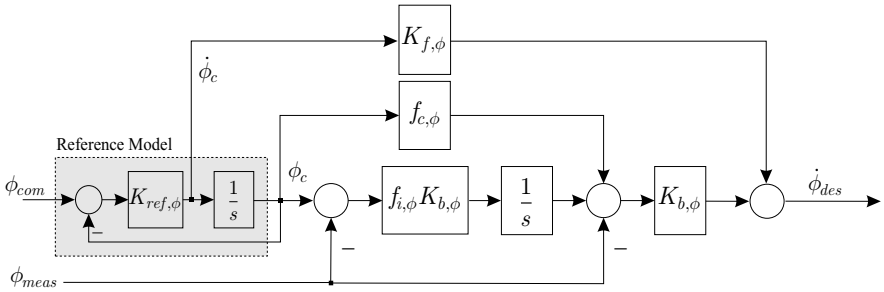


Fig. 8.19 Controller of the desired dynamics for the bank angle

8.6.3 Desired Dynamics of the Sideslip Angle

The controller for the desired dynamics of the sideslip angle β is shown in Fig. 8.20. The controller gains are $f_{c,\beta} = 0.5$, $f_{i,\beta} = 0.25$, and $K_{b,\beta} = 5\,[\mathrm{s}^{-1}]$.

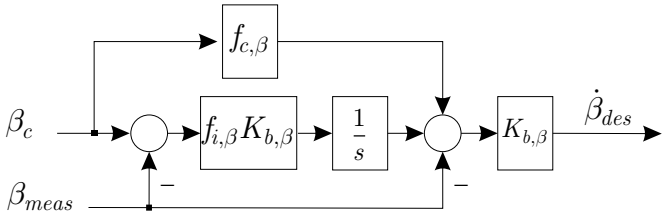


Fig. 8.20 Controller of the desired dynamics for the sideslip angle

8.6.4 Simulation Results

Figure 8.21 shows the results of a nonlinear simulation for aggressive lateral commands. The simulation also includes actuator dynamics and saturations. The top plot in Fig. 8.21 shows the reference signal of the roll angle. The aircraft is capable of tracking a step roll angle input with no overshoot with zero steady-state error and very good time response. There is a systematic small tracking error when the reference roll angle signal is a ramp, but this is acceptable. The tracking of the roll and yaw rates commands are accurate and very fast, with no tracking error for a step input or a ramp, which is very satisfactory. The controller for the sideslip angle has good performance as shown in the bottom plot of Fig. 8.21. The overall control performance of the lateral-directional motion is very satisfactory.

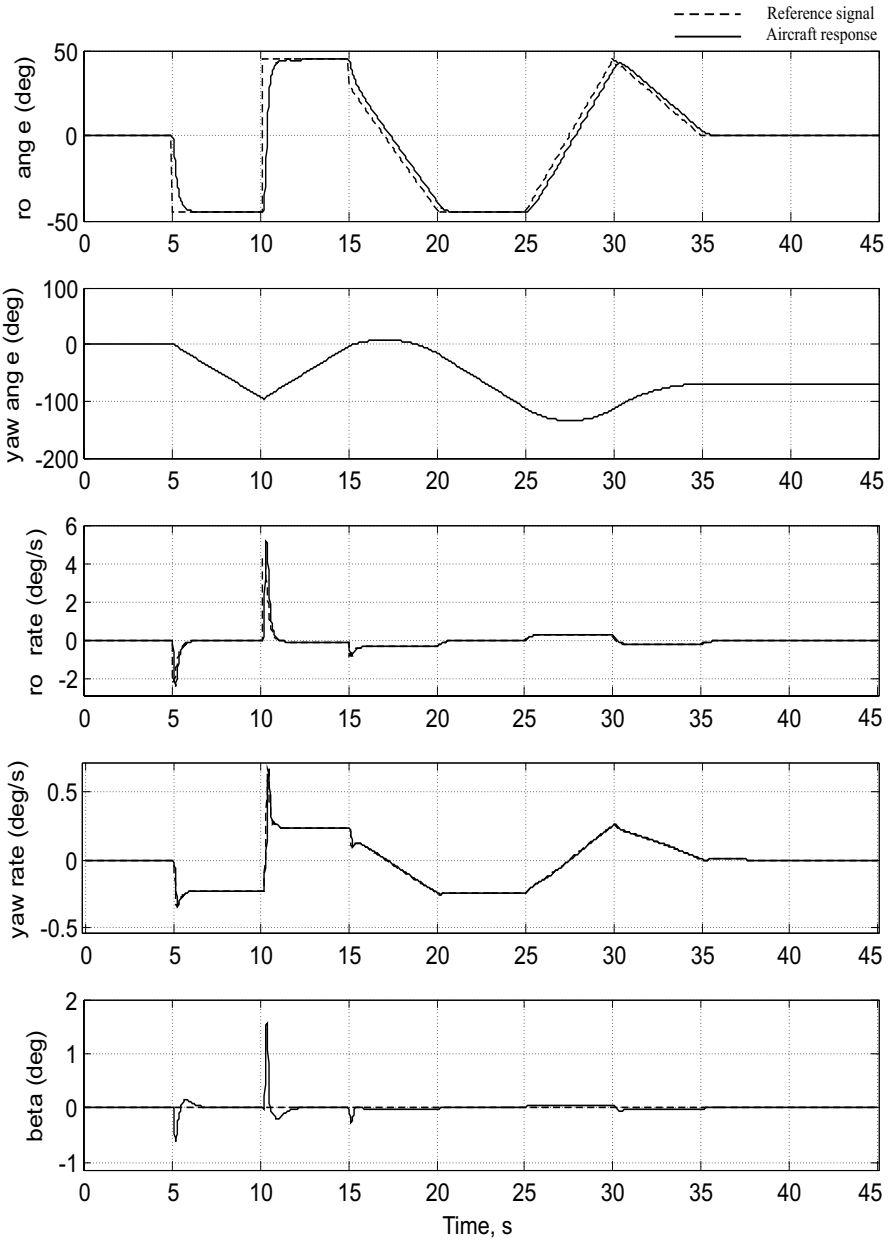


Fig. 8.21 Response of the controlled aircraft for aggressive lateral commands. Nonlinear simulations in the nominal case (no plant uncertainty). V_T=30 m/s, h=500 m

References

1. B. Stevens and F. Lewis. *Aircraft Control and Simulation, Second Edition*. Wiley, New York, NY, 2003.
2. R. F. Stengel. *Flight Dynamics*. Princeton University Press, 2004.
3. S. Skogestad and I. Postlethwaite. *Multivariable Feedback Control, Second Edition*. Wiley, 2005.

Chapter 9
Reconfigurable Guidance System

This chapter presents a guidance algorithm for a UAV. It combines a nonlinear lateral guidance control law, originally designed for UAVs tracking circles for mid-air rendezvous, with a new simple adaptive path-planning algorithm. Preflight path planning consists only of storing a few waypoints guiding the aircraft to its targets. The chapter presents an efficient way to model no-fly zones (NFZ), to generate a path in real time to avoid known or "pop-up" obstacles, and to reconfigure the flight path in the event of reduced aircraft performance. Simulation results show the good performance of this reconfigurable guidance system which, moreover, is computationally efficient [1, 2].

9.1 Introduction

Over the last two decades, many path-planning algorithms have been investigated, especially for ground robots, for a single UAV, and more recently for a formation of UAVs. Among the methods used in path planning, we can mention the PRM method [3], which explores all the possible paths within the space surrounding the vehicle and finally selects the lowest cost route. However, the computational load makes the PRM method impractical for real-time path planning in small UAVs. An extension to the PRM method has recently been presented in [4]. It is called modified *rapidly-exploring random trees*, which is capable of efficiently searching for feasible paths in the space while taking into account constraints from the vehicle performance. However, efforts are still going on to implement an on-the-fly path-replanning system as pop-up obstacles are discovered or when the performance of the vehicle degrades.

There are other methods based on *potential field* functions. However, the primitive forms of potential field functions present some difficulties when choosing an appropriate potential function, and the algorithm may be stuck

at some local minimum [5]. Since then, a whole family of potential field methods with superior performance has been developed. They are known as *navigation functions* [6, 7]. Other path-planning techniques are based on optimization methods, such as *mixed integer linear programming* or MPC techniques [8], which still involve intensive computations.

In this chapter, we present a reconfigurable guidance algorithm for a UAV. It newly combines the lateral guidance control law from [9] and [10], originally designed for UAVs tracking circles for mid-air rendezvous, with a new, simple adaptive path-planning algorithm, which takes advantage of the curve path-following property of the above-mentioned lateral guidance law. This path-planning method generates on-line a flight path based on predefined waypoints, takes into account the aircraft performance, avoids known or appearing obstacles, is simple to implement, and requires low computational power.

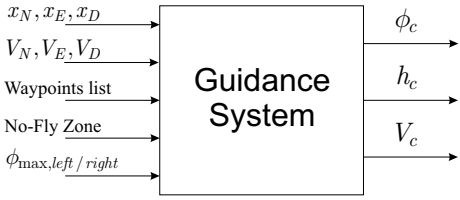


Fig. 9.1 Guidance system inputs and outputs

As shown in Fig. 9.1, the guidance system needs six inputs. The first input concerns the aircraft's current ground position ($x_N,\ x_E,\ x_D$). The second input is the aircraft's ground velocity ($V_N,\ V_E,\ V_D$). The mission of the aircraft is defined by a list of waypoints through which the aircraft is to fly. Furthermore, if in the area of the flight operation some obstacles or NFZ are known in advance or appear during the flight, their location and dimensions can be specified to the guidance system via the fourth input. A constraint on the maximum bank angle $\phi_{max,left/right}$ is given to the guidance system. Finally, the parameter τ_{roll} is provided as an estimate of the maximum time needed to bank the aircraft to ϕ_{max}. Note that the last four inputs can be changed dynamically, and the first two inputs are obviously constantly updated.

The outputs of the guidance system are the commanded bank angle ϕ_c, whose value is computed by the lateral guidance system detailed in Sects. 9.2 and 9.3. The altitude command signal h_c is computed by the altitude guidance system described in Sect. 9.4. Finally, the commanded aircraft velocity V_c can be adaptively controlled by the guidance system in order to efficiently avoid obstacles and reach the goals of the mission optimally. Note, however, that in this chapter the velocity command V_c is kept to a constant value.

9.2 Lateral Guidance System

9.2.1 Lateral Guidance Control Law for Trajectory Tracking

Consider Fig. 9.2, where an aircraft has to be guided to track the desired path. From the current location of the aircraft O, we can draw a circular arc that intersects the desired path at a "reference point" P, where R is the radius of the circle-arc OP, L_1 is the segment that joins the center of the aircraft O to the reference point P, and η is the angle between the aircraft's velocity vector and the line L_1.

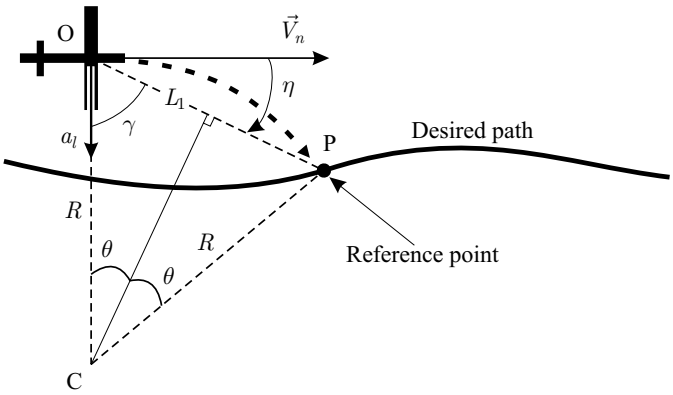


Fig. 9.2 Guidance law geometry

The lateral acceleration required to bring the aircraft to the reference point following the arc of a circle is

$$a_l = \frac{V_n^2}{R} \,, \tag{9.1}$$

where the ground speed of the aircraft (taken in the local navigation frame) is $V_n = \sqrt{V_N^2 + V_E^2}$. Let us express R in terms of the distance L_1 and the angle η. The triangle (OCP) is isosceles in C, therefore, we have $L_1 = 2R\sin\theta$, or also $L_1 = 2R\cos\gamma$. Moreover, the angle $\gamma = \frac{\pi}{2} - \eta$, and consequently, the length L_1 can be expressed in terms of the angle η as follows:

$$L_1 = 2R\sin\eta \,, \tag{9.2}$$

$$\iff R = \frac{L_1}{2\sin\eta} \,. \tag{9.3}$$

The lateral acceleration in (9.1) can now be written as

$$a_l = \frac{2V_n^2}{L_1} \sin \eta \ . \tag{9.4}$$

In turn, the lateral acceleration a_l is converted to a bank angle command, $\phi_{com} \approx a_l/g$ (see Appendix D.2).

9.2.2 Advantages and Properties of the Method

9.2.2.1 Remarkable Performance for Curve Path Following

This control law is remarkable in the sense that it is particularly suited to fly circles. Indeed, if the aircraft is following a desired circular path, then the acceleration command a_l generated by the guidance system is exactly the same as the associated centripetal acceleration. In other words, the guidance logic chooses a reference point on the desired path at a distance L_1 ahead of the aircraft, and generates the acceleration command that would lead the vehicle to hit the point after flying a circular arc, thus leading to zero steady-state error for a circular path. As shown in Chap. 3 of [9], the performance of such a lateral guidance law for circle following in the presence of wind is superior to that obtained with PD or PID controllers. With the nonlinear guidance law $a_l = \frac{2V_n^2}{L_1} \sin \eta$, the vehicle ground speed V_n is used to generate the acceleration command, which intrinsically takes into account the inertial velocity changes due to the wind effects, and adapts to the situation accordingly.

9.2.2.2 Properties Associated with the Angle η

Equation 9.4 shows that the direction of the acceleration a_l depends on the sign of η. For example, if the reference point is on the right side of the direction of the aircraft velocity vector, then the aircraft will be commanded to accelerate to the right, and finally the aircraft will tend to align its velocity vector with the direction of $\boldsymbol{L_1}$.

In practice the distance L_1 is fixed to a certain value, and two cases arise:

- If the aircraft is far away from the desired path, then the direction of $\boldsymbol{L_1}$ makes a large angle with the desired path. Therefore, the guidance law chooses the reference point on the desired trajectory in such a way that the aircraft rotates its velocity direction to approach the desired path at a large angle.
- If the aircraft is close to the desired path, then the direction of L_1 makes a small angle with the desired path. Therefore, the guidance law chooses the reference point on the desired trajectory in such a way that the aircraft rotates its velocity direction to approach the desired path at a small angle.

Since the angle η contains the information about the upcoming path, this geometric factor has the effect of a feedforward control.

9.2.3 Drawback of the Method

A drawback of the control law in (9.4) is that it does not contain any element of an integral control. Therefore, the lateral guidance law requires to be provided with an unbiased lateral acceleration measurement, and a non-biased bank angle estimate.

9.2.4 Selection of L_1

The design parameter in the lateral guidance logic is the distance L_1 between the vehicle and the reference point as shown in Fig. 9.3. It is explained in [9] that for a small magnitude of η, the guidance formula can be approximated in terms of the cross-track error y as follows:

$$a_l = \frac{2V_n^2}{L_1} \sin\eta \approx 2\frac{V_n}{L_1}\left(\dot{y} + \frac{V_n}{L_1} y\right) \ . \tag{9.5}$$

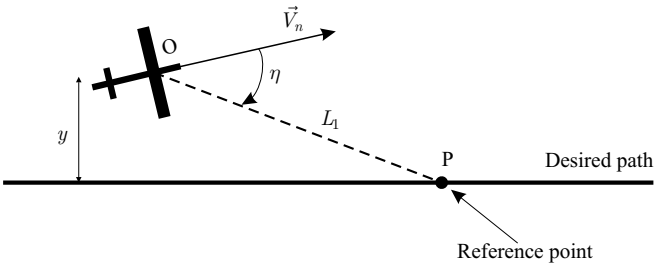


Fig. 9.3 Guidance law geometry

Equation 9.5 shows that the guidance law is equivalent to a PD controller, in which the ratio between the vehicle speed V_n and the distance L_1 is an important factor that behaves as the gain of the controller. A small value of L_1 leads to a high control gain and *vice versa*. The control gain is limited by the inner loop bank control bandwidth (2−3 rad/s). With a nominal flight velocity of around 25 m/s, the distance L_1 has been chosen to be $L_1 = 150\,\text{m}$.

9.2.5 Path-planning Objective

The main objective of this chapter is to generate on-line an appropriate reference path from which the reference point P can be selected and used by the lateral guidance control law.

9.3 Regular Waypoint Tracking

The regular waypoint tracking algorithm guides the aircraft through the predefined waypoints. An imaginary segment joins two consecutive points, and we define the *segment* k to be the segment that joins the waypoints WP_k and WP_{k+1}. The reference point, P, from the guidance law presented above, lies on this segment and is L_1 distant from the center of the aircraft; see Fig. 9.4.

9.3.1 Computation of the Reference Point P

The angle of the segment k with respect to North is defined as follows:

$$\psi_{seg(k)} = \tan^{-1}\left(\frac{WP_{k+1,E} - WP_{k,E}}{WP_{k+1,N} - WP_{k,N}}\right) \in [-\pi;\pi]\,. \tag{9.6}$$

The coordinates of the current location of the center of the aircraft are in the North−East plane (X_N, X_E). The angles χ and λ, and the distance d_1 as shown in Fig. 9.4 can be computed as follows:

$$\begin{aligned}
\chi &= \tan^{-1}\left(\frac{X_E - WP_{k,E}}{X_N - WP_{k,N}}\right) \in [-\pi;\pi]\,,\\
\lambda &= |\psi_{seg(k)}| - |\chi|\,,\\
d_1 &= \sqrt{(X_E - WP_{k,E})^2 + (X_N - WP_{k,N})^2}\,.
\end{aligned} \tag{9.7}$$

The distance between WP_k and the reference point P is given by

$$[WP_k, P] = [WP_k, H] + [H, P] = d_1\cos\lambda + \sqrt{L_1^2 - {d_1}^2\sin^2\lambda}\,. \tag{9.8}$$

Finally, the coordinates of the reference point P can be computed with

$$\begin{aligned}
P_N &= WP_{k,N} + [WP_k, P]\cos\psi_{seg(k)}\,,\\
P_E &= WP_{k,E} + [WP_k, P]\sin\psi_{seg(k)}\,.
\end{aligned} \tag{9.9}$$

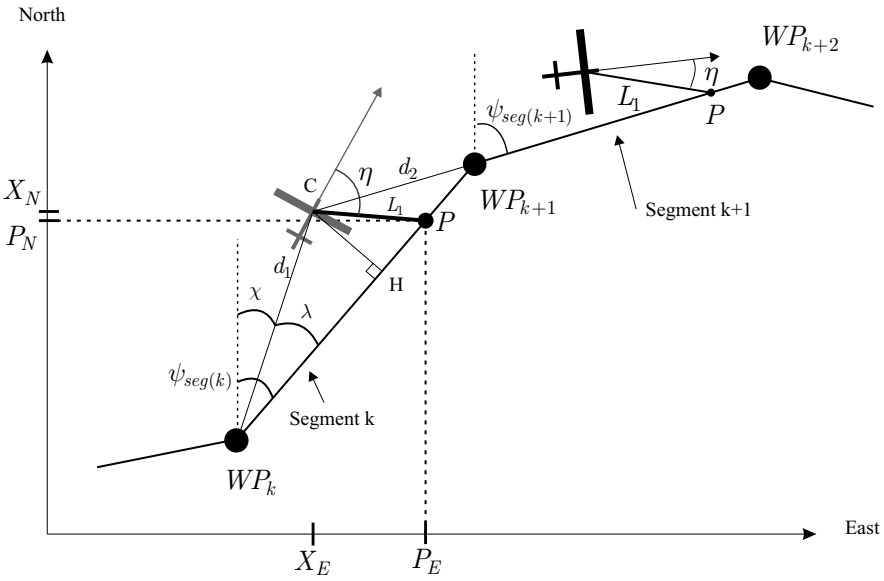


Fig. 9.4 Regular waypoint tracking

9.3.2 Logic for Segment Switching

As the aircraft flies, the reference point P also moves along the desired trajectory, which consists of consecutive segments. Therefore, the guidance algorithm has to select properly the current segment on which the reference point P is to be selected.

Equation 9.10 gives the two conditions that are continuously checked:

$$d_2 \geq L_1, \qquad d_1 \cos\lambda < \|WP_k, WP_{k+1}\| \; . \tag{9.10}$$

If one of the two conditions is not satisfied anymore, the guidance system has to select a reference point P on the next segment $[WP_{k+1}, WP_{k+2}]$. On the other hand, if the two conditions are still satisfied, the guidance system computes a reference point P on the segment $[WP_k, WP_{k+1}]$. In that case, the path-planning system has to further check the lateral distance between the aircraft and the segment followed in order to make sure that it selects a point on the current segment that is L_1 distant from the aircraft.

Two subcases can be distinguished:

- If $|\lambda| > \pi/2$, the aircraft is somewhere behind the first point of the current segment. In that case, the distance L_1 is selected as $L_1 = \max(L_1, d_1)$.
- If $|\lambda| < \pi/2$, the guidance algorithm proceeds in checking the lateral distance $[C, H] = d_1 \sin\lambda$. If $[C, H] > L_1$, then the distance L_1 is assigned a new value with $L_1 = 1.1 \times [C, H]$.

In this way, we always ensure that the arm L_1 is long enough to intersect the desired trajectory, so that a reference point P always exists.

9.3.3 Computation of the Roll Angle Command ϕ_{com}

The direction of the aircraft's ground speed is computed with

$$\psi = \tan^{-1}\left(\frac{V_E}{V_N}\right) \quad \in [-\pi; \pi] \; . \tag{9.11}$$

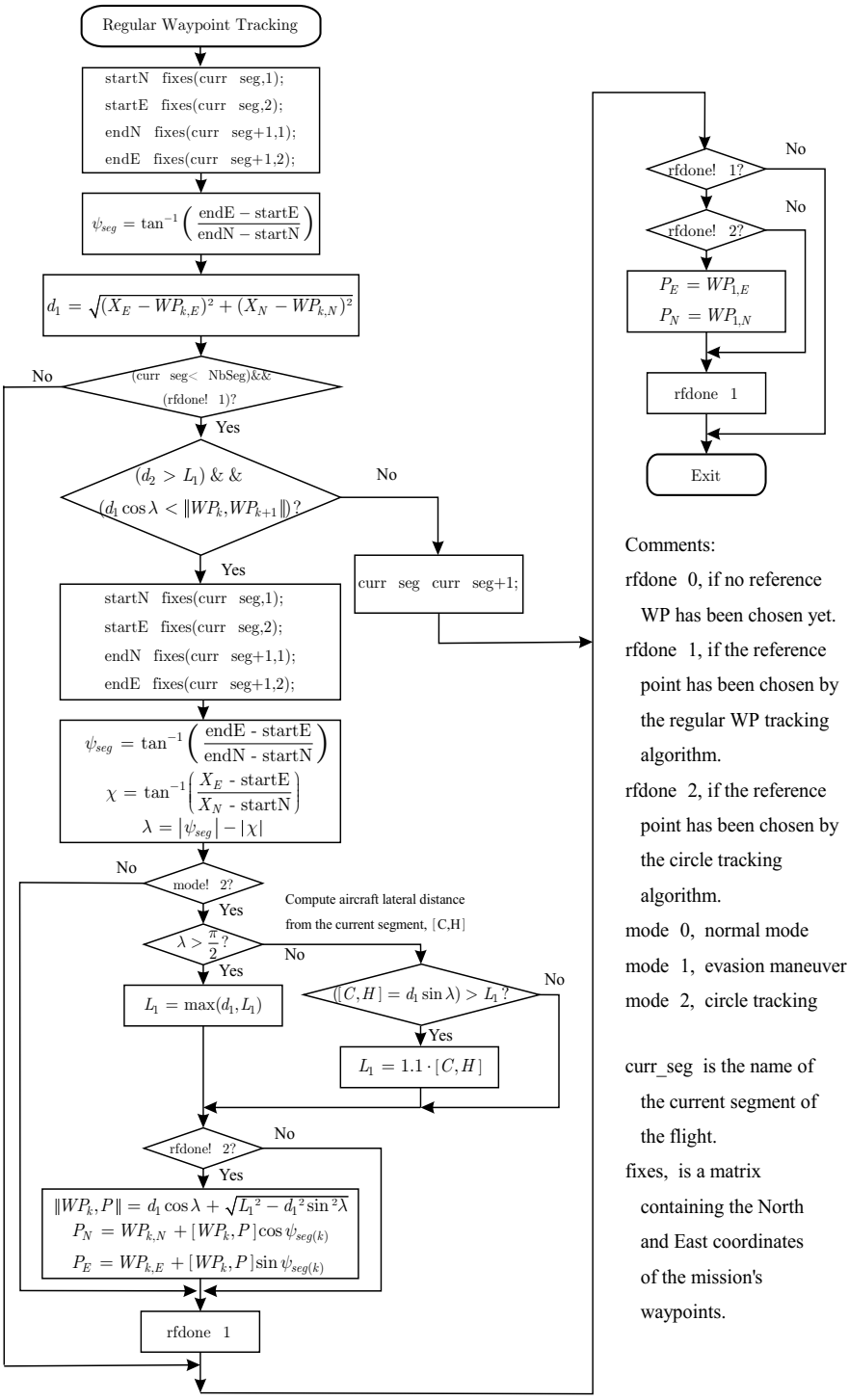
The lateral guidance control law needs the angle η, which is computed as follows (see Fig. 9.6):

$$\begin{aligned} \eta &= \Omega - \psi \; , \\ \Omega &= \tan^{-1}\left(\frac{P_E - X_E}{P_N - X_N}\right) \quad \in [-\pi; \pi] \; . \end{aligned} \tag{9.12}$$

In practice, we want the angle η to be in the range $[-\pi; \pi]$, therefore the following code implementation is used

$$\begin{aligned} &\text{while } (\eta > \pi) \;\; \eta = \eta - 2\pi \; , \\ &\text{while } (\eta < -\pi) \;\; \eta = \eta + 2\pi \; . \end{aligned} \tag{9.13}$$

Once the angle η is in the range $[-\pi; \pi]$, the angle η is further limited (saturated if needed) to be in the range $[-\pi/2; \pi/2]$. The reason for this is that when the angle η becomes large and approaches $\pi/2$ (or $-\pi/2$, respectively) we want the aircraft to bank to the right (or to the left, respectively) as much as is possible to quickly come closer again to the reference trajectory. Indeed, since the angle η is used in the roll angle command with $\phi_{com} = 2V_n^2 \sin\eta/(L_1 g)$, if η exceeded $\pi/2$ $(-\pi/2)$ in the range $[\pi/2; \pi]$ $([-\pi/2; -\pi])$ the term $\sin\eta$ would decrease in amplitude, which means that the commanded roll angle also decreases, and the aircraft would come back more slowly to the reference trajectory, which is not desired. Finally, the control signal for the roll angle ϕ_{com} is saturated within the range $[-\phi_{max}; \phi_{max}]$.



Comments:

rfdone 0, if no reference WP has been chosen yet.

rfdone 1, if the reference point has been chosen by the regular WP tracking algorithm.

rfdone 2, if the reference point has been chosen by the circle tracking algorithm.

mode 0, normal mode

mode 1, evasion maneuver

mode 2, circle tracking

curr_seg is the name of the current segment of the flight.

fixes, is a matrix containing the North and East coordinates of the mission's waypoints.

Fig. 9.5 Regular waypoint tracking

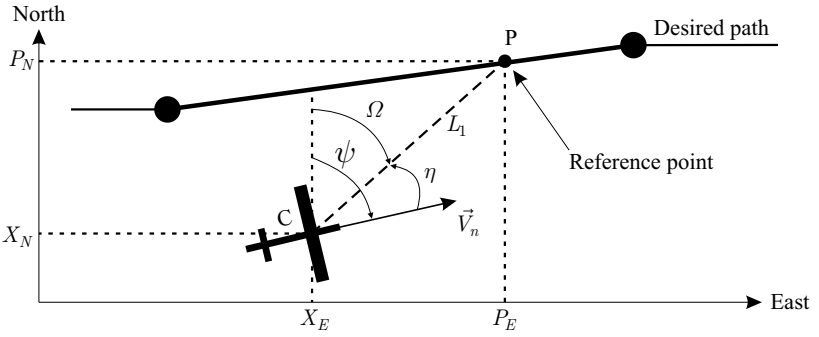


Fig. 9.6 Determination of the angle η

9.4 Altitude Guidance Law

The altitude control signal is h_c and is computed as follows:

$$h_c = WP_{k,D} + \gamma d_1 \cos\lambda \,, \tag{9.14}$$

with the desired flight path angle defined as follows:

$$\gamma = \tan^{-1}\left(\frac{WP_{k+1,D} - WP_{k,D}}{\sqrt{(WP_{k+1,N} - WP_{k,N})^2 + (WP_{k+1,E} - WP_{k,E})^2}}\right) \,, \tag{9.15}$$

with λ and d_1 defined in (9.7).

Figure 9.7 shows the aircraft flying from waypoint WP_k to WP_{k+1}. The desired altitude is h_c computed as a function of the aircraft's ground position. The aircraft's current altitude is X_D, which is obviously too high compared with h_c in Fig. 9.7, and the altitude controller of Section 7.5 must correct this altitude error.

Remarks:

- Equations 9.14 and 9.15 are used to generate the altitude reference signal in the normal waypoint tracking mode.
- In the modes *evasion maneuver* (mode 1) or *circle tracking* (mode 2), the altitude reference signal is kept at the value where it was before the guidance system entered one of these modes. Indeed, in mode 1 or 2 the aircraft takes a path that is not known in advance and for which no altitude may be specified. Once the guidance system switches back to *normal waypoint*

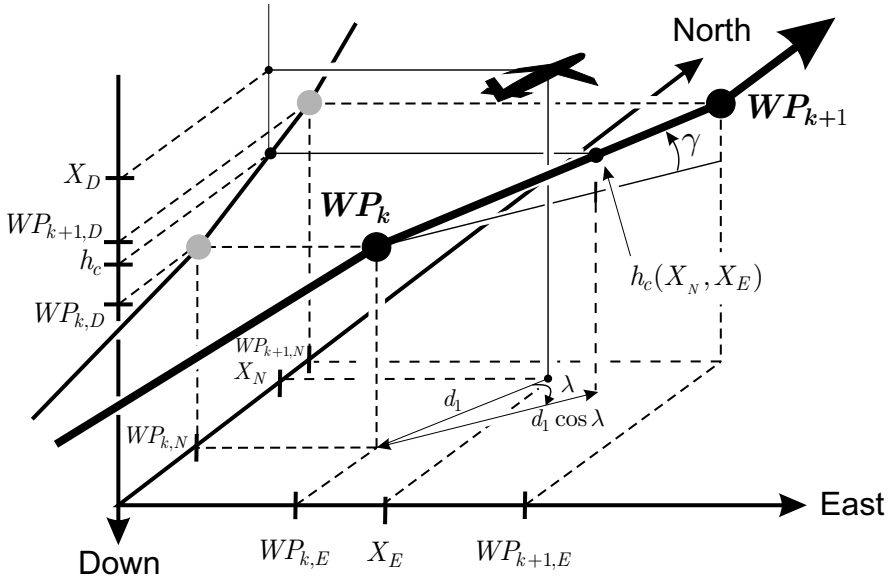


Fig. 9.7 Altitude tracking geometry

tracking mode (mode 0), the altitude control signal is again constantly recomputed.

9.5 NFZ and Obstacles

Before the flight, the location of any known NFZ are stored in the memory of the autopilot. If the UAV is equipped with scanning sensors that can detect pop-up obstacles, our path-planning system will recompute on the fly a new trajectory. It determines whether an NFZ or an obstacle interferes with the planned path by using an imaginary "detection line" of length R_{LA} in front of the aircraft, as shown in Fig. 9.8. The distance R_{LA} defines the so-called "look-ahead distance". If any part of this detection line penetrates an NFZ or an obstacle, avoidance action (mode 1) is immediately taken as described in the next section.

9.5.1 Definition of an NFZ

An NFZ is any airspace in which an aircraft is not permitted to fly. This airspace can be of any arbitrary shape. However, in order to simplify the

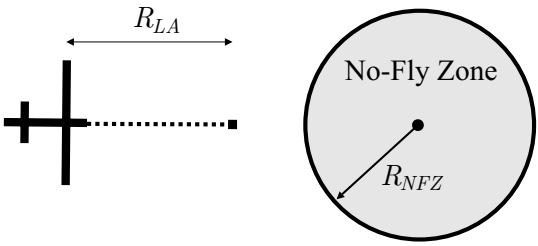


Fig. 9.8 NFZ and look-ahead distance R_{LA}

guidance algorithm, two conditions are imposed on how the NFZ is represented.

First, the vertical limits of the NFZ are not considered such that the NFZ is essentially a 2D surface. The aircraft is not allowed to pass over the NFZ. Second, the shape of the NFZ is chosen to be a circle. In this way, the avoidance maneuver can be an arc of a circle in order to benefit from the guidance control law especially suited to track circles, which are described by only two parameters, namely their center and their radius.

Although this chapter discusses the avoidance of one circular NFZ only, the algorithm can be extended to multiple NFZs with some simple modifications. Also, a complex NFZ shape can be represented by multiple circles.

9.5.2 Choice of an Appropriate Look-ahead Distance R_{LA}

The distance R_{LA} defines the so-called "look-ahead distance". If any part of this detection line penetrates an NFZ or an obstacle, avoidance action is immediately taken as described in the next section.

The look-ahead distance R_{LA} is chosen such that the aircraft will fly an arc that stays just outside the NFZ and start the evasion maneuver as late as possible. The value of R_{LA} depends on the radius R_{NFZ} of the NFZ, the ground speed of the aircraft V_n, and the maximum bank angle of the aircraft ϕ_{max}. Given these parameters, and assuming a coordinated turn, the minimum turn radius that the aircraft can fly is given by

$$R_{min} = \frac{V_n^2}{g \tan(\phi_{max})} ,$$
$$V_n = \sqrt{V_N^2 + V_E^2} . \tag{9.16}$$

The subscript n attached to the ground speed of the aircraft V_n indicates that the speed is taken in the local navigation frame.

In the case of an NFZ with infinite radius, the aircraft would have to make a 90° turn, in which case $R_{LA,min} = R_{min}$. For any NFZ with a finite radius, the aircraft has to turn less than 90° to avoid it. Figure 9.9 shows the situation where the aircraft is at the point where it begins its turn and is guided so that its path becomes tangent to the edge of the NFZ. A triangle can be set up with vertices at the center of the NFZ, at the center of the aircraft, and at a point R_{min} off the right wing-tip. The minimum look-ahead distance $R_{LA,min}$ is constructed using the Pythagorean theorem

$$(R_{LA,min} + R_{NFZ})^2 + R_{min}^2 = (R_{min} + R_{NFZ})^2 \, . \tag{9.17}$$

By expanding both sides and regrouping the terms, we obtain the following equation, where the distance $R_{LA,min}$ is the unknown variable

$$R_{LA,min}^2 + 2R_{LA,min}R_{NFZ} - 2R_{min}R_{NFZ} = 0 \, . \tag{9.18}$$

Equation 9.18 admits two real solutions of which we only keep the positive one

$$R_{LA,min} = \sqrt{R_{NFZ}}\sqrt{R_{NFZ} + 2R_{min}} - R_{NFZ} \, . \tag{9.19}$$

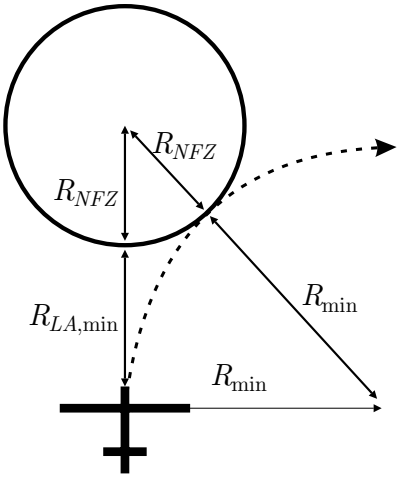


Fig. 9.9 Approaching an NFZ or an obstacle. Definition of the look-ahead distance R_{LA}

To obtain the final value for R_{LA}, compensation must be made for the delay needed to initiate the turn and to bank to ϕ_{max}. The assumption is made that while the aircraft is initiating the turn, it continues to fly level, and then as soon as it reaches ϕ_{max} it takes a minimum radius turn. The characteristic time τ_{roll} to roll to ϕ_{max} can be multiplied by the aircraft's speed to obtain the distance the aircraft will travel during this delay, which

is added to $R_{LA,min}$. The resulting look-ahead distance is

$$R_{LA} = R_{LA,min} + V_n \, \tau_{roll} \; . \tag{9.20}$$

9.6 Detection of the NFZ

As mentioned before, the algorithm monitors an "imaginary" line of length R_{La} ahead of the aircraft, and checks if it penetrates any NFZ or obstacle. First, the distance D_{NFZ} from the aircraft to the center of the NFZ is calculated; see Fig. 9.10:

$$D_{NFZ} = \sqrt{(NFZ_N - X_N)^2 + (NFZ_E - X_E)^2} \; . \tag{9.21}$$

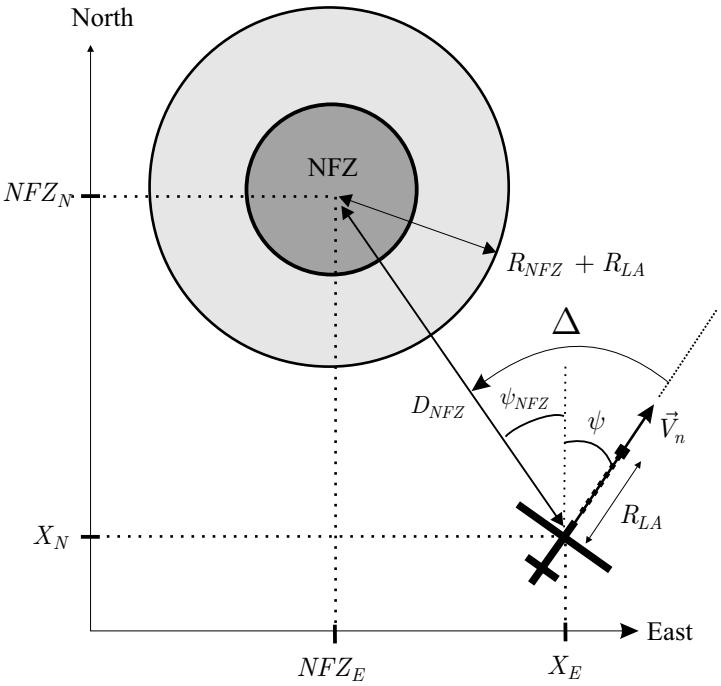


Fig. 9.10 Geometry of the NFZ approach

If the condition

$$D_{NFZ} \leq R_{NFZ} + R_{LA} \tag{9.22}$$

is satisfied, where the length R_{NFZ} is the radius of the NFZ, then the aircraft is considered to be within range of the NFZ, which means that the aircraft

is in the light-gray area in Fig. 9.10. In this case, a further check is made to see if a part of the detection line is touching the NFZ.

There are two possible cases shown in Figs. 9.11 and 9.12 depending on the orientation of the aircraft. The distances D_{NFZ} and R_{LA} are known. We also define the angle ψ_{NFZ} between the North direction at the center of the aircraft and the segment of length D_{NFZ} as follows:

$$\psi_{NFZ} = \tan^{-1}\left(\frac{NFZ_E - X_E}{NFZ_N - X_N}\right) . \tag{9.23}$$

A pair of triangles is created as shown in Fig. 9.11. The length of the edges y and a can easily be calculated, using

$$\begin{aligned} \Delta &= |\psi_{NFZ} - \psi| , \ \in [-\pi; \pi] , \\ y &= D_{NFZ} \cdot \sin \Delta , \\ a &= D_{NFZ} \cdot \cos \Delta . \end{aligned} \tag{9.24}$$

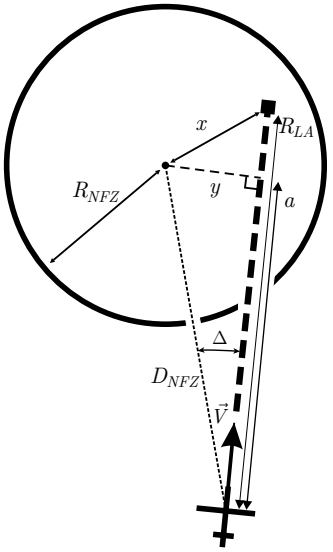


Fig. 9.11 Diagram of NFZ detection algorithm, Case 1 (detected)

Case 1: $a \leq R_{LA}$

Case 1 applies if $a \leq R_{LA}$. The limiting case occurs when edge a is tangent to the NFZ. Therefore, y will have a length equal to R_{NFZ}. Thus, the NFZ touches the detection line if $y \leq R_{NFZ}$.

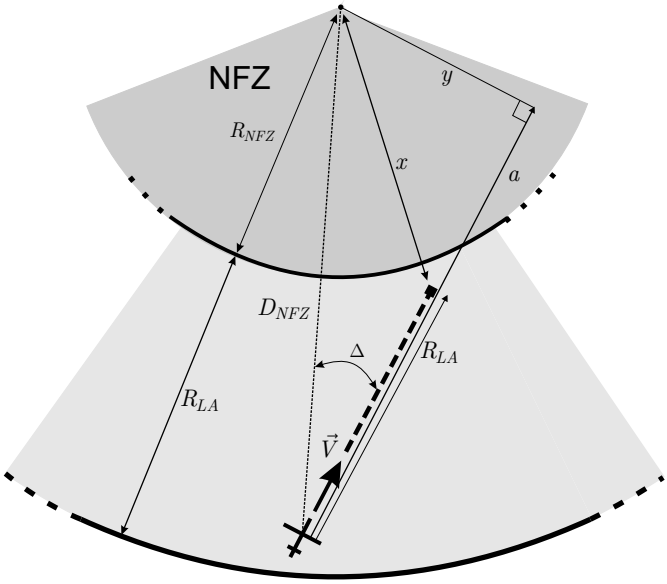


Fig. 9.12 Diagram of NFZ detection algorithm, Case 2 (not detected)

Case 2: $a > R_{LA}$

Case 2 applies if $a > R_{LA}$. The limiting case occurs when the end of the detection line is on the edge of the NFZ. This can be checked by comparing the length of edge x with the radius of the NFZ, so that the NFZ touches the detection line if $x \leq R_{NFZ}$, where $x = \sqrt{y^2 + (a - R_{LA})^2}$.

The check for Case 1 or Case 2 is only done if Δ is less than or equal to 90°. If Δ is greater than 90°, then the center of the NFZ lies behind the aircraft and no action is taken. Figure 9.13 shows the diagram of the NFZ detection algorithm.

9.7 NFZ Avoidance Algorithm

The NFZ avoidance algorithm guides the aircraft around any NFZ that the aircraft encounters. The avoidance method is designed to be simple to implement while allowing the aircraft to reach waypoints close to the edge of the NFZ.

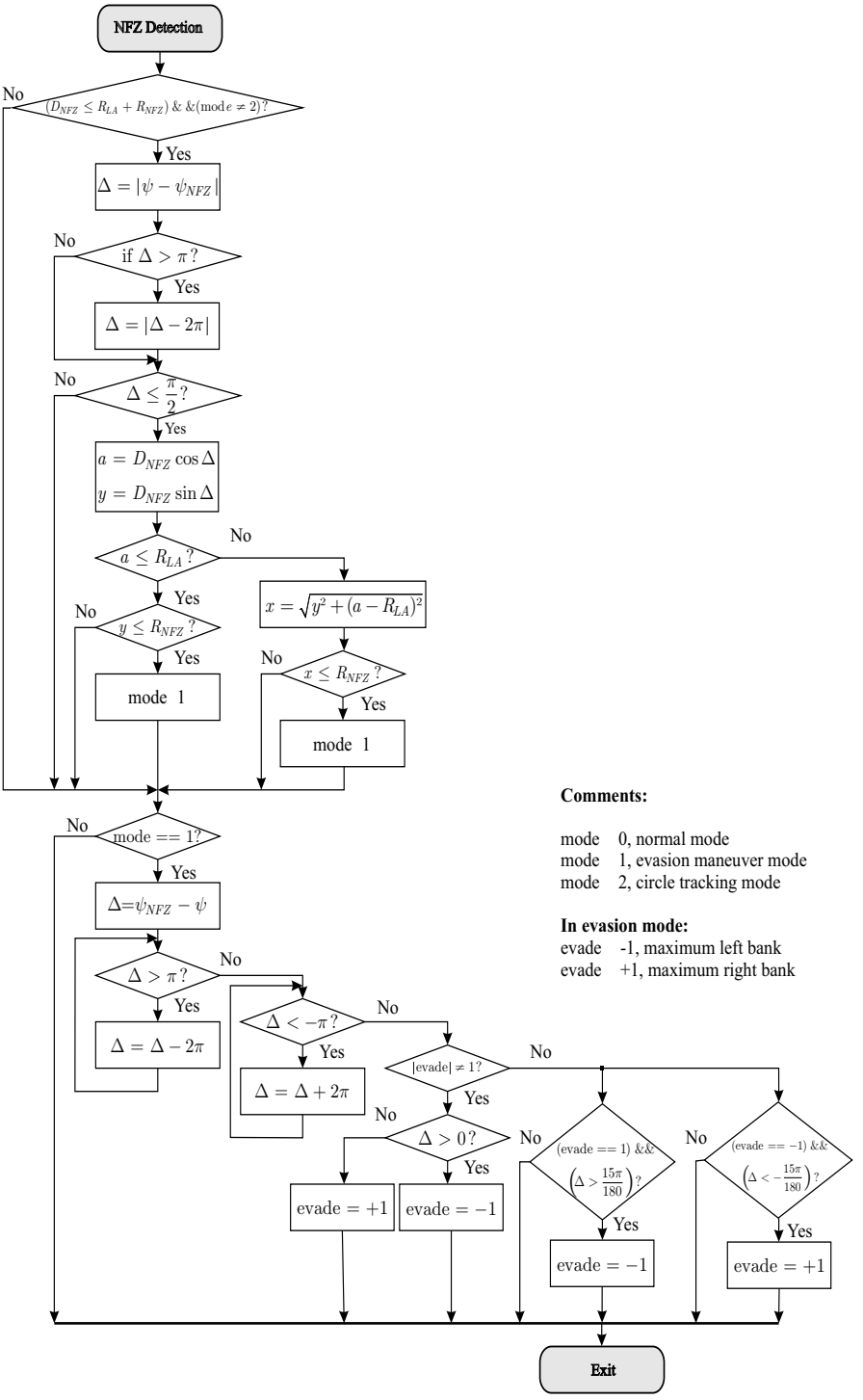


Fig. 9.13 Diagram of NFZ detection algorithm

9.7.1 On-line Selection of an Avoidance Path Template

One key feature of this avoidance method is the on-line generation of a circular arc around the NFZ as a reference path, drawn as a dashed line in Fig. 9.14. Such a path minimizes the distance the aircraft flies to avoid the NFZ. Moreover, we saw at the beginning of this chapter that the lateral guidance control law is particularly efficient in tracking circles.

Furthermore, choosing the reference path to be circular allows the template path to be easily defined in relationship to the NFZ dimensions. It is indeed defined by the center of the NFZ and a path radius, R_1, which is simply the NFZ radius plus a safety margin. The aircraft follows this path until it is able to continue towards the next waypoint in a straight line and without passing through the NFZ.

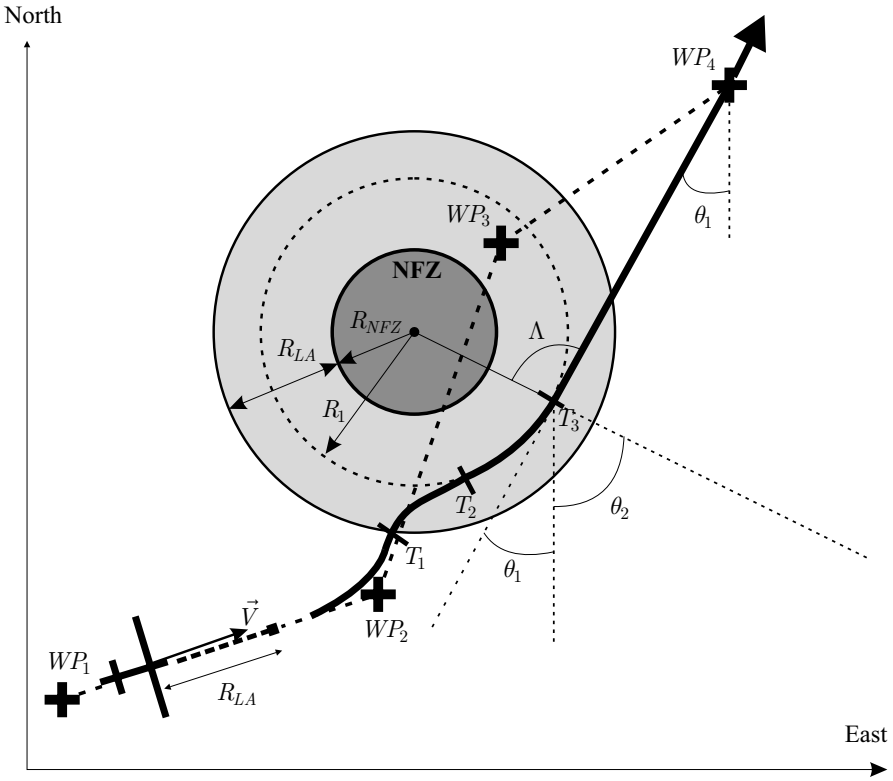


Fig. 9.14 Circular template path, waypoint tracking and reconfiguration

9.7.2 Entering the Circular Path Template

As soon as the obstacle is detected to be an immediate threat to the aircraft at point T_1, the guidance system begins an evasion maneuver with the minimum turn radius possible to reach tangentially the template path at point $T2$; see Fig. 9.14.

9.7.3 Choice of the Avoidance Side

Whether the guidance algorithm chooses to go left or right around the NFZ is determined by which side of the NFZ center the aircraft is already flying towards. If the aircraft's velocity vector is pointing to the right of the NFZ center, then the aircraft will fly around the NFZ on the right-hand side. If the velocity vector is pointing to the left-hand side, then the aircraft flies around the NFZ on the left-hand side. A circular NFZ makes this decision easy.

In practice, we compute $\Delta = \psi_{NFZ} - \psi \quad (\in [-\pi; \pi])$, and if $\Delta > 0$ then the aircraft flies on the left-hand side of the NFZ, otherwise it evades on the right-hand side; see Fig. 9.15.

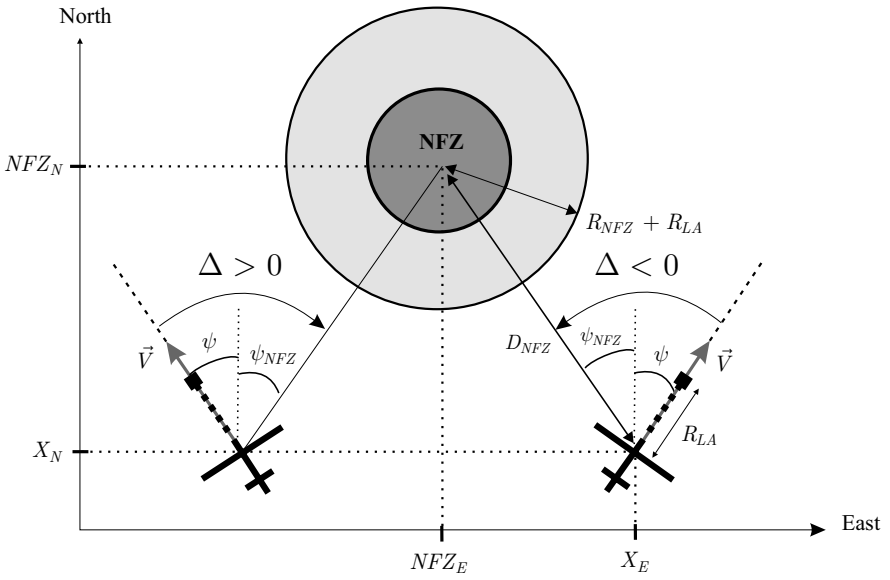


Fig. 9.15 NFZ evasion side

9.7.4 Generating the Template Path

Once the evasion maneuver is complete, the extremity of the monitoring line lies outside the NFZ. If the next waypoint to reach is obstructed by the NFZ, the guidance system guides the aircraft around the NFZ until there is a clear line of sight to the next valid waypoint. In this case, the guidance system has

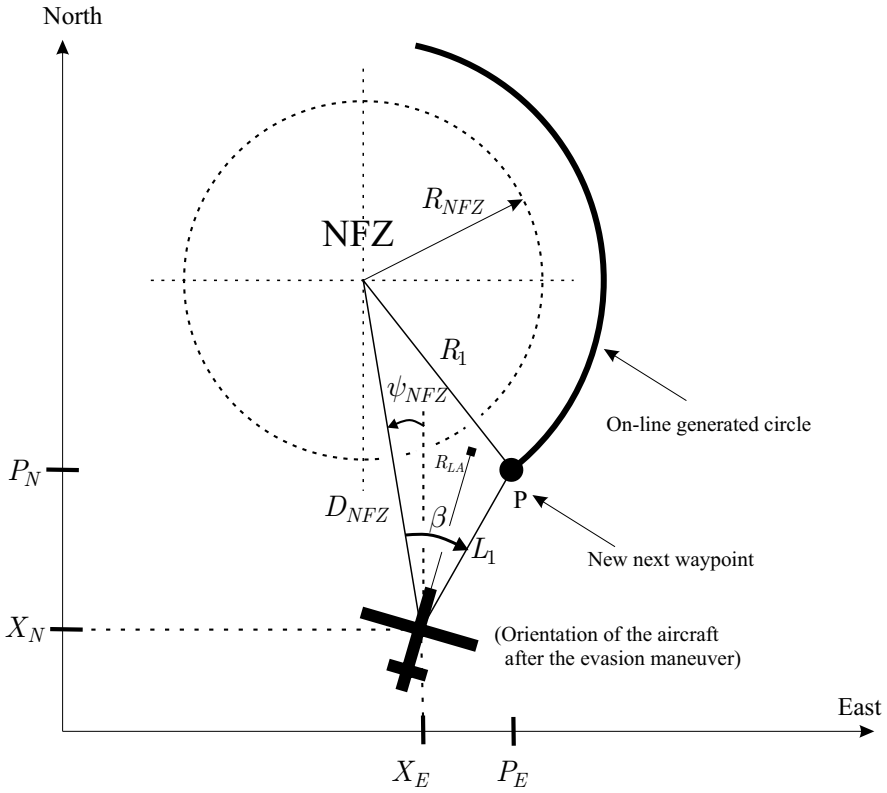


Fig. 9.16 Circular template path generation

to guide the aircraft to a point P that is on the circle R_1 and at a distance L_1 ahead of the aircraft. Using the law of cosines, the angle β (Fig. 9.16) between the segment that joins the center of the aircraft to the center of the NFZ and the segment from the center of the aircraft to the new reference point P is expressed as

$$\beta = \arccos\left(\frac{{D_{NFZ}}^2 + {L_1}^2 - {R_1}^2}{2D_{NFZ}\,L_1}\right) . \tag{9.25}$$

In order to make sure that the generated circle to avoid the NFZ is a feasible path for the aircraft, the radius R_1 is selected as

$$R_1 = \max\left(R_{min}, R_{NFZ} + \text{safety value}\right) . \tag{9.26}$$

The coordinates of the reference point P can then be computed as follows:

- If the NFZ is to be avoided on the right-hand side, we compute $\Psi_{avoid} = \psi_{NFZ} + \beta$
- If the NFZ is to be avoided on the left-hand side, we compute $\Psi_{avoid} = \psi_{NFZ} - \beta$

and finally

$$\begin{aligned} P_N &= X_N + L_1 \cos(\Psi_{avoid}) , \\ P_E &= X_E + L_1 \sin(\Psi_{avoid}) . \end{aligned} \tag{9.27}$$

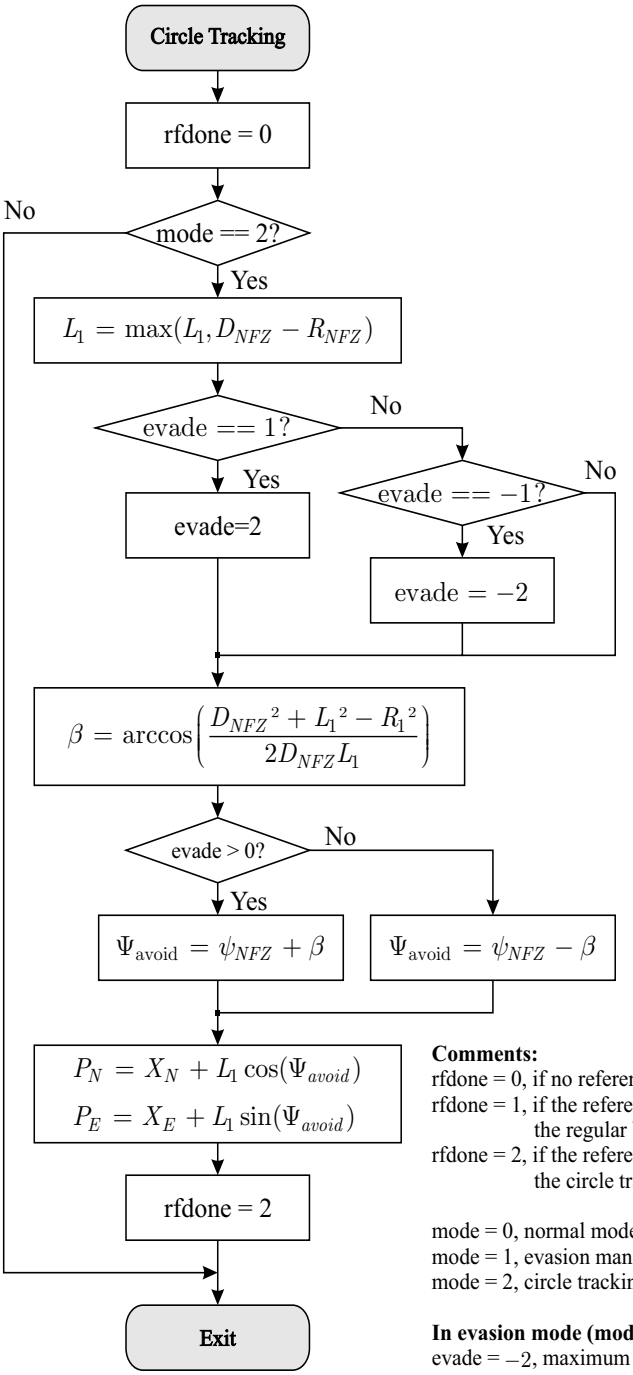
9.7.5 Leaving the Circular Path Template

The aircraft follows the circular path until it is able to continue towards the next waypoint in a straight line and without passing through the NFZ. The point at which the guidance algorithm transitions back to normal guidance (mode 0) towards the next waypoint is T_3 in Fig. 9.14, and it occurs when there is a clear line of sight from the aircraft's current position to the next waypoint.

The next waypoint has to be reachable, which means that it should lie outside a no-fly area. Therefore, while initiating the evasion maneuver, the guidance system analyzes the waypoint list and checks if the waypoint WP_{k+1} that was tracked before the evasion maneuver is still a reachable point. The following condition must hold:

$$d_{NFZ/WP_{k+1}} = \sqrt{(NFZ_N - WP_{k+1,N})^2 + (NFZ_E - WP_{k+1,E})^2} > R_1 . \tag{9.28}$$

If it turns out that this waypoint is not reachable, the guidance system selects the next waypoint in the list, and so on. Figure 9.14 illustrates the waypoint reconfiguration. The aircraft makes a left turn at waypoint WP2 in the direction of WP3. Then the NFZ is detected and the guidance system guides the aircraft around the obstacle. The unreachable waypoint WP3 is discarded and instead WP4 becomes the next target.



Comments:
rfdone = 0, if no reference WP has been chosen yet.
rfdone = 1, if the reference point has been chosen by the regular WP tracking algorithm.
rfdone = 2, if the reference point has been chosen by the circle tracking algorithm.

mode = 0, normal mode
mode = 1, evasion maneuver mode
mode = 2, circle tracking mode

In evasion mode (mode 1):
evade = −2, maximum left bank
evade = +1, maximum right bank
evade = −1, circle tracking left
evade = +2, circle tracking right

Fig. 9.17 Template path generation diagram

When should the guidance transition T_3 shown in Fig. 9.14 occur? A simple criterion is to monitor the angle Λ as shown in Fig. 9.14, which is the angle between the segment made by the aircraft's center and the next valid waypoint, and the segment made by the aircraft's center and the center of the NFZ. As soon as $|\Lambda| > \pi/2$, then there is a clear line of sight to the next valid waypoint. The angle Λ can be expressed simply as follows:

$$\Lambda = \theta_1 - \theta_2 \quad \in [-\pi; \pi] \,, \tag{9.29}$$

with

$$\begin{aligned} \theta_1 &= \tan^{-1}\left(\frac{WP_{k+1,E} - X_E}{WP_{k+1,N} - X_N}\right) \,, \\ \theta_2 &= \psi_{NFZ} = \tan^{-1}\left(\frac{NFZ_E - X_E}{NFZ_N - X_N}\right) \,. \end{aligned} \tag{9.30}$$

The decision steps for the next valid waypoint are summarized in Fig. 9.18.

9.7.6 Properties of the Guidance Schedule

The guidance schedule presented here has several desirable properties. It attempts to minimize the number of waypoints that are unreachable by initiating the avoidance maneuver as late as possible. It does not require any complex logic to decide how to avoid the NFZ. Finally, it minimizes the distance and time needed to return to the original flight path. It does this by flying directly to the next waypoint as soon as is safely possible.

9.8 Simulation

9.8.1 Simulation Set-up

Simulations were done using the nonlinear six degree-of-freedom computer model of a radio controlled aerobatic aircraft described in Chap. 3. The flight controllers used are described in Chaps. 7 and 8. The airspeed, the altitude, and the sideslip angle are kept constant.

Three similar scenarios were simulated, with the results presented below. In all scenarios, the aircraft is following a desired path that passes through an NFZ. The simulation was done with a maximum bank angle of $\phi_{max} = 30°$.

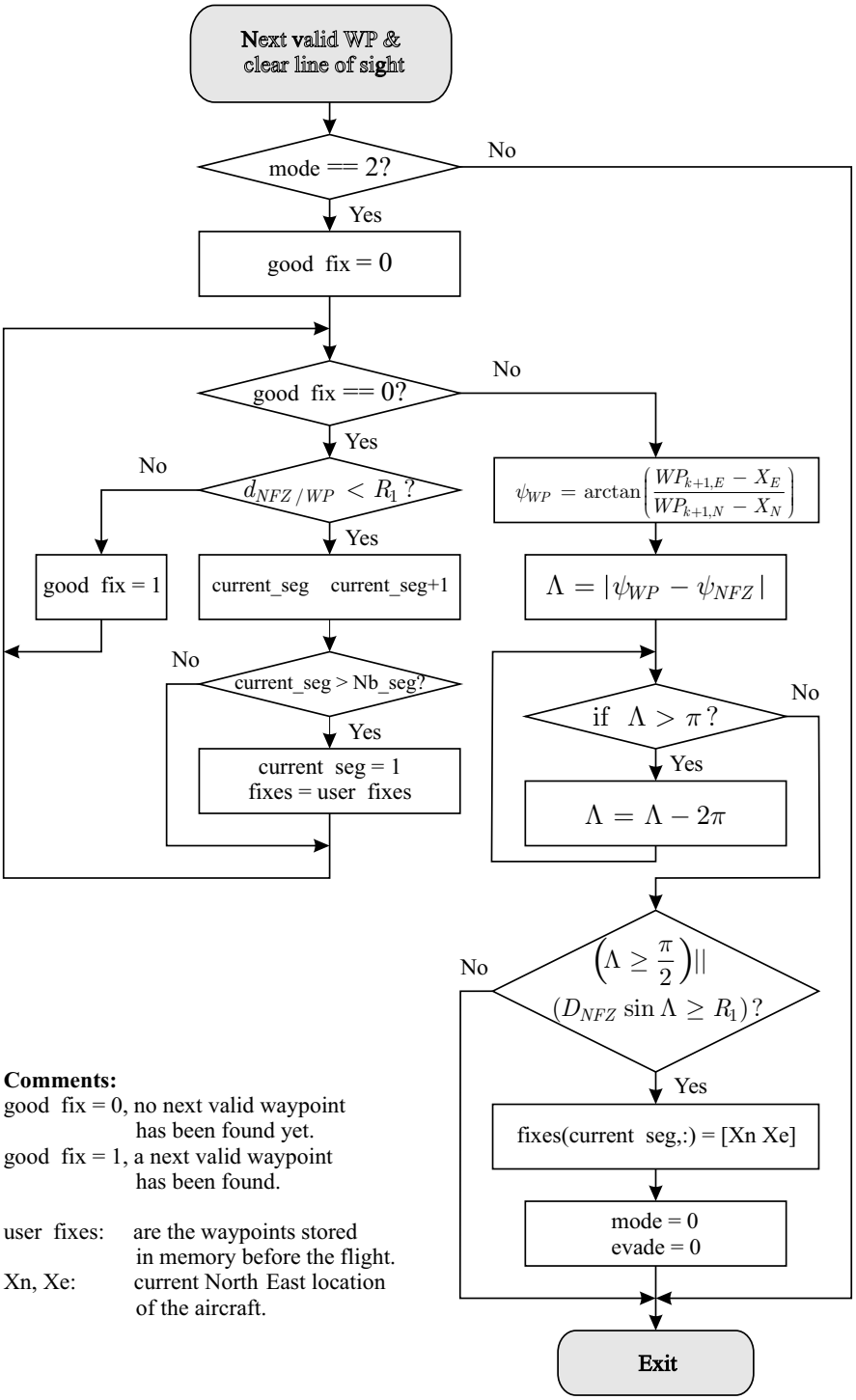


Fig. 9.18 Next valid point and clear line-of-sight determination

9.8.2 Simulation Results

9.8.2.1 No Wind

This first scenario, shown in Fig. 9.19, highlights the basic response of the aircraft to an NFZ blocking its path. The aircraft begins South of the NFZ and flies North along the desired path defined by the waypoints 1 to 5 and returns back to the runway. The desired path passes through an NFZ, but the aircraft deviates around it before returning to the desired path. The simulation was run at three different flight speeds, 15, 30, and 45 m/s. It can be seen that the aircraft begins its turn much later when flying at 15 m/s than when flying at 45 m/s. The airplane stays outside the NFZ at all three speeds.

9.8.2.2 With Wind

This second scenario, shown in Fig. 9.20, highlights the response of the aircraft in wind conditions. The desired path remains the same as in the first scenario. The path taken by the aircraft without wind and with wind are shown for comparison. The aircraft is flying at a nominal airspeed of 30 m/s.

A first flight is made with a 6 m/s crosswind blowing from West to East. In this case, the path followed by the aircraft is almost identical to the one without wind.

Another flight simulation is made with wind blowing from South to North with a speed of 6 m/s. The trajectory in the latter windy condition differs from the the nominal track (without wind) in the two turns that avoid the obstacle, where there is a maximum difference of 20 m. This is due to the fact that the wind speed adds to the airspeed resulting in a higher ground speed V_n than the one obtained with no wind. Thus, the evasion maneuver starts earlier when the aircraft first encounters the obstacle in between points 1 and 2. In between points 4 and 5, the wind is facing the aircraft, and therefore, the ground speed is lower than the one obtained with no wind. This explains why the evasion maneuver starts later.

In both cases, the NFZ is avoided. After the obstacle has been avoided, the guidance system resumes normal waypoint tracking. As expected from the lateral guidance law described in Sect. 9.2 there is no steady-state error when flying the circular arc around the NFZ as shown in Fig. 9.20.

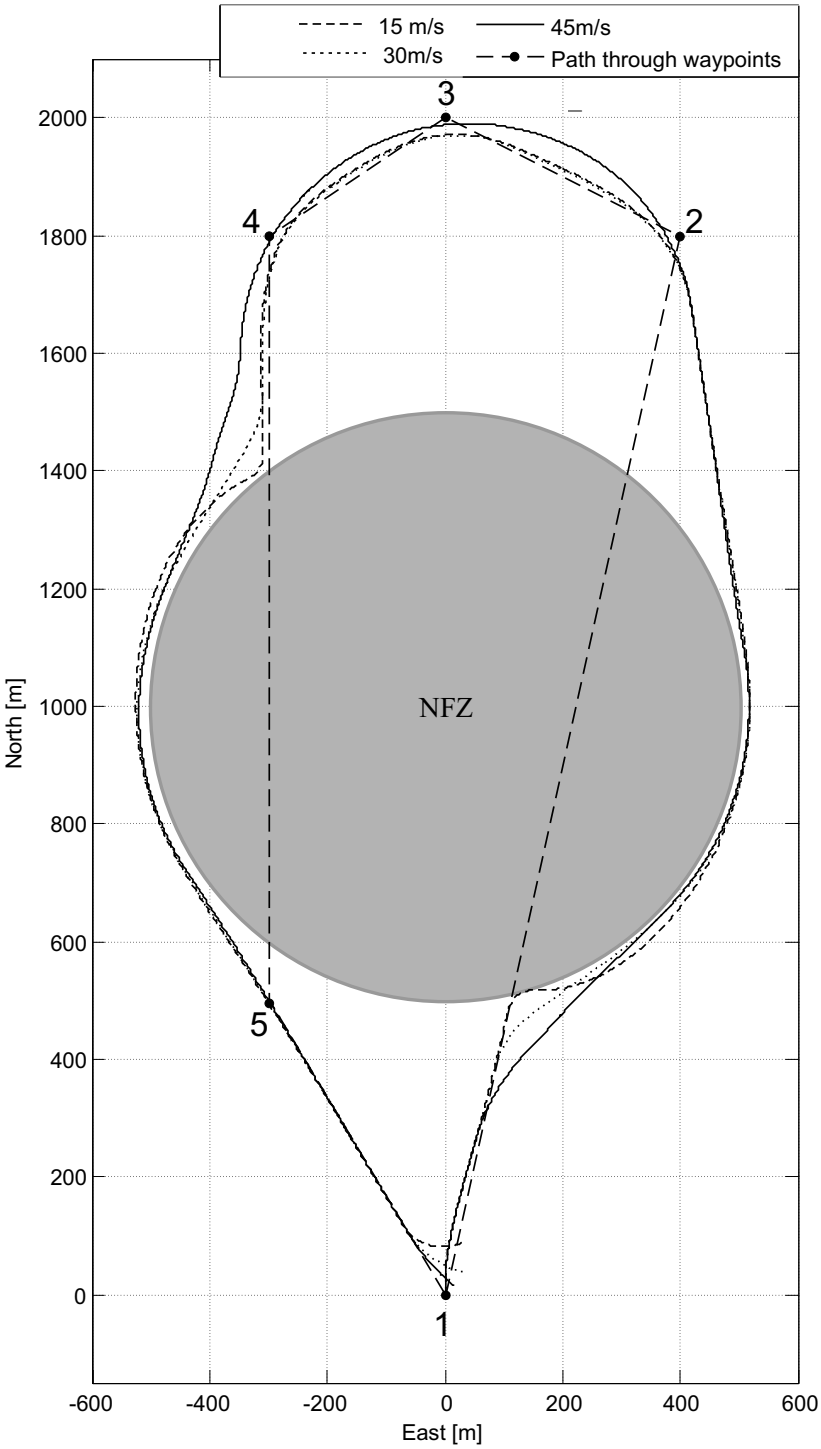


Fig. 9.19 Obstacle avoidance in no-wind condition at different speeds

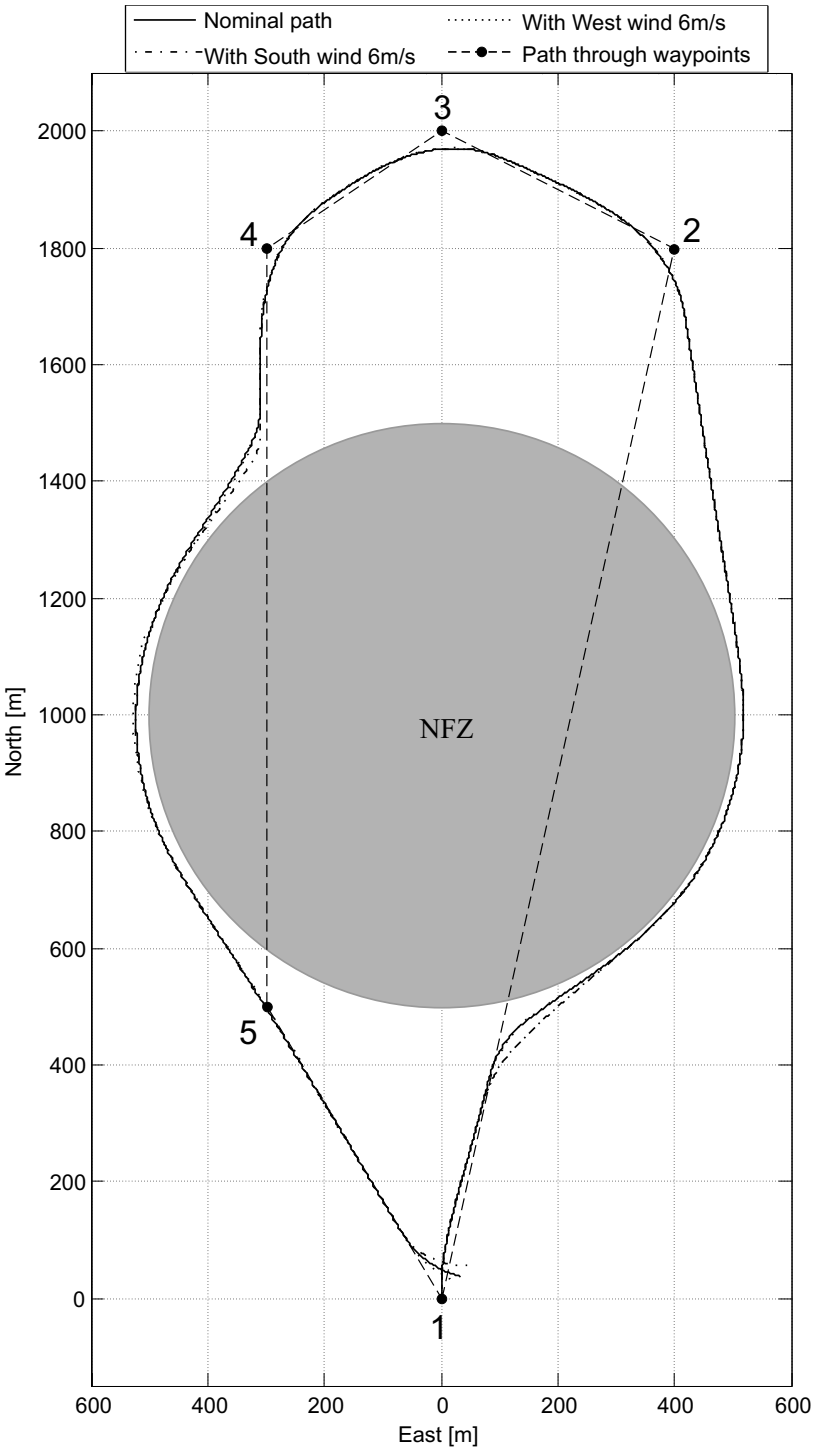


Fig. 9.20 Obstacle avoidance in wind conditions, $V_T = 30\,\mathrm{m/s}$

9.9 Conclusions

This chapter presented a guidance algorithm that combines simplicity of implementation and ability to avoid an NFZ. The algorithm successfully demonstrated in simulation its ability to guide the aircraft around the NFZ and then to resume flying along the desired path. The guidance system intrinsically takes into account the wind condition via the ground speed of the aircraft. Finally, the method is computationally efficient.

References

1. G. Ducard and H. P. Geering. A Computationally Efficient Guidance System for a Small UAV. In *Proceedings of the 4th International Conference on Informatics in Control, Automation and Robotics*, Angers, France, May 2007.
2. G. Ducard, K. C. Kulling, and H. P. Geering. A Simple and Adaptive On-Line Path Planning System for a UAV. In *Proceedings of the IEEE 15th Mediterranean Conference on Control and Automation*, pages 1–6, Athens, Greece, June 2007. T34-009.
3. L. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic Roadmaps for Path Planning in High-dimensional Configuration Spaces. *IEEE Transactions on Robotics and Automation*, 12(4):566–580, August 1996.
4. J. N. Amin, J. D. Boskovic, and R. K. Mehra. A Fast and Efficient Approach to Path Planning for Unmanned Vehicles. In *Proceedings of AIAA Guidance, Navigation, and Control Conference and Exhibit*, Keystone, CO, August 21−24 2006.
5. Y. Koren and J. Borenstein. Potential Fields Methods and their Inherent Limitations for Mobile Robot Navigation. In *Proceedings of IEEE Conference on Robotics and Automation*, Sacramento, CA, April 1991.
6. S. G. Loizou and K. J. Kyriakopoulos. Closed-loop Navigation for Multiple Holonomic Vehicles. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2861–2866, Minneapolis, Minnesota, 2002.
7. E. Rimon and D. Koditschek. Exact Robot Navigation Using Artificial Potential Functions. *IEEE Transactions on Robotics and Automation*, 8(5):501–518, October 1992.
8. Y. Kuwata, A. Richards, T. Schouwenaars, and J. P. How. Decentralized Robust Receding Horizon Control for Multi-vehicle Guidance. In *Proceedings of IEEE American Control Conference*, pages 2047–2052, Minneapolis, MN, June 2006.
9. S. Park. *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles*. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Available at http://hdl.handle.net/1721.1/16662, Cambridge, MA, 2004.
10. S. Park, J. Deyst, and J. P. How. A New Nonlinear Guidance Logic for Trajectory Tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, 2004.

Chapter 10
Evaluation of the Reduction in the Performance of a UAV

After an actuator failure, the performance of the aircraft is degraded. If an FDI system is available in the flight control system, the knowledge of the failure can be used to evaluate the new aircraft performance. Based thereon, a supervision system decides whether the mission can still be continued or if it should be aborted and have the aircraft redirected to the base station. In both cases, the aircraft should still be guided along a trajectory that is compatible with the new flying properties of the airplane. This chapter focuses on an aileron failure and shows how the degraded flying performance can be evaluated and used to reconfigure the guidance system. Simulation results show that, if the reduced performance due to the actuator failure is taken into account, the safety of the mission is improved [1].

10.1 Introduction

In this chapter, results from an on-line FDI system are used to estimate the reduction in performance of a UAV after an actuator failure. It is assumed that an FDI system is capable of estimating or measuring the position of the failed control surface. Based thereon, two simple criteria are developed in order to quantify the reduction of the flight performance of the aircraft. These two criteria are ϕ_{max}, the maximum bank angle allowed, and τ_{roll}, the time the aircraft needs to roll to ϕ_{max}. A guidance algorithm has been proposed in the previous chapter, which takes into account these two criteria in order to redefine on-line the flight path and to avoid an NFZ even after degraded flight performance.

This chapter focuses on an aileron that gets stuck and therefore alters the ability of the aircraft to roll. The supervision module is designed to make proper decisions on how to avoid an NFZ and on how to reach the next waypoint of the predefined trajectory. Simplicity and low computational power are among the requirements for this system, since it is intended to

operate in a small UAV with limited processing power. Finally, simulation results show that the system presented increases flight safety and enables the aircraft to complete its mission [1].

10.2 FDI System

The FDI system is designed to monitor the proper functioning of actuators. The FDI system detects the occurrence of an actuator fault, isolates the failed actuator, and finally estimates or measures the deflection of the failed control surface.

10.2.1 FDI with Control Surface Deflection Sensor

One simple method to detect that an actuator has failed consists of adding a sensor on the control surface to measure its deflection. If the result of this measurement is in accordance with the control signal issued by the controller, then the actuator is assumed to work properly, otherwise the actuator is considered to be faulty. If the monitored actuator turns out to have failed, its corresponding sensor will measure the deflection of the control surface, as shown by the dashed line in Fig. 10.1.

10.2.2 FDI Without Control Surface Deflection Sensor

If no sensor is used to measure the control surface deflection, then the FDI system has to be designed to detect which actuator has failed and to estimate at which deflection the control surface is stuck or floating. Such an FDI system is detailed in Chap. 4. It is based on the EMMAE method.

10.3 Degraded Turn Performance Evaluation

As Fig. 10.1 shows, the FDI system indicates to a supervision module which actuator has failed and at which deflection. The task of the supervision module is to determine the new flying performance of the aircraft by computing two parameters. They are the maximum bank angle ϕ_{max} that the guidance algorithm is allowed to command, and τ_{roll}, the time needed for the aircraft to bank to ϕ_{max}.

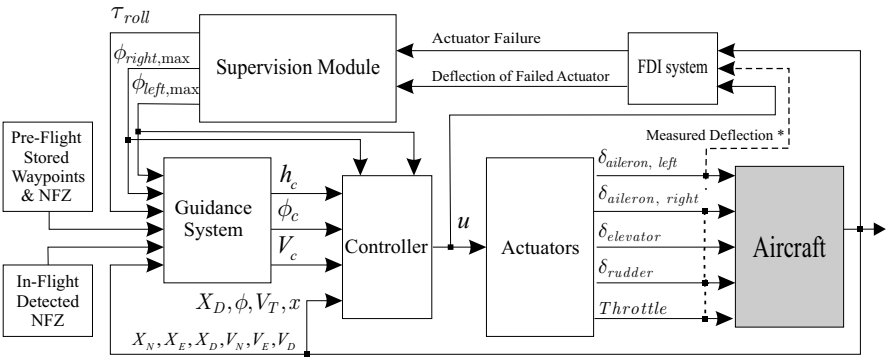


Fig. 10.1 Reconfigurable guidance and control system (* "Measured Deflection" signal is not necessary with the EMMAE-FDI system)

In normal operation, when the aircraft is flying turns, the flight controller makes sure that the roll angle does not exceed a maximum value $\phi_{max,\,nom}$ ($\pm 30°$ in our case).

Let us consider the case where the left aileron gets stuck as shown in Fig. 10.2. This causes the roll performance to differ between a left bank and a right bank maneuver.

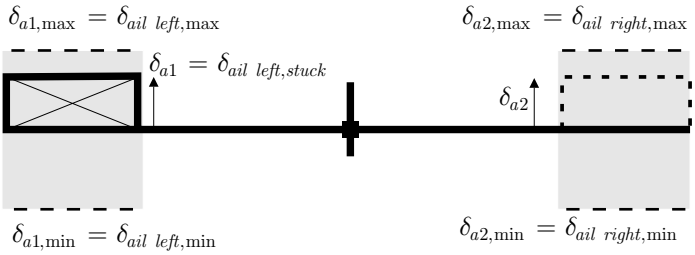


Fig. 10.2 Rear view of the aircraft with its left aileron stuck

The contribution of the ailerons for roll torque generation is expressed as follows:

$$L_{ailerons} = \bar{q} S b C_{La} (\delta_{a1} - \delta_{a2}) \ , \tag{10.1}$$

where δ_{a1} (δ_{a2}, respectively) represents the deflection of the left (right, respectively) aileron, the dynamic pressure is $\bar{q} = \rho {V_T}^2/2$, with the air density ρ, the airspeed V_T, the wing total surface S, the wing span b, and the dimensionless roll torque effectiveness of the ailerons C_{La}.

The relative deflection between the two ailerons is defined as follows:

$$\Delta\delta_{ail} = \delta_{a1} - \delta_{a2} \ . \tag{10.2}$$

During straight and level flight, in order to compensate for the failure of the left aileron, the controller has to deflect the right aileron to the same position, such that $\Delta\delta_{ail} = 0$, as is depicted in Fig. 10.2.

Therefore, if the aircraft is to bank to the right, the maximum roll torque on the right that can be generated occurs when $\delta_{a2} = \delta_{a2,\,max}$, and, conversely, the maximum roll torque on the left occurs when $\delta_{a2} = \delta_{a2,\,min}$. Let us define the relative deflection of the ailerons in the faulty case, for maximum right and left torque generation:

$$\begin{aligned} \left(\Delta\delta_{ail,\ right\ bank}\right)_{max} &= \delta_{a1,\,stuck} - \delta_{a2,\,max}\ , \\ \left(\Delta\delta_{ail,\ left\ bank}\right)_{max} &= \delta_{a1,\,stuck} - \delta_{a2,\,min}\ . \end{aligned} \tag{10.3}$$

10.3.1 Determination of the Maximum Bank Angle for Left/Right Turn

According to the scenario depicted in Fig. 10.2, the relative deflection of the ailerons is as follows:

$$\left|\left(\Delta\delta_{ail,\,right\ bank}\right)_{max}\right| < \left|\left(\Delta\delta_{ail,\,left\ bank}\right)_{max}\right|\ . \tag{10.4}$$

Consequently, it will take longer for the aircraft to reach a desired right bank angle than a left bank angle, since the maximum rate of turn $|p_{\ right\ max}| < |p_{\ left\ max}|$. After a time interval Δt, the maximum bank angles that can be reached are

$$\begin{aligned} \phi_{left} &= -\left|p_{\ left\ max}\right| \Delta t\ , \\ \phi_{right} &= \left|p_{\ right\ max}\right| \Delta t\ . \end{aligned} \tag{10.5}$$

In order to preserve the stability of the aircraft, we limit the left bank angle to a smaller value so that it takes the same time for the aircraft to come back from $\phi_{left\ max}$ as from $\phi_{right\ max}$ to $\phi = 0°$, which means that the following condition should hold

$$\frac{\phi_{left\ max}}{\left|p_{\ left\ max}\right|} = -\frac{\phi_{right\ max}}{\left|p_{\ right\ max}\right|}\ . \tag{10.6}$$

We are now interested in finding an explicit linear relationship between the roll rate p and the relative aileron deflection $\Delta\delta_{ail}$. We thus recall that the dynamic equation for the angular rates is

$$\dot{\boldsymbol{\omega}} = \left(\boldsymbol{I}^b\right)^{-1}\left([L, M, N]^T - \boldsymbol{\omega} \times \boldsymbol{I}^b\boldsymbol{\omega}\right)\ , \tag{10.7}$$

with $\boldsymbol{\omega} = [p, q, r]^T$. The variables p, q, and r are the roll, pitch, and yaw rates, and the aircraft body axes inertia matrix is $\boldsymbol{I}^b$. The variables L, M, and N stand for the roll, pitch, and yaw moments acting on the aircraft, respectively. They are the sum of the moments due to aerodynamic effects of the airframe (af) itself (fuselage and wing) and the total moment vector produced by the control actuators (δ):

$$[L, M, N]^T = [L, M, N]^T_{af} + [L, M, N]^T_{\delta} \ . \tag{10.8}$$

For our experimental small UAV, the roll moment L is modeled as follows:

$$L = \bar{q}Sb \left(C_{La}\Delta\delta_{ail} + C_{L\tilde{p}}\tilde{p} + C_{L\tilde{r}}\tilde{r} + C_{L\beta}\beta\right) \ , \tag{10.9}$$

where β is the sideslip angle of the aircraft. The inertia matrix is of the form

$$\boldsymbol{I}^b = \begin{pmatrix} I_{xx} & 0 & I_{xz} \\ 0 & I_{yy} & 0 \\ I_{xz} & 0 & I_{zz} \end{pmatrix} \ . \tag{10.10}$$

With the assumption that $I_{xz} \ll I_{xx}, I_{yy}, I_{zz}$, we can formulate a linearized dynamic equation for the roll rate as

$$\dot{p} = \frac{\rho V_T I_{zz} S b^2 C_{L\tilde{p}}}{4(I_{xx}I_{zz} - {I_{xz}}^2)} \ p + \frac{\rho {V_T}^2 I_{zz} S b C_{La}}{2(I_{xx}I_{zz} - {I_{xz}}^2)} \ \Delta\delta_{ail} \ , \tag{10.11}$$

which can also be written as

$$\dot{p} = -V_T\theta_p \ \ p + {V_T}^2\theta_{ail} \ \ \Delta\delta_{ail} \ , \tag{10.12}$$

with θ_p and θ_{ail} being positive constants:

$$\begin{aligned} \theta_p &= -\frac{\rho I_{zz} S b^2 C_{L\tilde{p}}}{4(I_{xx}I_{zz} - {I_{xz}}^2)} \ , \\ \theta_{ail} &= \frac{\rho I_{zz} S b C_{La}}{2(I_{xx}I_{zz} - {I_{xz}}^2)} \ . \end{aligned} \tag{10.13}$$

If the aircraft airspeed V_T is almost constant, we can integrate (10.12) and obtain a first-order response of the roll rate after a step input in $\Delta\delta_{ail}$:

$$p = V_T \frac{\theta_{ail}}{\theta_p} \ \Delta\delta_{ail} \ \left(1 - e^{-V_T\theta_p \ t}\right) \ . \tag{10.14}$$

We can conclude from (10.14) that the ratio $\Re_{p_{max}}$ of the maximum roll rates in each direction is proportional to the ratio of the relative deflection of the ailerons for maximum right and left torque generation as follows:

$$\frac{|p_{\,left\ max}|}{|p_{\,right\ max}|} = \frac{\left|(\Delta\delta_{ail,\,left\ bank})_{max}\right|}{\left|(\Delta\delta_{ail,\,right\ bank})_{max}\right|} = \Re_{p_{\,max}} \ . \tag{10.15}$$

We can now select the maximum bank angles that we allow in each direction by choosing

$$\begin{aligned} \phi_{right\ max} &= \phi_{max,\,nom} = +30^\circ \ , \\ \phi_{left\ max} &= -\phi_{right\ max}/\Re_{p_{max}} \ . \end{aligned} \tag{10.16}$$

10.3.2 Determination of the Minimum Radius of Right/Left Turns

By intentionally reducing the bank angle of the aircraft in one direction (in our example to the left), we are increasing the minimum radius of the left turn that the aircraft can fly according to the following expression

$$\begin{aligned} R_{left,\,min} &= \frac{{V_n}^2}{g\tan(\phi_{left\ max})} \ , \\ R_{right,\,min} &= \frac{{V_n}^2}{g\tan(\phi_{right\ max})} \ . \end{aligned} \tag{10.17}$$

10.3.3 Determination of the Maximum Roll Rates

From (10.14), we see that the steady-state value for the roll rate is $p_s = V_T \frac{\theta_{ail}}{\theta_p}\ \Delta\delta_{ail}$, which gives us a simple value for the maximum turn rate achievable in each direction:

$$\begin{aligned} p_{s\ left\,max} &= V_T \frac{\theta_{ail}}{\theta_p}\ (\Delta\delta_{ail,\,left\ bank})_{max} \ , \\ p_{s\ right\,max} &= V_T \frac{\theta_{ail}}{\theta_p}\ (\Delta\delta_{ail,\,right\ bank})_{max} \ . \end{aligned} \tag{10.18}$$

These maximum roll rates will be used to compute the maximum time to roll to ϕ_{max} in each direction, as shown in the next paragraph.

10.3.4 Determination of the Maximum Time τ_{roll} to Roll to ϕ_{max}

The variable τ_{roll} represents the maximum time needed to bank from $\phi = 0$ to the maximum left roll angle $\phi_{left\ max}$ or right roll angle $\phi_{right\ max}$

$$\tau_{roll} = \max(\tau_{roll \to \phi_{right\ max}}, \tau_{roll \to \phi_{left\ max}}) \ . \tag{10.19}$$

In order to get an explicit value of τ_{roll} for the on-line reconfiguration of the guidance system, we neglect the transients in the roll rate (see (10.14)), and we integrate (10.18) with respect to time:

$$\begin{aligned} \phi_{left\ max} &= p_{s,\ left\, max} \cdot \tau_{roll \to \phi_{left\ max}} \ , \\ \phi_{right\ max} &= p_{s,\ right\, max} \cdot \tau_{roll \to \phi_{right\ max}} \ . \end{aligned} \tag{10.20}$$

Finally, the characteristic time to roll to ϕ_{max} is computed as

$$\tau_{roll} = \max \left(\frac{\phi_{left\ max}}{p_{s,\ left\, max}} \ , \frac{\phi_{right\ max}}{p_{s,\ right\, max}} \right) \ . \tag{10.21}$$

10.4 Interface with the Guidance System

The guidance system developed in the previous chapter can dynamically reconfigure the trajectory flown by the aircraft in the case of an aileron failure. For this, the variables $\phi_{left/right,\ max}$ and τ_{roll} are to be given to the guidance system.

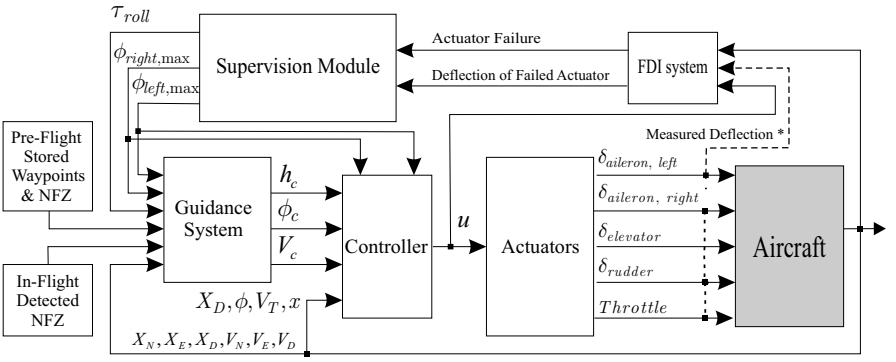


Fig. 10.3 Guidance system inputs and outputs (* "Measured Deflection" signal is not necessary with the EMMAE-FDI system)

10.5 Stability Discussion

The main goals of the algorithm presented in this chapter are to limit how far the aircraft is allowed to roll with the parameter ϕ_{max}, and to help adapt the guidance system of the aircraft, but not the low-level controllers. This adaptation is done through a linear estimation of the lateral performance of the aircraft. This adaptation helps prevent the guidance system from commanding an infeasible path that may violate an NFZ or collide with an obstacle. The guidance algorithm assumes the availability of a bank-angle controller, whose responsibility is to ensure the stability of the aircraft. In order to prevent commanded paths that would be too aggressive for the aircraft, the guidance algorithm includes logic that smooths the commanded path and avoids large steps in the bank angle control signal.

At the end of 2007, an interesting paper was written by Park *et al.* [2] on the performance and Lyapunov stability of the nonlinear path following guidance method described in [3, 4] and utilized in Chap. 9. The adaptive nature of the guidance method makes its stability independent of the vehicle velocity. Robust stability of the guidance law is shown in [2] in the presence of saturated lateral acceleration, which is a limitation for every aircraft and in particular in the context of actuator fault or failure.

10.6 Simulation Results

Simulations were done using the nonlinear six degree-of-freedom computer model of a radio controlled aerobatic aircraft described in Chap. 3. The flight controllers used are described in Chaps. 7 and 8. The NFZ is located in front of the aircraft 1000 m North, 0 m East, and has a radius of 500 m. The aircraft is commanded to fly to a desired waypoint at 1800 m North, 200 m East, at an airspeed of $V_T = 30\,\mathrm{m/s}$.

10.6.1 No Failure

This first scenario, shown in Fig. 10.4, highlights the basic response of the aircraft to an NFZ blocking its path. The aircraft begins South of the NFZ and flies North along the desired path. The desired path then continues through the NFZ, but the aircraft autonomously deviates around it to the right before returning to the desired path North and East of the NFZ.

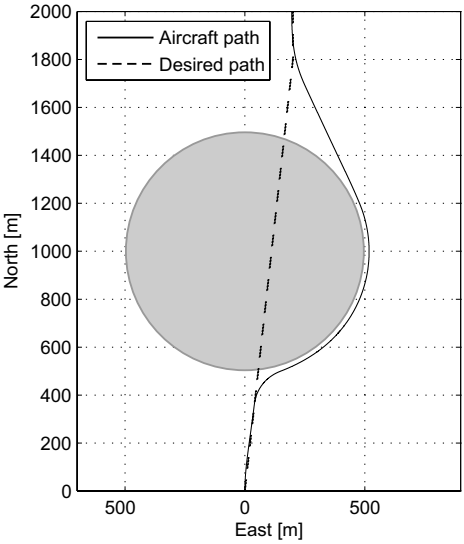


Fig. 10.4 No failure

10.6.2 With Failure but No Reconfiguration

The left aileron is now stuck at mid-motion range. In this scenario the guidance system is not reconfigured. We can see in Fig. 10.5 that the aircraft initiates its evasion procedure at the same time as in the no-failure case. However, due to the degraded performance, the aircraft is not able to turn right as fast as before. Therefore, the aircraft penetrates the NFZ or collides with the obstacle.

10.6.3 With Failure and With Reconfiguration

The left aileron is still stuck at mid-motion range. Now, the guidance system is reconfigured based on the knowledge of the fault and the new parameters for $\phi_{max,left/right}$ and τ_{roll}. As a result, the look-ahead distance R_{La} is longer, causing the aircraft to start its evasion maneuver earlier than in the previous cases. Also, the turn's radius is larger, which makes the aircraft circumvent the obstacle from a larger distance without penetrating it.

Although in this chapter the case where the left aileron fails was studied, the same methodology applies if it is the right aileron that fails.

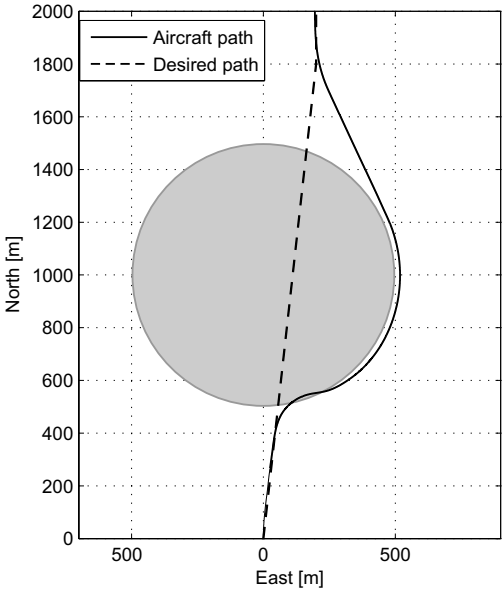


Fig. 10.5 With failure but no reconfiguration

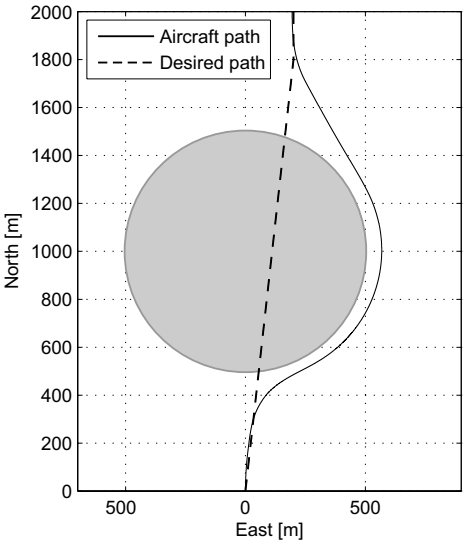


Fig. 10.6 With failure and with reconfiguration

10.7 Performance Degradation Around the Pitch and Yaw Axes

10.7.1 Pitch Axis

The ability of the aircraft to rotate around its pitch axis is essentially due to the elevators. For small private aircraft like a propeller Cessna, there is only

one elevator on the tail fin for pitch rate control. If this actuator unfortunately fails or breaks there is no other means to control the aircraft around the pitch axis. If the deflection of the failed elevator is luckily close to the trim position, then the altitude can still be maintained and adjusted by appropriate throttle commands. However, any landing attempt without a working elevator would be a highly risky exercise.

The small UAV that we consider in this book has two independent horizontal control surfaces on the tail fin as shown in Fig. 3.1. They are controlled by the control allocation module described in Chap. 5, which explains how control commands are reallocated among actuators in order to efficiently compensate for the failure of one of the two elevators.

Therefore, in this chapter, no special guidance reconfiguration is developed in the event of an elevator failure. It is enough for the safety of the mission if the altitude at least can be maintained constant despite the failure. This is guaranteed as long as the elevator does not lock at its maximum deflection. An elevator failure at a large deflection will generate significant drag, all the more that the failure is compensated by having the other elevator deflect the same but opposite deflection. This may generate significant drag, thus slowing down the aircraft, which will loose altitude unless the airspeed is maintained. This is the responsibility of the airspeed controller.

10.7.2 Yaw Axis

If the rudder fails and, for example, locks at a non-zero deflection angle, then some inevitable sideslip angle will build up. In order to maintain the flight path along the desired trajectory, the guidance system generates a bank angle command ϕ_c that counteracts the lateral undesired acceleration caused by the failed rudder. A rudder failure is less critical than an elevator failure and can usually be compensated by a combined deflection of ailerons and elevators. This is automatically done by the flight control system.

10.8 Conclusion

This chapter presents a simple method to evaluate the performance of a UAV after an aileron failure. Since the method is intended to be implemented on a small UAV control unit with limited processing power, simplicity is required. The key to simplicity is to interpret any reduction in the aircraft performance about the roll axis as a limitation on the bank angle, $\phi_{max,left/right}$, and as longer time to reach the maximum bank angle, τ_{roll}. These two criteria are passed to a guidance algorithm that is simple and adaptable to the aircraft's performance reduction described by the two criteria mentioned. The

algorithm demonstrated in simulation its ability to autonomously guide the aircraft around the NFZ and then resume flying along the desired path. In the event of a failed actuator, simulations have shown the benefits of reconfiguring the guidance system to take into account the degraded flight performance of the airplane in order to redefine on-line a feasible flight path that avoids the obstacle.

References

1. G. Ducard, K. C. Kulling, and H. P. Geering. Evaluation of Reduction in the Performance of a Small UAV after an Aileron Failure for an Adaptive Guidance System. In *Proceedings of the IEEE American Control Conference*, pages 1793–1798, New York, NY, July 2007.
2. S. Park, J. Deyst, and J. P. How. Performance and Lyapunov Stability of a Nonlinear Path-Following Guidance Method. *AIAA Journal of Guidance, Control, and Dynamics*, 30(6):1718–1728, November-December 2007.
3. S. Park. *Avionics and Control System Development for Mid-Air Rendezvous of Two Unmanned Aerial Vehicles*. PhD thesis, Department of Aeronautics and Astronautics, Massachusetts Institute of Technology, Available at http://hdl.handle.net/1721.1/16662, Cambridge, MA, 2004.
4. S. Park, J. Deyst, and J. P. How. A New Nonlinear Guidance Logic for Trajectory Tracking. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Providence, RI, 2004.

Chapter 11
Conclusions and Outlook

11.1 Future Work

11.1.1 FDI System

The FDI system developed in this book runs n (number of actuators) $+ 1$ EKFs in parallel. Current and future research work of the author focuses on a technique to reduce the computational complexity needed to achieve fault diagnoses with at least the same level of reliability and performance.

11.1.2 Reconfigurable Guidance System

The reconfigurable guidance system presented in Chaps. 9 and 10 adapts the trajectory of the aircraft based on modifying the minimum radius of turn R_{min} that the aircraft can fly. In both chapters, the airspeed was regulated to be constant around $V_T = 30\,\mathrm{m/s}$. However, the guidance system can be reconfigured by also adapting the airspeed. Indeed, if the speed is lowered, the aircraft can make a coordinated turn with a smaller turn radius, which should help avoid more easily some obstacles or threats.

After an actuator fault or damage, the flying performance of the aircraft inevitably degrades. This performance reduction requires more energy to achieve the mission. Indeed, since the trajectories are reconfigured, the length of the new flight path may be significantly extended. Therefore, the post-fault mission is also to be redesigned based on the amount of energy that is still available aboard the UAV. This is a topic for future research.

Chapter 9 presented an efficient method to avoid a known NFZ or a pop-up obstacle. The system was demonstrated in the event of a single NFZ blocking the aircraft's path from one waypoint to the next. The method can be extended to cope with multiple NFZs or obstacles.

11.2 The Future of Fault-tolerant Flight Control Systems for UAVs

Many fault-tolerant control systems have already been developed for UAVs, and a few have been successfully demonstrated during real flight tests. In the near future, more fault-tolerant flight control and guidance algorithms will be tested in real UAVs. A reconfigurable flight system will be successful in practice for UAV applications, if it combines reliability, reasonable complexity, and computation efficiency. This is even more true if a fault-tolerant flight control system is to be designed for a formation of UAVs, which is a new and appealing research area.

11.3 General Conclusion

In this book, new approaches for fault-tolerant flight control systems are investigated. The complete fault-tolerant guidance and control system is shown in Fig. 11.1.

The core of these fault-tolerant systems is the EMMAE-FDI algorithm. An efficient and nonlinear FDI system is developed for actuators and sensors over the entire operating envelope of an aircraft. Locked-in-place and floating actuator faults can be handled. The EMMAE-FDI system is also capable of handling two simultaneous actuator failures with no increase of the computational load and with no additional actuator position sensor.

The robustness of the FDI system is enhanced by the use of auxiliary excitation signals in order to check systematically any suspicious behavior of the aircraft. The robustness of the EMMAE-FDI system has been successfully demonstrated in simulation with a nonlinear model of a model aircraft flying a realistic flight scenario under severe wind conditions. The EMMAE-FDI algorithm, combined with an active supervision module, offers fast and ac-

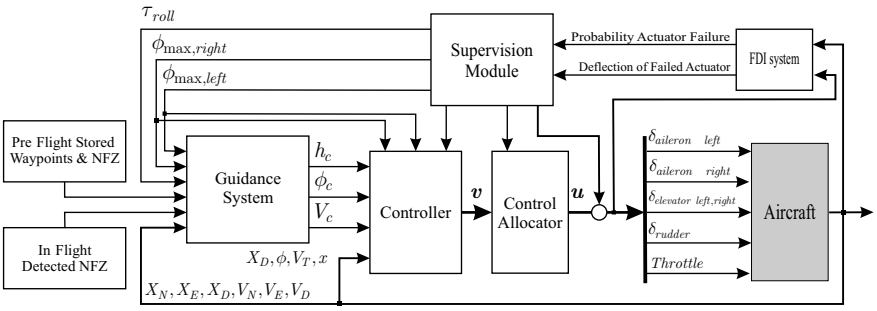


Fig. 11.1 Complete fault-tolerant guidance and control system

curate FDI. When a fault is clearly isolated, the faulty actuator deflection that is estimated by the FDI module can be used advantageously to modify on-line the settings of a control allocator.

The control allocation module presented in this book uses explicit laws for fast operation and low computational load, such that this algorithm can run in a small processor or microcontroller with limited floating-point operation capability. The control allocation method is designed to be reconfigurable based on the results obtained from the EMMAE-FDI system. Depending on the fault configuration, a supervision module has to select the appropriate law to distribute the commands and compensate for actuator faults. There is no need to redesign the flight controllers when such faults occur, since the control allocator compensates for the fault.

Simulation results show that this method yields satisfactory results, provides optimal solutions in some cases, and is much simpler and faster than conventional methods based on optimization algorithms. Indeed, no optimization solver has to be run on-line, and no pseudoinverse matrix has to be computed. This method also requires very few floating-point operations and is deterministic. The technique is therefore suitable to be implemented in a small processor or a microcontroller, where the computational power is rather limited and where the program has to run in a real-time environment.

In the flight control system described in this book, robust nonlinear controllers suitable for a wide range of operating conditions have been designed, thus removing the need for designing several linear controllers and gain-scheduling them. The controllers are based on a combination of explicit model following, NDI, and nonlinear transformations of selected longitudinal state variables. This book explains a practical approach for stability and robustness analyses of the plant undergoing the dynamic inversion process and provides a systematic procedure for the selection of uncertain model parameters involved in the controllers.

As part of the reconfigurable flight control system, a novel approach for adaptive guidance is presented. It combines a nonlinear lateral guidance control law, originally designed for UAVs tracking circles for mid-air rendezvous, with a new simple adaptive path-planning algorithm. Preflight path planning consists only of storing a few waypoints guiding the aircraft to its targets. Moreover, this guidance system integrates an efficient way to model NFZ, to generate a path in real time to avoid known or "pop-up" obstacles, and to reconfigure the flight path in the event of reduced aircraft performance. Simulation results show the good performance of this reconfigurable guidance system, which is also computationally efficient.

Since after an actuator failure the performance of the aircraft inevitably degrades, the knowledge of the failure is used to evaluate the new aircraft performance. Based thereon, a supervision system decides whether the mission can still be continued or if it should be aborted, in which case the aircraft is redirected to the base station. In both cases, the aircraft should still be guided along a trajectory that is compatible with the new flying properties

of the airplane. An example dealing with an aileron failure shows how the degraded flying performance can be evaluated and used to reconfigure the guidance system. Simulation results show that, if the reduced performance due to the actuator failure is taken into account, the safety of the mission is considerably improved.

Appendix A
V_T, α, and β Differential Equations

Chapter 3 establishes the equations for the aircraft motion. Equation 3.13 is recalled below for convenience

$$\sum_j \boldsymbol{F_j} = \left[\frac{d}{dt}\Big(m\boldsymbol{v}\Big)\right]_n , \tag{A.1}$$

$$= \left[\frac{d(m\boldsymbol{v})}{dt}\right]_b + \boldsymbol{\omega_{b/n}} \times (m\boldsymbol{v}) . \tag{A.2}$$

Equation A.1 corresponds to the law of Newton, where the subscript n denotes the navigation frame. In (A.2), the time derivation is computed in the aircraft body-fixed frame (b). Time derivation can also be done in the wind frame (w) as follows:

$$\frac{1}{m}\sum_j \boldsymbol{F_j} = \left[\frac{d\boldsymbol{v}}{dt}\right]_b + \boldsymbol{\omega_{b/n}} \times \boldsymbol{v} = \left[\frac{d\boldsymbol{v}}{dt}\right]_w + \boldsymbol{\omega_{w/b}} \times \boldsymbol{v} + \boldsymbol{\omega_{b/n}} \times \boldsymbol{v} . \tag{A.3}$$

All of the terms in (A.3) are projected in the wind frame, yielding

$$\frac{1}{m}\sum_j \boldsymbol{F_j}^w = \left[\frac{d\boldsymbol{v^w}}{dt}\right]_w + \boldsymbol{\omega_{w/b}}^w \times \boldsymbol{v}^w + \boldsymbol{\omega_{b/n}}^w \times \boldsymbol{v}^w . \tag{A.4}$$

The forces due to the engine, aerodynamics, and gravity are inserted in (A.4) as follows:

$$\frac{1}{m}\left(m\boldsymbol{g}^w + \boldsymbol{F_{engine}}^w + \boldsymbol{F_{aero}}^w\right) - \boldsymbol{\omega_{w/b}}^w \times \boldsymbol{v}^w - \boldsymbol{\omega_{b/n}}^w \times \boldsymbol{v}^w = \left[\frac{d\boldsymbol{v^w}}{dt}\right]_w , \tag{A.5}$$

with

$$\begin{aligned}\boldsymbol{g}^w &= \boldsymbol{C_b^w C_n^b} \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} = \begin{bmatrix} g_x^w \\ g_y^w \\ g_z^w \end{bmatrix}, \\ &= g \begin{bmatrix} \sin\alpha\cos\beta\cos\theta\cos\phi + \sin\beta\cos\theta\sin\phi - \cos\alpha\cos\beta\sin\theta \\ \cos\alpha\sin\beta\sin\theta + \cos\beta\cos\theta\sin\phi - \sin\alpha\sin\beta\cos\theta\cos\phi \\ \sin\alpha\sin\theta + \cos\alpha\cos\theta\cos\phi \end{bmatrix}, \end{aligned} \tag{A.6}$$

and

$$\boldsymbol{F_{engine}}^w = \boldsymbol{C_b^w} \begin{bmatrix} F_T \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{F_{aero}}^w = \begin{bmatrix} X^w \\ Y^w \\ Z^w \end{bmatrix}, \quad \boldsymbol{v}^w = \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix}, \tag{A.7}$$

$$\boldsymbol{\omega_{w/b}}^w = \begin{bmatrix} -\dot{\alpha}\sin\beta \\ -\dot{\alpha}\cos\beta \\ \dot{\beta} \end{bmatrix}, \quad \boldsymbol{\omega_{b/n}}^w = \boldsymbol{C_b^w} \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} p^w \\ q^w \\ r^w \end{bmatrix}. \tag{A.8}$$

Equation A.5 is computed using the terms in (A.6)−(A.8) as follows:

$$\begin{aligned}\begin{bmatrix} g_x^w \\ g_y^w \\ g_z^w \end{bmatrix} + \frac{1}{m} \left(F_T \begin{bmatrix} \cos\alpha\cos\beta \\ -\sin\beta\cos\alpha \\ -\sin\alpha \end{bmatrix} + \begin{bmatrix} X^w \\ Y^w \\ Z^w \end{bmatrix} \right) - \begin{bmatrix} p^w \\ q^w \\ r^w \end{bmatrix} \times \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} \\ - \begin{bmatrix} -\dot{\alpha}\sin\beta \\ -\dot{\alpha}\cos\beta \\ \dot{\beta} \end{bmatrix} \times \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \dot{V}_T \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{A.9}$$

Computing the two cross-products of (A.9) yields

$$\begin{aligned}\begin{bmatrix} g_x^w \\ g_y^w \\ g_z^w \end{bmatrix} + \frac{1}{m} \left(F_T \begin{bmatrix} \cos\alpha\cos\beta \\ -\sin\beta\cos\alpha \\ -\sin\alpha \end{bmatrix} + \begin{bmatrix} X^w \\ Y^w \\ Z^w \end{bmatrix} \right) \\ - \begin{bmatrix} 0 \\ r^w V_T \\ -q^w V_T \end{bmatrix} - \begin{bmatrix} 0 \\ \dot{\beta} V_T \\ \dot{\alpha}\cos\beta V_T \end{bmatrix} = \begin{bmatrix} \dot{V}_T \\ 0 \\ 0 \end{bmatrix}. \end{aligned} \tag{A.10}$$

Finally, the nonlinear differential equations for the airspeed V_T, the sideslip angle β, and the angle of attack α are obtained as follows:

$$\begin{aligned}\dot{V}_T &= \frac{1}{m} \left(X^w + F_T\cos\alpha\cos\beta + mg_x^w \right), \\ \dot{\beta} &= -r^w + \frac{1}{mV_T} \left(Y^w - F_T\cos\alpha\sin\beta + mg_y^w \right), \\ \dot{\alpha} &= \frac{1}{\cos\beta} \left(q^w + \frac{1}{mV_T} (Z^w - F_T\sin\alpha + mg_z^w) \right). \end{aligned} \tag{A.11}$$

Appendix B
Discretization of Linear State Space Models

B.1 Continuous Model

Consider the following continuous state space model:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{A}\boldsymbol{x}(t) + \boldsymbol{B}\boldsymbol{u}(t) + \boldsymbol{w}(t) \ , \tag{B.1}$$

where $\boldsymbol{A}$ and $\boldsymbol{B}$ are the dynamics and input matrices, and where $\boldsymbol{w}(t)$ represents continuous process noise with a covariance matrix $\boldsymbol{Q} = E\{\boldsymbol{w}\boldsymbol{w}^T\}$. The time derivative of the exponential matrix is

$$\frac{d(e^{\boldsymbol{A}t})}{dt} = \boldsymbol{A}e^{\boldsymbol{A}t} = e^{\boldsymbol{A}t}\boldsymbol{A} \ . \tag{B.2}$$

Multiplying (B.1) on the left-hand side by the term $e^{-\boldsymbol{A}t}$ yields

$$\begin{aligned} & e^{-\boldsymbol{A}t}\dot{\boldsymbol{x}}(t) = e^{-\boldsymbol{A}t}\boldsymbol{A}x(t) + e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t) + e^{-\boldsymbol{A}t}\boldsymbol{w}(t) \ , \\ \Longleftrightarrow \ & e^{-\boldsymbol{A}t}\dot{\boldsymbol{x}}(t) - e^{-\boldsymbol{A}t}\boldsymbol{A}\boldsymbol{x}(t) = e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t) + e^{-\boldsymbol{A}t}\boldsymbol{w}(t) \ . \end{aligned} \tag{B.3}$$

The left-hand side of (B.3) is equal to $\frac{d(e^{-\boldsymbol{A}t}\boldsymbol{x}(t))}{dt}$, therefore (B.3) is rewritten as

$$\frac{d(e^{-\boldsymbol{A}t}\boldsymbol{x}(t))}{dt} = e^{-\boldsymbol{A}t}\boldsymbol{B}\boldsymbol{u}(t) + e^{-\boldsymbol{A}t}\boldsymbol{w}(t) \ . \tag{B.4}$$

Integrating the above equation on the time interval $[0, t]$ yields

$$\begin{aligned}
\int_0^t \frac{d(e^{-\boldsymbol{A}\tau}\boldsymbol{x}(\tau))}{d\tau}d\tau &= \int_0^t e^{-\boldsymbol{A}\sigma}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_0^t e^{-\boldsymbol{A}\nu}\boldsymbol{w}(\nu)d\nu \ , \\
e^{-\boldsymbol{A}t}\boldsymbol{x}(t) - \boldsymbol{x}(0) &= \int_0^t e^{-\boldsymbol{A}\sigma}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_0^t e^{-\boldsymbol{A}\nu}w(\nu)d\nu \ , \\
e^{-\boldsymbol{A}t}\boldsymbol{x}(t) &= \boldsymbol{x}(0) + \int_0^t e^{-\boldsymbol{A}\sigma}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_0^t e^{-\boldsymbol{A}\nu}\boldsymbol{w}(\nu)d\nu \ , \\
\boldsymbol{x}(t) &= e^{\boldsymbol{A}t}\boldsymbol{x}(0) + \int_0^t e^{\boldsymbol{A}(t-\sigma)}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_0^t e^{\boldsymbol{A}(t-\nu)}\boldsymbol{w}(\nu)d\nu \ .
\end{aligned} \tag{B.5}$$

The continuous transition matrix is defined as follows:

$$\boldsymbol{\Phi}(t) = e^{\boldsymbol{A}t} \tag{B.6}$$

and finally the continuous solution for $\boldsymbol{x}(t)$ is written as

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t)\boldsymbol{x}(0) + \int_0^t \boldsymbol{\Phi}(t-\sigma)\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_0^t \boldsymbol{\Phi}(t-\nu)\boldsymbol{w}(\nu)d\nu \ . \tag{B.7}$$

B.2 Discrete Model

The discrete fundamental or transition matrix can be found by evaluating the continuous transition matrix at the sampling time T_s or $\boldsymbol{\Phi}_{\boldsymbol{k}} = \boldsymbol{\Phi}(T_s)$. As demonstrated in several practical examples in the book by Zarchan and Musoff[1], the transition matrix can be approximated by taking only the first two elements of the Taylor series. Adding higher-order terms does not bring significant improvements in the performance of the filter. Therefore, the discrete transition matrix is computed as follows:

$$\boldsymbol{\Phi}_{\boldsymbol{k}} \approx \boldsymbol{I} + \boldsymbol{A}T_s \ , \tag{B.8}$$

where the system dynamics matrix $\boldsymbol{A}$ can be obtained by linearizing the nonlinear equations and is successively (after every sampling time step T_s) evaluated at the latest available state estimate $\hat{\boldsymbol{x}}(k|k)$ of the EKF according to

$$\boldsymbol{A} = \left.\frac{\partial \boldsymbol{f}(\boldsymbol{x})}{\partial \boldsymbol{x}}\right|_{\boldsymbol{x}=\hat{\boldsymbol{x}}(k|k)} \ . \tag{B.9}$$

Let us define the discrete value taken by the state vector after the k^{th} time sample with $\boldsymbol{x}[k] = \boldsymbol{x}(kT_s)$.

[1] P. Zarchan and H. Musoff. *Fundamentals of Kalman Filtering: A Practical Approach, Second Edition.* Volume 208, Progress in Astronautics and Aeronautics, AIAA Inc., Reston, VA, 2005.

We can therefore rewrite the solution for $\boldsymbol{x}(t)$ obtained in (B.5) at $t = (k+1)T_s$ as

$$\begin{aligned}\boldsymbol{x}[k+1] &= e^{\boldsymbol{A}(k+1)T_s}\boldsymbol{x}(0) + \int_0^{(k+1)T_s} e^{\boldsymbol{A}((k+1)T_s-\sigma)}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma \\ &\qquad + \int_0^{(k+1)T_s} e^{\boldsymbol{A}((k+1)T_s-\sigma)}\boldsymbol{w}(\sigma)d\sigma \ , \\ &= \int_{kTs}^{(k+1)T_s} e^{\boldsymbol{A}((k+1)T_s-\sigma)}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_{kTs}^{(k+1)T_s} e^{\boldsymbol{A}((k+1)T_s-\sigma)}\boldsymbol{w}(\sigma)d\sigma \\ &+ e^{\boldsymbol{A}T_s}\left[e^{\boldsymbol{A}kT_s}\boldsymbol{x}(0) + \int_0^{kT_s} e^{\boldsymbol{A}(kT_s-\sigma)}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_0^{kT_s} e^{\boldsymbol{A}(kT_s-\sigma)}\boldsymbol{w}(\sigma)d\sigma\right] .\end{aligned} \tag{B.10}$$

The bracketed term in (B.10) is actually $\boldsymbol{x}[k] = \boldsymbol{x}(kT_s)$, and the second and third terms can be simplified by using the substitution $\tau = (k+1)T_s - \sigma$ as follows:

$$\begin{aligned}\int_{kTs}^{(k+1)T_s} e^{\boldsymbol{A}((k+1)T_s-\sigma)}\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma &= -\int_{T_s}^{0} e^{\boldsymbol{A}\tau}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau \ , \\ &= \int_0^{T_s} e^{\boldsymbol{A}\tau}\boldsymbol{B}\boldsymbol{u}(\tau)d\tau \ .\end{aligned} \tag{B.11}$$

Assuming also that the input $\boldsymbol{u}$ is constant over the integration interval (zero-order hold) yields

$$\begin{aligned}\boldsymbol{x}[k+1] &= e^{\boldsymbol{A}T_s}\boldsymbol{x}[k] + \left(\int_0^{T_s} e^{\boldsymbol{A}\tau}d\tau\right)\boldsymbol{B}\boldsymbol{u}[k] + \left(\int_0^{T_s} e^{\boldsymbol{A}\tau}\boldsymbol{w}(\tau)d\tau\right) \ , \\ &= \boldsymbol{\Phi_k}\boldsymbol{x}[k] + \left(\int_0^{Ts} \boldsymbol{\Phi}(\tau)d\tau\right)\boldsymbol{B}\boldsymbol{u}[k] + \left(\int_0^{Ts} \boldsymbol{\Phi}(\tau)\boldsymbol{w}(\tau)d\tau\right) \ , \\ &= \boldsymbol{\Phi_k}\boldsymbol{x}[k] + \boldsymbol{B_k}\boldsymbol{u}[k] + \boldsymbol{w_k} \ .\end{aligned} \tag{B.12}$$

B.2.1 Derivation of the Discrete Process-noise Covariance Matrix $\boldsymbol{Q_k}$

The discrete process-noise covariance matrix $\boldsymbol{Q_k}$ is defined as

$$
\begin{aligned}
\boldsymbol{Q_k} &= E\{\boldsymbol{w_k}\boldsymbol{w_k}^T\} \,, \\
&= E\left\{\left[\int_0^{T_s} \boldsymbol{\phi}(\tau)\boldsymbol{w}(\tau)d\tau\right]\left[\int_0^{T_s} \boldsymbol{\phi}(\nu)\boldsymbol{w}(\nu)d\nu\right]^T\right\} \,, \\
&= \int_0^{T_s}\int_0^{T_s} \boldsymbol{\phi}(\tau)E\left\{\boldsymbol{w}(\tau)\boldsymbol{w}(\nu)^T\right\}\boldsymbol{\phi}(\nu)^T d\tau d\nu \,, \\
\boldsymbol{Q_k} &= \int_0^{T_s} \boldsymbol{\phi}(\tau)\boldsymbol{Q}\boldsymbol{\phi}(\tau)^T d\tau \,. \qquad \text{(B.13)}
\end{aligned}
$$

B.2.2 Transition Matrix for KFs

Let us consider the following linear and continuous state space model:

$$\dot{\boldsymbol{x}}(t) = \boldsymbol{F}\boldsymbol{x}(t) + \boldsymbol{G}\boldsymbol{u}(t) + \boldsymbol{w}(t) \,, \qquad \text{(B.14)}$$

where $\boldsymbol{F}$ and $\boldsymbol{G}$ are the time-invariant dynamics and input matrices, and where $\boldsymbol{w}(t)$ represents continuous process noise with a covariance matrix $\boldsymbol{Q} = E\{\boldsymbol{w}\boldsymbol{w}^T\}$.

Equation B.14 can be integrated as follows (also see Appendix B.1):

$$\boldsymbol{x}(t) = \boldsymbol{\Phi}(t)\boldsymbol{x}(0) + \int_0^t \boldsymbol{\Phi}(t-\sigma)\boldsymbol{B}\boldsymbol{u}(\sigma)d\sigma + \int_0^t \boldsymbol{\Phi}(t-\nu)\boldsymbol{w}(\nu)d\nu \,, \qquad \text{(B.15)}$$

where the continuous transition matrix is $\boldsymbol{\Phi}(t) = e^{\boldsymbol{F}t}$. A Taylor series expansion is used to approximate $\boldsymbol{\Phi}(t)$, yielding

$$\boldsymbol{\Phi}(t) = e^{\boldsymbol{F}t} = \boldsymbol{I} + \boldsymbol{F}t + \frac{(\boldsymbol{F}t)^2}{2!} + ... + \frac{(\boldsymbol{F}t)^n}{n!} + ... \quad . \qquad \text{(B.16)}$$

The discrete fundamental or transition matrix can be found by evaluating the continuous fundamental matrix at the sampling time T_s: $\boldsymbol{\Phi_k} = \boldsymbol{\Phi}(T_s)$. As demonstrated in several practical examples in the book by Zarchan and Musoff[1], the transition matrix can be approximated by taking only the first two elements of the Taylor series; adding higher-order terms does not bring any significant improvements in the performance of the filter. Therefore, in this work the discrete transition matrix is computed as

$$\boldsymbol{\Phi_k} \approx \boldsymbol{I} + \boldsymbol{F}T_s \,. \qquad \text{(B.17)}$$

[1] P. Zarchan and H. Musoff. *Fundamentals of Kalman Filtering: A Practical Approach, Second Edition.* Volume 208, Progress in Astronautics and Aeronautics, AIAA Inc., Reston, VA, 2005.

Appendix C
Nonlinear Transformations Used in the Longitudinal Controllers

C.1 Nonlinear Transformation $T1$ Between Second Time Derivative of Altitude $\ddot{h}$ and the Aircraft Normal Acceleration a_n

The altitude h corresponds to the down position of the center of gravity of the aircraft x_D in the inertial navigation frame (n). Newton's law states that

$$\sum_j \boldsymbol{F}_j = \Big[\frac{d}{dt}\Big(m\boldsymbol{v}\Big)\Big]_n = m\begin{pmatrix}\ddot{X}_N\\ \ddot{X}_E\\ \ddot{X}_D\end{pmatrix}, \tag{C.1}$$

i.e.,

$$\begin{pmatrix}0\\0\\mg\end{pmatrix} + C_b^n\left[\begin{pmatrix}F_{thrust}\\0\\0\end{pmatrix} + \begin{pmatrix}X^b\\Y^b\\Z^b\end{pmatrix}_{aero}\right] = m\begin{pmatrix}\ddot{x}_N\\ \ddot{x}_E\\ \ddot{x}_D\end{pmatrix}, \tag{C.2}$$

with

$$C_b^n = \begin{pmatrix}\cos\theta\cos\psi & \sin\theta\sin\phi\cos\psi - \cos\phi\sin\psi & \sin\theta\cos\phi\cos\psi + \sin\phi\sin\psi\\ \cos\theta\sin\psi & \sin\theta\sin\phi\sin\psi + \cos\phi\sin\psi & \sin\theta\cos\phi\sin\psi - \sin\phi\cos\psi\\ -\sin\theta & \sin\phi\cos\theta & \cos\phi\cos\theta\end{pmatrix}. \tag{C.3}$$

Equation C.2 may be rewritten as

$$\begin{pmatrix}0\\0\\mg\end{pmatrix} + C_b^n m\begin{pmatrix}f_x\\f_y\\f_z\end{pmatrix} = m\begin{pmatrix}\ddot{x}_N\\ \ddot{x}_E\\ \ddot{x}_D\end{pmatrix}, \tag{C.4}$$

where f_x, f_y, f_z represent the accelerometer measurement data. Since the vertical acceleration $\ddot{x}_D$ is of interest, the corresponding equation in (C.2) is expanded as follows:

$$\ddot{x}_D = g - \sin\theta\ f_x + \sin\phi\cos\theta\ f_y + \cos\phi\cos\theta\ f_z\ . \tag{C.5}$$

Considering that the roll angle ϕ and the sideslip angle are small and close to zero yields $Y^b \approx 0$. Therefore the term $\sin\phi\cos\theta\ Y^b/m$ is neglected and (C.5) becomes

$$\ddot{x}_D = g\left[1 - \frac{f_x}{g}\sin\theta + \frac{f_z}{g}\cos\phi\cos\theta\right]\ . \tag{C.6}$$

For moderate values of the pitch angle θ, the term $-\sin\theta\ f_x/g$ is much smaller than 1 and is therefore neglected. The specific normal acceleration is $a_n = f_z/g$. Finally, the nonlinear transformation $T1$ between the second time derivative of the altitude $\ddot{h}$ and the aircraft's normal acceleration a_n is expressed as follows:

$$\ddot{x}_D = \ddot{h} = g\left[1 + \cos\phi\cos\theta\ a_n\right]\ . \tag{C.7}$$

C.2 Nonlinear Transformation $T2$ Between the Angle of Attack α and the Aircraft Normal Acceleration a_n

In Chap. 3, the relationship between the aircraft-body accelerations and the forces acting on the aircraft is established and recalled here for convenience:

$$\begin{pmatrix} -g\sin\theta \\ g\sin\phi\cos\theta \\ g\cos\phi\cos\theta \end{pmatrix} + \frac{1}{m}\left[\begin{pmatrix} F_{thrust} \\ 0 \\ 0 \end{pmatrix} + C_w^b \begin{pmatrix} X^w \\ Y^w \\ Z^w \end{pmatrix}\right] - \begin{pmatrix} qw - rv \\ ru - pw \\ pv - qu \end{pmatrix} = \begin{pmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{pmatrix}\ , \tag{C.8}$$

with

$$C_w^b = \begin{pmatrix} \cos\alpha\cos\beta & -\cos\alpha\sin\beta & -\sin\alpha \\ \sin\beta & \cos\beta & 0 \\ \sin\alpha\cos\beta & -\sin\alpha\sin\beta & \cos\alpha \end{pmatrix}\ . \tag{C.9}$$

The last row of (C.8) is written as follows:

$$\dot{w} = g\cos\phi\cos\theta + \frac{1}{m}\left[X^w\sin\alpha\cos\beta - Y^w\sin\alpha\sin\beta + Z^w\cos\alpha\right] - pv + qu\ . \tag{C.10}$$

Therefore, the z-body axis aerodynamic force is

$$Z^b = \frac{1}{m}\left[X^w\sin\alpha\cos\beta - Y^w\sin\alpha\sin\beta + Z^w\cos\alpha\right]. \tag{C.11}$$

The aircraft's normal and specific acceleration[1] measured by the z-accelerometer is

$$a_n = \frac{1}{mg}\left[X^w \sin\alpha\cos\beta - Y^w \sin\alpha\sin\beta + Z^w\cos\alpha\right] , \tag{C.12}$$

with

$$X^w = \bar{q}S\left[C_{X1} + C_{X\alpha}\alpha + C_{X\alpha 2}\alpha^2 + C_{X\beta 2}\beta^2\right] , \tag{C.13}$$

$$Y^w = \bar{q}SC_{Y1}\beta \quad \text{and} \quad Z^w = \bar{q}S\left[C_{Z1} + C_{Z\alpha}\alpha\right] . \tag{C.14}$$

Considering α to be small (in practice limited to 0 and 0.21 rad) yields $\alpha^2 \ll \alpha$ and thus only the first-order term in α is considered in (C.13). Under normal flight conditions the sideslip angle β is also close to zero, which leads to the following simplified version for the normal specific acceleration a_n:

$$\begin{aligned} a_n &\approx \frac{1}{mg}\left[\alpha\bar{q}S(C_{X1} + C_{X\alpha}\alpha) + \bar{q}S(C_{Z1} + C_{Z\alpha}\alpha)\right] , \\ &\approx \frac{\bar{q}S}{mg}\left(\left[C_{X1} + C_{Z\alpha}\right]\alpha + C_{Z1}\right) . \end{aligned} \tag{C.15}$$

Simulation results show that a_n is almost perfectly estimated using (C.15). The acceleration a_n is of the form $a_n \approx \bar{q}(\xi_1\alpha + \xi_2)$, where ξ_1 and ξ_2 are constant values and $\bar{q} = \rho V_T^2/2$.

C.3 Nonlinear Transformation $T3$ Between $\dot{\alpha}$ and the Pitch Rate q

The angle of attack α is related to the aircraft-body velocities u and w by

$$\tan(\alpha) = \frac{w}{u} . \tag{C.16}$$

Differentiating (C.16) with respect to time yields

$$\dot{w} = \dot{u}\ \tan\alpha + u\left(1 + \tan^2\alpha\right)\dot{\alpha} . \tag{C.17}$$

Given the assumption that α and $\dot{u}$ are small, (C.17) can be simplified to

$$\dot{w} = u\dot{\alpha} . \tag{C.18}$$

The last row of (C.8) provides a relationship between the pitch rate q and the z-body axis velocity w as follows:

[1] Measured in multiples of g.

$$g\cos\phi\cos\theta + Z^b/m - pv + qu = \dot{w} \\ = u\dot{\alpha}\ . \tag{C.19}$$

The x-body axis velocity u can be approximated by the total airspeed V_T, yielding

$$g\cos\phi\cos\theta + Z^b/m - pv + qV_T = V_T\dot{\alpha}\ . \tag{C.20}$$

The nonlinear transformation $T3$ aims at establishing a relationship between $\dot{\alpha}$ and the pitch rate q. Therefore, (C.20) is rewritten as

$$\dot{\alpha} = q + \frac{g}{V_T}\left(\cos\phi\cos\theta + \frac{Z^b}{mg}\right) - \frac{pv}{V_T}\,, \tag{C.21}$$

where the term Z^b/mg corresponds to the normal acceleration a_n measured at the aircraft center of gravity by the accelerometer in the z-body direction, and where the term pv/V_T is usually very close to zero during normal flight condition with small side-slipping. Finally, the expression for the nonlinear transformation $T3$ is obtained as follows:

$$\dot{\alpha} = q + \frac{g}{V_T}\left(\cos\phi\cos\theta + a_n\right)\ . \tag{C.22}$$

Appendix D
Nonlinear Transformation Used in the Lateral-directional Controller

D.1 Dynamics of the Sideslip Angle

This section establishes the relationship between the dynamics of the sideslip angle $\dot{\beta}$ and the roll and yaw rates, p and r, respectively.

By definition the sideslip angle β is obtained as follows:

$$\beta = \sin^{-1}\left(\frac{v}{V_T}\right) , \tag{D.1}$$

where the aircraft's y-body axis velocity is v, and the aircraft's airspeed is V_T.

Equation D.1 is rearranged as $V_T \sin\beta = v$, and is differentiated with respect to time, yielding

$$\dot{V}_T \ \sin\beta + V_T\dot{\beta}\cos\beta = \dot{v} . \tag{D.2}$$

Based on the assumptions that the sideslip angle remains small and close to zero and that the airspeed V_T is almost constant, (D.2) is simplified as follows:

$$\dot{\beta} \approx \frac{\dot{v}}{V_T} . \tag{D.3}$$

An expression for the dynamics of the side velocity v is found in (C.8) and is recalled here for convenience:

$$\dot{v} = g\,\sin\phi\cos\theta + \frac{1}{m}\left(X^w\sin\beta + Y^w\cos\beta\right) + pw - ru . \tag{D.4}$$

Using the facts that $u \approx V_T\,\cos\alpha$, $w \approx V_T\sin\alpha$, $\beta \approx 0$, $Y^w \approx m\,a_y$, where a_y is the accelerometer y-axis measurement, yields the simplified expression for $\dot{v}$,

$$\dot{v} \approx g\sin\phi\cos\theta + a_y + pV_T\sin\alpha - rV_T\cos\alpha . \tag{D.5}$$

The final expression for $\dot{\beta}$ is obtained by combining (D.3) and (D.5) yielding

$$\dot{\beta} = \frac{1}{V_T}\left[g \sin\phi \cos\theta + a_y\right] + p \sin\alpha - r \cos\alpha \; . \tag{D.6}$$

D.2 Roll Angle Command Signal and Equation Governing a Coordinated Turn

Let us assume that the aircraft is flying straight, level, and at constant altitude. In such a case, the lift force $\mathbf{Z^b}$ generated by the wings compensates the weight $m\mathbf{g}$ of the aircraft. When flying a turn, the aircraft banks by a roll angle ϕ, and the lift force provides a lateral component that accelerates the aircraft as shown in Fig. D.1. This lateral force F_l is computed as follows:

$$F_l = Z^b \sin\phi \; . \tag{D.7}$$

The lateral acceleration is obtained as

$$a_l = \frac{Z^b \sin\phi}{m} \; . \tag{D.8}$$

If the magnitude of Z^b is replaced by mg, the lateral acceleration is simplified to

$$a_l = g \sin\phi \; . \tag{D.9}$$

Conversely, in order to produce the lateral acceleration a_l, the aircraft has to bank to a roll angle ϕ equal to

$$\phi = \sin^{-1}(\frac{a_l}{g}) \approx \frac{a_l}{g} \; . \tag{D.10}$$

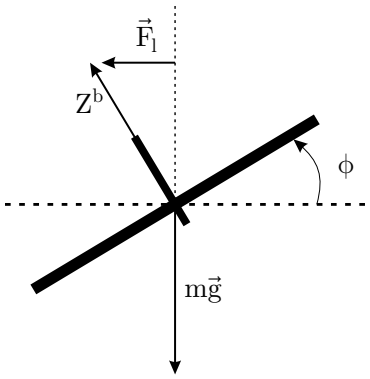


Fig. D.1 Lateral acceleration in a turn

D.3 Law of Cosines

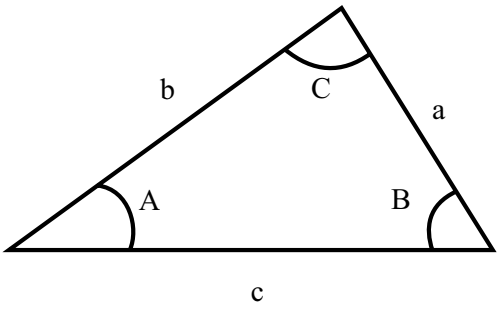


Fig. D.2 Triangle parameters for the law of cosines

The law of cosines gives any edge of a triangle as a function of the two adjacent edges and the opposite angle with

$$\begin{aligned} a^2 &= b^2 + c^2 - 2bc\cos A\,, \\ b^2 &= a^2 + c^2 - 2ac\cos B\,, \\ c^2 &= a^2 + b^2 - 2ab\cos C\,. \end{aligned} \tag{D.11}$$

Appendix E
Linearization of the Aircraft Model at 30 m/s

The aircraft is linearized around the operating point corresponding to a straight and level flight at a constant altitude of 500 m above sea level and at a constant speed of 30 m/s.

E.1 Longitudinal Linear Model

$$
\begin{aligned}
A_{Long} &= \begin{pmatrix} -4.7796 & 0 & -4.5420 & 0 \\ 0 & -0.0830 & -0.8660 & -9.81 \\ 1 & -0.0215 & -3.6573 & 0 \\ 1 & 0 & 0 & 0 \end{pmatrix}, \\
B_{Long} &= \begin{pmatrix} 27.4128 & 0 \\ 0 & -1.1 \cdot 10^{-4} \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \\
C_{Long} &= \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}, \\
D_{Long} &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.
\end{aligned}
\tag{E.1}
$$

E.2 Lateral Linear Model

$$
\begin{aligned}
A_{Lat} &= \begin{pmatrix} -11.4540 & 2.7185 & -19.4399 & 0 \\ 0.5068 & -2.9875 & 23.3434 & 0 \\ 0.0922 & -0.9957 & -0.4680 & 0.3256 \\ 1 & 0.0926 & 0 & 0 \end{pmatrix}, \\
B_{Lat} &= \begin{pmatrix} 78.4002 & -2.7282 \\ -3.4690 & 13.9685 \\ 0 & 0 \\ 0 & 0 \end{pmatrix}, \\
C_{Lat} &= \begin{pmatrix} 1\,0\,0\,0 \\ 0\,1\,0\,0 \\ 0\,0\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix}, \\
D_{Lat} &= \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.
\end{aligned}
\tag{E.2}
$$

Appendix F
Nomenclature

Symbol	Definition
a	temperature gradient
α	angle of attack
b	wing span
(b)	body-fixed frame
$\langle b \rangle$	body-fixed coordinate system
β	sideslip angle
$\bar{c}$	mean aerodynamic chord
$C_{F_{T1}} ... C_{F_{T3}}$	thrust force derivatives
$C_{L...}$	roll torque derivatives
$C_{M...}$	pitch torque derivatives
$C_{N...}$	yaw torque derivatives
$C_{X...}$	drag force derivatives
C_{Y1}	side force derivative
C_{Z1}, $C_{Z\alpha}$	lift force derivatives
$\boldsymbol{C}_a^b$	direction cosine matrix, transformation from (a) to (b)
$\boldsymbol{C_v}$	virtual control command, input of the control allocator
D	propeller diameter
δ_a	aileron deflection
δ_e	elevator deflection
δ_r	rudder deflection
δ_t	throttle position
F_T	component of $\boldsymbol{F_T}$ in $\boldsymbol{x}_b$ direction
g	component of $\boldsymbol{g}$ in $\boldsymbol{z}_n$ direction
$\boldsymbol{g}$	earth gravity vector
h	altitude above sea level
θ	pitch angle

Symbol	Definition
$\boldsymbol{I}^b$	airplane inertia matrix in the body-fixed frame
J	thrust advance ratio
L^b	roll torque, $\boldsymbol{x}_b$ direction
m	airplane mass
M^b	pitch torque, $\boldsymbol{y}_b$ direction
$\boldsymbol{M}$	external torque vector affecting the rigid body
(n)	local navigation frame
$\langle n \rangle$	local navigation coordinate system
n	engine speed
n_c	commanded engine rotational speed
N^b	yaw torque, $\boldsymbol{z}_b$ direction
p	roll rate, $\boldsymbol{x}_b$ direction
$\tilde{p}$	dimensionless roll rate, $\boldsymbol{x}_b$ direction
$\boldsymbol{p}$	inertial position vector of the rigid body
$\bar{q}$	dynamic pressure
q	pitch rate, $\boldsymbol{y}_b$ direction
$\tilde{q}$	dimensionless pitch rate, $\boldsymbol{y}_b$ direction
q_0	1st element of the quaternion $\boldsymbol{q}^n$
q_1	2nd element of the quaternion $\boldsymbol{q}^n$
q_2	3rd element of the quaternion $\boldsymbol{q}^n$
q_3	4th element of the quaternion $\boldsymbol{q}^n$
$\boldsymbol{q}^n$	quaternion that describes the rotation from $\langle n \rangle$ to $\langle b \rangle$
r	yaw rate, $\boldsymbol{z}_b$ direction
$\tilde{r}$	dimensionless yaw rate, $\boldsymbol{z}_b$ direction
R	air gas constant
ρ	air density
S	wing surface
T	ambient temperature
T_0	ambient temperature at mean sea level
τ_n	time constant of the engine turn rate
u	inertial velocity component, $\boldsymbol{x}_b$ direction
u_T	airspeed component, $\boldsymbol{x}_b$ direction
v	inertial velocity component, $\boldsymbol{y}_b$ direction
v_T	airspeed component, $\boldsymbol{y}_b$ direction
v_N	component of v in $\boldsymbol{x}_n$ direction
v_E	component of v in $\boldsymbol{y}_n$ direction
v_D	component of v in $\boldsymbol{z}_n$ direction
V_T	airspeed component, $\boldsymbol{x}_w$ direction
$\boldsymbol{v}$	inertial velocity vector of the rigid body
$\boldsymbol{V_T}$	airspeed vector

Symbol	Definition
ϕ	roll angle
w	inertial velocity component, $\boldsymbol{z}_b$ direction
w_T	airspeed component, $\boldsymbol{z}_b$ direction
W_N	component of $\boldsymbol{W}$ in $\boldsymbol{x}_n$ direction
W_E	component of $\boldsymbol{W}$ in $\boldsymbol{y}_n$ direction
W_D	component of $\boldsymbol{W}$ in $\boldsymbol{z}_n$ direction
$\boldsymbol{W}$	wind velocity vector
x_N	component of $\boldsymbol{p}$ in $\boldsymbol{x}_n$ direction
x_E	component of $\boldsymbol{p}$ in $\boldsymbol{y}_n$ direction
x_D	component of $\boldsymbol{p}$ in $\boldsymbol{z}_n$ direction
X^w	drag force, $\boldsymbol{x}_w$ direction
Y^w	side force, $\boldsymbol{y}_w$ direction
ψ	heading angle
X^w	lift force, $\boldsymbol{z}_w$ direction
$\boldsymbol{\omega_{b/a}}$	rotational velocity vector of $\langle b \rangle$ with respect to $\langle a \rangle$
γ	flight path angle
$\hat{y}$	model-based estimate of the measurement data
$y(t)$	measurement data

Table F.1 Guidance system variables

Symbol	Definition
a_l	lateral acceleration
D_{NFZ}	distance between the aircraft's center of gravity and the NFZ center
L_1	distance between the aircraft's center of gravity and the guidance reference point P
P	guidance reference point
P_E	East position of the reference point P
P_N	North position of the reference point P
ϕ_{max}	maximum roll angle allowed
R_{LA}	look-ahead distance
R_{NFZ}	radius of the circle defining the NFZ
R_{min}	minimum turn radius
τ_{roll}	time to roll to ϕ_{max}
WP_k	waypoint number k
X_E	East position of the aircraft's center of gravity
X_N	North position of the aircraft's center of gravity
y	cross-track error distance

Table F.2 Aircraft model parameters

Parameter	Value	Unit	Definition
m	28	[kg]	airplane mass
$\boldsymbol{I}^b$	$\boldsymbol{I}^b = \begin{bmatrix} 2.56 & 0 & 0.5 \\ 0 & 10.9 & 0 \\ 0.5 & 0 & 11.3 \end{bmatrix}$	[kg·m^2]	airplane inertia matrix
S	1.80	[m^2]	wing surface
$\bar{c}$	0.58	[m]	mean aerodynamic chord
b	3.1	[m]	wing span
D	0.79	[m]	propeller diameter
$C_{F_{T1}}$	8.42×10^{-2}	[–]	thrust derivative
$C_{F_{T2}}$	-1.36×10^{-1}	[–]	thrust derivative
$C_{F_{T3}}$	-9.28×10^{-1}	[–]	thrust derivative
C_{Z1}	1.29×10^{-2}	[–]	lift derivative
$C_{Z\alpha}$	-3.25	[–]	lift derivative
C_{X1}	-2.12×10^{-2}	[–]	drag derivative
$C_{X\alpha}$	-2.66×10^{-2}	[–]	drag derivative
$C_{X\alpha 2}$	-1.55	[–]	drag derivative
$C_{X\beta 2}$	-4.01×10^{-1}	[–]	drag derivative
C_{Y1}	-3.79×10^{-1}	[–]	side force derivative
C_{La}	6.79×10^{-2}	[–]	roll derivative
$C_{L_{a1}} = -C_{L_{a2}}$	-3.395×10^{-2}	[–]	roll derivative
$C_{L_{e1}} = -C_{L_{e2}}$	-0.485×10^{-2}	[–]	roll derivative
$C_{L\beta}$	-1.30×10^{-2}	[–]	roll derivative
$C_{L\tilde{p}}$	-1.92×10^{-1}	[–]	roll derivative
$C_{\tilde{r}}$	3.61×10^{-2}	[–]	roll derivative
C_{M1}	2.08×10^{-2}	[–]	pitch derivative
C_{Me}	5.45×10^{-1}	[–]	pitch derivative
$C_{M_{e1}} = C_{M_{e2}}$	2.725×10^{-1}	[–]	pitch derivative
$C_{M_{a1}} = C_{M_{a2}}$	0.389×10^{-1}	[–]	pitch derivative
$C_{M\alpha}$	-9.03×10^{-2}	[–]	pitch derivative
$C_{M\tilde{q}}$	-9.83	[–]	pitch derivative
$C_{N_{\delta_r}}$	5.34×10^{-2}	[–]	yaw derivative
$C_{N\beta}$	8.67×10^{-2}	[–]	yaw derivative
$C_{N\tilde{r}}$	-2.14×10^{-1}	[–]	yaw derivative
τ_n	0.4	[s]	time constant of the engine

Index

Other titles published in this series (continued):

Soft Sensors for Monitoring and Control of Industrial Processes
Luigi Fortuna, Salvatore Graziani, Alessandro Rizzo and Maria G. Xibilia

Adaptive Voltage Control in Power Systems
Giuseppe Fusco and Mario Russo

Advanced Control of Industrial Processes
Piotr Tatjewski

Process Control Performance Assessment
Andrzej W. Ordys, Damien Uduehi and Michael A. Johnson (Eds.)

Modelling and Analysis of Hybrid Supervisory Systems
Emilia Villani, Paulo E. Miyagi and Robert Valette

Process Control
Jie Bao and Peter L. Lee

Distributed Embedded Control Systems
Matjaž Colnarič, Domen Verber and Wolfgang A. Halang

Precision Motion Control (2nd Ed.)
Tan Kok Kiong, Lee Tong Heng and Huang Sunan

Optimal Control of Wind Energy Systems
Iulian Munteanu, Antoneta Iuliana Bratcu, Nicolaos-Antonio Cutululis and Emil Ceangă

Identification of Continuous-time Models from Sampled Data
Hugues Garnier and Liuping Wang (Eds.)

Model-based Process Supervision
Arun K. Samantaray and Belkacem Bouamama

Diagnosis of Process Nonlinearities and Valve Stiction
M.A.A. Shoukat Choudhury, Sirish L. Shah, and Nina F. Thornhill

Magnetic Control of Tokamak Plasmas
Marco Ariola and Alfredo Pironti

Real-time Iterative Learning Control
Jian-Xin Xu, Sanjib K. Panda and Tong H. Lee

Deadlock Resolution in Automated Manufacturing Systems
ZhiWu Li and MengChu Zhou

Model Predictive Control Design and Implementation Using MATLAB®
Liuping Wang

Predictive Functional Control
Jacques Richalet and Donal O'Donovan

Fault-tolerant Control Systems
Hassan Noura, Didier Theilliol, Jean-Christophe Ponsart and Abbas Chamseddine
Publication due August 2009

Control of Ships and Underwater Vehicles
Khac Duc Do and Jie Pan
Publication due September 2009

Detection and Diagnosis of Stiction in Control Loops
Mohieddine Jelali and Biao Huang (Eds.)
Publication due October 2009

Stochastic Distribution Control System Design
Lei Guo and Hong Wang
Publication due November 2009

Advanced Control and Supervision of Mineral Processing Plants
Daniel Sbárbaro and René del Villar (Eds.)
Publication due December 2009

Active Braking Control Design for Road Vehicles
Sergio M. Savaresi and Mara Tanelli
Publication due January 2010