TECHNISCHE UNIVERSITÄT MÜNCHEN

Flight Control Function Enhancements For Novel Aerial Vehicles

David Rainer Seiferth, M.Sc.

Vollständiger Abdruck der von der TUM School of Engineering and Design zur Erlangung des akademischen Grades eines

Doktors der Ingenieurwissenschaften

genehmigten Dissertation.

Vorsitzender:	Prof. Dr.-Ing. Klaus Drechsler
Prüfer der Dissertation:	1. Prof. Dr.-Ing. Florian Holzapfel
	2. Prof. Dr. Rubens Junqueira Magalhães Afonso

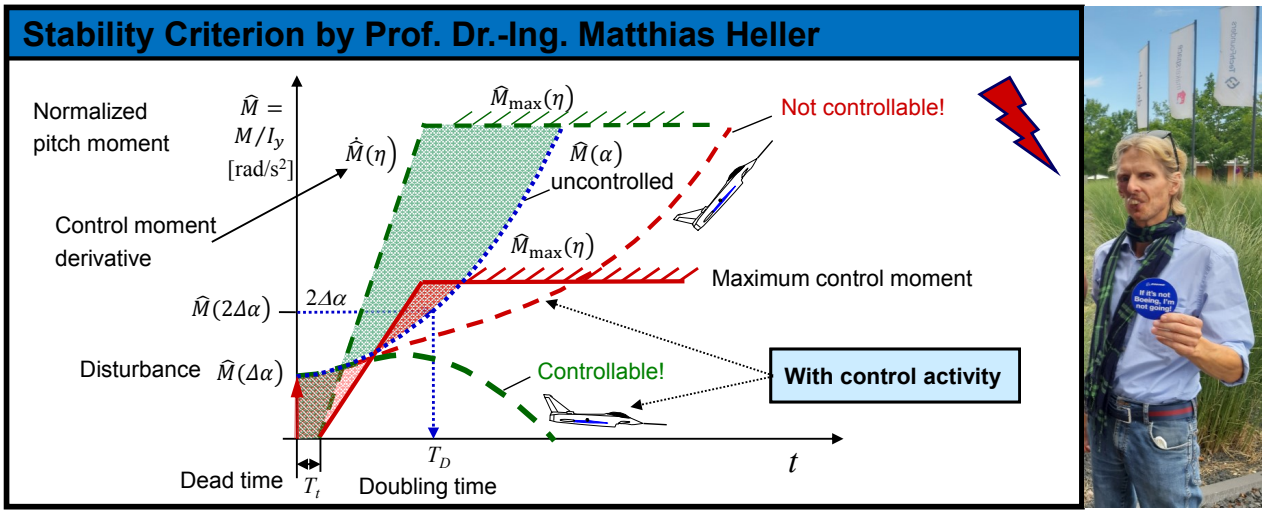
Die Dissertation wurde am 07.09.2021 bei der Technischen Universität München eingereicht und durch die TUM School of Engineering and Design am 24.02.2022 angenommen.

Abstract

The aerospace evolution in the 21st century demands new functional capabilities in the flight control system of novel aerial vehicle types, which vary significantly in size, configuration, take-off weight, and concept of operation.
This thesis contributes three flight control function enhancements for any of these aerial vehicle types.
The first functionality is an automatic geofencing concept, which prevents the aerial vehicle from leaving a pre-defined flight area in two- or three-dimensional airspace. The algorithm is embedded in a conventional flight control system and tested in model-in-the loop simulations. The geofencing functionality serves as risk mitigation strategy and is thus especially important for the certification of new unmanned or optionally-piloted aerial vehicles.
Secondly, this thesis elaborates different robustness metrics for the application to linearized closed-loop systems of new aerial vehicles and provides thus the important principles to assess the robustness of such aerial vehicles.
Besides the classical single-input-single-output margins, multiple-input-multiple-output robustness metrics are introduced and evaluated with regard to their conservatism. In addition, the classical and advanced μ-analysis are detailed, which assess the robustness of the closed-loop system with regard to stability or a specified conformity borderline considering a detailed uncertainty model with deterministic intervals. However, newly-developed uncertainty models are specified by stochastic probability distributions. Therefore, a new probabilistic-μ approach is proposed, which maximizes a lower bound of the probability to achieve stability or a conformity borderline.
The robustness metrics are applied within this thesis to an electric vertical take-off and landing (eVTOL) demonstrator. An incremental nonlinear dynamic inversion (INDI) controller designed for hover and transition is considered within the presented study. A horizontal velocity envelope is determined, in which the detailed robustness metric requirements are fulfilled. The results can be used as envelope during the first flights of the aerial vehicles.
At last, the development of a ground controller for runway centerline tracking during automatic take-off and landing of a novel diamond-shaped unmanned aerial vehicle (UAV) is shown. The entire development, from the initial design and implementation until on-ground tests, is introduced step-by-step within this thesis. The controller is verified in flight during the first flight campaign of the aerial vehicle in Overberg, South Africa.

Prof. Dr.-Ing. Matthias Heller

† 2. April 2020



Stability criterion illustration by Prof. Dr.-Ing. Matthias Heller[1]

This page is reserved to my former supervisor Prof. Dr.-Ing. Matthias Heller. He was flight mechanic expert at Airbus Defense and Space and professor for highly-augmented flight systems at the Institute of Flight System Dynamics of the Technical University of Munich. Furthermore, he was the originator of the lectures "Flight Dynamic Challenges for Highly-Augmented Aircraft" and "Flying / Handling Qualities towards Pilot-in-the-Loop Oscillations". Matthias Heller is the inventor of the stability criterion for naturally unstable aircraft, which is illustrated in the figure above. Using the stability criterion, one can determine if the unstable aircraft design is controllable under the consideration of chosen actuator dynamics and dead times in the system.

Thank you, Matthias Heller for the chance to be supervised by you. During the time of your supervision I learned so much in flight mechanics and control, but also about life. I still admire your spirit for aerospace and your love to live, no matter what comes. My thoughts will be always with you!

[1] Translated from: Matthias Heller, Lecture notes for "Flight Dynamic Challenges for Highly-Augmented Aircraft", Technical University of Munich, Winter Semester 2018/19.

Acknowledgment

First, I would like to thank my supervisors Prof. Dr.-Ing. Florian Holzapfel and Prof. Dr. Rubens Afonso. Thank you, Florian Holzapfel to give me the Ph.D. opportunity at your institute. When I started studying at the Technical University of Munich, you inspired me to dedicate my studies to flight dynamics and control. Although the time during my Ph.D. program was probably the hardest one of my life, you gave me numerous opportunities which I never have imagined. After M. Heller passed away, you agreed to be my Ph.D. supervisor without batting an eye. Thank you!

I would also like to thank Rubens Afonso. You had always an open ear and supported me any time, when I needed, especially in the elaboration of the reduced conservatism proof. Thank you!

Special thanks goes to my colleagues and friends at the Institute of Flight System Dynamics. Especially, I would like to thank the SAGITTA core team Martin Kügler, Stanislav Braun and Markus Geiser. Martin Kügler, you helped me to cope with the task in the hot phase of the project and it was always a pleasure for me to discuss with you topics in aerospace. Thank you, Stanislav Braun and Martin Geiser for your commitment to the project. Without you, it would have not been possible that SAGITTA became eventually a success.

Another special thanks goes to Benedikt Grüter and Johannes Diepolder. Thank you, Benedikt Grüter for the intense discussions on the geofencing concept and for the good times we had together on our business trips. Thank you, Johannes Diepolder. Without our discussions on the probabilistic-μ approach, the algorithm would have never been as good as it is now. Especially in the last phase of this thesis, you always supported me such that I was able to finish this thesis. Thank you! I would like to thank my friend Julian Leucht for reviewing this thesis.

Special thanks goes also to my parents, my grandparents and my brother. You always supported me in what I was doing, although it was hard for you to understand, why, for example, a n-dimensional uncertainty space really matters. Last but not least, I would like to thank my girlfriend for the understanding to spend evenings and weekends on this thesis over the years. With the distraction, you gave me, I was able to clear my mind and to find the right balance. Thank you!

Garching, May 21, 2022 David Seiferth

Contents

Abstract i

Acknowledgment v

1 Introduction 1

1.1 The Recent Aerospace Evolution 2

1.1.1 New Markets 2

1.1.2 New Configurations 2

1.1.3 New Technologies 3

1.1.4 New Regulations 5

1.2 Objectives, State-of-the-Art and Contributions 8

1.2.1 Automatic Geofencing For Safety Enhancement of Aerial Vehicle Operation 9

1.2.2 Stability and Robustness Analysis of Control Laws for VTOL Aircraft 13

1.2.3 Design, Development and Testing of Automatic Ground Control Functions of a Novel UAV for Automatic Take-Off and Landing 18

1.3 Thesis Composition 21

2 Automatic Geofencing For Safety Enhancement of Unmanned Aerial Vehicle Operation 22

2.1 Geofencing Terminology and Development Environment 24

2.2 Two-Dimensional Geofencing Algorithm 25

2.2.1 The Pre-flight Offline Algorithm 25
2.2.1.1 Flight Area Requirements and Definition 25
2.2.1.2 Verification of the Flight Area 26
2.2.2 Safety Area Generation 29
2.2.2.1 Safety Area Requirements 29
2.2.2.2 Safety Margin For Dedicated Evasive Maneuver 29
2.2.2.3 Voronoi Map Creation 30
2.2.2.4 Offset Polygon Generation 32
2.2.2.5 Safety Area Derivation 35
2.2.2.6 Safety Area Validity 38
2.2.3 Two-Dimensional Online Geofencing Algorithm 40
2.2.3.1 Two-Dimensional Safe Area Detection Algorithm 40
2.2.3.2 Evasive Maneuver Planning and Conduction 41
2.2.3.3 Online Algorithm Testing 42
2.3 Functional Extension For Geofencing in Three Dimensional Space 44
2.3.1 Pre-Flight Offline Algorithm For Three-Dimensional Space 44
2.3.1.1 3D-Flight Area Definition and Specification 44
2.3.1.2 z-Level and y-Level Creation for 3D-Flight Area 50
2.3.1.3 Intersection Point Calculation 52
2.3.1.4 Safety Area Generation for y- and z-Levels 54
2.3.2 Online Safe Area Detection Algorithm
For Three-Dimensional Space 55
2.3.2.1 Polygon Interpolation 56
2.3.2.2 Point-in-Polygon Algorithm 58
2.3.2.3 Model-in-the-Loop Verification 58
2.4 Evasive maneuver 59
2.4.1 Constraints 59
2.4.2 Equations of Motion 60
2.4.3 Control Logic for Maneuver Trigger 63

2.4.4 Lateral Case 64
2.4.5 Vertical Case 66
2.4.6 Combined Lateral and Vertical Case 68
2.5 Testing and Verification 70
2.5.1 Model-in-the Loop Simulation 70
2.5.2 Simulation Results 71
2.5.2.1 Lateral Maneuver 71
2.5.2.2 Vertical Maneuver 71
2.5.2.3 Combined Maneuver 75
2.6 Summary and Outlook 78

3 Stability and Robustness Analysis of Control Laws for VTOL aircraft 79

3.1 Published Stability, Robustness and Performance Requirements within Certification Documents 81
3.2 Feedback Control Revisited 83
3.2.1 Nomenclature 83
3.2.2 Closed-Loop Transfer Function, Sensitivity Transfer Function and Complementary Sensitivity Transfer Function 83
3.2.3 Loop Breaking in MIMO Systems 87
3.2.4 Stability, Robustness and Performance 90
3.3 Uncertainties in MIMO Systems 91
3.3.1 Uncertainty Origin 91
3.3.2 Definition of Uncertainties 91
3.3.3 Real versus complex scalar uncertainties 92
3.3.4 Uncertainties with Limited Intervals versus Unbounded Stochastic Uncertainties 94
3.3.5 Structured Uncertainty versus Unstructured Uncertainties 95
3.3.6 Uncertainty Representation 97
3.3.7 Uncertainty Modeling 100

3.3.7.1 Uncertainty Classes within Limited Intervals 100
3.3.7.2 Uncertainty Classes under Consideration of Unbounded Stochastic Uncertainties 106
3.4 Stability Criteria 107
3.4.1 Poles 107
3.4.2 Nyquist Criterion 108
3.4.3 Bode's Stability Condition 108
3.5 Robustness and Performance Criteria 108
3.5.1 SISO Robustness Metrics 109
3.5.1.1 Gain and Phase Margin 109
3.5.1.2 Bandwidth 109
3.5.2 SISO Nichols Disk Margin 112
3.5.3 Generic MIMO Block Structure and its stability condition 114
3.5.3.1 Small Gain Theorem 114
3.5.3.2 Linear Fractional Transformation 115
3.5.3.3 Stability Condition of the $\underline{M}\ \underline{\Delta}$-Structure 117
3.5.4 MIMO Gain and Phase Margin 118
3.5.4.1 Complementary Sensitivity Function 118
3.5.4.2 Sensitivity Function 123
3.5.4.3 Balanced Sensitivity Function 126
3.6 Reduced Conservatism Evaluation of MIMO Margins 129
3.6.1 Mathematical Preliminaries 129
3.6.2 Lower Gain Margins 130
3.6.2.1 Sensitivity Transfer Function 130
3.6.2.2 Complementary Sensitivity Transfer Function 131
3.6.3 Upper Gain Margins 131
3.6.3.1 Sensitivity Transfer Function 131
3.6.3.2 Complementary Sensitivity Transfer Function 132
3.6.4 Phase Margins 133

3.6.5 Conclusion and MIMO Margin Conservatism Discussion 135
3.7 μ_{Δ}-Uncertainty Analysis for Guaranteed Stability 137
3.7.1 Upper and Lower Bound 139
3.7.2 Tighter Lower and Upper Bound 141
3.8 Advanced Robust Performance Criteria 144
3.8.1 $\mu_{\Delta\Gamma}$-Uncertainty Analysis for Guaranteed Handling Qualities 144
3.8.2 Worst-Case Uncertainty Transfer Function for Stability Compliance 146
3.8.3 Worst-Case Uncertainty Transfer Function for Handling Quality Compliance 152
3.8.4 Verification of Worst-Case Uncertainty Transfer Functions 154
3.9 Probabilistic-μ with Mapped Uncertainties 159
3.9.1 General Idea 159
3.9.2 The Algorithm 160
3.9.3 Transformation Mapped Uncertainties 162
3.9.4 μ-Algorithm Exploitation 163
3.9.5 Probability Calculation 164
3.9.6 Maximization of Probability 165
3.9.7 Verification 166
3.9.7.1 Two-Dimensional Uncertainty Space 167
3.9.7.2 Three- and Four-Dimensional Uncertainty Space 168
3.9.8 Conclusion and Outlook 170
3.10 Application of Robustness Metrics on an eVTOL Demonstrator 171
3.10.1 eVTOL Demonstrator Configuration, System Architecture and Development Process 172
3.10.2 Nonlinear Simulation Model and its Linear Representation 172
3.10.3 Incremental Nonlinear Dynamic Inversion Attitude Control Structure 189

3.10.4 Building and Verification between Nonlinear and Linear Closed-Loop Model 196
3.10.5 Selection of Analysis Points 200
3.10.6 Results of the SISO Metrics 202
3.10.7 Results of MIMO Gain and Phase Margins 210
3.10.7.1 Evaluation with regard to Horizontal Flight Envelope of Closed-Loop System 211
3.10.7.2 Evaluation with respect to Elaborated Conservatism Proof . 218
3.10.8 μ-Analysis with limited uncertainty intervals 220
3.10.8.1 μ-Analysis with real and complex uncertainties 221
3.10.8.2 μ-Analysis with real uncertainties 227
3.11 Comparison between SISO Metrics, MIMO Metrics and μ-analysis Results . 230
3.12 Application of probabilistic-μ analysis 233
3.12.1 Algorithm Set-up 233
3.12.2 Results 235
3.12.2.1 Probability over grid without maximization 235
3.12.2.2 Selected trim points with maximization: 236
3.12.3 Conservatism and Ideas for Future Work 240
3.12.3.1 Conservatism 240
3.12.3.2 Ideas for future work 241

4 Automatic Ground Control Functions of a Novel UAV 243

4.1 The SAGITTA Demonstrator 244
4.1.1 Configuration 244
4.1.2 Concept of Operation 244
4.1.3 System Architecture 249
4.1.4 Flight Control System 249
4.1.4.1 Design and Structure 249
4.1.4.2 Software Development Process 250

4.2 Modeling & Simulation of the SAGITTA Demonstrator 251
4.2.1 SAGITTA Simulation Model 251
4.2.2 SAGITTA Landing Gear Model 252
4.2.2.1 Wheel Model 255
4.2.2.2 Vehicle Dynamics Model 255
4.2.2.3 Model Implementation in Modelica and Export 256
4.3 Design of the Ground Controller 258
4.3.1 Primary Objectives and derived Requirements 258
4.3.2 Structure of the Ground Controller 258
4.3.3 General Design Objectives 259
4.3.4 Modes of the Ground Controller 262
4.3.5 Development of the Taxi Test Software for
Online Gain Set Selection 268
4.3.6 Pre-Selection of Gain Sets 272
4.4 Real-Live On-Ground Testing For First Flight 276
4.4.1 Sensor Integration and Testing 277
4.4.1.1 Sensor Selection and Integration 277
4.4.1.2 Laser Leveling of Navigation System 277
4.4.1.3 Magnetometer Calibration 278
4.4.1.4 Towing Tests for Sensor Verification 279
4.4.2 Taxi Test Setup and Monitoring 284
4.4.3 External Pilot Controlled Taxi Tests and Landing Gear Model Validation 285
4.4.3.1 Overview and Conduction 285
4.4.3.2 The EP Controlled Taxi Test Maneuvers 287
4.4.4 Discoveries and Improvements 295
4.4.5 Online Gain Set Layout of the Ground Controller 297
4.4.5.1 Single Lane Change 298
4.4.5.2 Double Lane Change 298
4.4.5.3 Test of back-up law control structure 298

4.4.5.4 Straight Acceleration Run . . . 299
4.4.6 Verification and Validation for First Flight . . . 304
4.4.7 Final Taxi Tests at Ground Trial Facility . . . 304
4.4.7.1 Double Lane Change . . . 304
4.4.7.2 Straight Acceleration Run . . . 304
4.4.7.3 Full Flight Mission within Aircraft-in-the-Loop Simulation . . . 308
4.5 In-Flight Validation . . . 309
4.5.1 Pre-Flight Verification on-site . . . 309
4.5.2 First Flight Mission . . . 312
4.5.3 First Flight Monitoring . . . 312
4.5.4 Pre-Flight Checks For First Flight . . . 314
4.5.5 The Maiden Flight of the SAGITTA Demonstrator . . . 316
4.5.5.1 Automatic Take-Off Maneuver . . . 321
4.5.5.2 Automatic Landing Maneuver . . . 322
4.6 Summary . . . 328

5 Conclusion, Summary and Outlook 329

A Stability and Robustness I

A.1 MIMO Margins . . . I
A.1.1 Complementary Sensitivity Function . . . I
A.1.1.1 Derivation of transfer functions . . . I
A.1.1.2 Derivation of the MIMO phase margin of the complementary sensitivity transfer function . . . III
A.1.2 Sensitivity Function . . . V
A.1.2.1 Derivation of transfer function . . . V
A.1.2.2 Derivation of the MIMO Phase Margin of the sensitivity transfer function . . . V
A.1.3 Balanced Sensitivity Function . . . VII

B List of Scientific Publications **X**

C Acronyms **XII**

List of Figures **XXVII**

List of Tables **XXVII**

Bibliography **XXXVII**

Chapter 1

Introduction

Although drones were already built in the last century for military purposes, unmanned and optionally piloted aerial vehicles with new configurations have received significant attention in public at the beginning of the 21st century and become a growing industry. Significant progress in different technology areas enable nowadays the development of new types of aerial vehicles, which are utilized for many different applications in the private, business, public and other commercial sectors.
According to a study from the European Union (Ref. [1]), the market valuation for Unmanned Aerial Vehicles (UAVs) will exceed 10 billion Euro annually by 2035 and 15 billion Euro annually by 2050. It is forecasted, that the number of UAVs used for leisure activities will grow from 1 million to about 7 millions, and aerial vehicles for the business and public sector from thousands to hundreds of thousands.
The study in Ref. [2] predicts the market value for Urban Air Mobility (UAM) vehicles has the potential of 500 billion USD considering completely unconstrained scenarios and points out that 2.5 billion of that potential market value is realistic in the near term.
As a result, the new markets and the respective aerial vehicles demand novel flight control systems. This thesis contributes the following three flight control system enhancements for novel aerial vehicles of the evolving UAV and UAM market:

- Design, Development and Verification of a Three-Dimensional Geofencing Algorithm for Safety Enhancement
- Stability and Robustness Analysis for Flight Control Laws of new Vertical Take-off and Landing Aerial Vehicles
- Design, Development and Testing of Ground Control Functions of a novel diamond-shaped UAV

Before the presented enhancements are elaborated, the reasons and facts for the new aviation markets, the emergence of novel aerial vehicle configurations and their new regulations are elaborated.

1.1 The Recent Aerospace Evolution

1.1.1 New Markets

New technologies have driven engineers and businesses to design and develop UAVs for different applications, what has opened large new markets and industry segments. The UAV markets are usually subdivided based on the application of the UAV and can be structured into five groups: toys, hobby, professional, commercial and military (Ref. [3]). Professionals use their UAV, for example, for aerial filming and photography. In the commercial and industry sector the utilization of UAVs is manifold. According to Ref. [1], for example, UAVs open agriculture precision possibilities to increase the productivity and to leverage environmental-friendly methods. UAVs can also reduce the personal risks by performing dangerous activities, such as power lines maintenance and infrastructure inspections. The public security sector benefits from it by effective executions of monitoring and surveillance missions. Other industry branches are starting also to benefit from UAV advantages, for instance, for mining and construction and more.
It is predicted in Ref. [3] that the revenue for the hobby segment will be 5031.36 million USD in year 2022. Further, it is indicated that the commercial UAV market will be 5334.68 million USD worldwide in year 2022. This will be an increase of 380.3 % from the year 2018 to 2022.

Recently the public attention focuses on the urban air mobility market (UAM). A market study from Ref. [4] elaborates, that there are several UAM use cases, which are expected to become reality in future. One use case is the so-called last-mile delivery. In this scenario, small packages are delivered from a transportation hub to an arbitrary final destination within a distance of approximately 10 miles by smaller UAVs with payloads about 3 kg, which would be operated automatically or piloted manually and possibly beyond-visual-line of sight (BVLOS). The market study expects that after year 2030 this use case could be profitable, when the total number of delivery increases. The possibly most well-known use case is the *air taxi* scenario. The *air taxi* scenario is a ridesharing operation that enables customers to book vertical take-off and landing aircraft (VTOLs) to transport one to five passengers over a distance from 10 to 70 miles from a specified pick-up location to a desired destination. Other UAM use cases are *air shuttles* or *air ambulances.* (Ref. [2])

1.1.2 New Configurations

The technologies and the new markets nowadays allow well-established and newly-founded companies the development of novel aerial vehicle configurations, which differ substantially from aircraft or helicopter configurations designed in the last century of aviation. Major technology drivers lead to a high diversity of the new airspace participants in weight, propulsion,

configuration and speed. To give an impression of the aerial vehicle diversity, some aerial vehicles, which are on the market or flight demonstrators, are discussed next.
The *Nano Black Hornet* from the company *FLIR*, for example, weights about 18 g and can be used by fire brigades or the military as personal reconaissance system to improve the situational awareness (Ref. [5]). The *DJI* company, which is ranked on the 1st place as leading UAV company, sells, for example, among other quadcopters, the *DJI Phantom 4* with a take-weight of about 1.4 kg and a flight time of about 28 min. Looking at a higher weight class, the *Trinity F90* from the company *Quantum Systems GmbH* in Germany is a vertical-take-off-and-landing (VTOL) UAV with three tilting propellers and a maximum take-off weight of 5.0 kg. With transition vehicles the advantages of a helicopter and an aircraft are combined (Ref. [6]). The *Luna* from the company *EMT Ingenieurgesellschaft* in Germany is a fixed-wing UAV with a wingspan of about 4.1 m and a take-off weight of about 40 kg is launched by a bungee catapult and propelled by a pusher propeller (Ref. [7]). The flying demonstrator *SAGITTA*, whose development is also part of this thesis, is a fixed-wing UAV with novel diamond-shaped configuration, a take-off weight of about 150 kg and propelled with two gas turbines. In contrast to the UAV market, companies also design and develop currently *passenger aerial vehicles* or also called *air taxis*, which are mostly electrically powered and with VTOL capability. Thereby, prototypes from, for example, the Chinese company *eHang* or the german company *Volocopter GmbH*, rely on small electric multi-rotors. On the contrary, the American company *Joby Aviation* or the German company *Lilium GmbH* focus on passenger aerial vehicles with transition capabilities. Prototypes of several eVTOL aircraft are currently in flight testing with some companies even starting trial operations, such as Wisk in New Zealand (Ref. [8]). Others have begun flight testing in cooperation with certification authorities.

1.1.3 New Technologies

A combination of the new technologies presently opens up a completely new chapter in the history of aerospace. Major new technologies developed over the last thirty years now enable innovations for new aerial vehicles and new markets of UAVs. There are several major technology drivers, which contribute as the keys to success of UAV and eVTOL developments. To understand the reasons, an overview of important technology drivers is given next.
Micro-electro-mechanical sensors (MEMS) achieve nowadays sufficient performance with much less cost compared to optical or laser sensors. According to Ref. [9], MEMS sensors are more than ten times cheaper with smaller volume and less weight. As a result, the reduction in weight, volume and cost enables new possibilities in terms of unconventional configurations.

Another major driver is the progress made in the accuracy of position determination using global navigation satellite systems. Since 1995, several positioning improvement programs have advanced the accuracy of the American global positioning system (GPS) (Ref. [9]). The deactivation of Selective Availability (Ref. [10]) in the GPS system in May 2000 significantly improved the GPS applicability for aviation usage. According to Ref. [11], the accuracy improved 100 m vertically and 50 m horizontally. A report (Ref. [12]), created for the FAA in 2016, states the GPS system has meanwhile an 95 % error accuracy of 3.8 meters vertically and 1.8 m horizontally.
Beside the GPS system, other countries have launched their own Global Navigation Satellite System (GNSS) program. The Russian GLONASS, the Chinese BeiDou and the European Galileo system enable nowadays new possibilities for aerospace innovations and lead to higher reliability and availability for GNSS systems and position determination on earth in general. Augmentation systems, such as Satellite Based Augmentation Systems (SBAS) and Differential GPS (DGPS), enhance even further the accuracy of these technologies.
In addition to the MEMS and GNSS improvements, the computational power of microprocessors increased in an unprecedented scale in the last thirty years. Clock speed, number of transistors and the data width of the CPU are just a few properties, which underline the developments in this domain. Since 2004, *Intel* and other companies have even introduced microprocessors with multiple cores and millions more transistors. Released in 2000, the *Intel Pentium 4* processor consists of 42 million transistors, achieving a clock-speed of *1.5* GHz. In contrast, the dual-core GT2 Skylake processor, introduced in 2015, has already *1750* million transistor with a clock rate of up to 4.5 GHz (see e.g. Ref. [13]). The new computational power enables new methodologies within the design and development, and also during the verification and testing phases of new aircraft. In flight, the advanced computational resources enable the execution of complex and advanced software functionalities, which can be executed in real-time.
The energy storage optimization is another important technology step for the growing UAV market. Within the last thirty years, the battery technology industry was able to increase the energy density per weight significantly. The major focus lies thereby on Lithium-ion batteries as power or energy source. Comparing the year 1990 to 2015, the specific energy of the Lithium-ion batteries could be tripled to more than 300 Wh/kg using specific Lithium-ion materials (Ref. [14]). The increased demand in aerospace and the automobile industry will improve the energy storage capacity even further.
The drastic growth of (AC) permanent magnet motors open the dreams for larger electrical (unmanned) aerial vehicle in future. For small airplanes (AC) permanent magnet motors with a power of more than 100 kW already exist. Thereby, they can reach a specific power more than 2 KW/kg (Ref. [15]).

Other key drivers for the increasing UAV market are the improvements made in the communication and information technology, the progress in machine learning and artificial intelligence and advanced available camera technology. As a consequence, these new technologies enable the development of unprecedented configurations and open up markets for aerial vehicles with new purposes and applications.

1.1.4 New Regulations

Due to the growing market, the number of UAVs within the airspace increases continuously. The variety of applications and configurations leads to a high diversity within the current and future airspace. In order to regulate the new participants, the European Union Safety Agency (EASA) and Federal Aviation Administration (FAA) are currently drafting new regulations for UAVs. Therefore, the EASA introduces three categories of unmanned aerial system (UAS) operation on a risk- and performance based approach and Ref. [16, p.14] defines the three categories as follows:

- *the **open** category is a category of UAS operation that, considering the risks involved, does not require a prior authorisation by the competent authority nor a declaration by the UAS operator before the operation takes place;*
- *the **specific** category is a category of UAS operation that, considering the risks involved, requires an authorisation by the competent authority before the operation takes place, taking into account the mitigation measures identified in an operational risk assessment, except for certain standard scenarios where a declaration by the operator is sufficient or when the operator holds a light UAS operator certificate (LUC) with the appropriate privileges*
- *the **certified** category is a category of UAS operation that, considering the risks involved, requires the certification of the UAS, a licensed remote pilot and an operator approved by the component authority, in order to ensure an appropriate level of safety*

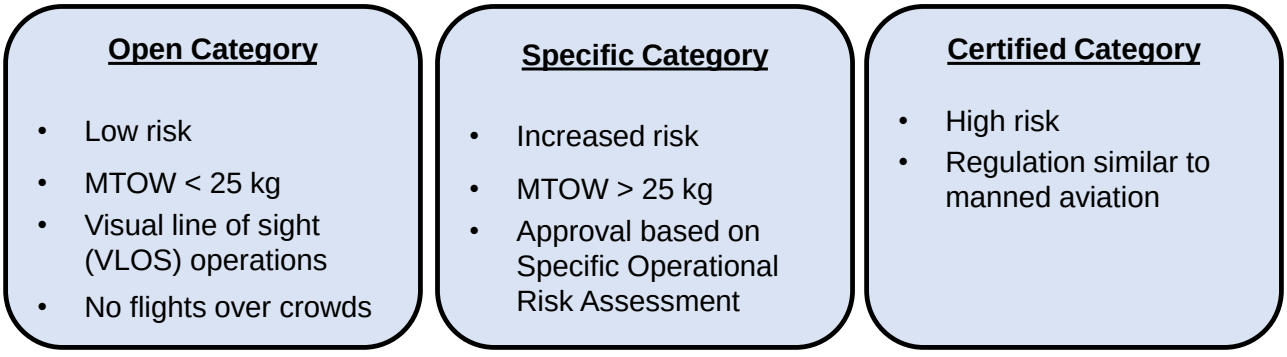


Figure 1.1: Categories of UAS operations (based on Ref. [16])

It is distinguished between the term UAV and UAS. An UAS consists of the UAV, the flying part, a command and control station, at least one data link and other components, which are necessary for the operation. (Ref. [17])
The Notice of Proposed Amendment (Ref. [16]) from the EASA specifies further the definition of the three categories. Important specifications are summarized in Fig. 1.1.
Based on the markets' growth, EASA prioritizes currently the elaboration of the regulation framework for the 'open' and 'specific' category. The regulation for the 'certified' category is currently in progress.
If the UAV operation *"poses more significant aviation risks to persons overflown or which involves sharing the airspace with manned aviation"* (Ref. [17, p.26]), the operation is assigned to the *specific* category. In October 2019, EASA has published the Guidance Material (GuM) and Acceptable Means of Compliance (AMC) for the open and the specific category.
To evaluate the risk of UAS operation for the specific category, the so-called Specific Operation Risk Assessment (SORA) should be conducted. The SORA is a methodology developed by the Joint Authorities for Rulemaking of Unmanned Systems (JARUS). According to the AMC and GuM material, the SORA can be used as *"an acceptable means to demonstrate compliance with Article 11 of the UAS Regulation, that is to evaluate the risks and determine acceptability of a proposed operation of an UAS within the 'specific' category."* (Ref. [18, p.11])
The SORA proposes a methodology to gain solid confidence that a specific UAV mission can be conducted safely. The SORA process comprises ten steps to assess the risk of the UAS operation. Since risk mitigation strategies are relevant items within the SORA, the ten steps are summarized shortly in the following.
First, the UAS operator details *"technical, operational and system relevant information"*, which is necessary to evaluate the risk of the UAS mission. This document is called the *Concept of Operation (ConOps)*. Subsequently, the initial ground and air risk classes are specified depending on the operational scenarios. The ground risk class specifies *"the risk of a person being struck by the UAS ..."* (Ref. [19, p.19]). In contrast, the air risk class details *"the risk of mid-air collisions"* (Ref. [19, p.22]). Both classes can be reduced by risk mitigation strategies. The ground risk class is reduced, for example, by the integration of a functional parachute system or an available emergency response plan.
Strategic and tactical mitigations can decrease the air risk class. The strategic mitigations comprise the restriction of the UAS' operational volume and the adherence to common structures and rules within a certain class of airspace (Ref. [20]). Tactical mitigations shall reduce the residual risk of mid-air collisions, for instance, by *detect and avoid (DAA)* or geofencing algorithm (Ref. [21]).

The final ground and air risk classes determine the Specific Assurance and Integrity Levels (SAIL), which *"represents the level of confidence that the UAS operation will stay under control"*. (Ref. [19, p.26])
The SAIL is eventually used to determine the *Operational Safety Objectives (OSO)* and their respective level of robustness. The OSO provided in the SORA document are objectives selected by many experts and are grouped in different categories based on their threat (e.g. technical issue with the UAS, human error). The level of robustness of the corresponding OSO is achieved by fullfilling the respective level of integrity and level of assurance. These levels are explained in the AMC and GuM of the SORA. Equivalently to the OSOs, the level of robustness applies also for the mitigation strategies, such as a geofencing algorithm.
According to the AMC and GuM (Issue 1), such algorithms *"refer to possible future applications of automated traffic management systems for unmanned aircraft ..."* and *"may not exist yet"* (Ref. [18, p.74]). If the applicant can provide confidence that the safety objectives and the mitigations are considered, the UAS operation is approved according to the SORA.
Thus, the SORA shifts the certification from compliance-based to a risk-based assessment. As mentioned, the drafting of regulations for the Certified category is still under development. Therefore *Special Conditions for Light Unmanned Aircraft Systems* (Ref. [22]) have been published by the EASA, which do not mainly fall into the specific category.

Further new regulations for new types of vehicles, such as the *air taxis*, are currently drafted. The *Special Condition for small-category VTOL aircraft* (Ref. [23]) adresses *"vertical take-off and landing (VTOL) aircraft, which differ from conventional rotorcraft or fixed-wing aircraft"* (Ref. [23, p.1]). Therefore, *"a complete set of dedicated technical specifications in the form of a special condition for VTOL aircraft has been developed"* (Ref. [23, p.1]) from the EASA. However, e.g. required Handling Qualities, are no longer specified by specific methods, and *"... each applicant can choose the methodology to determine the HQ"* (Ref. [24, p.10]).

In consequence, there is a high need for new verification and safety functions for the certification of new participants in the airspace with respect to SORA, but also other new certification standards.

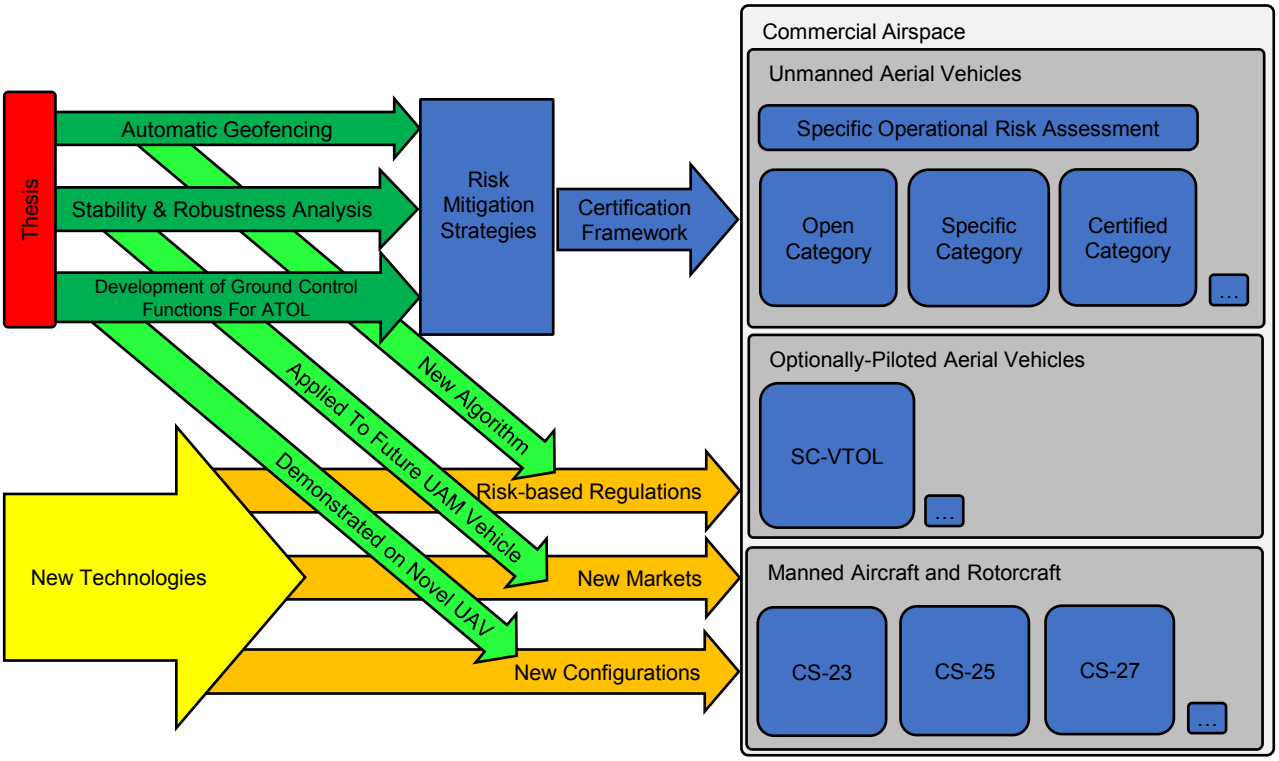


Figure 1.2: Thesis contributions in context of new technologies, regulations and commercial airspace

1.2 Objectives, State-of-the-Art and Contributions

As a result, the aerospace evolution demands new flight control systems for the presented new aerial vehicles. This includes, with regard to this thesis, to design new algorithmic approaches for safety augmentation, to find new methodologies for the evaluation of the robustness of flight control laws and to develop ground control system functions for automatic take-off and landing.

Thus, this thesis elaborates three independent topics, which contribute to the successful development and safe operation of future aerial vehicles in the new markets. It is of particular note, that the three contributions are applied to three different aerial vehicle configurations.

Fig. 1.2 illustrates the thesis contributions in context of the explained new technologies, regulations and configurations. First, the thesis provides a new risk mitigation strategy to restrict the operational volume for safety enhancement. This is beneficial for the compliance of the new risk-based regulations. Secondly, appropriate and new stability and robustness measures are applied to a future UAM vehicle, which can be used as cutting-edge for the new market. Eventually, the design, development and testing of ground control functions for a novel UAV configuration are demonstrated. The development is validated in a successful first and second flight.

In the following, the three contributions of this work are introduced.

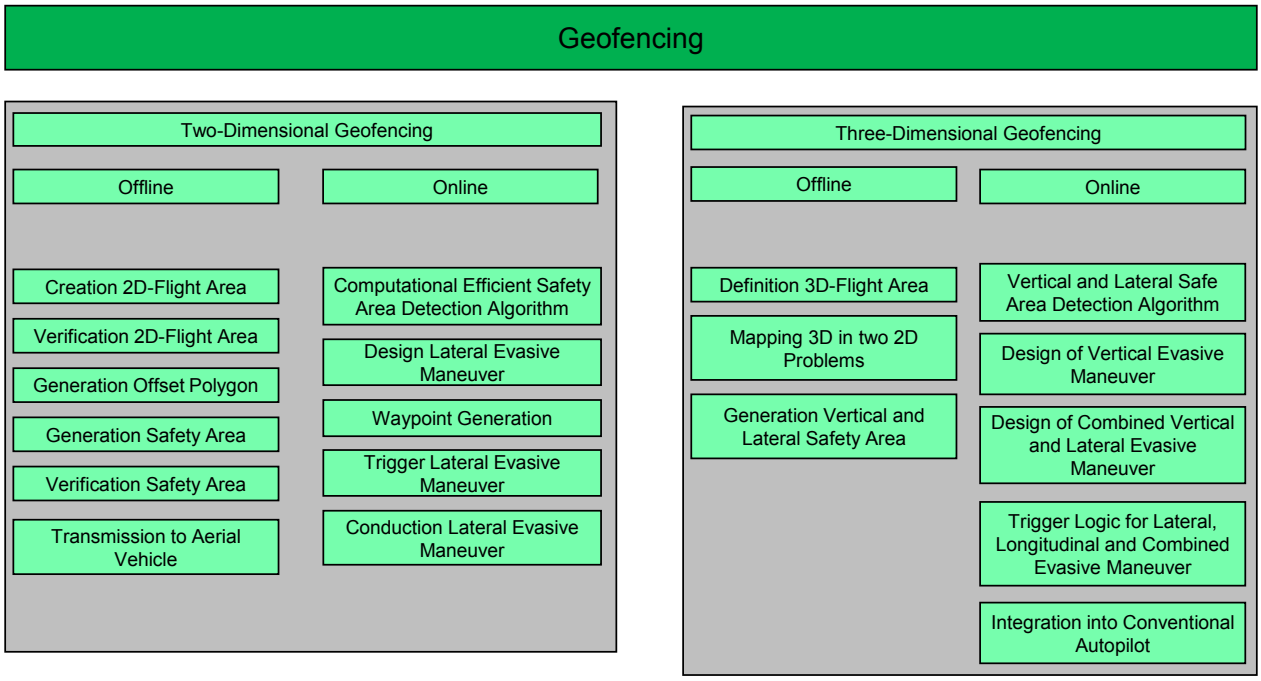


Figure 1.3: Overview and structure of the geofencing algorithm

1.2.1 Automatic Geofencing For Safety Enhancement of Aerial Vehicle Operation

Objectives:

The global market for UAVs is expected to grow significantly in the near future. As already mentioned, drones are utilized, for instance, for pipeline, power or infrastructure inspection, precision agriculture, highway monitoring or surveillance (Ref. [25]). For such applications, the UAVs are operated beyond visual line of sight. To avoid any hazardous or catastrophic events, the UAV needs to stay within a predefined flight zone in national airspaces. In order to simplify and generalize the permission processes, national and international aviation authorities currently formulate and draft regulations for remotely piloted aircraft systems. In many proposals (cf. Ref. [25, 19]), risk mitigation strategies are suggested to reduce the likelihood of hazards by appropriate and robust threat and harm barriers. One risk mitigation strategy is the limitation of the aircraft's operational space by reliable geofencing functionalities (Ref. [19]). Geofencing denotes an automatic function, which prevents the UAV to penetrate predefined flight areas.
Therefore, one objective of this thesis is the design, development and implementation of an automatic geofencing algorithm. The algorithmic approach shall comprise a safe area detection algorithm, which identifies the leaving of the flight area automatically and can trigger the termination system. However, beyond the safe area detection, the algorithm shall plan and conduct evasive maneuvers to prevent a leaving of the predefined airspace in three-

dimensioal space for wingborne flight. The algorithm shall be capable to handle arbitrary geometrical flight areas with non-flight zones, which fulfill defined requirements. The geofencing functionality enhances the safety capabilities of the aerial vehicle and reduces the probability of risk on-ground and in-air.

State-of-the-art

Industry: The company *DJI* already provides geofencing capabilities in the flight control systems of their unmanned air vehicles. According to e.g Ref.[26], the *DJI* vehicles have three flight modes: P-mode (Positioning), S-mode (Sport), and A-mode (Attitude). In P-mode, the vehicle *"utilizes the GPS and Obstacle Sensing System to automatically stabilize itself, navigate between obstacles or track a moving object"* (Ref. [26, p.11]). The A-mode is active, *"when neither the GPS nor the Obstacle Sensing System is active, the aircraft will use only its barometer for positioning to control the altitude"* (Ref. [26, p.11]). Geofence system in P-Mode enables flying in a maximum geofence (within the range of 20 m to 500 m) and a maximum radius geofence (within the range of 15 m to 500 m). If the vehicle approaches non-flight zones, the air vehicle stops before entering the boundary. The defined non-flight zones include restricted areas such as borders, government property and airports.
If the system is flown in A-mode(Attitude), the GPS system is not active, so the non-fly zones are not enforced. Ref. [26] provides no geofencing information to the S-mode.
DJI maintains the lists of non-flight zones, but the lists are strictly advisory and are not complete (Ref. [27]). At the time of the composition of this thesis, *DJI* sells only aerial vehicles, which produce their main lift by propellers.

Academia: In academia, some work with regard to geofencing capabilities has been published. Ref. [27] analyzes a geofencing system with independence from the autopilot system of small multicopters and suggests therefore respective different guidance modes to ensure the UAV is kept within a pre-defined test range. Ref. [28] proposes a safe area detection algorithm considering polygons and circles within ground control stations and implements the algorithm under the consideration of a graphical user interface. The work in Ref. [29] describes the scaling for geofence boundaries in order to gain the minimum distance from the original, predefined flight area. Thereby, new boundaries are generated by proposed scaling algorithms of polygons. The work concentrates on horizontal plane constraints. In Refs. [30, 31] a computationally efficient geofence boundary violation detection method is presented for three-dimensional space utilizing the Triangle Weight Characterization with Adjacency algorithm. It can be applied to polyhedrons, which are formed by extruding horizontal plane polygons vertically.

Contribution I:
Development of an Automatic Geofencing Algorithm For Three-Dimensional Space

This contribution comprises the design and development of a new algorithmic approach for automatic geofencing. The presented algorithm prevents the aerial vehicle from leaving an arbitrary two- or three-dimensional flight area in wingborne flight. Subsequently, the different elements of the algorithm are highlighted and contrasted to presented state-of-the-art:

- PROPER DEFINITION AND VERIFICATION OF ANY ARBITRARY SELECTED FLIGHT AREA PRIOR TO FLIGHT: The proposed algorithm is based on a suggested flight area definition, which specifies the allowed set of flight areas in a mathematical manner. To guarantee the geofencing functionality reliably the application of a verification algorithm is proposed, which checks if the specified requirements are met. To the author's knowledge, such a generic and consistent definition and verification hasn't been published yet.

- DEFINITION AND GENERATION OF ADEQUATE HORIZONTAL AND VERTICAL SAFETY MARGINS OF THE TWO- AND THREE-DIMENSIONAL FLIGHT AREAS: To assure the aerial vehicle does not leave the flight area in any case, safety areas are generated by the utilization of Voronoi-diagrams, which enable the generation of offset-polygons for arbitrary valid flight areas. The newly generated offset-polygons in addition are adapted to gain the final safety area such that the flight operator is restricted as little as possible in terms of his selected flight area. The safety area is generated by the consideration of the aerial vehicle's vertical and lateral turn radii in nominal conditions. In contrast to Ref. [29], the voronoi approach is chosen, which enables the generation of offset polygons within a generic approach for arbitrary flight areas. In addition, the approach is extended to three-dimensional space, which hasn't been published before.

- DEVELOPMENT OF AN EFFICIENT AND RELIABLE SAFE AREA DETECTION ALGORITHM IN TWO- AND THREE DIMENSIONAL SPACE: The proposed concept separates the automatic geofencing algorithm into an offline and an online part to reduce the computational effort during flight to a minimum. The offline part deals with computational costly tasks, which can be conducted before flight. This encompasses inter alia the generation of the voronoi-diagrams, the creation of the safety area, and the verification of the selected areas. The online part deals only with the determination of the current area, the aerial vehicle is located in, as well as the trigger and conduction of the evasive maneuver. In addition, in case of three-dimensional geofencing, the interpolation between areas at different altitudes is necessary. In contrast to Ref. [30, 31], the proposed algorithm is able to handle any polyhedrons, which do not require necessarily horizontal ceilings and basis. Further, the specific separation between offline and online tasks is novel within this context.

- DESIGN OF EVASIVE MANEUVERS TO AVOID THE LEAVING OF THE FLIGHT AREA BY CONSIDERING AIRCRAFT LIMITATIONS: In contrast to the already existing geofencing capabilities in the industry and academia, the proposed geofencing algorithm is mainly designed for wingborne flight, in which the respective vehicle gains the main lift by aerodynamic forces. Therefore, the evasive maneuvers are designed under the consideration of the aerial vehicle limitations. In this regard, it should be noted that lateral, longitudinal and simultaneous lateral and longitudinal evasive maneuvers are designed within this thesis under consideration of aircraft limitations for nominal conditions without wind. The designed maneuvers are conducted, such that the aircraft is prevented to leave the defined three-dimensional flight area. To the author's knowledge, such maneuvers have been never before proposed with regard to geofencing functions.
- VERIFICATION OF THE GEOFENCING ALGORITHM IN NONLINEAR SIMULATIONS: The algorithm is embedded in a flight control system and demonstrated within a high-fidelity simulation of a general aviation fly-by-wire demonstrator in order to show the functionality of the geofencing algorithm in a simulated real-world environment.

The structure and the content of the algorithm is illustrated as an overview in Fig. 1.3. It is differentiated between geofencing in two-dimensional and three-dimensional space and distinguished between required offline and online functions. Offline functions are executed before flight and online functions are executed on-board during flight. As above elaborated and to the author's knowledge, the presented, complete geofencing algorithm, from the definition of the flight area until the design of the evasive maneuvers in three-dimensional space in wingborne flight and its verification in simulations, have never been proposed before (except the respective author's publications of this thesis).

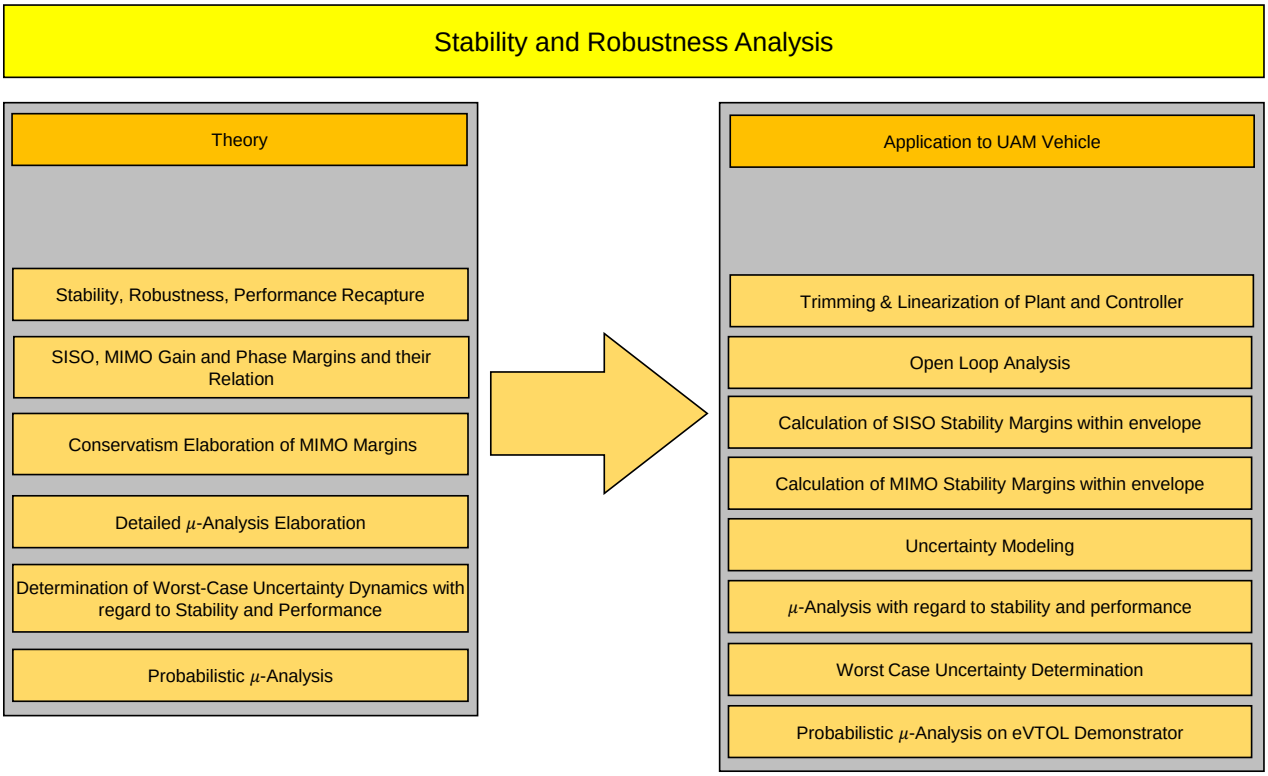


Figure 1.4: Overview and structure of the stability and robustness analysis for an UAM vehicle

1.2.2 Stability and Robustness Analysis of Control Laws for VTOL Aircraft

Objectives

Flight control laws have to fulfill defined stability and robustness requirements for first flight and for further certification processes. These requirements ensure robustness margins for the flight controller, if deviations between the utilized, but probably uncertain closed-loop models and the reality occur. Especially the new VTOL capable aerial vehicle configurations, which are pushed on the market, demand new approaches in terms of stability and robustness assessment.

In consequence, this thesis shall elaborate appropriate robustness measures for new vehicle configurations. Therefore, this thesis details, beside the Single-Input-and-Single-Output (SISO) gain and phase margins and Nichols diamonds, Multiple-Input-Multiple-Output (MIMO) stability and robustness margins, also with regard to their conservatism. Specifically for new configurations with several redundant control effectors, robustness margins need to be considered, which take multiple channels simultaneously into account. Since MIMO margins can lead to conservative robustness measures, the μ-analysis approach shall be detailed to get robustness margins with lower conservatism by considering modeled physical uncer-

tainties. Thereby, the worst-case uncertainties and uncertainty dynamics for guaranteed handling qualities shall be calculated and determined. In addition, the μ-analysis shall be extended to a probabilistic μ-approach which takes probabilistic uncertainties into account and eventually can give a guaranteed statement, what the probability is that the considered aerial vehicle is stable.
On top, this thesis shall apply the elaborated and new introduced robustness methodology approaches to a critical nonlinear flight control law of a novel vertical take-off and landing aerial vehicle. Thereby, the controller comprises the incremental nonlinear dynamic inversion control approach. The focus does not lie on the controller design, but on the assessment of its stability and robustness.

State-of-the-art

"The application of automatic control systems to aircraft promises to bring about the most important new advances in aeronautics in the future" (Ref. [32], 1951).
Stability and control has been always a major challenge in the design process of flight controls laws. Classical flight control laws were designed mainly by single-input-single-output frequency response and root locus techniques, going back to the approaches from in W. Bollay, 1951 (Ref. [33]). Since the 1970s, multivariable control synthesis and analysis techniques became popular (Ref. [34]) and were further developed in the 1980s and 1990s (e.g. eigenstructure assignment, dynamic inversion, μ-synthesis). Nowadays, a high variety of different control design strategies were already applied to different types of aerial vehicles. From linear and nonlinear (see e.g Ref. [35]) to robust or adaptive controllers (see e.g. Ref. [36]), various different design approaches exist. A good outline of control design strategies is given in Ref. [37].
In order to give an overview what measures are used for robustness assessment, a few reference examples are listed in the following, in which the robustness of aerial vehicles with different applied control strategies was assessed.
Ref. [38] used eigenstructure assignment (EA) to design a robust controller for the lateral dynamics of a hypersonic aerial vehicle. The robustness is mainly evaluated by Nichols plots and classical μ-analysis. In Ref. [34], the linear quadratic regulator (LQR) procedure was applied to the control law of the Boeing 767 autopilot. Further, in the same paper, MIMO margins are calculated for a fighter jet as robustness measure. Ref. [39] applied EA to the lateral flight controller of diamond-shaped UAV and assessed the robustness also by the calculation of SISO margins and Nichols plots. Recently, in Ref. [40], a PI-Algorithm design procedure is applied to a VTOL aerial vehicle. The controller robustness is evaluated as well by Nichols plots, underlined by the application of the classical μ-analysis considering uncertainties only at the plant input. Ref. [41] applied the incremental nonlinear dynamic inversion procedure to a multicopter and evaluated the robustness by parameter variation

within specifc intervalls in nonlinear simulations, which is feasible for a few parameter, but can become costly for high-dimensional uncertainty problems and cannot guarantee the worst-case.

Although a large diversity of control strategies exist, the number of accepted and respected methodologies for robustness assessment are still limited. As the mentioned examples show, the commonly used robustness methodologies are the SISO margins and the Nichols plot considering a linear closed-loop system. In addition to the SISO robustness measures, MIMO margins of the sensitivity transfer functions are suggested in Ref. [42] to consider cross-couplings between different channels. Ref. [43] further introduced the MIMO margins of the balanced sensitivity transfer function and Ref. [44] generalized these margins towards disk margins.

Refs. [45, 46, 47] introduced the structured singular value μ for a more detailed and less conservative robustness assessment.

Lately, Ref. [48] suggests to use the time-delay margin as robustness measure, in order to be able to assess also the robustness of adaptive controller. In contrast to the mentioned deterministic approaches above, probabilistic approaches become more and more popular over the last years. Ref. [49] compares two different methodologies in probabilistic robustness analysis. There are probabilistic sampling techniques to compute the probability for stability by sampling over the whole uncertainty space. The drawbacks of these methods are that they require high number of samples if the system becomes higher-dimensional to approximate the probability precisely and that they compute a soft bound on the probability for stability only. Worst-case methods such as the classical μ-analysis find the worst-case uncertainties which bring the systems towards instability within a defined and bounded uncertainty space. These techniques can give a hard bound on stability, but are deterministic and rather conservative. Ref. [50] made one of the first attempts to extend the μ-approach towards probabilistic-μ using a branch and bound algorithm by the consideration of uniform distributed uncertainties. Recently, Ref. [51] and Ref. [52] proposed a probabilistic-μ approach with truncated normally-distributed uncertainties, which applies a dichotomic algorithm for automatic uncertainty space exploration and classification using the principle of branch-and-bound techniques. μ is calculated for specific uncertainty sets. If μ is greater than one for the set, the set is divided into two parts and μ is calculated again for the two separated uncertainty sets. If μ is smaller than one, the respective set is considered for the probability calculation. The procedure is continued until a certain area size is reached. The probability of all accepted stable uncertainty sets is eventually accumulated. This approach can be computational costly in higher-dimensional uncertainty space, since the number of areas to be analyzed increases exponentially by each dimension.[2]

[2] This paragraph with regard to the state-of-the-art of the probabilistic approaches has been published recently in Ref. [53] as part of the introduction by the author.

Contributions II:
Novel Stability & Robustness Analysis for Future UAM

This thesis elaborates robustness metrics for new types of aerial vehicle and applies the introduced robustness measures on a novel VTOL demonstrator. The contributions are parted in the following segments:

1. DETAILED ELABORATION OF DIFFERENT SISO AND MIMO ROBUSTNESS MARGINS AND THEIR RELATION WITH RESPECT TO CONSERVATISM: The Single-Input-and-Single-Output (SISO), different Multiple-Input-Multiple-Output (MIMO) gain and phase margins and the classical μ-analysis are derived in a consistent manner. The focus lies thereby on the meaning and the physical interpretation of the margins with respect to uncertainties such that they can be applied on new types of aerial vehicles. For that reason, the different modeling approaches for real and complex uncertainties are detailed as well. In addition, the relation between the introduced margins and their corresponding relevance is analyzed and discussed.

2. CALCULATION OF REAL AND COMPLEX WORST-CASE UNCERTAINTIES AND THEIR RESPECTIVE TRANSFER FUNCTION FOR GUARANTEED STABILITY AND GUARANTEED HANDLING QUALITIES BY UTILIZATION OF μ-ANALYSIS METHODS: A new approach is introduced to compute the respective transfer function of worst-case uncertainties, which bring the closed-loop system towards the specified stability or handling quality borderline. The derived transfer functions underline the existence of the calculated worst-case uncertainties and represent uncertainty dynamics which drive the system to the respective borderline. This is demonstrated on a real-live eVTOL demonstrator.

3. INTRODUCTION OF NOVEL PROBABILISTIC-μ APPROACH: A new probabilistic-μ approach is proposed, which utilizes probabilistic uncertainties instead of limited uncertainty intervals, to compute a lower bound on the probability, that the closed-loop system stays stable. The new μ-approach solves thereby a bi-level optimization problem, in which the probability for stability is maximized. In contrast to existing probabilistic μ-approaches, the new approach can take probability distributions into account, which are not truncated, and sets up a bi-level optimization problem, instead of a costly branch-and-bound algorithm.

4. APPLICATION OF PRESENTED ROBUSTNESS METRICS AND μ-ANALYSIS TO A REAL CLOSED-LOOP SYSTEM CONSISTING OF NONLINEAR PLANT DYNAMICS OF AN EVTOL DEMONSTRATOR AND AN INCREMENTAL NONLINEAR DYNAMIC INVERSION FLIGHT CONTROLLER: The introduced SISO and MIMO gain and phase margins are applied on an eVTOL demonstrator, which is controlled by an incremental nonlinear dynamic inversion controller. Furthermore, μ-analysis is conducted for the real closed-loop system

under consideration of detailed uncertainty modeling including real and complex uncertainties. In this regard, to the author's knowledge, SISO and MIMO margins and the μ-analysis have been never before applied to such a system.

5. APPLICATION OF NOVEL PROBABILISTIC-μ APPROACH TO EVTOL DEMONSTRATOR: Finally, the proposed probabilistic-μ approach is applied to the closed-loop system of the new eVTOL demonstrator. The real-live application example considers thereby probabilistic uncertainties with Gaussian distributions in different linear systems.

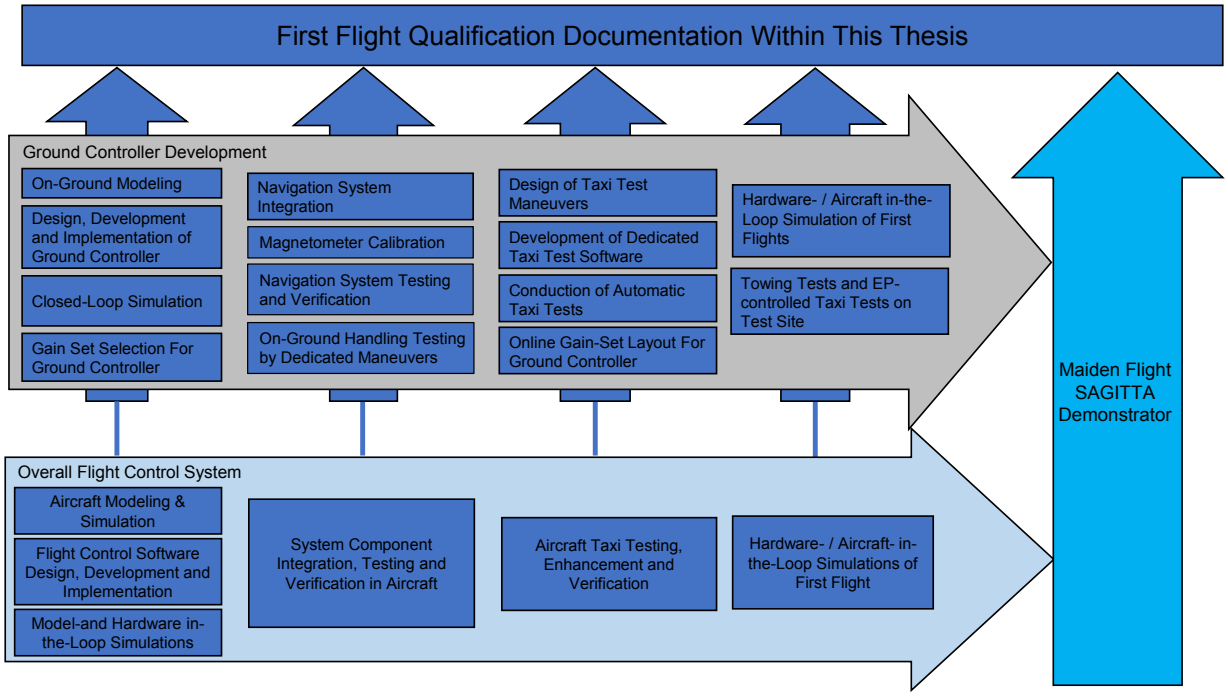


Figure 1.5: Comprehensive overview of the necessary steps for first flight qualification

1.2.3 Design, Development and Testing of Automatic Ground Control Functions of a Novel UAV for Automatic Take-Off and Landing

Objectives

The third objective is the first flight qualification of automatic ground control functions of a novel diamond-shaped UAV with about 150 kg weight for automatic take-off and landing. The objective is to show the entire development of required ground control functions within the flight control system of the novel UAV SAGITTA from the initiation until first flight. Thereby, the respective UAV shall be operated fully-automatically including take-off and landing, also at the first flight.

The focus lies on the on-ground modeling of the UAV, the design and development of the ground control functions, the integration and testing of the navigation system and the planning and the conduction of several (automatic) taxi tests. The approach is underlined by presenting taxi test results and flight test data of the successful first flight. The development of other flight control software modules is not part of this thesis. However, these modules are explained and referenced at the appropriate sections.

State-of-the-art

Only few work has been published about flight control systems of novel UAVs with automatic take-off and landing capabilities, which fly fully-automatically. Some UAV were/are in development, mainly in the military domain and denoted as unmanned combat aerial vehicles (UCAVs), which fly fully-automatically including take-off and landing. In Ref. [54] and Ref. [39], the most important UCAVs are summarized. For example, the Dassault nEUROn UAV, which was developed under lead of Dassault Aviation, took its successful first flight in December 2013. Another famous example is the Northrop Grumman X-47B, which performed a successful first-flight in February 2011 and has demonstrated several land- and carrier based landings (Ref. [55]). In general, these type of UAVs can be categorized as blended-wing body configuration with λ-wing (Ref. [39, p.2]) and a take-off mass of several tons. Information of these projects is limited, especially of their flight control system, due to their military nature. Beside UCAVs, as elaborated in Ref. [54, p.10], only few civilian UAVs with fully-automatic take-off and landing capabilities have been demonstrated in real flight.
However, work with respect to the development of the flight control system of the SAGITTA demonstrator has been already published. The development of the lateral flight control law is released in Ref. [39]. The design of its automatic take-off and landing maneuvers is detailed in Ref. [54] and the robust system automation of the demonstrator is elaborated in Ref. [56].
With regard to the development of the ground control functions, some work has been published about aircraft on-ground modeling, aircraft ground testing and low-speed taxiing.
In the automotive industry a lot of research is conducted to model the vehicle ground dynamics accurately. The driving dynamics can be modeled in different levels of fidelity. Ref. [57] explains many possibilities of vehicle on-ground modeling from the automotive perspective in detail. A simple on-ground aircraft model for ground controller design in roll-out and take-off phases between 40 and 100 knots is discussed in Ref. [58]. A gain-scheduled on-ground controller is designed in Ref. [59] for low-speed on-ground maneuvering, but only tested in simulation.
In terms of on-ground testing, the thesis from Ref. [60] gives a comprehensive overview about practical steps to follow until flight testing of UAVs can be conducted. Amongst others, this includes simulation, system integration test, data link tests, navigation control ground testing and weight and balance testing. On-ground taxi tests are not included.
In consequence, to the author's knowledge, no work has been published (beside the publications, which are part of this thesis), which describe the entire development, from the on-ground modeling, the design and the development of the ground controller, until the on-ground testing and first flight verification for a novel diamond-shaped UAV in detail.

Contributions III:
Design, Development and Testing of Automatic Ground Control Functions of a Novel UAV for Automatic Take-Off and Landing

However, the present work focuses on the necessary development steps for the ground control functions of the SAGITTA flight control system including the entire on-ground testing of the flight control system. Based on the described objective and state-of-the-art, Contributions III comprises the following major parts:

- GROUND CONTROL SYSTEM DESIGN FOR AUTOMATIC CENTERLINE TRACKING AND RUNWAY ALIGNMENT: A ground control system design for a novel diamond-shaped UAV demonstrator is proposed. The ground control system consists of a ground controller for centerline tracking and runway alignment, an automatic braking system and a direct law for remote pilots for manually controlled taxiing. A new aero-crossfeed command to the split-flaps is designed to support yaw control within the inner loop control law. The ground controller is active for automatic take-off until the aircraft is airborne and activated as soon as the weight-on-wheels sensors detected on-ground of the aircraft.
- INTEGRATION OF SENSOR COMPONENTS OF THE FLIGHT CONTROL SYSTEM AND PLANNING AND CONDUCTION OF TOWING AND TAXI TESTS INCLUDING ONLINE GAIN SET LAYOUT OF THE GROUND CONTROLLER: Necessary steps for the integration and verification of different sensor components are documented, evaluated and demonstrated on the mentioned novel diamond-shaped UAV in on-ground and towing tests. For the development and the verification of the ground controller, several taxi tests are planned, conducted and evaluated. Thereby, an online gain set layout of the proposed ground controller is introduced and demonstrated in real-live by dedicated taxi tests. The taxi tests encompass remote pilot controlled taxi tests and automatic taxi tests with different excitation maneuvers and at step-wise increased velocities.
- CONDUCTION OF PRE-FLIGHT TESTS ON-SITE AND IN-FLIGHT VALIDATION: The ground controller is verified in the first and second flight of the SAGITTA demonstrator during automatic take-off and landing maneuvers. The corresponding flight test results demonstrate the functionality of the proposed ground controller and highlight the applied strategies for first flight qualification within this work and other project publications.

The necessary steps for a successful first flight qualification of the ground control functions and thus the content of these contributions are shown comprehensively in Fig. 1.5.

1.3 Thesis Composition

The main chapters of this thesis are:

- Design, development and verification of a three-dimensional safe area detection and avoidance algorithm
- Stability and robustness analysis for the critical flight control software of a vertical take-off and landing aerial vehicle
- Design, development and testing of automatic ground control functions of a novel UAV for automatic take-off and landing

In summary, the proposed algorithm in Chapter 2 increases the safety of any aerial vehicle by a positional containment in three-dimensional space. In Chapter 3, the stability and robustness analysis for a vertical take-off and landing vehicle with novel configuration is elaborated. Chapter 4 shares the ground control function development of a new diamond-shaped UAV for automatic take-off and landing.

Chapter 2

Automatic Geofencing For Safety Enhancement of Unmanned Aerial Vehicle Operation

The global market for UAVs is expected to grow significantly in the near future. Currently, drones are utilized, for instance, for pipeline, power or infrastructure inspection, precision agriculture, highway monitoring or surveillance (Ref. [25]). For such applications, the UAVs are operated beyond visual line of sight. To avoid any hazardous or catastrophic events, the UAV needs to stay within a predefined flight zone in national airspaces. In order to simplify and generalize the permission processes, national and international aviation authorities currently formulate and draft regulations for remotely piloted aircraft systems. In many proposals (cf. Refs. [25, 19]), risk mitigation strategies are suggested to reduce the likelihood of hazards by appropriate and robust threat and harm barriers. One risk mitigation strategy is the limitation of the aircraft's operational space by reliable geofencing functionalities (Ref. [19]). Geofencing denotes an automatic function, which prevents the UAV to penetrate predefined flight area boundaries.

This chapter focuses on the development of the basic functions for a fully functional geofencing algorithm that is required for the flight control system. The presented chapter does not include the human-machine interface for the pilot to select the flight area, neither the development of respective flight control laws or system automation functions. However, the chapter concentrates on the fundamental geofencing functions to avoid the penetration of specified flight area boundaries and applies them exemplary to an assumed flight control system. This flight control system comprises classical flight control functions, such as a conventional autopilot and waypoint controller. For that reason, the presented geofencing algorithm can be embedded to conventional flight control systems.

The main contributions of this chapter are as follows:

- Proper definition and verification of any arbitrary selected flight area prior to flight
- Definition and generation of adequate horizontal and vertical safety margins of the two- and three-dimensional flight areas
- Development of an efficient and reliable safe area detection algorithm in two- and three dimensional space
- Design of evasive maneuvers to avoid the leaving of the flight area by considering aircraft limitations
- Verification of the geofencing algorithm in simulation

In order to reduce efforts and cost for possible certification processes of the risk mitigation strategy, the proposed algorithm shall feature low complexity, but fulfill the required task with highest safety and reliability. Consequently, the algorithm's simplicity requires less software and hardware verification and validiation effort leading to reduced expenses within certification processes.
The published and applied geofencing algorithm in industry have already been presented as state-of-the-art in Sec. 1.2.1 and contrasted with the following introduced geofencing algorithm, which corresponds to Contribution II of this thesis.
The chapter is subdivided into five parts. First, in Sec. 2.1, the geofencing terminology within this thesis is introduced and the context of the geofencing development is given. Sec. 2.2 presents a fully-automatic geofencing algorithm for two-dimensional space. The two-dimensional algorithm is extended in Sec. 2.3 to three-dimensional space, which enables the containment of an aerial vehicle in arbitrary flight areas. The design of the respective evasive maneuvers is elaborated in Sec. 2.4. Finally, the introduced geofencing algorithm is verified and tested in model-in-the-loop simulations in Sec. 2.5.

2.1 Geofencing Terminology and Development Environment

Before the geofencing algorithms are introduced, the important wordings within the subsequent sections are clarified:

- **Geofencing** denotes an automatic function, which prevents the UAV to penetrate predefined flight area boundaries.
- A **flight area** is defined as the region, in which the UAV is allowed to fly in and consists of one flight zone and an arbitrary number of non-flight zones within the flight zone.
- The person, who is responsible for the flight of the respective UAV, is denoted as the **flight operator**. Within this thesis, it is assumed, that the flight operator selects the flight area before flight.
- A **flight zone** is represented by one specified polygon.
- A **non-flight zone** is also defined by one polygon.
- The **safety area** is a region equivalent to the flight area, but narrowed down by a safety margin and is defined as the region, in which the pilot is allowed to operate in.
- The presented geofencing algorithm is divided into an offline and an online part. The **offline** algorithm is executed before flight in the ground station and necessary information are sent to the aircraft before flight.
- The **online** algorithm is executed on the flight control computer during the flight of the respective aerial vehicle.
- The **safe area detection algorithm** is part of the online algorithm and identifies, if the aerial vehicle is located in the flight area, the safety area or outside of these areas.

The development of the presented geofencing algorithm has been initiated in the context of a development project between the *Institute of Flight System Dynamics* at the *Technical University of Munich* (TUM-FSD) and an industrial partner. The algorithm has been augmented for three-dimensional space at TUM-FSD. This thesis covers primarily the functional design of the algorithm. The offline functions have been developed in *MATLAB* from *Mathworks*. *C*-code has been generated by the *MATLAB Coder* Toolbox for the two-dimensional algorithm. The online functions are developed in *MATLAB/ Simulink* in a modular way to an assumed classical flight control system. The respective *C*-code can be generated by the *Simulink Coder* from *Mathworks*.

2.2 Two-Dimensional Geofencing Algorithm[3]

Geofencing is applicable to two-dimensional and/or to three-dimensional space. The next section introduces geofencing for two-dimensional space in the horizontal plane and has already been published by the author in Ref. [61]. The presented definitions, requirements and functions are reused in Sec. 2.3, in which the geofencing functions are developed for the horizontal and the vertical dimension.

2.2.1 The Pre-flight Offline Algorithm

2.2.1.1 Flight Area Requirements and Definition

Before flight, the responsible flight operator chooses the flight area including possible non-flight zones on the ground station of the UAV. The specification of the flight should be based on environmental and airspace restrictions according to local and national aviation authorities. It is inevitable to check the flight area for proper definition in order to ensure that the selected flight area is feasible and appropriately defined. On the one hand, the safe area detection algorithm, which is explained later in Sec. 2.2.3.1, demands requirements to the flight area to function correctly. On the other hand, the flight area needs to be feasible for the specific UAV in terms of aerial vehicle's flight limitations. Since the vertices are specified by the flight operator in the WGS84-coordinate system, they are transformed to a local cartesian coordinate frame. Appropriate coordinate transformations are elaborated in Ref. [62]. For the presented geofencing algorithm, proposed within this work, the following requirements are set for the flight area:

- A flight area can be made up of one or more polygons. A single polygon is defined by a set of ordered vertices.
- A polygon is the region of a plane bounded by a finite collection of line segments forming a simple closed curve. A simple closed curve is defined as follows (Ref. [63, p.1f.]):

  Definition: *Let* $v_0, v_1, \ldots v_{n-1}$ *be* n *points in the plane [...], implying a cyclic ordering of the points, with* v_0 *following* v_{n-1}*, since* $(n-1)+1 \equiv n \equiv 0$*. Let* $e_0 = v_0v_1, \ldots, e_i = v_iv_{i+1}, \ldots, e_{n-1} = v_{n-1}v_0$ *be line segments connecting the points. Then these line segments bound a polygon iff* [i.e. if and only if]:

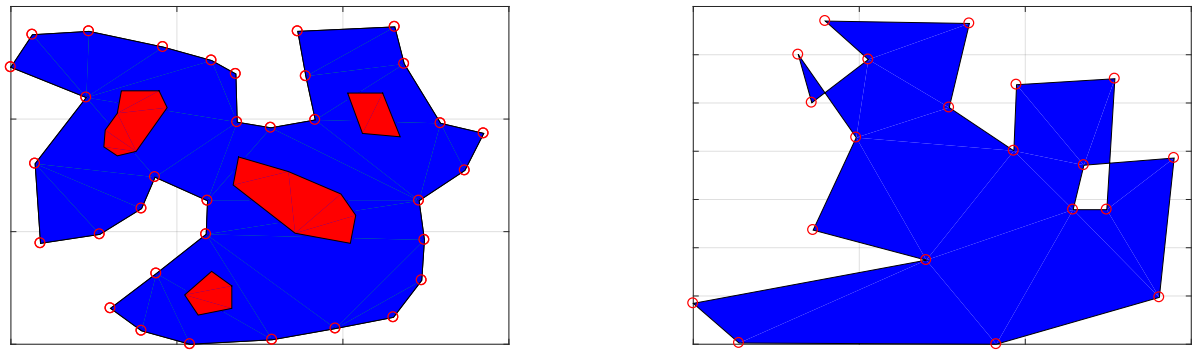
  1. *The intersection of each pair of segments adjacent in the cyclic ordering is the single point shared between them:* $e_i \cap e_{i+1}$*, for all* $i = 0, \ldots, n-1$*.*
  2. *Nonadjacent segments do not intersect:* $e_i \cap e_j = \emptyset$ *for all* $j \neq i+1$*.*

[3] This section is a revised and extended version of the work published by the author in Ref. [61].

The reason these line segments define a curve is that they are connected end to end; the reason the curve is closed is that they form a cycle; the reason the closed curve is simple is that nonadjacent segments do not intersect. The points v_i are called the vertices of the polygon, and the line segments are called its edges. Note that a polygon of n vertices has n edges.

- Attributes can be assigned to these polygons (e.g. flight zone, non-flight zone). The polygons, either non-flight zone or flight zone, shall not overlap; i.e. no intersection between lines of different polygons are allowed. Thus, non-flight zones have to be completely contained within the flight zone.
- A flight area consists of one flight zone and can contain non-flight zones.

Fig. 2.1 illustrates an exemplary valid and invalid flight area. The red circles represent the vertices, which the operator has selected on the ground station. The red polygons display the non-flight zones. The area subtraction of the non-flight zones from the flight zone results in the flight area, which is shown as the blue area. Fig. 2.1(b) represents an invalid flight zone, since self-intersections occur.



(a) Proper defined flight zone with non-flight zones

(b) Invalid flight zone due to self-intersections

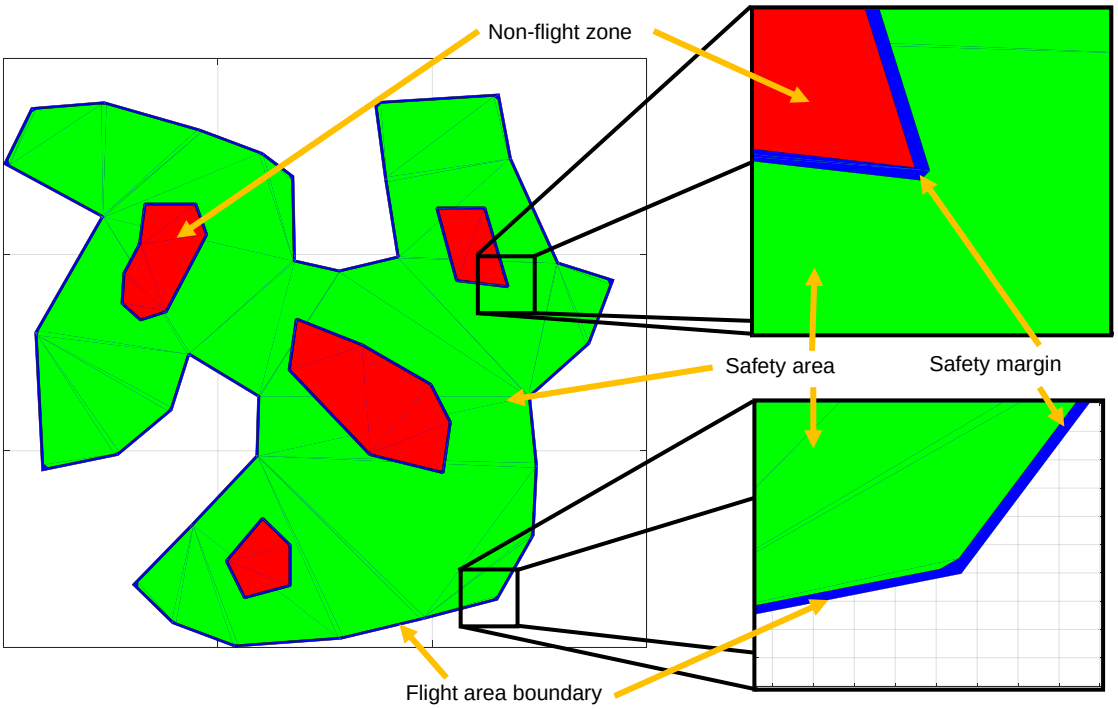
Figure 2.1: Valid and invalid defined flight areas

2.2.1.2 Verification of the Flight Area

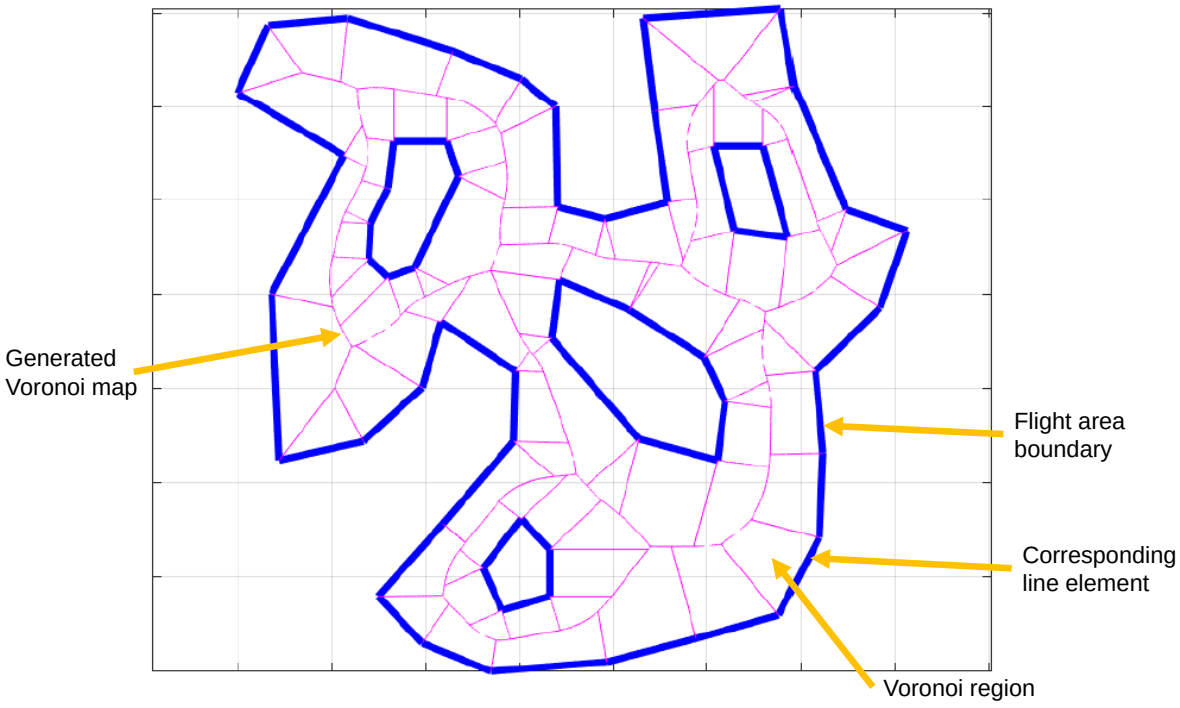
The safe area detection algorithm of the geofencing module requires a proper definition of the flight area. In order to enable an effective safe area detection, the following algorithm checks, if the flight area definition is in compliance with the requirements and the definitions of the flight area described above.
Derived from the flight area definition, the checking algorithm needs to verify, that the selected polygons are simple and that the polygons do not overlap. Consequently, it is sufficient that the validity algorithm analyzes any intersection between the N line segments (i.e.

$N = \sum_k^K n_k$ with n_k line segments of the polygon k and K number of polygons). If an intersection is detected, the defined flight area is inappropriate, and the flight operator is not permitted to start his mission, since the functionality of the safe area detection algorithm would fail.

In general, a set of intersection algorithm can be found in literature (Refs. [64, 65, 66]). The simplest intersection algorithm checks all lines against each other for intersections, resulting in the worst case computation effort of $O(N^2)$. As it is suggested in Ref. [64], the popular *Bentley-Ottmann Algorithm* is utilized within this work. If the number of intersections is small, what is assumed here, this algorithm requires $O(N \log N)$ computational effort only. Consequently, this results in reduction of the computational resources. (Refs. [64, 66])



(a) Flight zone with non-flight zones (red) and created safety area (green)



(b) Generated Voronoi diagram with flight area boundary (blue) and Voronoi lines (magenta)

Figure 2.2: Example of generated flight area and safety area

2.2.2 Safety Area Generation

2.2.2.1 Safety Area Requirements

In the published certification specification document of unmanned air systems (CS-UAS) from JARUS (Ref. [67]) which serves as a certification specification guideline of the specific and the certified category it is distinguished between a *hard* geofence and a *soft* geofence. The *hard* geofence is *"the border [...] which shall not be crossed"*[4]. The *soft* geofence is the *"border of the area, on which action must be taken to prevent crossing the hard fence."*[4]
This differentiation is also applied within the presented geofencing algorithm. The flight area boundary, which the flight operator selects, is equivalent to the *hard* fence. The UAV is only allowed to operate within the so-called safety area, whose boundary represents the *soft* fence. If the aircraft leaves the safety area for any reason, it shall perform an evasive maneuver within a safety margin and be navigated back to the safety area. For the horizontal, two-dimensional geofencing the evasive maneuver is designed as a dedicated turn with a pre-defined bank angle or load factor and is performed within the safety margin. The safety area is based on the flight area, but it is narrowed by the safety margin. Therefore, a proper generation of the safety margin is inevitable, in order to avoid the aircraft from leaving the flight area. In contrast, it is desirable for the operator to have no restriction regarding the mission of the UAV. Hence, the safety area should decrease the flight area as few as possible. The generation of the safety area needs to be compatible to any arbitrary flight area, which fulfills the definition described in Sec. 2.2.1.1. Fig. 2.2(a) demonstrates a proper safety zone in green color.

2.2.2.2 Safety Margin For Dedicated Evasive Maneuver

Fig. 2.10 illustrates the basic procedure of the two-dimensional and horizontal geofencing. The aircraft operates during its mission within the safety area. If the aerial vehicle hits the boundary of the safety area, the aircraft performs a dedicated turn as long as it is located within the safety margin. The turn is based on a predefined bank angle, load factor and speed. After reentry into the safety area, the aircraft can proceed with a defined strategy.
When the aerial vehicle violates the safety boundary, the required space for the turn depends on the flight course, as Fig. 2.10 illustrates. If the aerial vehicle hits a straight line boundary perpendicularly (as on the right side of Fig. 2.10), the largest safety margin is required (compared to all other cases, inter alia to the case on the left side of Fig. 2.10). The worst case scenario determines the safety margin to ensure that the flight area boundary is not violated. As the Sec. 2.2.3.2 points out, a waypoint controller is activated after violation of the safety area.

[4] see Ref. [67, p.15f]

The first step of the safety area generation is the shifting of the flight zone borders inwards and vice versa the non-flight zone boundaries outwards by the safety margin.
The size of the safety margin is based on a predefined turn radius of the UAV. The turn radius is calculated in Sec. 2.4, which introduces the maneuvers in three-dimensional space. The introduced lateral maneuver can be applied for the two-dimensional case. The turn radius R_L is finally calculated as follows. The course rate $\dot{\chi}$ is computed by:

$$\dot{\chi} = \frac{g}{V_K} \tan \mu_K = \frac{g}{V_K} \sqrt{n_z^2 - 1} \tag{2.1}$$

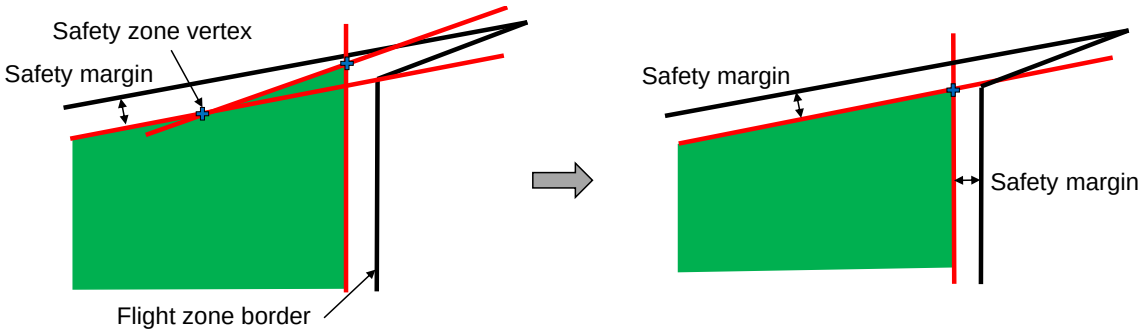
with the load factor n_z, the kinematic bank angle μ and the kinematic velocity V_K. With $V_K = \dot{\chi} \cdot R_L$ the turn radius is gained by:

$$R_L = \frac{V_K^2}{g \tan \mu} = \frac{V_K^2}{g \sqrt{n_z^2 - 1}} \tag{2.2}$$

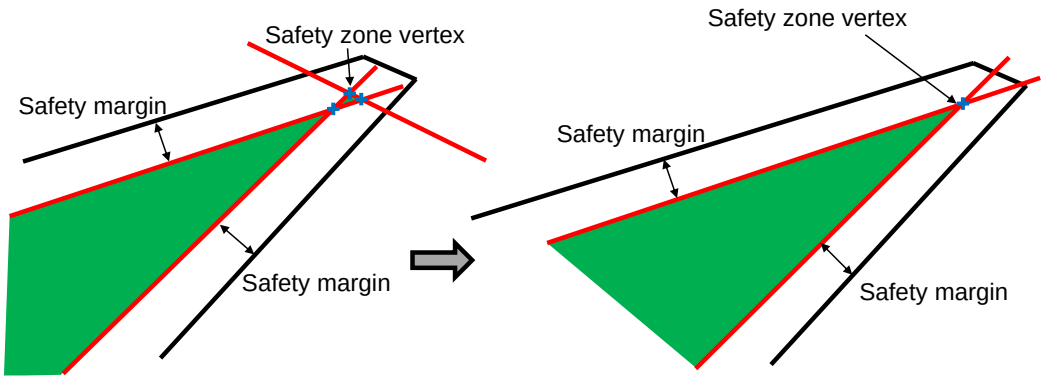
The turn radius R_L is limited either by the maximum lift coefficient $C_{L,max}$ or the maximum load factor $n_{z,max}$ of the vehicle. For the determination of the safety margin, the minimum turn radius at cruise speed or, as required, at the worst-case velocity, is calculated by applying a reasonable or maximum bank angle, or load factor respectively.
It is of particular note that the presented calculated safety margin does not consider wind explicitly. The consideration of wind is not in the scope of this thesis. However, Sec. 2.4.2 discusses shortly, how wind disturbances can be taken into account in future work.

2.2.2.3 Voronoi Map Creation

For generating the offset polygon it is not sufficient offsetting each line segment by the safety margin and calculating the intersection points between the new created offset segments. As Fig. 2.3 shows, this procedure would result in possibly inappropriate safety areas. The case on the left side in Fig. 2.3(a) illustrates, that only offsetting could result in inadequate intersection points and thus, in inappropriate offset polygons. The case in Fig. 2.3(a) can be resolved by skipping the short segment for offsetting and calculating the intersection point between the two longer segments (see right side of Fig. 2.3(a)). But further issues arise, as another case in Fig. 2.3(b) demonstrates. There, intersections between the offset line segments result in incorrect offset polygons. Although the case in Fig. 2.3(b) can be resolved as well (right side of Fig. 2.3(b)), other vertex constellation led to wrong generation of offset polygons. In order to follow a general approach, which is valid for any proper defined flight area, Voronoi diagrams are identified as general method to generate correct offset polygons. An example of a generated Voronoi diagram from a flight area is demonstrated in Fig. 2.2(b). A Voronoi diagram divides the flight area, consisting of one flight zone and several non-flight zones, in Voronoi regions. Thereby, the Voronoi algorithm considers each vertex and each



(a) Difficulties in offset polygon generation: Exemplary Issue I



(b) Difficulties in offset polygon generation: Exemplary Issue II

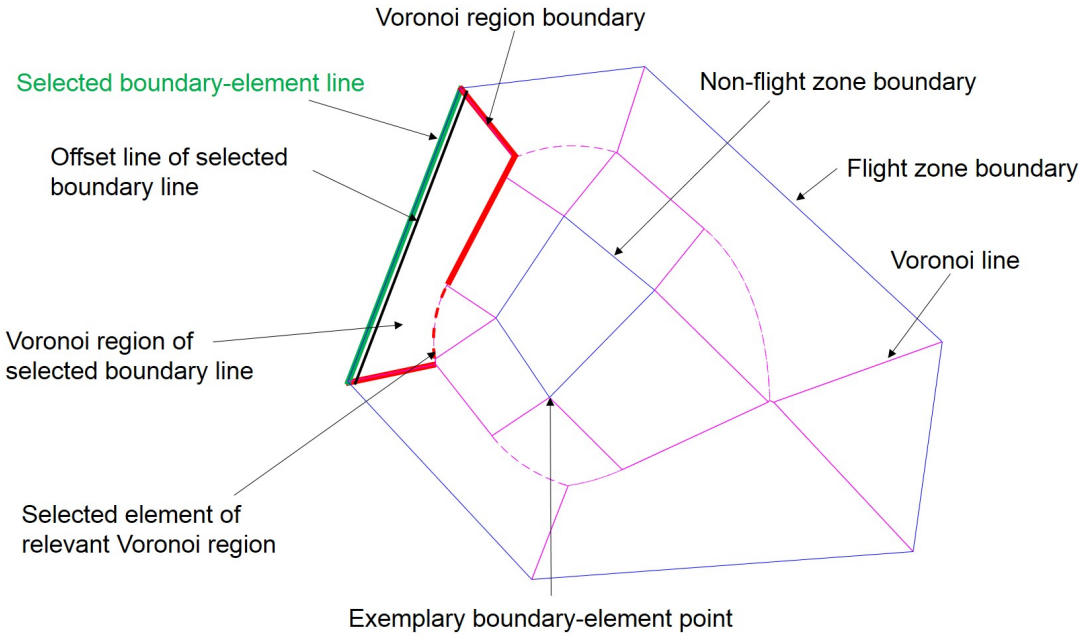
Figure 2.3: Issues of default offset polygon generation

line segment as element. Each element (vertex or line segment) corresponds to one Voronoi region. Each Voronoi region comprises all points which are closer to the respective element than to any other element. The area of a Voronoi region can be zero. The boundaries of a Voronoi region are straight lines, specified by two points, or parabolas defined by homogenous coordinates. The algorithm for the creation of the Voronoi diagram for the above defined flight areas is elaborated in an own paper (Ref. [68]).

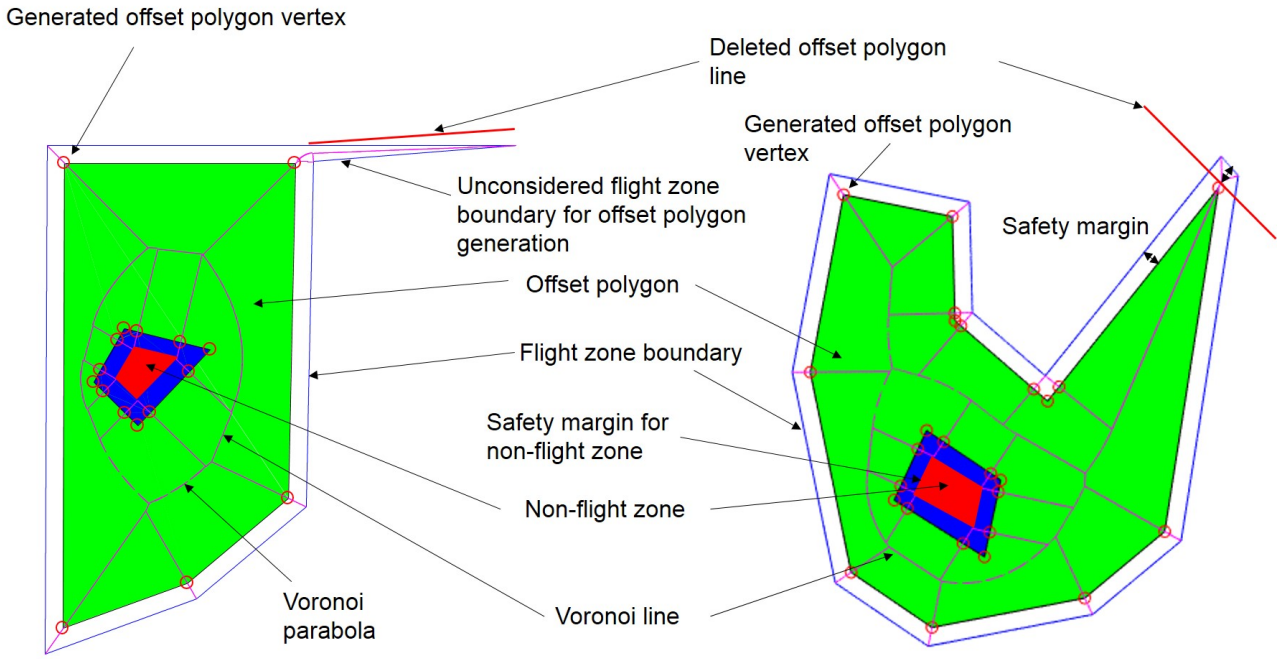
2.2.2.4 Offset Polygon Generation

Utilizing the feature of Voronoi diagrams offset polygons can be conveniently created in a general approach. The offset polygon generation is summarized in Algorithm 1. Fig. 2.4(a) illustrates the procedure of the algorithm:

- Each element (vertex or line segment) of each polygon (either flight zone or non-flight zone) is selected by the algorithm once.
- If the element is a line (marked in green in Fig. 2.4(a)), the line segment is shifted by the safety margin (marked in black color). Then the intersection points are calculated between the offset line and all elements of the corresponding Voronoi region.
- If the number of computed intersection points for the selected line is two, the intersection points are included into the vertex list of the corresponding safety area.
- If the selected element of the flight area is a vertex, a circle with the radius equal to the safety margin and the vertex as circle center is defined in homogeneous coordinates. Then the intersection points are calculated between the circle and all elements of the respective Voronoi region.
- If the number of computed intersection points for the selected vertex is equal to two, the intersection points are included into the vertex list of the corresponding safety zone. An additional vertex is calculated, which is tangential to the corresponding circle. This is necessary, since in the final online algorithm the flight area is completely defined by vertices, which are connected by line segments. The online algorithm doesn't consider circle segments as boundary. The creation of the mentioned additional vertex point (cyan color) is illustrated in Fig. 2.5 with two example points. The selected vertex element of the flight zone are marked as magenta rectangles.
- If less or more than two intersection points are computed, the element is not considered for the safety zone. This is the case for example, if adjacent line segments are close to each other, as in the issue cases in Fig. 2.3. In contrast, Fig. 2.4(b) illustrates, how the exemplary difficulties in Fig. 2.3 are overcome implicitly by the proposed algorithm.



(a) Offset polygon creation using generated Voronoi diagram



(b) Implicit avoidance of special cases by utilizing Voronoi diagram approach

Figure 2.4: Offset polygon generation

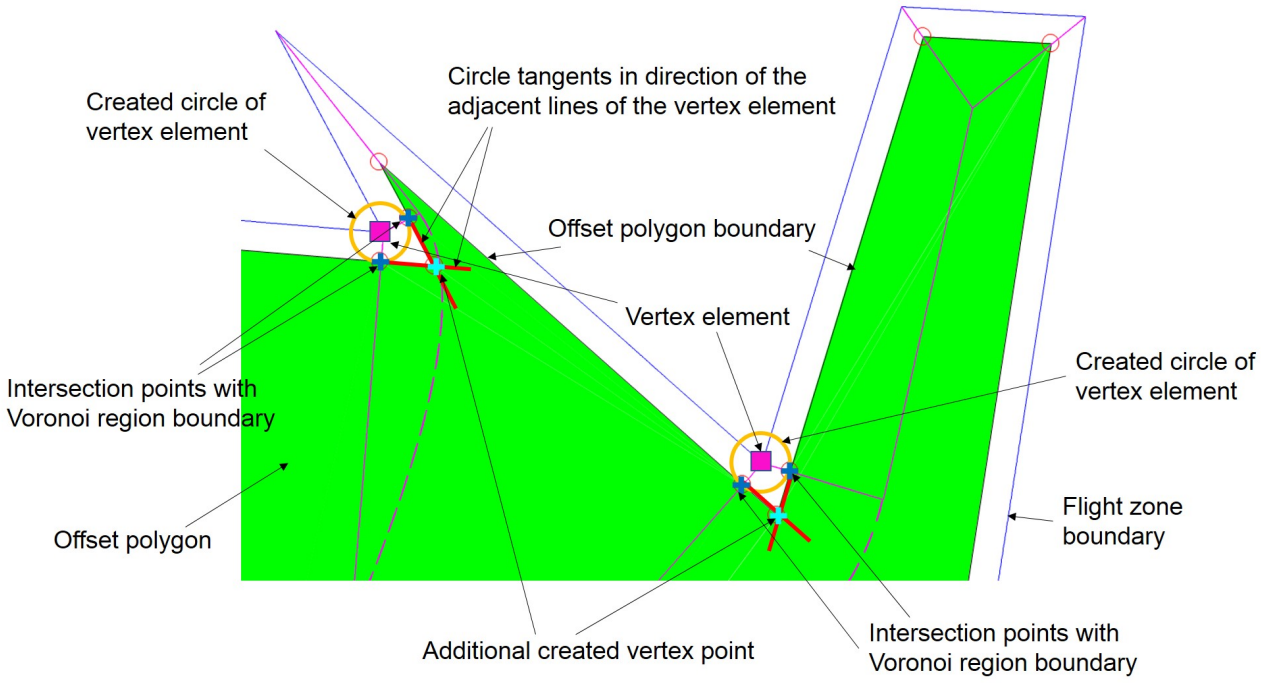


Figure 2.5: Vertex creation for offset polygon generation

Algorithm 1 Offset-Polygon-Generation

```
Voronoi-Map-Generation
for Choose boundary element do
    if Check if boundary element is line then
        for Calculate intersections with corresponding Voronoi region boundaries do
            Count intersections between line boundary element and lines of Voronoi region
            Count intersections between line boundary element and parabolas of Voronoi region
        end for
    else if Check if boundary element is point then
        for Calculate intersections with corresponding Voronoi region boundaries do
            Count intersections between circle boundary element and lines of Voronoi region
            Count intersections between circle boundary element and parabolas of Voronoi region
        end for
    end if
    if Intersection counter equal to 2 then
        if If offset element line then
            Take intersections as vertices
        else if If offset element point then
            Calculate both tangents of corresponding circle
            Create additional vertex
            Take points as vertices
        end if
    end if
end for
```

2.2.2.5 Safety Area Derivation

Although the generation of the offset polygons from the flight area is a crucial step, it is not sufficient for a proper geofencing functionality. On the one hand, as the left side of Fig. 2.6(a) illustrates, the aircraft violates the flight area in case the aircraft flies towards a vertex (i.e. in the range of two times the safety margin between the vertex and the location of the violation), if the evasive maneuver is started after passing the boundary of the generated offset polygon and if the angle α between two adjacent line segments of the vertex is $\alpha \leq$180 deg. On the other hand, if the angle α between two adjacent line segments is $\alpha >$180 deg, the boundary of the generated offset polygons is too conservative, i.e. it covers areas, where it is not required to already perform the evasive maneuver. This area is hatched in black color on the left side in Fig. 2.7. In this case, the area would be restricted unnecessarily.
Therefore, the boundaries are further adjusted for the angles $\alpha \leq$180 deg and $\alpha >$180 deg. The angle α is the angle between two adjacent line segments. It always lies on the inner side, if the corresponding vertex is part of the flight zone. It lies on the outer side, if the corresponding vertex is part of a non-flight zone. As a result, the angle points always to the inside of the flight area:

- $\alpha \leq$180 deg
  - With the generated offset polygons being the switching boundary for the evasive maneuver, the aerial vehicle leaves the furthest from the flight area boundary if it flies directly towards a vertex on the bisecting line of the angle α (see Fig. 2.6(a)). For this scenario, the aerial vehicle is prevented to leave the flight area by restricting the area by a new line segment. Instead of the vertex of the offset polygon, two new vertices are generated. The coordinates of the new vertices are calculated by adding the normalized direction vector of both line segment multiplied by the distance d to the corresponding vertex of the offset polygon. Based on Fig. 2.6(a), the distance d is calculated as follows:

    $$d = \frac{r_t}{\sin(\alpha/2)} \tag{2.3}$$

    The newly created vertices are entered into the vertex list of the final safety area. Since the new vertices cover the worst case for angles $\alpha \leq$180 deg, different approaches towards the boundary are covered implicitly. The resulting safety area for the presented case is shown on the right side of Fig. 2.6(a).
  - As the left side of Fig. 2.6(b) demonstrates, the adaptions above cannot be applied to every vertex with an angle $\alpha \leq$180 deg. If the vertices of the offset polygon are close to each other, the creation of the new vertices lead to invalid polygons. As

shown exemplary on the right side in Fig. 2.6(b), for each polygon offset vertex one new vertex is skipped. This scenario occurs, if the distance d of the vertex I is larger than the distance between the vertex I and one new generated vertex of the polygon offset vertex II.

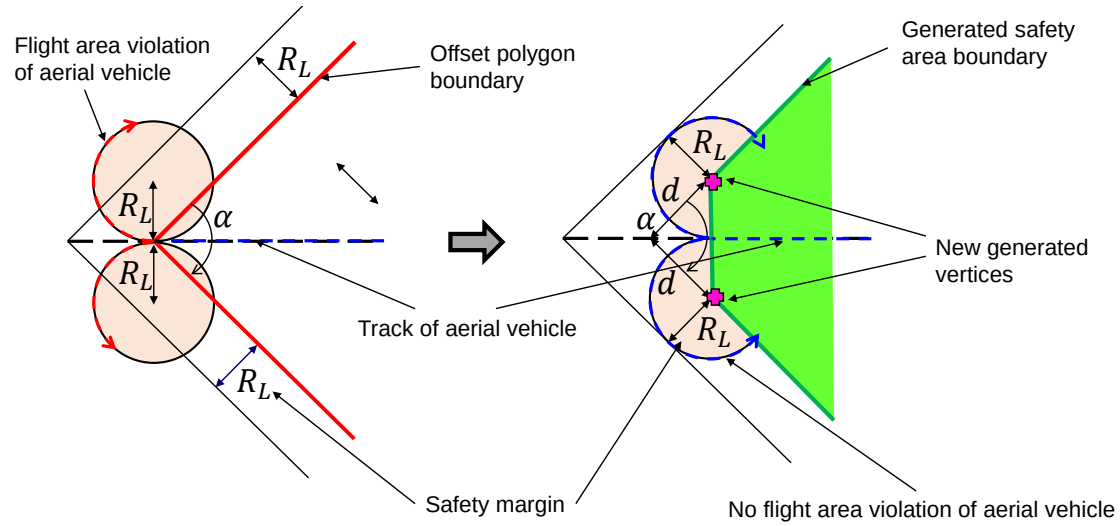
- $\alpha >$180 deg
  - If the angle α is $\alpha >$180 deg, a reduced safety margin at the corresponding corner is suitable as visualized in Fig. 2.7. The ideal boundary consists of a circle segment at the vertex according to the right side of Fig. 2.7. Since the polygons in the presented work consist of line segments only, an additional line is created, which is tangential to the mentioned circle. The additional line is defined by two new created vertices. The distance d is calculated as follows:

$$\sin(\beta/2) = \frac{R_L}{R_L + x} \text{ and } \cos(\beta/2) = \frac{x}{d}$$

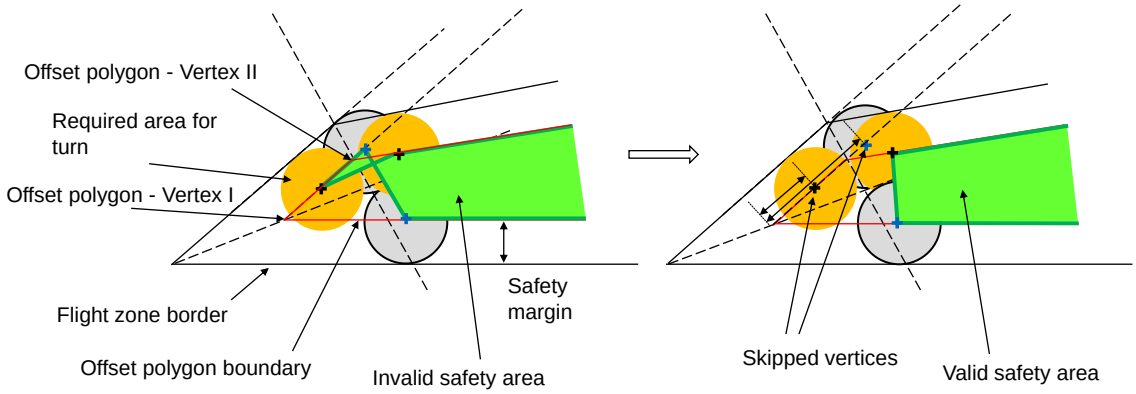
$$R_L + \cos(\beta/2) \cdot d = \frac{R_L}{\sin(\beta/2)}$$

$$d = \frac{r_t}{\sin(\beta/2)\cos(\beta/2)} - \frac{r_t}{\cos(\beta/2)}$$

The newly generated vertices for each zone are collected in a clockwise order as a simple closed curve according to the definition in Sec. 2.2.1.1 for each dimension.



(a) Challenges in assuring the required safety margin close to a vertex



(b) Challenges in assuring the required safety margin, if vertices are close to each other

Figure 2.6: Difficulties in safety area generation for angles $\alpha \leq$180 deg

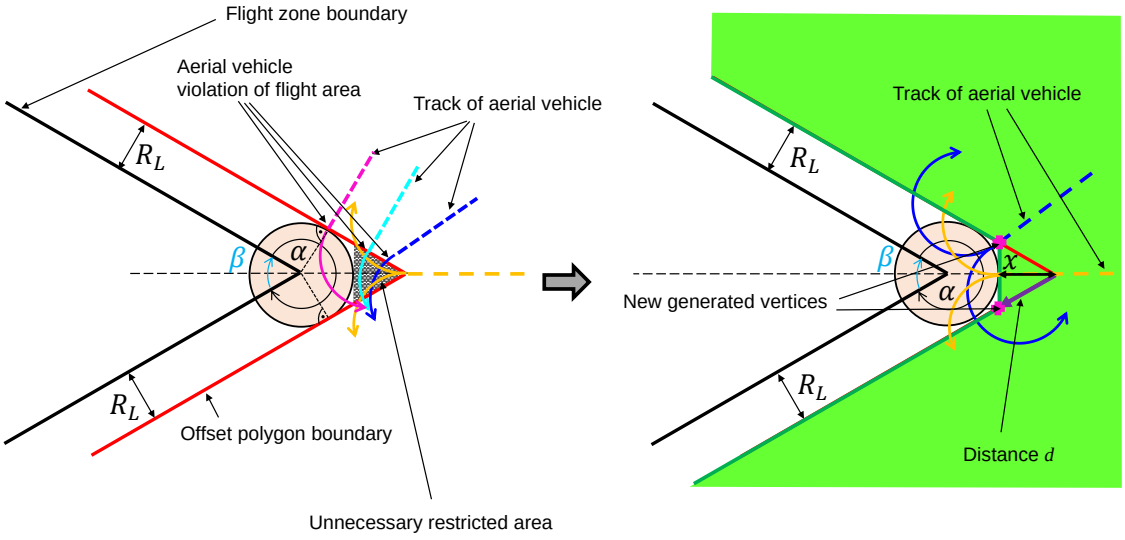


Figure 2.7: Difficulties in safety area generation for angles $\alpha >$180 deg

2.2.2.6 Safety Area Validity

Equivalent to the flight area validity, the validity of the safety area is required to be verified. Two algorithms have been identified, which are executed before flight after the flight area validity check (see Sec. 2.2.1.2).

Intersection Check: Analogously to the flight area validity check (see Sec. 2.2.1.2), the generated safety area is checked for self-intersections in order to ensure the proper functioning of the safe area detection algorithm.

Safety Margin Algorithm Check: The safety area ensures that the aircraft can perform an evasive maneuver in at least one turn direction within the specified safety margin. To verify the generated safety area is in distance of the safety margin to the flight area boundary, an additional checking algorithm is developed. This algorithm has been executed for the verification of the presented safety area generation. Yet this algorithm is also executable after the selection of the flight area as an extended verification test. The algorithm is based on the design of the evasive maneuver, which is introduced in Sec. 2.2.3.2. However, the evasive maneuver design is based on the assumption, that the aerial vehicle is able to follow the track of a circle after the aerial vehicle has initiated its turn. The turn can be performed in two directions, but the aerial vehicle should be able to perform the turn at least in one direction without violation of the flight area boundary (see Sec. 2.2.3.2 and Sec. 2.4). The algorithm utilizes this behavior to check if the aerial vehicle is able to turn along the whole flight area boundary without penetration. The algorithm is visualized in Fig. 2.8. In the algorithm, the contact point of two circles moves along the safety area boundary in a specified line step size. The center of the two circles are connected by a line. The line between the two circles and the safety boundary line form an angle.

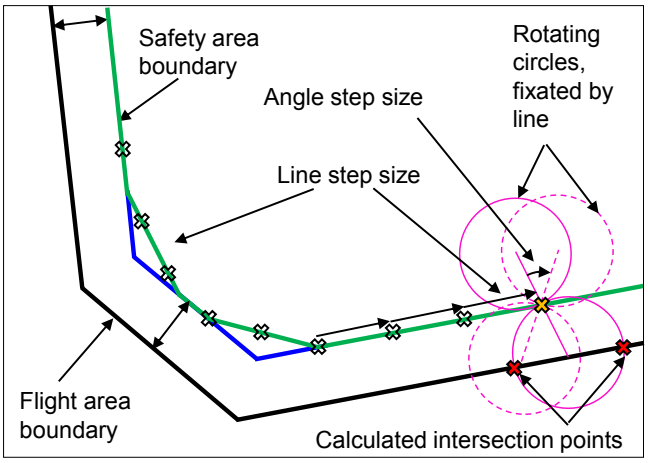


Figure 2.8: Illustration of safety margin checking algorithm

At each line step the angle is changed by a specified angle step size until the line reached a 180 deg rotation. Within such a step the number of intersections between the two circles and the flight area boundary lines are calculated. If the number of intersections of one circle is lower than two in one step the safety area is valid at the analyzed location and angle. The rotation is performed at each line step.
The connected circles go along the safety area and rotate in a specified angle step size at each location (see Fig. 2.8). In that way, it is assured that the safety margin is hold for at least one turn direction. If the intersection number of both circles in one step is greater than one, the safety margin is not ensured. Assuming the aircraft is able to turn along the circle, it is always assured that the aircraft does not leave the flight area boundary.

2.2.3 Two-Dimensional Online Geofencing Algorithm

The online part of the geofencing algorithm is executed on the flight control computer and shall be active during the whole mission of the UAV. It consists of the safe area detection algorithm presented in Sec. 2.2.3.1 and the execution of an evasive maneuver in case the aircraft has left the safety area, which is elaborated in Sec. 2.2.3.2. After leaving the safety area has been detected, a proper waypoint is generated within the safety area. The waypoint controller of the flight control software is activated and the aircraft is guided towards this waypoint. When the aircraft has reentered the safety area, a defined strategy can be proceeded. However, these strategies are beyond the content of this thesis.

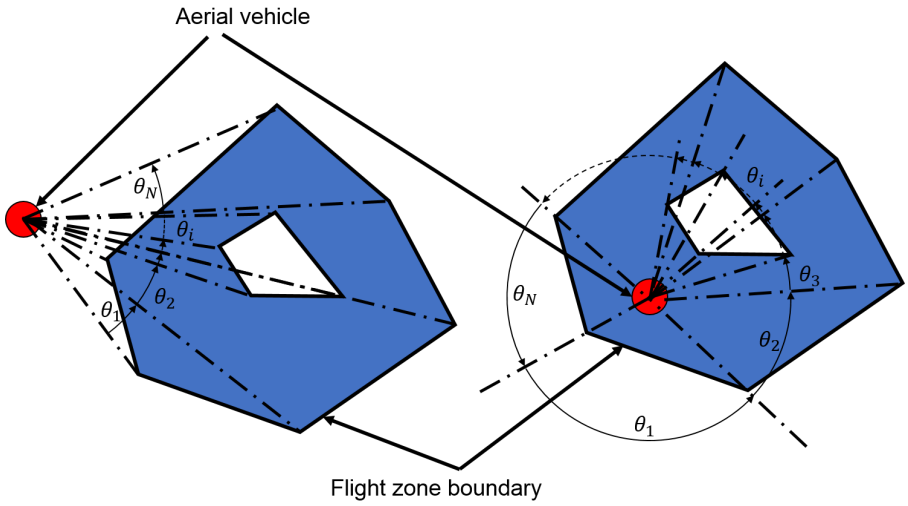
2.2.3.1 Two-Dimensional Safe Area Detection Algorithm

The safe area detection functionality itself comprises a point in polygon algorithm. The algorithm requires a flight area A consisting of one flight zone and optional number of non-flight zones, which do not overlap and are defined as simple closed curves, as described in Sec. 2.2.1.1. The point p within the algorithm represents the position of the aircraft, which is measured by the navigation system. The algorithm detects if $p \in A$. In computational geometry, usually either the crossing number algorithm or the winding number algorithm are used for this type of problem (Ref. [63]). Both algorithm are illustrated in Fig. 2.9:

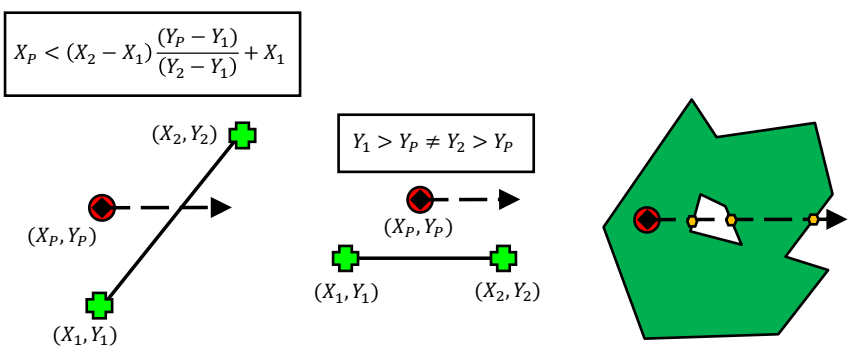
- Winding Number Algorithm:
  As the name states, the algorithm calculates the winding number , which is defined as follows:
  $$w_n = \frac{1}{2\pi} \sum_{i=1}^{N} \theta_i \tag{2.4}$$
  with the number of lines N and the angle θ_i. The angle θ_i is spanned between the two adjacent vertices and the position of the aircraft as angular point. If w_n is equal to one, the point p is within the flight area A, otherwise the point p is outside.

- Crossing Number Algorithm:
  The crossing number algorithm draws a line from the point p in any arbitrary direction and counts the number of intersection between the drawn line and any line of the flight area. If the number of intersection is odd, the point is within the flight area, otherwise the point is outside of the polygon. In Fig. 2.9(b), the line, which is drawn from the aircraft position along one coordinate is shown as dashed line. The solid line represents a line of a polygon the dashed line may hit. As shown in the middle part of Fig. 2.9(b), if the dashed line doesn't intersect with one of the polygon lines, no intersection emerges.

Since the winding number algorithm requires the utilization of trigonometric functions and



(a) Winding number algorithm



(b) Crossing Number Algorithm

Figure 2.9: Illustration of point-in-polygon algorithms: X_P, Y_P represent the coordinate of the aircraft position (red dot), X_1, Y_1 and X_2, Y_2 are coordinates of start and end point of a line.

floating point computations, the execution of the crossing number algorithm is twenty times faster according to Ref. [63]. For that reason, the crossing number algorithm is utilized as safe area detection algorithm. The point p represents the position of the aircraft and is measured by the navigation system. The algorithm requires a flight area which is comprised by a flight zone and an optional number of non-flight zones and fulfills the flight area requirements explained above.

2.2.3.2 Evasive Maneuver Planning and Conduction

When the safe area detection algorithm observes a leaving of the safety area, an evasive maneuver is conducted. As mentioned above, in the proposed two-dimensional geofencing algorithm, a coordinated turn is performed. The coordinated turn is realized by a waypoint navigation controller. When the aircraft leaves the safety area, an appropriate waypoint is

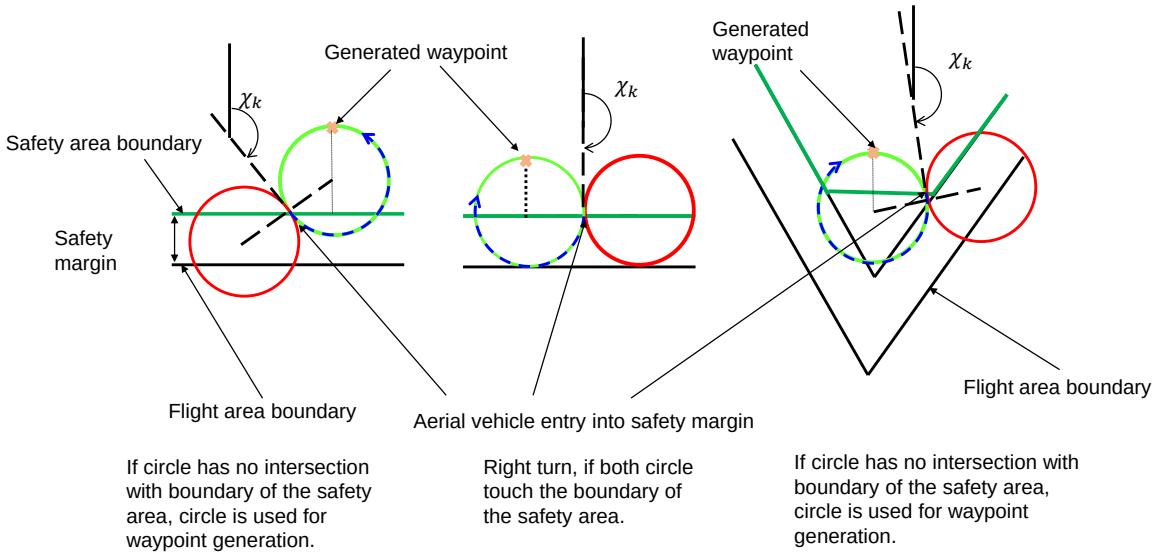


Figure 2.10: Appropriate online selection of turn direction

generated and the waypoint controller is activated. Fig. 2.10 illustrates how the waypoint is generated. First, it is detected, how the aerial vehicle course relates to the direction of the violated boundary segment. Depending on the relation, two circles are generated as demonstrated in the Fig. 2.10. It is calculated, if intersection points between the two generated circles with the violated or adjacent boundary lines exist. If no intersection point or a tangential point between one circle and the corresponding line segments exists, then a waypoint is generated on that circle (see Fig. 2.10). If two intersection points with the line segment (or the extended line segment respectively) exist, the aerial vehicle would violate the flight area, if the path of the circle is followed. If this is the case, the other circle is followed. It can be geometrical proven, that if two intersections of the extended line with one circle exist, the second circle will not intersect with the corresponding lines. In this way, the appropriate turn direction and the waypoint are determined. If both circles have one intersection point, then a defined turn direction is chosen. In reality, the aerial vehicle in wingborne flight is not capable to perform a geometrical circle path, since the aerial vehicle first requires time to establish a bank angle. This period is neglected in this algorithm, but can be considered by an increased safety margin. An appropriate trajectory generation and waypoint controller is elaborated inter alia in Refs. [69, 70]. Clothoid Segments can be used to deal with the transition phase between straight and arc flight (see Ref. [71]).

2.2.3.3 Online Algorithm Testing

Fig. 2.11 shows an exemplary model-in-the-loop simulation in order to underline the functionality of the presented geofencing algorithm. For this simulation, a two dimensional kinematic model is used. The algorithm is tested on models with six degree of freedom for the three-dimensional case, as discussed in the next session. Within this example the aerial vehicle

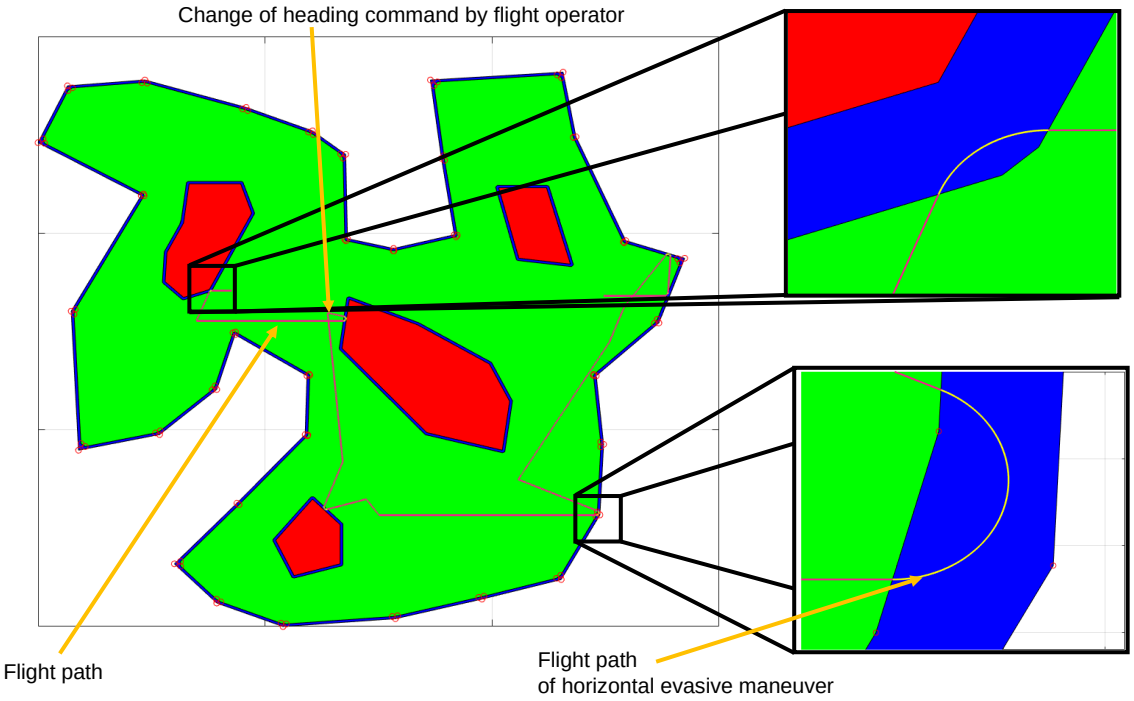


Figure 2.11: Verification of geofencing algorithm in model-in-the-loop simulation

follows a heading command. Turns within the green safety area occur, when a different heading command is given to the implemented autopilot. The flight path of the aerial vehicle is illustrated in magenta color, if the aerial vehicle is located within the green safety area. The color of the flight path is the output of the safe area detection algorithm explained in Sec. 2.2.3.1. If the aerial vehicle enters the blue safety margin, the flight path changes to yellow color. Then, the two circles are generated and the correct turn direction is chosen according to the previous subsection. The waypoint controller is activated and the aircraft follows the generated trajectory as long as the aircraft is located within the safety margin. As mentioned above, after reentering the safety area, different strategies are possible. In this example, the aerial vehicle flies further in straight level flight. Thereby, a heading command is given, which is equivalent to the course of the vehicle, when it has reentered the safety area. As long as the flight operator does not intervene, the aerial vehicle flies according the heading command. This strategy is chosen for illustration purposes of the geofencing algorithm.

2.3 Functional Extension For Geofencing in Three Dimensional Space[5]

Although two-dimensional geofencing approaches are sufficient for low-cost and small UAVs with smaller range and low altitude, heavier UAV will participate in commercial air spaces, where three-dimensional geofencing becomes relevant. Fig. 2.12 illustrates the air space structure using the example of the german air traffic control authority (Ref. [73]). Since the commercial air space structure changes by altitude, geofencing and safe area detection algorithm are necessary in three-dimensional space. Further, detected obstacles in flight by advanced sensor system will have a three-dimensional shape. Consequently, the following section describes a fully-automatic geofencing algorithm in three-dimensional space. It proposes a mathematical approach to detect, if the aircraft leaves the flight area, which consists of a three-dimensional boundary with a variable number of non-flight zones. The zones can have any arbitrary, three-dimensional shape with plane polygonal faces as long as the boundary planes are not self-intersecting. Analogously to the two-dimensional case, the extension of the offline algorithm is proposed in Sec. 2.3.1 first. Subsequently, the safe area detection algorithm, executed online, for the three-dimensional case is elaborated in Sec. 2.3.2. The following Sec. 2.3 has been published by the author in Ref. [72].

2.3.1 Pre-Flight Offline Algorithm For Three-Dimensional Space

2.3.1.1 3D-Flight Area Definition and Specification

The flight area is defined by vertices specified in a local reference coordinate frame in three-dimensional space by the cartesian coordinates x,y,z or by WGS84 coordinates. As in the two-dimensional case, depending on the range of the aerial vehicle, different coordinate transformations can be used to calculate the cartesian coordinates from the WGS84 coordinates. Before the flight, the operator determines the airspace for the planned mission based on the current airspace requirements and regulations. The user selects the flight zone and the non-flight zones within the ground control station successively. Either he chooses predefined shapes and enters the required coordinates, or he defines a new geometrical shape. Each zone can have different geometrical shapes. In order to be compliant with the proposed algorithm, the geometrical shapes need to fulfill certain geometrical requirements. The proposed algorithm is capable to process any three-dimensional geometric shape, which represents a polyhedron. The polyhedron is defined as a three-dimensional solid, *"in which the faces are planar, simple polygonal regions, and different faces intersect only along common edges or vertices"* (Ref. [74]).

[5] This section has been published by the author in Ref. [72] and revised within this thesis.

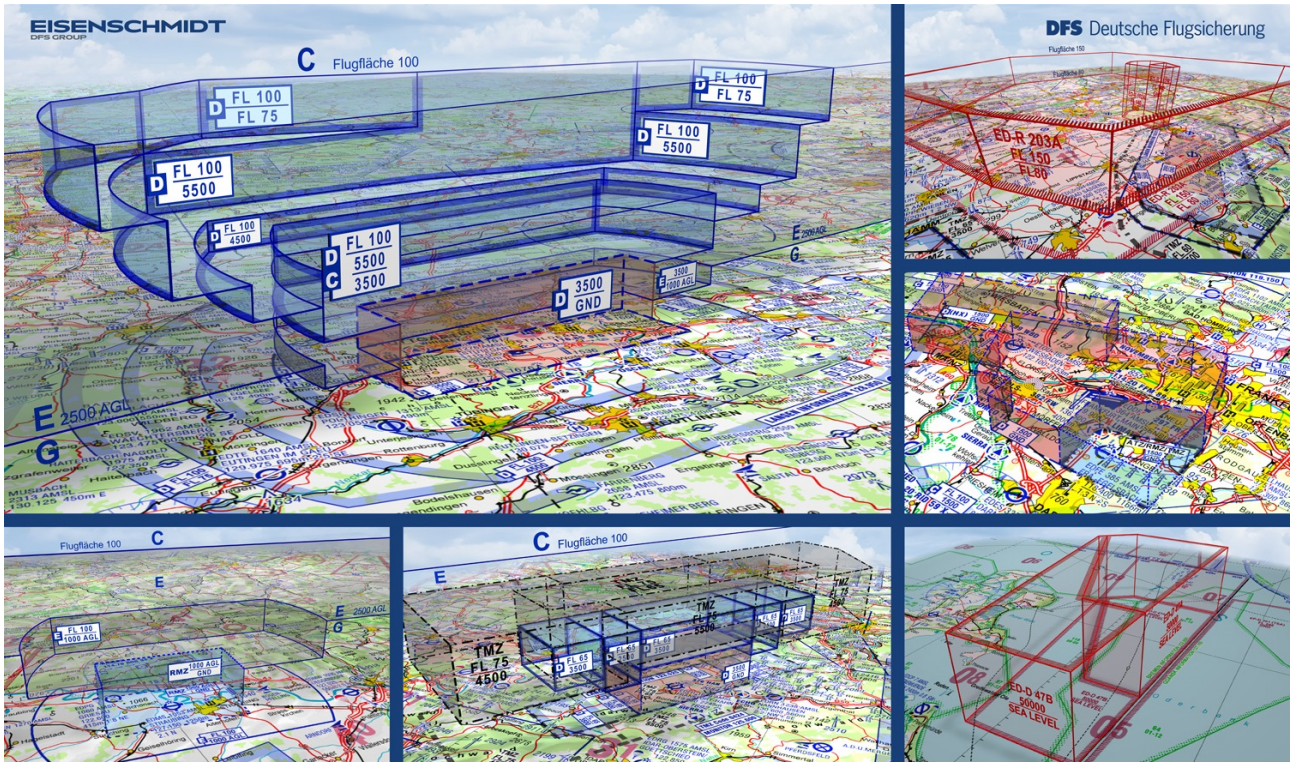
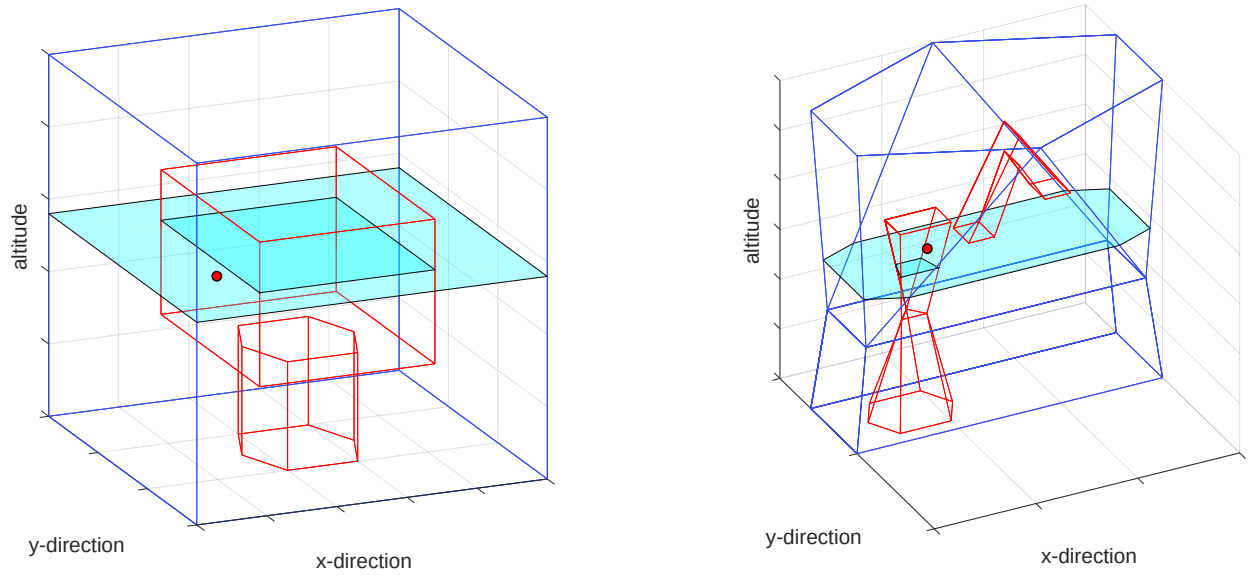


Figure 2.12: Exemplary air space structure of German Air Traffic Control Authority (image provided by Ref. [73])



(a) Example I: Flight zone with two non-flight zones

(b) Example II: Flight zone with three non-flight zones

Figure 2.13: Conceptional flight area examples in three-dimensional space with current calculated xy-plane at current aircraft (red circle) altitude z

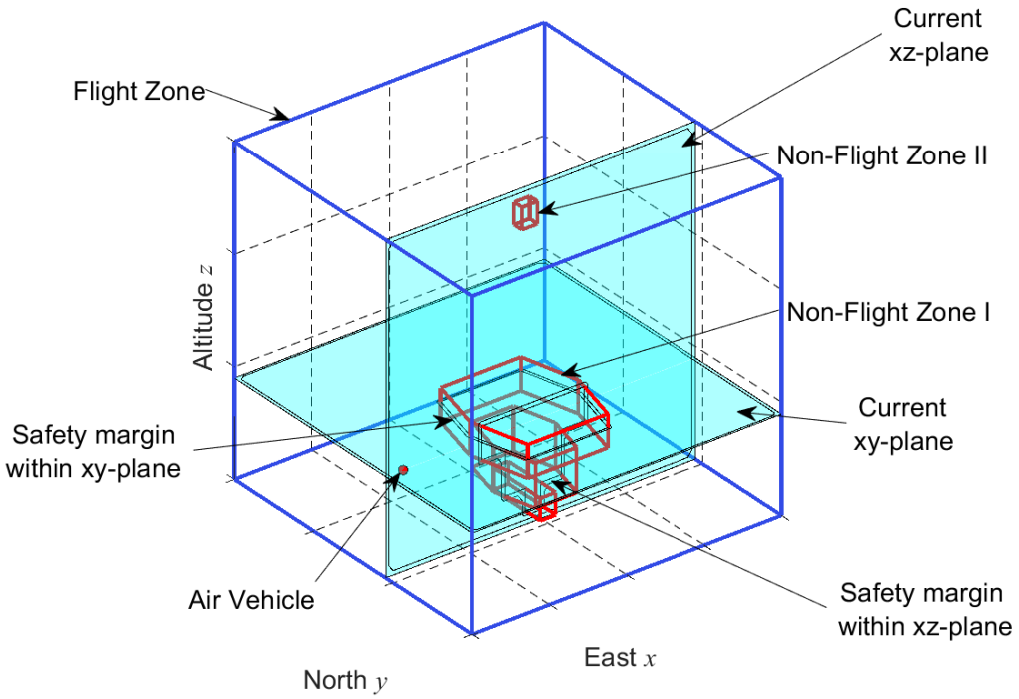
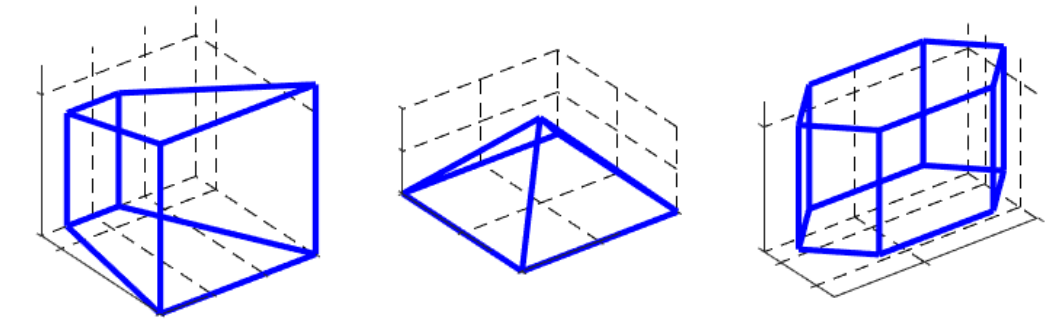
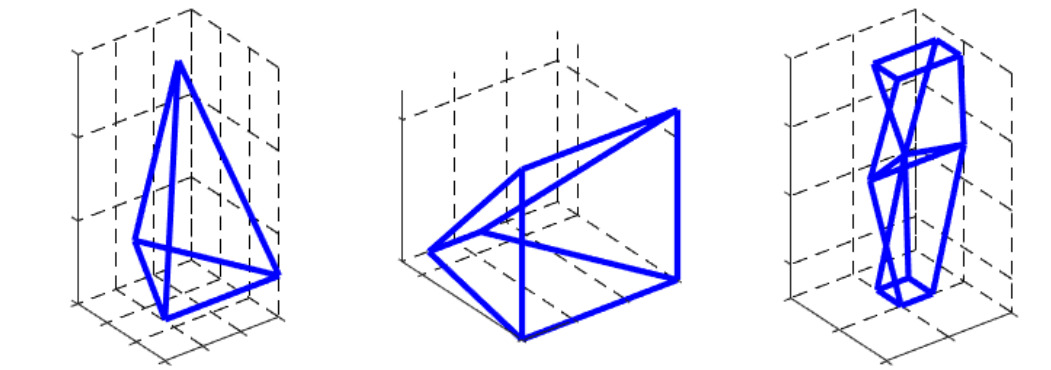


Figure 2.14: Exemplary flight area with two non-flight zones and with current computed xy- and xz-plane at current aircraft (red circle) position



a) Stretched Cuboid b) Pyramid c) Hexagonal Prism



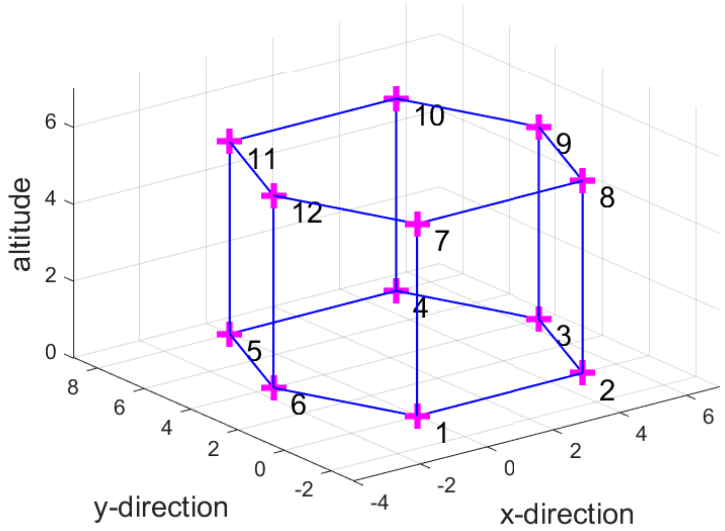
d) Tetrahedron e) Stretched Wedge f) Arbitrary possible shape

Figure 2.15: Examples of flight zone geometries

The zones can have any arbitrary shape with (planar) polygon faces, but no self-intersections within each polygon are allowed. In Fig. 2.15 geometrical examples of flight zones or non-flight zones are shown. These shapes could, for example, represent the predefined zones, which the user can preselect. As the examples underline, a high variety of different shapes is possible. To ensure an user-friendly application, known airspace restrictions can be predefined within the ground control station of the aerial vehicle.
The flight area restrictions and limitations of the user-defined flight area are as follows:

- The flight area consists of one flight zone and a defined number of non-flight zones
- Zones comprise a body in three-dimensional vector space
- Each body is spanned by different flat faces, which have the shape of polygons. The faces are not allowed to have self-intersections
- All non-flight zones lie within the flight zone
- Zones can contact, but are not allowed to intersect each other
- All relations and connections between points are linear

Examples of proper flight areas in three-dimensional space are depicted in Fig. 2.13 and Fig. 2.14. The flight zone and the non-flight zones are visualized in blue and red color.
The defined flight area is uniquely specified by the array $\boldsymbol{A} = \{\boldsymbol{V}_1, ..., \boldsymbol{V}_M |\ \boldsymbol{V}_i \in \mathbb{R}^{N_i \times \left(N_i+3\right)}, i = 1, ..., M\}$ with M number of zones and N_i number of vertices of the ith zone. The ith flight zone is defined by the vertex matrix $\boldsymbol{V}_i$. Valid geometrical shapes are saved in the vertex matrix $\boldsymbol{V}_i$ for each zone to process the flight area to the algorithm. The vertex matrix comprehends the x, y, z-coordinates in the first three columns. The subsequent columns contain a square connection matrix, which comprises the digits zero or one only. This matrix defines, which vertices are connected to each other. *Zero* indicates that there is no connection, *one* indicates that there is a connection.
The flight area is saved in the two different arrays $\boldsymbol{A}_{xy}$ and $\boldsymbol{A}_{xz}$. The distinction is necessary due to the introduction of xz-planes and xy-planes. This is discussed in the next two subsections. The two arrays differ in the order of the vertices and the composition of the different zones. However, the resulting flight areas are equivalent.
The composition of the vertex matrix $\boldsymbol{V}$ for y-planes and z-planes is in accordance to the explanation above, but the vertex order and thus also the connection matrix are different. Fig. 2.16 illustrates the vertex matrix $\boldsymbol{V}$ for the $\boldsymbol{A}_{xy}$-array and the xy-planes. Fig. 2.17 demonstrates the vertex matrix $\boldsymbol{V}$ for the $\boldsymbol{A}_{xz}$-array and the xz-planes. The vertex order for the y-planes is ascending with the y-coordinate. Vertices at one y-coordinate are ordered such that valid two-dimensional polygons result by connecting vertex by vertex with ascending order. Similarly, the vertex order for the z-planes is ascending with the z-coordinate.

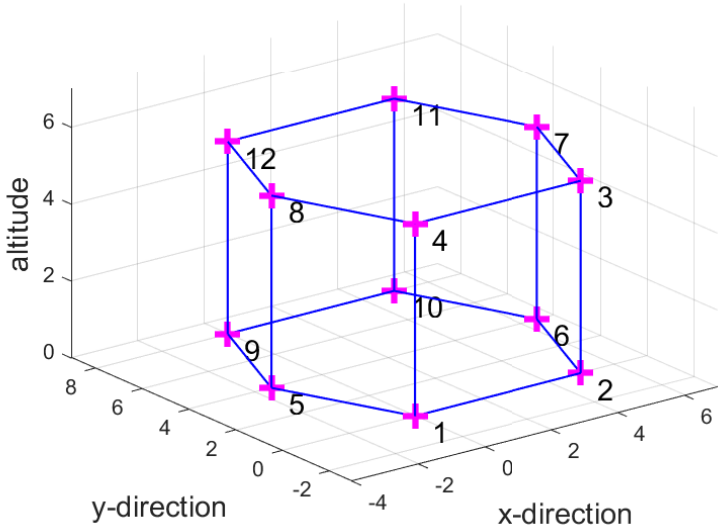


a) Vertex order for $\boldsymbol{A}_{xy}$-array

Vertex order	Coordinates			Connection List $N \times N$											
m	x	y	z	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	0
2	5	0	0	1	0	1	0	0	0	0	1	0	0	0	0
3	6.5	4	0	0	1	0	1	0	0	0	0	1	0	0	0
4	5	8	0	0	0	1	0	1	0	0	0	0	1	0	0
5	0	8	0	0	0	0	1	0	1	0	0	0	0	1	0
6	-1.5	4	0	1	0	0	0	1	0	0	0	0	0	0	1
7	0	0	5	1	0	0	0	0	0	0	1	0	0	0	1
8	5	0	5	0	1	0	0	0	0	1	0	1	0	0	0
9	6.5	4	5	0	0	1	0	0	0	0	1	0	1	0	0
10	5	8	5	0	0	0	1	0	0	0	0	1	0	1	0
11	0	8	5	0	0	0	0	1	0	0	0	0	1	0	1
12	-1.5	4	5	0	0	0	0	0	1	1	0	0	0	1	0

b) Vertex Matrix $\boldsymbol{V}$ of Hexagon for $\boldsymbol{A}_{xy}$-array

Figure 2.16: Example of flight zone definition by the vertex matrix for z-level discretization



a) Vertex order for $\boldsymbol{A}_{xz}$-array

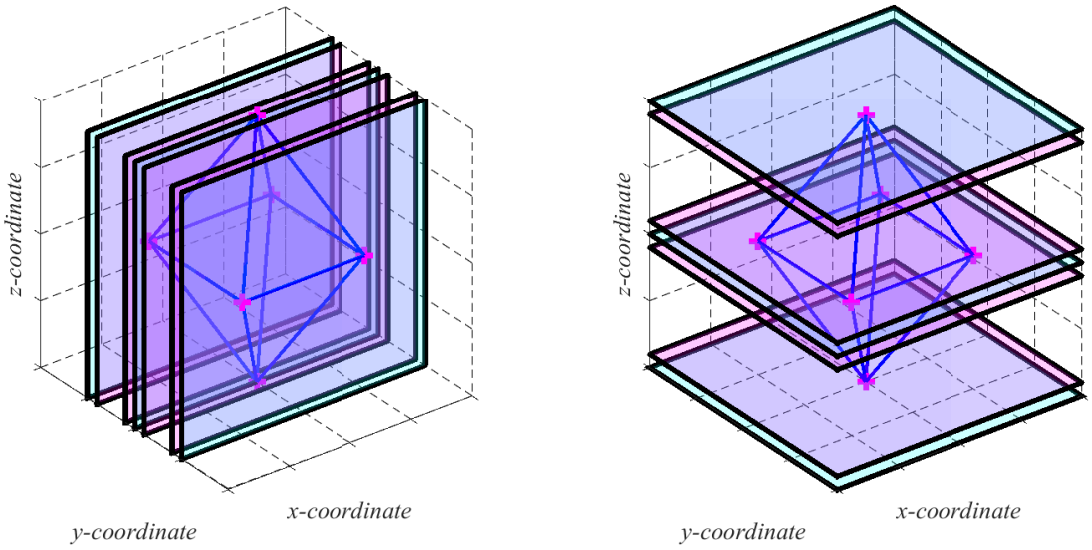
Vertex order	Coordinates			Connection List $N \times N$											
m	x	y	z	1	2	3	4	5	6	7	8	9	10	11	12
1	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0
2	5	0	0	1	0	1	0	0	1	0	0	0	0	0	0
3	5	0	5	0	1	0	1	0	0	1	0	0	0	0	0
4	0	0	5	1	0	1	0	0	0	0	1	0	0	0	0
5	-1.5	4	0	1	0	0	0	0	0	0	1	1	0	0	0
6	6.5	4	0	0	1	0	0	0	0	1	0	0	1	0	0
7	6.5	0	5	0	0	1	0	0	1	0	0	0	0	1	0
8	-1.5	4	5	0	0	0	1	1	0	0	0	0	0	0	1
9	0	8	0	0	0	0	0	1	0	0	0	0	1	0	1
10	5	8	0	0	0	0	0	0	1	0	0	1	0	1	0
11	5	8	5	0	0	0	0	0	0	1	0	0	1	0	1
12	0	8	5	0	0	0	0	0	0	0	1	1	0	1	0

b) Vertex matrix $\boldsymbol{V}$ of hexagon for $\boldsymbol{A}_{xz}$-array

Figure 2.17: Example of flight zone definition by the vertex matrix for y-level discretization

2.3.1.2 z-Level and y-Level Creation for 3D-Flight Area

Subsequently, xy- and xz-planes are introduced. Tolerance planes are additionally created for each of these introduced xy- and xz-planes. These planes are necessary for the online safe area detection algorithm, which is further detailed in Sec. 2.3.2. The online algorithm interpolates the line edges of the three-dimensional zones between the introduced planes, in order to gain the current two-dimensional flight and safety area in the xy-plane and xz-plane. This enables the safe area detection by two point-in-polygon algorithm. The example in Fig. 2.13 shows the current computed two-dimensional flight area in the xy-plane. Fig. 2.14 illustrates the two-dimensional flight and safety area in xy-plane and xz-planes.
The defined flight area is discretized along the y-coordinate with y_p^*-planes and along the z-coordinate (i.e. the altitude) with z_p^*-planes. The y_p^*-planes and the z_p^*-planes are saved in the vector $\boldsymbol{y^*} = \{y_1^*, ..., y_{P_y}^* \mid y_{p_y}^* \in \mathbb{R}, p_y = 1, ..., P_y\}$ and $\boldsymbol{z^*} = \{z_1^*, ..., z_{P_z}^* \mid z_{p_z}^* \in \mathbb{R}, p_z = 1, ..., P_z\}$. A z_p^*-discretization plane is introduced at every z-coordinate, where the total number of vertices within the xy-planes (flight zone with non-flight zones) changes.
Equivalently, a y_p^*-discretization plane is created at every y-coordinate, where the total number of vertices within the xz-planes (flight zone with non-flight zones) changes.
The z_p^*-discretization uses the $\boldsymbol{A}_{xy}$-array. The $\boldsymbol{A}_{xz}$-array is utilized for the y_p^*-discretization in case the planes
Fig. 2.18 shows an example of the introduced y_p^*- and z_p^*-planes with an exemplary octahedron as zone. The mentioned planes are drawn in cyan color. These planes differ from the tolerance planes, marked with magenta color. The tolerance y- and z-layer are introduced to overcome the following main issues, which are caused by the y_p^*- and z_p^*-planes.
Non-flight zones could contact each other, if one zone ends and another starts at the same y-level and/or z-level. This case occurs, for example, in Fig. 2.13(a). There, the non-flight zone at the bottom has the shape of a hexagon with six vertices at its bottom z-plane and top z-plane. A non-flight-zone with the shape of a cuboid begins on top of the hexagon. Hence, the number of vertices changes at the z-plane, where both zones contact. As elaborated further in Sec. 2.3.2, the online algorithm interpolates between the saved z-level polygons, when the aerial vehicle is positioned in-between. It cannot be interpolated between the polygons if the vertices number changes at one z-level and/or y-level since the algorithm does not know, which vertices to choose in the right order at the contacting y-level and/or z-level for the interpolation. A wrong interpolation could result in self-intersecting two-dimensional polygons, which would lead to a wrong trigger of the evasive maneuver.
Another issue arises, if the number of vertices at different y-levels and/or z-levels changes in one zone. An example is the geometrical shape in Fig. 2.15(f). The polygon in the xy-plane changes the number of vertices from four vertices, to three vertices and again to four vertices along the z-coordinate. As mentioned above, at the z-levels, where the number of vertices changes, a valid two-dimensional polygon cannot be guaranteed. The polygons are



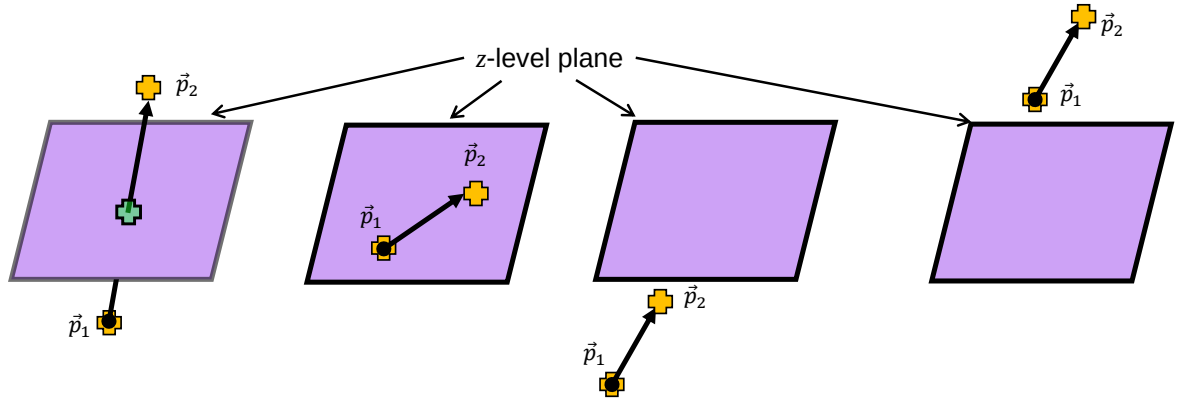
(a) y-levels (b) z-levels

Figure 2.18: Introduction of xz- and xy-planes for an exemplary zone with the shape of an octahedron

not allowed to overlap or contain self-intersections. Invalid polygons would also occur, if the zone's lowest/highest y- or z-plane comprises a single point. At this y- and/or z-plane, the single vertex can not be interpreted as a valid polygon. In Fig. 2.18 an octahedron example is shown. The lowest/highest z-level of the zone contains only a single point. The solution of the listed issues is the introduction of lower and upper tolerance y- and z-planes for each y- and z-layer, as follows. To avoid a single point as a polygon or in general invalid polygons, tolerance y- and z-levels are introduced. In case of a single point, adding or subtracting a small tolerance to the planes results in either a two-dimensional polygon or no polygon. The tolerance planes also solve the problem for adjacent zones. If the aerial vehicle is located in between the two tolerance planes in case of adjacent zones, the polygons of the closest tolerance plane are used as relevant polygons. The introduced dead band can be kept very small such that the approximation is minor and not of practical relevance.
In case of the z-levels, for the lowest plane of the flight area only an upper tolerance plane is introduced and, vice versa for the highest plane only a lower tolerance plane is created. For the y-plane with the lowest y-value of the flight area only a tolerance plane with higher y-value is introduced and, vice versa for the y-layer with the highest value only a tolerance plane with lower y-value is created.
As a result, this algorithm part outputs any occurring unique y- and z-level of the flight area, defined by the vector $\boldsymbol{y} = \{y_1, ..., y_{K_y} |\ y_k \in \mathbb{R}, k_y = 1, ..., K_y\}$ and the vector $\boldsymbol{z} = \{z_1, ..., z_{K_z} |\ z_k \in \mathbb{R}, k_z = 1, ..., K_z\}$. The vector $\boldsymbol{y}$ and $\boldsymbol{z}$ contains the y- and z-values of the shifted y- and z-planes only (shown as magenta planes in Fig. 2.18), and does not comprise the discretization values of the vectors $\boldsymbol{y}^*$ and $\boldsymbol{z}^*$.

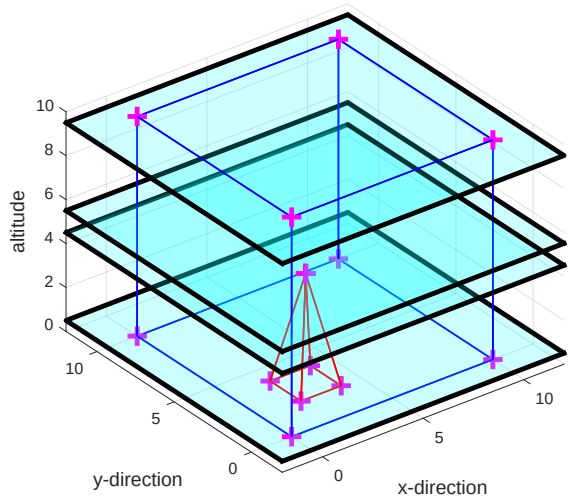
2.3.1.3 Intersection Point Calculation

After the unique y- and z-levels are determined, the intersection points between the flight/safety area and the y- and z-planes are calculated. For each zone defined by $\boldsymbol{V}$ within the matrix arrays $\boldsymbol{A}_{xy}$ and $\boldsymbol{A}_{xz}$ the intersection points between every zone's edge (defined by the straight-line equation between the two vertices p_1 and p_2) and every y-level/z-level saved in the vector $\boldsymbol{y}/\boldsymbol{z}$ are calculated. In case the corresponding line is located within the plane, below or above, the intersection point does not exist and the line is not considered further for the intersection point calculation. This is illustrated in Fig. 2.19.
The mentioned order of the vertices is essential to regain valid polygons with in the y-/z-levels. The intersection points are calculated from the first vertex ascending to the last vertex (i.e. at first, the intersection points are calculated between a y-/z- plane and the lines, which originate from the first vertex).
The calculated intersection points at one specific layer (i.e. y- or z-plane) in the order the points are calculated form again a polygon . For that reason, the three-dimensional flight area is saved in the arrays $\boldsymbol{A}_{xy}$ and $\boldsymbol{A}_{xz}$ for the y-planes and z-planes separately. The correct array definition is substantial in order to gain again a valid polygon, which represents a two-dimensional layer of original flight area. The flight operator should not be involved in the definition since he shall be able to select between already defined correct zones. For some polyhedrons (e.g. octahedron) a correct vertex order is not achievable in the order the intersection points are computed. Therefore, the line order for the calculation of the intersection points is saved in a matrix separately to the vertex matrix $\boldsymbol{V}$. Since intersection points are calculated twice in both directions of each line edge, equivalent intersection points, which have been already calculated, are not taken into account further. The intersection points are saved in the polygon matrix $\boldsymbol{P}$ subsequently for any y- and z-levels and zone. An exemplary matrix $\boldsymbol{P}$ is shown in Fig. 2.20. The approach to calculate the intersection points and finally the matrices $\boldsymbol{P}_{xy}$ and $\boldsymbol{P}_{xz}$ is summarized in Alg. 2.



a) Intersection occurs (b) Vertices lie within plane (c) No intersection (d) No intersection

Figure 2.19: Case illustration, which points are considered for intersection point calculation



a) Flight area with pyramidal non-flight zone (red), illustrated with introduced z-levels and an exaggerated tolerance tol=0.5

x	y	z	i	n	z-layer	Area
0	0	0.50	0	1	1	Cuboid
10.00	0	0.50	0	2		
10.00	10.00	0.50	0	3		
0	10.00	0.50	0	4		
2.10	2.10	0.50	1	1	1	Pyramid
3.90	2.10	0.50	1	2		
3.90	3.90	0.50	1	3		
2.10	3.90	0.50	1	4		
0	0	4.50	0	1	2	Cuboid
10.00	0	4.50	0	2		
10.00	10.00	4.50	0	3		
0	10.00	4.50	0	4		
2.90	2.90	4.50	1	1	2	Pyramid
3.10	2.90	4.50	1	2		
3.10	3.10	4.50	1	3		
2.90	3.10	4.50	1	4		
0	0	5.50	0	1	3	Cuboid
10.00	0	5.50	0	2		
10.00	10.00	5.50	0	3		
0	10.00	5.50	0	4		
0	0	9.50	0	1	4	Cuboid
10.00	0	9.50	0	2		
10.00	10.00	9.50	0	3		
0	10.00	9.50	0	4		

b) Example of polygon matrix $\boldsymbol{P}_{xy}$ with flight zone ($i = 0$) and non-flight zone ($i = 1$)

Figure 2.20: Polgyon matrix for illustrated flight area

Algorithm 2 Intersection point algorithm for z-levels

1: Input: Flight area $\boldsymbol{A}_{xy}$
2: Extract all unique z-planes from flight area $\boldsymbol{A}_{xy}$
3: Add/ subtract the defined tolerance tol to the z-planes
4: Save generated z-planes in vector $\boldsymbol{z}$ (see 2.3.1.2)
5: **for** Select kth z-Level of vector $\boldsymbol{z}$ from first to last **do**
6: **for** Select ith flight zone from first to last **do**
7: Calculate intersection points between the edges of the ith flight zone and the kth z-level using the vertices matrix $\boldsymbol{V}_i$ of the ith flight zone and the vector $\boldsymbol{z}$ (see Alg. 3)
8: **end for**
9: Add x, y, z-coordinates of intersection points with zone identification number i to polygon matrix $\boldsymbol{P}_{xy}$ of the ith flight zone (see polygon matrix $\boldsymbol{P}_{xy}$ in Fig. 2.20)
10: **end for**
11: Output: Polygon matrix $\boldsymbol{P}_{xy}$

Algorithm 3 Calculation intersection points for z-layers

```
Input: Vertex Matrix V_i, z-level z_k, zone identification number i
for Select first dimension of connection matrix do
    for Select second dimension of connection matrix do
        if Entry of connection matrix is true then
            Calculate directional vector and reference point of the zones edge
            if z-level z_k between the two points of the zone's edge AND not within z-level z_k
            AND directional vector points in positive z-direction (see Fig. 2.19) then
                Calculate intersection point
            end if
        end if
    end for
end for
Output: x,y,z-coordinates of intersection points between kth z-level and the polygon
edges of ith flight zone
```

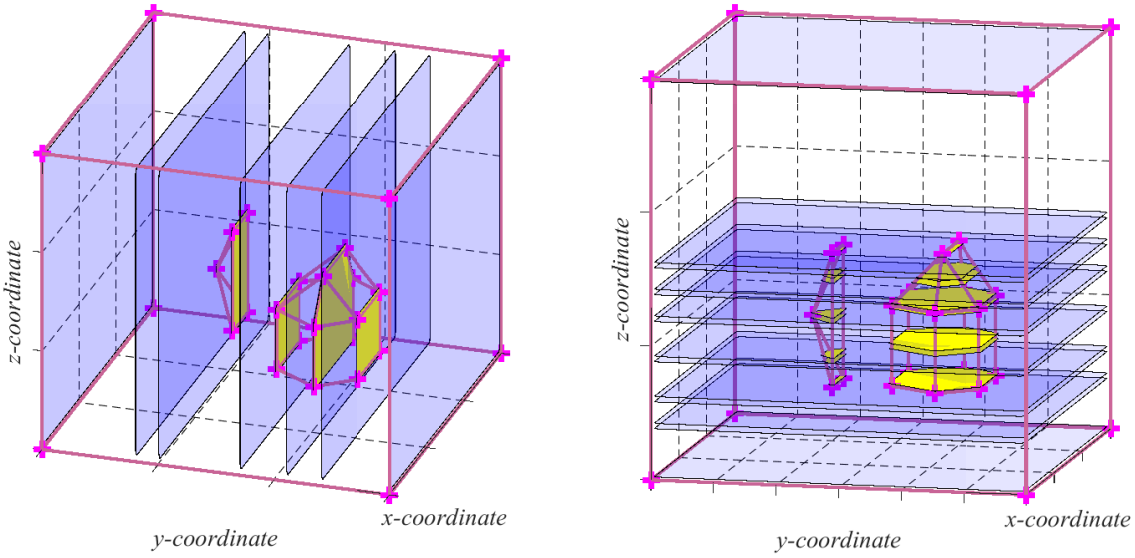
2.3.1.4 Safety Area Generation for y- and z-Levels

With the introduction of y- and z-levels the safety area generation is mapped from the three-dimensional case to two two-dimensional cases. For any calculated y- and z-plane and their corresponding zones (defined by the above mentioned calculated intersection points), two-dimensional safety areas are generated according to the two-dimensional case, explained in Sec. 2.2.2. The results are visualized in Fig. 2.21. The figure shows an exemplary flight zone with the shape of a cuboid and two non-flight zones with the shape of a pyramid and a more complex hexagonal prism. Fig. 2.21(a) and Fig. 2.21(b) show the introduced y-levels and z-levels in purple color. The two-dimensional zones of the generated safety area for each plane are illustrated in yellow color.

The two-dimensional flight zones with non-flight zones and the respective safety areas are stored in the polygon matrices (e.g. $\boldsymbol{P}_{xy}$ for the z-levels).

As a result, four different polygon matrices are saved. For each y-layer as well as for each z-layer the polygon matrices of the flight area and the generated safety area are saved separately. If the aerial vehicle is in-between these planes the online part of the algorithm interpolates between these zones.

The size of the two-dimensional safety margin is different in the vertical and the lateral case. The required worst-case safety margins with respect to the vertical and lateral evasive maneuvers are elaborated in detail in Sec. 2.4.



(a) y-levels (b) z-levels

Figure 2.21: Generated two-dimensional safety areas within the introduced xz- and xy-planes

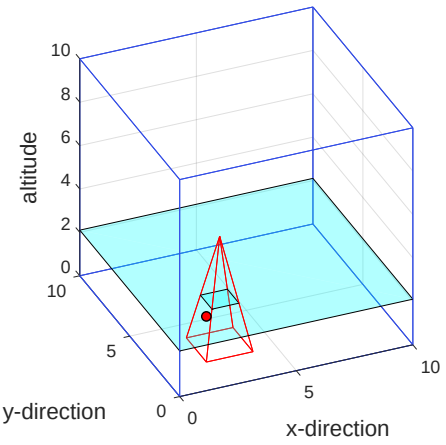
2.3.2 Online Safe Area Detection Algorithm For Three-Dimensional Space

The calculation of the y- and z-levels and the polygons at these y- and z-levels are performed offline and transmitted to the UAV before flight. This information is sufficient for the online part of the proposed algorithm. Utilizing this data the online algorithm is able to monitor the location of the aerial vehicle with respect to predefined three-dimensional flight area. The algorithm calculates the current lower and upper y- and z-layer. The vertices of relevant y- and z-planes are forwarded to the detection algorithm itself. At every calculation cycle, the current two-dimensional polygons defined by xz-coordinates at the y-coordinate and xy-coordinates at the z-coordinate respectively of the aerial vehicle are computed by interpolating between the connected points of the two relevant y- and z-planes. Using the calculated polygons (contour lines), two two-dimensional point-in-polygon algorithms are applied to detect if the aircraft is within or outside of the flight area and the safety area in horizontal and vertical direction. The procedure of the online algorithm becomes visible in the previously shown Fig. 2.13 and Fig. 2.14. The red circle marks the position of the aerial vehicle. The light blue plane represents the xy-plane at the altitude level z of the aerial vehicle. The black edges in the z-level-plane illustrate the two-dimensional flight zone together with the non-flight zones (dark blue). These inner and outer contours represent the relevant polygons for the two-dimensional point-in-polygon algorithm for the horizontal boundaries. The online algorithm is comprised in Alg. 4 and further detailed in the next subsections.

2.3.2.1 Polygon Interpolation

The two-dimensional polygons at the current position y_{pos} in y-direction and the altitude z_{pos} of the aerial vehicle are computed for the point-in-polygon algorithm from the polygon matrix (e.g. Fig. 2.20). For this purpose, the current y-layer in the back and in front of the aerial vehicle, and equivalently the lower and upper z-layer of the aerial vehicle are calculated. The vertices at the aerial vehicle position are determined from these layers using the polygon matrices (e.g. $\boldsymbol{P}_{xy}$).
At this step an important differentiation within the proposed algorithm is performed. If the aerial vehicle is located close (i.e. smaller than the tolerance tol) to the rear/front y-layer or a lower/upper z-layer, the polygon vertices of the rear/front y-layer or lower/upper z-layer are used for the two-dimensional point-in-polygon algorithm. If the distance is greater than the tolerance tol, an interpolation between the polygon vertices of the rear/back y-layer or lower/upper z-layer are performed. This differentiation is necessary, since the number of vertices per plane can change. The number of vertices between y-/z-planes changes if multiple zone edges meet in one vertex. As mentioned before, an example is the geometrical shape in Fig. 2.15(f). In this example, the polygon in the xy-plane changes the number of vertices along the z-coordinate. Since the interpolation between the z-layers can only be conducted if the same number of vertices in the lower and the upper z-layer exists, the introduced tolerance planes in the offline part support this part of the algorithm, too. If the aircraft is located within the defined tolerance band tol, the vertices of that z-layer are used. If the aircraft is located further away, the interpolation is conducted.
The comprehensive structure of the polygon matrices allow an interpolation of the zones' edges of the current rear/front y-layer and/or lower/upper z-layer. This is illustrated by the polygon matrix in Fig. 2.20 and the interpolated polygon matrix at the current z-layer z_{pos} in Fig. 2.22, which is used for the two-dimensional point-in-polygon algorithm (see next subsection). The polygon matrix in Fig. 2.20 contains every vertex transmitted from the offline calculation.
In contrast, Fig. 2.22 comprises the two-dimensional polygons, which are interpolated from the polygon matrix in Fig. 2.20. For the interpolation, at first the current lower and upper z-layer are identified. Next, using the current z-position of the aerial vehicle the polygons between the vertices of the lower and upper z-layer are interpolated. For example, if the aircraft is located between the z-layer 1 and 2 in the shown flight area (see Fig. 2.20), the polygons of the zone $i = 0$ and $i = 1$ at the current altitude z_{pos} of the aircraft are interpolated. For instance, the vertex $n = 2$ of the non-flight zone $i = 1$ at the aircraft altitude z_{pos} can be determined by the linear interpolation of vertex $n = 2$ by using the vertex coordinates of the z-layer 1 and 2 and the aircraft altitude z_{pos}. The interpolation is conducted for any vertex of the z-layers. Consequently, the polygons at the current z-position are obtained by calculating the intersection between the straight lines (i.e. the zone edges from the lower to the upper

z-level) and the current z-position. The polygon validity is still ensured, since the order is fixed in the polygon matrix, and the order of the polygons at the current z-level during the interpolation does not change either. The output are the calculated vertices of the flight zone and non-flight zones at the current z-position.



a) Same flight area as in Fig. 2.20 with two- dimensional polygons (visualized within cyan plane)

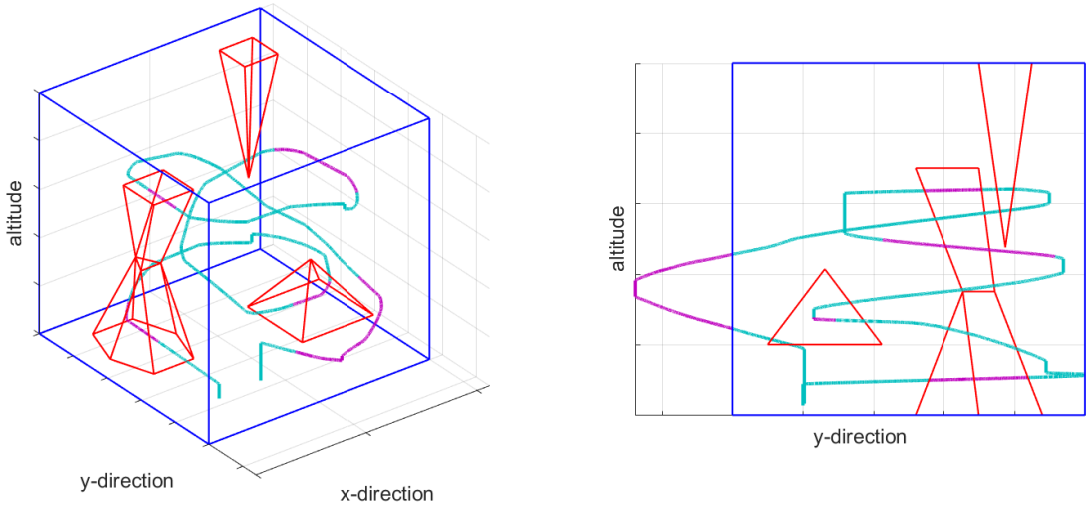
x	y	z	i	n	Description
0	0	2.1033	0	1	Vertices
10.0000	0	2.1033	0	2	of flight area
10.0000	10.0000	2.1033	0	3	polygon
0	10.0000	2.1033	0	4	(outer rectangle)
2.4207	2.4207	2.1033	1	1	Vertices
3.5793	2.4207	2.1033	1	2	of non-flight zone
3.5793	3.5793	2.1033	1	3	polygon
2.4207	3.5793	2.1033	1	4	(rectangle in pyramid)

b) Interpolated two-dimensional polygon matrix for point-in-polygon algorithm

Figure 2.22: Example of two-dimensional polygon definition at the aircraft position z_{pos}

Algorithm 4 Online algorithm for z-layers

1: Input: Polygon matrix $\boldsymbol{P}$, z-level vector $\boldsymbol{z}$, current aerial vehicle position z_{pos}
2: Determine current upper and lower z-level using vector $\boldsymbol{z}$ and the current z-position of the aerial vehicle z_{pos}
3: Extract current upper and lower polygons from polygon matrix $\boldsymbol{P}$
4: **if** Absolute difference between lower or upper z-level and current z-position is smaller than the introduced tolerance tol **then**
5: Use polygon vertices of lower or upper z-level respectively
6: **else**
7: Calculate polygon vertices at current z-level by linear interpolation between vertices of the lower and upper z-level
8: **end if**
9: Combine polygons to two-dimensional flight area (see Fig. 2.22)
10: Identification, if aerial vehicle is inside or outside of the flight area by two-dimensional point-in-polygon algorithm (see Sec. 2.3.2.2)
11: Output: Inside/ outside of three-dimensional flight area (see Fig. 2.23)



(a) Flight path of aerial vehicle within complex flight area

(b) Flight path in y-altitude view

Figure 2.23: Model-in-the-loop verification: outside (magenta), inside (cyan)

2.3.2.2 Point-in-Polygon Algorithm

As now the three-dimensional problem is reduced to two two-dimensional cases, two simultaneously executed point-in-polygon algorithms for the xy- and the xz-plane as proposed in Sec. 2.2.3.1 may be used to determine, if the aerial vehicle is inside or outside of the flight area and the safety area. The *crossing number* algorithm counts the number of intersections from the aircraft position in an arbitrary direction. If the number of intersections between the arbitrary direction and the polygon edges is odd, the aerial vehicle is inside the two-dimensional flight area, otherwise the aerial vehicle is outside. Due to the interpolation above, the proposed algorithm transforms the three-dimensional case to two two-dimensional problem to enable the utilization of the *crossing number* algorithm. The approach allows the detection by a simple implementation with low computational effort and allows even non-convex polygons (e.g. non-flight zones).

2.3.2.3 Model-in-the-Loop Verification

The proposed algorithm is verified in model-in-the-loop simulations. The results for a complex flight area are shown in Fig. 2.23. The flight path of an exemplary vertical take-off and landing vehicle is shown. The color of the flight path marks the output of the proposed safe area detection algorithm. If the aerial vehicle is *outside* of the flight area (i.e. outside of the flight zone or inside the non-flight zones), the flight path is marked magenta. If the aerial vehicle is *inside*, the flight path is illustrated in cyan color. The proposed algorithm performs as desired, if the flight area fulfills the requirements in Sec. 2.3.1.1.

2.4 Evasive maneuver[6]

This section describes safety margin calculations and the design of different evasive maneuvers to prevent the aircraft from leaving the flight area. It is important to note, that the evasive maneuvers are designed to achieve a designated acceleration in the respective direction. That means, the lateral maneuver shall achieve a certain lateral load factor $n_{des,L}$ and the vertical maneuver shall reach the specified load factor $n_{des,V}$. The combined maneuver shall reach the lateral and vertical accelerations simultaneously. The subsequent Sec. 2.4 is based on the work from the author in Ref. [75].
Aircraft specific limitations and its flight dynamics need to be considered for the design of the evasive maneuver. Therefore, aircraft flight performance and autopilot constraints are discussed first. Subsequently, the equations of motion are derived. Sec. 2.4.3 details the considered modes within the presented geofencing algorithm. Finally, the evasive maneuver for the lateral and vertical case are introduced. The last subsection discusses a combined lateral and vertical maneuver, which occurs in the rare case, if flight area boundaries in the lateral and vertical case are violated simultaneously. The derived considerations in this section are applied to model-in-the-loop simulations of a general aviation aircraft in the Sec. 2.5.

2.4.1 Constraints

For the evasive maneuver design, the aircraft airworthiness limitations need to be taken into account to ensure a safe operation of the aircraft. The following aircraft specific constraints are considered:

- Angle of Attack: $\alpha_{min} < \alpha < \alpha_{max}$
- Load Factor: $n_{z,min} < n_z < n_{z,max}$
- Climb Angle: $\gamma_{min} < \gamma < \gamma_{max}$
- Airspeed: $V_{A,min} < V_A < V_{A,max}$

Further aircraft/envelope constraints (e.g. engine constraints) can be considered in the future. In addition to the basic flight constraints above, the following autopilot limitations are taken into consideration in order to support the aircraft flight limits:

- Bank Angle: $-\phi_{max} < \phi_{cmd} < \phi_{max}$
- Pitch Angle: $\theta_{min} < \theta_{cmd} < \theta_{max}$

If these constraints are considered properly within the evasive maneuver design, the geofencing module is able to bring the aircraft back into the safety area without exceeding the aircraft airworthiness limitations.

[6] This section represents a revised part of the work in Ref. [75], which has been published by the author.

2.4.2 Equations of Motion

In order to calculate the required distance, which is necessary to conduct a certain evasive maneuver in vertical and lateral direction, the equation of motions are derived. As elaborated further in the next sections, the lateral maneuver comprises a turn flight with constant bank angle ϕ. The vertical evasive maneuver is designed as a pull-up or push-down maneuver with constant climb angle rate $\dot{\gamma}$. These maneuvers are designed to be conducted separately or in combination.
The translational equations of motion in an arbitrary coordinate frame n can be written as:

$$\left(\left(\frac{d\vec{V}_k}{dt} \right)_n^g + \left(\vec{\Omega}^{ng} \right)_n \times \left(\vec{V}_K \right)_n \right) = \frac{1}{m} \underline{\boldsymbol{M}}_{nA} \begin{bmatrix} X^A \\ Y^A \\ Z^A \end{bmatrix}_A + \frac{1}{m} \begin{bmatrix} X^P \\ Y^P \\ Z^P \end{bmatrix}_n + \underline{\boldsymbol{M}}_{ng} \vec{\boldsymbol{G}}_g \tag{2.5}$$

with the angular rate vector $\vec{\Omega}_n^{ng}$ between the geodetic frame g and the arbitrary frame n;
with the kinematic velocity $\left(\vec{V}_K\right)_K = \begin{bmatrix} V_K & 0 & 0 \end{bmatrix}_K^T$;
with the aerodynamic force $\vec{\boldsymbol{A}} = \begin{bmatrix} X^A & Y^A & Z^A \end{bmatrix}_A^T$ given in the aerodynamic frame A;
with the propulsion force $\vec{\boldsymbol{P}} = \begin{bmatrix} X^P & Y^P & Z^P \end{bmatrix}_n^T$;
with the gravity vector $\vec{\boldsymbol{G}}_g$;
and the transformation matrices $\underline{\boldsymbol{M}}_{nA}$ and $\underline{\boldsymbol{M}}_{nG}$.
For the elaboration of the minimum turn radius for coordinated turn climb flight and the pull-up/push down maneuver, the most appropriate coordinate frame is the kinematic frame (with the kinematic course χ and the climb angle γ), designated as K. The evaluation at this coordinate frame allows the direct derivation of the load factor and the minimum turn radius. The angular rate $\left(\vec{\Omega}^{Kg}\right)_K$ between the kinematic frame K and the geodetic frame g can be derived as followed:

$$\left(\vec{\Omega}^{Kg}\right)_K = \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \dot{\chi} \end{bmatrix} + \begin{bmatrix} 0 \\ \dot{\gamma} \\ 0 \end{bmatrix} = \begin{bmatrix} -\sin\gamma\dot{\chi} \\ \dot{\gamma} \\ \cos\gamma\dot{\chi} \end{bmatrix} \tag{2.6}$$

This leads to the expression:

$$\left(\vec{\Omega}^{Kg}\right)_K \times \left(\vec{V}_K\right)_K = \begin{bmatrix} -\sin\gamma\dot{\chi} \\ \dot{\gamma} \\ \cos\gamma\dot{\chi} \end{bmatrix} \times \begin{bmatrix} V_K \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ \cos\gamma\dot{\chi}V_K \\ -\dot{\gamma}V_K \end{bmatrix} \tag{2.7}$$

Using the term $\left(\vec{\Omega}^{Kg}\right)_K \times \left(\vec{V}_K\right)_K$ the equations of motion in the kinematic frame are finally

$$\begin{bmatrix} \dot{V}_K \\ \cos\gamma\dot{\chi}V_K \\ -\dot{\gamma}V_K \end{bmatrix} = \frac{1}{m}\underline{\boldsymbol{M}}_{KA}\begin{bmatrix} X^A \\ Y^A \\ Z^A \end{bmatrix}_A + \frac{1}{m}\begin{bmatrix} X^F \\ Y^F \\ Z^F \end{bmatrix}_K + \begin{bmatrix} \cos\gamma & 0 & -\sin\gamma \\ 0 & 1 & 0 \\ \sin\gamma & 0 & \cos\gamma \end{bmatrix}\begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{2.8}$$

with $\left(\frac{d\vec{V}_K}{dt}\right)_K^g = \begin{bmatrix} \dot{V}_K & 0 & 0 \end{bmatrix}^T$.

Replacing the aerodynamic force $\vec{\boldsymbol{A}} = \begin{bmatrix} -D & 0 & -L \end{bmatrix}^T$ with the drag force D and the lift force L, the assumed propulsion force $\vec{\boldsymbol{P}} = \begin{bmatrix} T & 0 & 0 \end{bmatrix}^T$ with the thrust T and the transformation matrices $\underline{\boldsymbol{M}}_{KA}$ and $\underline{\boldsymbol{M}}_{Kg}$ leads to:

$$\begin{bmatrix} \dot{V}_K \\ \cos\gamma\dot{\chi}V_K \\ -\dot{\gamma}V_K \end{bmatrix} = \frac{1}{m}\begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\mu & -\sin\mu \\ 0 & \sin\mu & \cos\mu \end{bmatrix}\begin{bmatrix} -D \\ 0 \\ -L \end{bmatrix} + \frac{1}{m}\begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}_K + \begin{bmatrix} -\sin\gamma \\ 0 \\ \cos\gamma \end{bmatrix} g \tag{2.9}$$

and finally to:

$$\begin{bmatrix} \dot{V}_K \\ \cos\gamma\dot{\chi}V_K \\ -\dot{\gamma}V_K \end{bmatrix} = \frac{1}{m}\begin{bmatrix} -D \\ L\sin\mu \\ -L\cos\mu \end{bmatrix} + \frac{1}{m}\begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix}_K + \begin{bmatrix} -\sin\gamma \\ 0 \\ \cos\gamma \end{bmatrix} g \tag{2.10}$$

Note that wind is not considered in the presented derivation of the equations of motions. It becomes apparent in Eq. (2.9) , as the transformation matrix between the kinematic frame (K) and the aerodynamic frame (A) is assumed to be:

$$\underline{\boldsymbol{M}}_{KA} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\mu & -\sin\mu \\ 0 & \sin\mu & \cos\mu \end{bmatrix} \tag{2.11}$$

with the aerodynamic and kinematic bank angle $\mu_A = \mu_K = \mu$ under consideration of no wind. Wind can be considered in future by replacing the transformation matrix with $\underline{\boldsymbol{M}}_{KA} = \underline{\boldsymbol{M}}_{KB}(\mu_K, \alpha_K, \beta_K)\underline{\boldsymbol{M}}_{BA}(\alpha_A, \beta_A)$. B represents thereby the frame of the aircraft body axes and it is differentiated between the kinematic and aerodynamic angle of attack α_K, α_A as well as angle of sideslip β_K, β_A (see Ref. [76]).

The equations in Eq. (2.10) can be written in scalar form as:

$$m\dot{V}_K = (-D) + T - mg\sin\gamma \tag{2.12}$$

$$\cos\gamma\dot{\chi}mV_K = L\sin\mu \tag{2.13}$$

$$-mV_K\dot{\gamma} = -L\cos\mu + mg\cos\gamma \tag{2.14}$$

Eq. (2.12) - Eq. (2.14) are used as the basis for the design of the evasive maneuvers in the next subsections.
As stated, the subsequent derivation of the safety margin calculation does not take wind into account. However, future work can replace inter alia the transformation matrix, and derive the safety margins under wind considerations.

2.4.3 Control Logic for Maneuver Trigger

As detailed in the next sections, the presented geofencing algorithm reuses standard autopilot functionalities to command and control specific evasive maneuvers in order to prevent the aerial vehicle from leaving the flight area. For the vertical and the lateral violation, three different geofencing modes are considered. The trigger of the lateral and vertical maneuver are handled independently. That means, the conditions for the lateral and vertical case are different.
The first mode is called the *default* mode. This mode is active if the aerial vehicle operates as usual in the flight area without any zone violations. Within the *default* mode, the geofencing is inactive and does not effect the operation of the aerial vehicle.
If the aerial vehicle flies into the safety margin (i.e. leaves the safety area), the mode switches to the *evasive* mode. In the lateral case, the heading controller becomes active to execute the evasive maneuver. If the aerial vehicle enters the vertical safety margin, the altitude channel of the autopilot becomes active. The command values to the autopilot are explained in the next section. As long as the aerial vehicle is located within these safety margins, the geofencing algorithm stays within the *evasive* mode.
The evasive maneuvers are designed such that the aerial vehicle leaves the safety margin and returns to the safety area. If the aerial vehicle reenters the safety area, the *returned* mode becomes active. In the lateral case, it is switched from the heading controller to a bank controller for wings-level flight.
If the aerial vehicle returns back to the safety area in the vertical case, it is switched to a pitch controller and pitch angle is commanded for a shallow climb or descend.
The *return* mode is active as long as the flight operator does not give any new inputs. If the geofencing mode is *returned*, the operator can give new autopilot commands and the geofencing mode switches back to *default*. If the *evasive* mode is active, the flight operator cannot intervene. In the next sections, the evasive maneuvers are designed, such that the evasive maneuver can be conducted, as required, in vertical and lateral case simultaneously or successively.

2.4.4 Lateral Case

If the safety area of the xy-plane is violated, a lateral maneuver is initiated. The maneuver is designed as steady coordinated turn with constant kinematic bank angle μ and climb angle γ. In Sec. 2.5.1, the maneuver is exemplary controlled by a cascaded waypoint controller, with an outer-loop heading and an inner-loop bank controller. The created waypoint and the maneuver itself are illustrated in Fig. 2.24. The waypoint creation is explained in Sec. 2.2.3.2. This section derives the required safety margin for the lateral maneuver by calculating the necessary turn radius.
With $\dot{\gamma} = 0$ and the division from Eq. (2.13) by Eq. (2.14) we get the turn rate as a function of the kinematic speed V_K and the kinematic bank angle μ:

$$\dot{\chi} = \frac{g}{V_K} \tan \mu \tag{2.15}$$

From Eq. (2.14) we can calculate the load factor n_z as a function of the climb angle γ and the kinematic bank angle μ:

$$\frac{\cos \gamma}{\cos \mu} = \frac{L}{mg} = n_z \tag{2.16}$$

With the trigonometric relation

$$\tan^2 \mu = \frac{1}{\cos^2 \mu} - 1 \qquad \Longleftrightarrow \qquad \sqrt{1 + \tan^2 \mu} = \frac{1}{\cos \mu} \tag{2.17}$$

and Eq. (2.15), the turn rate can be calculated as a function of the climb angle γ and the load factor n_z:

$$\dot{\chi} = \frac{\sqrt{n_z^2 - \cos^2 \gamma}}{V_K \cos \gamma} g \tag{2.18}$$

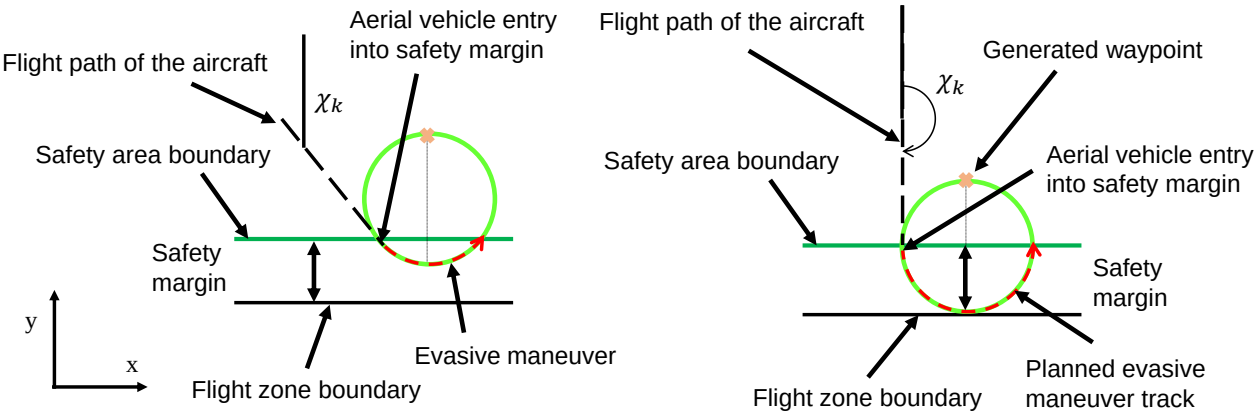


Figure 2.24: Design of the lateral maneuver

With the kinematic relation

$$\dot{\chi}\, R_L = V_K \cos\gamma \tag{2.19}$$

the turn radius R_L can be calculated by:

$$R_L = \frac{V_K \cos\gamma}{\dot{\chi}} = \frac{V_K \cos\gamma}{\frac{g}{V_K}\tan\mu} = \frac{V_K^2 \cos\gamma}{g\tan\mu} \tag{2.20}$$

The turn radius is limited by the load factor constraint $n_{z,max}$

$$R_L = \frac{V_K^2 \cos^2\gamma}{g\sqrt{n_{z,max}^2 - \cos^2\gamma}} \tag{2.21}$$

and the maximum angle of attack α_{max}:

$$R_L = \frac{V_K^2 \cos^2\gamma}{g\sqrt{\left(\frac{L_{max}}{mg}\right)^2 - \cos^2\gamma}} = \frac{V_K^2 \cos^2\gamma}{g\sqrt{\left(\frac{\rho V_A^2 C_{L,max}}{2mg}\right)^2 - \cos^2\gamma}} = \frac{V_K^2 \cos^2\gamma}{g\sqrt{\left(\frac{\rho V_A^2 C_L(\alpha_{max})}{2mg}\right)^2 - \cos^2\gamma}} \tag{2.22}$$

As Eq. (2.20) highlights, the worst-case turn radius R_L occurs at the highest speed V_{max} and the climb angle $\gamma = 0$ under the consideration of a velocity range, which enables the conduction of the evasive maneuvers at desired load factors. Note that $C_{L,max}$ defines in the present case the maximal lift derivative for maneuvering, which occurs significantly before stalling.
In case of the worst-case wind speed V_W, which is tailwind for the lateral case, the turn radius becomes:

$$R_L = \frac{(V_A + V_W)^2 \cos^2\gamma}{g\sqrt{\left(\frac{L_{max}}{mg}\right)^2 - \cos^2\gamma}} = \frac{(V_A + V_W)^2 \cos^2\gamma}{g\sqrt{\left(\frac{\rho V_A^2 C_L(\alpha_{max})}{2mg}\right)^2 - \cos^2\gamma}} \tag{2.23}$$

The worst-case turn radius R_L is used as quantity for the definition of the lateral safety margin, which is necessary to generate the safety areas in the horizontal z-planes. The lateral maneuver is designed to achieve a desired lateral load factor $n_{des,L}$ by a specific bank angle command, which causes a specific turn radius in the lateral plane. As a result, the desired lateral load factor defines the necessary turn radius for the maneuver and thus, the size of the safety margin. The lateral maneuver is applied for model-in-the-loop simulations in Sec. 2.5.1. The analysis with regard to wind is not within the scope of this thesis.

2.4.5 Vertical Case

If the safety area in the vertical case is violated, a pull-up maneuver or push-down maneuver is initiated. The maneuver is designed assuming a constant climb angle rate $\dot{\gamma} = const$ and small bank angles ϕ. The combined maneuver in the next subsection elaborates the pull-up/push-down maneuver with higher bank angles. The maneuver is controlled by a cascaded altitude and pitch angle controller.
The maneuver consists of three phases. In the first phase, the aircraft descends or climbs arbitrarily within flight envelope. After the safety area violation in the vertical plane, the second phase starts and consists of the pull-up/push-down maneuver. When the aerial vehicle reenters the safety area, a shallow climb is initiated by an appropriate pitch angle command. The maneuver is visualized in Fig. 2.25. In the following, the required margin for the pull-up/push-down phase is calculated. With the assumption $\dot{\gamma} = const$ and $\mu = 0$, we can derive from Eq. (2.14) and Eq. (2.16) the following equation:

$$\dot{\gamma} = \frac{g}{V_K}\Big(n_z - \cos\gamma\Big) \tag{2.24}$$

By using the kinematic relationship,

$$\dot{\gamma} = \frac{V_K}{R_V} \tag{2.25}$$

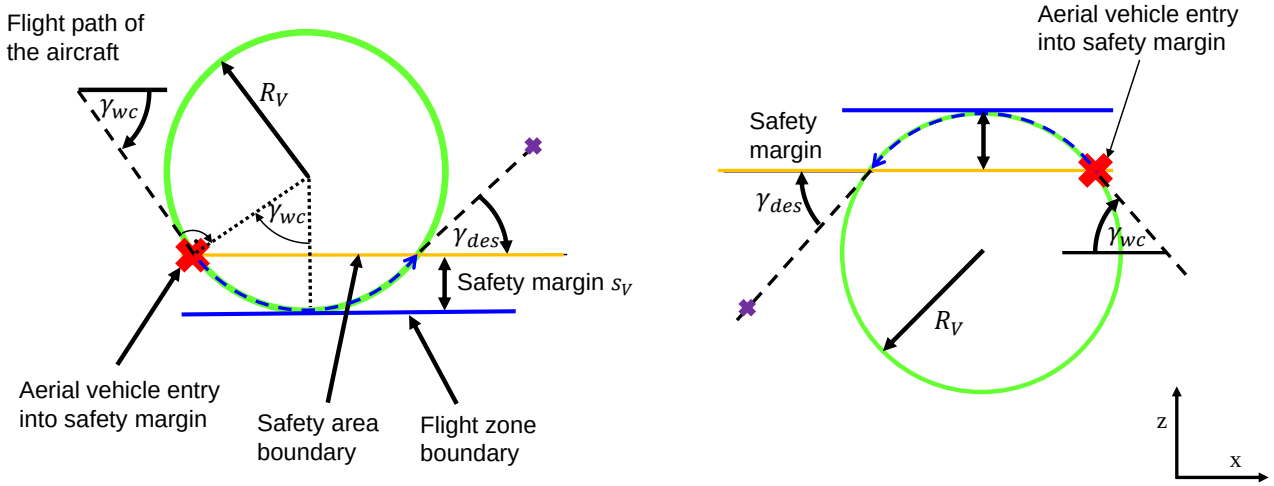
we can calculate the necessary radius for the pull-up/push-down maneuver by:

$$R_V = \frac{V_K^2}{(n_z - \cos\gamma)g} \tag{2.26}$$

Under consideration of Fig. 2.25 the necessary margin is calculated by:

$$s_V = R_V(1 - \cos\gamma) \tag{2.27}$$

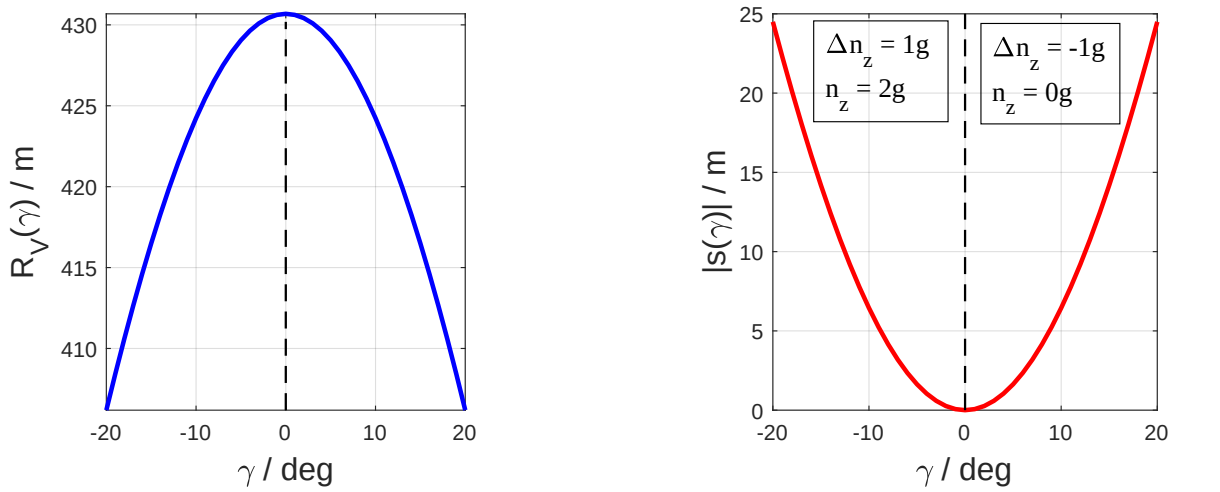
As Eq. (2.27) shows, the required margin s_V in the worst case occurs at the highest speed V_K and the allowed worst-case climb angle γ_{WC}. Thereby, a velocity range should be taken into account such that the desired load factor is achievable. The worst case becomes visible in Fig. 2.26, which illustrates the functions $R_V(\gamma)$ and $s_V(\gamma)$ in Eq. (2.26) and Eq. (2.27).
The computed margin s_V in the worst case is considered as safety margin for the generation of the safety areas in the vertical planes. Similarly to the lateral maneuver, the vertical maneuver is designed to achieve a certain vertical load factor $n_{des,V}$. Consequently, the desired vertical load factor defines the required safety margin by respecting the worst-case speed and climb angle.



(a) Pull-up maneuver

(b) Push-down maneuver

Figure 2.25: Design of the vertical maneuver



(a) Vertical turn radius R_V

(b) Necessary margin s for vertical evasive maneuver

Figure 2.26: Illustration of vertical turn radius R_V and necessary vertical margin s_V in dependence of γ under consideration of the velocity V_K =65 m/s and $|\Delta n_{des,V}| = 1$

2.4.6 Combined Lateral and Vertical Case

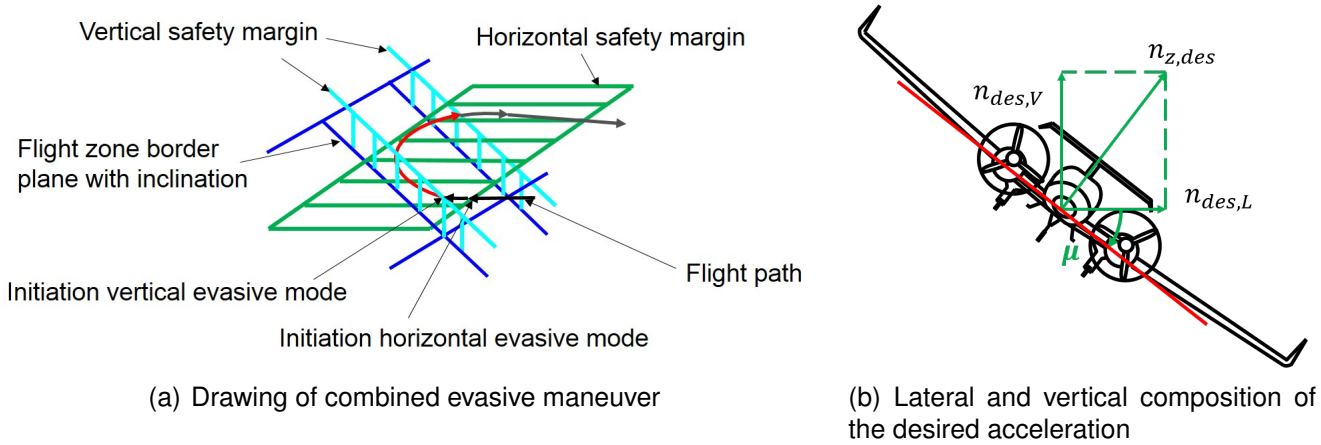
The following discussed maneuver consists of the pull-up maneuver in combination with the lateral maneuver. The combination of the push-down and lateral maneuver is not in the scope of this thesis. The combined maneuver occurs if the safety area is violated in the vertical and in the lateral plane simultaneously. Both maneuvers could be triggered at the same time or successively in different order. Fig. 2.27(a) visualizes an exemplary combined maneuver when the aerial vehicle hits an inclined plane boundary. After a turn is initiated, caused by violation of the lateral safety margin, a pull-up maneuver is triggered, since, while turning, the aerial vehicle also enters the vertical safety margin.
Subsequently, it is explained, what consequences result, when the pull-up and lateral maneuver are executed simultaneously and how the maneuvers are adapted in order to reach the desired lateral and vertical load factor during the maneuver. The combination of the push-down maneuver and the lateral maneuver is open for future work. From Eq. (2.14) the climb angle rate with kinematic bank angle μ is determined by:

$$\dot{\gamma} = \frac{g}{V_K}\Big(n_z \cos\mu - \cos\gamma\Big) \tag{2.28}$$

Using Eq. (2.13) and Eq. (2.14) the turn rate can be calculated by:

$$\dot{\chi} = \frac{\tan\mu\Big(V_K\dot{\gamma} + g\cos\gamma\Big)}{\cos\gamma\, V_K} \tag{2.29}$$

The equations expose two important phenomena of the combined pull-up and lateral maneuver. The turn rate $\dot{\chi}$ increases, when the pull-up (i.e. $\dot{\gamma} > 0$) is performed and the aerial vehicle turns with an established bank angle. At the same time, the achieved climb angle rate $\dot{\gamma}$ is reduced compared to the vertical maneuver without bank angle, since the aircraft is accelerated not only upwards, but also sidewards due to the established bank angle.
If the bank angle is high it is evident that the aircraft does not reach the desired climb angle rate, because the acceleration causes mainly a turn rate.
Although it is adverse, if high bank angles cause a reduced climb angle rate during the pull-up maneuver, the observations mentioned above are nevertheless used to achieve desired lateral accelerations $n_{des,L}$ and desired vertical accelerations $n_{des,V}$. Since the pull-up maneuver provokes in addition a higher turn rate (and thus, in theory, a smaller turn radius), the maneuver is designed such that the bank angle is reduced during the maneuver. The bank angle reduction leads to a lower turn rate. This fact is equalized by the pull-up maneuver, which causes a higher turn rate. The reduced bank angle provokes at the same time a higher vertical acceleration. The maneuver is designed to reach the desired lateral acceleration $n_{des,L}$ and the vertical acceleration $n_{des,V}$. If both maneuvers are triggered, the maximal commanded bank angle is reduced.



(a) Drawing of combined evasive maneuver

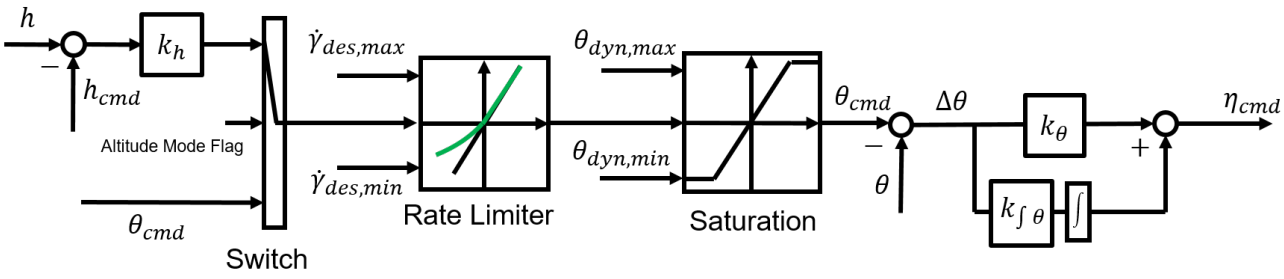
(b) Lateral and vertical composition of the desired acceleration

Figure 2.27: Illustration of combined pull-up and lateral maneuver

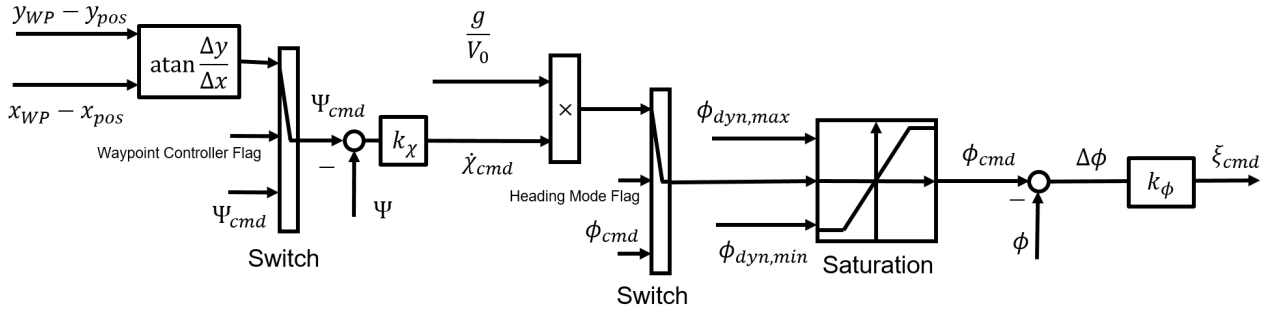
Considering Fig. 2.27(b) as an illustration, the required bank angle can be calculated by:

$$\tan \mu = \frac{n_{des,L}}{n_{des,V}}. \tag{2.30}$$

That means, the desired accelerations considered in the lateral and the pull-up maneuver can be still reached if a combined maneuver is conducted by reducing the absolute maximum bank angle command. As a result, in order to achieve the desired accelerations within the combined maneuver, a decision logic is implemented, which sets the appropriate bank angle and pitch angle limitations. If the lateral or vertical maneuver are performed separately, the default limitations are applied. If both maneuvers are performed simultaneously, the bank angle is limited according to Eq. (2.30). If the bank angle is already above a certain threshold ϕ_T, the pull-up maneuver is not initiated as long as the bank angle is above the threshold ϕ_T. After the aerial vehicle has reduced the bank angle below ϕ_T, the vertical maneuver is initiated. For example, if the aerial vehicle has already established the commanded bank angle to achieve the desired lateral acceleration, and successively the vertical maneuver is triggered, the bank is reduced first. Finally, the vertical maneuver is triggered. The delayed trigger of the vertical maneuver might be considered in the vertical safety margin calculation. However, this is beyond the scope of this thesis.
The next section applies the approach in model-in-the-loop simulations and demonstrates the different evasive maneuvers.



(a) Autopilot structure for longitudinal motion



(b) Autopilot structure for lateral motion

Figure 2.28: Autopilot structure within model-in-the-loop environment: x_{WP}, y_{WP} represent waypoint coordinates in local cartesian frame

2.5 Testing and Verification[7]

2.5.1 Model-in-the Loop Simulation

For the verification of the proposed algorithm, the presented geofencing approach is integrated in a model-in-the-loop environment. The high-fidelity simulation model of the *Diamond DA42* general aviation aircraft is used, which was already utilized for the model-based development of a fly-by-wire flight control system (see e.g. Ref. [77]). A standard and exemplary autopilot is implemented to demonstrate the presented geofencing concept and the execution of the evasive maneuvers. The integration of the geofencing algorithm into a flight control system designed for flight is not within the scope of this thesis.
The autopilot structure is shown in Fig. 2.28. For the longitudinal motion, a cascaded controller is integrated with a pitch controller as inner loop and altitude controller as outer loop. The illustrated rate limiter ensures, that the desired climb angle rate and thus the desired vertical acceleration is reached. The values $\dot{\gamma}_{des,min/max}$ are calculated using Eq. (2.24) with the desired vertical load factor and the current speed V_K. The autopilot of the lateral motion consists of a bank controller, a heading controller and a waypoint controller. An usual yaw

[7]This section represents a revised part of the work in Ref. [75], which has been published by the author.

damper and autothrottle is implemented to enhance the dutch roll stability and to control the speed. The simulation model has been linearized in one trim point in a steady straight-level flight. The control gains are determined using root locus methods. Switches between the different loops enable the activation of the different control laws. The saturation blocks allow a dynamic limitation adaption of the pitch and bank angle command during the evasive maneuvers.

2.5.2 Simulation Results

2.5.2.1 Lateral Maneuver

The desired lateral load factor for the present simulation model is set to $n_{des,L} = \sqrt{3}$, which is equal to a level turn with kinematic bank angle μ =60 deg. The safety margin is calculated according to Eq. (2.20) with the worst-case speed V_{max} =85 m/s. Fig. 2.24 illustrates the simulated lateral maneuver at the kinematic speed V_K =65 m/s. The aerial vehicle approaches the xy-boundary perpendicular which is the worst-case approach. After $t = 7$ s the safety area of the z-plane is violated, the lateral evasive maneuver is triggered. The waypoint, as visualized in Fig. 2.29, is generated. The waypoint controller commands a new heading command to reach the generated waypoint, which results in a bank angle command of ϕ_{cmd} =60 deg. The load factor n_z plot in Fig. 2.29 indicates, that a load factor $n_z = 2$ is reached during the turn, which is equivalent to the desired lateral load factor $n_{des,L} = \sqrt{3}$. After the heading error $\Delta\Psi$ is reduced, the heading controller decreases the bank angle command. The subplot x_{pos} visualizes the covered track of the aircraft relative to the safety margin and displays, how far the aerial vehicle entered the safety margin. The last subplot shows the mentioned lateral geofencing modes according to Sec. 2.4.3.

2.5.2.2 Vertical Maneuver

The vertical maneuver can be a pull-up or push-down maneuver depending on how the aerial vehicle hits the vertical safety margin. Within the presented model-in-the-loop simulation, the pull-up/ push-down maneuver is realized by activating the altitude channel of the illustrated autopilot and commanding a higher/lower altitude, when the vertical maneuver is triggered and a certain bank angle limit is not exceeded. The pitch angle command channel is adjusted accordingly by the saturation function of the pitch angle command signal. The pitch angle command is limited to θ_{max} =25 deg to avoid the exceedance of the maximum angle of attack and to prevent the aerial vehicle from low speeds. In this section, a pull-up maneuver is presented with worst-case speed and worst-case climb angle. The speed V_{cmd} =65 m/s is commanded to the autothrottle. During descend the autothrottle is not able to maintain the commanded speed and the aircraft accelerates towards the worst-case speed. The desired

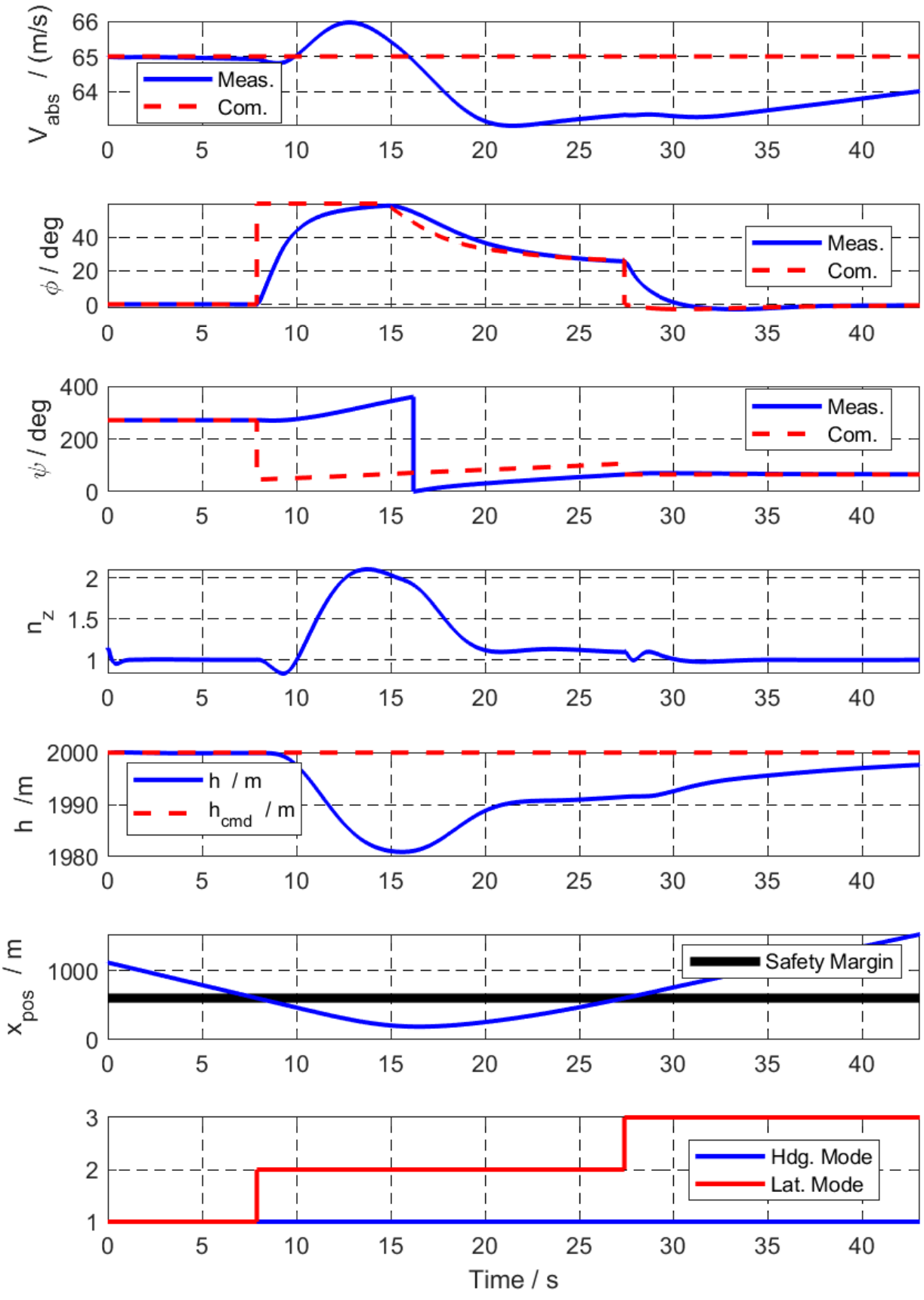


Figure 2.29: Conducted lateral maneuver in model-in-the-loop simulations

vertical load factor is set to $n_{des,V}$ =2, which is equivalent to a pull-up maneuver with 1g vertical acceleration assuming γ=0. Fig. 2.30 demonstrates the simulation results. After t =20 s, the aircraft starts to descend with a worst-case climb angle $\gamma_{WC} = -10$ deg. During the descent the aircraft accelerates up to its maximum speed. The vertical safety margin is hit at t =51 s. Then, the vertical evasive maneuver is triggered, which causes a higher altitude command. The resulting pitch angle command is restricted by the climb angle command rate limiter to achieve the desired vertical acceleration. After the aerial vehicle has reentered the vertical safety area, the pitch angle saturation function reduces the commands to force the aerial vehicle to shallow climb. As mentioned above, depending on the current geofencing mode, the pitch angle limiter changes the maximum and minimum command values. The load factor plot shows the desired vertical acceleration during the pull-up maneuver.

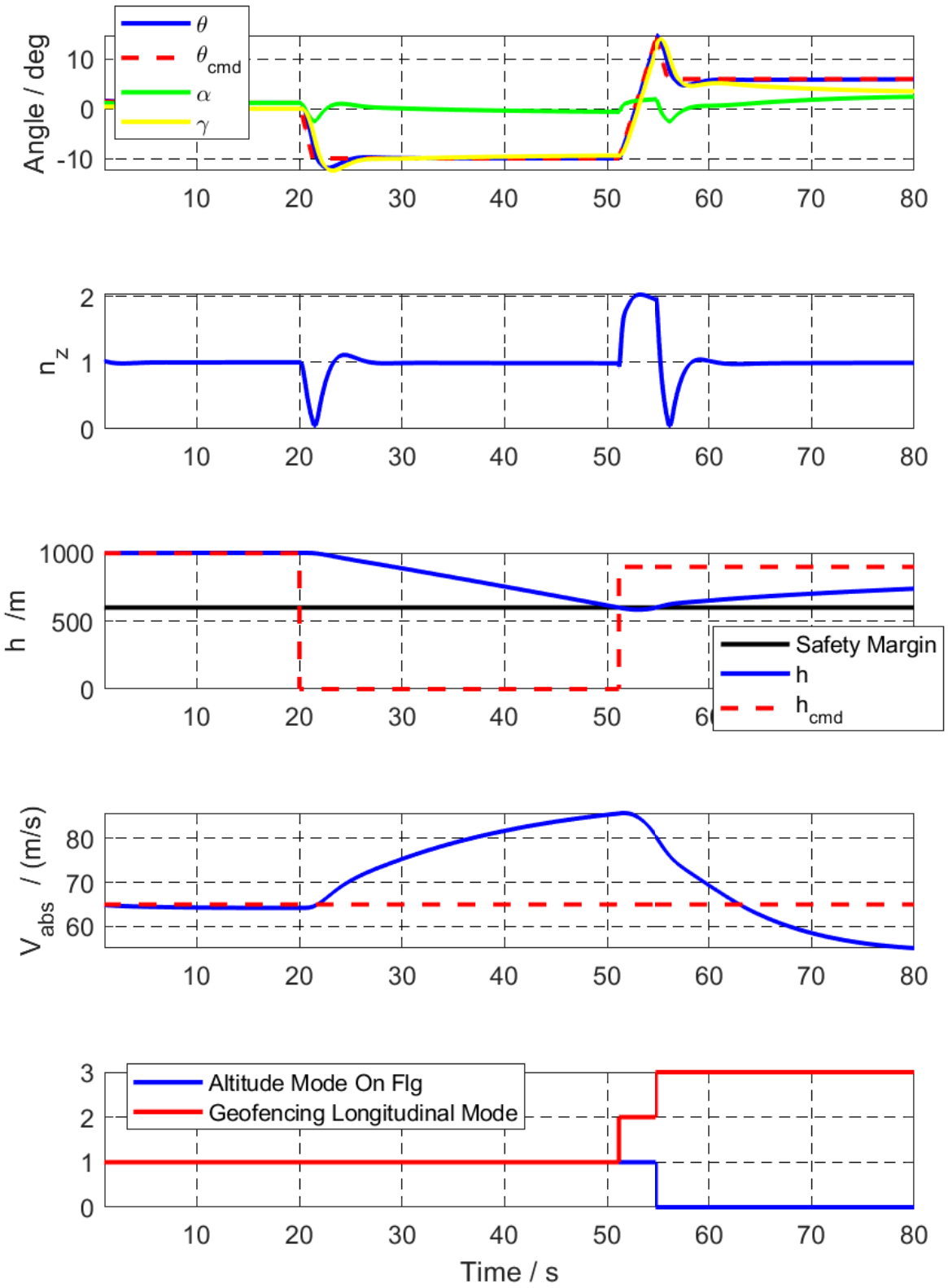


Figure 2.30: Conducted vertical maneuver in model-in-the-loop simulations

2.5.2.3 Combined Maneuver

The discussed combined maneuver consists of the above mentioned pull-up/ push-down maneuver and the lateral maneuver. The maneuvers can be triggered simultaneously or successively in different order. In this section, a combined maneuver is shown, in which a pull-up maneuver is triggered, after the commanded bank angle, due to the execution of a lateral maneuver, is established. As in Sec. 2.4.6 discussed, the combined maneuver is designed such that the desired lateral and vertical load factor are still achieved. According to Fig. 2.27(b) and the above sections, the overall load factor of the described maneuver is calculated by:

$$n_z = \sqrt{n_{des,V}^2 + n_{des,L}^2} = \sqrt{2^2 + \sqrt{3}^2} = \sqrt{7} \tag{2.31}$$

The load factor is verified in the following by the discussed model-in-the-loop simulation. The flight path of the maneuver is illustrated in Fig. 2.31. The respective important parameters are shown in Fig. 2.32. In Fig. 2.31, the three-dimensional flight area is shown in blue color. The aerial vehicle descends towards an inclined plane (i.e. slope) boundary. The outer color of the flight path marks the lateral geofencing mode, the inner color visualizes the vertical geofencing mode. By approaching the inclined plane, the lateral safety margin is violated first, equivalently to Fig. 2.27(a). This is visible by the outer line, which changes its color from magenta towards black color. In Fig. 2.32, it can be seen in the last subplot, where the lateral and vertical geofencing modes are shown. When the lateral geofencing mode changes its mode to *evasive*, the bank angle command changes steplike to ϕ_{cmd} =60 deg. Two seconds later at about t =41 s, the vertical safety margin is also violated. This is visible in the last subplot, when the vertical geofencing mode changes. Because a high bank angle is already established, the bank angle command first is reduced to ϕ_{cmd} =40 deg (which is according to Eq. (2.30) and the desired load factor). When the bank angle is reduced, the pull-up maneuver is initiated. This can be seen in the pitch angle subplot in Fig. 2.32 by the increasing pitch angle command, which again is slowed down by the rate limiter to achieve the desired climb angle rate. At about t =47 s the maximum pitch angle θ_{max} is reached. Fig. 2.31 indicates the switching by changing the inner line color from blue to red color. The load factor subplot in Fig. 2.32 underlines, that the desired acceleration is reached, as calculated in Eq. (2.31), since the load factor is approximately $n_z = \sqrt{7}$ when both maneuvers are conducted. After t =50 s, the aerial vehicle exits the vertical safety margin, and a smaller pitch angle is command for shallow climb. In Fig 2.31 the inner line changes to grey color. Since the heading error $\Delta\Psi$ is decreased, the bank angle command reduces too. Finally, the lateral maneuver is also completed, indicated by the outer line color change from black to cyan color.

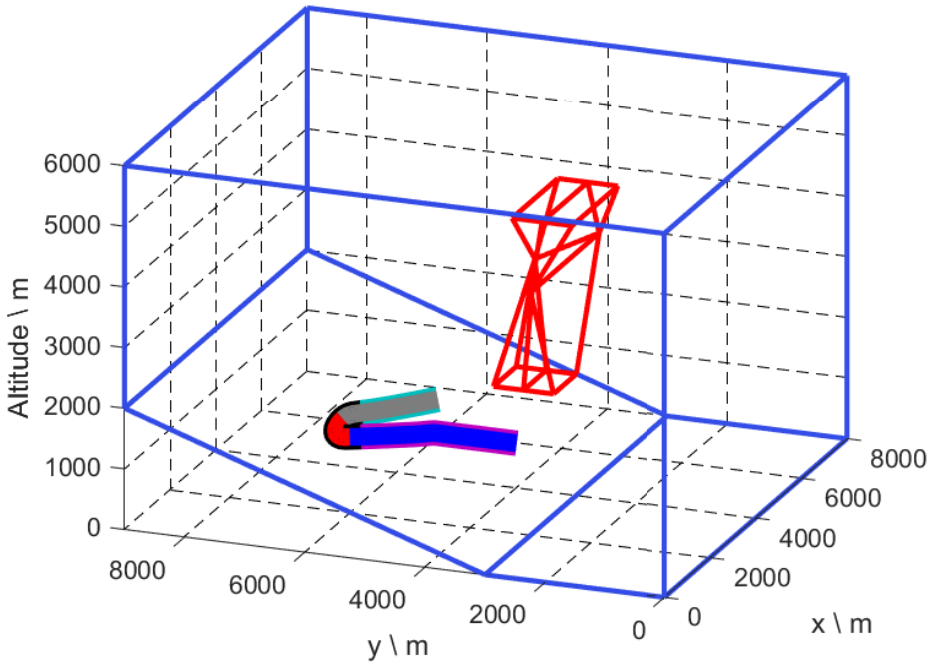


Figure 2.31: Flight path of simulated combined vertical and lateral maneuver

The simulation of the combined maneuver underlines, that the desired lateral and vertical load factor can be still fulfilled by dedicated reduction of the bank angle command. As result, the calculated safety margins, calculated separately for the lateral and the vertical maneuver, are also sufficient for the combined maneuver.

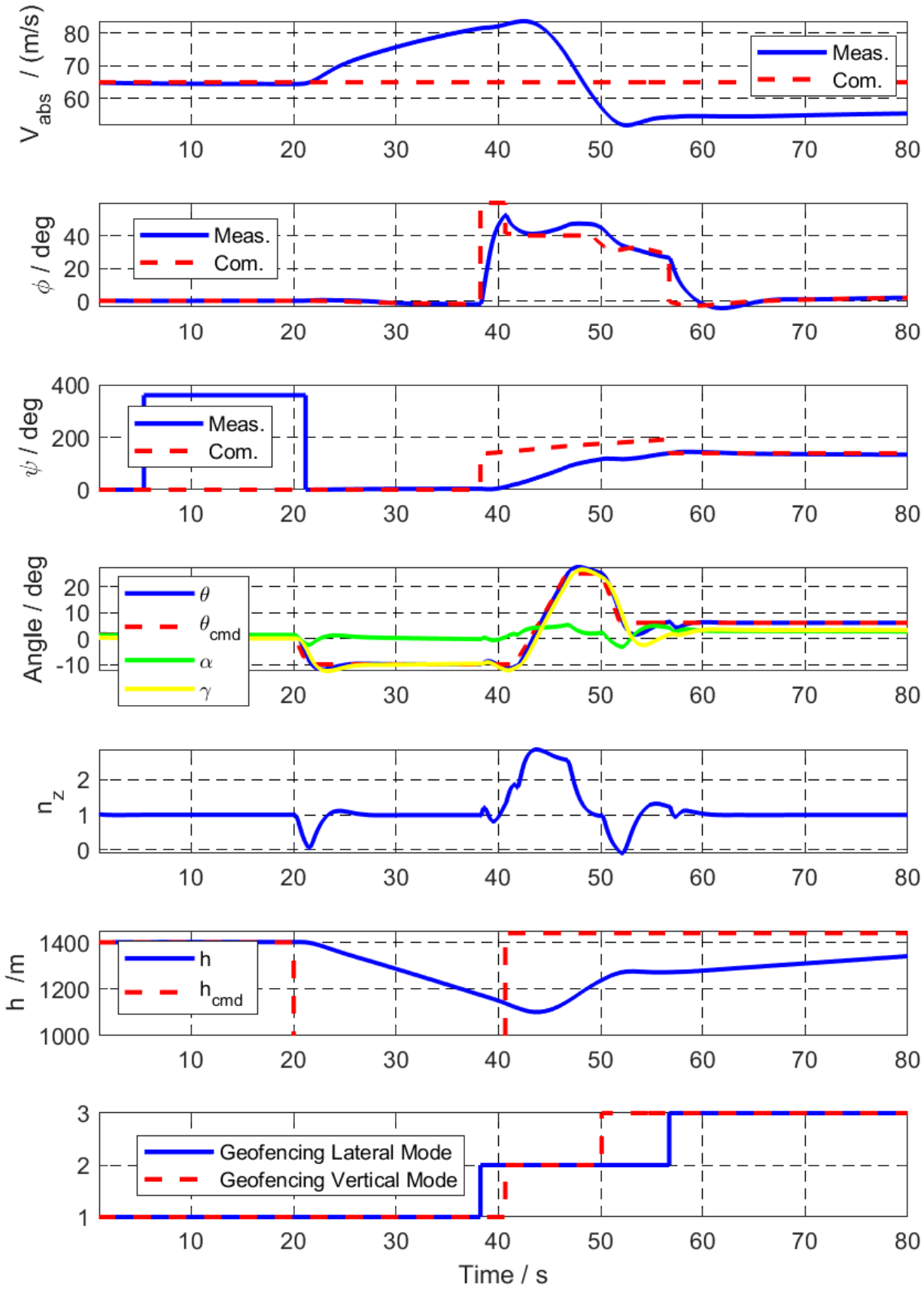


Figure 2.32: Simulation results of a combined maneuver

2.6 Summary and Outlook

Within this chapter, the first complete concept of three-dimensional geofencing for wing-borne flight has been proposed. The introduced algorithm encompasses the fundamental functions, which are necessary for a fully-automatic geofencing algorithm. The concept focuses on the functions with respect to flight control systems. Thereby, the algorithm has been developed to handle a high-variety of flight areas. The geofencing functions, which are executed during flight, are designed such that only a limited number of equations are necessary to calculate. The geofencing algorithm has been demonstrated in model-in-the-loop simulations under the consideration of a conventional flight control system.

Two important points are left open for future work. Firstly, the investigation of a combined push-down and turn maneuver is still open. Secondly, the consideration of disturbances including wind in the calculation of the safety margins needs to be taken into account in more detail.

Beside the mentioned remaining points in the design of the algorithm, the presented geofencing algorithm has reached a maturity, in which future work can focus on the testing and verification of the presented functions, the embedding of these functions into flight control systems designed for the respective real aircraft and finally the validation in flight tests. However, the presented work establishes the basis for using the geofencing algorithm as risk mitigation strategy.

It is of particular note that the presented geofencing functions are not limited to the application of UAVs. The presented geofencing algorithm can also be used as an additional safety enhancement function for manned aircraft. Moreover, the proposed algorithm can even be utilized for the development of *detect and avoid* algorithms.

Chapter 3

Stability and Robustness Analysis of Control Laws for VTOL aircraft

In the last chapter a risk mitigation strategy is introduced to enhance the overall safety of the aerial vehicles by the integration of an explicit and additional functionality to the flight control system.
The present chapter elaborates various stability and robustness metrics for novel aerial vehicles. The application of the presented methodologies increases the level of safety for flight control systems and ensures correct flight controller design with respect to uncertainties and disturbances.
The newly emerging and over-actuated configurations of optionally-piloted or unmanned aerial vehicles demand stability and robustness analysis metrics, which take multiple control effector channels simultaneously into account. In case of a highly actuated system, the known single-input-single-output (SISO) robustness metrics might be misleading, since they consider uncertainties only in one channel at the analysis point. Thus, in addition to the classical SISO robustness metrics, the present chapter elaborates methodologies to analyze the stability and robustness sufficiently for multi-control effector aerial vehicles, which are multiple-input-and-multiple-output (MIMO) systems. The presented metrics allow the quantification of robustness and stability of closed-loop systems by respecting simultaneous uncertainties in multiple channels. Therefore, different MIMO margins are explained and evaluated with regard to their conservatism. Furthermore, the μ-analysis method is explained, which calculates the robustness of the system under consideration of a detailed uncertainty modeling. On top of the elaborated margins and the μ-analysis, a novel probabilistic-μ approach is proposed, which considers unbounded stochastic uncertainties. The approach calculates a lower bound of the probability for achieving stability or a specific handling quality borderline in the frequency domain.
By fulfilling specified stability and robustness requirements of the presented metrics it can be guaranteed that the flight controller functions in off-nominal plant conditions in reality. The

adherence of the presented robustness metrics enables the necessary level of safety for the flight controller. This is especially beneficial for first flights of aerial vehicles, which have high uncertainties in their plant behavior and a large variety of control input channels.
The robustness metrics are applied to the emergency flight controller of a novel eVTOL demonstrator for its first flights.
This chapter encompasses the *Contributions II*, introduced in Sec. 1.2.2, and provides the following main content:

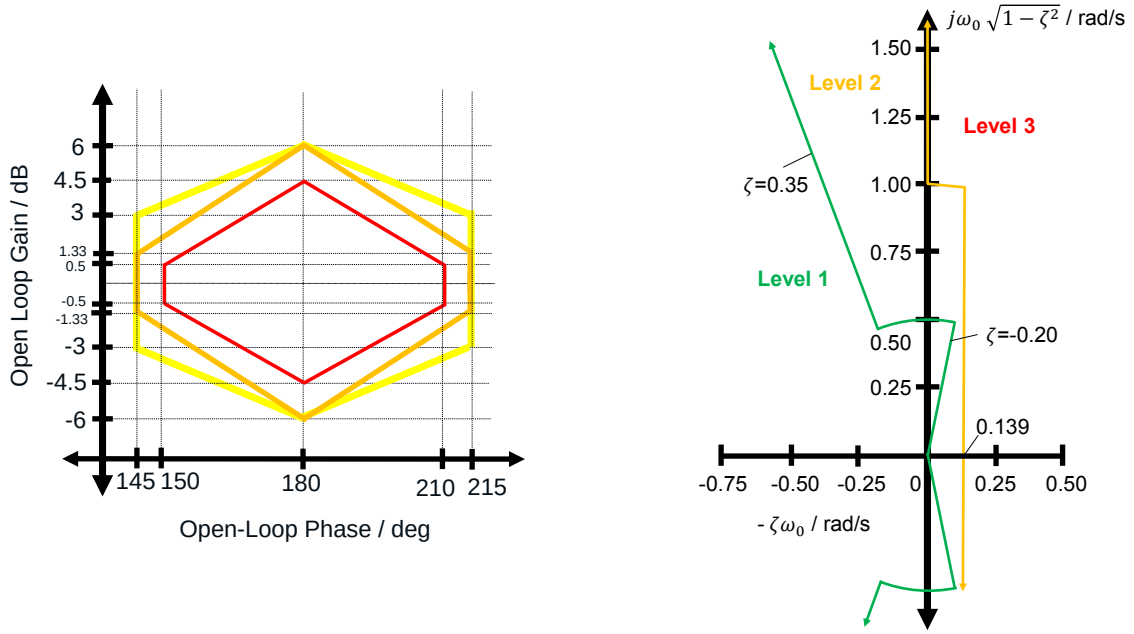
- Detailed elaboration of different SISO and MIMO gain and phase margins and their relation with respect to conservatism
- Calculation of real and complex worst-case uncertainties and their respective transfer function for guaranteed stability and guaranteed handling qualities by utilization of μ-analysis methods
- Open-loop analysis of an eVTOL demonstrator by the utilization of nonlinear, complex and uncertain plant dynamics
- Application of the presented stability and robustness metrics to the closed-loop system of an eVTOL demonstrator consisting of the nonlinear plant dynamics and an incremental nonlinear dynamic inversion flight controller
- Introduction of the novel probabilistic-μ analysis under consideration of probabilistic uncertainties instead of uncertainties with limited intervals as in the classical μ-analysis
- Application of the presented probabilistic-μ analysis to eVTOL demonstrator

This chapter is split up into two main parts. The first part covers the theoretical background for stability and robustness assessment of novel aerial vehicles.
At first, current available stability and robustness requirements for aerial vehicle certification are reviewed. Subsequently, Sec. 3.2 revisits relevant aspects of linear feedback control. On this basis, Sec. 3.3 introduces the background of uncertainties and their respective modeling in MIMO systems. Sec. 3.5 elaborates approaches for stability and robustness metrics including the MIMO metrics. Thereon, Sec. 3.6 assesses the conservatism of the different MIMO margins under the consideration of Fan's Theorem. Sec. 3.7 and Sec. 3.8 present the conventional and advanced μ-analysis and proposes an approach to determine worst-case transfer functions for complex uncertainties with regard to handling quality borderlines. Furthermore, Sec. 3.9 suggests a novel probabilistic-μ approach in order to consider unbounded stochastic uncertainties. This is in contrast to the classical μ-analysis. The uncertainties considered in the classical μ-analysis are only defined by limited intervals and without the specification of probability density functions.

The second part, which encompasses Sec. 3.10 and Sec. 3.12, apply the discussed robustness metrics including the new probabilistic-μ approach to the closed-loop system of an eVTOL demonstrator.

3.1 Published Stability, Robustness and Performance Requirements within Certification Documents

The classical robustness requirements for aircraft with conventional fixed-wing configuration are the SISO gain and phase margins, which can be found in the *Military Specification: Flight Control Systems* from the U.S. Air Force (MIL-F-9490) (cf. Ref. [78]) or the Aerospace Standard AS94900 from the Society of Automotive Engineers (cf. Ref. [79]). The SISO gain margin (GM) and phase margin (PM) shall thereby be always |GM|$\geq$6 dB and |PM|$\geq$45 deg, which are required between the minimum and maximum operational airspeed.
Equivalent margin requirements can be found also in the Aeronautical Design Standard for Military Rotorcraft (ADS-33, Ref. [80]) for Attitude Command / Attitude Hold Response Types (cf. Ref. [80, p.76]). The NATO Standardization Agreement for Unmanned Aerial Vehicles Systems Airworthiness Requirements (STANAG 4671) states that *"the UAV, augmented by the FCS ... must be longitudinally, directionally and laterally stable in any condition"* (Ref. [81, p.33]). The STANAG 4671 is intended for military UAVs with maximum take-off weight between 150 kg and 20.000 kg.
Since a combination of gain and phase can also lead to instability, the Nichols diamond was established. The Nichols diamond is a disk margin, which makes sure that an overall sufficient distance to the critical point in the Nyquist plane is achieved. The derivation of the Nichols diamond is explained in Sec. 3.5.2. The Nichols diamond, which is based on the mentioned |GM|$\geq$6 dB and |PM|$\geq$45 deg, is shown in yellow color in Fig. 3.1(a). For the SAGITTA demonstrator, reduced Nichols diamonds are published in Ref. [39] for nominal and degraded conditions. The respective diamonds are drawn in orange and red color in Fig. 3.1(a).
Specific MIMO margin requirements are limited in public documents. Within certification documents no specifications are made to MIMO margin requirements. In Ref. [38, p.121] a first guideline is given for the MIMO phase margin, which shall be between 20 deg and 30 deg. Refs. [82, 34] apply MIMO margins to evaluate the robustness of flight control systems for different aircraft. The closed-systems achieve thereby MIMO |GM|>3 dB and MIMO |PM|>30 deg. These MIMO margin values are used within the subsequent studies as reference MIMO margin requirements.
Performance requirements and guidance material (including pole locations of eigenmotions) for conventional aircraft are published in the *Military Specification: Flying Qualities of Piloted*



(a) Nichols diamonds

(b) ADS-33 handling quality requirements for pitch and roll oscillations (reworked from Ref. [80, p.77])

Figure 3.1: Robustness and handling quality requirements

Airplanes (MIL-F-8785C, Ref. [83]) and the *Military Standard: Flying Qualities of Piloted Aircraft* (MIL-STD-1797A, Ref. [84]). Performance requirements for rotorcraft can be found in the ADS-33. Fig. 3.1(b) shows required pole locations in the complex plane for the roll and pitch eigenmotion for different handling quality levels. The shown handling quality borderlines of the ADS-33 are later used in Sec. 3.10 as handling quality requirement within the robustness assessment of an eVTOL demonstrator.

The *Proposed Means of Compliance with the SC-VTOL* (MOC-SC-VTOL), Ref. [85]) published by the EASA gives a comprehensive set of technical specifications for new VTOL aircraft. The presented approach is in contrast to existing requirement methodologies, since no specific method is required to assess stability and handling qualities (HQ). *"HQ are evaluated without any specific generally recognised method, and are mainly evaluated to measure the workload to determine the minimum crew in respect to the kind of operations."* (Ref. [85, p.10]). As a result, the approach for certification is left open to the applicant, as long as the applicant can demonstrate sufficient stability and handling qualities by the chosen method. Therefore, the subsequent chapter elaborates an appropriate approach to assess the stability and robustness of a new VTOL demonstrator. The proposed approach can be generalized and applied to any kind of such vehicles.

3.2 Feedback Control Revisited

For the sake of clarity, important aspects of feedback control for linear-time-invariant systems are recapitulated, in order to introduce the notation for the following chapter and to have a common view on the elaboration of the stability and robustness metrics. The focus of the next section is set on the calculation of different sensitivity functions for multiple input and multiple output linear-time invariant systems, which are essential for the calculation of the robustness metrics. Furthermore, Sec. 3.2.3 clarifies how loop breakings of MIMO closed-loop systems are executed to apply SISO robustness criteria for MIMO systems. Sec. 3.2.4 explains the differences between stability, robustness and performance and links the respective sections within this chapter to these terms.

3.2.1 Nomenclature

First, important nomenclature utilized within this chapter is highlighted to have a common understanding of the terminology. It is distinguished between the notations x, $\boldsymbol{x}$, X and $\underline{\boldsymbol{X}}$. Variables which are written such as $x \in \mathbb{R}$ and $y \in \mathbb{C}$ are scalars and are thus one-dimensional variables. Variable notations written in bold script such as $\boldsymbol{x} \in \mathbb{R}^{n\times 1}$ and $\boldsymbol{y} \in \mathbb{C}^{n\times 1}$ are used for vectors with $n \in \mathbb{N}^{+}\backslash\{0\}$ dimensions. If not otherwise stated, variable notations with capital letter such as X and Y are utilized for one-dimensional transfer functions. In contrast, variables such as $\underline{\boldsymbol{X}} \in \mathbb{R}^{n\times m}$ and $\underline{\boldsymbol{Y}} \in \mathbb{C}^{n\times m}$ are written in bold script with underline and represent MIMO transfer functions or real/ complex matrices with $n \times m$ dimensions and $n,m \in \mathbb{N}^{+}\backslash\{0\}$. Higher-dimensional arrays such as $\underline{\underline{\boldsymbol{\Pi}}}^{n_1\times\ldots\times n_i}$ with $i,n_1,\ldots,n_i \in \mathbb{N}^{+}\backslash\{0\}$ are written in bold script and two underlines.
The presented notation has been chosen to highlight the difference between the SISO transfer function $X(s)$ and the MIMO transfer function $\underline{\boldsymbol{X}}(s)$.

3.2.2 Closed-Loop Transfer Function, Sensitivity Transfer Function and Complementary Sensitivity Transfer Function

The linear time-invariant (LTI) and multiple-input-multiple-output (MIMO) closed-loop system with the plant transfer function $\underline{\boldsymbol{G}}(s)$ and controller transfer function $\underline{\boldsymbol{K}}(s)$ in Fig. 3.2 is considered for the derivation of the respective closed-loop transfer function, the sensitivity transfer functions and complementary sensitivity transfer functions. The plant $\underline{\boldsymbol{G}}(s)$ is a $n \times m$ dimensional transfer function with $n,m \in \mathbb{N}^{+}\backslash\{0\}$. The controller $\underline{\boldsymbol{K}}(s)$ is a $m \times n$ dimensional transfer function. The output signal vector $\boldsymbol{y}$ of the plant transfer function is a $n \times 1$ dimensional signal vector and is calculated by:

$$\boldsymbol{y}(s) = \underline{\boldsymbol{G}}(s)\boldsymbol{u}(s) \tag{3.1}$$

with the input signal vector $\boldsymbol{u}(s)$, which is a $m \times 1$ dimensional vector. Using the transfer function of the controller $\underline{\boldsymbol{K}}(s)$ and the input signal $\boldsymbol{r}(s)$, which has the same dimension as $\boldsymbol{y}$, the closed-loop response is computed by:

$$\boldsymbol{y} = \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)(\boldsymbol{r} - \boldsymbol{y}) \tag{3.2}$$

$$(\underline{\boldsymbol{I}} + \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s))\boldsymbol{y} = \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)\boldsymbol{r} \tag{3.3}$$

$$\boldsymbol{y} = \underbrace{(\underline{\boldsymbol{I}} + \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s))^{-1}\underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)}_{\underline{\boldsymbol{T}}_O}\boldsymbol{r} \tag{3.4}$$

$\underline{\boldsymbol{T}}_O$ is called the output complementary sensitivity function. The closed-loop system is often affected by disturbances, which influence the behavior of the closed-loop system. The disturbances can feed in at different locations of the system and in different ways. The so-called sensitivity function is usually defined as the transfer function from the disturbance signal to specific signals within the closed-loop system, while the command signal is set to $\boldsymbol{r} = 0$. The following descriptions are based on the illustrations in Fig. 3.3 and 3.4.
In the literature (e.g. Ref. [86, p.22]), it is usually written about one sensitivity function without further specification, which is actually named the output sensitivity function $\underline{\boldsymbol{S}}_O$:

$$\boldsymbol{y} = \boldsymbol{dy}_S - \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)\boldsymbol{y} \tag{3.5}$$

$$\boldsymbol{y} = \underbrace{(\underline{\boldsymbol{I}} + \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s))^{-1}}_{\underline{\boldsymbol{S}}_O}\boldsymbol{dy}_S \tag{3.6}$$

The output sensitivity function $\underline{\boldsymbol{S}}_O$ is the transfer function from the disturbance $\boldsymbol{dy}$ to the output signal $\boldsymbol{y}$. Thereby, the signal $\boldsymbol{dy}_S$ is injected after the plant. Considering an aerial vehicle as plant, in reality, the disturbance $\boldsymbol{dy}_S$ can occur, for example, due to wind gusts. Equivalently, the input sensitivity function $\underline{\boldsymbol{S}}_I$ is the transfer function from the input $\boldsymbol{du}_S$ to the input signal $\boldsymbol{u}$ of the plant. This disturbance can occur for example due to actuator malfunction:

$$\boldsymbol{u} = \boldsymbol{du}_S - \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)\boldsymbol{u} \tag{3.7}$$

$$\boldsymbol{u} = \underbrace{(\underline{\boldsymbol{I}} + \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s))^{-1}}_{\underline{\boldsymbol{S}}_I}\boldsymbol{du}_S \tag{3.8}$$

The respective open-loop transfer function of $\underline{\boldsymbol{S}}_I$ can be written as $\underline{\boldsymbol{L}}_I(s) = \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)$, which is in contrast to the open-loop transfer function $\underline{\boldsymbol{L}}_O(s) = \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)$ of the output sensitivity transfer function $\underline{\boldsymbol{S}}_O$. Besides the described sensitivity functions, so-called complementary sensitivity functions exist, as already mentioned above.

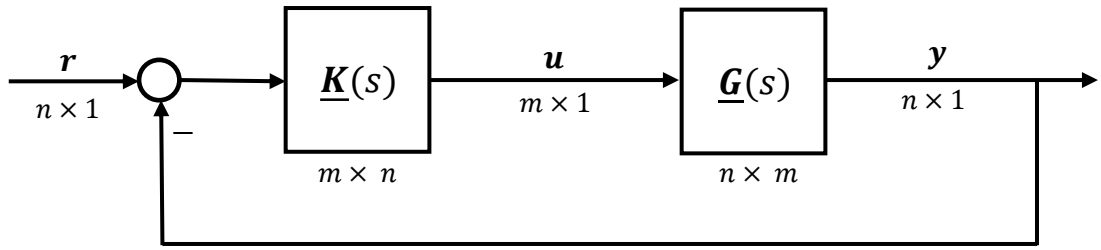


Figure 3.2: Considered MIMO closed-loop system with signal dimensions n and m

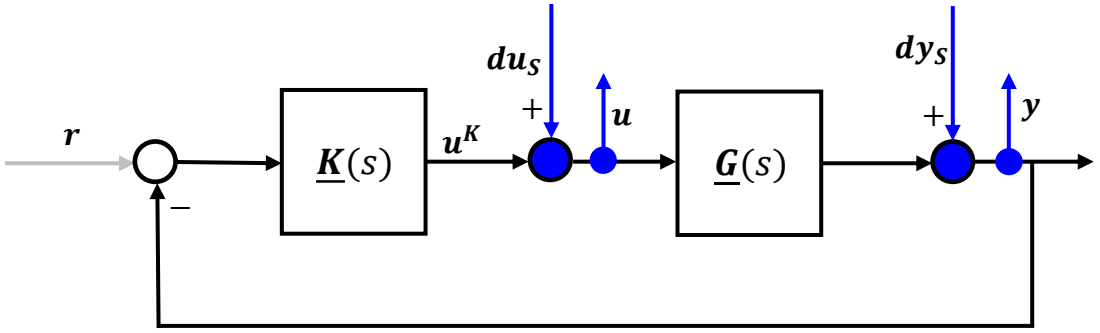


Figure 3.3: Block diagram for the derivation of the input and output sensitivity transfer functions $\underline{S}_I$ and $\underline{S}_O$

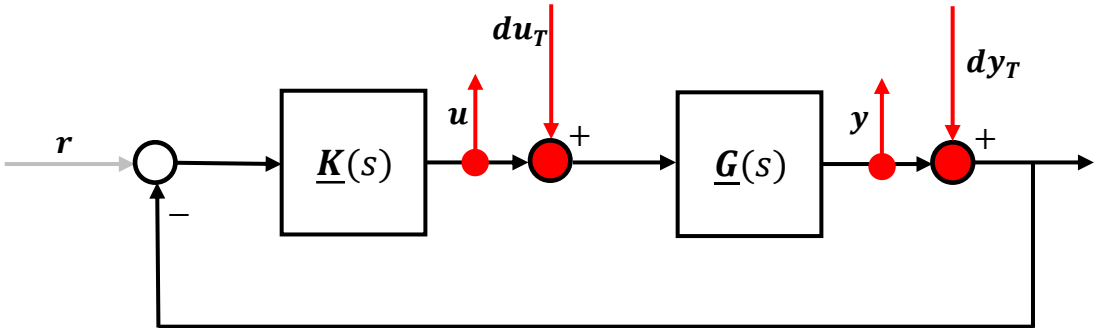


Figure 3.4: Block diagram for the derivation of the input and output complementary sensitivity transfer functions $\underline{T}_I$ and $\underline{T}_O$

The input complementary sensitivity transfer function $\underline{\boldsymbol{T}}_I$ is defined as the transfer function from the signal $\boldsymbol{du}_T$ to the input signal $\boldsymbol{u}$:

$$\boldsymbol{u} = -\underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)(\boldsymbol{u} + \boldsymbol{du}_T) \tag{3.9}$$

$$\boldsymbol{u} = -\underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)\boldsymbol{u} - \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)\boldsymbol{du}_T \tag{3.10}$$

$$\boldsymbol{u} = -\underbrace{(\underline{\boldsymbol{I}} + \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s))^{-1}\underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)}_{\underline{\boldsymbol{T}}_I}\boldsymbol{du}_T \tag{3.11}$$

The output complementary sensitivity function $\underline{\boldsymbol{T}}_O$ is defined as follows:

$$\boldsymbol{y} = -\underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)(\boldsymbol{y} + \boldsymbol{dy}_T) \tag{3.12}$$

$$\boldsymbol{y} = -\underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)\boldsymbol{y} - \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)\boldsymbol{dy}_T \tag{3.13}$$

$$\boldsymbol{y} = -\underbrace{(\underline{\boldsymbol{I}} + \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s))^{-1}\underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)}_{\underline{\boldsymbol{T}}_O}\boldsymbol{dy}_T \tag{3.14}$$

By comparing the complementary sensitivity function $\underline{\boldsymbol{T}}_O$ to the closed-loop transfer function in Eq. (3.2), it can be seen that they are equivalent. The name results, due to the following relationship:

$$\underline{\boldsymbol{S}} + \underline{\boldsymbol{T}} = \underline{\boldsymbol{I}} \tag{3.15}$$

with the identity matrix $\underline{\boldsymbol{I}}$. The equation is valid for either the sensitivity functions $\underline{\boldsymbol{S}}_I$ and $\underline{\boldsymbol{T}}_I$ at the input or $\underline{\boldsymbol{S}}_O$ and $\underline{\boldsymbol{T}}_O$ at the output. As seen by the derivations above, the distinction between the open-loop transfer functions $\underline{\boldsymbol{L}}_I(s) = \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)$ and $\underline{\boldsymbol{L}}_O = \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{K}}(s)$, and thus also for the sensitivity and the complementary sensitivity functions is necessary for MIMO systems, since they are not equivalent in general.
In case the closed-loop system is a SISO system, it becomes apparent, that the following applies:

$$L_O = L_I = L = K(s)G(s) = G(s)K(s) \tag{3.16}$$

due to the commutative nature of SISO transfer functions. As a result, the input and output (complementary) sensitivity functions become equivalent for SISO systems. That means, the following holds:

$$T_O = T_I = T \quad \text{and} \quad S_O = S_I = S \tag{3.17}$$

It is important to note, that Eq. (3.17) holds in the general case only for SISO systems. In summary, it can be said, that sensitivity functions of MIMO systems are dissimilar at different analysis points. As a result, several analysis points are chosen for the robustness assessment of MIMO systems. This becomes visible in Sec. 3.10, which shows the application of robustness metrics to an eVTOL demonstrator.

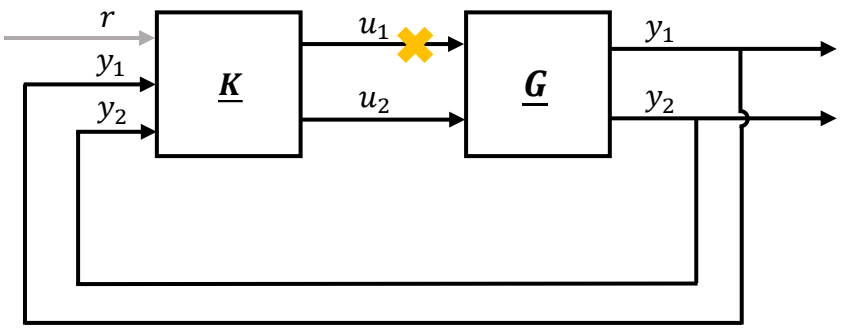


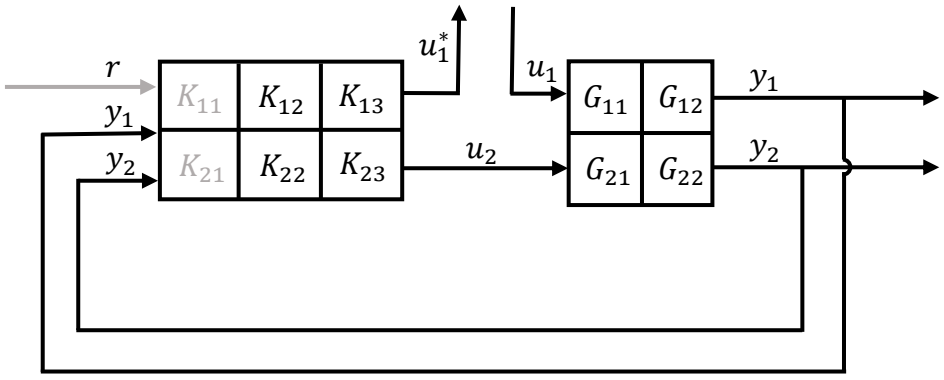
Figure 3.5: Illustrative MIMO Closed-loop system

3.2.3 Loop Breaking in MIMO Systems

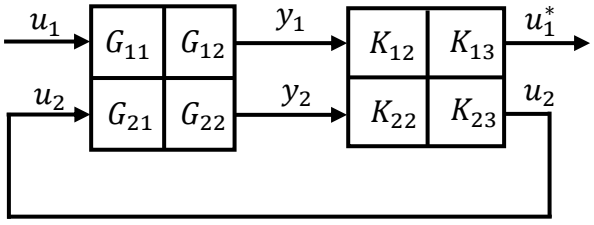
Individual loop channels of MIMO closed-loop systems are broken, in order to analyze single channels with respect to perturbations and to apply SISO stability and robustness criteria. On the one hand, this approach enables the calculation of channel-specific SISO gain and phase margins, which are documented in certification processes and can be used for conformance to robustness requirements. On the other hand, it can identify lacks in stability and robustness margins for specific channels or degrees of freedom.
One drawback is that only one channel is analyzed, whereas the other channels in parallel are kept closed. This may lead to misinterpretations, if many channels are in parallel and only one channel is broken.
For example, the calculated phase margin can be interpreted as the maximum delay, which can be inserted into the respective channel, until the system becomes unstable. Since the other channels are kept closed, the calculated delay represents only the maximum delay for the respective channel, whereas for the other channels in parallel no delay is considered. Thus, it is important to keep the number of necessary cuts low. In aerospace, the MIMO closed-loop breaking is also known as the *Bottleneck-Cut* (cf. Ref. [87]).
Subsequently, one possible way is explained, how the respective SISO open-loop transfer function $L(s)$ for an exemplary MIMO closed-loop system is calculated to lay the foundation for loop breakings in MIMO systems.
Fig. 3.5 illustrates the exemplary MIMO system with the MIMO controller transfer function $\underline{\boldsymbol{K}}(s)$ and the plant transfer function $\underline{\boldsymbol{G}}(s)$:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \underbrace{\begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix}}_{\underline{\boldsymbol{G}}(s)} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} u_1^* \\ u_2 \end{bmatrix} = \underbrace{\begin{bmatrix} K_{12} & K_{13} \\ K_{22} & K_{23} \end{bmatrix}}_{\underline{\boldsymbol{K}}(s)} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \tag{3.18}$$

In order to analyze the first input channel by keeping the remaining closed, the loop is broken at the first input u_1. To apply SISO stability criteria, it is necessary to obtain the transfer



(a) *Bottleneck-Cut* at first input channel



(b) Structural transformation

Figure 3.6: Exemplary transformation procedure for the calculation of a SISO open-loop transfer function $L(s)$ after a *Bottleneck-Cut* at the first plant input channel

function from the input u_1 to the signal u_1^* (see Fig. 3.6(a)). For the calculation of $L(s)$, the command channel is set to $r(s) = 0$ and the system is restructured, such that Fig. 3.6(b) results. By multiplying the plant transfer function by the controller transfer function the following equation can be derived:

$$\begin{bmatrix} u_1^* \\ u_2 \end{bmatrix} = \begin{bmatrix} K_{12} & K_{13} \\ K_{22} & K_{23} \end{bmatrix} \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} = \begin{bmatrix} K_{12}G_{11} + K_{13}G_{21} & K_{12}G_{12} + K_{13}G_{22} \\ K_{22}G_{11} + K_{23}G_{21} & K_{22}G_{12} + K_{23}G_{22} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \tag{3.19}$$

By writing the matrix multiplication by single equations, one yields:

$$u_1^* = \big(K_{12}G_{11} + K_{13}G_{21}\big)u_1 + \big(K_{12}G_{12} + K_{13}G_{22}\big)u_2 \tag{3.20}$$

$$u_2 = \big(K_{22}G_{11} + K_{23}G_{21}\big)u_1 + \big(K_{22}G_{12} + K_{23}G_{22}\big)u_2 \tag{3.21}$$

To get the transfer function from u_1 to the signal u_1^*, the second channel u_2 needs to be closed. Therefore, Eq. (3.21) is rearranged, such that one gets the equation:

$$u_2 = \frac{\big(K_{22}G_{11} + K_{23}G_{21}\big)}{1 - \big(K_{22}G_{12} + K_{23}G_{22}\big)}u_1 \tag{3.22}$$

Eq. (3.22) is inserted in Eq. (3.20) to get eventually the SISO open-loop transfer function $L(s)$:

$$u_1^* = \Bigg(\underbrace{\big(K_{12}G_{11} + K_{13}G_{21}\big) + \big(K_{12}G_{12} + K_{13}G_{22}\big) \frac{\big(K_{22}G_{11} + K_{23}G_{21}\big)}{1 - \big(K_{22}G_{12} + K_{23}G_{22}\big)}}_{L(s)} \Bigg) u_1 \tag{3.23}$$

By determining the one-dimensional open-loop transfer function $L(s)$, further SISO stability and robustness metrics can be applied, which are elaborated further in the next sections. The loop breaking is embedded in Ref. [88] as a convenient function for a given control structure.

3.2.4 Stability, Robustness and Performance

Based on the work in Ref. [42], there are typically four analysis types for linear feedback control, which are summarized in the following and relevant for the stability and robustness analysis of an eVTOL.

Nominal Stability: Nominal stability tests evaluate if the nominal system without any perturbations is stable. Therefore, stability criteria such as the analysis of the closed-loop poles and the Nyquist theorem are applied. Typical stability criteria within this work are recapitulated in Sec. 3.4.

Robust Stability: The system is checked for stability in off-nominal plant conditions with respect to a set of uncertainties. Possible uncertainties are explained in Sec. 3.3. For the analysis of robust stability, robustness criteria are applied to the system. This includes SISO and MIMO gain and phase margins, but also μ_Δ-analysis. Relevant robustness criteria, which are applied within this work for the eVTOL demonstrator, are introduced in Sec. 3.5.

Nominal Performance: This analysis type focuses on the closed-loop performance of the nominal plant. Typically, this includes the analysis of the system step response in the time-domain, such as rise time, settling time overshoot and steady-offset. Singular value plots of the sensitivity functions are also typical analysis tests. Different bandwidths of the closed-loop system are also parameters for the evaluation of the nominal performance. There is a dependency between the robustness criteria and the performance evaluation (e.g. singular value plots). Performance measurements mainly encompass the time-domain parameters and the bandwidths of the closed-loop system. Within this work, only the bandwidths are discussed as performance parameters. Nominal performance evaluations in the time domain are not within the scope of this thesis.

Robust Performance: The robust performance can be evaluated analogously to the nominal performance, but with off-nominal plant and a set of induced perturbations. Instead of this time-consuming method, Sec. 3.8 introduces the so-called conform μ_Γ-analysis method, which can guarantee certain performance criteria specified by a conformity border, for a perturbed closed-loop system.

In consequence, it is distinguished between analysis of the nominal system and the system perturbed by uncertainties. In general, robustness of a control system is defined as the insensitivity *"to differences between the actual system and the model of the system which was used to design the controller"* (Ref. [86, p.253]). Using the above test metrics for robust stability and performance, the control system can be analyzed with regard to robustness.

3.3 Uncertainties in MIMO Systems

Since the behavior of the actual / real system is unknown, the challenge arises, how to represent the difference between the actual system and the assumed, nominal system. The disparity between the nominal system and the real system is expressed by uncertainties.
This section elaborates the uncertainty origins, the mathematical definition, different integration methods and the physical uncertainty modeling. The explained uncertainty origins clarify the necessity of the robustness criteria, explained in Sec. 3.5. The following subsections about uncertainty modeling are relevant for the μ- and probabilistic-μ analysis, introduced later in this chapter. This section is based on Ref. [86, Ch. 7+8]. However, it is structured differently to emphasize the various uncertainty properties in a concise overview and compares the classical uncertainties defined by limited intervals without specification of a probability density distribution and unbounded stochastic uncertainties.

3.3.1 Uncertainty Origin

Uncertainties within plant models have several reasons. Some uncertainties specifically for novel aerial vehicles are listed below:

- Flight dynamic model errors in aerodynamics and propulsion forces (e.g. propeller-airframe interactions)
- Unconsidered delays in communication buses
- Unknown structural effects (e.g. flapping)
- Unmodeled high-order dynamics
- Sensor and actuator perturbations and delays
- Parameter variation due to nonlinearities and deviations from trim conditions

According to Ref. [86, p.254] model uncertainties can also include actuator and sensor failures, physical limitations and changes in control objectives, which cause deviations from the nominal plant.

3.3.2 Definition of Uncertainties

Uncertainties are conventionally (as in Ref. [86]) defined by limited intervals without specification of probability density functions. Assumed uncertainties in the plant model are represented in the *plant uncertainty set* $\underline{\underline{\mathbf{\Pi}}}$. The set $\underline{\underline{\mathbf{\Pi}}}$ contains all possible perturbed plant models $\underline{\boldsymbol{G}}_p \in \underline{\underline{\mathbf{\Pi}}}$ and thus describes an infinite number of possible models. The perturbed model is within the set $\underline{\boldsymbol{G}}_p \in \underline{\underline{\mathbf{\Pi}}}$. The set $\underline{\underline{\mathbf{\Pi}}}$ only allows H_∞-bounded stable perturbations, which are denoted by $\underline{\mathbf{\Delta}}$ as multidimensional uncertainty or Δ as scalar uncertainty and are normalized, such that $||\underline{\mathbf{\Delta}}||_\infty \leq 1$. This is equivalently specified in Ref. [86, p.254].

The H_∞ norm is defined as follows:

Definition: The H_∞ norm of a MIMO transfer function $\underline{\boldsymbol{f}}(s)$ is the peak value of of $\underline{\boldsymbol{f}}(s)$ along the frequency and is defined as follows:

$$||\underline{\boldsymbol{f}}(s)||_\infty = \max_{\omega} \bar{\sigma}(\underline{\boldsymbol{f}}(s)) \quad \text{with} \quad s = j\omega \tag{3.24}$$

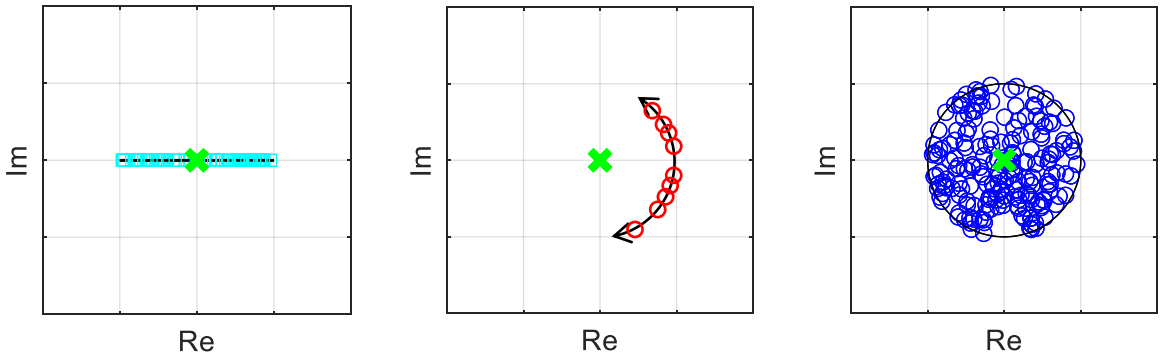
with $\bar{\sigma}(\underline{\boldsymbol{f}}(s))$ indicating the maximum singular value of $\underline{\boldsymbol{f}}(s)$.

Depending on the perturbation assumption, the uncertainty $\underline{\boldsymbol{\Delta}}(j\omega)$ can be a function of the frequency ω. It is distinguished between scalar and multidimensional uncertainties. Scalar perturbations are discussed in the next section. Multidimensional uncertainties $\underline{\boldsymbol{\Delta}} \in \mathbb{C}^{n\times m}$, which are more detailed in Sec. 3.3.5, consist typically of scalar perturbations Δ_{ij} or matrices $\underline{\Delta}_{ij}$.

3.3.3 Real versus complex scalar uncertainties

There are typically two possible ways in order to express a scalar uncertainty mathematically under consideration of a limited uncertainty interval. One mathematical uncertainty expression is called *real uncertainty*, the second is denoted as *complex uncertainty*.
This differentiation is important for the uncertainty modeling/ representation with limited intervals. The uncertainty modeling is explained in Sec. 3.3.6. The classical optimization algorithm of the μ-boundaries and the respective solution depends on the mathematical expressions of the considered uncertainties too. This is explained in Sec. 3.7.

Real uncertainty: A *real uncertainty* $\Delta(j\omega) = \Delta^R \in \mathbb{R}$ is used for a real-valued parameter, whose value is uncertain within the normalized interval $-1 \leq \Delta^R \leq 1$. The uncertainty range can be adapted accordingly by weights using linear mappings, in order to yield the uncertainty in the desired value range. The real perturbation itself is independent from the frequency ω and changes only the gain of the parameter. Thus, the real perturbation does not represent any dynamic behavior.
Fig. 3.7(a) draws the real perturbation as a line in the complex plane. The rectangles represent uniformly distributed samples as examples of possible uncertainty values. The cross represents the nominal parameter. Since the *real uncertainty* is only a variation in gain, it is equivalent to shifting the cross to the left or the right, in parallel to the real axis. In consequence, the real uncertainty changes only the real part of the nominal value.



a) Real perturbation

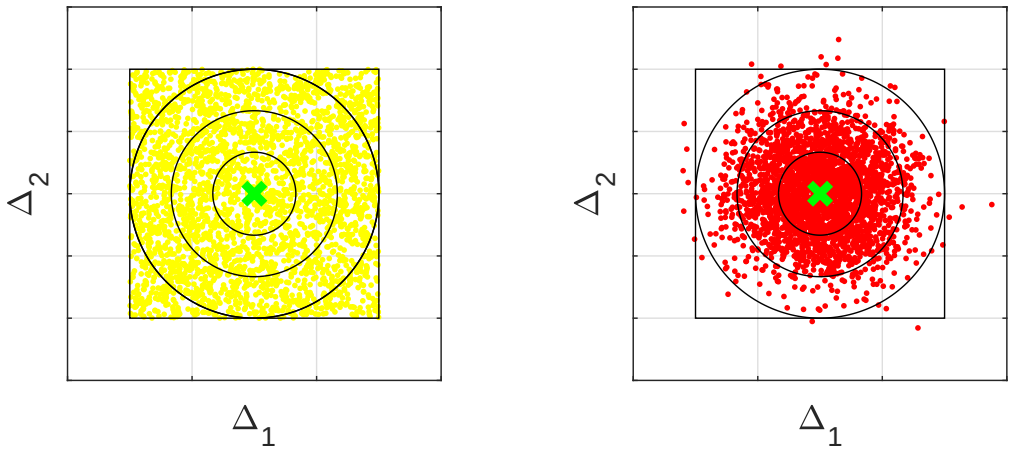
b) Complex perturbation with constant amplitude and varying phase

c) Complex perturbation with varying real and imaginary part

Figure 3.7: Real perturbations Δ^R and complex perturbation Δ^C with radius $|w_I(j\omega)| = 1$ and uniformly distributed samples

Complex uncertainty: A *complex uncertainty* $\Delta(j\omega) = \Delta^C \in \mathbb{C}$ describes at the frequency ω the uncertainty of a parameter, which is uncertain in gain and/or phase, such that $||\Delta^C(j\omega)||_\infty \leq 1$. The *complex uncertainty* depends on the frequency ω and can change gain and/or phase at different frequencies. For that reason, the complex perturbation can be represented as a complex number or a transfer function $\Delta(j\omega)$.
Fig. 3.7(b) shows exemplary an uncertainty which changes the phase only. In contrast, Fig. 3.7(c) illustrates the combination of Fig. 3.7(a) and (b), which is a *complex uncertainty* within the complex plane. The circle represents $||\Delta(j\omega)||_\infty \leq 1$ and the circle center is the nominal value of the uncertain parameter at the frequency ω. Since the complex perturbation can change the real and the imaginary part of the nominal parameter, it can change gain and phase at the frequency ω. As a result, the complex perturbation Δ^C covers all uncertainties within the circle $||\Delta(j\omega)||_\infty \leq 1$. The blue circle markers illustrate uniformly distributed samples as proper uncertainty examples, which lie within the circle $||\Delta(j\omega)||_\infty \leq 1$. Equivalently to the real perturbation, the circle radius of the complex perturbation and thus the uncertainty range can be adapted by weighting functions. These weighting functions are introduced within the next section.

In order to analyze a perturbed system, uncertainties from the μ-analysis can be inserted into the model as real or complex number, or as transfer functions $\Delta(s)$. The calculation of the worst-case transfer function is explained in Sec. 3.8.2.



(a) Samples from uniform distribution (truncated)

(b) Samples from Normal distribution (unbounded)

Figure 3.8: Truncated and unbounded stochastic uncertainties in comparison

3.3.4 Uncertainties with Limited Intervals versus Unbounded Stochastic Uncertainties

The real and complex uncertainties with limited uncertainty intervals are the classical ways to express uncertainties mathematically. The uncertainties are modeled, such that their samples only occur within dedicated intervals. No probability density functions are attached in the uncertainty definition in the regular case. This is contrary to unbounded stochastic uncertainties, which are represented by probability distributions with unlimited support (e.g. normal distributions, student-t distributions). They are considered also within this thesis for robustness assessment (see Sec. 3.9). Samples from the respective probability density functions do not occur necessarily within a specific interval. Fig. 3.8 contrasts both modeling approaches exemplarily. Fig. 3.8(a) shows samples from a two-dimensional uniform distribution. The samples occur only within limited intervals, which is in this case a rectangle. Fig. 3.8(b) illustrates samples from a two-dimensional multivariate Gaussian distribution. The three circles represent multiple standard deviations (i.e. σ, 2σ and 3σ). It is clearly visible that samples occur outside of dedicated intervals. In case of unbounded stochastic uncertainties the dilemma with limited intervals can be on the one hand that the solution of the classical μ-analysis can be too optimistic or on the other hand the results can be conservative. Either outliers are not taken into account in the limited intervals, or the occurrence of unlikely uncertainties is considered equivalently as the occurrence of probable uncertainties. Both types of uncertainties are considered further within this thesis. The uncertainties with limited intervals are utilized in the classical μ-analysis. The new proposed probabilistic-μ approach within this thesis in Sec. 3.9 considers unbounded stochastic intervals. Note that the new proposed probabilistic-μ approach can be extended in future work to also consider stochastic uncertainties with limited intervals and probability density function attachment.

3.3.5 Structured Uncertainty versus Unstructured Uncertainties

The above described scalar perturbations within limited intervals are embedded as scalar or as part of multidimensional uncertainties in MIMO systems. The uncertainty structure of $\underline{\boldsymbol{\Delta}}$ defines the way how the uncertainty $\underline{\boldsymbol{\Delta}}$ affects the input signals or the output signals of the considered, nominal model. It is distinguished between unstructured and structured uncertainties. Unstructured uncertainties are mainly used to create a simple uncertain model. For detailed robustness statements, structured uncertainties are typically appropriate.

Unstructured Uncertainty: The multidimensional uncertainty $\underline{\boldsymbol{\Delta}}$ is denoted as *unstructured uncertainty*, if the matrix $\underline{\boldsymbol{\Delta}}$ is full and complex, such that

$$\underline{\boldsymbol{\Delta}} = \begin{bmatrix} \Delta_{11} & \cdots & \Delta_{1n} \\ \vdots & \ddots & \vdots \\ \Delta_{n1} & \cdots & \Delta_{nn} \end{bmatrix} \tag{3.25}$$

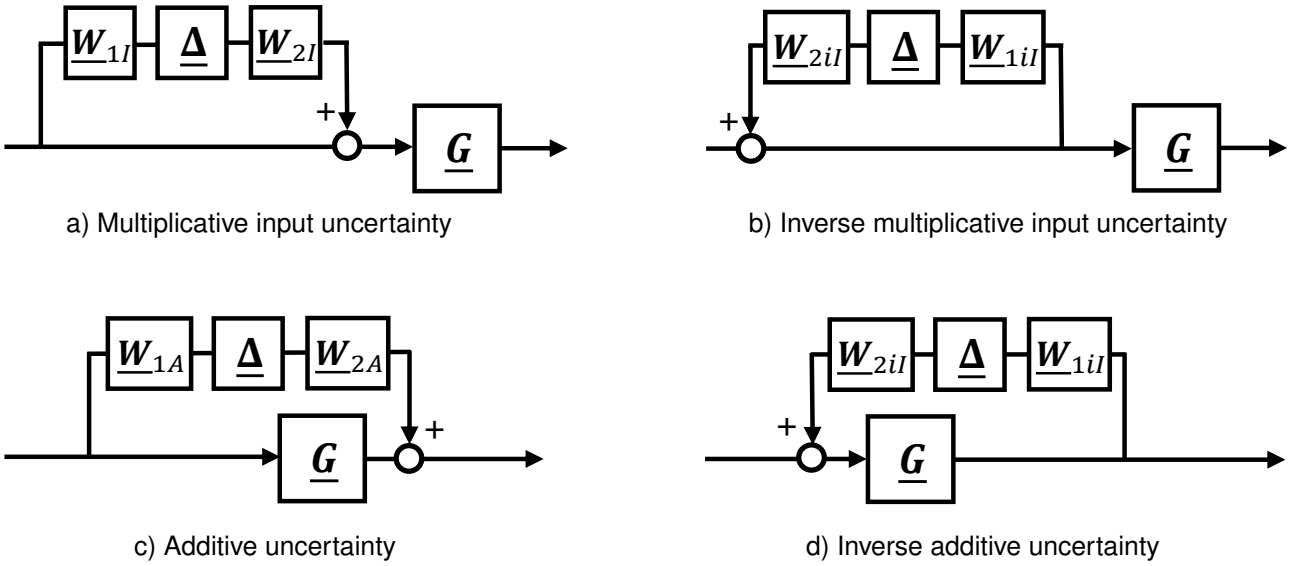
with $\underline{\boldsymbol{\Delta}} \in \mathbb{C}^{n\times n}$. The dimension is chosen such that the uncertainty fits properly into the model. According to Ref. [86], unstructured uncertainties are utilized to obtain a simple uncertainty model. However, this could lead to conservative results, since coupling effects might be modeled by the unstructured uncertainty, which are not physically present.

Structured Uncertainty: The uncertainty $\underline{\boldsymbol{\Delta}}$ is denoted as *structured uncertainty*, if the matrix is a block-diagonal matrix with $\underline{\boldsymbol{\Delta}} = \mathrm{diag}\{\underline{\Delta}_{ii}\}$, such that

$$\underline{\boldsymbol{\Delta}} = \begin{bmatrix} \underline{\Delta}_{11} & 0 & \cdots & 0 \\ 0 & \underline{\Delta}_{22} & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \underline{\Delta}_{nn} \end{bmatrix} \quad \text{with} \quad \underline{\boldsymbol{\Delta}} \in \mathbb{C}^{n\times n},\ n \in \mathbb{N}^{+} \tag{3.26}$$

The entries $\underline{\Delta}_{ii}$ are underlined, in order to emphasize that these entries can be matrices themselves. However, the diagonal entries can have a scalar expression. In contrast to the unstructured uncertainty, the uncertainty matrix of the structured uncertainty has a block diagonal structure. Sec. 3.7 introduces the block diagonal structured uncertainty matrices in a more detailed mathematical way in the context of the structured singular value μ_{Δ}. Within this section, the mathematical definition is not shown explicitly. This section shall only contrast unstructured and structured uncertainty matrices.

As stated in the generic uncertainty definition in Sec. 3.3.2, the structured and unstructured uncertainties are typically normalized, such that $||\underline{\boldsymbol{\Delta}}||_\infty \leq 1$. The uncertainty range is adapted by so-called weighting functions (see Sec. 3.3.6). The distinction between structured and unstructured uncertainty becomes important as the structure is integrated into the model. This is further explained in the next Sec. 3.3.6. However, it influences also the μ-boundary results, since considering block-diagonal uncertainty matrices can lead to tighter μ-boundaries, which is beneficial for being less conservative (see Sec. 3.7).
Note that, within this thesis and within the application section in Sec. 3.10, each uncertainty is modeled as a scalar uncertainty. However, these scalar uncertainties can be unified to structured uncertainties. Unstructured uncertainties are not used in Sec. 3.10.



a) Multiplicative input uncertainty

b) Inverse multiplicative input uncertainty

c) Additive uncertainty

d) Inverse additive uncertainty

Figure 3.9: Different forms of uncertainties with weighting matrices

3.3.6 Uncertainty Representation

There are different ways how uncertainties are embedded within the model structure, since the uncertainty can affect the nominal model in various ways. This section elaborates, how the uncertainties can be structurally integrated in the model. Important possibilities are illustrated in Fig. 3.9. The uncertainty representations are introduced for MIMO uncertainties $\underline{\boldsymbol{\Delta}}$, which are multiplied with weighting matrices. The introduced uncertainty representations express the most generic cases. However, further within this thesis, these representations are mainly applied for scalar uncertainties with SISO weighting transfer function. Thereby, the relation to the real uncertainty parameters is kept straightforward.
The introduced uncertainty representation can be used with structured or unstructured uncertainties. As mentioned above, unstructured uncertainties are utilized to obtain simple uncertainties models and are not used within this thesis. For detailed uncertainty models structured uncertainties are more appropriate. The explained uncertainty representation is on the basis of Ref. [86, Ch.8] and is recapitulated next. The uncertainty representation depends on the uncertainty modeling and on the meaning of the uncertain parameter. The uncertainty modeling is covered in Sec. 3.3.7.

Multiplicative Uncertainties

Uncertainties are called *multiplicative input uncertainties* if they are integrated such that

$$\underline{\boldsymbol{G}}_P(s) = \underline{\boldsymbol{G}}(s)\big(\underline{\boldsymbol{I}} + \underline{\boldsymbol{W}}_{2I}(s)\ \underline{\boldsymbol{\Delta}}(s)\ \underline{\boldsymbol{W}}_{1I}(s)\big). \tag{3.27}$$

with the weighting matrices $\underline{\boldsymbol{W}}_{1I}(s)$ and $\underline{\boldsymbol{W}}_{2I}(s)$. In this way, the uncertainty affects the input signal, in a multiplicative manner. Equivalently, a *multiplicative output uncertainty* affects the

output signal of the model $\underline{\boldsymbol{G}}$, such that

$$\underline{\boldsymbol{G}}_P(s) = (\underline{\boldsymbol{I}} + \underline{\boldsymbol{W}}_{2O}(s)\ \underline{\boldsymbol{\Delta}}(s)\ \underline{\boldsymbol{W}}_{1O}(s))\underline{\boldsymbol{G}}(s). \tag{3.28}$$

The weighting functions $\underline{\boldsymbol{W}}_{1I}(s), \underline{\boldsymbol{W}}_{2I}(s)$ and $\underline{\boldsymbol{W}}_{1O}(s), \underline{\boldsymbol{W}}_{2O}(s)$ are MIMO stable transfer functions, which specify further the magnitude (or magnitude range) of the uncertainty. The uncertainty structure defines how the model inputs / outputs are affected. In case the uncertainty matrix is a diagonal matrix, the input / output signals remain independently. If the uncertainty matrix is a full uncertainty matrix, additional signal cross-coupling effects occur. Multiplicative uncertainties occur, if the uncertainties are modeled inter alia as parametric (e.g. gain uncertainty) or lumped uncertainties. These are two uncertainty modeling approaches, which are all explained in Sec. 3.3.7.

Inverse Multiplicative Uncertainties

Uncertainties can affect the model in an inverse (feedback) way. If the uncertainty occurs at the input, it is denoted as *inverse multiplicative input uncertainty*:

$$\underline{\boldsymbol{G}}_P(s) = \underline{\boldsymbol{G}}(s)(\underline{\boldsymbol{I}} - \underline{\boldsymbol{W}}_{2iI}(s)\ \underline{\boldsymbol{\Delta}}(s)\ \underline{\boldsymbol{W}}_{1iI}(s))^{-1} \tag{3.29}$$

If the uncertainty occurs at the output, it is named as *inverse multiplicative output uncertainty*:

$$\underline{\boldsymbol{G}}_P(s) = (\underline{\boldsymbol{I}} + \underline{\boldsymbol{W}}_{2iO}(s)\ \underline{\boldsymbol{\Delta}}(s)\ \underline{\boldsymbol{W}}_{1iO}(s)(s))^{-1}\underline{\boldsymbol{G}}(s) \tag{3.30}$$

Some parametric uncertainties can be modeled as inverse multiplicative uncertainties. This includes, for example, the uncertainty in a time constant of a first order low-pass filter. Analogously to the multiplicative uncertainties, the uncertainty structure defines how the input/output signals are coupled.

Additive Uncertainties

Uncertainties can also influence the model in an additive manner. They are called *additive uncertainties* and are expressed with:

$$\underline{\boldsymbol{G}}_P(s) = \underline{\boldsymbol{G}}(s) + \underline{\boldsymbol{W}}_{2A}(s)\ \underline{\boldsymbol{\Delta}}(s)\ \underline{\boldsymbol{W}}_{1A}(s) \tag{3.31}$$

This type of uncertainty is similar to the multiplicative input uncertainty. If Eq. (3.31) and 3.27 are compared, the additive uncertainty can be converted to a multiplicative input uncertainty by adjusting the weighting function accordingly.

Inverse Additive Uncertainties

The output signal of the model can be also fed back to the input of the model by an uncertainty, which is called *inverse additive uncertainty*:

$$\underline{\boldsymbol{G}}_P(s) = \underline{\boldsymbol{G}}(s)(\underline{\boldsymbol{I}} - \underline{\boldsymbol{W}}_{2iA}(s)\ \underline{\boldsymbol{\Delta}}(s)\ \underline{\boldsymbol{W}}_{1iA}(s)\underline{\boldsymbol{G}}(s))^{-1} \tag{3.32}$$

The presented four uncertainty integration methods describe how multidimensional uncertainties can be embedded into the model. The next section explains how physical model parameter can be expressed as one of these four uncertainty representations and how the corresponding weighting functions are calculated.

3.3.7 Uncertainty Modeling

Depending on the origin, uncertainties can be parted in different classes. These classes are distinguished by their modeling method. The classes define the way how uncertain model parameters are expressed, such that they can be embedded as one of the uncertainty representations introduced above. The subsequent presented classes are utilized in the conventional and advanced μ-analysis and represent uncertainties within limited intervals. Four important uncertainty classes are presented within this section, based on the work in Ref. [86, Ch. 7]. The section is concluded with a short discussion, whether the presented classes can also be used under the consideration of unbounded stochastic uncertainties.

3.3.7.1 Uncertainty Classes within Limited Intervals

Parametric Uncertainties

This class of perturbation can be used, if the dynamical structure of the uncertain system is known, but the exact values of the parameters are unknown. A parametric uncertainty α is specified by the bounded interval $[\alpha_{min}, \alpha_{max}]$ and the mean (nominal) value $\bar{\alpha}$. The uncertainty is then represented by the equation

$$\alpha = \bar{\alpha}(1 + r_\alpha \Delta_\alpha) \tag{3.33}$$

with the relative weight $r_\alpha = \frac{\alpha_{max} - \alpha_{min}}{\alpha_{max} + \alpha_{min}}$, and the scalar perturbation with $||\Delta_\alpha||_\infty \leq 1$. How Eq. (3.33) is integrated into the nominal model, such that $\underline{\boldsymbol{G}}_p \in \underline{\underline{\boldsymbol{\Pi}}}$, depends on the meaning of the parametric uncertainty itself. Possible integration approaches are discussed in Sec. 3.3.6. Parametric uncertainties typically are specified as *real uncertainty* (compare to Sec. 3.3.3). This is due to the fact that parametric uncertainties are used in the nominal models for variables, which are real numbers by definition. They are expressed by multiplicative uncertainties or inverse multiplicative uncertainties.

For clarification, parametric uncertainties are shown in the following section, on the basis of an uncertain actuator model including dead time in Ref. [89, 90]. The actuator is modeled as a second order low pass filter:

$$G(s) = k \cdot \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \cdot e^{-\theta s} \tag{3.34}$$

The gain k, the relative damping ζ, and the eigenfrequency ω_0 are uncertain. The unknown deadtime θ represents an additional uncertain parameter. The respective parametric uncertainties of the actuator model and their representation are summarized in Tab. 3.1.

Fig. 3.10 illustrates the parametric uncertainties of the actuator model in a Nyquist plot. The line in green color represents the nominal transfer function. Each blue line illustrates one per-

Parameter	Symbol	Mean	Interval	Relative Weight	Uncertainty
Frequency	ω_0	$\bar{\omega}_0$	$[\omega_{min}, \omega_{max}]$	$r_\omega = \frac{\omega_{max}-\omega_{min}}{\omega_{max}+\omega_{min}}$	$\Delta_\omega \in \mathbb{R}$ with $\lvert\Delta_\omega\rvert \leq 1$
Damping	ζ	$\bar{\zeta}$	$[\zeta_{min}, \zeta_{max}]$	$r_\zeta = \frac{\zeta_{max}-\zeta_{min}}{\zeta_{max}+\zeta_{min}}$	$\lvert\Delta_\zeta\rvert \leq 1$
Gain	k	$\bar{k}$	$[k_{min}, k_{max}]$	$r_k = \frac{k_{max}-k_{min}}{k_{max}+k_{min}}$	$\lvert\Delta_k\rvert \leq 1$
Dead Time	θ	$\bar{\theta}$	$[\theta_{min}, \theta_{max}]$	$r_\theta = \frac{\theta_{max}-\theta_{min}}{\theta_{max}+\theta_{min}}$	$\lvert\Delta_\theta\rvert \leq 1$

Table 3.1: Parametric uncertainties within actuator model

turbed transfer function in the uncertainty set. The given parameters in Tab. 3.1 are varied over the whole uncertainty set in a discrete manner. Each blue line is a specific combination of the parametric uncertainties. The red diamonds mark the parametric uncertainties at four different frequencies. As Fig. 3.10 shows, the parametric uncertainties cover a particular region in the complex plane. By the combination of several real perturbations, it is possible to cover certain regions at a specific frequency in the Nyquist plot.
Parametric uncertainties with limited intervals are often replaced by lumped uncertainties, which are explained next.

Lumped Uncertainties

Parametric uncertainties with limited interval can be joined to one lumped uncertainty. The lumped uncertainty is specified by the uncertainty $||\boldsymbol{\Delta}_I||_\infty \leq 1$ and the weighting function $w_I(s)$. Typically, the respective transfer function $\underline{\boldsymbol{G}}_P \in \underline{\underline{\boldsymbol{\Pi}}}$ is represented using a multiplicative input or output uncertainty. The multiplicative uncertainty is one of a few uncertainty integration possibilities, which are explained in Sec. 3.3.6.
The weighting function depends on the frequency $s = j\omega$ and is established such that any parametric uncertainty is covered by the magnitude of the weighting function $w_I(s)$. In case of a scalar uncertainty, the magnitude $|w_I(j\omega_i)|$ (i.e. absolute value) represents the radius of a complex uncertainty at the frequency ω_i in the complex plane since the weighting function is multiplied by the uncertainty $||\Delta_I||_\infty \leq 1$.
This fact is illustrated in Fig. 3.10(b). The parametric uncertainties of the actuator example are visualized at four different frequencies in red diamonds. The parametric uncertainties can be replaced by one lumped uncertainty. The lumped uncertainty is displayed by the four circles with radius $w_I(s)$ at these four frequencies ω_i. Depending on the frequency, the magnitude of the weighting function changes.
The approach is rather conservative, since the lumped uncertainty covers a larger uncertainty set than the parametric uncertainties. But the conservatism might be justified, since

the assumed model has a quite specific structure, which might not be substantiated.

The calculation of the weighting function $w_I(s)$ is explained next. Therefore, it is searched for the maximal relative error $l(\omega)$ between the nominal plant $\underline{\boldsymbol{G}}(s)$ and all possible perturbed transfer functions $\underline{\boldsymbol{G}}_P(j\omega)$, which are obtained by varying the parametric uncertainties over the whole possible set. If the nominal model is scalar, the error can be calculated with:

$$l_i(\omega) = \max_{G_p \in \Pi} \left| \frac{G_p(j\omega) - G(j\omega)}{G(j\omega)} \right|, \quad \text{such that } |w_I(j\omega)| \geq l_i(\omega),\ \forall\omega \tag{3.35}$$

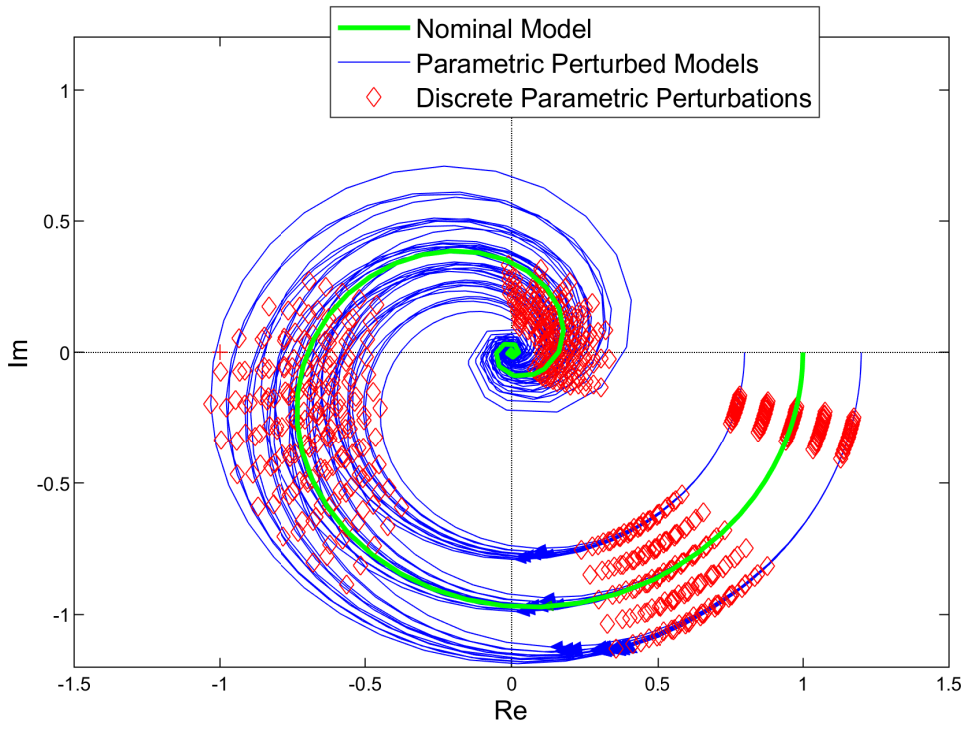
In multidimensional cases, the error is calculated with:

$$\text{Input:} \quad l_I(\omega) = \max \bar{\sigma}\left(\underline{\boldsymbol{G}}^{-1}\left(\underline{\boldsymbol{G}}_P - \underline{\boldsymbol{G}}\right)\right), \quad \text{such that } |w_I(j\omega)| \geq l_I(\omega),\ \forall\omega \tag{3.36}$$

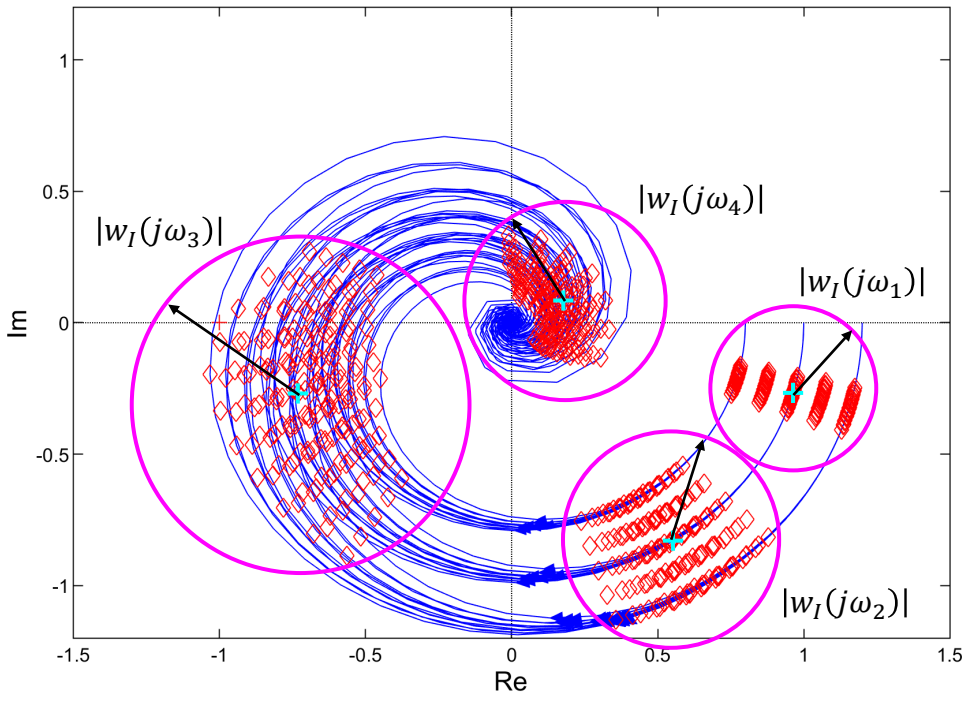
$$\text{Output:} \quad l_O(\omega) = \max \bar{\sigma}\left(\left(\underline{\boldsymbol{G}}_P - \underline{\boldsymbol{G}}\right)\underline{\boldsymbol{G}}^{-1}\right), \quad \text{such that } |w_O(j\omega)| \geq l_O(\omega),\ \forall\omega \tag{3.37}$$

The magnitude of the corresponding weighting functions needs to be equal or greater than the relative error over all frequencies.

The weighting functions are designed, such that the gain of the weighting function $|w_I(j\omega)|$, $|w_O(j\omega)|$ is always equal or greater than the maximal relative error $|l_I(\omega)|$, $|l_O(\omega)|$. According to Ref. [86], in order to obtain an appropriate weighting function, it can be started with a simple first order approximation, which can be adapted successively by increasing the complexity and order. The weighting function shall be close to the error function to be less conservative, but it shall also cover all perturbed models. The Refs. [86, 89] explain in detail, how weighting function candidates are obtained under consideration of the maximal relative error $\max\ |l(\omega)|$. In this thesis and also for the present example, the procedures described in Ref. [86, 89] are applied.
The parametric uncertainties of the example in Eq. (3.34) are varied over their whole intervals in a discrete manner. The resulting perturbed plant models are displayed as magnitude and phase plot in the first two subplots of Fig. 3.11. These lines are equivalent to the blue lines in the Nyquist plot in Fig. 3.10. The error $|l(\omega)|$ is calculated for each perturbed model and plotted in the last subplot. In the last subplot in Fig. 3.11 the maximal relative error $\max\ |l(\omega)|$ using Eq. (3.35) together with one calculated weighting function candidate is shown. Fig. 3.10(b) shows in addition the magnitude of a weighting function candidate for the four discrete frequencies, which are visualized explicitly for respective discrete parametric perturbations in Fig. 3.10(a), mentioned above (i.e. red diamonds). The weighting function candidate needs to cover all possible parametric uncertainties at the corresponding frequency. This is visualized by the four circles for each discrete frequency in magenta color.



(a) Discrete parametric uncertainties at four different frequencies ω



(b) Lumped uncertainties with weighting function at four different frequencies ω

Figure 3.10: Nyquist plot of uncertain actuator model

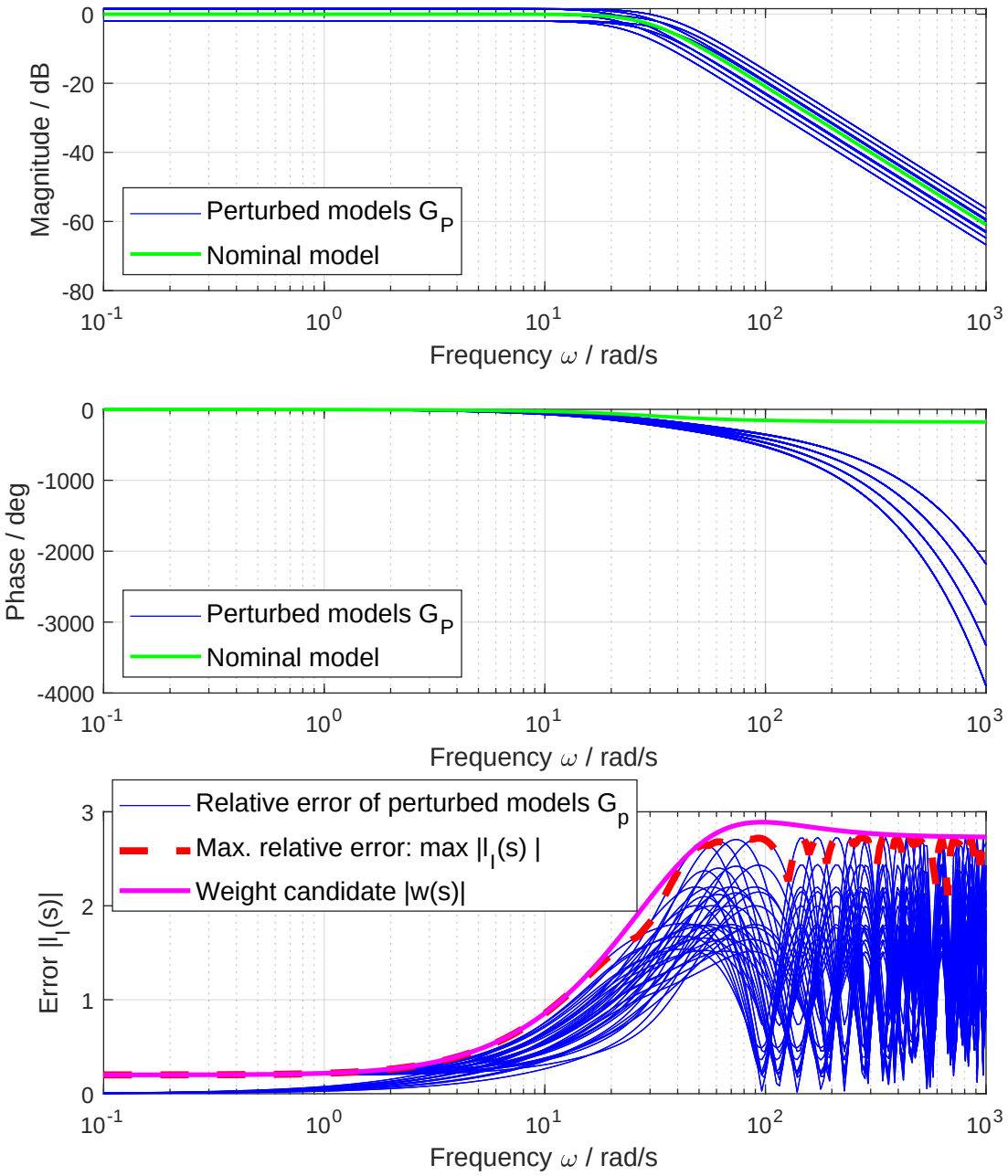


Figure 3.11: Design of weighting function $w_I(s)$

Neglected Dynamics

Neglected dynamics are dynamics, which are not considered in the nominal model. This can include, for example, lags or deadtimes. These neglected dynamics can be taken into account in the robustness and uncertainty analysis. They are represented as complex perturbations. The uncertainty set of plants is written as:

$$\underline{\boldsymbol{G}}_P(s) = \underline{\boldsymbol{G}}(s)\underline{\boldsymbol{f}}(s) \tag{3.38}$$

with the neglected function $\underline{\boldsymbol{f}}(s)$. Equivalently to the lumped uncertainties, the neglected dynamics are integrated into the nominal system by the utilization of weighting functions. With Eq. (3.35) and Eq. (3.38), the error becomes in the scalar case

$$l_N(\omega) = \max_{G(s)f(s)\in\Pi} |f(j\omega) - 1|. \tag{3.39}$$

Analogously to the section above, the error function can be approximated by a weighting function.

Unmodeled Dynamics

Unmodeled dynamics are similar to neglected dynamics, but they further include dynamics which are completely unknown. This might be especially the case at higher frequencies. Unmodeled dynamics are represented as complex perturbation using respective weighting functions. Ref. [86, p.269] suggests the following simple weighting function:

$$w_U(s) = \frac{\tau s + r_0}{(\tau/r_\infty)s + 1} \tag{3.40}$$

where r_0 is the relative uncertainty value at steady state after a step input. The variable τ represents the respective time constant of the weight. r_∞ is the weight gain at high frequencies.

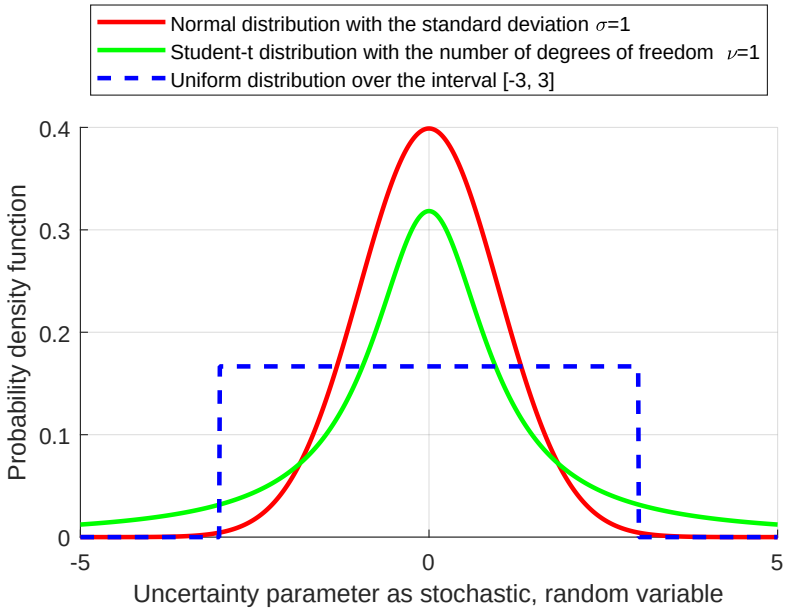


Figure 3.12: Exemplary probability density functions with limited and unlimited support

3.3.7.2 Uncertainty Classes under Consideration of Unbounded Stochastic Uncertainties

Unbounded stochastic uncertainties can be considered in the novel probabilistic-μ-approach. In contrast to uncertainties with limited interval, unbounded stochastic uncertainties do not have interval boundaries. They are defined by a probability density function with unlimited support. The uncertainty can thus not necessarily be located within a specific and limited interval. This becomes visible in Fig. 3.12, which shows the probability density function of three distributions. The uniform distribution covers a specific and limited interval. However, the Normal and Student't distribution are functions with unlimited support. Limited intervals cannot be identified for these distributions without neglecting unlikely, but relevant samples. It appears that unbounded stochastic uncertainties are appropriate as parametric uncertainties. Since the three remaining presented classes (i.e. lumped, neglected, unmodeled uncertainties) are not parameter specific or take several parametric uncertainties into account, it is difficult to find their stochastic and non-conservative representation, especially if the uncertainty is unbounded. In consequence, only parametric uncertainties are used in the probabilistic-μ approach.

3.4 Stability Criteria

The central design challenge for feedback controllers is to ensure stability and robustness of the closed-loop system. Different stability definitions exist (e.g. Lyapunov stability, static/ dynamic stability, bounded-input-bounded-output stability), but for linear systems the difference has no practical relevance. For the sake of completeness and in order to give an overview, the stability definition used within this work and its main criteria are elaborated next, based on the work in Ref. [86, Sec. 2.3, Sec. 4.9]. Robustness criteria are introduced in Sec. 3.5. Within this work, stability of linear time-invariant systems is defined as:

Definition: *"A system is stable if none of its components contain unstable modes and the injection of bounded external signals at any place in the system result in bounded output signals measured anywhere in the system."* (Ref. [86, p.127])

Three major criteria can be applied for the stability assessment of linear time-invariant closed-loop systems, which are elaborated next.

3.4.1 Poles

It is well-known that closed-loop systems are stable if their poles are in the left half plane. The poles are the roots of $\det(\underline{\boldsymbol{L}}(s)+1)=0$ (considering $\underline{\boldsymbol{L}}(s)$ as the MIMO open-loop transfer function with negative feedback loop) and the eigenvalues of the system matrix $\underline{\boldsymbol{A}}$ of the closed-loop linear state space model. This can be observed by looking at the analytical solution of the closed-loop state space model. The linear closed-loop state model is defined as:

$$\dot{\boldsymbol{x}}(t) = \underline{\boldsymbol{A}}\boldsymbol{x}(t) + \underline{\boldsymbol{B}}\boldsymbol{u}(t) \tag{3.41}$$

with the ouput equation

$$\mathbf{y}(t) = \underline{\boldsymbol{C}}\boldsymbol{x}(t) + \underline{\boldsymbol{D}}\boldsymbol{u}(t). \tag{3.42}$$

The analytical solution of the state-space model is:

$$\boldsymbol{x}(t) = \exp\left(\underline{\boldsymbol{A}}(t-t_0)\right)\boldsymbol{x}(t_0) + \int_{t_0}^{t} \exp\left(\underline{\boldsymbol{A}}(t-\tau)\right)\underline{\boldsymbol{B}}\boldsymbol{u}(\tau)d\tau \tag{3.43}$$

with

$$\exp\left(\underline{\boldsymbol{A}}\cdot t\right) = \sum_{k=1}^{n} t_i \exp\left(\lambda_i \cdot t\right) q_i^H \tag{3.44}$$

with left t_i and right q_i eigenvectors of the system matrix $\underline{\boldsymbol{A}}$ (based on Ref. [86, p.114]). Thereby, the internal stability of the open-loop system needs to be considered (see Ref. [86, p.139]) to assess the overall stability of the closed-loop system. The eigenvalues λ_i repre-

sent the poles of the closed-loop system and are thus decisive for the closed-loop stability. In general, the poles can be calculated numerically using the system state-space matrix $\underline{\boldsymbol{A}}$ of the closed-loop system to analyze the stability. Unfortunately, this method has its limitation, if the system contains time delays, unless the time delay is approximated by a Padé approximation.

3.4.2 Nyquist Criterion

Since evaluating the poles of the open- and closed-loop system is not sufficient for systems with time delay elements, an important stability criterion is the Nyquist theorem. The generalized MIMO Nyquist Theorem is not in the scope of this thesis. It is elaborated in detail for example in Ref. [86, p.147ff.]. The Nyquist theorem is important for the stability condition elaboration of the μ-analysis.

3.4.3 Bode's Stability Condition

In case of a SISO open-loop transfer function $L(s)$, if the phase $\angle L(j\omega_c)$ of a stable open-loop transfer function $L(j\omega)$ crosses -180 deg only once, it is sufficient to apply the Bode stability criterion for stability of the closed-loop system.

Bode Stability Condition (Ref. [86, p.25]):
The closed-loop system is stable if and only if the loop gain $|L|$ is less than 1, at the phase crossover frequency ω_{180}, such that $\angle L(j\omega_{180}) = -180$ deg. That means:

$$|L(j\omega_{180})| < 1 \tag{3.45}$$

The Bode Stability Condition can be only applied to one-dimensional open-loop transfer functions.

3.5 Robustness and Performance Criteria

As elaborated above, it is necessary to analyze the stability of the linear closed-loop system. However, it is even more important to evaluate how much margin has the system towards the stability limit. Therefore, stability robustness criteria are developed for SISO and MIMO systems. First, the classical SISO robustness metrics are elaborated. Subsequently, the Nichols disk margin is discussed. Finally, Sec. 3.5.3 and Sec. 3.5.4 introduce three different MIMO gain and phase margins.

3.5.1 SISO Robustness Metrics

3.5.1.1 Gain and Phase Margin

SISO Gain Margin:
The gain margin derivation is based on Ref. [86, p.31]. The (upper) gain margin is defined as:

$$\mathrm{GM_U} = 1/|L(j\omega_{180})| \tag{3.46}$$

The phase crossover frequency ω_{180} is defined as the frequency where the Nyquist curve $L(j\omega)$ crosses the negative real axis between -1 and 0, such that $\angle L(j\omega_{180}) = 180$ deg. If multiple crossings within this interval exist, the closest crossing to -1 is taken. The gain margin is the factor by which the loop gain $|L(j\omega)|$ may be increased until the system becomes unstable.
For unstable plants, the curve $L(j\omega)$ can cross the real axis also within the interval between $-\infty$ and -1, and a lower gain margin G_L is defined as:

$$\mathrm{GM_L} = 1/|L(j\omega_{L,180})| \tag{3.47}$$

at $\omega_{L,180}$, wich is the frequency, where the curve $L(j\omega)$ crosses the real axis between $-\infty$ and -1.

SISO Phase Margin:
The phase margin is defined as (Ref. [86, p.32]):

$$\mathrm{PM} = \angle L(j\omega_c) + 180 \text{ deg} \tag{3.48}$$

The frequency ω_c is the frequency where $|L(j\omega)|$ crosses the unit circle with the center in the origin, i.e. $|L(j\omega_c)| = 1$. The phase margin indicates how much phase lag can be added to $L(j\omega)$ at the frequency ω_c.

SISO Delay Margin:
From the phase margin, the so-called delay margin can be derived. The delay margin represents the time the signal can be delayed such that instability occurs. It can be calculated by:

$$\mathrm{DM} = \mathrm{PM}/\omega_c \tag{3.49}$$

3.5.1.2 Bandwidth

The bandwidth of a closed-loop system is an important measure within feedback control. The bandwidth specifies the frequency range in which the controller is effective. A low

bandwidth is the indication for systems with slow rise time. In contrast, a high bandwidth exhibits a more sensitive system at higher frequencies and a fast rise time. According to [86, Sec. 2.4.5], it is mainly distinguished between three different bandwidth definitions in the analysis of a closed-loop system. Fig. 3.13 shows an illustrative control configuration for their elaboration. The following derivations are based on [86, Sec. 2.4.5] and result from Fig. 3.13.

$$e = r - y \tag{3.50}$$

$$\begin{aligned} y &= G(s)u + d && (3.51)\\ u &= K(s)e && (3.52)\\ y &= G(s)K(s)(r-y) + d && (3.53)\\ y(1+G(s)K(s)) &= G(s)K(s)r + d && (3.54) \end{aligned}$$

$$y = \frac{G(s)K(s)}{1+G(s)K(s)}r + \frac{1}{1+G(s)K(s)}d \tag{3.55}$$

The introduced bandwidths are based on SISO systems, but can also be applied to analyze MIMO systems by considering specific SISO transfer functions of the MIMO system. Note that the derived bandwidths in this section are not equivalent to the bandwidths used in the *Bandwidth Criterion* (Ref. [84]).

Sensitivity transfer function: However, considering Eq. (3.50) and Eq. (3.55) the tracking error is calculated by

$$e = r - y = \frac{1}{1+G(s)K(s)}r = S(s)r. \tag{3.56}$$

From this equation it can be said that the control error is minimized as long as the (output) sensitivity transfer function is effective. As a result, the bandwidth can be calculated based on the sensitivity transfer function. The bandwidth ω_B is defined as the frequency, where the amplitude of the sensitivity transfer function $|S| \approx 0.707 \approx$ -3 dB from above.

Complementary sensitivity transfer function: Equivalently, the bandwidth can be calculated based on the complementary sensitivity transfer function, instead of the sensitivity transfer function. This results from the following considerations:

$$y = \frac{G(s)K(s)}{1+G(s)K(s)}r = T(s)r \tag{3.57}$$

This leads to a different bandwidth definition. The control is effective as long as the the input

signal r acts on the output signal y. The complementary sensitivity transfer function T represents the relation between r and y. Thus, the bandwidth ω_{BT} is defined as the frequency, at which the amplitude of the transfer function T passes $|T| \approx 0.707 \approx$ -3 dB from above.

Crossover frequency ω_c: In addition to the two bandwidths ω_B and ω_{BT}, Ref. [86, p.37] defines the crossover frequency ω_c as a third bandwidth. The frequency ω_c is the gain crossover frequency, where $|L|$ passes 1 from above first:

$$|L(j\omega_c)| = 1 \tag{3.58}$$

This is equivalent to the frequency, which is used for the calculation of the SISO phase margin.

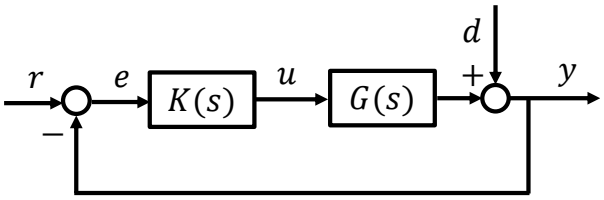
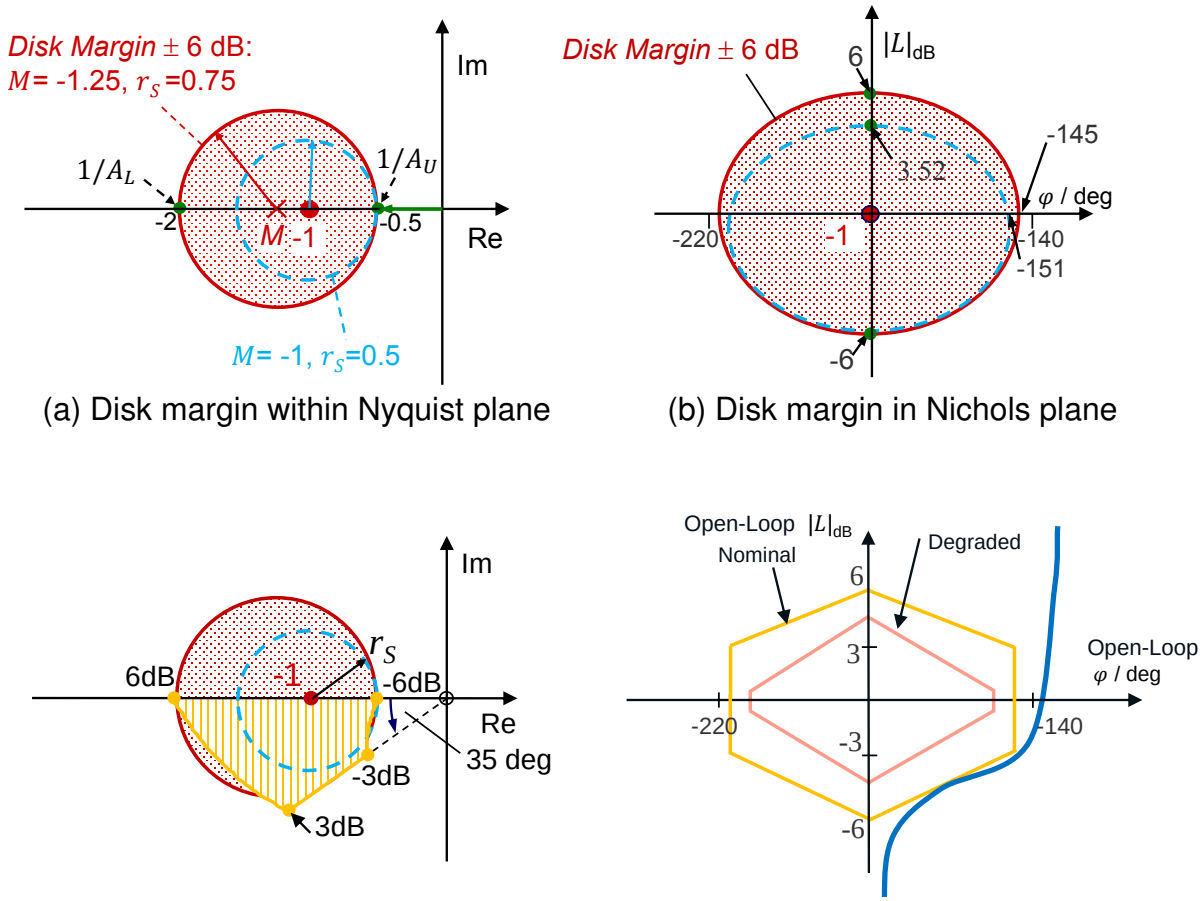


Figure 3.13: Illustrative SISO control configuration



(a) Disk margin within Nyquist plane

(b) Disk margin in Nichols plane

(c) Nichols diamond in Nyquist plane

(d) Nichols diamond in Nichols plane

Figure 3.14: Qualitative illustration of Nichols diamond derivation based on Ref. [87, p.42ff.]

3.5.2 SISO Nichols Disk Margin

The SISO gain margin indicates how much the open-loop transfer function can change in gain such that the system becomes unstable. The SISO phase margin states how much phase can be added to the system until it becomes unstable. A combination of the gain and the phase variation can also bring the system towards instability. Therefore, disk margins have been introduced (see e.g. Ref. [91, p. 75f.]). They make sure that, in case of a SISO system, the Nyquist plot has sufficient distance towards the critical point -1. The disk margin can have the shape of a circle with radius r_S in the complex plane to ensure a sufficient distance to the critical point. Instead of the disk margin in the complex plane, for robustness analysis in aerospace, the Nichols diamond has been established. The Nichols diamond got its name since it has literally the shape of a diamond in the Nichols diagram. The Nichols diamond represents also a disk margin, but instead of a circle it has the shape of a diamond. Fig. 3.14 demonstrates the derivation of the Nichols diamond from disk margins in the complex plane. The following derivation and the corresponding Fig. 3.14 is based on the work in Ref. [87, p.42ff.].

One can imagine to set a circle, for example, with radius $r_S = 0.5$ and the center at the critical point -1 as an appropriate disk margin (see blue, dashed circle in Fig. 3.14(a)). The drawback of such a circle is that the upper gain margin A_U and the lower gain margin A_L become asymmetric. This is visible in Fig. 3.14(b), which shows the circle transformed to the Nichols plane as blue, dashed line. This is observable by looking at the absolute upper and lower gain margin of the dashed, blue ellipse in Fig. 3.14(b), which is the transformed circle in Fig. 3.14(a).
In order to improve the asymmetry, the circle origin is shifted and the radius is increased, such that the absolute lower and upper gain margin become $\pm$6 dB (see red circle in Fig. 3.14(a)). Again, this circle can be transformed into the Nichols plane (see red ellipse in Fig. 3.14(b)). According to Ref. [87, p. 44ff.], especially in flight control for robustness analysis, such a disk margin with the shape of a circle in the complex plane is approximated by line curves, as visualized in Fig. 3.14(c). Note that these line curves are not necessarily straight, since the transformation between the Nichols plane and the complex plane is nonlinear. Transformed into the Nichols plane, the approximated area (illustrated as yellow, hatched area) becomes a Nichols diamond, which is shown in Fig. 3.14(d). The shown, greater diamond represents the disk margin for the nominal case with $\pm$6 dB lower and upper gain margin and a combined gain and phase margin with $\pm$3 dB gain and $\pm$35 deg phase. In Ref. [39], other Nichols diamond for off-nominal conditions are introduced.
The Nichols plot of a SISO open-loop transfer function $L(s)$ is not allowed to cut or contact the Nichols diamond to guarantee sufficient robustness margin. In order to obtain an appropriate open-loop transfer function from a MIMO closed-loop system, particular loop-breaking rules need to be considered, which are elaborated in Sec. 3.2.3.
The Nichols diamond is later used as disk margin for the robustness analysis of an eVTOL demonstrator within this work.

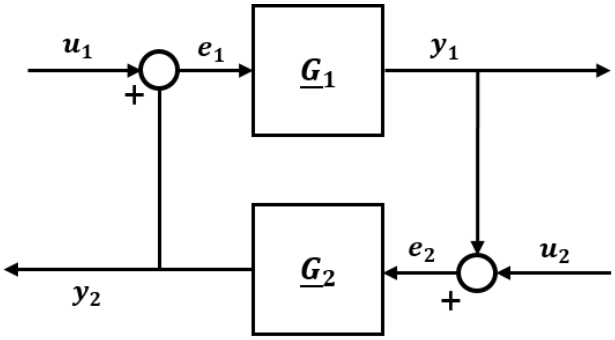


Figure 3.15: Small Gain Theorem (Ref. [42])

3.5.3 Generic MIMO Block Structure and its stability condition

3.5.3.1 Small Gain Theorem

The Small Gain Theorem is fundamental for stability analysis and for the further derivation of MIMO margins and the μ-analysis. For that reason, the weak version of the Small Gain Theorem and its proof is elaborated in the present section. The theorem and the proof is based on the work in Ref. [42] and visualized in Fig. 3.15.

Theorem (Ref. [42]): Assume that $\underline{\boldsymbol{G}}_1(s), \underline{\boldsymbol{G}}_2(s)$ are both stable, bounded-gain transfer function. Then if $||\underline{\boldsymbol{G}}_1(s) \cdot \underline{\boldsymbol{G}}_2(s)||_\infty < 1$, the closed-loop system is stable.
Furthermore, since $||\underline{\boldsymbol{G}}_1(s)\underline{\boldsymbol{G}}_2(s)|| \leq ||\underline{\boldsymbol{G}}_1||_\infty ||\underline{\boldsymbol{G}}_2||_\infty$, then:

$$||\underline{\boldsymbol{G}}_1||_\infty ||\underline{\boldsymbol{G}}_2||_\infty < 1 \Rightarrow \text{closed-loop is stable} \tag{3.59}$$

For the proof of the Small Gain Theorem, the transfer function from $\boldsymbol{u}_2$ to $\boldsymbol{e}_2$ is derived. We assume $\boldsymbol{u}_1 = 0$:

$$\boldsymbol{e}_2 = (\boldsymbol{I} - \underline{\boldsymbol{G}}_2\underline{\boldsymbol{G}}_1)^{-1}\boldsymbol{u}_2 \;\Leftrightarrow \boldsymbol{e}_2 = \boldsymbol{u}_2 + \underline{\boldsymbol{G}}_1\underline{\boldsymbol{G}}_2\boldsymbol{e}_2 \tag{3.60}$$

By taking the H_∞-norm, the following equation holds:

$$\Rightarrow ||\boldsymbol{e}_2||_\infty \leq ||\boldsymbol{u}_2||_\infty + ||\underline{\boldsymbol{G}}_1\underline{\boldsymbol{G}}_2||_\infty ||\boldsymbol{e}_2||_\infty \Leftrightarrow (1 - ||\underline{\boldsymbol{G}}_1\underline{\boldsymbol{G}}_2||_\infty)||\boldsymbol{e}_2||_\infty \leq ||\boldsymbol{u}_2||_\infty \tag{3.61}$$

The last equation holds, if

$$||\underline{\boldsymbol{G}}_1\underline{\boldsymbol{G}}_2||_\infty < 1 \Leftarrow ||\underline{\boldsymbol{G}}_1||_\infty ||\underline{\boldsymbol{G}}_2||_\infty < 1, \tag{3.62}$$

which is equivalent to the theorem above. If Eq. 3.62 is fulfilled, it is ensured, that the $||\boldsymbol{e}_2||$ is smaller than $||\boldsymbol{u}_2||$, which proofs the theorem. Eq. 3.62 is essential for the robustness analysis of uncertain MIMO systems.

3.5.3.2 Linear Fractional Transformation[8]

In order to apply the above Small Gain Theorem for the robustness and worst-case uncertainty evaluation of closed-loop systems, the Linear Fractional Transformation (LFT) is introduced next. A generic block structure is obtained by the LFT, which eases the robustness analysis in an universal manner.
An example of an arbitrary structure is shown in Fig. 3.16(a). The analysis for specific structures with uncertainties might be possible, but the generic approach enables the application of the robustness analysis to any system, although the physical meaning, as presented by the exemplary system, might disappear.
The LFT restructures the system to a generic block structure (see Fig. 3.16(b)) with the plant transfer function $\underline{\boldsymbol{P}}(s)$, the controller transfer function $\underline{\boldsymbol{K}}(s)$ and the uncertainty matrix $\underline{\boldsymbol{\Delta}}(s)$. The transformation collects any defined uncertainty parameter Δ_i into the matrix $\underline{\boldsymbol{\Delta}}(s)$ and any controller gain parameter into the transfer function $\underline{\boldsymbol{K}}(s)$. If the lower loop is closed (i.e. the control loop) of the new generic block structure, the so-called lower LFT-model can be calculated:

$$\boldsymbol{z} = \underline{\boldsymbol{F}}_l(\underline{\boldsymbol{P}}, \underline{\boldsymbol{K}})\boldsymbol{w} \tag{3.63}$$

$$\underline{\boldsymbol{F}}_l(\underline{\boldsymbol{P}}, \underline{\boldsymbol{K}}) = \underline{\boldsymbol{P}}_{22} + \underline{\boldsymbol{P}}_{23}\underline{\boldsymbol{K}}(\underline{\boldsymbol{I}} - \underline{\boldsymbol{P}}_{33}\underline{\boldsymbol{K}})^{-1}\underline{\boldsymbol{P}}_{32} \tag{3.64}$$

The lower LFT-model can be used for controller synthesis. If the upper loop is closed, the so-called upper LFT-model can be produced and computed:

$$\boldsymbol{z} = \underline{\boldsymbol{F}}_u(\underline{\boldsymbol{P}}, \underline{\Delta})\boldsymbol{w} \tag{3.65}$$

$$\underline{\boldsymbol{F}}_u(\underline{\boldsymbol{P}}, \underline{\Delta}) = \underline{\boldsymbol{P}}_{22} + \underline{\boldsymbol{P}}_{21}\underline{\boldsymbol{\Delta}}(\underline{\boldsymbol{I}} - \underline{\boldsymbol{P}}_{11}\underline{\boldsymbol{\Delta}})^{-1}\underline{\boldsymbol{P}}_{12} \tag{3.66}$$

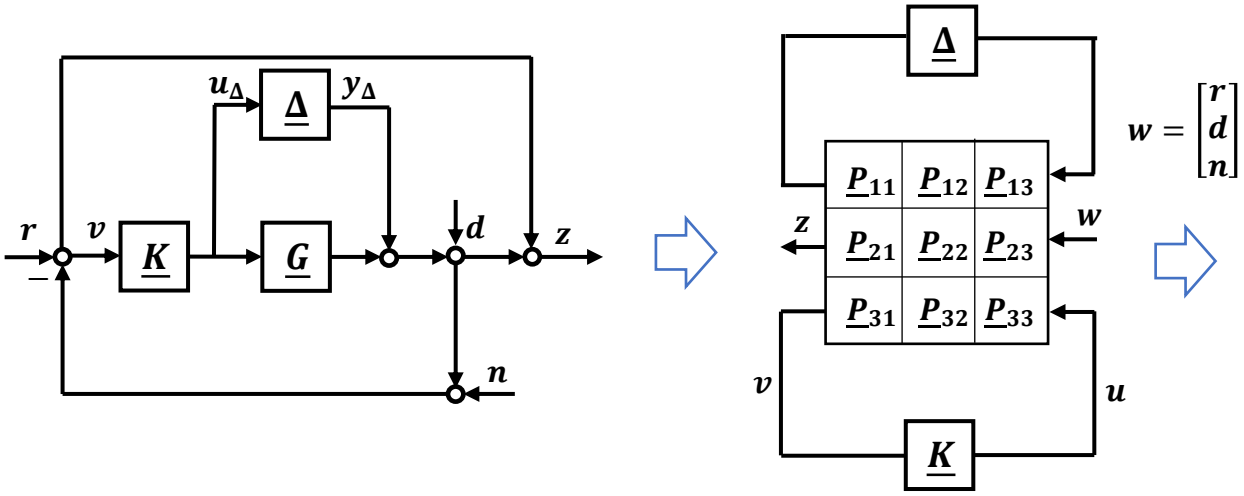
If the controller is already known, the controller can be unified with the plant. As a result, the block structure in Fig. 3.16(c) can be generated. The respective transfer function is:

$$\boldsymbol{z} = \underline{\boldsymbol{F}}_u(\underline{\boldsymbol{N}}, \underline{\Delta})\boldsymbol{w} \tag{3.67}$$

$$\underline{\boldsymbol{F}}_u(\underline{\boldsymbol{N}}, \underline{\Delta}) = \underline{\boldsymbol{N}}_{22} + \underline{\boldsymbol{N}}_{21}\underline{\Delta}(\underline{\boldsymbol{I}} - \underline{\boldsymbol{N}}_{11}\underline{\Delta})^{-1}\underline{\boldsymbol{N}}_{12} \tag{3.68}$$

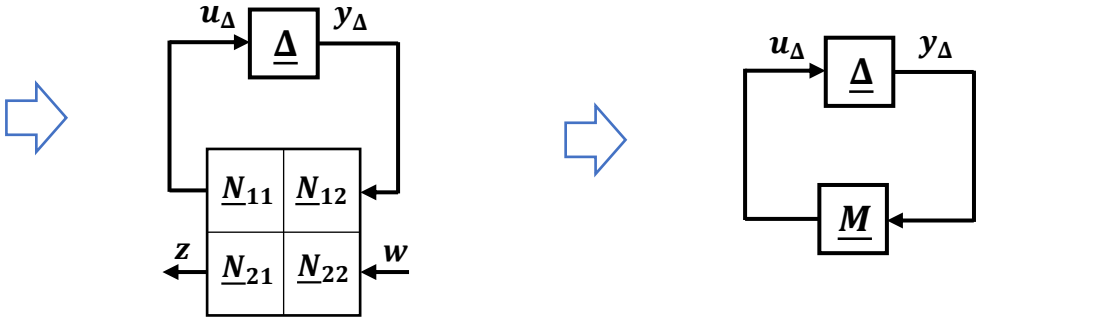
This transfer function is used for the robustness analysis in the following sections. The linear fractional transformation is explained in detail in Ref. [86, 92].

[8] The author has published this subsection as introduction to the new probabilistic-μ approach in Ref. [53]. This represents a revised version.



a) Linear feedback system with uncertainty (Ref. [93])

b) General control configuration



c) General block diagram with uncertainty

d) Block diagram for stability analysis

Figure 3.16: Linear Fractional Transformation

3.5.3.3 Stability Condition of the $\underline{M}$ $\underline{\Delta}$-Structure

After the generic block structure in Fig. 3.16(c) is obtained and the lower LFT-model is calculated, MIMO stability conditions can be derived for the system. It is assumed, that the transfer function $\underline{\boldsymbol{N}}$ is nominally stable. The uncertainty transfer function $\underline{\Delta}$ is also assumed to be stable. Considering the Eq. (3.68), it follows from these assumptions, that only the feedback term $(\boldsymbol{I} - \underline{\boldsymbol{N}}_{11}\ \underline{\boldsymbol{\Delta}})^{-1}$ is a source for instability of the upper LFT-model $\underline{\boldsymbol{F}}_u$. As a result, the stability of the upper LFT-model $\underline{\boldsymbol{F}}_u$ is identical to the $\underline{M}$ $\underline{\Delta}$-structure in Fig. 3.16(d) with the notation $\underline{\boldsymbol{N}}_{11} = \underline{\boldsymbol{M}}$.
According to the Small Gain Theorem in Eq. (3.59), the feedback system is stable, if:

$$||\underline{\boldsymbol{M}}\ \underline{\boldsymbol{\Delta}}||_\infty \leq ||\underline{\boldsymbol{M}}||_\infty \cdot ||\underline{\Delta}||_\infty < 1 \tag{3.69}$$

Applying the H_∞-norm leads to the following result:

$$\bar{\sigma}(\underline{\boldsymbol{M}}(j\omega)\ \underline{\boldsymbol{\Delta}}(j\omega)) \leq \bar{\sigma}(\underline{\boldsymbol{M}}(j\omega)) \cdot \bar{\sigma}(\underline{\boldsymbol{\Delta}}(j\omega)) < 1 \tag{3.70}$$

Further, applying the Nyquist criterion in Sec. 3.4.2 to the $\underline{M}$ $\underline{\Delta}$-system, the following Theorem in Ref. [86, p.308] holds:

Theorem: *Assume that the nominal system $\underline{\boldsymbol{M}}(s)$ is stable and that the perturbations $\underline{\boldsymbol{\Delta}}(s)$ are stable. [...] Then the $\underline{M}$ $\underline{\Delta}$-system is stable for all allowed perturbations (we have RS* [i.e. Robust Stability]) *if and only if:*
Nyquist plot of $\det(\underline{\boldsymbol{I}} - \underline{\boldsymbol{M}}(s)\ \underline{\boldsymbol{\Delta}}(s))$ *does not encircle the origin*, $\forall\underline{\boldsymbol{\Delta}}$

$$\Leftrightarrow \det(\underline{\boldsymbol{I}} - \underline{\boldsymbol{M}}(s)\ \underline{\boldsymbol{\Delta}}(s)) \neq 0,\ \ \forall\omega,\ \forall\underline{\boldsymbol{\Delta}} \tag{3.71}$$

$$\Leftrightarrow \lambda_i(\underline{\boldsymbol{M}}\ \underline{\boldsymbol{\Delta}}) \neq 1,\ \forall i,\ \forall\omega,\ \forall\underline{\boldsymbol{\Delta}} \tag{3.72}$$

The proof is also given in Ref. [86, p.309]. The two elaborated theorems will help to understand the derivation and the meaning of the MIMO gain and phase margins in the next section and the introduction of the μ-analysis in Sec. 3.7.

3.5.4 MIMO Gain and Phase Margin[9]

The SISO margins, introduced in Sec. 3.5.1, focus on the gain and phase margins at one channel while the remaining channels are kept closed. Since gain and phase shift due to uncertainties and delays occur in every channel simultaneously, it is beneficial to analyze the system by taking all channels into account. This section elaborates different MIMO gain and phase margins. In contrast to the SISO margins, the MIMO margins indicate a conservative robustness measure for gain, phase and delay shift in every parallel channel at a respective analysis point. This section introduces three specific MIMO margins. Ref. [95] has recently presented a generalized class of these margins. By varying a skew parameter, the generalized margins morph continuously between the presented special cases. However, these generalized margins are not within the scope of this thesis.
Typical analysis points for the MIMO gain and phase margins are located at the input ($\boldsymbol{I}$) or the output ($\boldsymbol{O}$) of the plant model, which are also used for the subsequent derivation. However, different analysis points can be more reasonable in real-life closed-loop systems (see Sec. 3.10.5).
In order to derive the three specific MIMO margins and to understand their meaning, complex and/or real uncertainties are inserted in different ways into typical feedback loops. The following MIMO gain and phase margin derivation is based on the work in Refs. [42, 43]. This thesis supplements additional derivation steps for the understanding of the different MIMO gain and phase margins. This section and the subsequent section, Sec. 3.6, represent *Contribution II.1*, which has been introduced in Sec. 1.2.2.

3.5.4.1 Complementary Sensitivity Function

This section introduces the MIMO gain and phase margin of the complementary sensitivity transfer function $\underline{\boldsymbol{T}}_I(s)$ or $\underline{\boldsymbol{T}}_O(s)$. Fig. 3.17(a) illustrates an exemplary MIMO closed-loop system for the derivation of the MIMO gain and phase margins of the complementary sensititvity transfer function $\underline{\boldsymbol{T}}_I(s)$. In the closed-loop system, additive uncertainties are inserted at each input channel of the plant $\underline{\boldsymbol{G}}$. The introduced uncertainties Δ_i are scalar, complex uncertainties, which are independent from each other. By structural transformation, the presented feedback loop can be brought to Fig. 3.17(b), with the diagonal uncertainty matrix

$$\underline{\boldsymbol{\Delta}} = \mathrm{diag}\{\Delta_1, ..., \Delta_{n_t}\} \tag{3.73}$$

with n_t number of uncertainties (which is for the presented example $n_t = 3$), the open-loop transfer function $\underline{\boldsymbol{L}}_I(s) = \underline{\boldsymbol{K}}(s) \cdot \underline{\boldsymbol{G}}(s)$ and $\underline{\boldsymbol{I}}$ as identity matrix . The structural modification supports the understanding of the MIMO gain and phase margins. If the uncertainties Δ_i are

[9] The author has published this section in Ref. [94]. This section represents a revised and extended version.

zero, the nominal system results. If the terms $1+\Delta_i$ with uncertainty Δ_i are only real numbers, the open-loop transfer function changes in gain. If the uncertainties Δ_i are complex numbers and the term $1+\Delta_i$ has the absolute value 1, the open-loop transfer function changes in phase. These findings are summarized below:

$$\Delta_i = 0 \rightarrow \text{nominal} \tag{3.74}$$

$$k_{T,i} = 1+\Delta_i = 1+r_{T,i}\, e^{j\phi} \in \mathbb{R} \quad \text{i.e. the term is real} \rightarrow \text{change in gain} \tag{3.75}$$

$$k_{T,i} = 1+\Delta_i = 1+r_{T,i}\, e^{j\phi} = e^{j\theta_i} \rightarrow \text{change in phase} \tag{3.76}$$

The variable $r_{T,i}$ is the amplitude $r_{T,i} = |\Delta_i|$ and the variable ϕ_i is the phase of the uncertainty Δ_i with $\Delta_i = r_{T,i}\, e^{j\phi_i}$. The variable θ_i represents in this case the phase, which is added by the term $1+\Delta_i$ to the respective channel of the uncertainty Δ_i, if only phase is added to the ith channel. These variables are visualized in Fig. 3.18. The variable $r_{T,i}$ is visualized as a blue circle with center at 1. Note that this results from the additive uncertainties, which were added to the system. The absolute value 1 is shown by the red, dashed circle. As a result, the scalar variable $k_{T,i}$ represents the factor by which the term $1+\Delta_i$ changes the gain and/or phase at the ith channel.

Fig. 3.17(b) shows the structural transformation from Fig. 3.17(a). The terms $1+\Delta_i$ are written in matrix notation. They represent the factors $k_{T,i}$ in each channel, with which the open-loop transfer function is multiplied. If the discussed closed-loop system is restructured by a LFT, as shown in Sec. 3.5.3.2, the structure in Fig. 3.17(c) results. Only the relevant part of the LFT-model, which affects the stability, is considered on the right side of 3.17(c). For this case, the matrix $\underline{\boldsymbol{M}}$ in Fig. 3.16(d) is equivalent to the matrix $\underline{\boldsymbol{T}}_I$, which is identical to the complementary sensitivity transfer function $\underline{\boldsymbol{T}}_I$, introduced in Sec. 3.2.2. As in Ref. [42, p.47], this surprising result is due to the fact, that the transfer function "seen" by the uncertainty $\underline{\boldsymbol{\Delta}}$ from $\boldsymbol{du}_T$ to $\boldsymbol{u}$ is equivalent to the complementary transfer function in Fig. 3.4. The derivation of the LFT-model in Fig. 3.17(c) is demonstrated in the Appendix A.1.1.

Considering the Small Gain Theorem in Sec. 3.5.3.1, the system in Fig. 3.17 is robustly stable if:

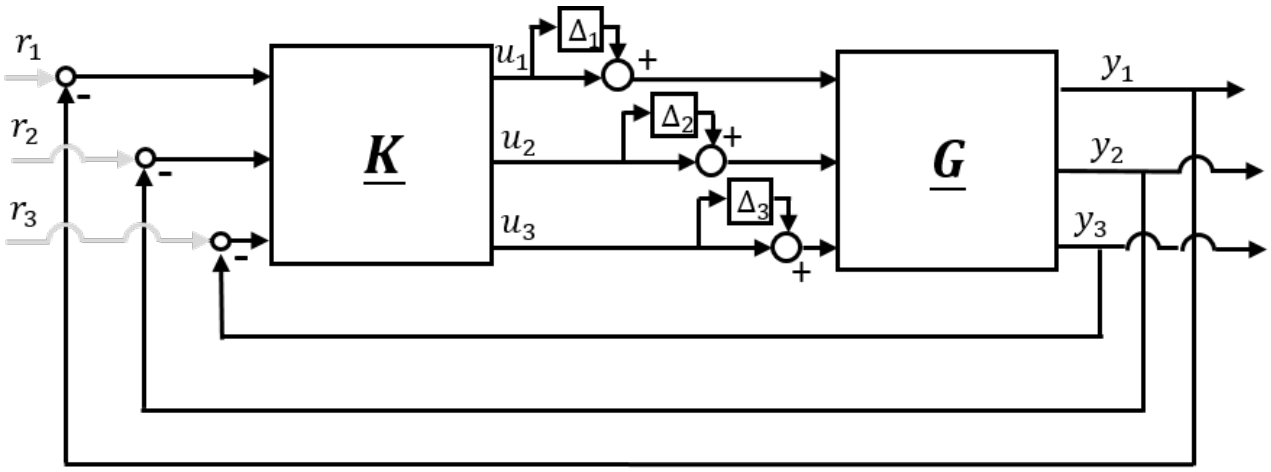
$$||\underline{\boldsymbol{T}}_I(s)\ \underline{\boldsymbol{\Delta}}(s)||_\infty \leq ||\underline{\boldsymbol{T}}_I(s)||_\infty \cdot ||\underline{\boldsymbol{\Delta}}(s)||_\infty < 1 \tag{3.77}$$

This equation can be rewritten:

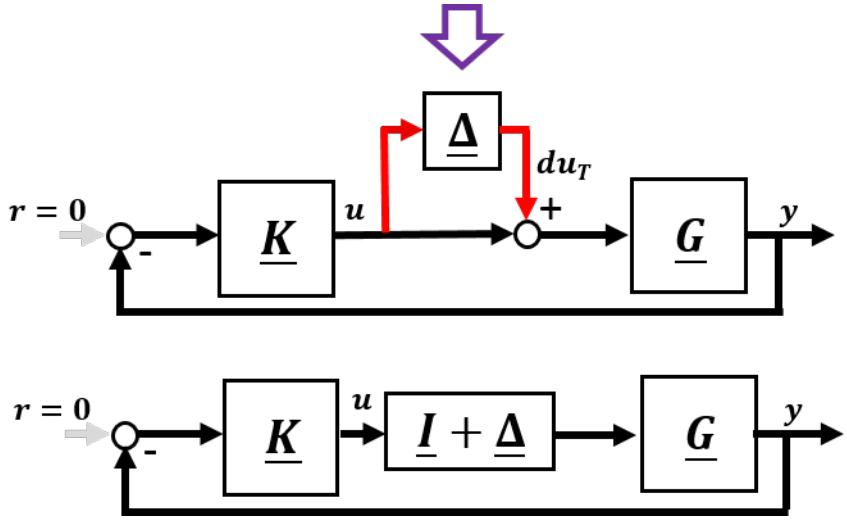
$$\bar{\sigma}(\underline{\boldsymbol{\Delta}}(s)) < r_T(\underline{\boldsymbol{T}}_I(s)) \quad \text{for all} \ \ s = j\omega \tag{3.78}$$

$$\bar{\sigma}(\Delta_i(s)) < r_T(\underline{\boldsymbol{T}}_I(s)) \quad \text{for all} \ \ s = j\omega, \text{and} \ \ i \tag{3.79}$$

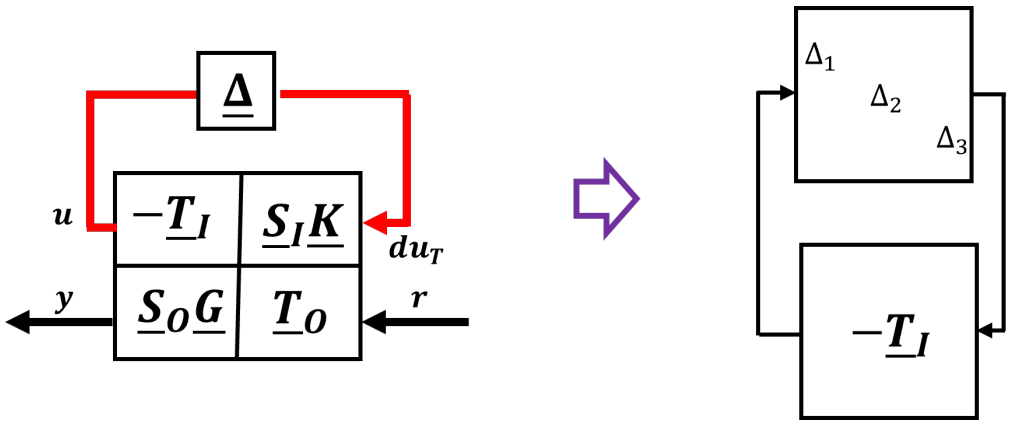
with the notation $1/\bar{\sigma}(\underline{\boldsymbol{T}}_I) = r_T(\underline{\boldsymbol{T}}_I)$. The maximum singular values of $\underline{\boldsymbol{T}}_I(s)$ and $\underline{\boldsymbol{\Delta}}(s)$ are denoted as $\bar{\sigma}(\underline{\boldsymbol{T}})$ and $\bar{\sigma}(\underline{\boldsymbol{\Delta}}(s))$. As a result, MIMO gain and phase margins can be derived.



(a) Scalar additive uncertainties at the inputs



(b) Structural transformation



(c) Linear Fractional Transformation for the application of the Small Gain Theorem

Figure 3.17: MIMO gain and phase margin derivation of the complementary sensitivity function $\underline{T}_I$

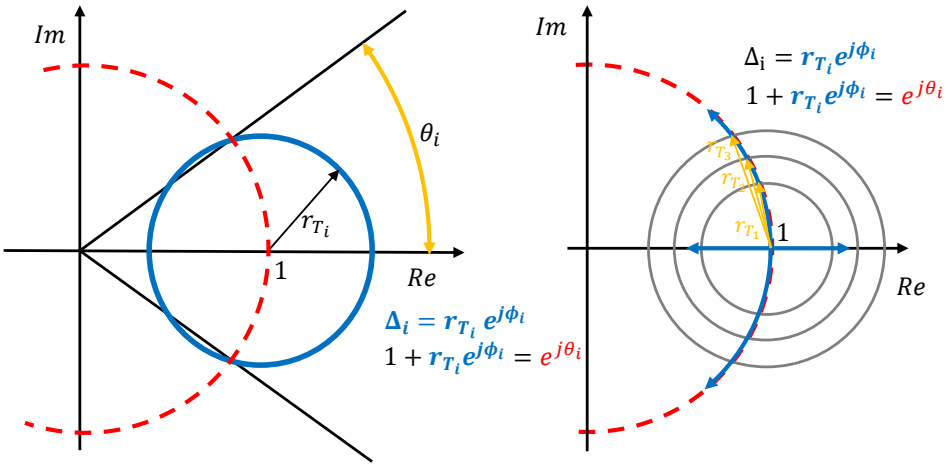


Figure 3.18: Illustration of the term $1+\Delta_i$, if it changes in gain and/or phase; the dashed red circle represents the term $e^{j\theta_i}$, the blue circle illustrates the term $1+\Delta_i$, the angle θ_i shows the phase, which is added to the system, if the uncertainty causes phase only, the grey circles on the right show different-sized uncertainties, depending on $r_{T,i}$

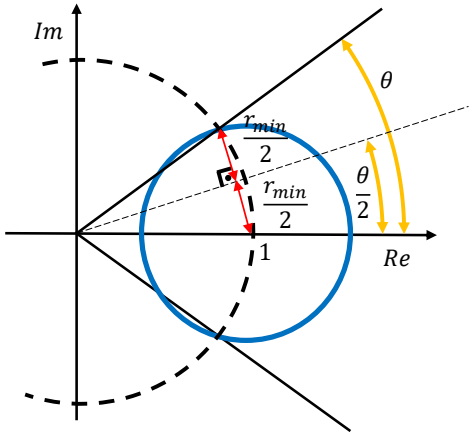


Figure 3.19: Geometrical interpretation for calculating the MIMO phase margin

For all real Δ_i, the system remains robustly stable if all uncertainties Δ_i stay inside the circle with radius $r_{T,min}$, centered at 1. Thus, the MIMO gain margin of the complementary sensitivity transfer function are:

$$\begin{aligned} \text{MIMO gain margin} &= [1 - r_{T,min}, 1 + r_{T,min}], \\ \text{where } r_{T,min} &= \inf_{\omega \in \mathbb{R}} r_T(\omega) \end{aligned} \tag{3.80}$$

Since the minimal possible maximum singular value is $\bar{\sigma}(\underline{\boldsymbol{T}}) = 1$, the maximal interval of the MIMO gain margin is $[0,2]$ or on a logscale $[-\infty,6\text{ dB}]$. The lower and upper MIMO gain margins of the input complementary sensitivity transfer function represent thus lowest and highest factors $k_{T,min} = 1 + r_{T,min}$ the open-loop transfer function $\underline{\boldsymbol{L}}_I$ can be multiplied with in

all input channels such that the closed-loop system maintains stability definitely.
The results can be seen in Fig. 3.18(a). Since for the MIMO gain margin the uncertainty changes only in gain, the interval is defined by the intersection points between the blue circle and the real axis, in case $r_{T,i} = r_{T,min}$.
In Ref. [42], the MIMO gain margin is defined as follows:

Definition: *A MIMO gain margin is a real interval* $[G_L, G_U]$ *such that the closed-loop system remains stable for all real* Δ_i *satisfying* $G_L \leq 1 + \Delta_i \leq G_U$. $[...]$ *Note that unlike the SISO gain margin, the MIMO margin is not unique. The extremal value of* G_U *depends on* G_L *and vice versa.* $[...]$.

Equivalently, the MIMO phase margin of the complementary sensitivity transfer function is derived. Assuming the terms $1 + \Delta_i$ lead only to phase rotation, the calculation then follows from geometrical interpretation of Fig. 3.19:

$$\text{MIMO Phase Margin} = [-\theta_T, \theta_T] \quad \text{with} \quad \theta = 2 \ \text{asin}\left(\frac{r_{T,min}}{2}\right) \tag{3.81}$$

The angle θ_T results from the intersection point between the circle with radius 1 centered at the origin and the circle with radius $r_{T,min}$ centered at 1. The two intersection points are relevant from the fact that the phase margin considers only phase rotation. At the intersection points the gain of $1 + \Delta_i$ is equal to 1. The MIMO phase margin can also be calculated analytically, but the geometrical interpretation is easier to derive. The algebraic derivation is provided in the Appendix A.1.1.2.
On the right side of Fig. 3.18 it is visualized how the phase margin changes varying $\bar{\sigma}(\underline{\boldsymbol{T}})$. The maximal interval of the phase margin is [-60 deg, 60 deg], if the case $\bar{\sigma}(\underline{\boldsymbol{T}}) = 1$ occurs. In contrast to the MIMO gain margin above, the MIMO phase margin is symmetric.
In consequence, the MIMO phase margin specifies the phase, which can be added by all terms $1 + \Delta_i = e^{j\theta_T}$ to the system, such that it remains stable.
Ref. [42] defines the MIMO phase margin for the complementary sensitivity transfer function as follows:

Definition: *The MIMO phase margin [...] is the largest real interval* $[-\theta, \theta]$ *such that the system remains stable for all simultaneous, independent phase rotations* $1 + \Delta_i = e^{j\theta_i}$ *satisfying* $-\theta \leq \theta_i \leq \theta$. *Defined in this way, the MIMO phase margin is unique because it is a symmetrical interval [...].*

The additive uncertainties can be also introduced at the output channels of the plant. The MIMO gain and phase margins are calculated equivalently as explained above by utilization of the transfer function $\underline{\boldsymbol{T}}_O(s)$, instead of $\underline{\boldsymbol{T}}_I(s)$.

3.5.4.2 Sensitivity Function

Similarly, the MIMO gain and phase margins of the sensitivity transfer function $\underline{\boldsymbol{S}}_I(s)$ or $\underline{\boldsymbol{S}}_O(s)$ are derived in this section. For the derivation, no additive uncertainties are inserted, but so-called inverse uncertainties are introduced into the closed-loop system. The present section derives the MIMO margins for $\underline{\boldsymbol{S}}_I(s)$, but the approach is analogous for the sensitivity function $\underline{\boldsymbol{S}}_O(s)$. The respective structure for $\underline{\boldsymbol{S}}_I(s)$ is visualized in Fig. 3.20(a), which shows three inverse uncertainties at the input channels. For the derivation, the channels are combined to one multidimensional signal and the open-loop transfer function is written as $\underline{\boldsymbol{L}}_I(s)$. For an easier MIMO gain and phase margin interpretation, the inverse uncertainties Δ_i are unified to $\underline{\boldsymbol{\Delta}}$ and replaced by $\underline{\boldsymbol{\Delta}} = \underline{\boldsymbol{I}} - \underline{\boldsymbol{C}}_S^{-1}$. $\underline{\boldsymbol{C}}_S$ is a multidimensional transfer function, which fulfills the equation $\underline{\boldsymbol{\Delta}} = \underline{\boldsymbol{I}} - \underline{\boldsymbol{C}}_S^{-1}$. The resulting structure is illustrated in Fig. 3.20(b). The feedback loop with the feedback $\underline{\boldsymbol{I}} - \underline{\boldsymbol{C}}_S^{-1}$ can be compressed to:

$$(\underline{\boldsymbol{I}} - \underline{\boldsymbol{\Delta}}(s))^{-1} = (\underline{\boldsymbol{I}} - (\underline{\boldsymbol{I}} - \underline{\boldsymbol{C}}_S^{-1}(s)))^{-1} = \underline{\boldsymbol{C}}_S(s) \tag{3.82}$$

In elementwise notation for the ith uncertainty, the following can be stated for the ith channel:

$$\Delta_i = 0 \rightarrow \text{nominal case}$$

$$c_{S,i} = \frac{1}{1-\Delta_i} = \frac{1}{1 - r_{S,i} e^{j\phi_{S,i}}} \in \mathbb{R} \rightarrow \text{change in gain}$$

$$c_{S,i} = \frac{1}{1-\Delta_i} = \frac{1}{1 - r_{S,i} e^{j\phi_{S,i}}} = e^{j\theta_{S,i}} \rightarrow \text{change in phase}$$

Note that $\underline{\boldsymbol{C}}_S(s) = \text{diag}\{c_{S,1}, ..., c_{S,n_s}\}$ with n_s, which is the number of input channels of the transfer function $\underline{\boldsymbol{G}}(s)$ and thus also the number of inverse uncertainties (see Fig. 3.20(a)). The variable $c_{S,i}$ can be interpreted as the factor the term $1/(1-\Delta_i)$ changes the gain of the ith channel, if the term $1/(1-\Delta_i)$ stays real. Analogously to Sec. 3.5.4.1, if the term $1/(1-\Delta_i)$ has the absolute value 1, the variable $c_{S,i}$ adds only the phase $\theta_{S,i}$ to the system. Note that, $\phi_{S,i}$ represents the phase of the uncertainty $\Delta_i = r_{S,i} e^{j\phi_{S,i}}$ with the amplitude $|\Delta_i| = r_{S,i}$.

By the mathematical property in Eq. (3.82), the structure at the bottom of Fig. 3.20(b) results. In order to utilize the Small Gain Theorem for the stability analysis, a LFT is applied (see Fig. 3.20(c)). Considering the introduced structure in this section, the relevant transfer function $\underline{\boldsymbol{M}}(s)$ for stability analysis from the LFT is equivalent to the sensitivity transfer function $\underline{\boldsymbol{S}}_I(s)$, which is introduced in Sec. 3.2.2. The reason is analogous to the MIMO gain and phase margin for the complementary sensitivity transfer function in Sec. 3.5.4.1. The transfer function $\underline{\boldsymbol{\Delta}}$ "sees" the sensitivity transfer function $\underline{\boldsymbol{S}}_I(s)$, which is the transfer function from $\boldsymbol{du}_S$ to $\boldsymbol{u}$ in Fig. 3.3.

The applied Small Gain Theorem results in the following condition:

$$||\underline{\boldsymbol{S}}_I(s)\ \underline{\boldsymbol{\Delta}}(s)||_\infty \leq ||\underline{\boldsymbol{S}}_I(s)||_\infty \cdot ||\underline{\boldsymbol{\Delta}}(s)||_\infty < 1 \tag{3.83}$$

$$\bar{\sigma}(\Delta_i(s)) < r_S(\underline{\boldsymbol{S}}_I(s)) \quad \text{for all} \ \ s = j\omega, \text{and} \quad i \tag{3.84}$$

with the notation $1/\bar{\sigma}(\underline{\boldsymbol{S}}_I) = r_S(\underline{\boldsymbol{S}}_I)$. Since the embedding of uncertainties into the structure is more complicated as in the case of the complementary sensitivity transfer function, the MIMO gain and phase margins cannot easily depicted as in Fig. 3.18. But using the structural transformation and the property shown in Eq. (3.82), the MIMO gain and phase margins can still be derived. Considering the transfer function $\underline{\boldsymbol{C}}(s)$ and the Small Gain Theorem, the interval of the MIMO gain margin for the sensitivity transfer function is:

$$\begin{aligned} \text{MIMO gain margin} &= \left[\frac{1}{1+r_{S,min}}, \frac{1}{1-r_{S,min}}\right], \\ \text{where } r_{S,min} &= \inf_{\omega\in\mathbb{R}} r_S(\omega). \end{aligned} \tag{3.85}$$

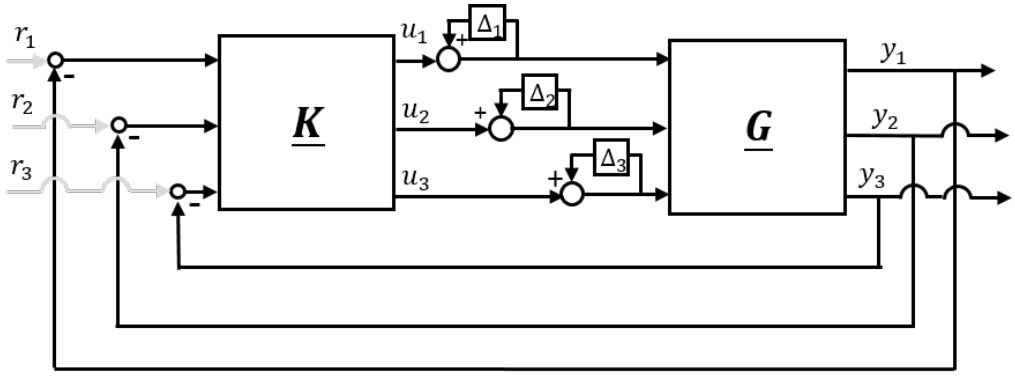
with the notation $1/\bar{\sigma}(\underline{\boldsymbol{S}}_I) = r_{min}(\underline{\boldsymbol{S}}_{\boldsymbol{I}})$. For all real Δ_i, the system remains stable if the variable $c_{S,i}$ corresponding to the uncertainty Δ_i stays within the given interval. The maximal interval of the MIMO gain margin occurs if $\bar{\sigma}(\underline{\boldsymbol{S}}_I) = 1$. This results in the interval $[0.5, \infty]$ or on the logscale [-6 dB,∞]. As a result, the interval is not symmetric.
If the uncertainties Δ_i lead only to phase rotation, the following equation holds:

$$\frac{1}{1-\Delta_i} = \frac{1}{1-r_{S,i}\ e^{j\phi_{S,i}}} = e^{j\theta_{S,i}} \tag{3.86}$$

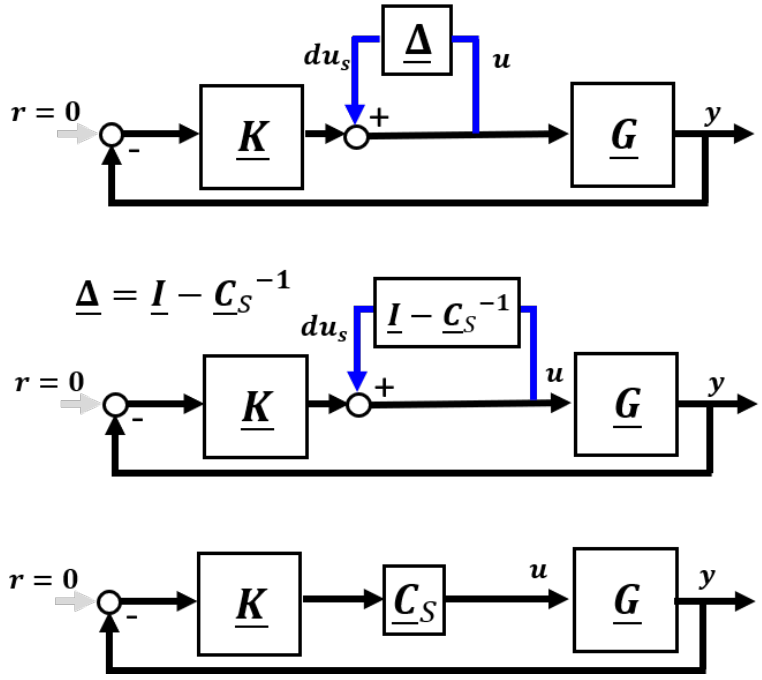
The equation is equal to $e^{j\theta_{S,i}}$, since the MIMO phase margin shall give an interval for the term $c_{S,i}$, which affect only the phase of the system without changing the gain. From Eq. (3.86), equivalently to the MIMO phase margin determination for the complementary sensitivity transfer function, the MIMO phase margin of the sensitvity transfer function is calculated:

$$\text{MIMO Phase Margin} = [-\theta_S, \theta_S] \quad \text{with} \quad \theta_S = 2\ \text{asin}\left(\frac{r_{S,min}}{2}\right) \tag{3.87}$$

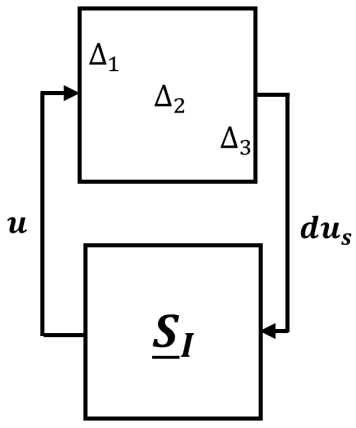
with $1/\bar{\sigma}(\underline{\boldsymbol{S}}_I) = r_{S,min}(\underline{\boldsymbol{S}}_{\boldsymbol{I}})$. As it is seen from the equation, the calculation for the MIMO phase margin of the sensitivity transfer function is analogous to the computation of the MIMO phase margin for the complementary sensitivity transfer function and is also symmetric. The phase margin cannot be derived equivalently by geometrical interpretation. However, it can be calculated by solving Eq. (3.86) for $\theta_{S,i}$ using $r_{S,min}$. The derivation is provided in the Appendix A.1.2.



(a) Scalar inverse uncertainties at the inputs



(b) Structural transformation



(c) LFT for the application of the Small Gain Theorem

Figure 3.20: MIMO gain and phase margin of the sensitivity function $\underline{S_I}$

3.5.4.3 Balanced Sensitivity Function

In addition to the last two MIMO margins, Refs. [34, 43] introduced a new robustness measure. The new MIMO gain and phase margin is based on the so-called balanced sensitivity function. The basic idea of the balanced sensitivity function is to combine the robustness measure of the complementary sensitivity transfer function and the sensitivity transfer function. The margins for the balanced sensitiviy transfer function can be derived equivalently for the input channels as well as for the output channels. In the following, the margins at the input channels are calculated.
Fig. 3.21(a) shows the closed-loop structure for the balanced sensitivity transfer function with the corresponding uncertainties, which consist of an additive and an inverse part at the plant input. The closed-loop system can be restructured as drawn in Fig. 3.21(b). A LFT leads to Fig. 3.21(c), which is essential for the stability analysis. The resulting and relevant transfer function $\underline{\boldsymbol{M}}(s)$ of the generic LFT model for the stability analysis is equivalent to the subtraction of the sensitivity transfer function $\underline{\boldsymbol{S}}_I$ by the complementary transfer function $\underline{\boldsymbol{T}}_I$:

$$\underline{\boldsymbol{S}}_I(s) - \underline{\boldsymbol{T}}_I(s) = (\underline{\boldsymbol{I}} + \underline{\boldsymbol{L}}_I(s))^{-1} - \underline{\boldsymbol{L}}_I(s)(\underline{\boldsymbol{I}} + \underline{\boldsymbol{L}}(s))_I^{-1} \tag{3.88}$$

This equation can be also derived by the utilization of the results in Sec. 3.5.4.1 and Sec. 3.5.4.2. Fig. 3.21(b) at the bottom illustrates a consolidated transfer function of Fig. 3.21(a). As the structure is compared to Figs. 3.3 and 3.4, the transfer function seen by the uncertainty Δ_i is equivalent to the subtraction of both sensitivity transfer functions:

$$\boldsymbol{u}_T = \underline{\boldsymbol{T}}_I(s)\ \boldsymbol{du}_T \tag{3.89}$$

$$\boldsymbol{u}_S = \underline{\boldsymbol{S}}_I(s)\ \boldsymbol{du}_S \tag{3.90}$$

$$\boldsymbol{u}_S - \boldsymbol{u}_T = \Big(\underline{\boldsymbol{S}}_I(s) - \underline{\boldsymbol{T}}_I(s)\Big)\boldsymbol{du}_{ST} \tag{3.91}$$

The condition for stability results from the Small Gain Theorem:

$$\bar{\sigma}(\Delta_i(s)) < r_{ST}(\underline{\boldsymbol{S}}_I(s) - \underline{\boldsymbol{T}}_I(s)) \quad \text{for all} \quad s = j\omega, \text{and} \quad i \tag{3.92}$$

with the notation $1/\bar{\sigma}(\underline{\boldsymbol{S}}_I(s) - \underline{\boldsymbol{T}}_I(s)) = r_{ST}(\underline{\boldsymbol{S}}_I(s) - \underline{\boldsymbol{T}}_I(s))$. Fig. 3.21(b) at the bottom illustrates, how the inserted uncertainty affects the gain and phase of the closed-loop system. The influence is expressed by the following equation:

$$\underline{\boldsymbol{C}}_{ST}(s) = \big(\underline{\boldsymbol{I}} + \underline{\boldsymbol{\Delta}}(s)\big)\big(\underline{\boldsymbol{I}} - \underline{\boldsymbol{\Delta}}(s)\big)^{-1} \tag{3.93}$$

Derived from this equation, the complex gain of the uncertainty Δ_i shifts the gain and phase of the closed-loop system in the following way:

$$\frac{1-\Delta_i}{1+\Delta_i} = \frac{1-r_{ST,i}\ e^{j\phi}}{1+r_{ST,i}\ e^{j\phi}} \quad \text{with } \Delta_i = r_{ST,i}\ e^{j\phi} \tag{3.94}$$

If only the gain changes, and considering the stability results above, the MIMO gain margin of the balanced sensitivity transfer function is calculated as follows:

$$\begin{aligned} \text{MIMO gain margin} &= \left[\frac{1-r_{ST,min}}{1+r_{ST,min}}, \frac{1+r_{ST,min}}{1-r_{ST,min}}\right], \\ \text{where } r_{ST,min} &= \inf_{\omega\in\mathbb{R}} r_{ST}(\omega). \end{aligned} \tag{3.95}$$

By respecting the following two necessary equations

$$\frac{1+r_{ST,min}\ e^{j\phi}}{1-r_{ST,min}\ e^{j\phi}} = e^{j\theta} \tag{3.96}$$

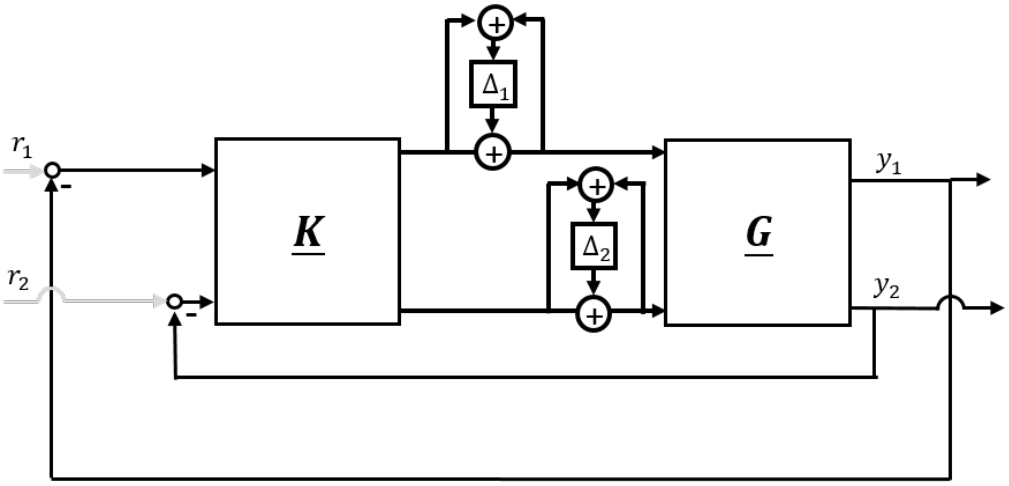
and

$$\left|\left|\frac{1+r_{ST,min}\ e^{j\phi}}{1-r_{ST,min}\ e^{j\phi}}\right|\right|_2 = 1, \tag{3.97}$$

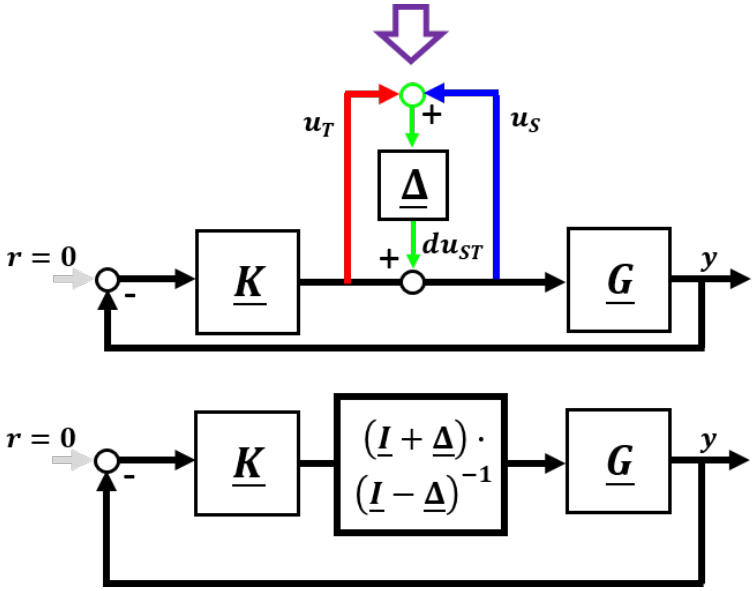
the MIMO phase margin of the balanced sensitivity function is obtained:

$$\begin{aligned} \text{MIMO phase margin} &= \left[-\theta_{ST}, \theta_{ST}\right] \\ \text{with } \theta_{ST} &= 2\,\text{atan}\left(r_{ST,min}\right) \end{aligned} \tag{3.98}$$

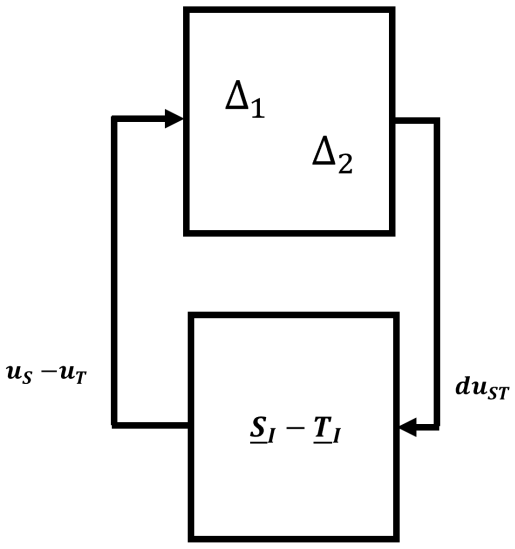
The derivation of the MIMO phase margin for the balanced sensitivity transfer function is provided in the Appendix A.1.3. Inserting the lowest possible singular value $\bar{\sigma}(\underline{\boldsymbol{S}}_I(s) - \underline{\boldsymbol{T}}_I(s)) = 1$ into the margin intervals, the largest possible intervals are derived. The MIMO gain margin can have the interval range $[0, \infty]$ or on a logscale $[-\infty$ dB,∞ dB$]$. The MIMO phase margin reaches the interval range [-90 deg, 90 deg]. In contrast to the MIMO gain margins of the sensitivity transfer function and the complementary sensitivity transfer functions, the MIMO gain margins of the balanced sensitivity transfer function are symmetric in dB. According to Ref. [34, p.132], *"replacing $\underline{\boldsymbol{S}}$ and $\underline{\boldsymbol{T}}$ by $\underline{\boldsymbol{S}}_I - \underline{\boldsymbol{T}}_I$ alone increases the guaranteed margins further to make them more useful in engineering"*.



(a) Combined additive and inverse uncertainties at plant input



(b) Structural transformation



(c) LFT for the application of the Small Gain Theorem

Figure 3.21: MIMO gain and phase margin derivation of the balanced sensitivity function

3.6 Reduced Conservatism Evaluation of MIMO Margins[10]

In Sec. 3.5.4 it was shown that the balanced sensitivity $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}$ has the potential to yield larger margins than $\underline{\boldsymbol{S}}$ or $\underline{\boldsymbol{T}}$ alone. For instance, the phase margin using $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}$ may be as large as 90 deg, whereas for $\underline{\boldsymbol{S}}$ or $\underline{\boldsymbol{T}}$ there is a theoretical limit of 60 deg. However, this alone does not prove that the actual phase margin obtained using $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}$ will surpass the ones obtained using $\underline{\boldsymbol{S}}$ or $\underline{\boldsymbol{T}}$. In the following, we present mathematical proofs that the balanced MIMO margin is always in-between the gain margins obtained from $\underline{\boldsymbol{S}}$ or $\underline{\boldsymbol{T}}$ and that the phase margin using $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}$ is greater or equal than the ones obtained using $\underline{\boldsymbol{S}}$ or $\underline{\boldsymbol{T}}$ for a certain domain of maximum singular values of the balanced and (complementary) sensitivity transfer functions. The presented proof has been published by the author in Ref. [94]. The underline and the bold script for matrices is omitted within the proof for readability.

3.6.1 Mathematical Preliminaries

Let $A, B \in \mathbb{C}^{l \times m}$ be two $l \times m$ complex matrices. Using Fan's-Theorem in Ref. [96] the following inequalities can be established:

$$|\bar{\sigma}(A) - \bar{\sigma}(B)| \leq \bar{\sigma}(A+B) \leq \bar{\sigma}(A) + \bar{\sigma}(B). \tag{3.99}$$

Considering $S + T = I$, the application of Fan's-Theorem to the sensitivity transfer functions leads to the following inequalities:

$$|\bar{\sigma}(S) - \bar{\sigma}(T)| \leq \bar{\sigma}(S+T) = 1 \leq \bar{\sigma}(S) + \bar{\sigma}(T) \tag{3.100}$$

$$|\bar{\sigma}(S) - \bar{\sigma}(T)| \leq \bar{\sigma}(S-T) \leq \bar{\sigma}(S) + \bar{\sigma}(T) \tag{3.101}$$

With $T = I - S$ and $S = I - T$, the subsequent equations result:

$$|2\bar{\sigma}(S) - 1| \leq \bar{\sigma}(S-T) = \bar{\sigma}(2S-I) \leq 2\bar{\sigma}(S) + 1 \tag{3.102}$$

$$|2\bar{\sigma}(T) - 1| \leq \bar{\sigma}(S-T) = \bar{\sigma}(I-2T) \leq 2\bar{\sigma}(T) + 1 \tag{3.103}$$

It should be noted that $S = S(s)$, $T = T(s)$, $S + T = S(s) + T(s)$ and $S - T = S(s) - T(s)$ are transfer functions and depend on $s = j\omega$. In consequence, the above inequalities hold at any frequency ω. This is visible in Fig. 3.22(a), which shows exemplary maximum singular value plots of a closed-loop system with plant and proportional-integral (PI) controller. The maximum singular value peaks occur at different frequencies. The maximum of the greatest singular value $\bar{\sigma}(S)$ is at the frequency $\omega_S = \arg\max_\omega \bar{\sigma}(S)$, whereas the maximum of the

[10] The author has published the following three subsections in Ref. [94] and revised within this thesis

greatest singular value $\bar{\sigma}(T)$ is at the frequency $\omega_T = \arg\max_\omega \bar{\sigma}(T)$ and the maximum of the greatest singular value $\bar{\sigma}(S-T)$ occurs at the frequency $\omega_{S-T} = \arg\max_\omega \bar{\sigma}(S-T)$. In the following, the notation $\underset{\omega_\alpha}{\bar{\sigma}}(\beta)$ is used to represent the value of $\bar{\sigma}(\beta(j\omega_\alpha))$.

3.6.2 Lower Gain Margins

In this subsection we establish the relation between the lower gain margins using $S-T$ with the ones obtained with S and T.

3.6.2.1 Sensitivity Transfer Function

It is trivial to see from the definition of ω_S and ω_{S-T} that

$$2\underset{\omega_S}{\bar{\sigma}}(S)+1 \geq 2\underset{\omega_{S-T}}{\bar{\sigma}}(S)+1. \tag{3.104}$$

On the other hand, by Eq. (3.102) at ω_{S-T} and Eq. (3.104)

$$2\underset{\omega_S}{\bar{\sigma}}(S)+1 \geq 2\underset{\omega_{S-T}}{\bar{\sigma}}(S)+1 \geq \underset{\omega_{S-T}}{\bar{\sigma}}(S-T). \tag{3.105}$$

We rearrange Eq. (3.105) as follows

$$\begin{aligned}
&2\underset{\omega_S}{\bar{\sigma}}(S)+1 \geq \underset{\omega_{S-T}}{\bar{\sigma}}(S-T)\\
\Longrightarrow\ &\underset{\omega_S}{\bar{\sigma}}(S) \geq \underset{\omega_{S-T}}{\bar{\sigma}}(S-T) - \underset{\omega_S}{\bar{\sigma}}(S) - 1\\
\Longrightarrow\ &\underset{\omega_S}{\bar{\sigma}}(S) + \underset{\omega_{S-T}}{\bar{\sigma}}(S-T)\underset{\omega_S}{\bar{\sigma}}(S)\\
&\quad \geq \underset{\omega_{S-T}}{\bar{\sigma}}(S-T) + \underset{\omega_{S-T}}{\bar{\sigma}}(S-T)\underset{\omega_S}{\bar{\sigma}}(S) - \underset{\omega_S}{\bar{\sigma}}(S) - 1\\
\Longrightarrow\ &\underset{\omega_S}{\bar{\sigma}}(S)\left[\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)+1\right]\\
&\quad \geq \underset{\omega_{S-T}}{\bar{\sigma}}(S-T) - 1 + \underset{\omega_S}{\bar{\sigma}}(S)\left[\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)-1\right]
\end{aligned} \tag{3.106}$$

$$\begin{aligned}
\Longrightarrow\ &\underset{\omega_S}{\bar{\sigma}}(S)\left[\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)+1\right] \geq \left[1+\underset{\omega_S}{\bar{\sigma}}(S)\right]\left[\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)-1\right]\\
\Longrightarrow\ &\frac{\underset{\omega_S}{\bar{\sigma}}(S)}{1+\underset{\omega_S}{\bar{\sigma}}(S)} \geq \frac{\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)-1}{\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)+1}\\
\Longrightarrow\ &\frac{1}{1+r_{S,min}(S)} \geq \frac{1-r_{ST,min}(S-T)}{1+r_{ST,min}(S-T)}.
\end{aligned} \tag{3.107}$$

The last inequality in Eq. (3.107) proves that the lower gain margin using the balanced sensitivity $S-T$ is always equal or lower than the one using S.

3.6.2.2 Complementary Sensitivity Transfer Function

Let us assume for the sake of contradiction that the lower gain margin of T is greater than that of $S-T$, i.e. from Eq. (3.80) and Eq. (3.95):

$$\begin{aligned} &\frac{\underset{\omega_T}{\bar{\sigma}}(T)-1}{\underset{\omega_T}{\bar{\sigma}}(T)} > \frac{\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)-1}{\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)+1} \\ \Longrightarrow &\Big(\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)+1\Big)\Big(\underset{\omega_T}{\bar{\sigma}}(T)-1\Big) > \Big(\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)-1\Big)\underset{\omega_T}{\bar{\sigma}}(T) \\ \Longrightarrow &\underset{\omega_{S-T}}{\bar{\sigma}}(S-T) < 2\underset{\omega_T}{\bar{\sigma}}(T)-1. \end{aligned} \tag{3.108}$$

This is a contradiction to the inequality in Eq. (3.103) from Fan's-Theorem, since:

$$\underset{\omega_{S-T}}{\bar{\sigma}}(S-T) \geq \underset{\omega_T}{\bar{\sigma}}(S-T) \geq 2\underset{\omega_T}{\bar{\sigma}}(T)-1 \tag{3.109}$$

As a result, it can be guaranteed that the lower boundary MIMO gain margin of T is always equal or smaller than $S-T$ via proof by contradiction.

3.6.3 Upper Gain Margins

3.6.3.1 Sensitivity Transfer Function

Further, it is proven that the upper MIMO gain margin of S is greater or equal than the upper MIMO gain margin of $S-T$. Let us assume for the sake of contradiction that the upper gain margin of S is less than that of $S-T$, i.e. from Eq. (3.85) and Eq. (3.95):

$$\begin{aligned} &\frac{\underset{\omega_S}{\bar{\sigma}}(S)}{\underset{\omega_S}{\bar{\sigma}}(S)-1} < \frac{\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)+1}{\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)-1} \\ \Longrightarrow &\Big(\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)-1\Big)\underset{\omega_S}{\bar{\sigma}}(S) < \Big(\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)+1\Big)\Big(\underset{\omega_S}{\bar{\sigma}}(S)-1\Big) \\ \Longrightarrow &2\underset{\omega_S}{\bar{\sigma}}(S)-1 > \underset{\omega_{S-T}}{\bar{\sigma}}(S-T) \end{aligned} \tag{3.110}$$

This is a contradiction to the inequality in Eq. (3.102) from Fan's-Theorem, since:

$$\underset{\omega_{S-T}}{\bar{\sigma}}(S-T) \geq \underset{\omega_S}{\bar{\sigma}}(S-T) \geq 2\underset{\omega_S}{\bar{\sigma}}(S)-1 \tag{3.111}$$

Thus, it can be guaranteed that the upper boundary MIMO gain margin of S is equal or always greater than $S-T$.

3.6.3.2 Complementary Sensitivity Transfer Function

It is trivial to see from the definition of ω_T and ω_{S-T} that

$$\underset{\omega_T}{2\bar{\sigma}(T)}+1 \geq \underset{\omega_{S-T}}{2\bar{\sigma}(T)}+1. \tag{3.112}$$

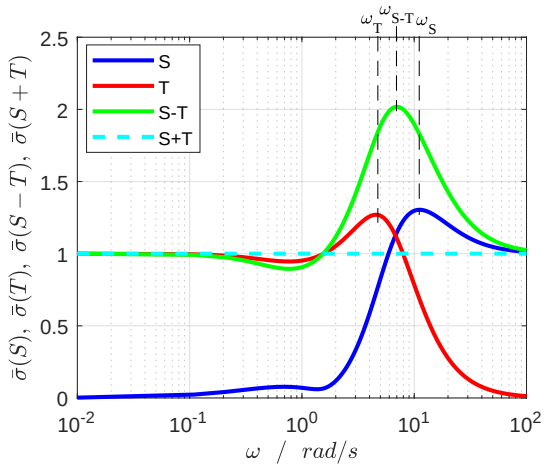
On the other hand, by Eq. (3.103) at ω_{S-T} and Eq. (3.112)

$$\underset{\omega_T}{2\bar{\sigma}(T)}+1 \geq \underset{\omega_{S-T}}{2\bar{\sigma}(T)}+1 \geq \underset{\omega_{S-T}}{\bar{\sigma}(S-T)}. \tag{3.113}$$

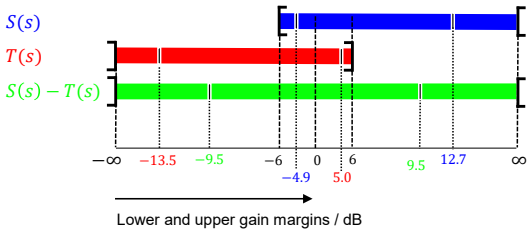
We rearrange Eq. (3.113) as follows:

$$\begin{aligned}
&\underset{\omega_T}{2\bar{\sigma}(T)}+1 \geq \underset{\omega_{S-T}}{\bar{\sigma}(S-T)} \\
\Longrightarrow\ &\underset{\omega_T}{\bar{\sigma}(T)} \geq \underset{\omega_{S-T}}{\bar{\sigma}(S-T)} - \underset{\omega_T}{\bar{\sigma}(T)} - 1 \\
\Longrightarrow\ &\underset{\omega_T}{\bar{\sigma}(T)} + \underset{\omega_{S-T}}{\bar{\sigma}(S-T)}\underset{\omega_T}{\bar{\sigma}(T)} \\
&\geq \underset{\omega_{S-T}}{\bar{\sigma}(S-T)} + \underset{\omega_{S-T}}{\bar{\sigma}(S-T)}\underset{\omega_T}{\bar{\sigma}(T)} - \underset{\omega_T}{\bar{\sigma}(T)} - 1 \\
\Longrightarrow\ &\underset{\omega_T}{\bar{\sigma}(T)} \left[\underset{\omega_{S-T}}{\bar{\sigma}(S-T)} + 1 \right] \\
&\geq \underset{\omega_{S-T}}{\bar{\sigma}(S-T)} - 1 + \underset{\omega_T}{\bar{\sigma}(T)} \left[\underset{\omega_{S-T}}{\bar{\sigma}(S-T)} - 1 \right] \\
\Longrightarrow\ &\underset{\omega_T}{\bar{\sigma}(T)} \left[\underset{\omega_{S-T}}{\bar{\sigma}(S-T)} + 1 \right] \geq \left[1 + \underset{\omega_T}{\bar{\sigma}(T)} \right] \left[\underset{\omega_{S-T}}{\bar{\sigma}(S-T)} - 1 \right] \\
\Longrightarrow\ &\frac{1+\underset{\omega_T}{\bar{\sigma}(T)}}{\underset{\omega_T}{\bar{\sigma}(T)}} \leq \frac{\underset{\omega_{S-T}}{\bar{\sigma}(S-T)}+1}{\underset{\omega_{S-T}}{\bar{\sigma}(S-T)}-1} \\
\Longrightarrow\ &1 + r_{min}(T) \leq \frac{1+r_{min}(S-T)}{1-r_{min}(S-T)}
\end{aligned} \tag{3.114}$$

The last inequality in Eq. (3.114) proves that the upper gain margin using the balanced sensitivity $S-T$ is always equal or greater than the one using T. The three lower and upper gain margins are calculated for the example in Fig. 3.22(a) and are shown in Fig. 3.22(b). The possible values are displayed in colored bands corresponding to each transfer function, whereas the actual values in the example are marked at the horizontal axis. The results of the example underline the provided proofs and findings above.



(a) Illustrative singular value plots



(b) Lower and upper gain margins for $S(s), T(s)$ and $S(s)-T(s)$.

Figure 3.22: Illustrative results for closed-loop system with plant and PI-controller

3.6.4 Phase Margins

As mentioned above, the phase margin of the balanced sensitivity transfer function has the potential to achieve larger values.
In this part, it will be shown that the phase margin of $S-T$ is less conservative than the phase margin of T or S if the maximum singular values of $S-T$ and T or S are in a derived sub-domain of the possible domain of definition.
Since the phase margin equations of S and T are equivalent, and Fan's-Theorem holds for S and T, the proof is only shown for $S-T$ and S, but the proof for $S-T$ and T is analogous.
The hypothesis is that the following inequality holds:

$$\underbrace{2\,\mathrm{asin}\left(\frac{1}{\underset{\omega_S}{2\bar{\sigma}(S)}}\right)}_{\text{Phase Margin } \theta_S} \leq \underbrace{2\,\mathrm{atan}\left(\frac{1}{\underset{\omega_{S-T}}{\bar{\sigma}(S-T)}}\right)}_{\text{Phase Margin } \theta_{ST}} \tag{3.115}$$

Canceling the common factor 2 in Eq. (3.115) and applying the sine function on both sides of the inequality yields

$$\frac{1}{\underset{\omega_S}{2\bar{\sigma}(S)}} \leq \sin\left(\mathrm{atan}\left(\frac{1}{\underset{\omega_{S-T}}{\bar{\sigma}(S-T)}}\right)\right), \tag{3.116}$$

where the inequality is not altered by the sine because both arguments are defined in the interval [-90 deg, 90 deg]. Using the trigonometric equation

$$\sin(\mathrm{atan}(x)) = \frac{x}{\sqrt{1+x^2}}, \tag{3.117}$$

and squaring both sides we obtain the equivalent conditions:

$$\frac{1}{4\underset{\omega_S}{\bar{\sigma}^2}(S)} \leq \frac{\frac{1}{\underset{\omega_{S-T}}{\bar{\sigma}^2}(S-T)}}{1+\frac{1}{\underset{\omega_{S-T}}{\bar{\sigma}^2}(S-T)}} \tag{3.118}$$

$$\frac{1}{4\underset{\omega_S}{\bar{\sigma}^2}(S)} \leq \frac{1}{\underset{\omega_{S-T}}{\bar{\sigma}^2}(S-T)+1} \tag{3.119}$$

$$\underset{\omega_{S-T}}{\bar{\sigma}^2}(S-T) \leq 4\underset{\omega_S}{\bar{\sigma}^2}(S)-1 \tag{3.120}$$

By taking the square root we obtain:

$$\boxed{\underset{\omega_{S-T}}{\bar{\sigma}}(S-T) \leq \sqrt{4\underset{\omega_S}{\bar{\sigma}^2}(S)-1} \leq 2\underset{\omega_S}{\bar{\sigma}}(S)} \tag{3.121}$$

One can show that the square root in Eq. (3.121) has an upper-limit of $2\underset{\omega_S}{\bar{\sigma}}(S)$. By the application of Fan's theorem and the consideration of the maximum singular values at their respective frequencies the following inequalities are derived:

$$2\underset{\omega_S}{\bar{\sigma}}(S)+1 \geq 2\underset{\omega_{S-T}}{\bar{\sigma}}(S)+1 \geq \underset{\omega_{S-T}}{\bar{\sigma}}(S-T) \geq 2\underset{\omega_S}{\bar{\sigma}}(S)-1 \geq 2\underset{\omega_{S-T}}{\bar{\sigma}}(S)-1 \tag{3.122}$$

The inequalities in Eq. (3.121) and Eq. (3.122) are illustrated in Fig. 3.23. The singular values $\underset{\omega_S}{\bar{\sigma}}(S)$ and $\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)$ between the two red lines are feasible solutions to fulfill the inequalities in Eq. (3.122). The blue line indicates the border of the inequality in Eq. (3.121). The derived sub-domain is shown in Fig. 3.23 as black-dashed area. If the values $\underset{\omega_{S-T}}{\bar{\sigma}}(S-T)$ and $\underset{\omega_S}{\bar{\sigma}}(S)$ are within this area, the MIMO balanced phase margin has reduced conservatism in comparison to the MIMO phase margin of S or T. The example in Fig. 3.22 and the values presented in the example in Ref. [34] fulfill Eq. (3.121) and Eq. (3.122) and show thus reduced conservatism.

As a result, it can be stated, if Eq. (3.121) is fulfilled, the balanced MIMO phase margin is less conservative than the phase margin of the sensitivity transfer function. The proof within this work is confined to the domain which fulfills the condition in Eq. (3.121). The applied Fan's theorem in Eq. (3.122) narrows down the domain of possible maximum singular values of S or T, but does not prove reduced conservatism of the balanced MIMO phase margin globally. However, a sub-domain of possible maximum singular values for definite reduced conservatism is found (see Fig. 3.23).

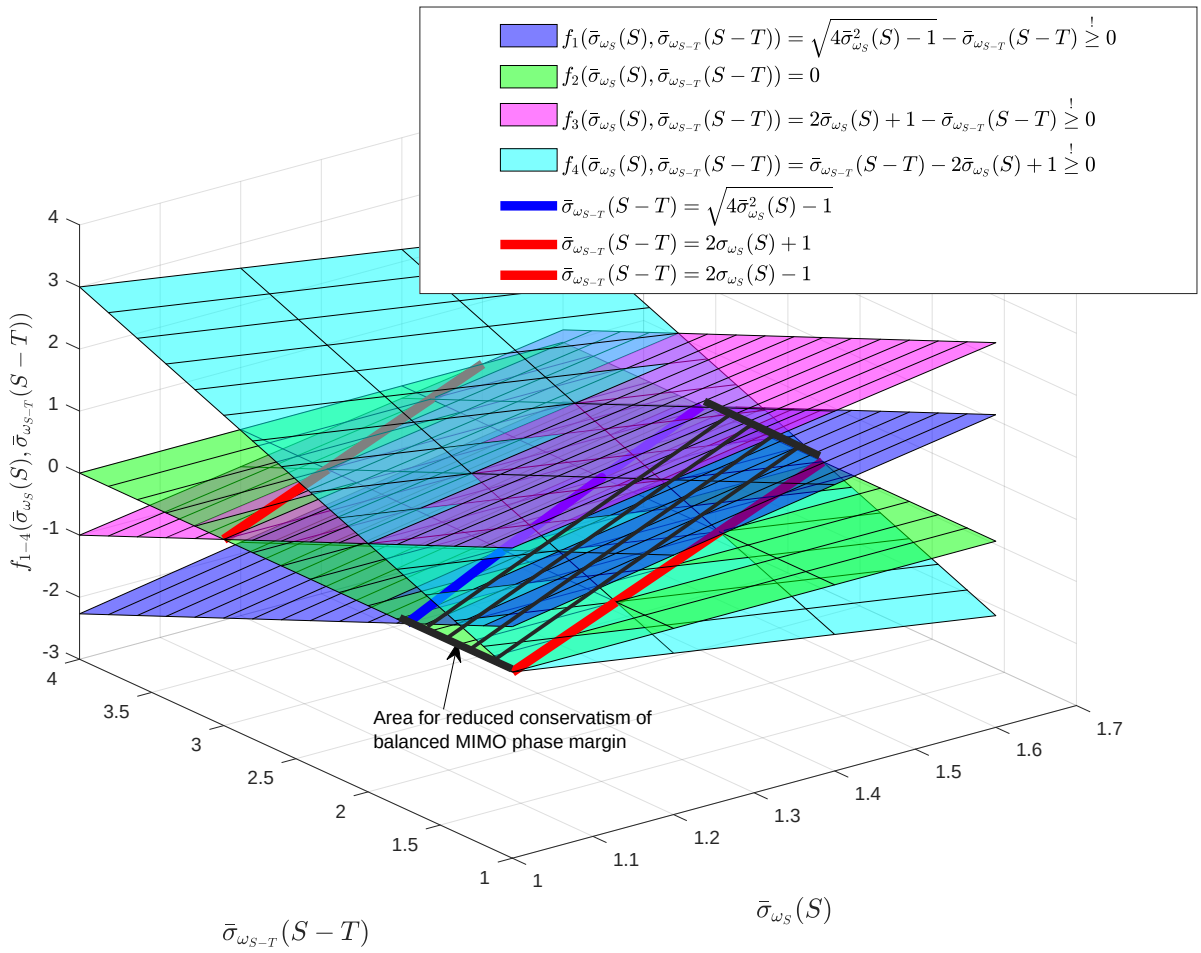


Figure 3.23: Illustration of MIMO phase margin proof for reduced conservatism of balanced sensitivity transfer function: the singular values $\bar{\sigma}_{\omega_S}(S)$ and $\bar{\sigma}_{\omega_{S-T}}(S-T)$ between the two red lines are feasible solutions to fulfill the inequalities in Eq. (3.122). The area marked in black fulfills Eq. (3.121) and Eq. (3.122). In this area the reduced conservatism of balanced MIMO phase margin is definite by the presented proof in comparison to the phase margin of S or T.

3.6.5 Conclusion and MIMO Margin Conservatism Discussion

It is proven above that the lower gain margin of $S-T$ is always equal or between the lower margin of S and the lower margin of T.

The upper gain margin of $S-T$ is always equal or between the upper margin of T and the upper margin of S.

As a result, the MIMO gain margins of $S-T$ are indeed balanced, since they are either dominated by S or T or both. Secondly, it is proven, that the MIMO phase margin of the balanced sensitivity transfer function is greater or equal to the phase margin of the sensitivity and complementary sensitivity function, if the maximum singular value of the balanced sensitivity transfer function fullfills a derived relation to the maximum singular value of the (complementary) sensitivity transfer function.

However, the question remains, which of the three MIMO margins are relevant. Since the

three MIMO margins are guaranteed robustness margins, the least conservative margins are of interest. The proof above shows the relation between the balanced sensitivity transfer function and the S or T.
In consequence, the relevant upper MIMO gain margin is the MIMO gain margin from the sensitivity transfer function. Analogously, it can be said, the relevant lower gain margin is the margin from the complementary transfer function (see i.a. Fig. 3.22). Both mentioned margins achieve the corresponding least conservative margin under consideration of the three presented MIMO gain margins.
Such a statement cannot be made about the phase margins. Although the balanced sensitivity transfer function $S-T$ can achieve higher margins, the reduced conservatism could not been proven by Fan's theorem globally. Instead, Sec. 3.10.7 presents MIMO margin results for a closed-loop system of an eVTOL demonstrator and shows that the balanced MIMO phase margin is not always greater than the phase margin from S or T. It can be said that the three presented phase margins are relevant and the greatest phase margin can be selected for the specific application. However, a sub-domain is identified, in which the balanced MIMO phase margin has indeed reduced conservatism.
The presented proofs are with respect to the specific margins of the sensitivity transfer function, the complementary sensitivity transfer function and the balanced sensitivity transfer function. Ref. [44] shows that these three margins are special cases of a general class of disk margins. A disk skew parameter can be specified, which is a real number. This parameter can morph continuously between the presented cases. The proof in this regard is left open for future work.

3.7 μ_Δ-Uncertainty Analysis for Guaranteed Stability

Doyle et. al (Ref. [46, 45, 47]) introduced the so-called μ_Δ-analysis, which is a powerful multi-loop robustness analysis method. The μ_Δ-analysis evaluates any arbitrary stable closed-loop LTI-system with regard to uncertainties. Therefore, the linear system, which is perturbed by specified uncertainties, is brought to the $\underline{M}\,\underline{\Delta}$-structure by a LFT. The $\underline{M}\,\underline{\Delta}$-structure is visualized in Fig. 3.24. Considering the $\underline{M}\,\underline{\Delta}$-structure, the μ_Δ-analysis solves a structured optimization problem. The so-called structured singular value $\mu_\Delta(\underline{M})$ is defined as follows:

$$\boxed{\mu_\Delta(\underline{M}) := \frac{1}{\min\limits_{\underline{\Delta}\in\underline{\underline{\Delta}}} \left(\bar{\sigma}(\underline{\Delta}) | \det(\underline{I} - \underline{\Delta}(s)\underline{M}(s)) = 0\right)}} \quad \text{with structured } \underline{\Delta} \in \underline{\underline{\Delta}} \tag{3.123}$$

Ref. [86, p.314] expresses the definition of μ_Δ in words by:

Find the smallest structured $\underline{\Delta}$ *(measured in terms of* $\bar{\sigma}(\underline{\Delta})$*),*
which makes $\det(\underline{I} - \underline{\Delta}(s)\underline{M}(s)) = 0$*; then* $\mu(\underline{M}) = 1/\bar{\sigma}(\underline{\Delta})$*.*

Thereby $\underline{\Delta}(s)$ is a structured complex/real uncertainty matrix, $\bar{\sigma}(\underline{\Delta})$ is the maximal singular value of $\underline{\Delta}(s)$, and $\underline{M}(s)$ is a complex matrix representing the closed-loop dynamics of the plant interconnected with the controller.

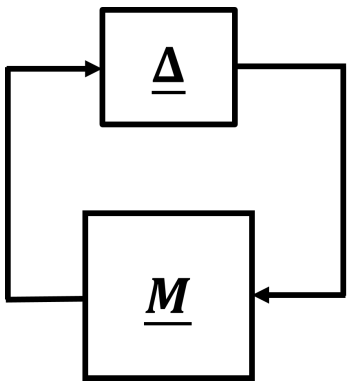


Figure 3.24: Exemplary $\underline{M}\,\underline{\Delta}$-structure with block-diagonal matrix $\underline{\Delta}$

The uncertainty $\underline{\Delta}(s)$ is structured and is within the set $\underline{\underline{\Delta}}$. Following the notation in Ref. [45], denote $\underline{I}_b$ the $b \times b$ identity matrix, consider the positive integers m_r, m_c and m_C with $m = m_r + m_c + m_C = n$, and define the block structure $\underline{\underline{B}}(m_r, m_c, m_C)$ as the m-tuple of positive integers

$$\underline{\underline{B}} = (b_1, ..., b_{m_r}, b_{m_{r+1}}, ..., b_{m_r+m_c}, b_{m_r+m_c+1}, b_m) \tag{3.124}$$

with $\Sigma_{i=1}^{m} b_i = n$. The set of allowed uncertainties is defined as:

$$\underline{\underline{\Delta}} = \{\underline{\Delta} = \textit{block-diag}\left(\Delta_1^R \underline{I}_{b_1}, ..., \Delta_{m_r}^R \underline{I}_{b_{m_r}}, \Delta_1^C \underline{I}_{b_{m_{r+1}}}, ..., \Delta_{m_c}^C \underline{I}_{b_{m_c+m_r}}, \underline{\Delta}_1^C, ..., \underline{\Delta}_{m_C}^C\right)\} \tag{3.125}$$

$$\text{with } \Delta_i^R \in \mathbb{R}, \Delta_i^C, \underline{\Delta}_i^C \in \mathbb{C}^{b_{m_r+m_c+i} \times b_{m_r+m_c+i}}$$

such that the matrix $\underline{\boldsymbol{\Delta}}$ is compatible with the matrix $\underline{\boldsymbol{M}}$ and *"that this block structure is sufficiently general to allow for repeated real scalars, repeated complex scalars, and full complex blocks"* (Ref. [45, p.1]).
As it can be seen, the definition of μ_Δ includes changing $\bar{\sigma}(\underline{\boldsymbol{\Delta}})$. It is favored that $\underline{\boldsymbol{\Delta}}$ is normalized, such that $\bar{\sigma}(\underline{\boldsymbol{\Delta}}) \leq 1$. This leads to another definition of μ_Δ:

$$\mu_\Delta(\underline{\boldsymbol{M}}) := \frac{1}{\min\limits_{c_\mu}\left(c_\mu \,|\, \det(\underline{\boldsymbol{I}} - c_\mu \underline{\boldsymbol{\Delta}}^*(s)\underline{\boldsymbol{M}}^*(s)) = 0, \bar{\sigma}(\underline{\boldsymbol{\Delta}}^*) \leq 1\right)} \quad \text{with } \underline{\boldsymbol{\Delta}}^* \in \underline{\underline{\boldsymbol{\Delta}}}', \tag{3.126}$$

under consideration of $\underline{\underline{\boldsymbol{\Delta}}}' = \{\underline{\boldsymbol{\Delta}}^* \in \underline{\underline{\boldsymbol{\Delta}}} : \bar{\sigma}(\underline{\boldsymbol{\Delta}}^*) \leq 1\}$, i.e. $\underline{\boldsymbol{\Delta}}^*$ has the same block-structure as $\underline{\boldsymbol{\Delta}}$, but it is scaled such that $\bar{\sigma}(\underline{\boldsymbol{\Delta}}^*) \leq 1$. This definition is also used subsequently within this thesis. Due to the normalization, the matrix $\underline{\boldsymbol{M}}$ is not equivalent to $\underline{\boldsymbol{M}}^*$, since the normalization constants are accumulated in $\underline{\boldsymbol{M}}^*$, such that the uncertainty matrix is normalized. This is visualized in Fig. 3.25. Note that, however, for readability the superscript $*$ and $'$ are dropped in the following of this thesis, such that $\underline{\boldsymbol{\Delta}} \in \underline{\underline{\boldsymbol{\Delta}}}$ with $\bar{\sigma}(\underline{\boldsymbol{\Delta}}) \leq 1$. Thus, the reciprocal of μ_Δ represents the smallest scalar factor the uncertainty matrix $\underline{\boldsymbol{\Delta}}$ can be multiplied with such that the closed-loop system is brought on the stability borderline for each frequency $\omega \in \mathbb{R}^+$. The inverse $\frac{1}{\mu_\Delta(\underline{\boldsymbol{M}})}$ is the smallest destabilizing factor c at one frequency, which satisfies the stability condition $\det(\boldsymbol{I} - c_\mu \underline{\boldsymbol{\Delta}}(s)\underline{\boldsymbol{M}}(s)) = 0$. The resulting real or/and complex uncertainties $c_\mu \underline{\boldsymbol{\Delta}}$ represent the minimal uncertainty parameters which drive the closed-loop system to the stability borderline at the frequency ω. Note that the respective $\underline{\boldsymbol{\Delta}}$ to c_μ fulfills always $\bar{\sigma}(\underline{\boldsymbol{\Delta}}) = 1$. The structured singular value μ_Δ is defined as zero, if no solution is found.

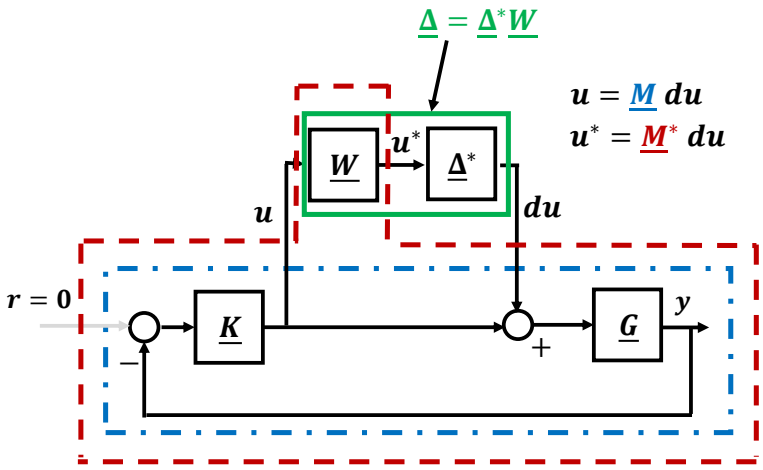


Figure 3.25: Normalization of uncertainty matrix $\underline{\boldsymbol{\Delta}}$ by matrix $\underline{\boldsymbol{W}}$ (same structure as $\underline{\boldsymbol{\Delta}}$), such that $\underline{\boldsymbol{\Delta}} = \underline{\boldsymbol{\Delta}}^* \underline{\boldsymbol{W}}$ with $\bar{\sigma}(\underline{\boldsymbol{\Delta}}^*) \leq 1$

The condition $\det(\boldsymbol{I} - \underline{\boldsymbol{\Delta}}(s)\underline{\boldsymbol{M}}(s))$ in the structured singular value definition represents the stability condition in Eq. (3.71).
In Ref. [97, p.133], the maximum peak of $\mu_\Delta(\underline{\boldsymbol{M}})$ is defined as another robustness margin:

$$\frac{1}{c_{\mu,max}} = \max_{\omega \in \mathbb{R}^+} \mu_\Delta(\underline{\boldsymbol{M}}(s)), \text{ with } s = j\omega \text{ and the frequency } \omega \tag{3.127}$$

If it is assumed that the nominal value of each uncertainty is equal to zero and the uncertainty intervals are symmetric, then the value $c_{\mu,max}$ represents the smallest factor with which the lower and upper uncertainty interval limit can be multiplied, to bring the system on the stability borderline. Thus, $c_{\mu,max}$ leads to $\det\left(\underline{\boldsymbol{I}} - c_{\mu,max}\underline{\boldsymbol{M}}\ \underline{\boldsymbol{\Delta}}\right) = 0$ at one frequency at least.
Note that μ_Δ can only be calculated in rare and simple cases analytically. Optimization algorithms are applied to calculate a lower and upper bound instead. The μ_Δ-analysis method can be used as a robustness margin, which considers detailed uncertainty models.
In general, the computational complexity of μ_Δ or its lower and upper bounds respectively has a combinatoric (non-polynomial) growth with the number of parameters considered (see Ref. [86, p.349]). Upper and lower bounds can be calculated for μ_Δ by available software packages (e.g. Ref. [98]), which apply inter alia the algorithms published in the references mentioned within this section. The following section explains the upper and lower boundary calculation.

3.7.1 Upper and Lower Bound

An upper bound for the structured singular value is found by the application of the Small Gain Theorem to the $\underline{\boldsymbol{M}}\,\underline{\boldsymbol{\Delta}}$-structure, illustrated in Fig. 3.24. For that reason, the following holds:

$$\bar{\sigma}(\underline{\boldsymbol{\Delta}})\bar{\sigma}(\underline{\boldsymbol{M}}) < 1 \tag{3.128}$$

$$\bar{\sigma}(\underline{\boldsymbol{\Delta}}) < \frac{1}{\bar{\sigma}(\underline{\boldsymbol{M}})} \leq \frac{1}{\mu_\Delta(\underline{\boldsymbol{M}})} \tag{3.129}$$

The equation holds, because the structured singular value μ_Δ is less conservative as the $\bar{\sigma}(\underline{\boldsymbol{M}})$-value. Thus, the value μ_Δ allows a greater $\sigma(\underline{\boldsymbol{\Delta}})$, i.e. greater uncertainty intervals. If the matrix $\underline{\boldsymbol{\Delta}}$ is a full block complex unstructured matrix, the smallest possible $\underline{\boldsymbol{\Delta}}$ yields $\mu_\Delta(\underline{\boldsymbol{M}}) = \bar{\sigma}(\underline{\boldsymbol{M}})$ (see Ref. [86], p.317).

A lower bound for the structured singular value μ_Δ is found by considering the constraint $\det(\underline{\boldsymbol{I}} - \underline{\boldsymbol{\Delta}}(s)\underline{\boldsymbol{M}}(s)) = 0$ of the optimization problem. As discussed in Sec. 3.5.3.3, the $\underline{\boldsymbol{M}}\,\underline{\boldsymbol{\Delta}}$-system becomes unstable, if $\det(\underline{\boldsymbol{I}} - \underline{\boldsymbol{\Delta}}(s)\underline{\boldsymbol{M}}(s)) = 0$. As a result,

$$\frac{1}{\lambda_i(\underline{\boldsymbol{M}})}\underline{\boldsymbol{I}} = \underline{\boldsymbol{\Delta}} \tag{3.130}$$

is a solution for the defined μ_Δ in Eq. (3.123) with λ_i as the ith eigenvalue of $\underline{\boldsymbol{M}}(s)$. The largest eigenvalue $\max_i|\lambda_i| = \rho(\underline{\boldsymbol{M}})$ defines the lower bound, which satisfies equation $\underline{\boldsymbol{M}}x_i = \lambda_i\, x_i$. The eigenvalue fulfills the following equation:

$$\det\left(\underline{\boldsymbol{I}} - \left(\frac{1}{\lambda_i}\underline{\boldsymbol{I}}\right)\underline{\boldsymbol{M}}\right) = 0 \;\text{ or }\; \det(\lambda_i\underline{\boldsymbol{I}} - \underline{\boldsymbol{M}}) = 0 \tag{3.131}$$

With $\frac{1}{\max|\lambda_i|}\underline{\boldsymbol{I}} = \underline{\boldsymbol{\Delta}}$, then $\bar{\sigma}(\underline{\boldsymbol{\Delta}}) = \frac{1}{\max|\lambda_i|} = \frac{1}{\rho(\underline{\boldsymbol{M}})}$, and the following bound holds:

$$\frac{1}{\mu_\Delta(\underline{\boldsymbol{M}})} \leq \frac{1}{\rho(\underline{\boldsymbol{M}})} \quad\text{and}\quad \mu_\Delta^{lb}(\underline{\boldsymbol{M}}) = \rho(\underline{\boldsymbol{M}}) \leq \mu_\Delta(\underline{\boldsymbol{M}}) \tag{3.132}$$

The lower bound μ_Δ^{lb} represents thus definitely one solution of the stability condition $\det(\boldsymbol{I} - \underline{\boldsymbol{\Delta}}^{lb}\,\underline{\boldsymbol{M}}(s)) = 0$ with $\underline{\boldsymbol{\Delta}}^{lb} = \frac{1}{\mu_\Delta^{lb}}\underline{\boldsymbol{I}}$. However, $\frac{1}{\rho(\underline{\boldsymbol{M}})}$ is possibly not the smallest factor with which the uncertainty matrix can be multiplied with, such that the system becomes unstable, since a greater μ_Δ might exist and thus also a smaller uncertainty matrix (measured by the maximum singular value). If the matrix $\underline{\boldsymbol{\Delta}}$ encompasses only the scalar complex uncertainty Δ^C (i.e. $\underline{\boldsymbol{\Delta}} = \underline{\boldsymbol{I}}\Delta^C$), the value μ_Δ is equivalent to the lower bound $\mu_\Delta(\underline{\boldsymbol{M}}) = \rho(\underline{\boldsymbol{M}})$ (cf. Ref. [86, p.317]).
The author wants to emphasize an important fact about the spectral radius $\rho(\underline{\boldsymbol{M}})$, which the interested researcher might overread in the original works in Refs. [99, 45, 46] and which is not detailed explicitly in Ref. [86].

For the calculation of the lower bound one needs to distinguish between the spectral radius $\rho_C(\underline{\boldsymbol{M}})$ and $\rho_R(\underline{\boldsymbol{M}})$. If $\underline{\boldsymbol{\Delta}}$ consists only of complex perturbations, then

$$\rho_C(\underline{\boldsymbol{M}}) = \rho(\underline{\boldsymbol{M}}) = \max\{|\lambda| : \lambda \textit{ is a complex eigenvalue of } \underline{\boldsymbol{M}}\}. \tag{3.133}$$

In contrast, if $\underline{\boldsymbol{\Delta}}$ consists of real and complex perturbations, then

$$\rho_R(\underline{\boldsymbol{M}}) = \rho(\underline{\boldsymbol{M}}) = \max\{|\lambda| : \lambda \textit{ is a real eigenvalue of } \underline{\boldsymbol{M}}\}. \tag{3.134}$$

If $\underline{\boldsymbol{M}}$ does not encompass the respective eigenvalue in the latter case, then $\rho_R(\underline{\boldsymbol{M}}) = 0$. By looking at Eq. (3.130) the reason for this important fact becomes clear. If the perturbations are only allowed to be real, the considered eigenvalue must be real, otherwise it does not fulfill the condition of a real perturbation. In consequence, it can be said that it is more difficult to find a solution of the lower bound in case of real uncertainties, since a reasonable spectral radius might not exist. To overcome this issue one might complexify the real uncertainties by adding a marginal complex part to the real uncertainties. However, this procedure makes the solution more conservative. Further details are beyond the scope of this thesis.

By combining the two derived boundaries, the following inequalities hold:

$$\frac{1}{\sigma(\underline{\boldsymbol{M}})} \leq \frac{1}{\mu_\Delta} \leq \frac{1}{\rho(\underline{\boldsymbol{M}})} \quad \text{or} \quad \boxed{\rho(\underline{\boldsymbol{M}}) \leq \mu_\Delta \leq \sigma(\underline{\boldsymbol{M}})} \tag{3.135}$$

with the applicable spectral radius $\rho(\underline{\boldsymbol{M}})$.

Additionally, Ref. [86, p.317] shows that the value $\mu_\Delta(\underline{\boldsymbol{M}})$ is equivalent to the upper bound, if the uncertainty matrix is a full, unstructured matrix. If the uncertainty matrix is a diagonal matrix, which contains only scalar complex perturbations, $\mu_\Delta(\underline{\boldsymbol{M}})$ is equivalent to the lower bound.
If the uncertainty matrix is block-diagonal, the lower and upper bound can be optimized by different solvers, in order to tighten both boundaries and to become less conservative. The calculation of tighter lower and upper boundaries are elaborated in the next section.

3.7.2 Tighter Lower and Upper Bound

For the reduction of conservatism the lower and upper μ-boundary can be tightened by the application of optimization methods. The lower bound can be maximized by the introduction of the matrix $\underline{\boldsymbol{U}}$, such that the following optimization problem of the lower bound can be formulated:

$$\max_{\underline{\boldsymbol{U}} \in \underline{\underline{\boldsymbol{U}}}} \rho_R(\underline{\boldsymbol{U}}\,\underline{\boldsymbol{M}}) \leq \mu(\underline{\boldsymbol{M}}) \quad \text{at every frequency } \omega \tag{3.136}$$

with the scaling matrix $\underline{\boldsymbol{U}} \in \underline{\underline{\boldsymbol{U}}}$ and in case of a mixed $\underline{\boldsymbol{\Delta}}$ with real and complex uncertainties. The matrix $\underline{\boldsymbol{U}}$ has the same block structure of $\underline{\boldsymbol{\Delta}}$. Ref. [45] defines the set $\underline{\underline{\boldsymbol{U}}}$ with the following properties:

$$\underline{\underline{\boldsymbol{U}}} = \{\underline{\boldsymbol{\Delta}} \in \underline{\underline{\boldsymbol{\Delta}}},\ \Delta_i^R \in [-1,1], \Delta_i^{C*}\Delta_i^C = 1, \underline{\Delta}_i^{C*}\underline{\Delta}_i^C = \underline{\boldsymbol{I}}_{k_{m_r+m_c+i}}\} \tag{3.137}$$

Thereby, $\Delta_i^{C*}, \underline{\Delta}_i^{C*}$ denote the complex conjugate transpose of $\Delta_i^C, \underline{\Delta}_i^C$. Generally, $\underline{\boldsymbol{U}}$ is not an unitary matrix. However, if $\underline{\boldsymbol{\Delta}}$ encompasses only complex perturbations, the matrix $\underline{\boldsymbol{U}}$ becomes unitary, such that $\underline{\boldsymbol{U}}^\star\underline{\boldsymbol{U}} = \underline{\boldsymbol{I}}$. Young et. al (Ref. [46]) shows that actually the following is an equality:

$$\max_{\underline{\boldsymbol{U}} \in \underline{\underline{\boldsymbol{U}}}} \rho_R(\underline{\boldsymbol{U}}\,\underline{\boldsymbol{M}}) = \max_{\underline{\boldsymbol{\Delta}} \in \underline{\underline{\boldsymbol{\Delta}}}} \rho_R(\underline{\boldsymbol{\Delta}}\,\underline{\boldsymbol{M}}) = \mu_\Delta(\underline{\boldsymbol{M}}) \quad \text{at every frequency } \omega \tag{3.138}$$

Due to *"tractability"* (Ref. [46, p.3]), the left hand side of the equation is further used. Nevertheless, this is a non-convex optimization problem. For that reason, μ_Δ is approximated from below with $\mu_\Delta^{lb} = \max_{\underline{\boldsymbol{U}} \in \underline{\underline{\boldsymbol{U}}}} \rho_R(\underline{\boldsymbol{U}}\,\underline{\boldsymbol{M}}) \leq \mu_\Delta(\underline{\boldsymbol{M}})$. Note that $\underline{\underline{\boldsymbol{U}}} \subset \underline{\underline{\boldsymbol{\Delta}}}$ (in which $\bar{\sigma}(\underline{\boldsymbol{\Delta}}) \leq 1$), due to its definition in Eq. (3.137). The elaborated proof and algorithm for mixed uncertainties

in this regard can be found in Ref. [46]. The solution of the lower bound optimization $\underline{\boldsymbol{U}}^{lb}$ is utilized to calculate the *worst-case* uncertainty which drives the system definitely on the stability borderline. The entries of matrix $\underline{\boldsymbol{U}}^{lb}$ together with the lower bound of the structured singular value $\mu_{\Delta}^{lb} = \max \rho_R(\underline{\boldsymbol{U}}\,\underline{\boldsymbol{M}})$ represent by

$$\left(1/\mu_{\Delta}^{lb}\right) \underline{\boldsymbol{U}}^{lb} = \underline{\boldsymbol{\Delta}}^{WC} \tag{3.139}$$

the values of the possible worst-case uncertainty matrix $\underline{\boldsymbol{\Delta}}^{WC}$, which leads to $\det(\underline{\boldsymbol{I}} - \underline{\boldsymbol{\Delta}}^{WC}(s)\underline{\boldsymbol{M}}(s)) = 0$. These derived uncertainties are not necessarily the smallest possible uncertainties for instability (measured by the spectral radius), since the uncertainty matrix of the lower μ-boundary is considered. The calculated uncertainties values bring the system to the stability borderline. However, smaller uncertainties (measured by the spectral radius) might exist, since not the exact μ-value is considered, but the lower μ-boundary. But they are at least uncertainties, which bring the system definitely to the stability limit of the closed-loop system.

Equivalently, the upper bound is reduced by setting up a second optimization problem. Therefore, the matrix $\underline{\boldsymbol{D}} \in \underline{\underline{\boldsymbol{D}}}$, which is a block-diagonal scaling matrix, is introduced. According to Ref. [45], the set $\underline{\underline{\boldsymbol{D}}}$ is defined by

$$\underline{\underline{\boldsymbol{D}}} = \{\textit{block-diag}(\underline{D}_1, ..., \underline{D}_{m_r+m_c}, d_1\underline{\boldsymbol{I}}_{k_{m_r+m_c+1}}, ..., d_{m_C}\underline{\boldsymbol{I}}_{k_m})\} \tag{3.140}$$

$$\text{with} \quad \det(\underline{D}_i) \neq 0,\ \underline{D}_i \in \mathbb{C}^{k_i \times k_i},\ 0 \neq d_i \ \in \mathbb{R}. \tag{3.141}$$

The matrix $\underline{\boldsymbol{D}}$ commutes with the uncertainty matrix $\underline{\boldsymbol{\Delta}}\,\underline{\boldsymbol{D}} = \underline{\boldsymbol{D}}\,\underline{\boldsymbol{\Delta}}$ (compare with uncertainty definition structure in Eq. (3.125), specifically the diagonal elements of $\underline{\boldsymbol{D}}$ and $\underline{\boldsymbol{\Delta}}$). The introduction of the matrix $\underline{\boldsymbol{D}}$ is visualized in Fig. 3.26. Due to its structure, the introduction of the matrix $\underline{\boldsymbol{D}}$ has no effect on the uncertainty matrix $\underline{\boldsymbol{\Delta}}$ and on the application of the Small Gain Theorem. It can be applied to the matrix $\underline{\boldsymbol{D}}\,\underline{\boldsymbol{M}}\,\underline{\boldsymbol{D}}^{-1}$, instead of $\underline{\boldsymbol{M}}$ (see e.g. Refs. [86, p.313], [46, p.1] for more details).
The property is used to minimize the upper μ-boundary:

$$\mu_{\Delta}(\underline{\boldsymbol{M}}) \leq \min_{\underline{\boldsymbol{D}} \in \underline{\underline{\boldsymbol{D}}}} \bar{\sigma}(\underline{\boldsymbol{D}}\,\underline{\boldsymbol{M}}\,\underline{\boldsymbol{D}}^{-1}) = \mu_{\Delta}^{ub} \quad \text{at every frequency } \omega \tag{3.142}$$

The improved upper μ-bound is the conservative, but relevant bound to verify whether the closed-loop system is able to handle the given uncertainties. Thus, the upper bound does not necessarily represent the uncertainties, which bring the system to the stability limit (due to its conservatism), but it gives the reciprocal factor c_μ the uncertainty intervals can be multiplied with, such that a stable system is still guaranteed (assuming the uncertainty intervals are symmetric and their nominal value is zero).

Several optimization approaches for the lower and upper boundary of the singular value μ were published. In Ref. [100], many algorithm for mainly the μ-lower bound are introduced and compared, but also a short overview for μ-upper bound algorithm is given. Within this work, the core structured singular value function *mussv* from Ref. [98] is used to calculate the lower and upper bound and the worst-case uncertainty matrix $\underline{\boldsymbol{\Delta}}^{WC}$.

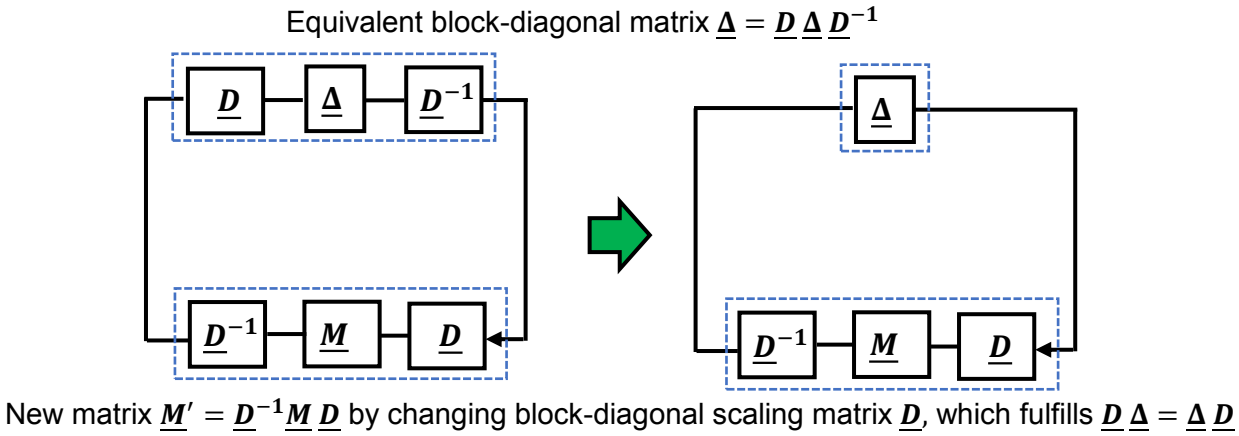


Figure 3.26: μ-upper bound minimization by introducing the scaling matrix $\underline{\boldsymbol{D}}$

3.8 Advanced Robust Performance Criteria

The methods, elaborated in the last sections, are able to analyze the closed-loop system with regard to robust stability. Another important property of closed-loop systems is robust performance. As mentioned in Sec. 3.2.4, classical robust performance metrics in the time domain are rise time, settling time, steady-state tracking error, decay ratio and steady-state offset in off-nominal conditions.
An advanced robust performance technique in the frequency domain is the $\mu_{\Delta\Gamma}$-analysis. To the author's knowledge, the $\mu_{\Delta\Gamma}$-analysis approach was introduced by Ref. [97, Ch.7] as an enhancement to the μ_{Δ}-analysis, named in context of the μ_{Δ}-analysis as *Extension to Modal Performance Analysis*. In Ref. [90], the approach is formalized and called the $\mu_{\Delta\Gamma}$-analysis. Thereby, the system is analyzed with regard to a conformity borderline, instead of the stability borderline.
Sec. 3.8.1 recapitulates the $\mu_{\Delta\Gamma}$ approach. Subsequently, Sec. 3.8.2 proposes an approach to determine uncertainty dynamics, which bring the system towards the conformity borderline.

3.8.1 $\mu_{\Delta\Gamma}$-Uncertainty Analysis for Guaranteed Handling Qualities

The $\mu_{\Delta\Gamma}$-analysis is able to calculate worst-case uncertainties with regard to respective conformity borders. These conformity borders can be extracted from aerospace certification documents such as, e.g., the ADS-33 (cf. Ref. [80]) and MIL-F-8785C (cf. Ref. [83]), which comprise specific handling quality borderlines from Level 1 to Level 3 in the frequency domain. In consequence, the $\mu_{\Delta\Gamma}$-analysis is a powerful approach to show compliance of an uncertain system with regard to handling quality requirements.
Considering the LTI closed-loop system $\underline{\boldsymbol{G}}$ with the state space model matrices $\underline{\boldsymbol{A}}, \underline{\boldsymbol{B}}, \underline{\boldsymbol{C}}, \underline{\boldsymbol{D}}$, conformity is defined as follows:

Conformity Definition (reworked from Ref. [90, p.94]):
Assume, the conformity border Γ *is a continuous function* $\Gamma(\omega) = \sigma_\Gamma(\omega) + j\omega$ *in the complex plane confined to the set* $\mathbb{D} = \{\omega | \omega \in]\infty, +\infty[\ \}$*, which is also symmetric with regard to the real axis. The LTI-system with* n *eigenvalues / poles* $p_i = \lambda_i(\underline{\boldsymbol{A}}) = \sigma_i + j\omega_i$ *with* $\{\lambda_i(\underline{\boldsymbol{A}})\} = \{s | \det(s\underline{\boldsymbol{I}} - \underline{\boldsymbol{A}})\}$ *is conform with regard to* $\Gamma(\omega)$ *if*

$$\mathrm{Re}(p_i) = \mathrm{Re}(\lambda_i(\underline{\boldsymbol{A}})) < \mathrm{Re}(\Gamma(\omega_i)), \forall i \tag{3.143}$$

$$\Leftrightarrow \mathrm{Re}(p_i) = \mathrm{Re}(\lambda_i(\underline{\boldsymbol{A}})) < \sigma_\Gamma(\omega_i), \forall i \tag{3.144}$$

$$\Leftrightarrow \sigma_i < \sigma_\Gamma(\omega_i), \forall i. \tag{3.145}$$

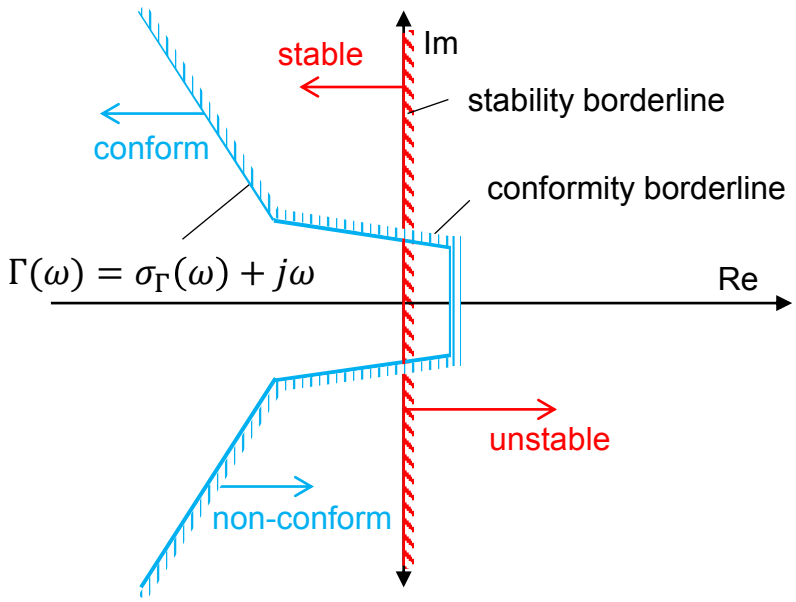


Figure 3.27: Exemplary conformity borderline $\Gamma(\omega)$ (reworked from (Ref. [90, p.92])

A linear closed-loop system is conform, if its poles are located on the left side of a conformity borderline Γ in the complex plane. This is analogous to the pole stability definition, which states that a system is stable, if its poles are on the left side of the imaginary axis in the complex plane. An exemplary conformity borderline is shown in Fig. 3.27.

The structured singular value can be calculated with regard to the conformity border Γ, instead of the stability borderline. The respective structured singular value is then called $\mu_{\Delta\Gamma}$. Based on Ref. [90, p.103], the structured singular value $\mu_{\Delta\Gamma}$ is defined as:

$$\mu_{\Delta\Gamma}(\underline{\boldsymbol{M}}) := \frac{1}{\min\limits_{c_\mu}\left(c_\mu \,|\, \det(\underline{\boldsymbol{I}} - c_\mu \underline{\boldsymbol{\Delta}}(s_\Gamma)\underline{\boldsymbol{M}}(s_\Gamma)) = 0, \bar{\sigma}(\underline{\boldsymbol{\Delta}}) \leq 1\right)} \quad \text{with } \underline{\boldsymbol{\Delta}} \in \underline{\underline{\boldsymbol{\Delta}}} \tag{3.146}$$

with regard to the conformity borderline $s_\Gamma = \sigma_\Gamma(\omega) + j\omega$. Note again that the uncertainty matrix is normalized, such that $\bar{\sigma}(\underline{\boldsymbol{\Delta}}) \leq 1$.

A necessary condition for robust compliance is nominal compliance. That means, it is important to check first if the nominal closed-loop system is conform with regard to the considered handlinq quality borderline. This needs to be tested before $\mu_{\Delta\Gamma}$ is calculated. As Ref. [90, Sec. 7.3.2] points out, equivalent properties of μ_Δ can be established for $\mu_{\Delta\Gamma}$. Also the same algorithm, introduced in Sec. 3.7.2, can be applied to narrow down the lower and upper $\mu_{\Delta\Gamma}$-boundary.

3.8.2 Worst-Case Uncertainty Transfer Function for Stability Compliance

One important result of the μ-lower bound calculation is the uncertainty matrix $\underline{\boldsymbol{\Delta}}^{WC}$. This matrix is a possible candidate for the worst-case uncertainty matrix. However, it is definitely uncertainty matrix, which brings the system towards the respective borderline. The entries of the matrix Δ_i^{WC} are complex or real numbers, depending on how the uncertainties are modeled. Real parametric uncertainties can be easily implemented as real numbers into the model.

This and the next section elaborate how scalar worst-case complex uncertainties can be modeled as transfer functions. The sections correspond to *Contribution II.2* of this thesis.

Thereby, the μ-lower bound is considered as worst-case, since the μ-upper bound gives a guarantee for stability/conformity, but does not guarantee that the stability/conformity condition is violated (due to the fact that the μ-upper bound is conservative). In contrast, the lower bound does not guarantee that the worst-case uncertainty is found, but gives at least uncertainty intervals, which bring the system definitely on the considered borderline. For that reason, the μ-lower bound is considered to calculate the worst-case transfer function.

In order to emphasize this relation, the proposed transfer function is denoted as worst-case transfer function candidate. If the μ-lower bound is equivalent to the μ-upper bound at the worst-case frequency ω_{wc}, the transfer function can be renamed to worst-case transfer function.

The necessary conditions for this transfer function are elaborated next. The presented approach can be applied to any worst-case scalar complex uncertainty, which was calculated for the stability borderline, but also for any valid conformity borderline.

At first the approach is explained for the stability borderline and subsequently, for the conformity borderline.

It appears that the calculation of the following introduced worst-case transfer function with regard to the stability borderline can be already computed by Ref. [98] with the command *robstab*, since the results of the presented transfer function have matched with the results from Ref. [98] in simple control examples. However, no documentation could be found in this regard about the background of the coefficient calculation. Thus, the following section contributes the derivation of the worst-case transfer function candidate with regard to the stability borderline and the respective proof. Moreover, to the author's knowledge, the calculation of the worst-case transfer function candidate with respect to the conformity borderline in the subsequent section has not been published before.

Assume, the scalar uncertainties $\Delta_i \in \mathbb{C}, \forall i$ are modeled as complex perturbations. Suppose the μ-lower bound is calculated for a closed-loop system including $\Delta_i, \forall i$. The corresponding worst-case uncertainties $\Delta_i^{WC}(j\omega = j\omega_{WC}) = \sigma_{\Delta i}^{WC} + j\omega_{\Delta i}^{WC}, \forall i$ result from the μ-lower bound

optimization under consideration of Eq. (3.139). The worst-case frequency is at the maximum of the lower bound μ_{Δ}^{lb}:

$$\omega_{WC} = \arg\max_{\omega} \mu_{\Delta}^{lb}(\omega) \tag{3.147}$$

A suitable transfer function candidate for the uncertainty Δ_i is:

$$\Upsilon_{\Delta}(s) = K_{\Delta} \underbrace{\frac{(s-x_{\Delta})^2}{(s+x_{\Delta})^2}}_{\upsilon_{\Delta 1}(s)} \underbrace{\frac{s}{s^2+\sqrt{2}\omega_{WC}s+\omega_{WC}^2}}_{\upsilon_{\Delta 2}(s)} \tag{3.148}$$

with the gain $K_{\Delta} \in \mathbb{R} \setminus \{0\}$ and the variable $x_{\Delta} \in \mathbb{R}^+$. K_{Δ} and x_{Δ} are used to adjust the transfer function $\Delta(s)$ to get the worst-case transfer function for Δ_i^{WC}. Thereby, the transfer functions $\upsilon_{\Delta 1}$ and $\upsilon_{\Delta 2}$ represent an *all-pass* and a *DT-2 element*. In the following, it is step-by-step explained, how the transfer function $\Delta(s)$ is found considering the worst-case uncertainty $\Delta^{WC}(j\omega_{WC}) = \sigma_{\Delta}^{WC} + j\omega_{\Delta}^{WC}$. The subscript i is dropped for readability:

Necessary Conditions: The worst-case uncertainty $\Delta^{WC}(j\omega_{WC})$ is a complex number, which is characterized by its amplitude $|\Delta^{WC}|$ and phase angle $\varphi(\Delta^{WC})$. In order to fulfill the property of the worst-case uncertainty, the corresponding transfer function shall have the amplitude $\Delta^{WC}(j\omega_{WC}) \in \mathbb{C}$ and phase $\varphi(\Delta^{WC})$ at the frequency ω_{WC}:

$$\text{Gain condition} : |\Delta^{WC}(j\omega_{WC})| \stackrel{!}{=} |\Upsilon_{\Delta}(j\omega_{WC})| \tag{3.149}$$

$$\text{Phase condition} : \varphi(\Delta^{WC}(j\omega_{WC})) \stackrel{!}{=} \varphi(\Upsilon_{\Delta}(j\omega_{WC})) \tag{3.150}$$

In addition, the transfer function shall have the maximum gain at this frequency ω_{WC}, otherwise the transfer function would not be a candidate for the worst-case transfer function.

$$\text{Worst-case condition} : |\Delta^{WC}(j\omega_{WC})| \stackrel{!}{=} \max_{\omega} |\Upsilon_{\Delta}(j\omega)| \tag{3.151}$$

***All-pass* Determination:** To fulfill the three conditions above, the transfer function Υ_{Δ} in Eq. (3.148) is chosen as a suitable candidate. Fig. 3.28 shows the amplitude and phase of the transfer functions $\upsilon_{\Delta 1}$ and $\upsilon_{\Delta 2}$, which are part of the transfer function Υ_{Δ}. As illustrated in Fig. 3.28, the phase of the *DT2*-element is equivalent to $\varphi(\upsilon_{\Delta 2}(j\omega_{WC})) = 0$ at the critical frequency ω_{WC}. This property allows, that the phase condition in Eq. (3.150) can be fulfilled by setting the variable x_{Δ} of the *all-pass* transfer function $\upsilon_{\Delta 1}(s)$ accordingly. It can be realized due to the superposition principle of the Bode diagram. Therefor, the phase of $\upsilon_{\Delta 1}$ is

calculated as follows. First, the first order *all-pass* $\upsilon'_{\Delta 1}$ is considered :

$$\upsilon'_{\Delta 1}(j\omega) = \frac{j\omega - x_\Delta}{j\omega + x_\Delta} = \frac{(j\omega - x_\Delta)^2}{(j\omega + x_\Delta)(j\omega - x_\Delta)} = \frac{x_\Delta^2 - 2(j\omega)x_\Delta + (j\omega)^2}{-x_\Delta^2 - \omega^2} \tag{3.152}$$

The respective real and imaginary parts are calculated by:

$$\text{Real Part: } \mathrm{Re}(\upsilon'_{\Delta 1}(j\omega)) = \frac{x_\Delta^2 - \omega^2}{-\omega^2 - x_\Delta^2} \tag{3.153}$$

$$\text{Imaginary Part: } \mathrm{Im}(\upsilon'_{\Delta 1}(j\omega)) = \frac{-2\omega x_\Delta}{-\omega^2 - x_\Delta^2} \tag{3.154}$$

The phase of $\upsilon'_{\Delta 1}$ is calculated by:

$$\tan\left(\varphi(\upsilon'_{\Delta 1}(j\omega))\right) = \frac{\mathrm{Im}(\upsilon'_{\Delta 1}(j\omega))}{\mathrm{Re}(\upsilon'_{\Delta 1}(j\omega))} = \frac{-2\omega x_\Delta}{x_\Delta^2 - \omega^2} \tag{3.155}$$

Consider the superposition principle of the Bode diagram, the phase $\varphi(\upsilon_{\Delta 1})$ of the second order *all-pass* element is twice the phase $\varphi(\upsilon'_{\Delta 1})$ of the first order *all-pass* element:

$$\varphi(\upsilon_{\Delta 1}) = 2\varphi(\upsilon'_{\Delta 1}) \tag{3.156}$$

To satisfy the phase condition in Eq. (3.150), the following equation needs to hold:

$$\varphi(\Delta^{WC}(j\omega_{WC})) \stackrel{!}{=} \varphi(\Upsilon_\Delta(j\omega_{WC})) \Leftrightarrow \varphi(\Delta^{WC}(j\omega_{WC})) \stackrel{!}{=} \varphi(\upsilon_{\Delta 1}(j\omega_{WC})) \tag{3.157}$$

The equation can be transformed to:

$$\tan\varphi\left(\frac{\Delta^{WC}(j\omega_{WC})}{2}\right) \stackrel{!}{=} \tan\varphi\left(\frac{\upsilon_{\Delta 1}(j\omega_{WC})}{2}\right) \tag{3.158}$$

With Eq. (3.155) and Eq. (3.156) Eq. (3.158) becomes:

$$\tan\varphi\left(\frac{\Delta^{WC}(j\omega_{WC})}{2}\right) \stackrel{!}{=} \frac{-2\omega_{WC}x_\Delta}{x_\Delta^2 - \omega_{WC}^2} \tag{3.159}$$

Finally, the equation can be restructured to:

$$\underbrace{y}_{=a} \cdot x_\Delta^2 + \underbrace{2\omega_{WC}}_{=b} \cdot x_\Delta \underbrace{-y\omega_{WC}^2}_{=c} = 0 \quad \text{with} \quad y = \tan\varphi\left(\frac{\Delta^{WC}(j\omega_{WC})}{2}\right) \tag{3.160}$$

This quadratic formula can be solved for x_Δ:

$$x_{\Delta_{1,2}} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-2\omega_{WC} \pm \sqrt{4\omega_{WC}^2 + 4y^2\omega_{WC}^2}}{2y} = \frac{-\omega_{WC} \pm \omega_{WC}\sqrt{1+y^2}}{y} \tag{3.161}$$

Considering $x_\Delta \in \mathbb{R}^+$, the number of possible solutions reduces to one valid solution.
As a result, the phase condition can be set by the determination of x_Δ alone. Instead of the first order filter $\upsilon'_{\Delta 1}$, the second order *all-pass* $\upsilon_{\Delta 1}$ is chosen to cover the entire phase interval from -360 deg to 0 deg. The amplitude of Υ_Δ is unaffected by the *all-pass*, since its amplitude $|\upsilon_{\Delta 1}(j\omega)| = 1, \forall\omega$ (see Fig. 3.28).

Gain K_Δ Determination: The gain K_Δ is calculated in the following such that the amplitude condition in Eq. (3.149) is satisfied:

$$|\Delta^{WC}(j\omega_{WC})| \stackrel{!}{=} |\Upsilon_\Delta(j\omega_{WC})| = |K_\Delta| \cdot \underbrace{|\upsilon_{\Delta 1}(j\omega_{WC})|}_{=1} \cdot |\upsilon_{\Delta 2}(j\omega_{WC})| \tag{3.162}$$

As noted, the amplitude of the *all-pass* element $\upsilon_{\Delta 1}(j\omega) = 1, \forall\omega$. Thus, K_Δ is determined by:

$$K_\Delta = \frac{|\Delta^{WC}(j\omega_{WC})|}{|\upsilon_{\Delta 2}(j\omega_{WC}|} \quad \text{with} \quad K_\Delta \in \mathbb{R} \setminus \{0\} \tag{3.163}$$

With determination of the gain K_Δ and the variable x_Δ, the worst-case transfer function Υ_Δ is fully defined. The resulting transfer function with the *all-pass* element $\upsilon_{\Delta 1}$, the *DT2*-element and the gain K_Δ is illustrated in Fig. 3.29.

The *DT2*-Element and the Worst-Case Proof: In order to prove that $\Upsilon_\Delta(j\omega)$ is a valid worst-case transfer function, the worst-case condition in Eq. (3.151) needs to be fulfilled. To prove the worst-case condition, one needs to show that the global maximum of $|\Upsilon_\Delta(j\omega)|$ occurs at ω_{WC}. This can be demonstrated analytically. Within this work, it is shown based on a graphical demonstration. Since $\upsilon_{\Delta 1}(j\omega) = 1, \forall\omega$, the focus can be set on the amplitude of the *DT2*-element, which is illustrated in Fig. 3.28. It can be seen, that the only one maximum occurs at the frequency ω_{WC}, which is the worst-case condition.
The reason becomes clear by the separation of the *DT2*-element into one *D*-element and *PT2*-element. The damping of the corresponding *PT2*-element is $\zeta = \sqrt{2}/2$, which is equivalent to the known *optimal damping* in flight control theory. The *optimal damping* is the lowest possible damping, at which no resonance peak in the amplitude over ω occurs. For that reason, the *DT2*-element is an appropriate transfer function as part of the worst-case transfer function Υ_Δ and leads to the global maximum at the worst-case frequency ω_{WC}.

It could be shown, that the transfer function candidate $\Upsilon_{\Delta}(j\omega)$ in Eq. (3.148) is a valid worst-case transfer function since the three necessary conditions can be fulfilled. Thus, a worst-case transfer function for scalar complex perturbations is found for the stability borderline of a closed-loop system.
Note that using the *optimal damping* is a particular solution for the worst-case transfer function to fulfill the presented conditions. However, other damping values can be also chosen. The presented method is therefore still valid, however, the worst-case frequency needs to be chosen such that it is equivalent to the resonance frequency instead of the eigenfrequency. This can be elaborated in future work.

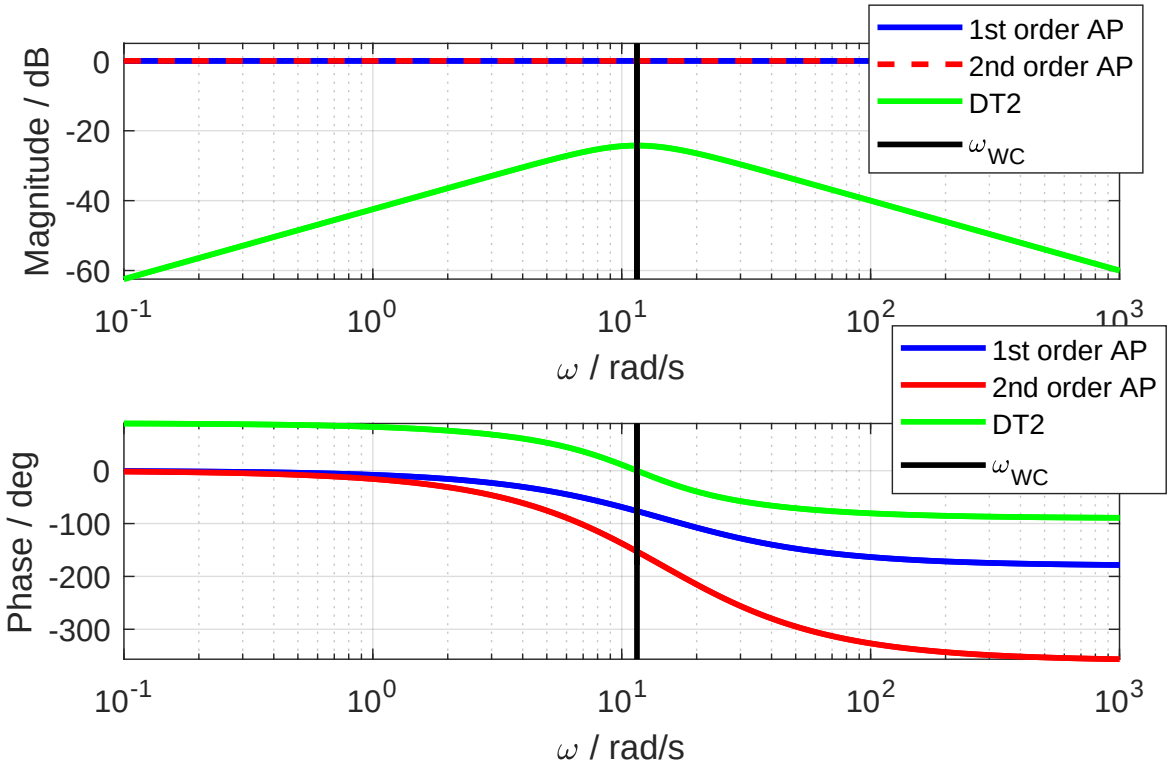


Figure 3.28: Bode diagram of the *all-pass* (AP) transfer function $\upsilon_{\Delta 1}(s)$ and the *DT2*-element $\upsilon_{\Delta 2}(s)$

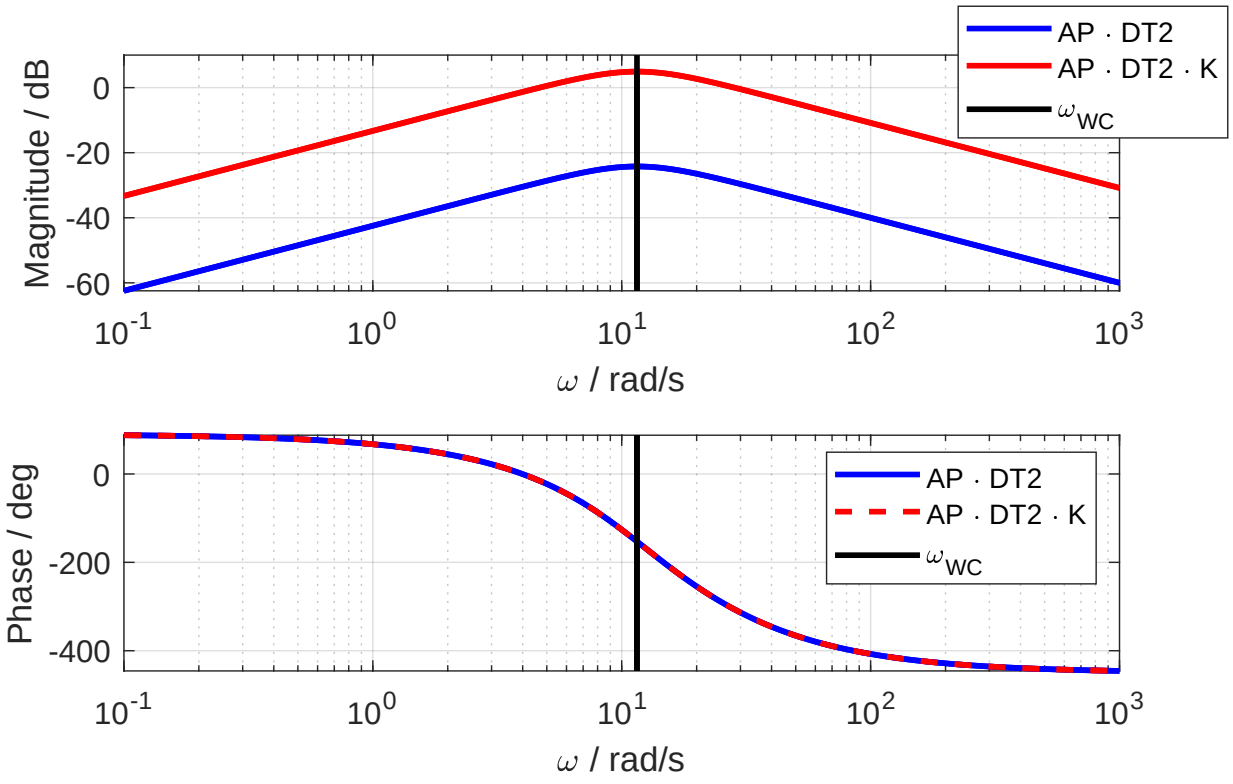


Figure 3.29: Bode diagram of the worst-case transfer function candidate $\Gamma_{\Delta}(s)$ (red), which consists of the second-order *all-pass* (AP) $\upsilon_{\Delta 1}(s)$, the *DT2*-element $\upsilon_{\Delta 2}(s)$ and the gain K

3.8.3 Worst-Case Uncertainty Transfer Function for Handling Quality Compliance

Worst-case transfer function candidates can be also derived for valid conformity borderlines. These uncertainty transfer functions bring the studied closed-loop system on the conformity borderline, which specifies, for example, required handling qualities.

Suppose, the scalar uncertainties $\Delta_i \in \mathbb{C}, \forall i$ are modeled as complex perturbations within the LTI closed-loop system. Note that the lower bound $\mu^{lb}_{\Delta\Gamma}$ with regard to the conformity borderline s_Γ is computed. Thereby, the respective worst-case uncertainties $\Delta^{WC}_{\Gamma,i} = \sigma^{WC}_{\Delta\Gamma,i} + j\omega^{WC}_{\Delta\Gamma,i}$ are obtained. The worst-case is defined in the following at the maximum of the lower bound $\mu^{lb}_{\Delta\Gamma}$:

$$s^{WC}_\Gamma = \arg\max_{s_\Gamma} \mu^{lb}_{\Delta\Gamma}(s_\Gamma) \tag{3.164}$$

Note that the structured singular value $\mu_{\Delta\Gamma}$ and its lower/upper bound are evaluated along the line $s_\Gamma = \sigma_\Gamma(\omega) + j\omega$. The subscript i is dropped in the following for readability.

Based on the worst-case conditions with respect to the stability borderlines elaborated in the last subsection, the following properties can be derived for the worst-case transfer function candidates regarding the conformity borderline:

$$\text{Amplitude condition} : |\Delta^{WC}_\Gamma| \stackrel{!}{=} |\Upsilon_\Gamma(s^{WC}_\Gamma)| \tag{3.165}$$

$$\text{Phase condition} : \varphi(\Delta^{WC}_\Gamma) \stackrel{!}{=} \varphi(\Upsilon_\Gamma(s^{WC}_\Gamma)) \tag{3.166}$$

$$\text{Worst-case condition} : |\Delta^{WC}_\Gamma| \stackrel{!}{=} \max_{s_\Gamma} |\Upsilon_\Gamma(s^{WC}_\Gamma)| \tag{3.167}$$

The amplitude, the phase and the worst-case condition are evaluated at s^{WC}_Γ. A suitable candidate for the worst-case transfer function with regard to the conformity borderline is:

$$\Upsilon_\Gamma(s) = K_\Gamma \underbrace{\frac{(s-\sigma_\Gamma(\omega_{WC})-x_\Gamma)^2}{(s-\sigma_\Gamma(\omega_{WC})+x_\Gamma)^2}}_{\upsilon_{\Gamma 1}(s)} \underbrace{\frac{s-\sigma_\Gamma(\omega_{WC})}{(s-\sigma_\Gamma(\omega_{WC}))^2+\sqrt{2}\omega_{WC}(s-\sigma_\Gamma(\omega_{WC}))+\omega^2_{WC}}}_{\upsilon_{\Gamma 2}(s)} \tag{3.168}$$

$$\begin{aligned}\Upsilon_\Gamma(s) = K_\Gamma & \underbrace{\frac{s^2-(2x_\Gamma+2\sigma_\Gamma(\omega_{WC}))s+x^2_\Gamma+2x_\Gamma\sigma_\Gamma(\omega_{WC})+\sigma^2_\Gamma(\omega_{WC})}{s^2+(2x_\Gamma-2\sigma_\Gamma(\omega_{WC}))s+(x^2_\Gamma-2x_\Gamma\sigma_\Gamma(\omega_{WC})+\sigma^2_\Gamma)}}_{\upsilon_{\Gamma 1}(s)} \\ & \cdot \underbrace{\frac{s-\sigma_\Gamma(\omega_{WC})}{(s^2+(-2\sigma_\Gamma(\omega_{WC})+\sqrt{2}\omega_{WC})s+\sigma^2_\Gamma(\omega_{WC})-\sqrt{2}\omega_{WC}\sigma_\Gamma(\omega_{WC})+\omega^2_{WC}}}_{\upsilon_{\Gamma 2}(s)}\end{aligned} \tag{3.169}$$

The transfer function candidate consists of a second order *all-pass* transfer function $\upsilon_{\Gamma 1}(s)$ and the *DT2*-element $\upsilon_{\Gamma 2}(s)$ with the variable $x_\Gamma \in \mathbb{R}^+$ and the gain $K_\Gamma \in \mathbb{R} \setminus \{0\}$.

To fulfill the conditions in Eq. (3.165)-Eq. (3.167), the variables x_Γ and K_Γ can be calculated equivalently to the variable x_Δ and the gain K_Δ explained in Sec. 3.8.2. The derivation is analogous, since the structure of the respective transfer function becomes similar at the worst-case, i.e. at $s = j\omega_{WC}$ for $\Upsilon_\Delta(s)$ and the imaginary axis as borderline, and $s = \sigma_\Gamma(\omega_{WC}) + j\omega_{WC}$ for $\Upsilon_\Gamma(s)$ and the conformity borderline. This becomes visible by evaluating $\Upsilon_\Gamma(s)$ at $s = \sigma_\Gamma(\omega_{WC}) + j\omega_{WC}$:

$$\Upsilon_\Gamma(s = \sigma_\Gamma(\omega_{WC}) + j\omega_{WC}) = K_\Gamma \underbrace{\frac{(j\omega_{WC} - x_\Gamma)^2}{(j\omega_{WC} + x_\Gamma)^2}}_{\upsilon_{\Gamma 1}(j\omega_{WC})} \underbrace{\frac{j\omega_{WC}}{(j\omega_{WC})^2 + \sqrt{2}\omega_{WC}(j\omega_{WC}) + \omega_{WC}^2}}_{\upsilon_{\Gamma 2}(j\omega_{WC})} \tag{3.170}$$

The approach to determine x_Γ is equivalent to the determination of x_Δ (see Eq. (3.152)-(3.161)) by using $\upsilon_{\Gamma 1}(j\omega_{WC})$ instead of $\upsilon_{\Delta 1}(j\omega_{WC})$. The gain K_Γ is calculated by Eq. (3.163) using $\upsilon_{\Gamma 2}(j\omega_{WC})$ and Δ_Γ^{WC}. Thus, the necessary phase and amplitude conditions can be met by determining the variable x_Γ and the gain K_Γ accordingly.

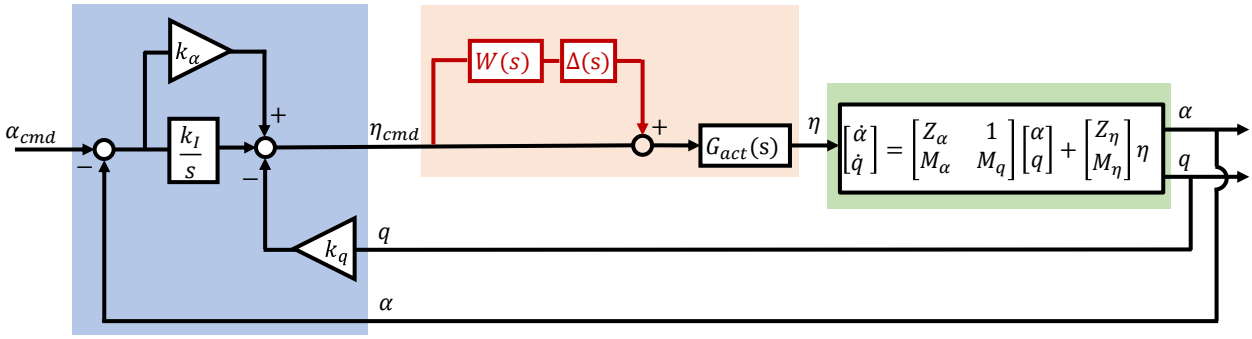


Figure 3.30: Structure of exemplary closed-loop system with one-dimensional multiplicative uncertainty $\Delta(s)$ and its weight $W(s)$

3.8.4 Verification of Worst-Case Uncertainty Transfer Functions

The introduced approach above is verified based on an illustrative aerospace example. The considered closed-loop system encompasses the longitudinal dynamics of a highly-agile aircraft and the respective flight controller, which comprises a pitch damper and an angle-of-attack α feedback. The closed-loop system with the open-loop plant and the controller are illustrated in Fig. 3.30. The parameters of the example are listed in Tab. 3.2. The longitudinal dynamics of the aircraft are represented by the following linear state-space model:

$$\begin{bmatrix}\dot\alpha\\ \dot q\end{bmatrix} = \begin{bmatrix}Z_\alpha & 1\\ M_\alpha & M_q\end{bmatrix}\cdot\begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix}Z_\eta\\ M_\eta\end{bmatrix}\eta \tag{3.171}$$

The considered actuator for this illustrative example is expressed by the *PT2*-element:

$$G_{act}(s) = \frac{\omega_0^2}{s^2 + 2\zeta\omega_0 s + \omega_0^2} \tag{3.172}$$

The used actuator dynamics including the eigenfrequency ω_0 and the damping ζ are based on the identified actuator in Ref. [90, p.14].
The actuator dynamics are affected by uncertainties due to e.g. unconsidered deadtimes, order reduction and/or parameter identification errors. For this example, the uncertainty modeling in Ref. [89] for the respective actuator is considered, which is according to the procedure in Sec. 3.3.7. The actuator uncertainties are modeled as one lumped and complex uncertainty, which is weighted by the following transfer function

$$W(s) = \frac{w_{n1}s^3 + w_{n2}s^2 + w_{n3}s + w_{n4}}{w_{d1}s^3 + w_{d2}s^2 + w_{d3}s + w_{d4}}, \tag{3.173}$$

such that the actuator dynamics affected by the weighted uncertainty $\Delta(s)$ become:

$$G_{act,\Delta} = G_{act,n}\cdot(1 + W(s)\Delta(s)) \tag{3.174}$$

Plant					**Controller**		
Z_α	M_α	M_q	Z_η	M_η	k_α	k_q	k_I
-0.317	-2.11	-0.5645	-0.1036	-4.81	-3.88	-2.15	-1.25

Actuator			
$\omega_{0,act}$	ζ_{act}	T_t	k_{act}
35 / rad/s	0.8	0.035 / s	1
Δ_ω	Δ_ζ	Δ_T	Δ_k
$\pm 20\%$	$\pm 10\%$	$\pm 20\%$	$\pm 10\%$

Uncertainty weight $W(s)$ with consideration of Δ_ω, Δ_ζ, Δ_T, Δ_k							
w_{n1}	w_{n2}	w_{n3}	w_{n4}	w_{d1}	w_{d2}	w_{d3}	w_{d4}
0.004444	0.4524	2.232	2.8	0.00172	0.2173	7.107	28

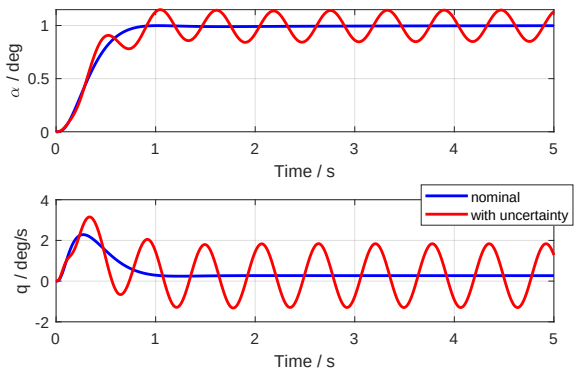
Table 3.2: Parameter values of the open-loop plant, the controller, the actuator $G_{act}(s)$ and its uncertainty weight $W(s)$ (actuator and weight values are based on the work in Ref. [90, 89])

The utilized uncertainty modeling takes into account uncertainties the actuator's time delay θ_T, the gain k_{act}, the damping ζ_{act} and the frequency ω_0. The uncertainties, illustrated in Tab. 3.2, are covered by the weight $W(s)$. The structure and the values of the weighting transfer function $W(s)$ were identified in Ref. [90, 89] following the procedure for lumped uncertainties presented in Sec. 3.3.7.1.
The flight controller sends the desired elevator command η_{cmd} to the actuator, calculated by the controller equation:

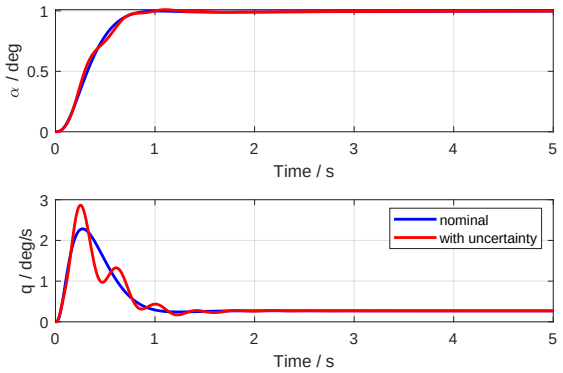
$$\begin{aligned} &\eta_{cmd}(t) = \\ &= k_I \int \big(\alpha_{cmd}(t) - \alpha(t)\big)dt - k_q\, q(t) + k_\alpha\big(\alpha_{cmd}(t) - \alpha(t)\big) \end{aligned} \tag{3.175}$$

The step response of the closed-loop system and the respective pole-zero maps are illustrated in Fig. 3.31(a)-(b) and in 3.31(c)-(d) in blue color. The approaches in Sec. 3.8.2 and Sec. 3.8.3 are subsequently applied to the introduced aerospace example for verification. Therefore, the lower bound μ^{lb} of the presented closed-loop system is calculated under consideration of the complex uncertainty Δ first. Note that the complex uncertainty is multiplied by the uncertainty weight $W(s)$ given in Tab. 3.2. Secondly, using the scaling matrix $\underline{\boldsymbol{U}}$, which is gained by the lower bound calculation (presented inter alia in Sec. 3.7.2), the worst-case uncertainty Δ_{WC} at the worst-case frequency ω_{WC} is computed as complex number. Respect, that the scaling matrix $\underline{\boldsymbol{U}}$ becomes a scalar number, since in the example only one scalar uncertainty is considered. Then, the worst-case transfer function candidates $\Upsilon_\Delta(s)$ and $\Upsilon_\Gamma(s)$ are calculated with respect to the stability borderline and an artificial conformity

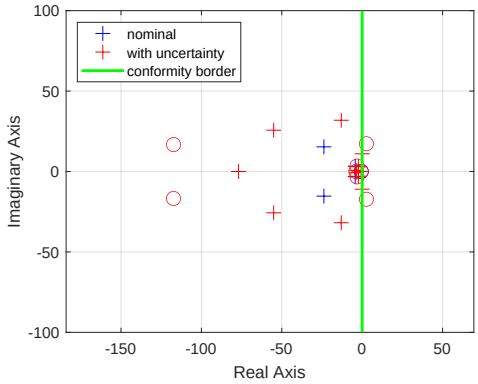
borderline for demonstration. The borderlines are shown in Fig. 3.31(c)-(d) within the pole-zero maps. In the last step, the complex uncertainty Δ is replaced by the worst-case transfer function candidates $\Upsilon_\Delta(s)$ and $\Upsilon_\Gamma(s)$, which are calculated based on the above introduced procedures.
The resulting step responses and pole-zero maps are shown in Fig. 3.31(a)+(c) for the stability borderline as well as in Fig. 3.31(b)+(d) for the considered conformity borderline. In addition, Fig. 3.31(e)-(f) show the Bode diagram of the transfer functions $\Upsilon_\Delta(s)$ and $\Upsilon_\Gamma(s)$ along the absolute value of the conformity borderline $|s_\Gamma|$ together with the worst-case frequency (vertical black line). The resulting step responses and pole-zero maps of the uncertain closed-loop system underline and verify the above introduced approaches. As it can be seen in Fig. 3.32(c)-(d), which shows an enlarged pole-zero map of the closed-loop system, particular complex pole pairs are shifted on the stability borderline or the conformity borderline respectively by the considered transfer functions $\Upsilon_\Delta(s)$ and $\Upsilon_\Gamma(s)$. The imaginary value, at which the particular complex poles hit the considered borderline, is equivalent to the worst-case frequency ω_{wc}. This is underlined by Fig. 3.32, which plots the upper bound μ^{ub} and lower bound μ^{lb} over the frequency. The maximum μ-value occurs at the worst-case frequency ω_{wc} by definition. Since the upper μ-bound is equivalent to the μ-lower bound for both borderlines, for this aerospace example the considered transfer functions Υ_Δ and Υ_Γ represent the worst-case uncertainty, which bring the closed-loop system on the stability borderline or the conformity borderline.
The step responses in Fig. 3.31(a)-(b) show, how the transfer functions Υ_Δ and Υ_Γ affect the closed-loop system in the time domain. The affected closed-loop system in Fig. 3.31(a) oscillates periodically after the step command. The frequency of the oscillations is the worst-case frequency ω_{wc}. This is plausible, since a complex pole pair is located on the stability borderline at this frequency. In contrast, Fig. 3.31(b) shows the step response of the closed-loop system with the transfer function $\Upsilon_\Gamma(s)$. The occuring oscillations are damped and vanish over time. Since a complex pole pair is shifted only on the conformity borderline, only damped oscillations occur. By analysing the corresponding $\mu(\omega)$ along the frequency in Fig. 3.32(c)-(d), it becomes visible that the maximum value of μ considering the stability borderline is smaller than considering the conformity borderline. This is also reasonable, since in total less uncertainty gain is necessary to bring the system on the conformity border than on the stability border. This is the case, since the nominal complex poles of the closed-loop system are closer to the conformity borderline than to the stability borderline. The made observations confirm that the introduced approach gives a worst-case transfer function with regard to the stability borderline or a specified conformity borderline. This is verified by the fact, that the poles of the affected closed-loop system hit the considered borderline at the worst-case frequency ω_{wc}. Thus, it can be said, if this condition is met, a worst-case transfer function with regard to a specific conformity borderline is found.



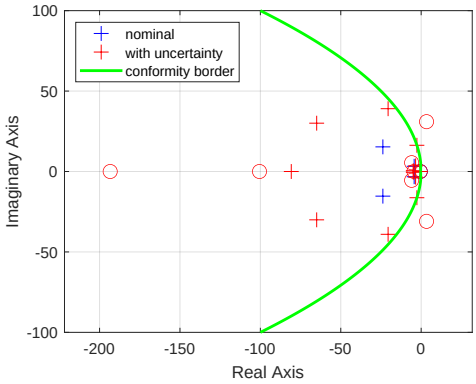
(a) Step response with worst-case transfer function candidate regarding the stability borderline



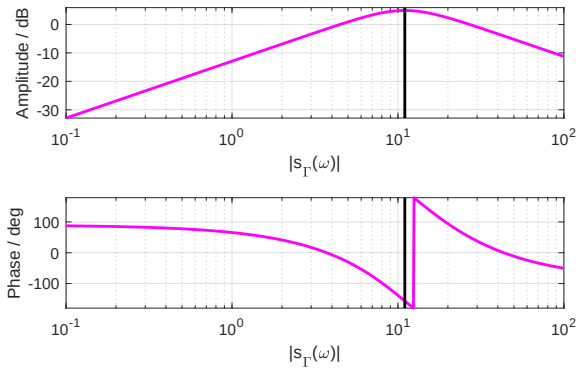
(b) Step response with worst-case transfer function candidate regarding the conformity borderline



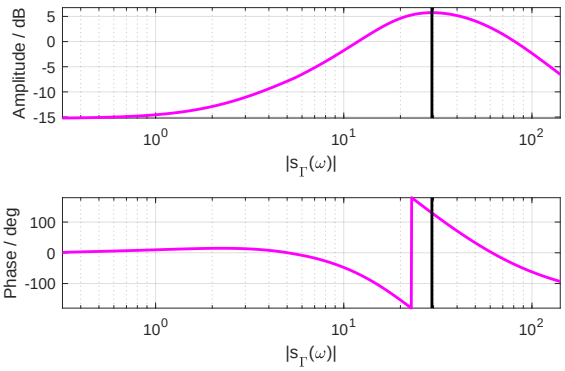
(c) Pole(+)-zero(o) map with worst-case transfer function candidate regarding the stability borderline



(d) Pole(+)-zero(o) map with worst-case transfer function candidate regarding the conformity borderline

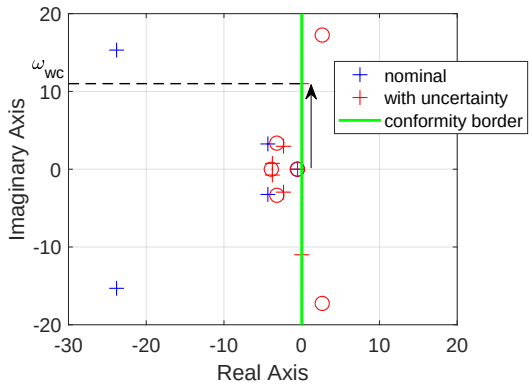


(e) Bode diagram of worst-case transfer function candidate with respect to stability borderline

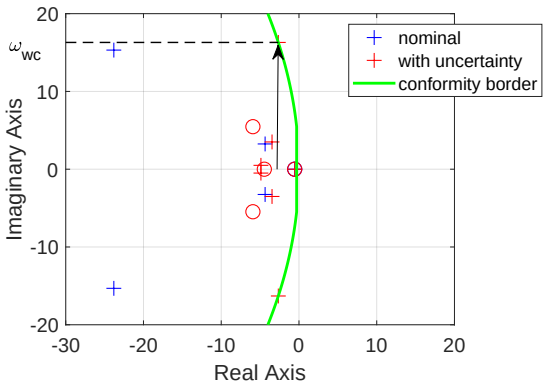


(f) Bode diagram of worst-case transfer function candidate with respect to conformity borderline

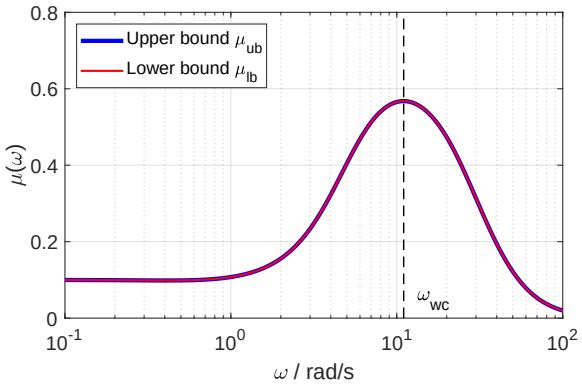
Figure 3.31: Comparison between closed-loop system including worst-case transfer function candidate with regard to the stability borderline (left, imaginary axis) and specified conformity borderline (right)



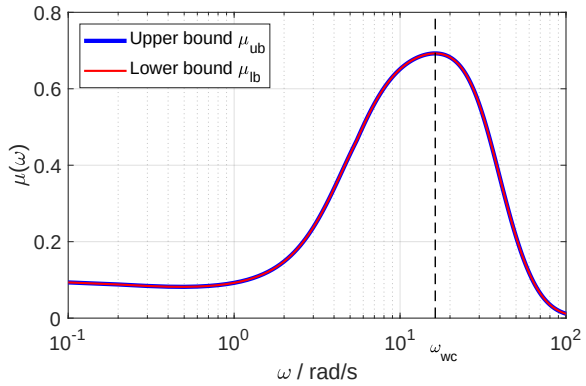
(a) Enlarged pole(+)-zero(o) map with stability borderline



(b) Enlarged pole(+)-zero(o) map with conformity borderline



(c) μ over frequency ω with maximum at ω_{wc} considering the stability borderline



(d) μ over frequency ω with maximum at ω_{wc} considering the conformity borderline

Figure 3.32: Verification plots of worst-case transfer function candidates with regard to the stability borderline (left) and the conformity borderline (right)

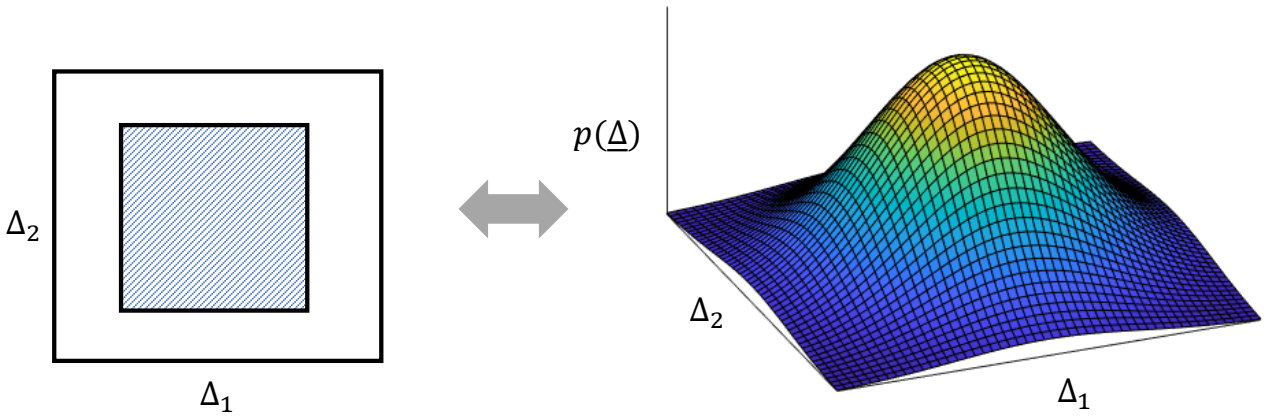


Figure 3.33: Uncertainties with limited intervals and probabilistic uncertainties

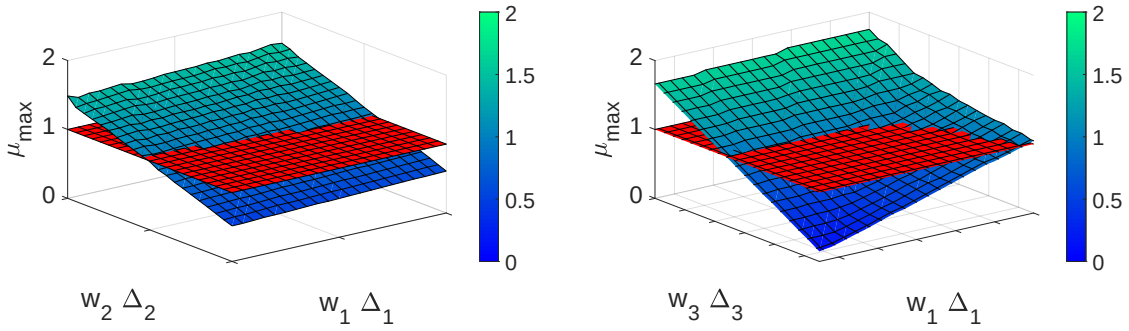
3.9 Probabilistic-μ with Mapped Uncertainties[11]

The μ-analysis, explained in Sec. 3.7, is a popular approach to evaluate the robustness of a LTI closed-loop system under consideration of uncertainties with limited intervals. According to Eq. (3.127), the classical μ-analysis gives the lowest factor, the limited intervals can be multiplied with, such that the system becomes unstable. It does not consider unbounded stochastic uncertainties and does not give a statement about the probability for stability.
In order to take unbounded stochastic uncertainties into account and to calculate a lower bound of the probability for stability, a new probabilistic-μ-approach is proposed. It has been published by the author in Ref. [53]. This section corresponds to *Contribution II.3* of this thesis, which has been introduced in Sec. 1.2.2. The algorithm can be also used to calculate a lower bound of probability to achieve conformity, instead of stability. Therefore, a conformity borderline is specified.

3.9.1 General Idea

The idea of the subsequently proposed algorithm is the exploitation of the μ-algorithm to find the complete set of uncertainties, which do not lead to an unstable closed-loop system, and finally calculate the probability for stability by integrating over the found set of allowed perturbations. Under consideration of the μ-definition, this set is bounded by $\mu = 1$. Finding the exact and complete set of all allowed perturbations for a high-dimensional problem is a nontrivial task, since the set can be a complex body in high-dimensional spaces. This is illustrated in Fig. 3.34 for a simple example. The figure shows the calculated μ upper bound of an exemplary closed-loop system in dependence of normalized and scalar real uncertainties Δ_1, Δ_2 and Δ_3, which are weighted by the scaling factors $w_1, w_2, w_3 \in \mathbb{R}^+$. For each weight,

[11] This section has been published recently in Ref. [53] by the author and extended in this thesis.



(a) μ^{ub} in dependence of $w_1\Delta_1$ and $w_2\Delta_2$

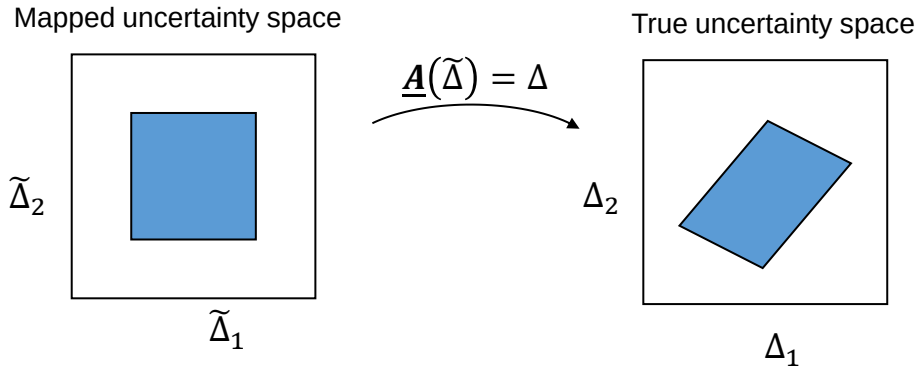
(b) μ^{ub} in dependence of $w_1\Delta_1$ and $w_3\Delta_3$

Figure 3.34: Upper bound μ^{ub} of an exemplary closed-loop system in dependence of weighting factors for the normalized uncertainties $\Delta_1, \Delta_2, \Delta_3$

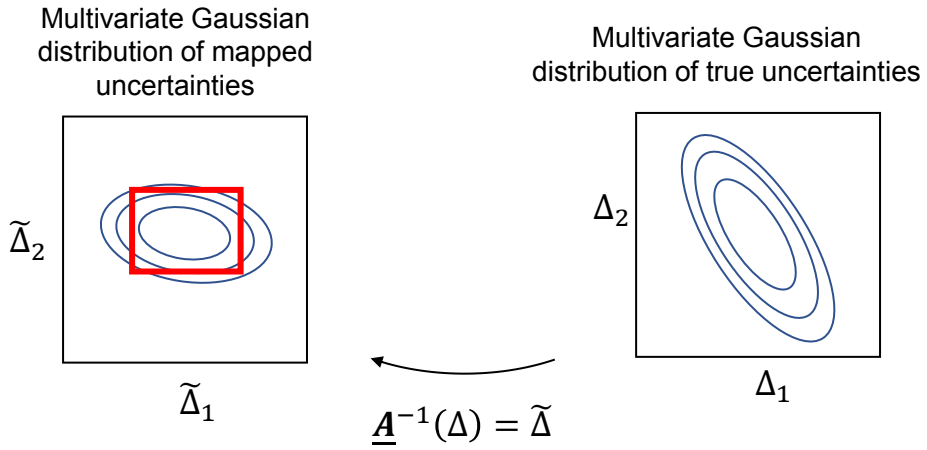
the μ upper bound has been computed. The weights in this case are stretching factors to explore the allowed uncertainty set. It is visible that the $\mu = 1$ boundary is not necessarily linear in dependence of the weighted uncertainties. If the closed-loop system consists of a high number of uncertainties, the set becomes thus very complex. The proposed algorithm tries to find an approximation of such a set by solving a bi-level optimization problem.

3.9.2 The Algorithm

The uncertainties $\underline{\Delta}_1, ..., \underline{\Delta}_n \in \mathbb{R}^{d_i \times d_i}$ are considered as random variables based on a multivariate Gaussian distribution with zero mean and variance $\Sigma \in \mathbb{R}^{N \times N}$. The variable N represents the sum of the uncertainty dimensions $N = \sum_{i=1}^{n} d_i^2$. The uncertainties are not necessarily independent, thus the variance matrix Σ can contain covariance entries. Other probability distributions are not considered in the approach within this thesis. However, the approach can be extended to essential probability distributions in the future. Before the proposed probabilistic-μ approach is explained in detail step-by-step in the next subsections, an overall picture of the algorithm is given first. The algorithm is summarized in Alg. 5. The approach represents a bi-level optimization problem. The outer optimization maximizes the probability for stability with transformation variables as optimization parameters (see Sec. 3.9.6).
The transformation itself converts mapped uncertainties into the true uncertainty space. Since the uncertainty space can be multi-dimensional, the mapped uncertainty space and the true uncertainty space represent hyperspaces. The set of the mapped uncertainties is adapted within the optimization such that the probability is maximized. Possible transformations are shown in Fig. 3.36 (see Sec. 3.9.3).
The inner optimization represents the classical μ-approach taking the mapped uncertainties into consideration, instead of the true uncertainties, i.e. the uncertainties which occur in reality. As shown in Sec. 3.7, μ_Δ tries to find the smallest maximum singular value which makes the system unstable.



(a) Transformation of mapped uncertainty space to true uncertainty space



(b) Transformation of the multivariate Gaussian distribution of the true uncertainties to the mapped uncertainty space. This procedure enables the integration of the probability density function over (hyper-)boxes (red).

Figure 3.35: Illustration of the transformations in the proposed probabilistic-μ approach

After μ^{ub} is calculated, the set of mapped uncertainties can be multiplied by $1/\mu^{ub}$ in order to extend the mapped uncertainty set towards the stability limit, such that μ^{ub} would theoretically be $\mu^{ub} = 1$ in the expanded uncertainty set. Note that the set of mapped uncertainties represents always a hypercube. This is in contrast to the true uncertainty set, which is distorted by the transformation. By the multiplication the maximal possible uncertainty set is considered, such that the system is still stable. Particular transformations adapt the possible set such that the probability is maximized (see Sec. 3.9.4).

The probability for stability within this work is finally calculated by transforming the multivariate Gaussian distribution of the true uncertainties into the mapped uncertainty space. In that way, the probability can be calculated by integrating the new multivariate Gaussian distribution over the hypercube multiplied by $1/\mu_\Delta$, instead of a distorted hyper-body in the true uncertainty space. The mapped uncertainties serve as auxiliary variables to calculate the probability for stability by integration over a non-distorted hypercube. (see Sec. 3.9.5 and Sec. 3.9.6)

Algorithm 5 Probabilstic μ-approach solving a bi-level optimization problem to maximize the probability for stability

Input: Uncertain linear time-invariant closed-loop system as exemplified in Fig. 3.16(a)
Multivariate Gaussian distribution $p\left(\boldsymbol{\Delta}|\boldsymbol{\mu},\underline{\boldsymbol{\Sigma}}\right)$ of the uncertainties $\boldsymbol{\Delta}=[\Delta_1,...,\Delta_N]^T$

Outer optimization: $\max\ P\left(\tilde{\boldsymbol{\Delta}}|\tilde{\boldsymbol{\mu}},\underline{\tilde{\boldsymbol{\Sigma}}}\right)$
with vector $\boldsymbol{p}_0$ and matrix $\underline{\boldsymbol{T}}$ as optimization variables (see Sec. 3.9.6)
- Replace $\boldsymbol{\Delta}$ within closed-loop system with mapped uncertainties by $\boldsymbol{\Delta}=\boldsymbol{p}_0+\underline{\boldsymbol{T}}\,\tilde{\boldsymbol{\Delta}}$ (see Sec. 3.9.3)
- Conduct Linear Fractional Transformation to gain matrix $\underline{\boldsymbol{M}}$
- Calculate upper μ-boundary μ^{ub} (see Sec. 3.9.4)
  Inner optimization: $\mu^{ub}(\underline{\boldsymbol{M}})=\min_{\underline{\boldsymbol{D}}\in\mathbb{D}}\bar{\sigma}(\underline{\boldsymbol{D}}\,\underline{\boldsymbol{M}}\,\underline{\boldsymbol{D}}^{-1})$
  with matrix $\underline{\boldsymbol{D}}$ as optimization variable
- Transform multivariate Gaussian distribution of the true uncertainties $\boldsymbol{\Delta}$ to the mapped uncertainty space (see Eq. (3.182))
- Calculate the probability $P\left(\tilde{\boldsymbol{\Delta}}|\tilde{\boldsymbol{\mu}},\underline{\tilde{\boldsymbol{\Sigma}}}\right)$ by integrating the multivariate Gaussian distribution $p\left(\tilde{\boldsymbol{\Delta}}|\tilde{\boldsymbol{\mu}},\underline{\tilde{\boldsymbol{\Sigma}}}\right)$ over a hypercube with vertices $\pm 1/\mu^{ub}$ in each dimension, utilizing Monte Carlo integration methods. The hypercube represents the allowed mapped uncertainties for stability (see Sec. 3.9.5). The probability $P\left(\tilde{\boldsymbol{\Delta}}|\tilde{\boldsymbol{\mu}},\underline{\tilde{\boldsymbol{\Sigma}}}\right)$ is a lower bound of the probability for stability of the uncertain closed-loop system.

Output: Guaranteed and maximized lower bound of probability for stability of closed-loop system with Gaussian distributed uncertainties

3.9.3 Transformation Mapped Uncertainties

Let $\underline{\tilde{\Delta}}_1,...,\underline{\tilde{\Delta}}_n\in\mathbb{R}^{d_i\times d_i}$ be the mapped uncertainties. These mapped uncertainties have the same dimensions as their true uncertainty counterpart. Equivalent to the classical μ-analysis, an uncertainty interval is specified for each mapped uncertainty for their initial values. The mapped uncertainties are normalized such that $\bar{\sigma}(\underline{\tilde{\boldsymbol{\Delta}}}(s))\leq 1$. Thus, the allowed mapped uncertainties are within the set of norm-bounded uncertainties $\underline{\tilde{\Delta}}_i\in\tilde{\mathbb{B}}_\Delta$ such that:

$$\tilde{\mathbb{B}}_\Delta=\{\underline{\tilde{\boldsymbol{\Delta}}}\in\underline{\underline{\boldsymbol{\Delta}}}:\bar{\sigma}(\underline{\tilde{\boldsymbol{\Delta}}})\leq 1\} \tag{3.176}$$

Geometrically, the set of the mapped uncertainties is a hyper-dimensional cube. This is visualized in Fig. 3.35 on the left side for a two-dimensional example.
The mapped uncertainties are transformed into the true uncertainty space with the following transformation:

$$\boldsymbol{\Delta}=\underline{\boldsymbol{A}}(\tilde{\boldsymbol{\Delta}}) \tag{3.177}$$

Thereby, $\boldsymbol{\Delta} = \mathrm{vec}(\underline{\boldsymbol{\Delta}}) \in \mathbb{R}^{N\times 1}$ and $\tilde{\boldsymbol{\Delta}} = \mathrm{vec}(\underline{\tilde{\boldsymbol{\Delta}}}) \in \mathbb{R}^{N\times 1}$ represent a vectorization of the uncertainty matrices $\underline{\boldsymbol{\Delta}}$ and $\underline{\tilde{\boldsymbol{\Delta}}}$. Within this work, the following transformation is chosen:

$$\begin{aligned}\boldsymbol{\Delta} =& \boldsymbol{p}^0 + \underline{\boldsymbol{T}}\,\tilde{\boldsymbol{\Delta}} \\ =& \begin{bmatrix} p_1^0 \\ p_2^0 \\ \vdots \\ p_N^0 \end{bmatrix} + \begin{bmatrix} t_{11} & t_{12} & \dots & t_{1N} \\ t_{21} & t_{22} & \dots & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ t_{N1} & \dots & \dots & t_{NN} \end{bmatrix} \cdot \begin{bmatrix} \tilde{\Delta}_1 \\ \tilde{\Delta}_2 \\ \vdots \\ \tilde{\Delta}_N \end{bmatrix}\end{aligned} \tag{3.178}$$

The transformation matrix $\underline{\boldsymbol{T}}$ is square and nonsingular. The affine transformation $\underline{\boldsymbol{A}}$ represents a displacement of the mapped uncertainties by the vector $\boldsymbol{p}^0 \in \mathbb{R}^{N\times 1}$ and a distortion of the mapped uncertainty space by the matrix $\underline{\boldsymbol{T}} \in \mathbb{R}^{N\times N}$. Changing the entries of the matrix $\underline{\boldsymbol{T}}$ in a certain way the transformation represents a specific type of distortion, e.g. rotation, stretching or shearing. Illustrative distortions for the two-dimensional case are depicted in Fig. 3.36.

The transformation parameters are optimization parameters within the proposed probabilistic-μ approach to maximize the probability for stability. The transformation in Eq. (3.178) is a suitable transformation scheme since it enables an analytical transformation of the multivariate Gaussian distribution of the true uncertainties in the mapped uncertainty space. This is explained further in Sec. 3.9.5. However, different transformations could be also possible, but this is not in the scope of this work.

3.9.4 μ-Algorithm Exploitation

An initial set $\tilde{\mathbb{B}}$ for the mapped uncertainties and initial values for the transformation parameters are chosen, such that the closed-loop system is stable and the μ-analysis can be applied. Within the first level of the bi-level optimization problem the upper bound of the structured singular value μ_Δ is calculated:

$$\mu_\Delta(\underline{\boldsymbol{M}}) \leq \mu^{ub}(\underline{\boldsymbol{M}}) = \min_{\underline{\boldsymbol{D}}\in\mathbb{D}} \bar{\sigma}(\underline{\boldsymbol{D}}\,\underline{\boldsymbol{M}}\,\underline{\boldsymbol{D}}^{-1}) \tag{3.179}$$

with the matrix $\underline{\boldsymbol{D}}$ which commutes with $\underline{\boldsymbol{\Delta}}$. The optimization of the upper bound is a convex optimization problem for real uncertainties $\underline{\Delta}_i \in \mathbb{R}^{d_i\times d_i}$. After the calculation of $\mu^{ub}(M)$ the set of allowed mapped uncertainties $\tilde{\mathbb{B}}_\Delta$ is extended by the factor of $1/\mu^{ub}$ such that $\mu^{ub} = 1$ results from the extended set:

$$\begin{aligned}\tilde{\mathbb{B}}'_\Delta = \{\underline{\tilde{\boldsymbol{\Delta}}}' \in \underline{\boldsymbol{\Delta}} : \bar{\sigma}(\underline{\tilde{\boldsymbol{\Delta}}}') \leq 1\} \\ \text{with } \underline{\tilde{\boldsymbol{\Delta}}}' = \underline{\tilde{\boldsymbol{\Delta}}} \cdot 1/\mu^{ub}\end{aligned} \tag{3.180}$$

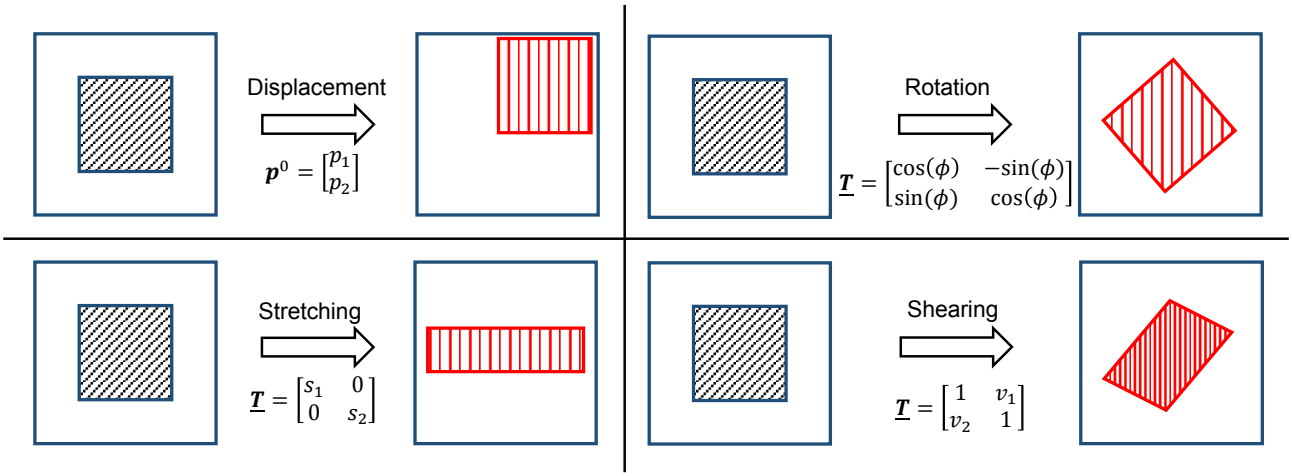


Figure 3.36: Possible mapped uncertainty hyperspace transformations

Note that the mapped uncertainties are normalized in each μ-calculation such that $\bar{\sigma}(\underline{\tilde{\boldsymbol{\Delta}}}') \leq 1$ (see Fig. 3.25). The upper bound is chosen in order to be conservative such that it is guaranteed that the obtained uncertainty set contains only uncertainties which do not make the system unstable. The uncertainty set is stretched by the reciprocal of μ^{ub} such that $\mu^{ub} = 1$. That means, an enlarged, allowed set of mapped uncertainties is found which is close to the stability limit at a specific uncertainty constellation. The goal of the algorithm is to find the allowed set of mapped uncertainties which maximizes the probability by varying the transformation variables. The found uncertainty set within each step is one possible set, in which stability is ensured for all uncertainty combinations. However, it does not necessarily represent the set which achieves the probability maximum.

3.9.5 Probability Calculation

In order to calculate the probability for stability, the continuous probability density function of the true uncertainties must be integrated. The calculated allowed uncertainty set explained in the last section is used to compute the probability by integrating the probability density function over the set. This is challenging for higher-dimensional probability density functions and uncertainty sets, since an analytical solution over a distorted hypercube does not exist for multi-dimensional continuous probability functions yet.
Within this work, it is assumed that the uncertainties are random variables of a multivariate Gaussian distribution. Since the integration of a multivariate Gaussian distribution over a distorted hypercube is still difficult, the multivariate Gaussian distribution is transformed from the true uncertainty space into the mapped uncertainty space.

The multivariate Gaussian distribution is defined as:

$$p(\boldsymbol{\Delta}|\boldsymbol{\mu},\underline{\boldsymbol{\Sigma}}) = \frac{1}{\sqrt{(2\pi)^N|\underline{\boldsymbol{\Sigma}}|}}\exp\left(\frac{1}{2}(\boldsymbol{\Delta}-\boldsymbol{\mu})^T\underline{\boldsymbol{\Sigma}}^{-1}(\boldsymbol{\Delta}-\boldsymbol{\mu})\right) \quad \text{with the random variable } \boldsymbol{\Delta}=[\Delta_1,...,\Delta_N]^T \tag{3.181}$$

The multivariate Gaussian distribution of the mapped uncertainty space is obtained by inserting the transformation Eq. (3.178) in Eq. (3.181) and comparing the coefficients:

$$\begin{aligned} p(\tilde{\boldsymbol{\Delta}}|\tilde{\boldsymbol{\mu}},\underline{\tilde{\boldsymbol{\Sigma}}}) &= \frac{1}{\sqrt{(2\pi)^N|\underline{\tilde{\boldsymbol{\Sigma}}}|}}\exp\left(\frac{1}{2}(\tilde{\boldsymbol{\Delta}}-\tilde{\boldsymbol{\mu}})^T\underline{\tilde{\boldsymbol{\Sigma}}}^{-1}(\tilde{\boldsymbol{\Delta}}-\tilde{\boldsymbol{\mu}})\right) \\ &\text{with the random variable } \tilde{\boldsymbol{\Delta}}=[\tilde{\Delta}_1,...,\tilde{\Delta}_N]^T \\ &\text{with mean } \tilde{\boldsymbol{\mu}}=\underline{\boldsymbol{T}}^{-1}(\boldsymbol{\mu}-\boldsymbol{p}^0) \\ &\text{and variance matrix } \underline{\tilde{\boldsymbol{\Sigma}}}=\underline{\boldsymbol{T}}^{-1}\underline{\boldsymbol{\Sigma}}(\underline{\boldsymbol{T}}^{-1})^T \end{aligned} \tag{3.182}$$

The transformations of the multivariate distribution enables the integration over the hypercube $\tilde{\mathbb{B}}'_\Delta$ which was stretched by the reciprocal of μ^{ub}. Different methods can be used which are able to compute accurately the integral of multivariate distributions over a high-dimensional box (cf. Refs. [101, 102]).

3.9.6 Maximization of Probability

The uncertainty space transformation, the μ-calculation and the uncertainty space extension until the stability limit, and the calculation of the probability for stability represents the objective function which is maximized:

$$\max_{\underline{\boldsymbol{T}},\,\boldsymbol{p}^0} P(\tilde{\boldsymbol{\Delta}}|\tilde{\boldsymbol{\mu}},\underline{\tilde{\boldsymbol{\Sigma}}}) \text{ such that } \underline{\boldsymbol{T}}_l \le \underline{\boldsymbol{T}} \le \underline{\boldsymbol{T}}_u \tag{3.183}$$

The inequality is taken thereby elementwise. The optimization variables are the transformation variables in Eq. (3.178). If the displacement $\boldsymbol{p}^0$ causes already an unstable nominal plant, a negative value is assigned to the objective function of the optimization as penalty. Constraints are also applied to the diagonal entries of the transformation matrix $\underline{\boldsymbol{T}}$ to avoid negative stretching of the hypercube. Note that $P(\tilde{\boldsymbol{\Delta}}|\tilde{\boldsymbol{\mu}},\underline{\tilde{\boldsymbol{\Sigma}}})$ represents the integral of the probability density function in Eq. (3.182) over a hypercube. The hypercube center is located in the origin and its vertices are in distance of $1/\mu^{ub}$ in negative and positive direction in each dimension.

In general, this is a global optimization problem with possible local maxima. Therefore, global search algorithms can be applied to search for the global maximum. Within this work the surrogate optimization solver is used from Refs. [103, 104]. However, the calculated

probability $P\left(\tilde{\Delta}|\tilde{\mu},\tilde{\underline{\Sigma}}\right)$ represents always a lower bound of the probability for stability. As a result, the global optimization only improves the solution, but does not necessarily need to find the global maximum.

3.9.7 Verification

The proposed probabilistic-μ algorithm is illustrated on a closed-loop system with two, three and four open-loop uncertainties. The standard settings in Ref. [104] for the surrogate optimization solver are used. For the three examples the solver was stopped by the default maximum function evaluation limit. Thus, the probability results presented in the following are guaranteed lower limits. The results are compared to the direct Monte Carlo method with 40000 samples.

The closed-system encompasses the longitudinal dynamics of a highly-agile aircraft and the respective flight controller, which comprises a pitch damper and an angle-of-attack α feedback. The closed-loop system with the open-loop plant and the controller are illustrated in Fig. 3.37. The parameters of the example are listed in Tab. 3.3. The open-loop dynamics are expressed by:

$$\begin{bmatrix}\dot{\alpha}\\ \dot{q}\end{bmatrix} = \begin{bmatrix}Z_\alpha & 1\\ M_\alpha & M_q\end{bmatrix} \cdot \begin{bmatrix}\alpha\\ q\end{bmatrix} + \begin{bmatrix}Z_\eta\\ M_\eta\end{bmatrix} \eta \tag{3.184}$$

Actuator dynamics of the elevator η are considered by the transfer function

$$G_{act}(s) = \frac{1}{T_a s + 1} \exp\left(-sT_t\right), \tag{3.185}$$

with the time constant T_a and deadtime T_t. The controller comprises a proportional gain k_q for the pitch damper and a proportional gain k_α and integral gain k_I for the control of the angle-of-attack α:

$$\eta_{cmd}(t) = k_I \int \left(\alpha_{cmd}(t) - \alpha(t)\right)dt - k_q\, q(t) + k_\alpha\left(\alpha_{cmd}(t) - \alpha(t)\right) \tag{3.186}$$

Z_α	M_α	M_q	Z_η	M_η
-0.317	-2.11	-0.5645	-0.1036	-4.81
T_a	T_t	k_α	k_q	k_I
0.05	0.05	-3.88	-2.15	-1.25

Table 3.3: Parameter values of the open-loop plant, the actuator and the controller

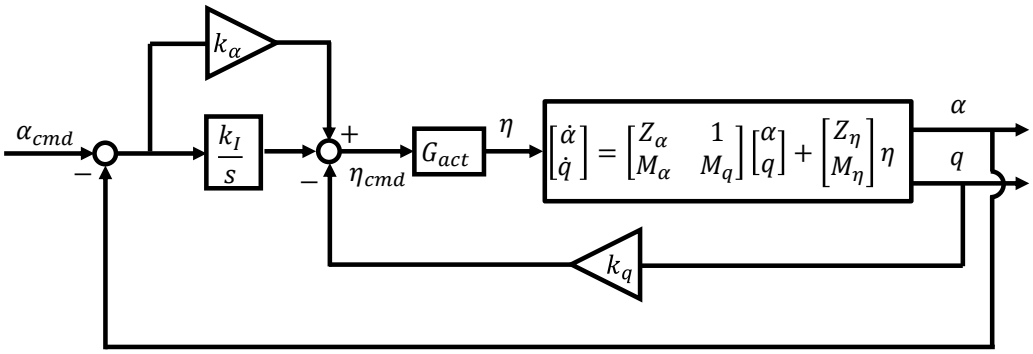


Figure 3.37: Structure of the closed-loop system

3.9.7.1 Two-Dimensional Uncertainty Space

Two uncertainties are introduced in the open-loop plant:

$$\overline{M}_\alpha = M_\alpha(1+\Delta_{M_\alpha}) \tag{3.187}$$

$$\overline{M}_q = M_q(1+\Delta_{M_q}) \tag{3.188}$$

The uncertainties are random variables from a multivariate distribution with the mean $\boldsymbol{\mu} = \mathbf{0}$ and the variance matrix $\underline{\boldsymbol{\Sigma}}$.

The results of the example are visualized in Fig. 3.38. The three subplots show in red color the uncertainty set of the true uncertainties Δ_{M_α} and Δ_{M_q}, which cause an unstable closed-loop system. The red area can be determined by calculating the eigenvalues of the closed-loop system for each discrete uncertainty combination. If one of the closed-loop system matrix eigenvalues is greater than zero, the uncertainty constellation is considered as unstable. Since the example comprises a two-dimensional uncertainty space, the unstable uncertainty space can be illustrated. Beside the red area, the three quadrilaterals in each subplot show true uncertainty sets for which stability is guaranteed. The quadrilaterals are calculated by the first level of the bi-level optimization and the μ-exploitation (see Sec. 3.9.4). Fig. 3.38 specifies the respective transformation variables $\boldsymbol{p}^0$ and $\underline{\boldsymbol{T}}$, and the corresponding probability for stability. Each quadrilateral illustrates the bounds of the probability density function integration. The three subplots represent therefore possible function evaluations of the second level bi-level optimization. The quadrilaterals touch almost the unstable uncertainty area, which is caused by the multiplication of the uncertainty set by the reciprocal of μ^{ub} (see Sec. 3.9.4). The space is thus extended such that the integration bounds are as large as possible. The right subplot in Fig. 3.38 shows the solution of the second level optimization in Eq. (3.183). The optimization adapts the transformation variables such that the probability for stability is maximized. The presented quadrilaterals illustrate the function evaluations of the optimization. The quadrilateral is displaced, rotated and sheared such that it is

Example	Variance	Novel μ	Direct MC
2-dim	$\underline{\Sigma} = \mathbf{I}^{2\times 2} \cdot 5$	99.87447%	99.93000%
3-dim	$\underline{\Sigma} = \mathbf{I}^{3\times 3} \cdot 3$	99.36881%	99.99500%
4-dim	$\underline{\Sigma} = \mathbf{I}^{4\times 4} \cdot 0.2$	98.63178%	98.81500%

Table 3.4: Probability comparison between the novel probabilistic-μ approach and the direct Monte Carlo (MC) method

adapted to the boundary shape of the unstable area. The maximized probability is compared to the direct Monte Carlo method in Tab. 3.4. The probability of the novel probabilistic-μ approach is close to the results of direct Monte Carlo method, which verifies the validity of the proposed algorithm. However, in contrast to the direct Monte Carlo method, the result of the probabilistic-μ algorithm represents a guaranteed probability for stability.

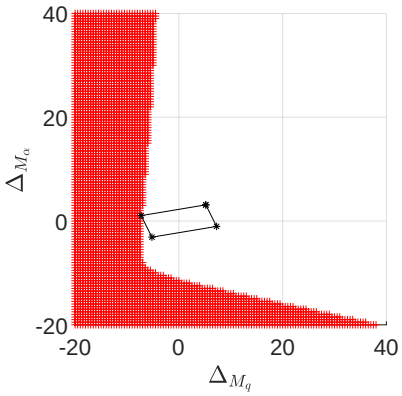
3.9.7.2 Three- and Four-Dimensional Uncertainty Space

In addition to the uncertainties in Eq. (3.187)-Eq. (3.188), the following third uncertainty and fourth uncertainty are introduced into the example:

$$\overline{Z}_\alpha = Z_\alpha(1+\Delta_{Z_\alpha}) \tag{3.189}$$

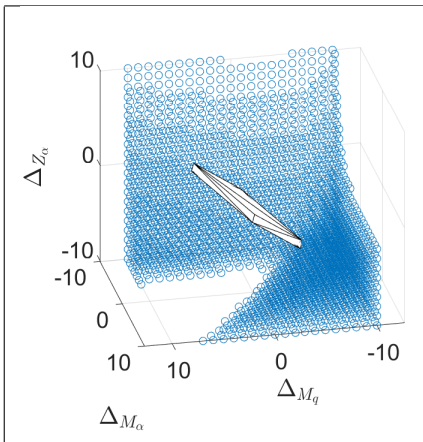
$$\overline{M}_\eta = M_\eta(1+\Delta_{M_\eta}) \tag{3.190}$$

Analogously to the two-dimensional case, illustrative function evaluations of the probability optimization for the three-dimensional uncertainty set are visualized in Fig. 3.39. The uncertainty set which causes an unstable closed-loop system is visualized as blue circulars. The uncertainty set, which can guarantee stability, becomes a three-dimensional, distorted cube for the three-dimensional uncertainty space. In addition to the three-dimensional case, a fourth uncertainty is introduced to the example. As Tab. 3.4 underlines, the algorithm is also feasible for three- or four-dimensional uncertainty sets.



Low probability	Medium probability	Maximized probability
$P(\tilde{\boldsymbol{\Delta}})$ = 67.54894%	$P(\tilde{\boldsymbol{\Delta}})$ = 91.55035%	$P(\tilde{\boldsymbol{\Delta}})$ = 99.87447%
$\boldsymbol{p}^0 = [0,0]^T$	$\boldsymbol{p}^0 = [16,8]^T$	$\boldsymbol{p}^0 = [17.60, 10.96]^T$
$\underline{\boldsymbol{T}} = \begin{bmatrix} 3 & -0.5 \\ 0.5 & 1 \end{bmatrix}$	$\underline{\boldsymbol{T}} = \begin{bmatrix} 1 & 0 \\ 0.5 & 1 \end{bmatrix}$	$\underline{\boldsymbol{T}} = \begin{bmatrix} 27.13 & 2.12 \\ -7.20 & 28.03 \end{bmatrix}$

Figure 3.38: Two-dimensional uncertainty space with three different guaranteed-stable uncertainty sets



Low probability	Medium probability	Maximized probability
$P(\tilde{\boldsymbol{\Delta}})$ = 5.56597%	$P(\tilde{\boldsymbol{\Delta}})$ = 69.46846%	$P(\tilde{\boldsymbol{\Delta}})$ = 99.36881%
$\boldsymbol{p}^0 = [0,0,0]^T$	$\boldsymbol{p}^0 = [2,3,1]^T$	$\boldsymbol{p}^0 = [0.59, 0.22, 0.71]^T$
$\underline{\boldsymbol{T}} = \begin{bmatrix} 1 & 0.3 & 1 \\ 0.1 & 1 & 1 \\ 1 & 1 & 2 \end{bmatrix}$	$\underline{\boldsymbol{T}} = \begin{bmatrix} 1.5 & 0 & 1 \\ 0.1 & 2 & 1 \\ 0 & 0 & 1 \end{bmatrix}$	$\underline{\boldsymbol{T}} = \begin{bmatrix} 25.23 & 0.44 & 0.07 \\ 0.30 & 25.04 & 0.14 \\ 0.11 & 0.04 & 25.12 \end{bmatrix}$

Figure 3.39: Three-dimensional uncertainty space with three different guaranteed-stable uncertainty sets

3.9.8 Conclusion and Outlook

A novel probabilistic-μ approach has been proposed within this section, which is able to handle uncertainties with multivariate Gaussian distributions of LTI systems. This contrasts to existing approaches, which can handle only truncated-normally or uniformly distributed uncertainties and are infeasible for higher-dimensional uncertainty sets. Compared to Monte Carlo techniques, the represented approach gives a guaranteed lower limit on the probability for stability. The proposed approach shall provide the basis for the novel probabilistic-μ approach as several ideas exist to augment the algorithm. The approach can be extended to arbitrary probability distributions by including advanced integration methods of high-dimensional probability density functions. Different transformation methods might lead to even less conservative probability results. The found uncertainty set of the proposed approach could also be used as initial set for existing sampling methods (see Sec. 3.12.3.2). The application, results and challenges of the presented algorithm to a high-dimensional uncertainty problem of an eVTOL are discussed in Sec. 3.12. Potential ideas for future work are in addition elaborated at the end of Sec. 3.12 in more detail.

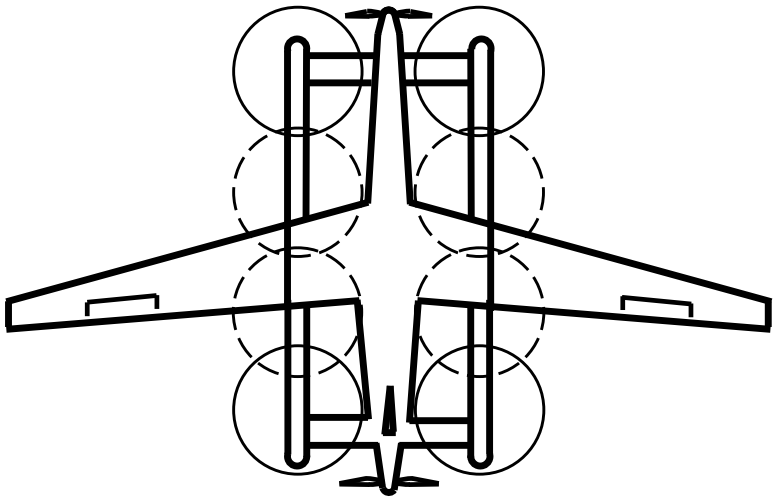


Figure 3.40: Sketch of eVTOL demonstrator aircraft

3.10 Application of Robustness Metrics on an eVTOL Demonstrator

The following section demonstrates the applicability of the presented robustness measures on novel aerial vehicle configurations with multiple and over-actuated control effectors. The results can set the basis for robustness measurements of future aerial vehicles such as transition vehicles and air taxis. Therefore, the elaborated and introduced robustness metrics including the μ-analysis and the probabilistic-μ approach are applied on a real-live eVTOL aircraft demonstrator, which is currently under development. The *Institute of Flight System Dynamics* of the *Technical University of Munich* designs and develops the flight control system of the eVTOL demonstrator. The subject of this section corresponds to Contribution II.3-II.6, introduced in Sec. 1.2.2.

The eVTOL demonstrator configuration, the system architecture and the development process of the flight control system is shortly detailed in the next section.
Subsequently, the considered closed-loop system and its linear representation are elaborated. The linear representations of the respective closed-loop components are essential for the robustness analysis. The nonlinear simulation model including sensor and actuator model and their linear representations are introduced in Sec. 3.10.2. The emergency control structure and its linear representation are explained in Sec. 3.10.3. The focus of the two sections is set on necessary details for building the linear closed-loop system, which is required for the robustness analysis. The application of the robustness metrics is finally elaborated in Sec. 3.10.5-Sec. 3.12 including the advanced μ_Γ analysis and the proposed probabilistic-μ approach.

3.10.1 eVTOL Demonstrator Configuration, System Architecture and Development Process

The eVTOL aircraft demonstrator is a transition vehicle designed for hover, transition and wingborne flight and is illustrated in Fig. 3.40. The eVTOL aircraft has take-off mass of about 250 kg and a wingspan of about 7 m. It is actuated by eight equivalent hover-propellers attached to two hover-beams for hover flight. The aircraft is propelled by one pusher and puller propeller for forward flight. Classical control surfaces such as elevators, ailerons and rudders are available for wingborne flight, in which the main aerodynamic lift is produced by the wings.
The aircraft is equipped with an Attitude Heading Reference System (AHRS) and an Inertial Navigation System (INS) to sense and calculate the necessary feedback signals for the flight control system. The eVTOL demonstrator is controlled by remote pilots during the first flights. The redundant flight control system of the aerial vehicle consists of a nominal and emergency flight controller. The emergency flight controller comprises a flight control law active in the hover plus transition phase and a flight control law for wing-borne flight. The hover control law gives commands only to the hover-propellers while the control surfaces are set to neutral. The emergency flight controller for the hover and transition flight phase is selected for the subsequent robustness analysis. The flight control software and the respective flight control laws for the eVTOL demonstrator are developed by a model-based approach using *MATLAB, Simulink and Stateflow*.

3.10.2 Nonlinear Simulation Model and its Linear Representation

For the model-based design approach a nonlinear simulation model has been developed and implemented at the *Institute of Flight System Dynamics* at the *Technical University of Munich*. It consists of the flight dynamic model of the eVTOL demonstrator, the dynamics of the actuators, the sensor systems, and the communication delays between the hardware components. For the subsequent derivation of the nonlinear simulation model the nomenclature introduced in Ref. [105] is applied.

Nonlinear Flight Dynamic Model: The nonlinear flight dynamic model comprises the six-degrees-of-freedom rigid-body *equations of motion* considering the forces and moments of the propulsion, the aerodynamics and the gravitation. The forces and moments are considered as external forces in the body-fixed coordinate frame B for the *equations of motion* of the aerial vehicle. The respective *equations of motion* are derived in Ref. [105]. Subsequently, the important details of the considered flight dynamic model are elaborated.

The propulsion force for each propeller c is calculated in a specific propulsion coordinate system P_c, which represents a sideslip-free coordinate frame, at the reference point R_c:

$$\left(\boldsymbol{F}_{P_{Total}}\right)_B = \sum_{c}^{N_c} \underline{\boldsymbol{M}}_{BP_c} \left(\boldsymbol{F}_{P_c}^{R_c}\right)_{P_c} \quad \text{with} \quad \boldsymbol{F}_{P_c}^{R_c} = \frac{D^4\ \rho\ (\omega_c^{Prot})^2}{(2\pi)^2} \cdot \begin{bmatrix} c_{P,x}\left(\omega_c^{Prot}, \boldsymbol{V}_A^{P_c}\right) \\ c_{P,y}\left(\omega_c^{Prot}, \boldsymbol{V}_A^{P_c}\right) \\ c_{P,z}\left(\omega_c^{Prot}, \boldsymbol{V}_A^{P_c}\right) \end{bmatrix} \tag{3.191}$$

The propulsion force depends on the propeller diameter D, the air density ρ and the derivatives $[c_{P,x},\ c_{P,y},\ c_{P,z}]^T$, which are functions of the rotational speed ω_c^{Prot} of the propeller c and the inflow velocity vector $\boldsymbol{V}_A^{P_c}$ at the respective reference point P_c of each propeller c. The propulsion moment for each propeller c is calculated by:

$$\left(\boldsymbol{M}_{P_{Total}}\right)_B = \sum_{c}^{N_c} \left(\boldsymbol{M}_{P_c}^{R_c}\right)_B \quad \text{with} \quad \left(\boldsymbol{M}_{P_c}^{R_c}\right)_{P_c} = \frac{D^5\ \rho\ (\omega_c^{Prot})^2}{(2\pi)^2} \cdot \begin{bmatrix} c_l\left(\omega_c^{Prot}, \boldsymbol{V}_A^{P_c}\right) \\ c_m\left(\omega_c^{Prot}, \boldsymbol{V}_A^{P_c}\right) \\ c_n\left(\omega_c^{Prot}, \boldsymbol{V}_A^{P_c}\right) \end{bmatrix} \tag{3.192}$$

Equivalently, the derivatives $[c_{P,l},\ c_{P,m},\ c_{P,n}]^T$ are functions of the rotational speed $\omega_{z,c}^{Prot}$ and the inflow velocity vector $\boldsymbol{V}_A^{P_c}$ at the respective reference point P_c at each propeller c. Note that, since in the subsequent studies the hover-propellers are relevant, the turn rates ω_c^{Prot} of the eight hover propellers are assembled in the vector

$$\boldsymbol{\omega} = [\omega_{R01}, \omega_{R02}, \omega_{R03}, \omega_{R04}, \omega_{L01}, \omega_{L02}, \omega_{L03}, \omega_{L04}]^T, \tag{3.193}$$

encompassing the turn rates of the right (R) and left (L) hover-propellers. The subscript c is thereby replaced by the respective position of the propeller (e.g. R01,...,L04). The number "01" indicates the propellers at the front of the aerial vehicle, "04" represents the propellers at the back of the aerial vehicle.

The aerodynamic forces and moments are represented by a multi-compartment aerodynamic model. Every aerodynamic surface (i.e. the wing, the front and back horizontal stabilizer, the lower and upper vertical stabilizer, the fuselage) is divided into different aerodynamic sections. For each of these sections the aerodynamic forces and moments are calculated as follows:

$$\left(\boldsymbol{F}_{A_{Total}}\right)_B = \sum_{sec}^{N_{sec}} \underline{\boldsymbol{M}}_{BA_{sec}} \left(\boldsymbol{F}_{A_{sec}}^{R_{sec}}\right)_{A_{sec}} \quad \text{with} \quad \boldsymbol{F}_{A_{sec}}^{R_{sec}} = \frac{1}{2}\rho \left(\boldsymbol{V}_{A_{sec}}^{R_{sec}}\right)^2 S_{ref_{sec}} \cdot \begin{bmatrix} c_{D_{sec}}(\alpha_{sec}) \\ 0 \\ c_{L_{sec}}(\alpha_{A_{sec}}) \end{bmatrix} \tag{3.194}$$

The aerodynamic derivatives $C_{D_{sec}}$ and $C_{L_{sec}}$ for each aerodynamic section are functions of the respective angle of attack α_{sec} at the reference point R_{sec} of the section. The aerodynamic

moments are calculated by:

$$\left(\boldsymbol{M}_{P_{Total}}\right)_B = \sum_{c}^{N_c} \left(\boldsymbol{M}_{A_{sec}}^{R_{sec}}\right)_B \quad \text{with} \quad \boldsymbol{M}_{A_{sec}}^{R_{sec}} = \frac{1}{2}\rho \left(\boldsymbol{V}_{A_{sec}}^{R_{sec}}\right)^2 S_{ref_{sec}} \cdot \begin{bmatrix} c_{l_{sec}}(\alpha_{sec}, \beta_{sec}, p) \\ c_{m_{sec}}(\alpha_{sec}, \beta_{sec}, q) \\ c_{n_{sec}}(\alpha_{sec}, \beta_{sec}, r) \end{bmatrix} \tag{3.195}$$

Equivalently, the aerodynamic derivatives $[c_{l_{sec}},\ c_{m_{sec}},\ c_{n_{sec}}]^T$ for each aerodynamic surface section are functions of the respective angle of attack α_{sec} and angle of sideslip β_{sec} at the reference point R_{sec} of the section. Thereby, it is important to note, that the functions of the aerodynamic derivatives cover the range $\alpha_{sec} \in [-\pi, \pi]$ since the eVTOL demonstrator operates as transition vehicle in this angle-of-attack range. The respective inflow velocity $\boldsymbol{V}_{A_{sec}}^{R_{sec}}$ at reference point R_{sec} is considered for each aerodynamic section. The weight force is calculated by

$$\boldsymbol{F}_G = \underline{\boldsymbol{M}}_{BO} \cdot m_{Total} \cdot \begin{bmatrix} 0 \\ 0 \\ g \end{bmatrix} \tag{3.196}$$

with the gravitational acceleration constant g and the mass m_{Total} of the aerial vehicle.
The forces and moments are transformed into the B-frame considering the transformation matrices $\underline{\boldsymbol{M}}_{BP_c}$ for each propeller c, $\underline{\boldsymbol{M}}_{BA_{sec}}$ for each aerodynamic section sec and $\underline{\boldsymbol{M}}_{BO}$.

Linear Representation of Nonlinear Flight Dynamic Model: For the application of the above introduced robustness criteria the nonlinear flight dynamic model is trimmed and linearized. The following plant state vector $\boldsymbol{x}$ and input vector $\boldsymbol{u}$ are therefore considered:

$$\boldsymbol{x} = [\lambda^G_{WGS84}, \phi^G_{WGS84}, h^G_{WGS84}, (u^G_A)_B, (v^G_A)_B, (w^G_A)_B, \phi, \theta, \psi, p, q, r]^T \tag{3.197}$$

$$\boldsymbol{\omega} = [\omega_{R01}, \omega_{R02}, \omega_{R03}, \omega_{R04}, \omega_{L01}, \omega_{L02}, \omega_{L03}, \omega_{L04}]^T \tag{3.198}$$

The first three entries of the state vector $\boldsymbol{x}$ represent the position of the aerial vehicle in the WGS84 coordinate frame. The state vector $\boldsymbol{x}$ also encompasses the three-dimensional aerodynamic velocities of the vehicles reference point (G, i.e. the center of gravity) in the body-frame B, the roll angle ϕ, the pitch angle θ and the heading angle ψ, as well as the roll rate p, the pitch rate q and the yaw rate r (based on the notations in Ref. [105]). The input vector $\boldsymbol{\omega}$ comprises the rotational speed of the left (L) and right (R) hover-propellers. For the subsequent robustness analysis of the emergency flight control law other control effectors such as the pusher/puller propeller or the control surfaces are not relevant and thus not considered in the input vector $\boldsymbol{\omega}$.
Nonlinear programming solvers are applied to find operational points (i.e. trim points) of the nonlinear flight dynamic model. The available trim- and linearization tool chain, developed at the *Institute of Flight System Dynamics* at the *Technical University of Munich*, is therefore utilized. Note that the subsequent C-coordinate system is used, which represents the O-frame (i.e. north-east-down frame), but turned around the z-axis in the direction of the aircraft. The following trim template and cost function are set up:

- Flight parameter:
  - The aerodynamic forward velocity $(u^G_A)_C$
  - The aerodynamic sideward velocity $(v^G_A)_C$
  - The aerodynamic vertical velocity $(w^G_A)_C$
  - The geodetic height h^G_{WGS84}
- Equality constraints:
  - Angular accelerations $\dot{p}, \dot{q}, \dot{r} = 0$
  - Translational accelerations $\dot{u}, \dot{v}, \dot{w} = 0$
- Solver variables:
  - Bank angle ϕ and pitch angle θ
  - The vector of the eight hover propeller turn rates $\boldsymbol{\omega} = [\omega_{L01}, \omega_{L02}, \omega_{L03}, \omega_{L04}, \omega_{R01}, \omega_{R02}, \omega_{R03}, \omega_{R04}]^T$
- Cost function:
  - The cost function $\min_{\boldsymbol{u}} \boldsymbol{u}^T\boldsymbol{u}$ with $\boldsymbol{u} = [\omega^2_{L01}, \omega^2_{L02}, \omega^2_{L03}, \omega^2_{L04}, \omega^2_{R01}, \omega^2_{R02}, \omega^2_{R03}, \omega^2_{R04}]^T$ is minimized under considerations of the equality constraints and the flight param-

eter. This cost function represents the used control allocation approach, which minimizes $\boldsymbol{u}^{\boldsymbol{T}}\boldsymbol{u}$. The control allocation is embedded within the incremental nonlinear dynamic inversion (INDI) control approach, which is used in the considered flight controller. The control allocation is further explained in Sec. 3.10.3.

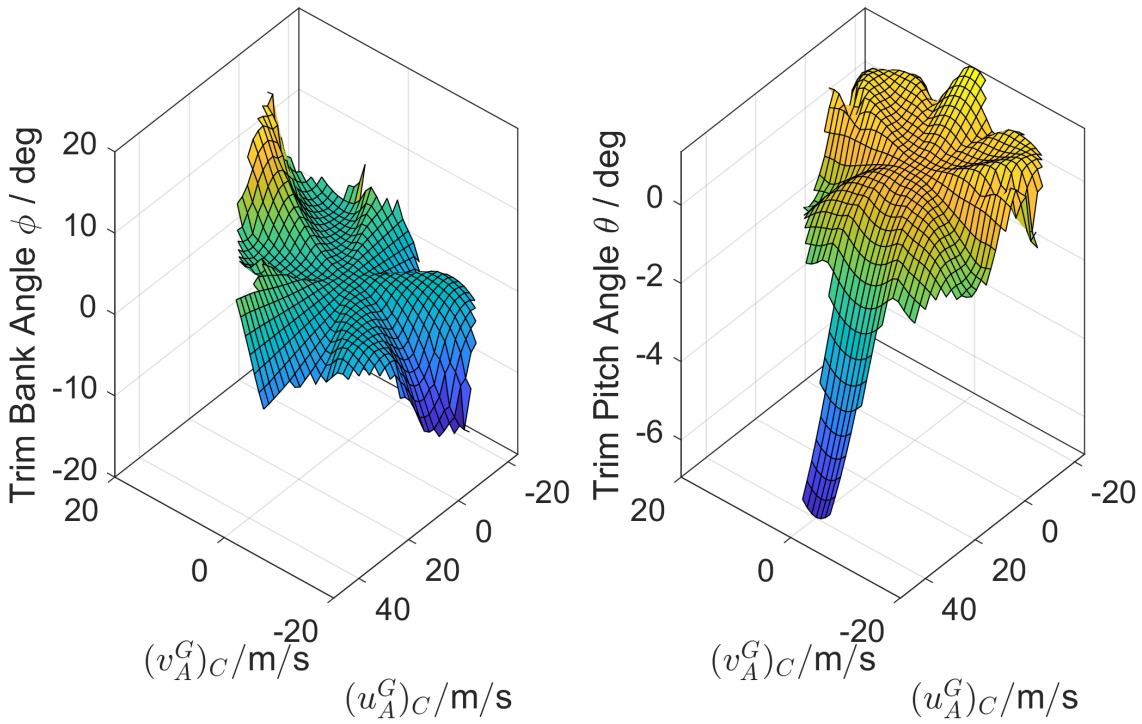


Figure 3.41: Calculated trim angles over the sidewards velocity $(v_A^G)_C$ and forward velocity $(u_A^G)_C$ grid

The nonlinear flight dynamic model is trimmed and linearized over the velocity envelope of the aerial vehicle by varying the aerodynamic forward velocity $(u_A^G)_C$, the aerodynamic sideward velocity $(v_A^G)_C$ and the aerodynamic vertical velocity $(w_A^G)_C$. Each trim point is defined by the calculated trim state vector $\boldsymbol{x}_0$ and the trim input vector $\boldsymbol{\omega}_0$. Only trim solutions are taken into account, which fulfill the equality constraints under consideration of propeller speed limitations.
For $(w_A^G)_C = 0$ the calculated trim bank angle ϕ and pitch angle θ are visualized over the forward and sideward velocity in Fig. 3.41.
Linear time-invariant(LTI) state-space models are gained for each found trim point:

$$\Delta\dot{\boldsymbol{x}} = \underline{\boldsymbol{A}}\Delta\boldsymbol{x} + \underline{\boldsymbol{B}}\Delta\boldsymbol{\omega} \tag{3.199}$$

$$\Delta\boldsymbol{y} = \underline{\boldsymbol{C}}\Delta\boldsymbol{x} + \underline{\boldsymbol{D}}\Delta\boldsymbol{\omega} \tag{3.200}$$

with the system matrix $\underline{\boldsymbol{A}}$, the input matrix $\underline{\boldsymbol{B}}$, the output matrix $\underline{\boldsymbol{C}}$ and the feedthrough

matrix $\underline{\boldsymbol{D}}$. The state vector $\Delta\boldsymbol{x}$, input vector $\Delta\boldsymbol{\omega}$ and output vector $\Delta\boldsymbol{y}$ are defined as:

$$\Delta\boldsymbol{x} = [\Delta(u_A^G)_B, \Delta(v_A^G)_B, \Delta(w_A^G)_B, \Delta\phi, \Delta\theta, \Delta\psi, \Delta p, \Delta q, \Delta r]^T \tag{3.201}$$

$$\Delta\boldsymbol{\omega} = [\Delta\omega_{R01}, \Delta\omega_{R02}, \Delta\omega_{R03}, \Delta\omega_{R04}, \Delta\omega_{L01}, \Delta\omega_{L02}, \Delta\omega_{L03}, \Delta\omega_{L04}]^T \tag{3.202}$$

$$\Delta\boldsymbol{y} = [\Delta(u_A^G)_B, \Delta(v_A^G)_B, \Delta(w_A^G)_B, \Delta\phi, \Delta\theta, \Delta\psi, \Delta p, \Delta q, \Delta r]^T \tag{3.203}$$

The greek letter Δ indicates in this context the deviation of the state/input vector to the trim point defined by $\boldsymbol{x}_0$ and $\boldsymbol{\omega}_0$. The Δ is neglected for readability in the following linear robustness analysis. The subscripts $R01, ..., R04, L01, ...L04$ in the input vector $\boldsymbol{\omega}$ indicate the position of the propeller, e.g. ω_{L01} is the turn rate of the left (L), front propeller (01), ω_{R04} is the turn rate of the right (R), back propeller (04).
The system matrix $\underline{\boldsymbol{A}}$ and the input matrix $\underline{\boldsymbol{B}}$ are specified with the following entries:

$$\underline{\boldsymbol{A}} = \begin{bmatrix} X_u & X_v & X_w & X_\phi & X_\theta & 0 & X_p & X_q & X_r \\ Y_u & Y_v & Y_w & Y_\phi & Y_\theta & 0 & Y_p & Y_q & Y_r \\ Z_u & Z_v & Z_w & Z_\phi & Z_\theta & 0 & Z_p & Z_q & Z_r \\ 0 & 0 & 0 & 0 & 0 & 0 & \phi_p & \phi_q & \phi_r \\ 0 & 0 & 0 & 0 & 0 & 0 & \theta_p & \theta_q & \theta_r \\ 0 & 0 & 0 & 0 & 0 & 0 & \psi_p & \psi_q & \psi_r \\ L_u & L_v & L_w & L_\phi & L_\theta & 0 & L_p & L_q & L_r \\ M_u & M_v & M_w & M_\phi & M_\theta & 0 & M_p & M_q & M_r \\ N_u & N_v & N_w & N_\phi & N_\theta & 0 & N_p & N_q & N_r \end{bmatrix} \tag{3.204}$$

$$\underline{\boldsymbol{B}} = \begin{bmatrix} X_{R01} & X_{R02} & X_{R03} & X_{R04} & X_{L01} & X_{L02} & X_{L03} & X_{L04} \\ Y_{R01} & Y_{R02} & Y_{R03} & Y_{R04} & Y_{L01} & Y_{L02} & Y_{L03} & Y_{L04} \\ Z_{R01} & Z_{R02} & Z_{R03} & Z_{R04} & Z_{L01} & Z_{L02} & Z_{L03} & Z_{L04} \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ L_{R01} & L_{R02} & L_{R03} & L_{R04} & L_{L01} & L_{L02} & L_{L03} & L_{L04} \\ M_{R01} & M_{R02} & M_{R03} & M_{R04} & M_{L01} & M_{L02} & M_{L03} & M_{L04} \\ N_{R01} & N_{R02} & N_{R03} & N_{R04} & N_{L01} & N_{L02} & N_{L03} & N_{L04} \end{bmatrix} \tag{3.205}$$

The entries are based on the naming convention in Ref. [105]. The non-zero entries in the fourth until the sixth row of the matrix $\underline{\boldsymbol{A}}$ represent the linearized matrix $\underline{\boldsymbol{G}}_{SDE}$ in the strap-down equation $\dot{\boldsymbol{\Phi}} = \underline{\boldsymbol{G}}_{SDE}(\boldsymbol{\Phi})\ \boldsymbol{\omega}_{pqr}$ with the attitude angle vector $\boldsymbol{\Phi}^T = [\phi, \theta, \psi]^T$ and $\boldsymbol{\omega}_{pqr}^T = [p, q, r]^T$.

Dynamical behavior of aerial vehicle over velocity envelope: The dynamical behavior of the calculated linear state-space models is analyzed. Thereby, it is distinguished between the six degree of freedom motion (i.e. the full motion), and the lateral and the longitudinal motion of the aerial vehicle. The state-space model of the full motion comprises the states detailed in Eq. (3.201). The states $(u_A^G)_B$, $(w_A^G)_B$, θ, q and $(v_A^G)_B$, ϕ, p, r are considered for the longitudinal and lateral motion.
Firstly, the maximum real part of the system matrix eigenvalues is used as an indicator how the aerial vehicle dynamics change over the trim envelope and thus identifies critical flight conditions in the trim grid for the flight control system. Fig. 3.42 and Fig. 3.43 illustrate the maximum real part of the system matrix eigenvalues of the full, the longitudinal and the lateral state-space models over envelope of the forward and sideward velocity.
As the left plot in Fig. 3.43 shows, the maximum real part of the eigenvalues in the longitudinal motion is positive and rises with increased backward velocity. It reaches the highest value at the highest backward velocity. In contrast, increased forward velocity has only small impact on the maximum real part of the longitudinal eigenvalues.
The right plot in Fig. 3.43 illustrates the maximum real part of the eigenvalues in the lateral motion. Similarly, with increased backward velocity the maximum real part of the lateral eigenvalues increases. In addition, greater maximum real parts of eigenvalues in the lateral motion occur at backward and simultaneously sideward movement.
Fig. 3.42 underlines this observation. The highest maximum real part of the eigenvalues of the full motion occur, if the air flows from the back and the side of the aerial vehicle at the same time. If the aerial vehicle moves backwards faster, the maximum real part of the eigenvalues is also increased.

Secondly, the change of the eigen motion characteristics within the velocity envelope is investigated. Fig. 3.44 and Fig. 3.45 plot the eigenvalues of the full, the longitudinal and the lateral movement for different velocities in the complex plane. The poles for hover are illustrated as red markers for comparison.
Fig. 3.44 shows the pole shift for increased forward (left column) and increased backward (right column) velocity without any sideward velocity:

- *Longitudinal motion with increased forward velocity:* With increased forward velocity two complex pole pairs in the longitudinal motion emerge from four real eigenvalues in hover (see red markers). The right complex pole pair close to the imaginary axis emerges in the right half of the complex plane, but stays with increased forward velocity close to the imaginary axis with small change in their eigenfrequency and relative damping. The second eigenmotion (i.e. the left pole pair) emerges in the left half of the complex plane. The eigenfrequency of this emerging pole pair increases at higher forward velocities, whereas the relative damping stays almost constant.

- *Longitudinal motion with increased backward velocity:* In contrast, with increased backward velocity only one complex pole pair and two real eigenvalue emerge. The eigenfrequency of the pole pair increases, whereas the relative damping decreases slightly. One real eigenvalue moves into the right half plane. The second real eigenvalue shifts to the left in the left half of the complex plane.

- *Lateral motion with increased forward velocity:* The lateral motion consists of two real eigenvalues and one complex pole pair, which are all close to the origin of the complex plane. With increased forward velocity, the eigenfrequency of the complex pole pair increases with almost constant relative damping. One real eigenvalues moves further to the left with increased forward velocity, the second real eigenvalue stays close to the origin.

- *Lateral motion with increased backward velocity:* The complex pole pair in hover increases first its eigenfrequency with increased backward velocity, until it reduces the eigenfrequency again and the complex pole pair becomes two real eigen values. One real eigenvalue in hover shifts further to the lift in left half of the complex plane, the other moves on the right half of the complex plane.

Comparing the eigenvalues of the full motion with the eigenvalues of the longitudinal and lateral motion, it can be stated, that the motion decoupling into lateral and longitudinal motion is possible, if the aerial vehicle does not move sidewards. This is visible in Fig. 3.44, since the eigenvalues of the full motion are qualitatively equivalent to the eigenvalues of the longitudinal and lateral motion combined.

Fig. 3.45 visualizes the pole shift of the eigenvalues for different sidewards velocities to the right with no forward/backward velocity:

- *Longitudinal motion with sidewards velocity:* With increased sidewards velocity one real eigenvalue moves into the right half plane and a complex pole pair emerges with small sideward velocity in the left half plane (see second plot in Fig. 3.45).

- *Lateral motion with sidewards velocity:* The complex pole pair of the lateral motion in hover increases its eigenfrequency with increased sidewards velocity, but eventually moves on the right side of the complex plane. One real eigenvalue moves as well on the right side of the complex plane (see plot in Fig. 3.45 at the bottom).

It should be noted, that by comparing the longitudinal motion and the lateral motion combined with the full motion at higher sideward velocities, coupling effects emerge and the decoupling becomes misleading. For low sideward velocities the tendency and the pole shifting is comparable, but at higher sideward velocities the pole shifting differs. This is visible, for

example, by comparing the locations of the real eigenvalues. The full motion comprises one real eigenvalue with higher real value in the right half plane and one real eigenvalue more left in the left half plane compared to the pole superposition of the lateral and longitudinal motions. This indicates that for higher sidewards velocity a decoupling is not allowed anymore since coupling terms between the lateral and longitudinal motions increase.

The eigenvalue plots underline the significant dynamical change of the transition vehicle's eigen motion at different forward/backward and sideward velocities. In addition, the plots discover the coupling effect of the longitudinal and lateral motion at higher sidewards velocities. Thus, the coupling and the dynamical change needs to be considered in the design of the respective flight control system and its robustness.

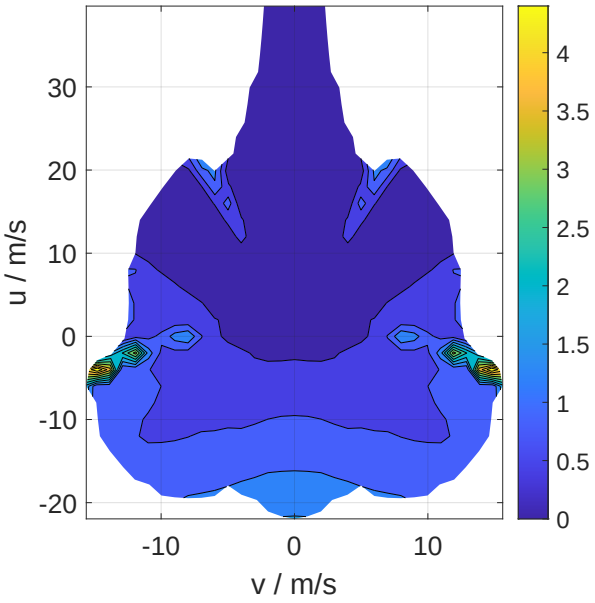


Figure 3.42: Maximum eigen value of flight dynamics over the velocity envelope considering the full aerial vehicle's motion

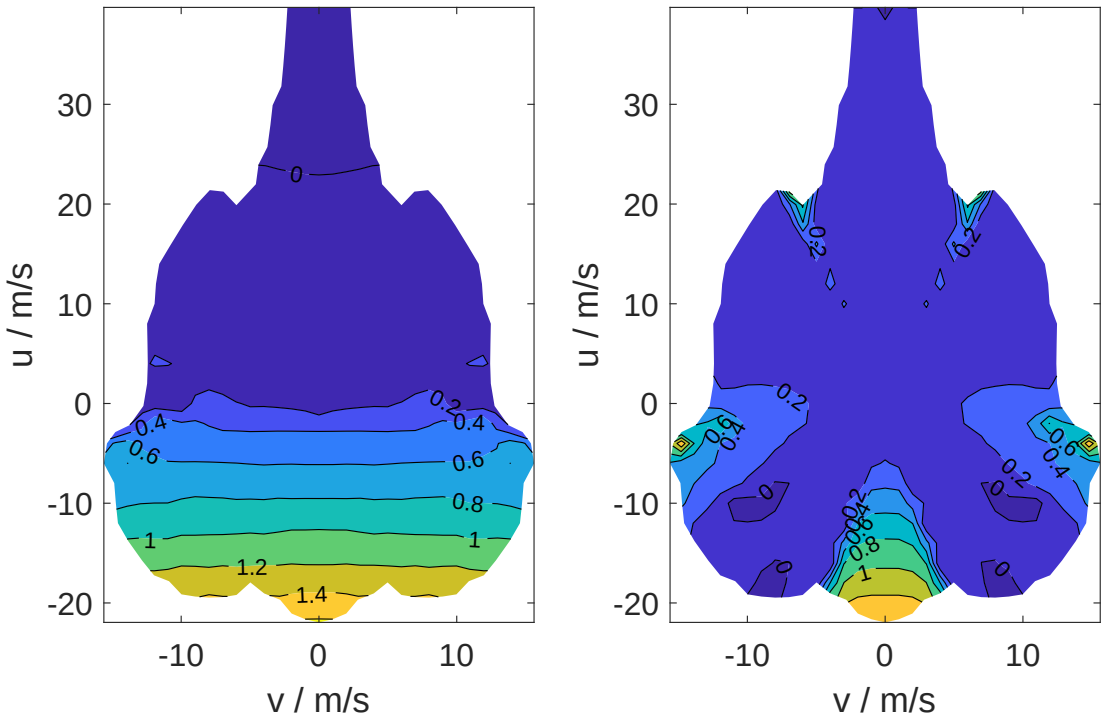


Figure 3.43: Maximum eigen value of the longitudinal and lateral flight dynamics over the velocity envelope

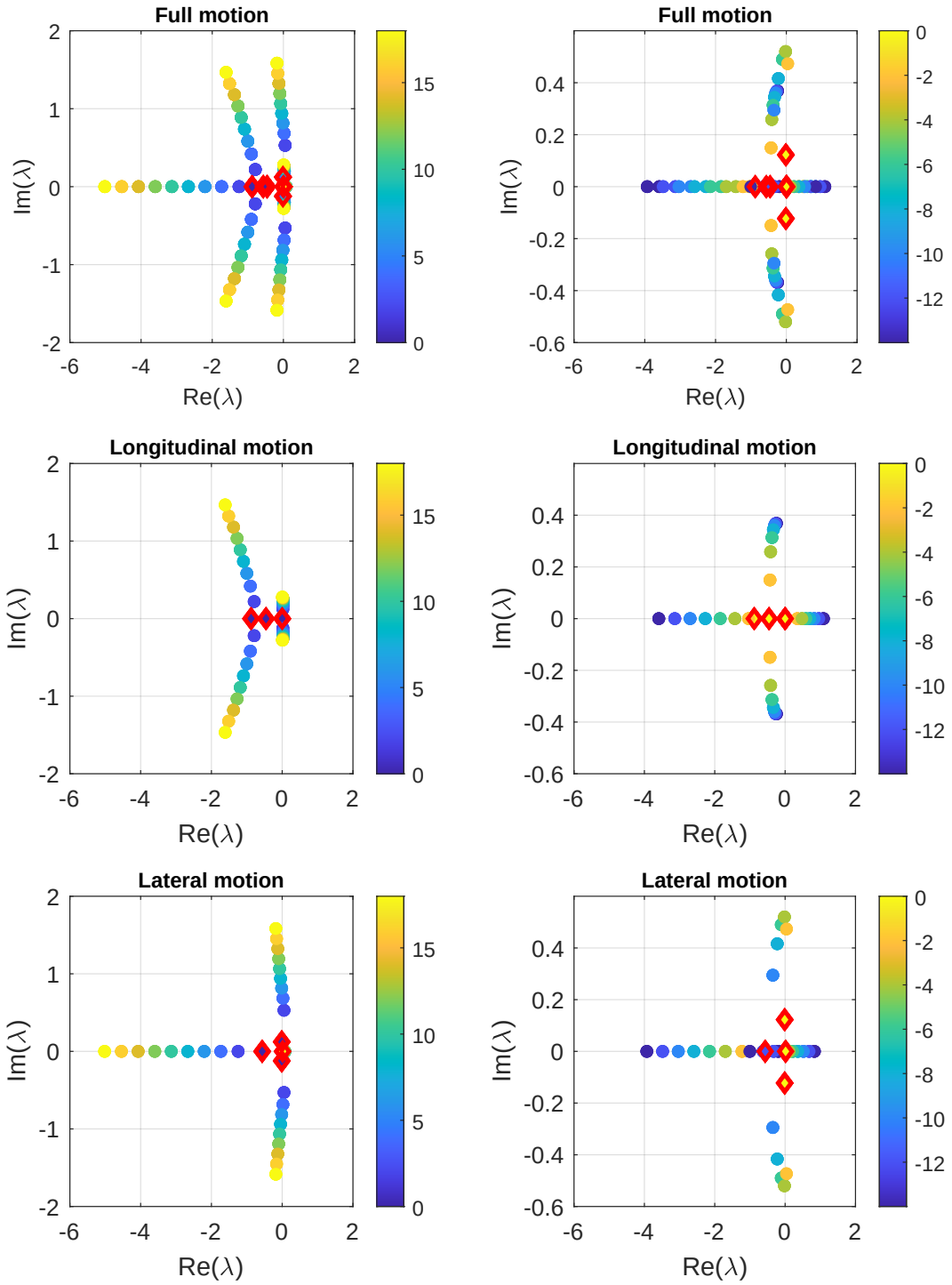


Figure 3.44: Pole shift for **forward** (left column) and **backward** (right column) movements (see forward/backward velocity in colorbar, denoted in m/s) under consideration of the full, longitudinal and lateral motion in comparison to hover (red)

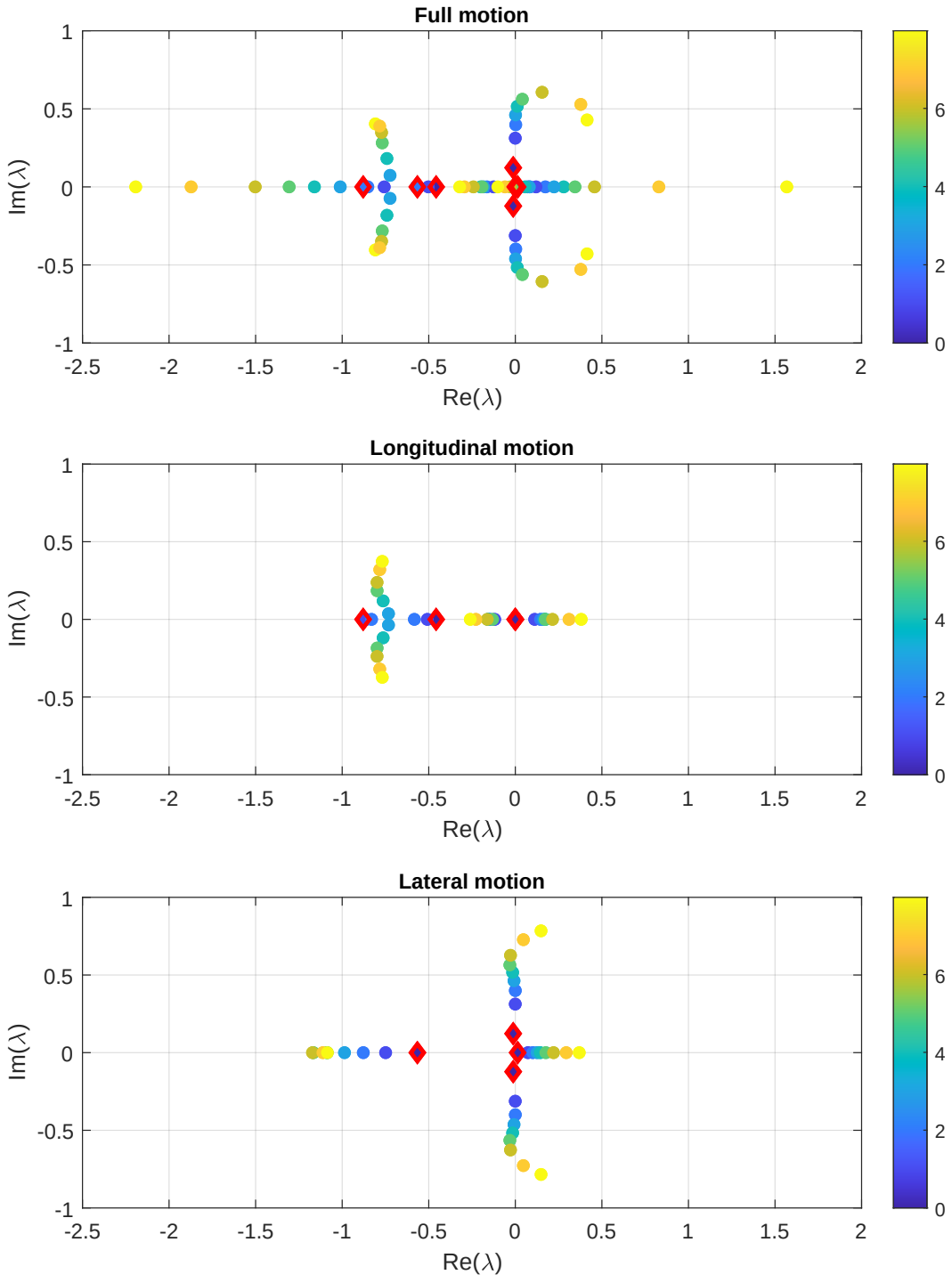


Figure 3.45: Pole shift for different **sidewards** movements (see sidewards velocity in colorbar, denoted in m/s) under consideration of the full, longitudinal and lateral motion in comparison to hover (red)

Sensor system model: In addition to the flight dynamic model, other system components such as the sensor systems are considered. According to the AHRS device manufacturer, the dynamics of the used sensor system comprise a first-order Resistor-Capacitor filter (RC) and a second-order digital filter for the translational and rotational acceleration measurements. The dynamics of the used AHRS are identified using the inertial laboratory available at the *Institute of Flight System Dynamics* at the *Technical University of Munich* (see Ref. [106]). The sensor device is attached to the turntable. Rotational sine-sweep commands are given at the three rotational axes of the turntable at different frequencies. The dynamic characteristics are then identified by comparing the measurements of the AHRS with the commands given to the turntable. An additional deadtime is identified for the attitude signals as well as for the acceleration signals. The sensor transfer functions for the translational accelerations and rotational rates are approximated by

$$G_{sen}(s) = \frac{1}{T_{RC}s+1} \cdot \frac{\omega_{sen}^2}{s^2+2\omega_{sen}\zeta_{sen}s+\omega_{sen}^2} \cdot \exp\left(-T_{t,acc/rat} \cdot s\right) \tag{3.206}$$

with the sensor deadtime $T_{t,acc/rat}$, the RC-filter time constant T_{RC} and the bandwidth frequency ω_{sen} for the second-order filter. The characteristics of the sensor bank and pitch angle signals are approximated by first-order lag elements

$$G_{sen,att}(s) = \frac{1}{T_{t,sen,att}s+1} \cdot \exp\left(-T_{t,sen,att} \cdot s\right) \tag{3.207}$$

with time constant $T_{sen,att}$ and deadtime $T_{t,sen,att}$.

Actuator models of hover propellers: The electric motors of the propellers are represented by second-order transfer functions including maximum moment saturations. In case no saturations occur, the respective control effector dynamics for each hover propeller can be written as:

$$G_{act,c}(s) = \frac{\omega_{z,c}^{Prot}(s)}{\omega_{cmd,c}(s)} = \frac{\omega_{act}^2}{s^2+2\omega_{act}\zeta_{act}s+\omega_{act}^2} \tag{3.208}$$

with the eigenfrequency ω_{act} and the damping ζ_{act}. The output signal of the transfer function $\omega_{z,c}^{Prot}$ corresponds to the rotational speed of the propeller c. The rotational speed command $\omega_{cmd,c}$ of the propeller c is given by the control law, which is elaborated in Sec. 3.10.3.

Communication delays: Under consideration of the system architecture, the hardware components and the sample rates of the interface buses, the worst-case communication delays between the sensor systems, the flight control computer and the actuator units have

been identified. The delays are modeled as one-dimensional deadtime element

$$G_{com_i}(s) = \exp\left(-T_{c_i}s\right) \tag{3.209}$$

with the communication deadtime T_{c_i} for each signal i. Thereby, the worst-case delay is considered for every signal within the robustness analysis.

Uncertainty modeling in nonlinear flight dynamic model: As the representative closed-loop system is analyzed with regard to uncertainties by the application of the μ-analysis and the proposed probabilistic-μ approach, the modeling of uncertainties in the system is required. For the respective eVTOL within this work, the uncertainties are modeled with regard to parameters integrated in the nonlinear flight dynamic model. The uncertainty modeling has been developed at the *Institute of Flight System Dynamics*. Details of the uncertainty modeling and their identification are beyond the scope of this thesis. However, a short overview is given subsequently.
The uncertainties are modeled by Gaussian distributions. The data of such uncertainties is taken from available test bench data, computational fluid dynamics (CFD) simulations (e.g. for aerodynamic data), or based on available data sheets and system assumptions.
The uncertainties are applied to scalar parameters, which are modeled as Gaussian- distributed uncertainties. The scalar parameters are inter alia the derivatives of the aerodynamic multi-compartment model (see e.g. derivatives Eq. (3.191)) and of the propeller model (see e.g. derivatives in Eq. (3.194)). Since some aerodynamic parameters (inter alia parameters modeled as look-up tables) are correlated, the procedure published in Ref. [107] is applied to model stochastic look-up tables. Additional uncertainties are, for example, the position of the center of gravity, the mass and the inertia tensors of the aircraft, and the inertia of the propellers.

Linear representations of the flight dynamic uncertainties: The available probabilistic uncertainties are used to gain on the one hand limited intervals to apply the classical μ-analysis to the system. On the other hand, the new probabilistic μ-approach is applied under consideration of the probabilistic uncertainties.
Therefore, 1000 samples are taken from each of these Gaussian-distributed uncertainties in the nonlinear flight dynamic model. Each perturbed flight dynamic model is trimmed and linearized. As result, at each considered trim point, 1001 state-space models of the flight dynamic model result, which includes one nominal state-space model and 1000 perturbed state-space models. A Gaussian distribution is fitted to each entry of the system matrix $\underline{\boldsymbol{A}}$ and input matrix $\underline{\boldsymbol{B}}$. The expected value and the standard deviation are calculated for each matrix entry of the system matrix $\underline{\boldsymbol{A}}$ and the input matrix $\underline{\boldsymbol{B}}$ (see Eq. (3.199)). For individual

matrix entries the expected value of the Gaussian distribution fit and the value of the nominal model has been compared. The expected value of the Gaussian distribution fit matched within an error interval of 1%. Thus, uncertainties for the linear model have been modeled with the nominal model value as expected value and the calculated standard deviation.
If this approximation is not desired in future work, one has to trim- and linearize the respective flight controller as well at the perturbed plant trim points. As a result, perturbed models for the linearized flight controller emerge and additional uncertainties within the controller state-space models appear. This is not in the scope of this thesis. The flight controller and its linearization are introduced in Sec. 3.10.3. The controller is trimmed and linearized at the nominal trim points. Within this analysis uncertainties of derivatives with respect to the attitude angles (e.g. $X_\phi, X_\theta, Y_\phi, Y_\theta, Z_\phi, Z_\theta$) are also neglected. Nevertheless, 76 uncertainties are considered later within the application of the classical μ-analysis and the probabilistic-μ analysis in the flight dynamic state-space model. The normalized deviations of the 76 uncertainties are shown as Gaussian distributions in Fig. 3.46 and Fig. 3.47. The considered uncertainties can be identified by the respective labels in each subplot.

Uncertainty parameters of system components: The actuators of each propeller are also perturbed by uncertainties. As the actuators are modeled as second-order lag elements, their eigenfrequency and damping values are modeled as parametric uncertain parameters, which are Gaussian-distributed with a standard deviation of $\sigma = 2\%$ of their nominal value. Since the sensor dynamics were identified accurately, no uncertainties were assumed to the parameters of the sensor dynamics.

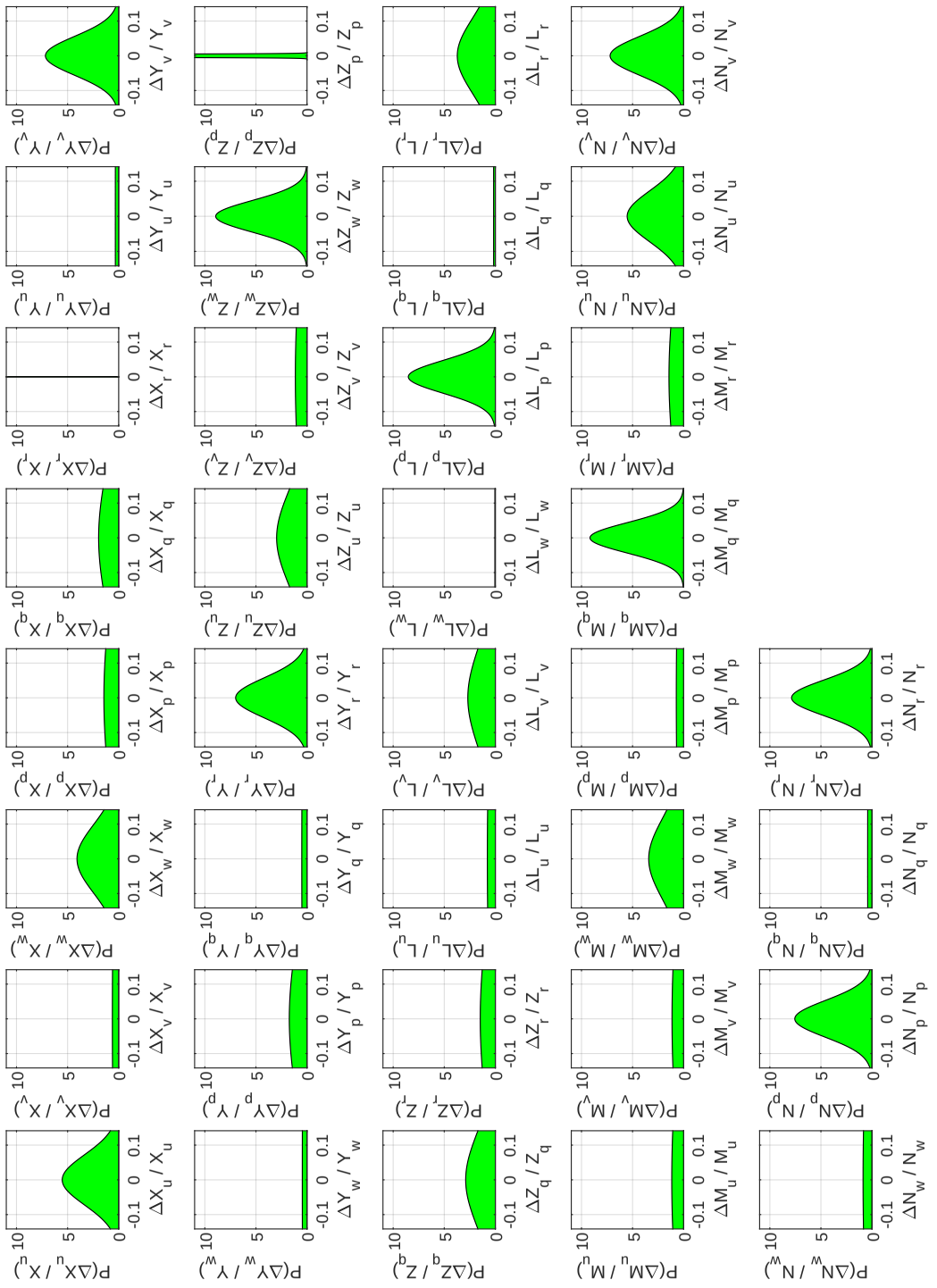


Figure 3.46: Fitted stochastic uncertainties from the nonlinear flight dynamic model within the plant system matrix $\underline{A}$ at the trim point $(u_A^R)_C = 0$, $(v_A^R)_C = -4$ m/s

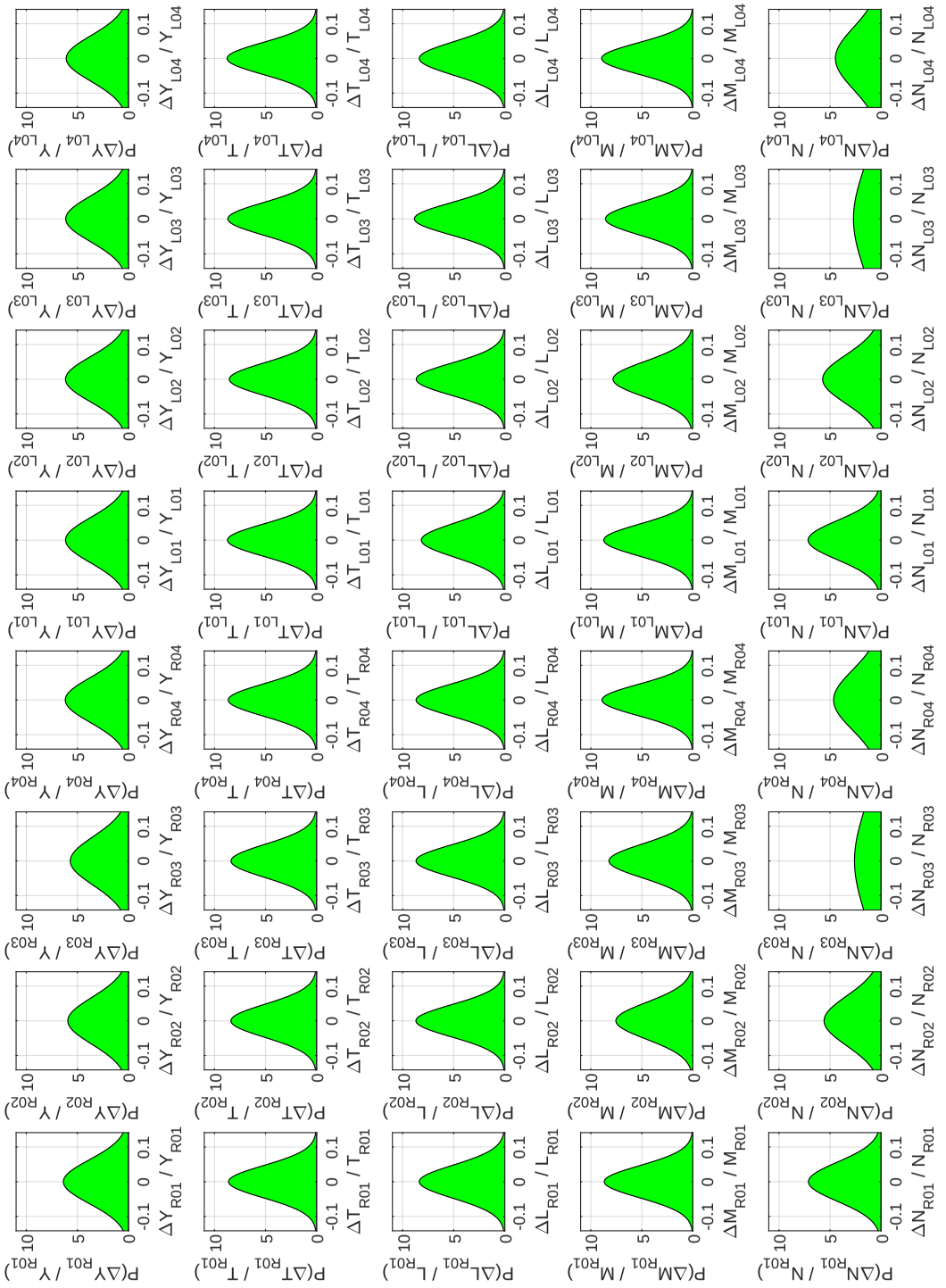


Figure 3.47: Fitted stochastic uncertainties from the nonlinear flight dynamic model within the plant input matrix $\underline{\underline{B}}$ at the trim point $(u_A^R)_C = 0$, $(v_A^R)_C = -4$ m/s

3.10.3 Incremental Nonlinear Dynamic Inversion Attitude Control Structure

An Incremental Nonlinear Dynamic Inversion(INDI) attitude control approach is chosen for the emergency flight control law of the eVTOL demonstrator. The INDI control approach has been already applied to different aerial vehicle configurations (see e.g. Refs. [41, 108, 109]). In contrast to the nonlinear dynamic inversion approach, INDI calculates the control input on the basis of desired acceleration increments utilizing acceleration measurements and/or estimations. This can have the advantage, that the control relies less on the accuracy of the dynamic model (see Ref. [108, p.12]).
Nevertheless, the work in Ref. [108, p.13] further points out, INDI needs to overcome two main challenges. The acceleration measurements require filtering due to their noise level, or estimation, which is used in the emergency control law, presented in the following. The estimation adds delay to the system. Furthermore, the approach requires accurate knowledge about the control effectiveness, since it is based on inverting the control dynamics.
The attitude emergency flight control law for the eVTOL demonstrator is based on the work in Ref. [41] and is presented in the following. Fig. 3.48 illustrates the respective linear closed-loop system. It consists of the above elaborated plant model, the actuator models $\underline{\boldsymbol{G}}_{act}(s)$, the sensor models $\underline{\boldsymbol{G}}_{sen}(s)$ and the control law, which is detailed in this section. The control law consists of the control algorithm $\underline{\boldsymbol{C}}(s)$, the control allocation $\boldsymbol{CA}$, estimation filters $\underline{\boldsymbol{E}}(s)$, and the command filter $\underline{\boldsymbol{F}}(s)$. Each controller component and their respective linear representations are introduced subsequently. After the linear representations of each controller component are calculated, the state-space models of the components are connected to gain the complete linear representation of the controller.

Controller Algorithm: Fig. 3.49 details further the INDI control structure. The control inputs of the pilot are thrust command T_{cmd}, pitch angle command θ_{cmd}, bank angle command ϕ_{cmd} and yaw rate command $\dot{\psi}_{cmd}$. The bank angle command and the pitch angle command signals are first filtered by reference models consisting of an first-order actuator model and a second-order dynamic model with the states $\phi_{act}, \phi_{ref}, \dot{\phi}_{ref}$ and $\theta_{act}, \theta_{ref}, \dot{\theta}_{ref}$. The yaw rate command is shaped by a reference model, which consists of a first-order actuator model and a first-order lag element. It comprises the states $\dot{\psi}_{act}, \dot{\psi}_{ref}$. The reference models represent the desired dynamics. The control errors between the reference model angles/ rates (i.e. ϕ_{ref}, θ_{ref} and $\dot{\phi}_{ref}, \dot{\theta}_{ref}$) and the measured angles/ rates are multiplied by control error gains and added to the desired angular accelerations $\ddot{\boldsymbol{\Phi}}_{ref} = [\ddot{\phi}_{ref}, \ddot{\theta}_{ref}, \ddot{\psi}_{ref}]^T$ of the reference models (see Fig. 3.49).

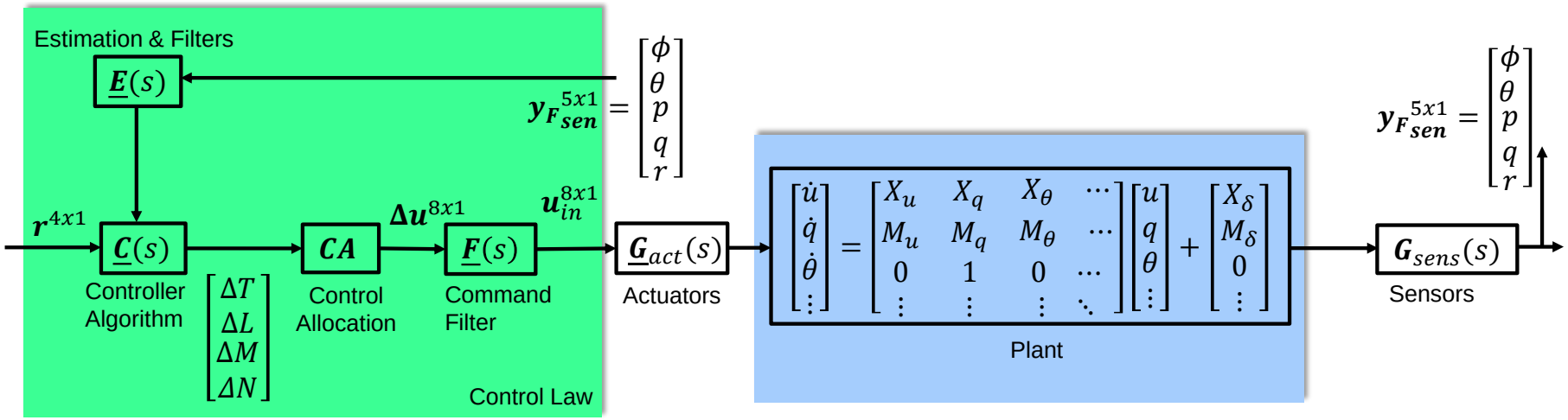


Figure 3.48: Linear closed-Loop system with detailed control law

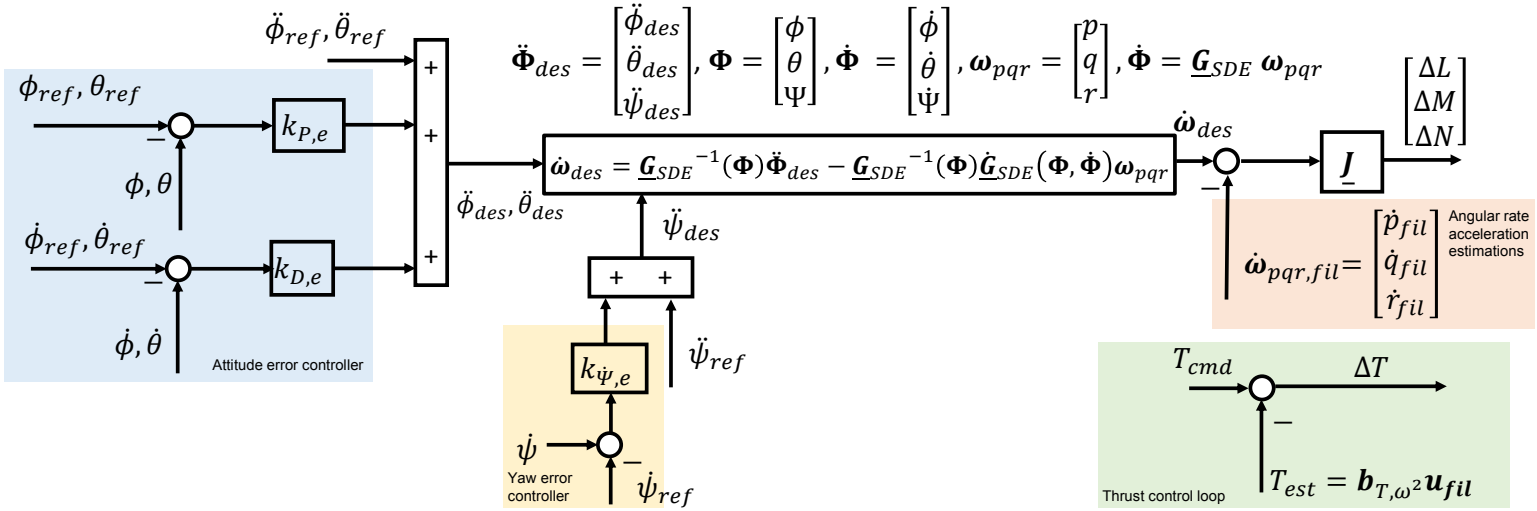


Figure 3.49: INDI controller algorithm $\underline{C}(s)$ without illustrated reference models

The desired angular accelerations $\ddot{\Phi}_{des}$ can be transformed to the desired angular accelerations $\dot{\omega}_{des} = [\dot{p}_{des}, \dot{q}_{des}, \dot{r}_{des}]^T$ under consideration of the strap-down equation:

$$\dot{\Phi} = \underline{G}_{SDE}(\Phi)\omega_{pqr} \tag{3.210}$$

$$\ddot{\Phi} = \underline{G}_{SDE}(\Phi)\dot{\omega}_{pqr} + \dot{\underline{G}}_{SDE}(\Phi, \dot{\Phi})\omega_{pqr} \tag{3.211}$$

with the angular rate vector $\omega_{pqr} = \begin{bmatrix} p, q, r \end{bmatrix}^T$ and the attitude vector $\Phi = \begin{bmatrix} \phi, \theta, \psi \end{bmatrix}^T$. The transformation is calculated by the following equation:

$$\dot{\omega}_{des} = \left(\underline{G}(\Phi)^{-1}\ddot{\Phi}_{des} - \underline{G}_{SDE}(\Phi)^{-1}\dot{\underline{G}}_{SDE}(\Phi, \dot{\Phi})\omega_{pqr} \right) \tag{3.212}$$

The desired angular accelerations $\dot{\omega}_{des}$ are subtracted by the filtered angular accelerations $\dot{\omega}_{pqr,fil}$ and multiplied by the assumed plant inertia tensor $\underline{J} \in \mathbb{R}^{3\times3}$:

$$\Delta M^{3\times1} = \underline{J}\left(\dot{\omega}_{des} - \dot{\omega}_{pqr,fil}\right) \tag{3.213}$$

This results in incremental pseudo-moment commands $\Delta M = [\Delta L, \Delta M, \Delta N]^T$, which are inputs to the control allocation. Note that an additional lead-lag filter is included after the calculated signal vector $\Delta M^{3\times1}$ and the control allocation, to achieve better robustness margin results. The lead-lag filter represents a linear SISO transfer function for each signal. The design of the lead-lag filter is beyond of this thesis. However, its values are tuned heuristically.
The thrust increment ΔT is computed as follows:

$$\Delta T = \left(T_{cmd} - b_{T,\omega^2} u_{fil}\right) \tag{3.214}$$

The vector $u_{fil} = [\omega^2_{L01,fil}, \omega^2_{L02,fil}, ..., \omega^2_{R04,fil}]$ represents the squared propeller turn rates $\omega_{fil} = [\omega_{L01,fil}, \omega_{L02,fil}, ..., \omega_{R04,fil}]$, which are estimated in the command filter by actuator models. The row vector $b_{T,\omega^2} \in \mathbb{R}^{1\times8}$ represents the respective row of the control effector matrix $\underline{B}_{\omega^2} = \begin{bmatrix} b_{T,\omega^2},\ b_{L,\omega^2},\ b_{M,\omega^2},\ b_{N,\omega^2} \end{bmatrix}^T$ with regard to the thrust increment ΔT. The control effectivity matrix $\underline{B}_{\omega^2}$ is derived from the input matrix $\underline{B}$ of the linearized flight dynamic model (see Eq. (3.199)). Further explanation with regard to the control effectivity matrix can be found in the control allocation section, elaborated later in this section. The command filter is explained subsequently (see also Fig. 3.50). The thrust control structure is illustrated in Fig. 3.49.

Linear Representation of Controller Algorithm: The nonlinear controller is trimmed and linearized to gain the linear state-space representation $\underline{C}$, which is further used to build up and analyze the linear closed-loop system for robustness. Therefore, an exact knowledge of the inputs, states and outputs of the controller is required:

- Command Inputs: $\psi_{cmd}, \phi_{cmd}, \theta_{cmd}, \Delta T_{cmd}$
- Estimation Inputs: $\phi_{est}, \theta_{est}, \psi_{est}, \dot{\phi}_{est}, \dot{\theta}_{est}, \dot{\psi}_{est}$
- States: $\phi_{ref}, \theta_{ref}, \dot{\phi}_{ref}, \dot{\theta}_{ref}, \dot{\psi}_{ref}, \dot{\phi}_{act}, \dot{\theta}_{act}, \dot{\psi}_{act}$
- Outputs: $\Delta\boldsymbol{\nu} = [\Delta T, \Delta L, \Delta M, \Delta N]^T$

Analogously to the trimming and linearization of the nonlinear flight dynamic model, an appropriate trim template is set up by defining the trim conditions, the solver variables and the equality conditions.

- Trim conditions: (1)-(3) $\phi_{est}, \theta_{est}, \psi_{est}$
- Solver variables:
  - (1)-(4) $\dot{\psi}_{cmd}, \phi_{cmd}, \theta_{cmd}, T_{cmd}$
  - (5)-(7) $\phi_{act}, \theta_{act}, \dot{\psi}_{act}$
  - (8)-(9) ϕ_{ref}, θ_{ref}
  - (10)-(12) $\dot{\phi}_{ref}, \dot{\theta}_{ref}, \dot{\psi}_{ref}$
- Equality conditions:
  - (1)-(2) $\dot{\phi}_{ref}, \dot{\theta}_{ref} = 0$
  - (3)-(5) $\dot{\phi}_{act}, \dot{\theta}_{act}, \dot{\psi}_{act} = 0$
  - (6)-(8) $\ddot{\phi}_{ref}, \ddot{\theta}_{ref}, \ddot{\psi}_{ref} = 0$
  - (9)-(12) $\Delta T, \Delta L, \Delta M, \Delta N = 0$

With the trim template the controller is trimmed by the application of nonlinear programming solvers and thereafter linearized by numerical perturbation. The trim conditions and trim points are chosen appropriately and equivalently to the plant trim points, which is important for a consistent building of the linear closed-loop systems. Note that the mentioned lead-lag filters are added to the resulting state-space model from the numerical trimming and linearization, such that finally the state-space model $\underline{C}$ results. For that reason the states of the lead-lag filters are not listed in the presented trim template.

Acceleration and Estimation Filter: The filtered angular accelerations $\boldsymbol{\omega}_{pqr,fil}$ are estimated using the angular rate measurements from the AHRS. The signals $\dot{\phi}$ and $\dot{\theta}$ are calculated from the rate measurements by the strap-down equations with the attitude measurements. The attitude signals and the translational and rotational accelerations are shaped by a second order roll-off filter.

Control Allocation: Within the presented controller, a *redistributed pseudo-inverse* control allocation approach is applied (see e.g. Ref. [110]). Subsequently, only relevant details of the control allocation for the robustness analysis are addressed. The control allocation maps the incremental pseudo control command $\Delta\boldsymbol{\nu} = [\Delta T, \Delta L, \Delta M, \Delta N]^T$ to incremental squared rotational speed commands $\Delta\boldsymbol{u} = \mathrm{diag}\left(\Delta\boldsymbol{\omega} \cdot \Delta\boldsymbol{\omega}^T\right)$ of the eight hover propeller. Therefore, the

control effectivity matrix $\underline{\boldsymbol{B}}_{\omega^2}$, which considers the squared rotational speeds of the propellers is used. The control effectivity matrix $\underline{\boldsymbol{B}}_{\omega^2}$ describes the influence of the eight hover-propeller on the translational acceleration $(f_z^G)_B$ and the angular accelerations $\dot{p}, \dot{q}, \dot{r}$, such that:

$$[\Delta T, \Delta L, \Delta M, \Delta N]^T = \underline{\boldsymbol{B}}_{\omega^2} \Delta \boldsymbol{u} \tag{3.215}$$

The vector $[\Delta T, \Delta L, \Delta M, \Delta N]$ represents the incremental force and moments commands, which shall produce a desired acceleration $(f_z^G)_B$ (which is the translational acceleration in the z_B-direction of the aerial vehicle) and angular accelerations $\dot{p}, \dot{q}, \dot{r}$. Note that the matrix $\underline{\boldsymbol{B}}_{\omega^2}$ represents the control effectivity matrix considering the squared rotational speeds of the propellers. The input matrix $\underline{\boldsymbol{B}}$ of the linear flight dynamic model considers the rotational speed of the propellers only (i.e. not squared). The relation between the corresponding entries of the input matrix $\underline{\boldsymbol{B}}$ and $\underline{\boldsymbol{B}}_{\omega^2}$ can be derived analytically. Although the input matrix $\underline{\boldsymbol{B}}$ changes over the envelope, the used control effectivity matrix is a constant matrix. To cover therefor a greater spectrum of trim points, the matrix is averaged over a trim grid of input matrices of the flight dynamic model. The averaged matrix is then used to eventually calculate the matrix $\underline{\boldsymbol{B}}_{\omega^2}$. The non-quadratic matrix $\underline{\boldsymbol{B}}_{\omega^2}$ is inverted by taking the Moore-Penrose inverse $\underline{\boldsymbol{B}}^+$:

$$\underline{\boldsymbol{B}}^+ = \mathrm{pinv}(\underline{\boldsymbol{B}}_{\omega^2}) \tag{3.216}$$

As a result, the incremental squared rotational speed command $\Delta \boldsymbol{u}$ for the propeller can be calculated:

$$\Delta \boldsymbol{u} = \underline{\boldsymbol{B}}^+ \Delta \boldsymbol{\nu} = \underline{\boldsymbol{B}}^+ \cdot [\Delta T, \Delta L, \Delta M, \Delta N]^T \tag{3.217}$$

The incremental squared rotational speed command $\Delta \boldsymbol{u}$ is further processed in the command filter $\underline{\boldsymbol{F}}$ to calculate absolute propeller turn rate commands.
In the actual implementation, an additional term $\Delta \boldsymbol{u}_N$, which utilizes the nullspace of the control effectors to ensure that $\boldsymbol{u}^T \boldsymbol{u}$ is minimized is added to the increment $\Delta \boldsymbol{u}$. However, the term $\Delta \boldsymbol{u}_N$ is neglected in the further analysis within this thesis. Numerical linearization is not necessary, because the control allocation is linear (see Eq. (3.217)) by its design and does not encompass dynamic behavior.

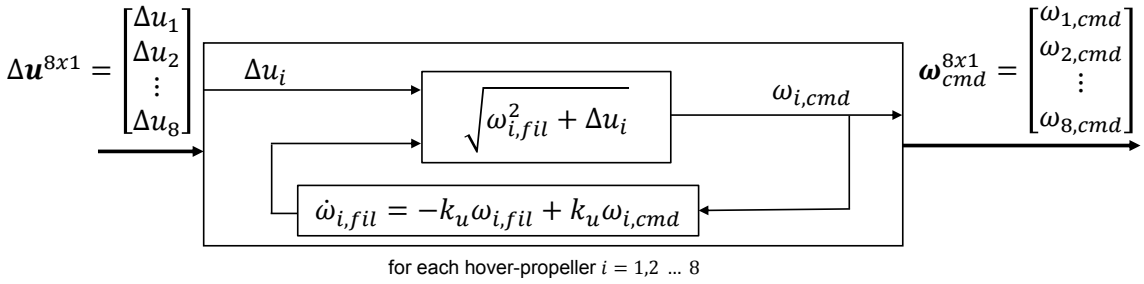


Figure 3.50: Command filter $\underline{\boldsymbol{F}}(s)$ with INDI approach

Command Filter: The command filter of the emergency control law is illustrated in Fig. 3.50. The incremental rotational propeller speed command signal $\Delta \boldsymbol{u}$ of each propeller is added to the squared filtered rotational speed signal $\boldsymbol{\omega}_{fil}$ of each propeller. After taking the square root, the resulting signal $\omega_{i,cmd}$ is the rotational speed command of the i-th hover-propeller, which is sent to the corresponding control effector actuator. The same signal $\omega_{i,cmd}$ is an input to the actuator models of the eight hover-propeller. The model output is equivalent to the squared filtered rotational speed signal $\boldsymbol{\omega}_{fil}$. The actuator models are represented by first-order lag elements with the time constant $T_u = 1/k_u$ for each hover -propeller actuator.

Linear Representation of Command Filter: For the linear representation of the command filters, the square root $\sqrt{\omega_{i,fil}^2 + \Delta u_i}$ and the actuator model are considered as two separated dynamical systems for each propeller. The trim values of the rotational speeds, gained as result by the trimming of the plant, are used for the linearization of the command filters.
Firstly, the square root is linearized considering the two input signals Δu_i and $\omega_{i,fil}$:

$$h_i(\omega_{i,fil}, \Delta u_i) = \omega_{i,cmd} = \sqrt{\omega_{i,fil}^2 + \Delta u_i} \tag{3.218}$$

Considering $\Delta u_i|_0 = 0$ and the terms

$$\left.\frac{\partial h_i}{\partial \omega_{i,fil}}\right|_0 = \left.\frac{2\omega_{i,fil}}{2\sqrt{\omega_{i,fil}^2 + \Delta u_i}}\right|_0 = 1 \tag{3.219}$$

$$\left.\frac{\partial h_i}{\partial \Delta u_i}\right|_0 = \left.\frac{1}{2\sqrt{\omega_{i,fil}^2 + \Delta u_i}}\right|_0 = \left.\frac{1}{2\omega_{i,fil}}\right|_0 \tag{3.220}$$

the linear representation of Eq. (3.218) becomes:

$$y_i = \left.\frac{\partial y_i}{\partial \omega_{i,rpm,fil}}\right|_0 \cdot \Delta\omega_{i,fil} + \left.\frac{\partial y_i}{\partial \Delta u_i}\right|_0 \Delta u_i = \Delta\omega_{i,fil} + \left.\frac{1}{2\omega_{i,fil}}\right|_0 \Delta u_i \tag{3.221}$$

It is not required to linearize the used actuator models in the command filter (see Fig. 3.50), since they are implemented as first-order lag elements with the output and state $\omega_{i,fil}$ and input $\omega_{i,cmd}$ and the time constant $T_u = 1/k_u$.
Finally, the MIMO state-space models of the command filter is gained by connecting the each actuator model with the corresponding linear equations of the square root $\sqrt{\omega_{i,fil}^2 + \Delta u_i}$ for each propeller i.

3.10.4 Building and Verification between Nonlinear and Linear Closed-Loop Model

After the plant, the sensors, the actuators and the controller are available as linear state-space models at defined trim points, the linear closed-loop systems are build.
It is required to ensure that the linear closed-loop model matches with the nonlinear model near the trim point in order to gain valid robustness margins, which shall also be representative in the nonlinear world near the trim point. Since the present closed-loop system is nonlinear (e.g. propeller forces comprise squared turn rates as nonlinearities and the aerodynamics model contains nonlinear look-up tables), the derived linear closed-loop system is verified by comparing the linear with the nonlinear closed-loop system at the trim point.
The closed-loop system is built by connecting the state-space models step-wise. The state-space models are linked by using the command *connect* from Ref. [88]. This needs to be done carefully, otherwise dynamics get lost unintentionally. Therefor, it is very important to verify the gained linear closed-loop system.
The nonlinear model states of the extended plant with sensor and actuator models and the controller are initialized at the respective trim point for a valid comparison. Within this work, three different step commands are given to each command channel of the closed-loop system with feedback signals. The results are illustrated for each command channel in Figs. 3.51-3.53 in the hover trim point. The signals are verified qualitatively by evaluation and quantitatively utilizing signal error norms. An empirically-determined error tolerance is chosen for the quantitative evaluation of the signal error norms.
The presented closed-loop system is characterized by the four input signals T_{cmd}, ϕ_{cmd}, θ_{cmd}, $\dot{\psi}_{cmd}$, in total 68 states (without counting dead times), and 10 chosen output signals. After the closed-loop system is build and verified, a major milestone for the robustness analysis is reached.

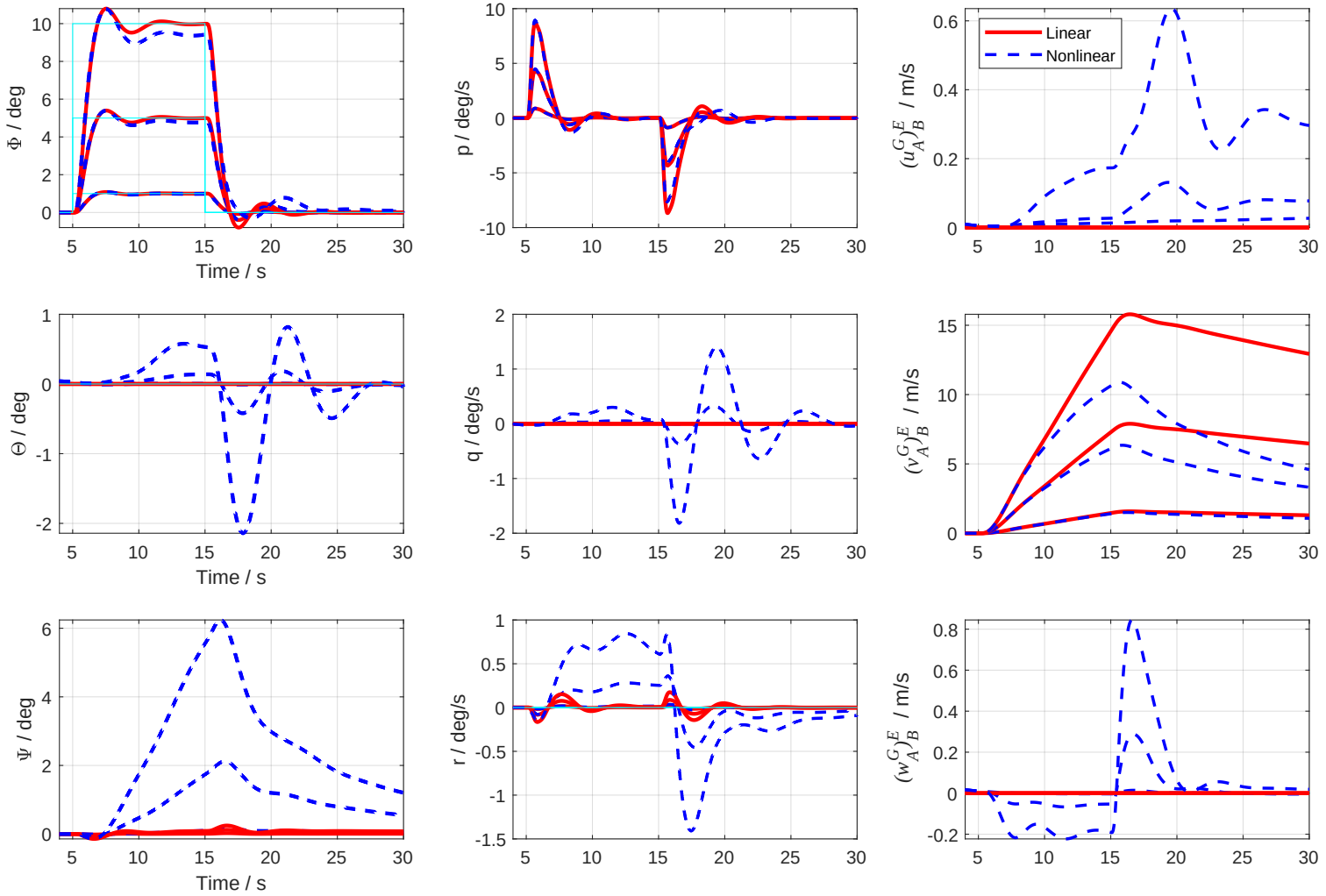


Figure 3.51: Verification of linear with nonlinear closed-loop system with three different step inputs in the **roll command channel** ϕ_{cmd}: linear model (red), nonlinear model (blue)

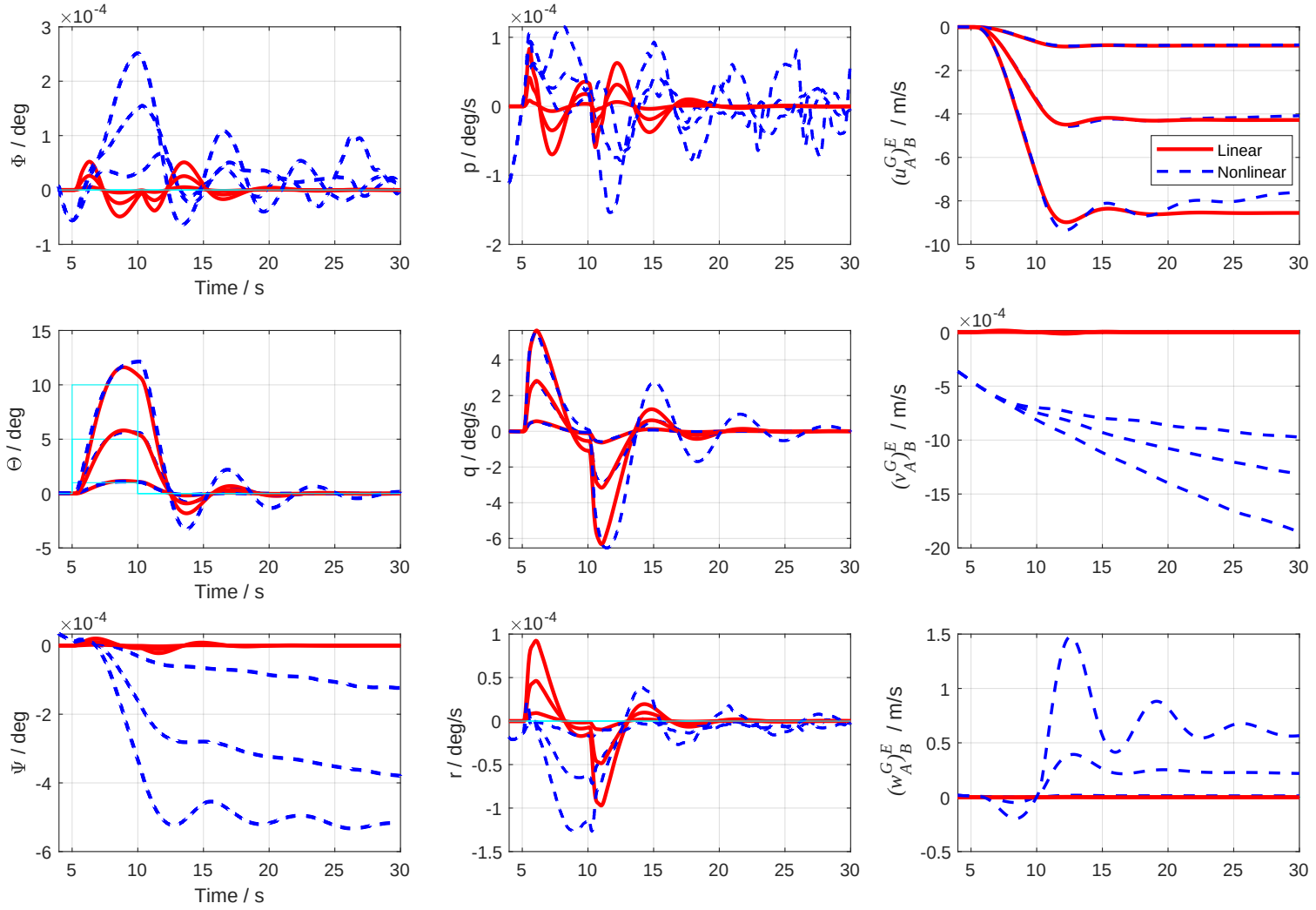


Figure 3.52: Verification of linear with nonlinear closed-loop system with three different step inputs in the **pitch command channel** θ_{cmd}: linear model (red), nonlinear model (blue)

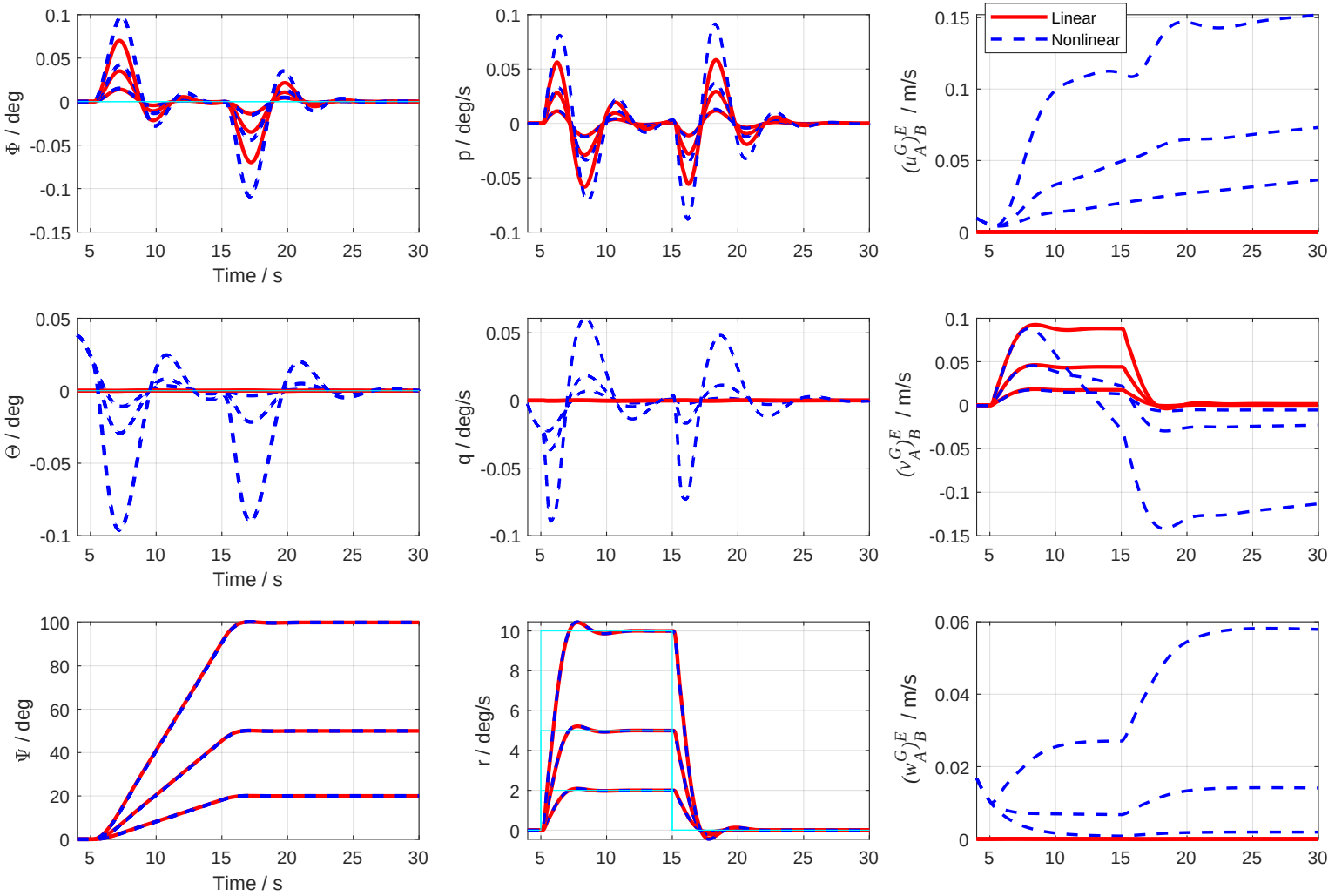


Figure 3.53: Verification of linear with nonlinear closed-loop system with three different step inputs in the **yaw command channel** ψ_{cmd}: linear model (red), nonlinear model (blue)

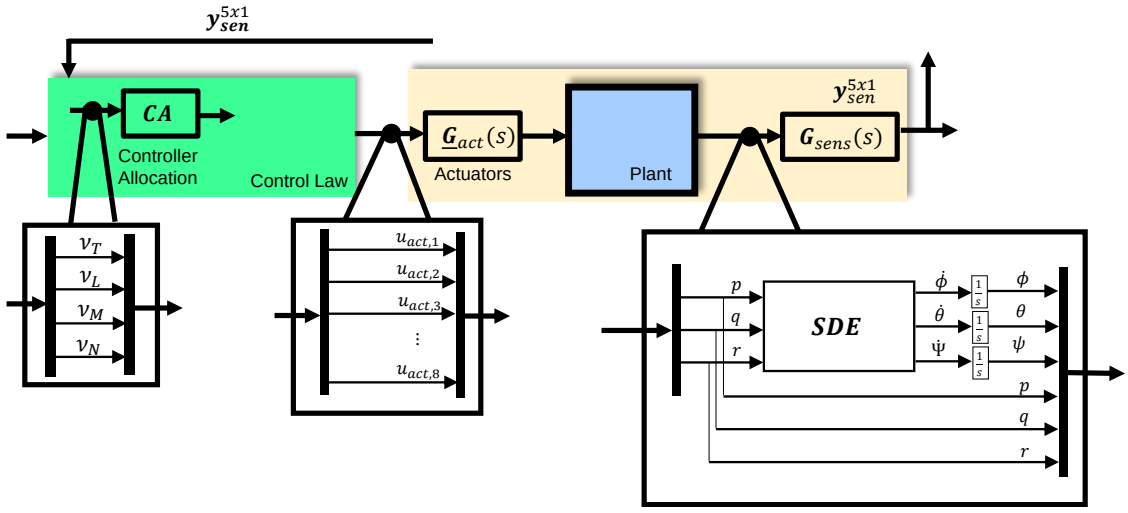


Figure 3.54: Selection of Analysis Points for SISO and MIMO margins

3.10.5 Selection of Analysis Points

The presented SISO and MIMO gain and phase margins can be calculated at different analysis points. It is important to mention, that the presented closed-loop systems are MIMO systems with a significant number of input and output channels, such that classical SISO margins are only applicable with caution. As elaborated in Sec. 3.2.3, for the application of SISO robustness metrics, single-loop breakings need to be conducted in case of a MIMO system by keeping channels in parallel closed. Thus, the number of channels in parallel needs to be considered in the interpretation and assessment of the SISO margins. Equivalently, the introduced MIMO margins can also be calculated at different analysis points. Note that the number of channels in parallel also needs to be considered in the assessment of the MIMO margin results since one uncertainty is introduced at each channel of the analysis point in the MIMO margin calculation (see Sec. 3.5.4). The more channels are in parallel at the analysis point, the more uncertainties are taken into account.
Within this work, the analysis points are chosen such that physical interpretations of the robustness margins are present. The analysis points are set at the actuators, the sensors and before the control allocation. Since in aerospace standards (e.g. Ref. [79]) analysis at the actuators and the sensors are required, within this work analysis points are also selected at these respective locations. The three locations are discussed in the following.

Actuators: The analysis point at the actuators is set at the input signals of the actuators. In this respect the analysis point is located at the system border between the flight control system (with actuator commands as output) and the plant dynamics (including actuator dynamics and flight dynamic model).

However, the calculation of the SISO margins at the actuators are misleading since, for the present eVTOL demonstrator, eight command channels are in parallel. This is a high number of parallel channels in comparison to aircraft with conventional configurations (see e.g. Ref. [79]). Therefore, the margins are deceiving since uncertainties occur mainly simultaneously in these eight channels and are not present in a single channel only.
In contrast, the MIMO margins take uncertainties simultaneously into account and are thus appropriate margins at the actuators. Nevertheless, the results shown in Sec. 3.10.6, demonstrate that also robustness evaluation at the actuator cuts can restrict the aircraft envelope.

Sensors: Within this work, the analysis point for the sensors is set at the angular rate signals of the plant model. This is illustrated in Fig. 3.54. At this system location the angular rates are measured by the inertial measurement unit (IMU) and further processed by the AHRS system to compute the euler angles. Since the euler angles of the eVTOL demonstrator result directly from the angular rate measurements, it is not plausible to set the analysis point at the feedback signal $\boldsymbol{y}^{5x1}$ (in case of the considered system), which encompasses the angular rates and the euler angles. If so, uncertainties would be considered in the margins, which do not occur in reality, since the euler angles are calculated by the AHRS filter. Thus, the analysis point is set at the sensor input. Although it is in contrast to other work (e.g. Ref. [38, p.120]), which set the analysis point at the sensor output, this is necessary for MIMO systems such as the eVTOL demonstrator to gain confident results. If future controllers require the translational acceleration, the controlled translational accelerations are considered at the analysis point as well.

Control Allocation: Another important analysis point can be set at the input of the control allocation (with the pseudo controls $\Delta\boldsymbol{\nu}$ as input). This analysis point has the advantage, compared to the actuator analysis point, that a lower number of signals are in parallel (in this case four signals), which is beneficial for the calculation of the SISO margins. In addition, these signals represent commands to change one specific degree of freedom of the aerial vehicle. The SISO margins can thus be interpreted as the maximum gain and/or phase the controller can compensate for one specific degree of freedom. In this respect, the margin can identify robustness weaknesses for specific motions.

3.10.6 Results of the SISO Metrics

The SISO gain and phase margins are calculated at the presented analysis points over a specified velocity grid and illustrated in Fig. 3.58 considering the control allocation analysis points. The SISO margins at the actuators and the sensors are presented in Fig. 3.59 and Fig. 3.60 as contour plots. The margins are calculated at specific trim points, which are illustrated with crosses (+). The color indicates, if the requirement is fulfilled (green) or is not fulfilled (yellow and red). The contour plots interpolate the color between the trim points. The corresponding nichols plots are presented in Fig. 3.55 - Fig. 3.56.
Additionally, an automatic routine assessed if the nichols diamond is cut under consideration of the open-loop transfer function at the respective trim point. The evaluation results for the three analysis points are shown in Fig. 3.61. The color at the the trim point, illustrated with crosses (+), indicates if the nichols diamond is cut (red) or is not cut (green). Again, the color is interpolated between the trim points. subsequently, the SISO robustness results are shown and evaluated. Thereafter, Sec. 3.10.7 shows the application of the MIMO margins.

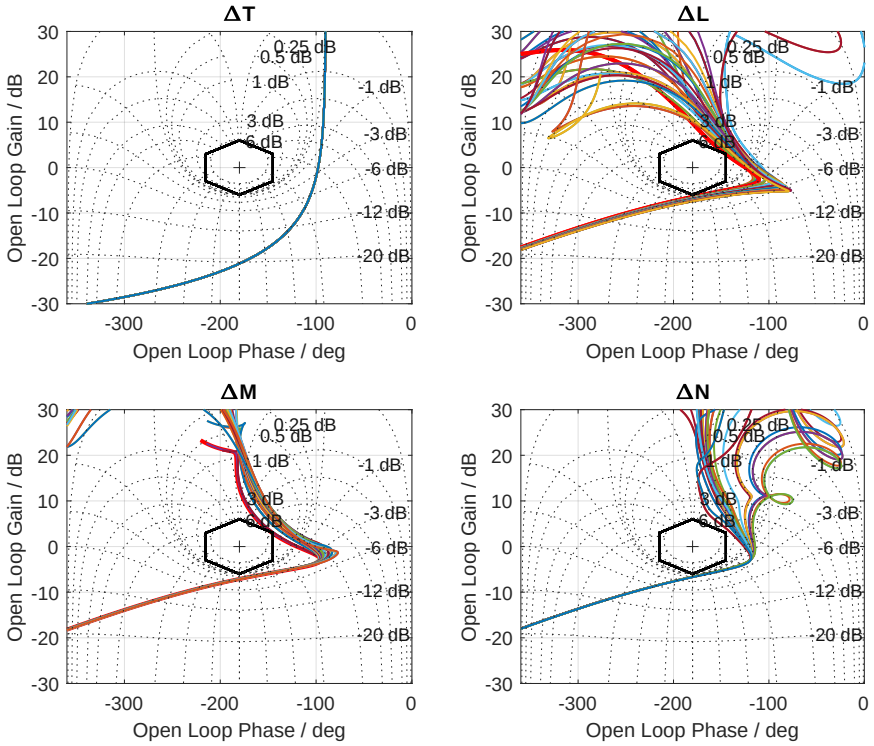


Figure 3.55: Nichols plots over envelope with sidewards velocity up to 4 m/s and forward/ backward velocity between -4 m/s and 14 m/s considering *bottleneck-cut* before control allocation: Lines are marked, if nichols diamond is cut (red)

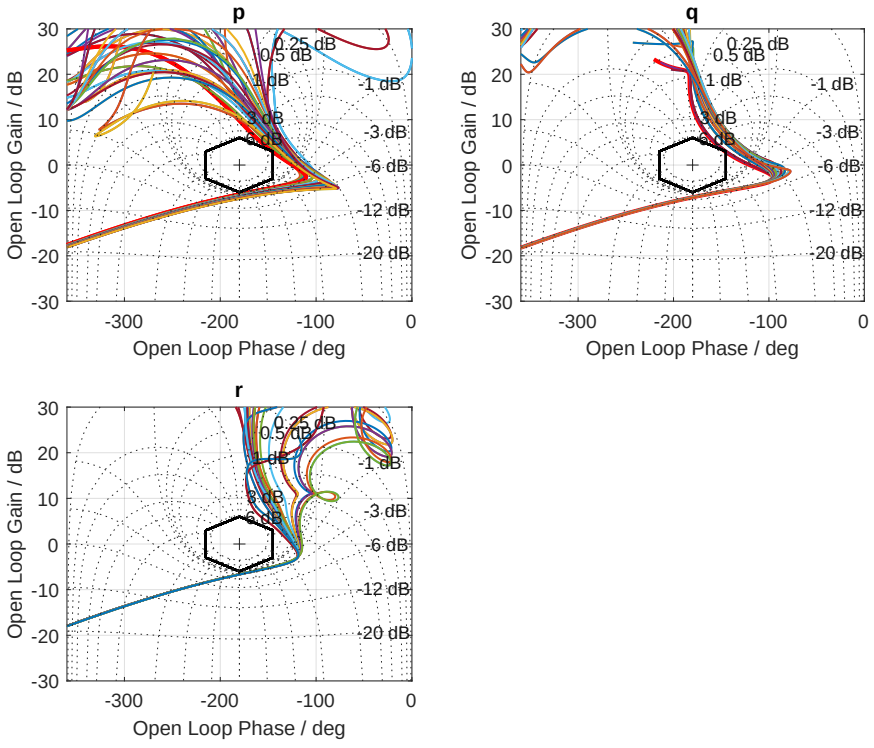


Figure 3.56: Nichols plots over envelope with sidewards velocity up to 4 m/s and forward/ backward velocity between -4 m/s and 14 m/s considering *bottleneck-cut* before sensors: Lines are marked, if nichols diamond is cut (red)

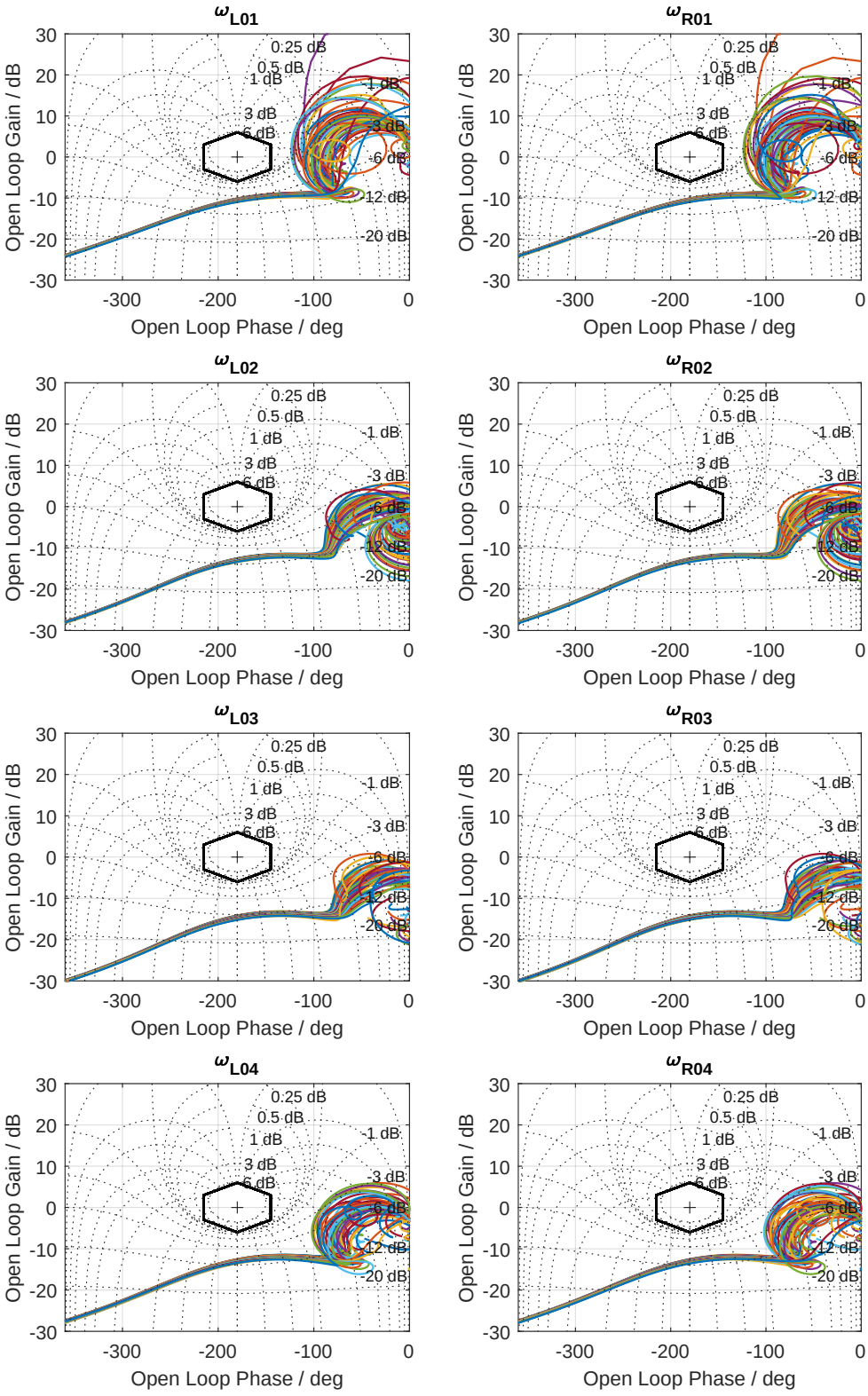


Figure 3.57: Nichols plots over envelope with sidewards velocity up to 4 m/s and forward/ backward velocity between -4 m/s and 14 m/s considering *bottleneck-cut* at control allocation.

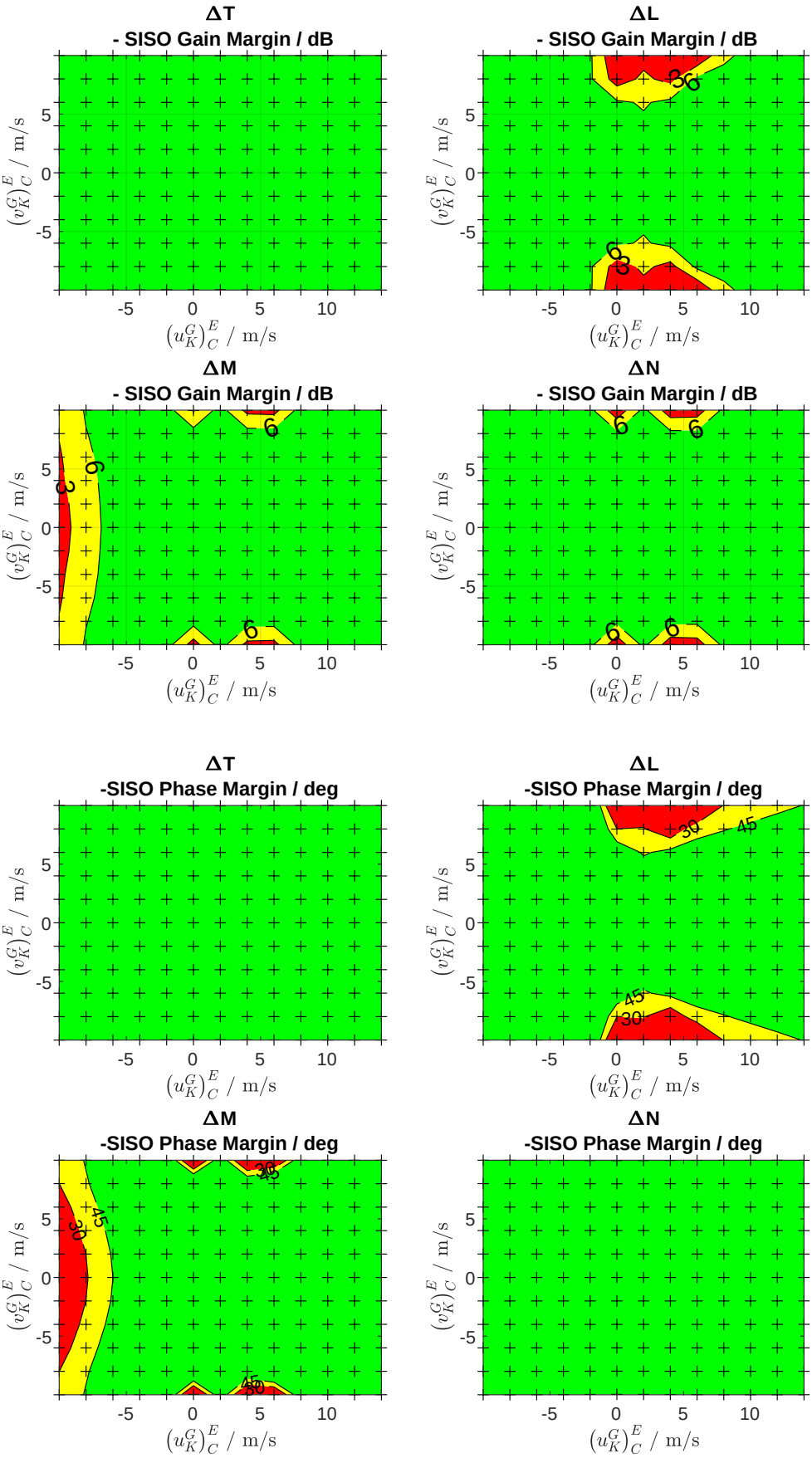


Figure 3.58: SISO gain and phase margin before control allocation

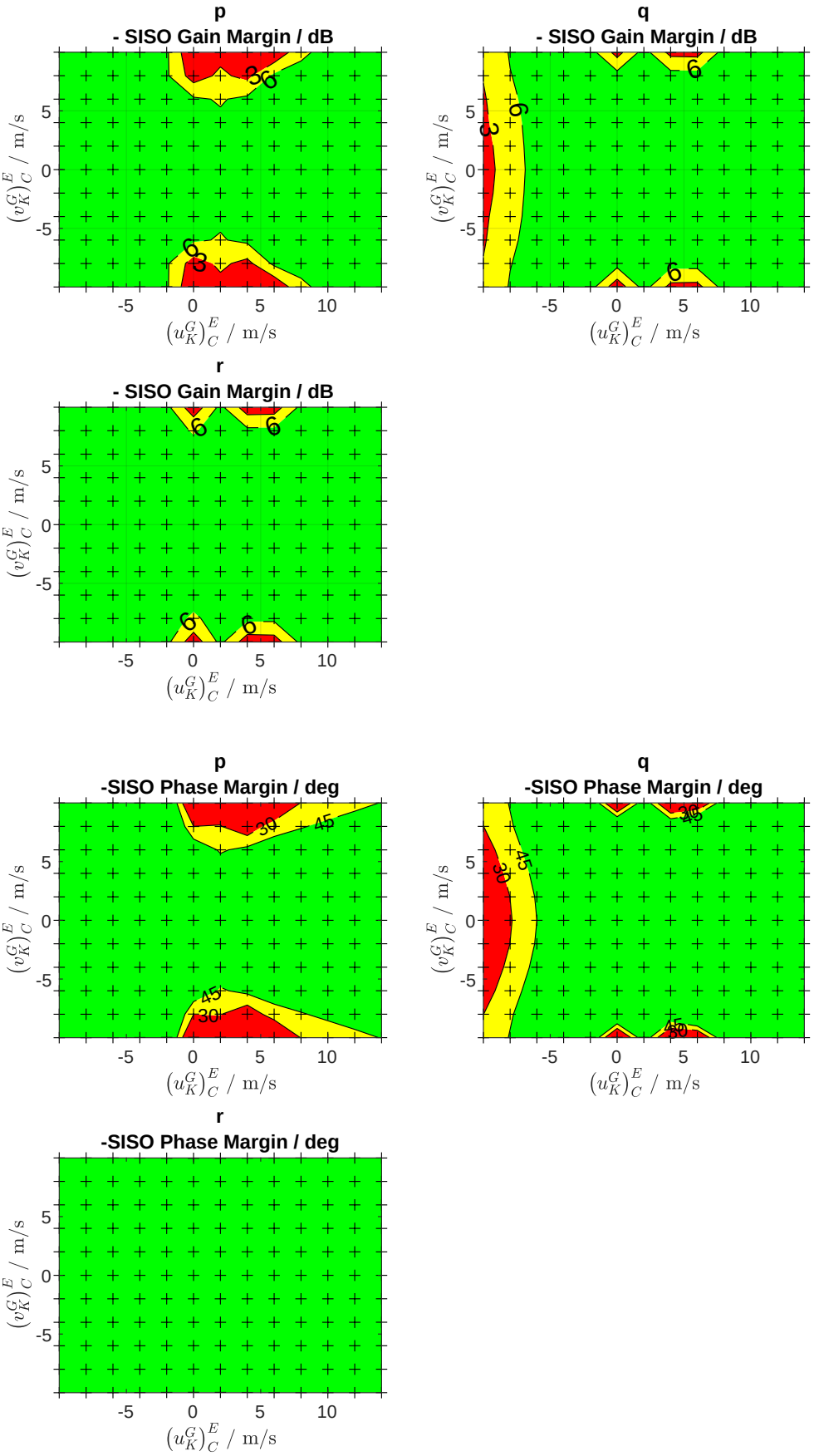


Figure 3.59: SISO gain and phase margin at sensor AP

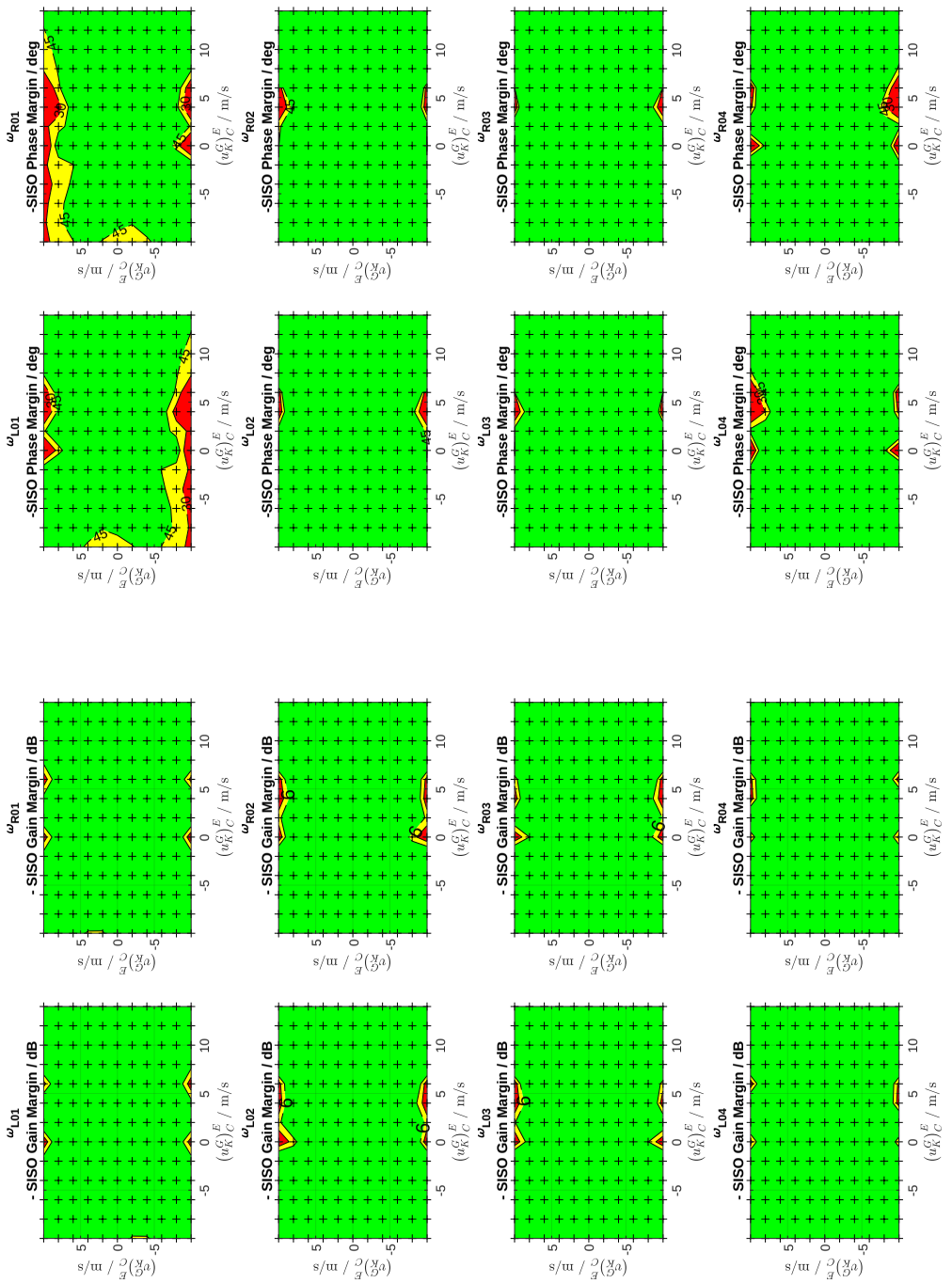


Figure 3.60: SISO gain (left side) and phase margin (right side) results at actuator AP over horizontal velocities

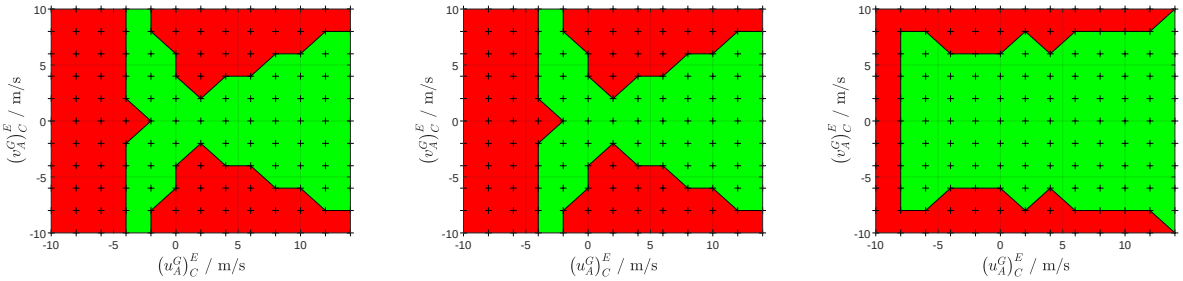
Control Allocation: As discussed, the control allocation analysis point is appropriate to analyze the robustness of the specific control axis of the closed-loop system. The SISO gain and phase margins are shown in Fig. 3.58. The SISO gain and phase margins at the vertical channel ΔT are much greater (GM$>>$6 dB and PM$>>$45 deg) than the required values. This is the case since the vertical channel has only a controller internal feedback loop from the actuator model and does not need to consider the flight dynamic and sensor model.
The SISO gain and phase margins at the roll channel ΔL are also greater as the required margins within the considered velocity range. Critical values occur at higher sidewards velocities (i.e. $> |4|$ m/s). This is caused by the dominant real pole in case of higher sideward velocities (see Fig. 3.45). This observation is visible in the gain margin as well as the phase margin contour plots. In these critical areas the SISO gain and phase margins decrease below the required margins. The crossover frequencies are the highest within the considered velocity range around the hover trim point. This results in the lowest delay margin around the hover trim point considering the corresponding phase margins.
The pole shifts over the velocity envelope influence also the SISO gain and phase margins of the SISO open loop transfer function by breaking the loop at the pitch channel ΔM. The SISO gain and phase margin reduces drastically at higher backward velocities (i.e. < -5 m/s) below the required margins, which is caused by the pole shift on the right half side of the complex plane at higher backward velocities (see Fig. 3.44). The yaw channel ΔN is less influenced by the horizontal velocities and the SISO margin requirements can be fullfilled. The associated nichols plots are presented in Fig. 3.55 at the control allocation analysis point. Thereby, only the corresponding open-loop transfer functions are drawn in different color, which do not cut the respective diamonds. This is the case for sidewards velocities $<$4 m/s and backward/forward velocities between > -4 m/s and $<$14 m/s. Higher forward velocities are not considered. At higher sidewards velocities the nichols diamond is cut and the required robustness margins cannot be achieved (marked in red). In consequence, the respective trim points do not fulfill the requirements.

Sensors: The SISO margins at the sensor analysis point (see Fig. 3.59) are qualitatively equivalent to the roll, pitch and yaw channel open-loop transfer functions at the control allocation analysis point. The SISO margins of the open-loop transfer function breaking the loop at the roll rate signal of the plant reduce at higher sidewards velocity at low forward/ backward velocities. Similarly, the margins reduce at simultaneous forward and sideward velocities.
By breaking the loop at the plant pitch rate channel the pitch rate SISO margins are calculated. The margins decrease significantly at higher backward velocities and at simultaneous higher forward and sideward velocities. In the yaw channel at the sensor analysis point, the

gain margin fulfills almost the considered trim grid completely. Only at high sidewards velocities the gain margin is below the required 6dB. The corresponding phase margin satisfies the required 45deg over the considered grid.
The nichols plots for the sensor analysis point are illustrated in Fig. 3.56. The nichols diamond is not cut over the considered envelope in the plot (see caption of Fig. 3.56), except for higher sideward velocities in the roll and pitch channel, marked in red. This corresponds to the findings in the phase margin plots of the roll and pitch channel and to the results of the control allocation analysis point.

Actuators: The results of the SISO gain and phase margins at the actuator analysis point are illustrated in Fig. 3.60. The respective nichols plots are shown over the envelope with sidewards velocity up to 4 m/s and forward/ backward velocity between -4 m/s and 14 m/s in Fig. 3.60.
The SISO gain and phase margin plots highlight the symmetry of the aircraft. The plots show similar results in terms of the air inflow for the left and the right propellers. This is significantly visible in the SISO phase margin plot for the actuator cut of the right front propeller ω_{R01} and the left front propeller ω_{L01}.
The SISO gain margin is almost fulfilled over the considered envelope. Only at high sidewards velocity the SISO gain margins falls below $6dB$.
The SISO phase margin plots indicate a margin loss at sidewards velocities at the front right (ω_{R01}) and front left (ω_{L01}) propeller. This is the case for the right propeller if an aerodynamic sideward velocity to the right is established. Equivalently, this is the case for the left propeller if an aerodynamic sideward velocity to the left is established.
The nichols plots in Fig. 3.57 show the same behavior. Nichols lines of the different trim points are the closest at the front left and right propellers. Note that in Fig. 3.57 a smaller flight envelope is shown as in the SISO gain and phase margin plots.

Analysis Point Comparison: The different analysis points can be compared over the considered envelope. Therefore, the results of the nichols plots are utilized. Corresponding contour plots are illustrated in Fig. 3.61. The area is marked in red color, if the nichols diamond is cut. The area is colored green, if the nichols diamond requirements are fulfilled. Again, the color is interpolated between the trim points. As it can be seen from the figure, the control allocation and the sensor analysis points achieve an equivalent envelope for horizontal velocities. The nichols diamond is cut at higher backward velocities and higher sideward velocities. The results at the actuator analysis point are different, as it can be seen in Fig. 3.61(c). Comparing to the other two analysis points, the actuator cut requirement with regard to the nichols diamond is less restrictive, except for simultaneous backward and sideward velocities (see Fig. 3.61).



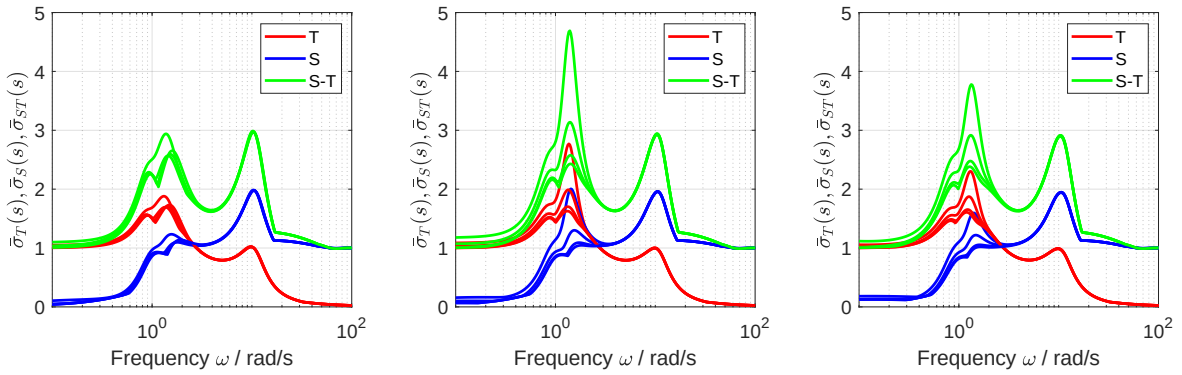
(a) Control allocation AP (b) Sensor AP (c) Actuator AP

Figure 3.61: Horizontal flight envelope under consideration of nichols plots at different analysis points: Nichols diamond is cut in at least one respective channel of AP (red), Nichols diamond is not cut in all respective channels of AP (green), considered trim point (+)

Subsequently in this section, the resulting envelope, which is gained by unification of the three envelopes, is compared to the MIMO margins and μ-analysis results.

3.10.7 Results of MIMO Gain and Phase Margins

Analogously to the SISO gain and phase margins, the presented MIMO gain and phase margins are applied at the three analysis points.
The different MIMO sensitivity transfer functions $\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$ are calculated using the built linear closed-loop system, introduced above under the consideration of the theory in Sec. 3.2. The functions provided in Ref. [88] were applied at the different analysis points to calculate the sensitivity and complementary transfer functions. These functions are used to calculate the maximal singular values $\bar{\sigma}(\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}), \bar{\sigma}(\underline{\boldsymbol{T}}), \bar{\sigma}(\underline{\boldsymbol{S}})$ over the frequency ω. The maximal singular values $\max_{\omega} \bar{\sigma}(\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}), \bar{\sigma}(\underline{\boldsymbol{T}}), \bar{\sigma}(\underline{\boldsymbol{S}})$ are used to calculate the respective MIMO gain and phase margins. Since three different analysis points are considered, three different sensitivity functions are available, and three margins (upper+lower gain margin and phase margin) exist, in total 27! contour plots over the horizontal flight envelope result.
The subsequent five pages illustrate the results of the upper and lower gain margin, and the phase margin over the horizontal velocity envelope. Fig. 3.63, Fig. 3.64 and Fig. 3.65 show the results at the control allocation analysis point, the sensor analysis point, and the actuator analysis point. The red and yellow color indicate that the respective margin is |GM|<3 dB or PM<30 deg. The areas in blue and green color visualize margins results above these limits. Note that the margins are calculated at the trim points marked as cross (+) and the margin results are interpolated between these crosses.
As elaborated in Sec. 3.5.4, since the MIMO margins of the complementary sensitivity function, the sensitivity function and the balanced sensitivity function are definite, the lowest conservative margins are relevant. The relevant lower and upper gain margin have been proven by the way analytically in Sec. 3.6. These lowest conservative margins are shown in



(a) $\left(u_A^G\right)_C^E$ =0 m/s (b) $\left(u_A^G\right)_C^E$ =2 m/s (c) $\left(u_A^G\right)_C^E$ =4 m/s

Figure 3.62: Singular value plots at control allocation analysis point with varying sidewards velocity from $\left(v_A^G\right)_C^E = -6$ m/s (line with highest peak) to $\left(v_A^G\right)_C^E$ =0 m/s (line with lowest peak)

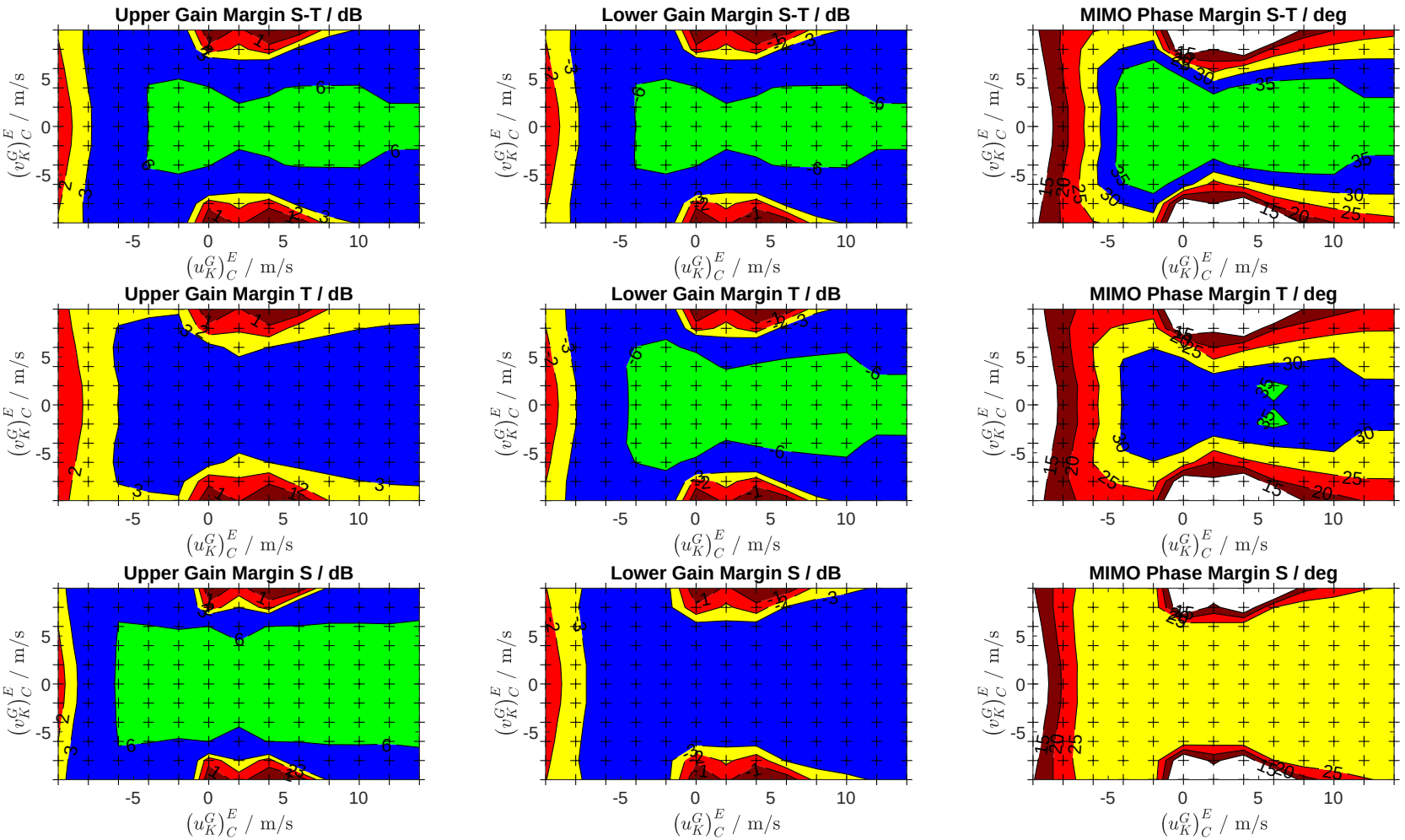
Fig. 3.66 for the three analysis points. These illustrated margins are the greatest margins for the upper gain margin and phase margin and the lowest margins for lower gain margin from the three sensitivity functions (i.e. $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$). Fig. 3.67 contrasts the margin results of the three analysis points in three contour plots.
The mentioned figures are detailed in the following. First, the results are evaluated over the horizontal flight envelope with respect to the MIMO robustness requirements, detailed in 3.1. Secondly in Sec. 3.10.7.2, the results are assessed with regard to the conservatism of the MIMO margins, proven above, and the SISO margins.

3.10.7.1 Evaluation with regard to Horizontal Flight Envelope of Closed-Loop System

Control allocation and sensor AP: Fig. 3.63 and Fig. 3.64 visualize the MIMO gain and phase margins at the control allocation and the sensor analysis point.
Both analysis point are analyzed in the same section since their result are similar. Respecting the lowest conservative results in Fig. 3.66, the margins |GM|$\geq$3 dB or PM $\geq$30 deg are achieved over a large area of the considered horizontal velocity envelope. The lower and upper gain margins with respect to the mentioned limits are thereby less restrictive than the phase margin (compare e.g. Fig. 3.66(c),(f),(i)). The phase margins at higher backward velocities (< -5 m/s) or at higher sideward velocity (> 6 m/s) are below PM$\geq$30 deg.
In this regard, Fig. 3.62 shows singular value plots at three different forward velocities for different sideward velocities using the control allocation analysis point. Basically, the figure represents a cut-out of the margin envelope plots. Their line peaks are relevant for the calculation of the MIMO gain and phase margins. The higher the sideward velocities, the higher is the maximum singular value peak. It is important to note, that $\bar{\sigma}(T(\omega \to 0)) \approx 1$ and $\bar{\sigma}(S(\omega \to \infty)) \approx 1$ needs to be given. This is the case in Fig. 3.62, which indicates inter alia,

that the command channel achieves the desired commands. This is also important for the margin results, otherwise it is possible that margins are not comparable.
The observations of the robustness drop at higher backward and higher sideward velocities match with the results of the SISO margins and can be explained inter alia by the pole shift of the linear flight dynamic model at these higher velocities (see Sec. 3.10.2 and Sec. 3.10.6).

Actuator AP: The margins are also calculated over the considered horizontal velocity envelope at the actuator analysis point. The results are shown in Fig. 3.65 for the three sensitivity transfer functions. In this regard, the lowest conservative margins are depicted in Fig.3.66 (b),(e),(h). The robustness margins decrease gradually at higher sideward and backward/forward velocity. However, an increase in backward velocity is significantly more sensitive to a robustness margin drop than forward velocity. An increase of sideward velocity is more sensitive than backward/forward velocities. In qualitative terms, the observations are analogously to the other two analysis points. However, the quantitative values of the actuator margins are reduced. This is explicitly visible in the gain and phase margin plots in Fig. 3.66(b),(e),(h). Especially, the blue area of the phase margin in Fig. 3.66(h), which fulfills PM$\geq$30 deg, is very restrictive. The underlying causes are discussed in the subsequent section.

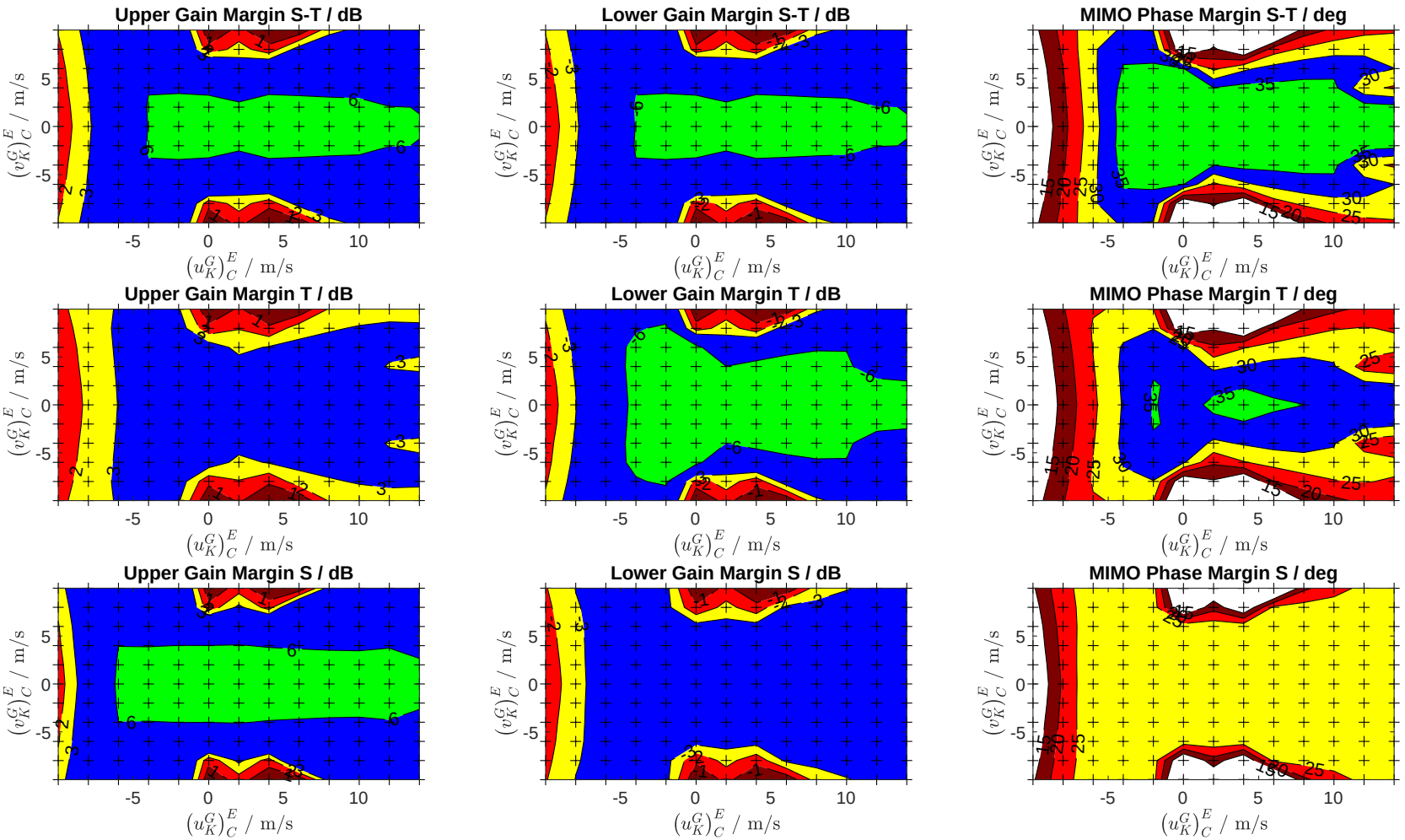


(a) Upper gain margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

(b) Lower Gain Margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

(c) Phase Margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

Figure 3.63: MIMO gain and phase margin over horizontal velocity grid at **control allocation analysis point**: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg

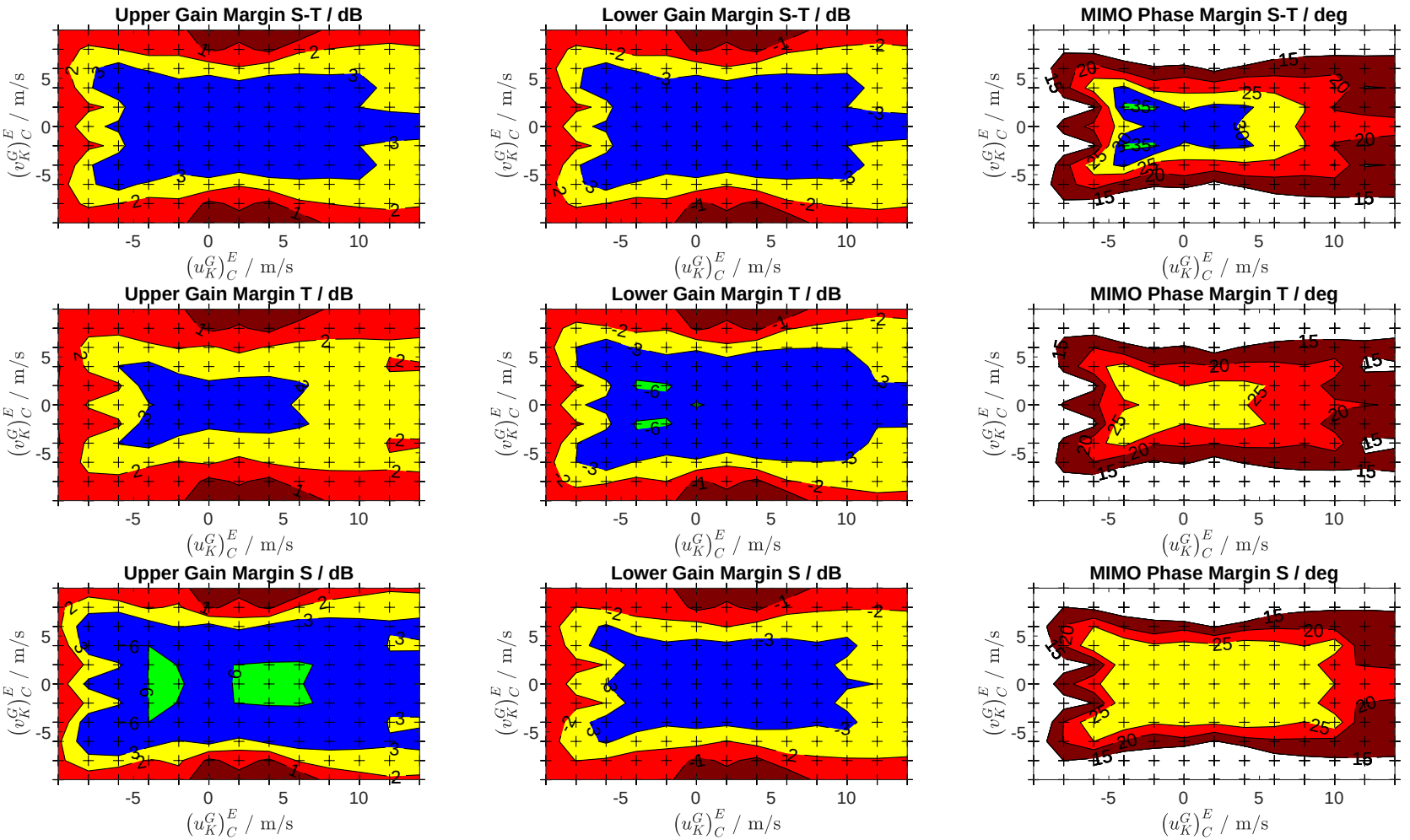


(a) Upper gain margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

(b) Lower gain margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

(c) Phase margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

Figure 3.64: MIMO gain and phase margin over horizontal velocity grid at **sensor analysis point**: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg

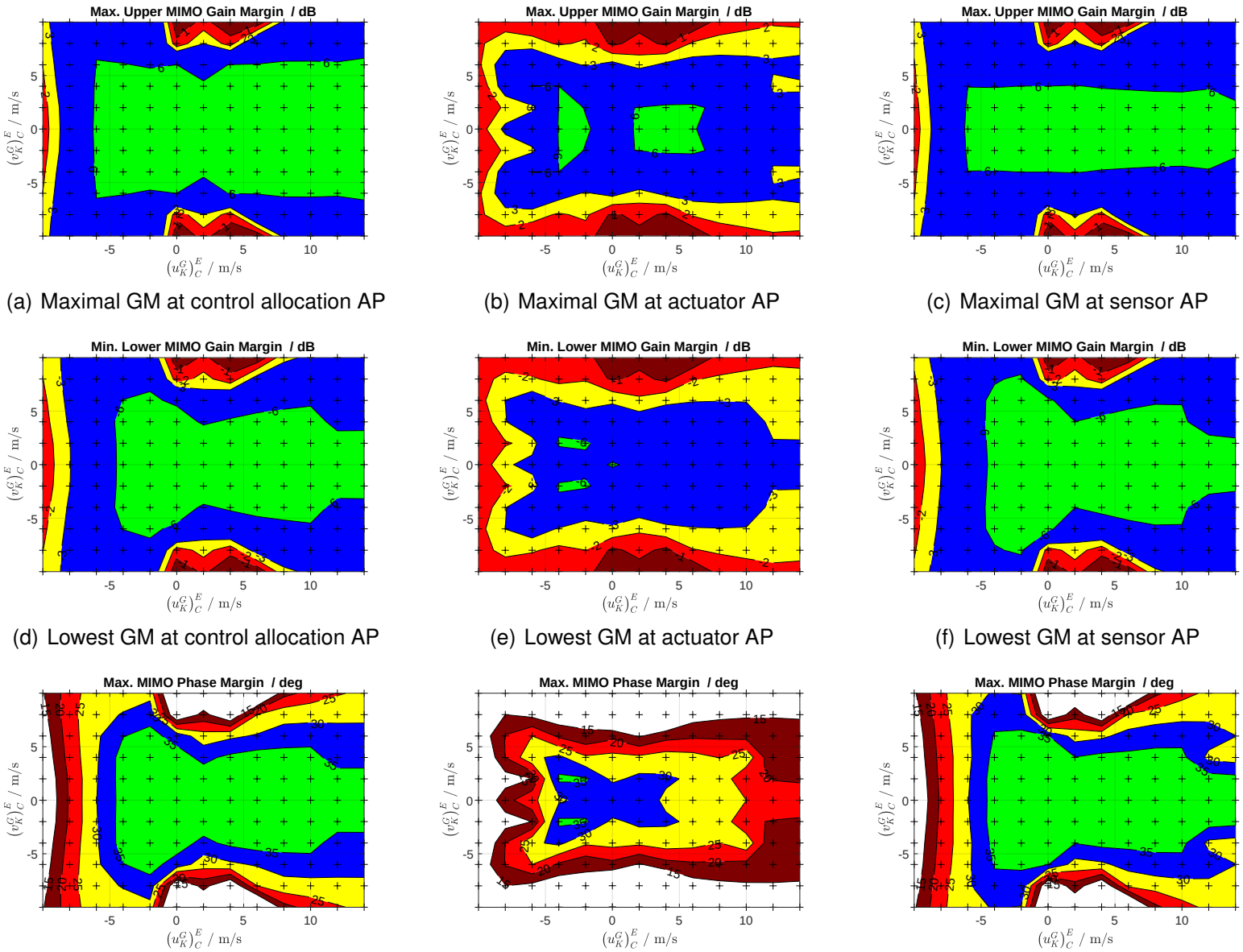


(a) Upper gain margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

(b) Lower gain margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

(c) Phase margin of $\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}, \underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}$

Figure 3.65: MIMO gain and phase margin over horizontal velocity grid at **actuator analysis point**: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg

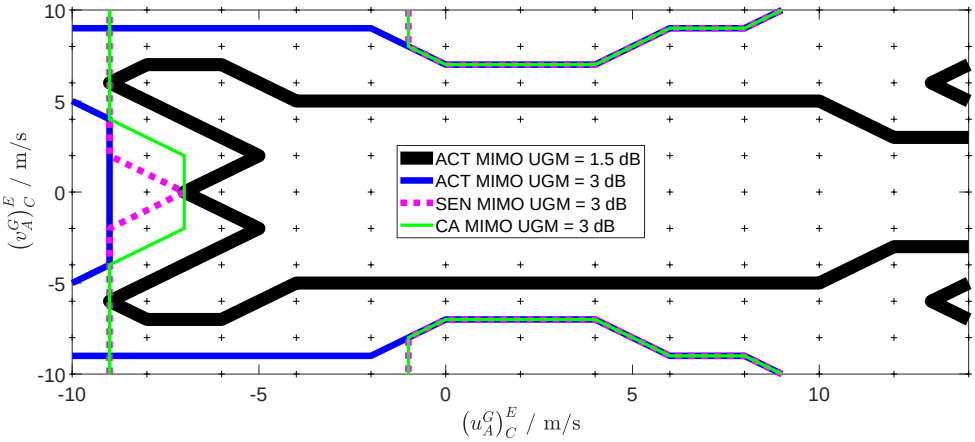


(a) Maximal GM at control allocation AP (b) Maximal GM at actuator AP (c) Maximal GM at sensor AP

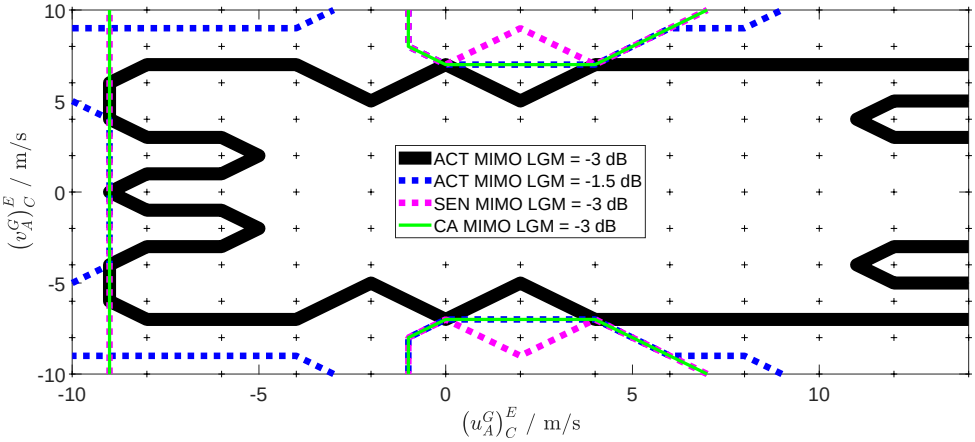
(d) Lowest GM at control allocation AP (e) Lowest GM at actuator AP (f) Lowest GM at sensor AP

(g) Maximal PM at control allocation AP (h) Maximal PM at actuator AP (i) Maximal PM at sensor AP

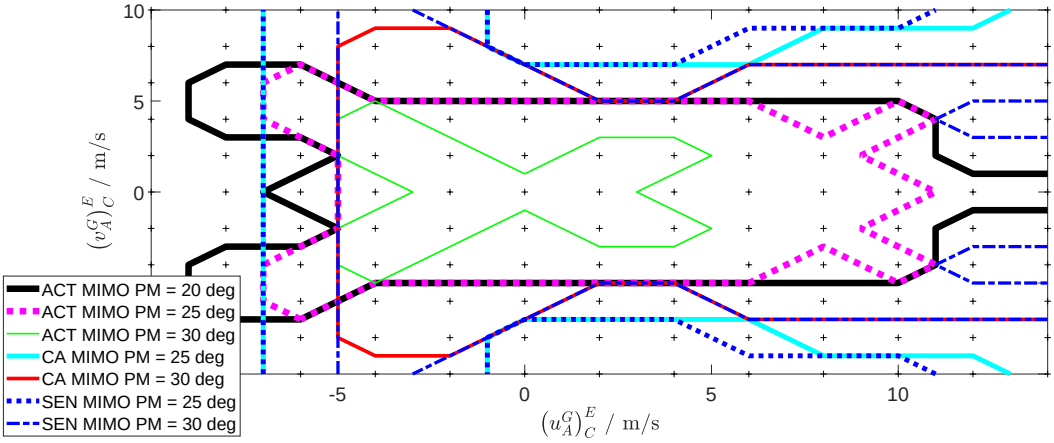
Figure 3.66: Lowest conservative MIMO gain and phase margins (i.e. maximal upper GM, minimal lower GM, maximal PM) at different analysis points (AP) over horizontal aerodynamic velocity envelope: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg



(a) Lowest conservative upper MIMO gain margin comparison



(b) Lowest conservative lower MIMO gain margin comparison



(c) Lowest conservative MIMO phase margin comparison

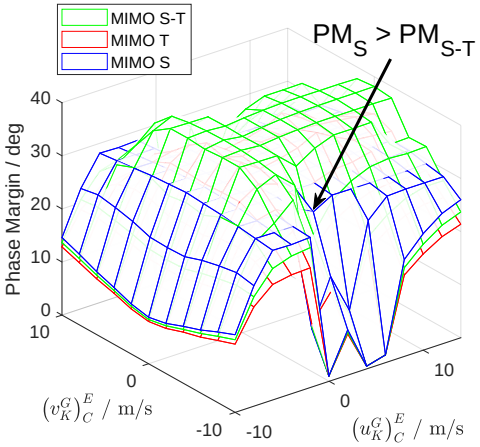
Figure 3.67: Comparison of MIMO gain and phase margins at different analysis points: only contour lines are shown which are relevant in respect of the requirement |GM|>3 dB and |PM|>30 deg

Analysis Point Comparison: The results of the three analysis points over the horizontal velocity flight envelope are compared in this section. In this regard, suitable contour plots are illustrated in Fig. 3.67, which shows subfigures for the upper (a) and lower (b) gain margin, and the phase margin (c). Each contour line represents thereby a specific margin value. Only contour lines are shown, which are relevant with respect to the requirement |GM|>3 dB and |PM|>30 deg.
The figures are directly derived from the lowest conservative margins, which are depicted in Fig. 3.66.
It can be seen in Fig. 3.67, that the contour lines |GM|=3 dB and |PM|=30 deg of the control allocation and the sensor analysis point match at most trim points. This becomes obvious for the lower and upper gain margin by comparing the green and magenta colored lines and for the phase margin by comparing the red with the blue-dashed contour lines. As the number of channels at the sensor and the control allocation analysis point are similar (i.e. three vs. four) and an analogy between the channels (i.e. ΔL,ΔM,ΔN and p,q,r) can be established, this observation can be explained. A difference can be identified between the sensor/control allocation analysis point and the actuator analysis point. The horizontal velocity flight envelope of the actuator analysis point, which fulfill |GM|$\geq$3 dB and |PM|$\geq$30 deg is significantly smaller in comparison to the other two analysis points. This holds for the upper, lower gain margin and the phase margin. However, considering the contour lines |GM|$\geq$1.5 dB and |PM|$\geq$25 deg for the actuator analysis point instead, the contour lines match significantly better with the sensor/control allocation analysis point contour lines of |GM|$\geq$3 dB and |PM|$\geq$30 deg. Since eight channels at the actuator analysis point and only three and four channels at the sensor/control allocation analysis point are in parallel, the different results of the analysis points are explainable.

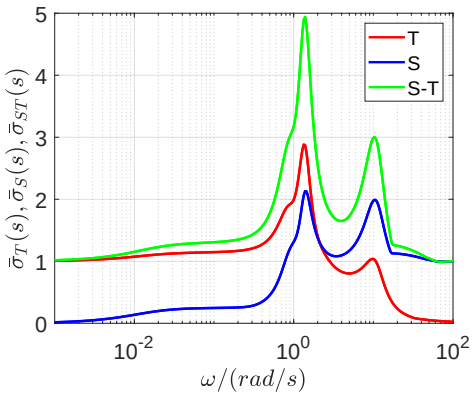
In consequence, the author believes that if MIMO robustness margin requirements are set up in future for eVTOLs, the respective location and the corresponding number of channels in parallel needs to be detailed in the requirement.
After the results of the μ-analysis are presented, Sec. 3.11 compares the SISO, MIMO and μ-analysis results. By that comparison, one gets deeper insight about the validity of the various margins.

3.10.7.2 Evaluation with respect to Elaborated Conservatism Proof

The results of the MIMO margins over the horizontal flight envelope can be also used to validate the analytic conservatism proof in Sec. 3.6 by the present application, which is discussed next. Therefore, Fig. 3.63-Fig. 3.65 can be contrasted columnwise. The blue area of the upper gain margin calculated by the sensitivity transfer function is always greater than the blue areas of the other two sensitivity transfer functions at the three analysis points.



(a) MIMO phase margin (PM) plot over the considered flight envelope: arrow shows exemplary trim point, at which the PM of $\underline{\boldsymbol{S}}$ is larger than the PM of $\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$



(b) Maximum singular value plots of $\underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}, \underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$ at the trim point $\left(u_A^G\right)_C^E$ =2 m/s, $\left(v_A^G\right)_C^E = -6$ m/s (marked in Fig. 3.68(a))

Figure 3.68: MIMO phase margin of $\underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}, \underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$ comparison at control allocation analysis point over the horizontal flight envelope

Similarly, the blue area of the lower gain margin calculated by the complementary sensitivity transfer function is always greater than the area of the other two transfer functions. These observations match with the results of the conservatism proof.
The MIMO phase margins match also with the results of the conservatism proof. The intention in Sec. 3.6.4 was to proof, that the phase margin of $\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$ is equal or greater than of $\underline{\boldsymbol{S}}, \underline{\boldsymbol{T}}$, based on the statement in Ref. [34]. The author spend months to find a global proof by the utilization of Fan's theorem. However, the reduced conservatism could be proven only for the condition in Eq. (3.121). The analysis of the present closed-loop system over the considered flight envelope has been conducted and the MIMO phase margins in Fig. 3.68(a) result. The figure confirms the findings of the proof. The phase margin of $\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$ is not always greater than $\underline{\boldsymbol{S}}$, at least for the presented closed-loop system, since at a few trim points the phase margin of $\underline{\boldsymbol{S}}$ is greater than of $\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$. Fig. 3.68(b) shows corresponding singular value plots for the marked trim point in Fig. 3.68(a). Inserting the maxima of the greatest singular values $\bar{\sigma}(\underline{\boldsymbol{S}})$ and $\bar{\sigma}(\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}})$ of the Fig. 3.68(b) in Eq. (3.121) demonstrates, that the phase margin of $\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$ and $\underline{\boldsymbol{S}}$ is not always less conservative, since at this trim point the condition is not fulfilled.
In summary, it can be stated, that the MIMO phase margin has the potential to reach less conservative margins, but it cannot be stated in general, that the MIMO phase margin of $\underline{\boldsymbol{S}}-\underline{\boldsymbol{T}}$ is always less conservative. This is shown by the example in Fig. 3.68(a).

3.10.8 μ-Analysis with limited uncertainty intervals

As alternative to the MIMO robustness margins, which analyze the system at specified analysis points, the μ-analysis elaborated in Sec. 3.7 and Sec. 3.8.1, takes modeled uncertainties in the entire closed-loop system into account. The μ-algorithm calculates thereby a factor the uncertainty intervals can be multiplied with such that the system is shifted on the stability/conformity borderline.
To the author's knowledge, the MIMO margins and the results of the μ-analysis can only be derived easily from each other, if the equivalent uncertainties of the MIMO margins are considered in the μ-analysis. The main advantage of the μ-analysis, however, is that it is able to consider detailed models of uncertainties. The μ-analysis can consider uncertainties which are identified by CFD, data sheets or test bench data. This is in contrast to the MIMO margins. The inserted uncertainties for their relation have not a direct relation to uncertainties identified/ measured in reality.
Nevertheless, the results of the MIMO margins and the μ-analysis together with the SISO margins can be compared and verified by each other. This is elaborated in Sec. 3.11.
Subsequently, the results of the μ-analysis are discussed first.

Considered conformity borderline: It is important to mention, that for the subsequent studies the conformity borderline for handling quality (HQ) Level 2 from the ADS-33 is used, which is illustrated at the beginning of this chapter in Fig. 3.1(b). This is valid in this context, since the considered controller shall function inter alia as an emergency control law for hovering, for which the HQ Level 2 conformity borderline is sufficient.
However, the HQ Level 2 conformity borderline distinguishes hardly with the stability borderline, since it matches at higher frequencies $\omega > 1$ rad/s (i.e. the imaginary axis of the complex plane).
In order to avoid misunderstandings in the following, the application of the μ-analysis considering the HQ Level 2 conformity borderline is called still the application of the classical μ-analysis, instead of advanced μ-analysis. This notation has been chosen to distinguish between the μ-analysis with limited interval and the new proposed probabilistic-μ approach.

Considered uncertainties with limited intervals: The 76 uncertainties of the flight dynamic model and the two uncertainties of the actuator model introduced in Sec. 3.10.2 are used for the analysis.
Since the uncertainties are modeled as stochastic Gaussian-distributed uncertainties, approximations are necessary to apply the classical μ-analysis as it can consider only uncertainties with limited intervals. Therefore, the stochastic Gaussian-distributed uncertainties are approximated by limited intervals which cover $[-3\sigma, 3\sigma]$ (which are three standard deviations σ) of each stochastic Gaussian distribution. The author is aware, that this is an

approximation, whose validity needs to be demonstrated.
On the one hand, the comparison between the classical μ-results and the calculated SISO and MIMO margins in the following section (see Sec. 3.11) underlines the eligibility of this approximation. On the other hand, taking stochastic uncertainties into account and giving definite statements about the stochastic robustness of the system is the million-dollar question. Therefor, the probabilistic-μ analysis has been introduced, which is also applied in the subsequent sections. The order of magnitude of each uncertainty is given in Tab. 3.5, since the exact values are confidential.

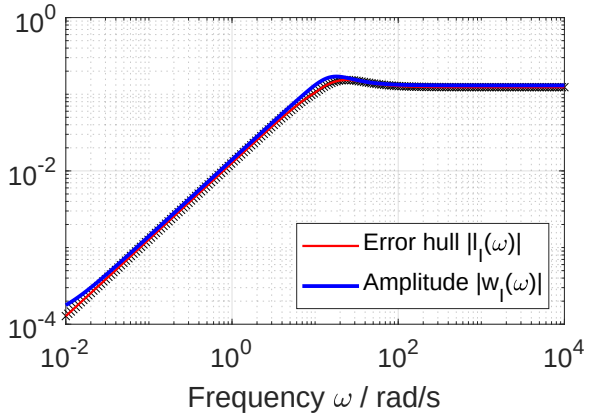
Two different studies are conducted. The first study presents in Sec. 3.10.8.1 the results of the classical μ-analysis under consideration of the parametric flight dynamic uncertainties in Tab. 3.5 as real uncertainties and complex uncertainties of the actuators. Thereby, the procedure in Sec. 3.8.3 is applied to calculate the worst-case dynamic behavior of the actuators.
The second study in Sec. 3.10.8.2 considers the actuator uncertainties also as real uncertainties and calculates the lower and upper bound of μ over the considered horizontal velocity flight envelope.

3.10.8.1 μ-Analysis with real and complex uncertainties

This section deals with the classic μ-analysis considering the parametric and real flight dynamic model uncertainties as well as complex uncertainties for the actuator models.
The complex uncertainties are multiplied by weighting functions, which cover the eigenfrequency and the relative damping uncertainties as one lumped uncertainty for each actuator. The weighting function is generated based on the procedure in Sec. 3.3.7.1 by varying over discrete parametric values of the uncertainties ω_{act} and ζ_{act}. The discrete values are thereby within the uncertainty interval $\left[-3\sigma, 3\sigma\right]$ of the respective parameter. The resulting weighting function is depicted in Fig. 3.69.

The objective of this section is to calculate the worst-case dynamics of the actuators which can bring the system to a specified HQ borderline (taking also the uncertainties of the flight dynamic model into account). Thereby, the computed weighting functions are utilized. The dynamics are calculated for the HQ borderline Level 2 as well as for a more restrictive and by the author defined HQ borderline at the trim point $(u_A^G)_C$ =0, $(v_A^G)_C = -4$ m/s (to demonstrate the potential for future applications).

Therefore, the μ lower and upper bounds are calculated using the entire established closed-loop system with the real uncertainties of the flight dynamic model and the weighted complex uncertainties. Note that the uncertainties listed in Tab. 3.5 of the flight dynamic model are



Weighting function:

$$w_I(s) = \frac{0.85s^3 + 25.51s^2 + 191.4s + 1.48}{6.518 \cdot 10^{-5}s^3 + 191s^2 + 2679s + 1.364 \cdot 10^4}$$

Figure 3.69: Weight approximation of eVTOL propeller actuator uncertainty

entirely modeled as independent and real parametric uncertainties and are embedded as scalar and multiplicative uncertainties into the model.

From the lower bound calculation the matrix $\underline{\boldsymbol{U}}_{lb}$ results. As elaborated above, with $1/\mu_{lb}\ \underline{\boldsymbol{U}}_{lb} = \underline{\boldsymbol{\Delta}}^{WC}$ the worst-case uncertainty matrix results. The core command *mussv* in Ref. [98] has been therefore used considering the respective conformity borderline. Note that the matrix $\underline{\boldsymbol{\Delta}}^{WC}$ has real value entries which correspond to the real uncertainties and complex value entries corresponding to the complex uncertainties (following the structure of the uncertainty matrix $\underline{\boldsymbol{\Delta}}$). Using the complex value entries in $\underline{\boldsymbol{\Delta}}^{WC}$, the worst-case transfer functions of the actuator models are calculated based on the procedure proposed in Sec. 3.8.3.

The results for the two different HQ borderlines are illustrated in Fig. 3.70 and Fig. 3.71. The figures present the following results:

- Sub-figure (a): The sub-figures in (a) represent the μ_Γ- plot over the frequency under consideration of the two different HQ borderlines. The two different HQ borderlines are illustrated in subfigure (b). Note that the second, by the author chosen HQ borderline is more restrictive, since it requires a higher performance to the closed-loop system in terms of e.g. the relative damping. The μ_Γ peak in Fig. 3.70(a) is below $\mu_\Gamma < 1$, indicating that the considered closed-loop system is sufficiently robust for the ADS-33 HQ Level 2 borderline to handle the considered uncertainties. In contrast, the closed-loop system cannot fulfill the author-defined HQ borderline. This is indicated by the peak values of μ_Γ being $\mu_\Gamma > 1$ in Fig. 3.71.
- Sub-figure (b): The sub-figures in (b) represent the pole map of the closed-loop system in the complex plane. The uncertainties are thereby replaced by the calculated worst-case uncertainties. Note that the deadtimes are replaced by pade approximations, otherwise the system can not be plotted in a pole map. In both figures, one complex

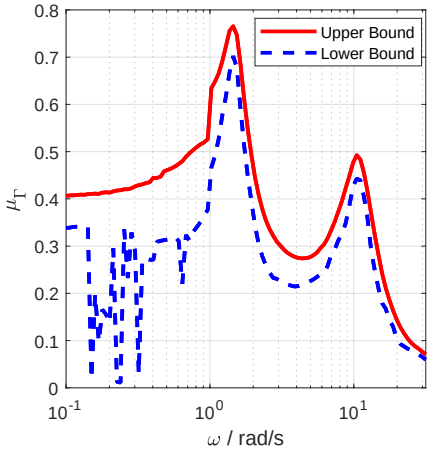
pole pair is shifted on the considered conformity borderline. This shows the validity of the presented approaches for a real-live system.

- Sub-figure (c): The sub-figures (c) show the computed worst-case actuator dynamics in comparison to the nominal model. The worst-case transfer function are computed according to the procedure in Sec. 3.8.3. Note that for the ADS-33 HQ Level 2 borderline, this worst-case dynamic is not covered by the presented uncertainty set, since the lower bound value of μ_Γ is $\mu_{lb} < 1$.
- Sub-figure (d): The sub-figures in (d) show selected responses of the entire closed-loop system after the step command $\phi_{cmd} = 1$ deg utilizing the calculated worst-case uncertainties. The step responses are in accordance with the pole map in sub-figure(b). Since in case of the ADS-33 HQ Level 2 compliance borderline, a pole pair is shifted on the imaginary axis, the oscillations in the bank angle and roll rate signal are undamped and sustain. This is in contrast to the HQ borderline in Fig. 3.71. There, the pole pair is shifted on the considered comformity borderline, such that the bank angle and roll rate signals are damped. Note that in the response plot only the deviations to the trim point are shown.

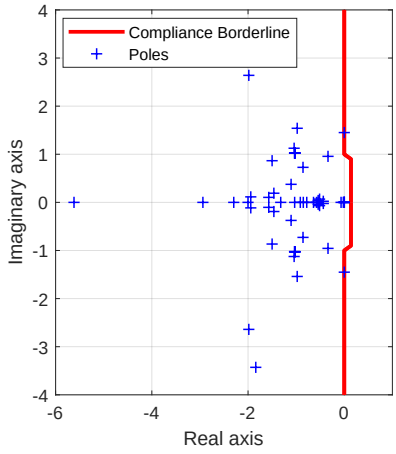
In order to guarantee a certain minimum level on handling qualities and to assure the pilot to control the vehicle and finish the mission safely, robust stability is not sufficient w.r.t. a safe operation of the aircraft in the presence of parameter variations. It is essential for pilots to control the aerial vehicle adequately also under uncertainties. Consequently, the pilot requires closed loop dynamics with poles in a certain area left of the imaginary axis in the s-plane. The μ_Γ-analysis with the presented procedure can calculate uncertainties such that specific handling qualities requirements are still fulfilled. This has been demonstrated within this section on a real-live example.

Uncertainty	Three standard deviations 3σ considering uncertainty interval $[-3\sigma, 3\sigma]$	Type
Uncertainties in the system matrix $\underline{A}$ of the linear flight dynamic model		
ΔX_u, ΔX_v, ΔX_w	$<25\%, >100\%, <30\%$	real
ΔY_u, ΔY_v, ΔY_w	$>100\%, <20\%, >100\%$	real
ΔZ_u, ΔZ_v, ΔZ_w	$<45\%, >100\%, <15\%$	real
ΔX_p, ΔX_q, ΔX_r	$<85\%, <65\%, <1\%$	real
ΔY_p, ΔY_q, ΔY_r	$<75\%, >100\%, <20\%$	real
ΔZ_p, ΔZ_q, ΔZ_r	$<1\%, <45\%, <85\%$	real
ΔL_u, ΔL_v, ΔL_w	$>100\%, <45\%, >100\%$	real
ΔM_u, ΔM_v, ΔM_w	$>100\%, >100\%, >35\%$	real
ΔN_u, ΔN_v, ΔN_w	$<25\%, <20\%, >100\%$	real
ΔL_p, ΔL_q, ΔL_r	$<15\%, >100\%, <35\%$	real
ΔM_p, ΔM_q, ΔM_r	$>100\%, <15\%, <85\%$	real
ΔN_p, ΔN_q, ΔN_r	$<20\%, >100\%, <20\%$	real
Uncertainties in the input matrix $\underline{B}$ of the linear flight dynamic model		
ΔY_{R01}, ... ,ΔY_{R04}	$<20\%, <25\%, <25\%, <20\%$	real
ΔY_{L01} ... ΔY_{L04}	$<20\%, <20\%, <20\%, <20\%$	real
ΔT_{R01},...,ΔT_{R04}	$<15\%, <15\%, <15\%, <15\%$	real
ΔT_{L01} ... ΔT_{L04}	$<15\%, <15\%, <15\%, <15\%$	real
ΔL_{R01} ... ΔL_{R04}	$<15\%, <15\%, <15\%, <15\%$	real
ΔL_{L01} ... ΔL_{L04}	$<15\%, <15\%, <15\%, <15\%$	real
ΔM_{R01} ... ΔM_{R04}	$<15\%, <20\%, <15\%, <15\%$	real
ΔM_{L01} ... ΔM_{L04}	$<15\%, <15\%, <15\%, <15\%$	real
ΔN_{R01} ... ΔN_{R04}	$<20\%, <25\%, <50\%, <30\%$	real
ΔN_{L01} ... ΔN_{L04}	$<20\%, <25\%, <45\%, <30\%$	real
Uncertainties in the actuator models		
$\Delta\zeta_{act}$	$\leq 6\%$	real or complex
$\Delta\omega_{act}$	$\leq 6\%$	real or complex

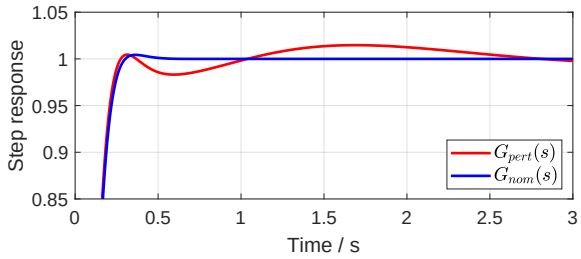
Table 3.5: List of considered uncertainties in μ-analysis with real and complex uncertainties at the trim point $(u_A^R)_C$ =0, $(v_A^R)_C = -4$ m/s for the two different studies. Note that this table shall give only the order of magnitude for the considered uncertainties, it does not give the exact values, since they are confidential. The values are directly derived from the stochastic uncertainties in Fig. 3.46-3.47. In case of μ-analysis with mixed uncertainties the actuator uncertainties are modeled as complex uncertainties with the weighting functions w_{L01} ... w_{R04}



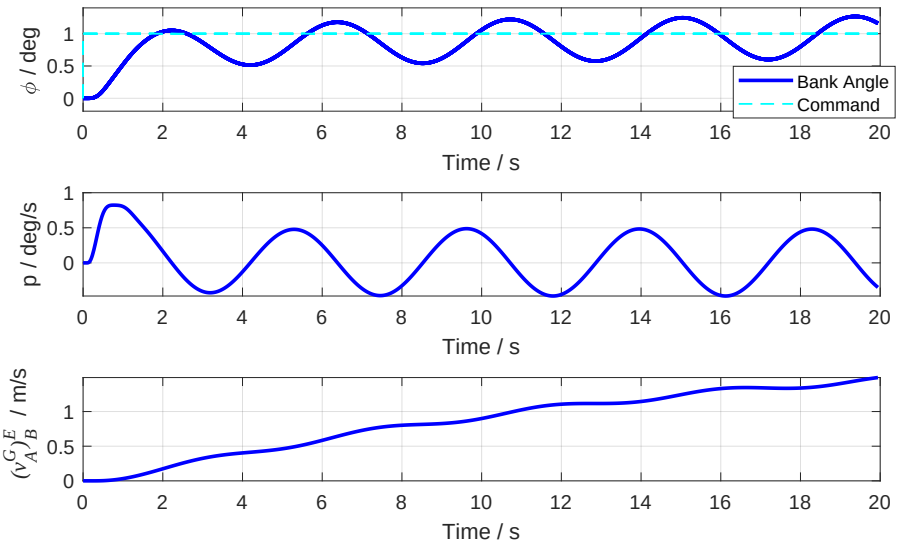
(a) μ_Γ-plot over the frequency considering the ADS-33 handling quality Level 2 conformity borderline



(b) Pole map in complex plane after worst-case uncertainty replacement, which shift one pole pair on the handling quality borderline. Closed-loop system is approximated by pade.

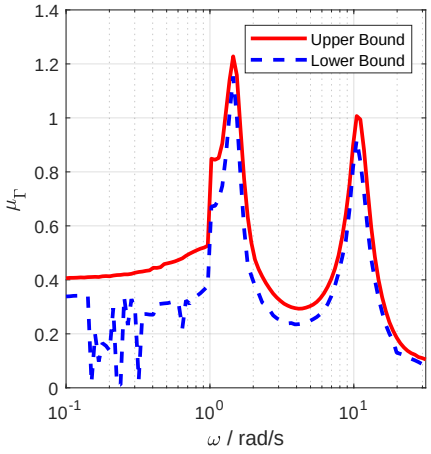


(c) Step response of nominal (i.e. G_{nom}) and computed worst-case actuator model $G_{pert} = G_{nom}(1 + w_I \Delta_{act}^{WC})$

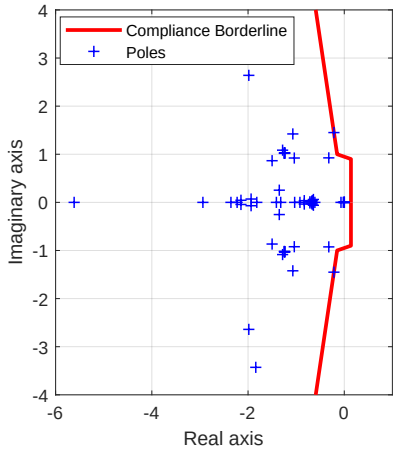


(d) Step responses of entire closed-loop system with computed worst-case uncertainties after bank angle command $\phi_{cmd} = 1$ deg: plot shows the state deviations from the trim point

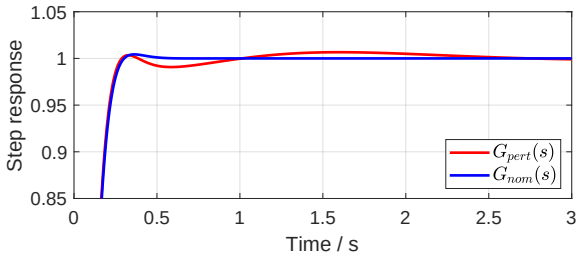
Figure 3.70: Closed-loop system behavior using computed worst-case uncertainties under consideration of real flight dynamic model uncertainties and complex actuator uncertainties and the **ADS-33 HQ Level 2 conformity borderline (see (b))** at the trim point $(u_A^R)_C$ =0, $(v_A^R)_C = -4$ m/s.



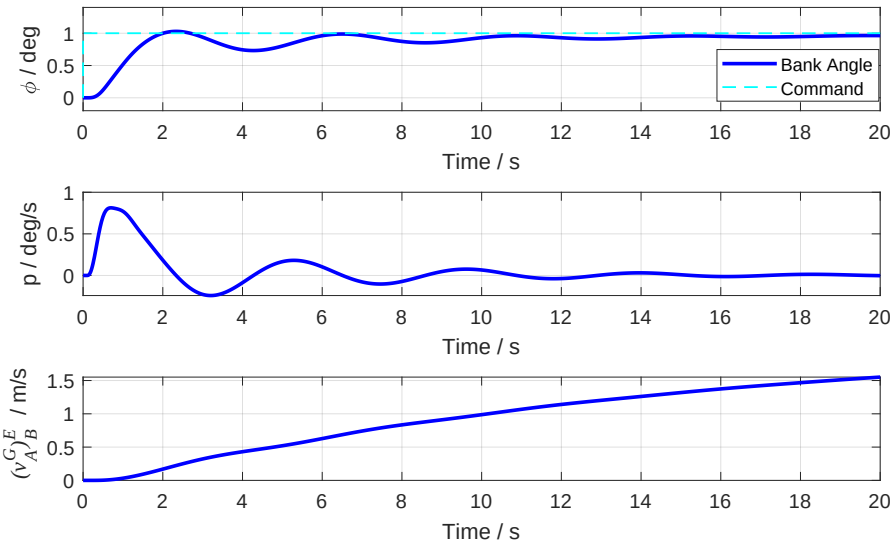
(a) μ_Γ-plot over the frequency considering a higher performance conformity borderline (see (b))



(b) Pole plot after worst-case uncertainty replacement, which shift one pole pair on the handling quality borderline



(c) Step response of nominal (i.e. G_{nom}) and computed worst-case actuator model $G_{pert} = G_{nom}(1 + w_I \Delta_{act}^{WC})$



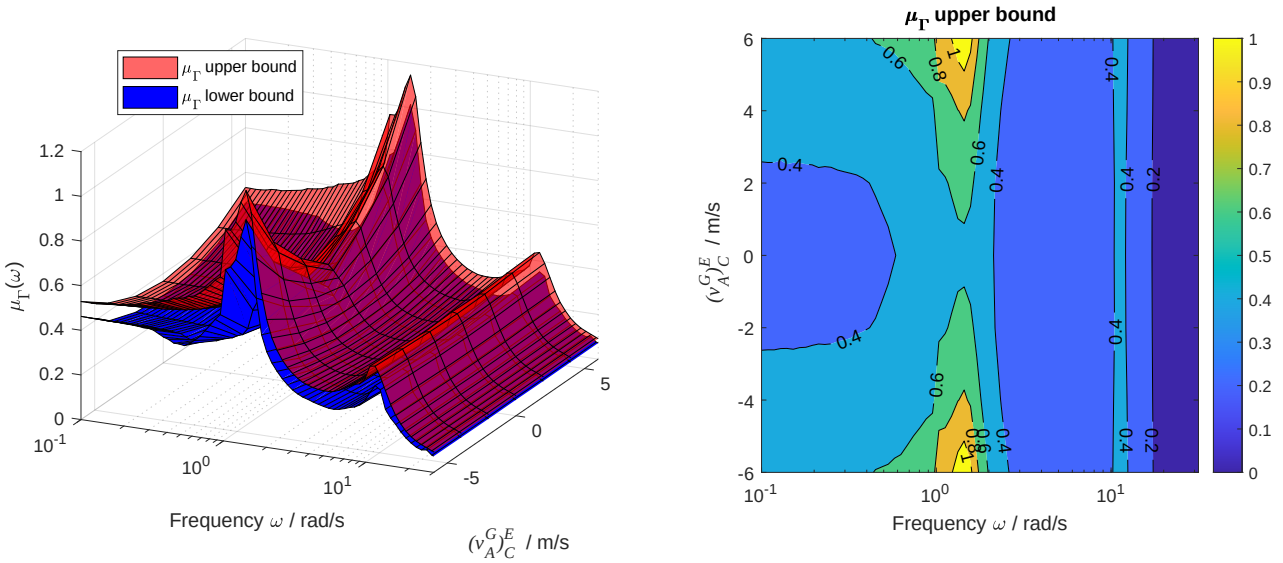
(d) Step responses of entire closed-loop system with computed worst-case uncertainties (with regard to **conformity borderline in (b)**) after bank angle command $\phi_{cmd} = 1$ deg: plot shows the state deviations from the trim point

Figure 3.71: Closed-loop system behavior using computed worst-case uncertainties under consideration of real flight dynamic model uncertainties and complex actuator uncertainties and the author-defined **higher performance HQ conformity borderline (see (b))** at the trim point $(u_A^R)_C$ =0, $(v_A^R)_C = -4$ m/s.

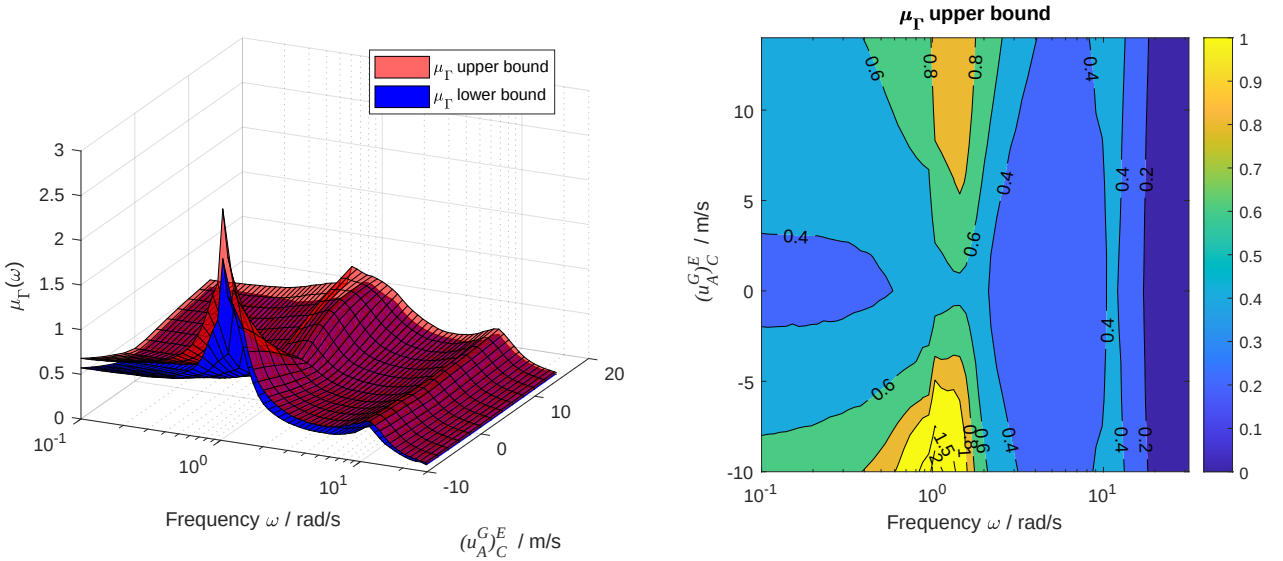
3.10.8.2 μ-Analysis with real uncertainties

Instead of using the lumped and complex uncertainties for the actuator models, one can also model the actuator uncertainties as real, parametric uncertainties. With the complex uncertainties worst-case dynamics in terms of transfer functions can be determined. On the other hand, as discussed, by the lumped uncertainties additional uncertainties are covered, which could lead to additional conservatism.
Therefor, within the rest of the chapter, the complex uncertainties are replaced by real, parametric uncertainties (i.e. the uncertainties of ω_{act} and ζ_{act} are now modeled as scalar, multiplicative uncertainties).
Under consideration of the uncertainties listed in Tab. 3.5, the lower and upper bound of the μ_Γ with respect to ADS-33 HQ Level 2 borderline has been calculated over the considered horizontal velocity flight envelope. The results over the horizontal velocity flight envelope are shown in Fig. 3.73(a) (μ_Γ-lower bound) (b) (μ_Γ-upper bound). The μ_Γ-boundaries might look familiar to the other robustness margins. In the figure, the peak value of the lower and upper bound at the respective trim points (+) are depicted. The lowest μ_Γ value occurs at the hover trim point (i.e. $(u_A^G)_C$ =0, $(v_A^G)_C$ =0 m/s). At higher sideward velocity both μ_Γ boundaries increase and reach $\mu_\Gamma \geq 1$ at about $(v_A^G)_C \approx$|6| m/s without any forward velocity. The same applies for higher backward velocities. The upper bound of μ_Γ is $\mu_\Gamma \geq 1$ at about $(u_A^G)_C \approx$ -5 m/s with $(v_A^G)_C$ =0 m/s. As for the SISO and MIMO margin discussed, this can be explained by the pole shifts, elaborated in Sec. 3.10.2.
Fig. 3.10.8.2 depicts the μ_Γ lower and upper bound results over the frequency and along the velocity $(u_A^G)_C$ or $(v_A^G)_C$. The upper bound results are in addition shown as contour plot on the right side.
Fig. 3.10.8.2(a) shows the result over the frequency along the velocity $(v_A^G)_C$ at the forward velocity $(u_A^G)_C$ =0. Fig. 3.10.8.2(b) illustrates in contrast at $(v_A^G)_C$ =0 the μ_Γ- results over the frequency ω along the backward/forward velocity. The three-dimensional plots over the frequency highlight the critical frequencies in comparison to different velocities. Based on the figure, the critical frequencies occur around $\omega \approx$1-2 rad/s. Another crest along the respective velocity, but at significant lower μ_Γ values, occurs at $\omega \approx$10 rad/s. By Fig. 3.10.8.2, the significant increase of μ_Γ at higher sideward/backward velocities becomes apparent.

In summary, the classical/ advanced μ-analysis is an enhanced approach to evaluate the robustness of a real-live closed-loop system under consideration of a complex uncertainty model by taking HQ borderlines into account.
The approach is even valid if the uncertainty is modeled as complex uncertainty, which can be expressed as dynamic transfer function. Furthermore, it can determine potential flight envelopes by analysing different trim points over a trim grid. Subsequently, the μ-results are contrasted with the calculated SISO and MIMO gain and phase margins results.

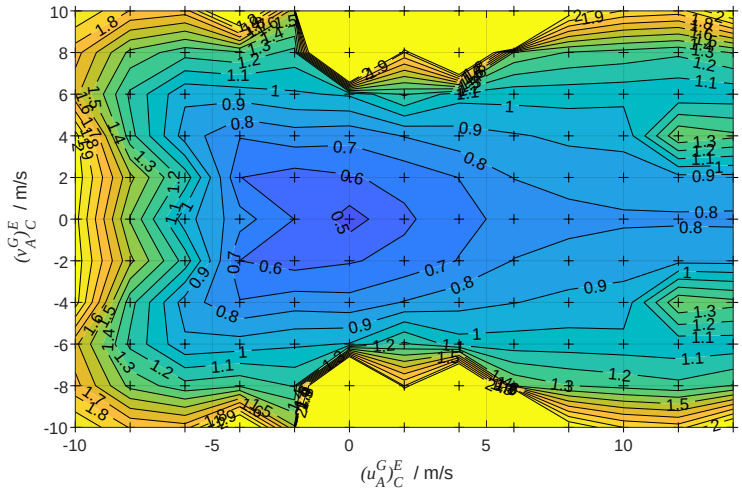


(a) μ_Γ boundary over the frequency under consideration of $[-3\sigma, 3\sigma]$ uncertainty interval for each real uncertainty Δ_i with $(u_A^G)_C^E$

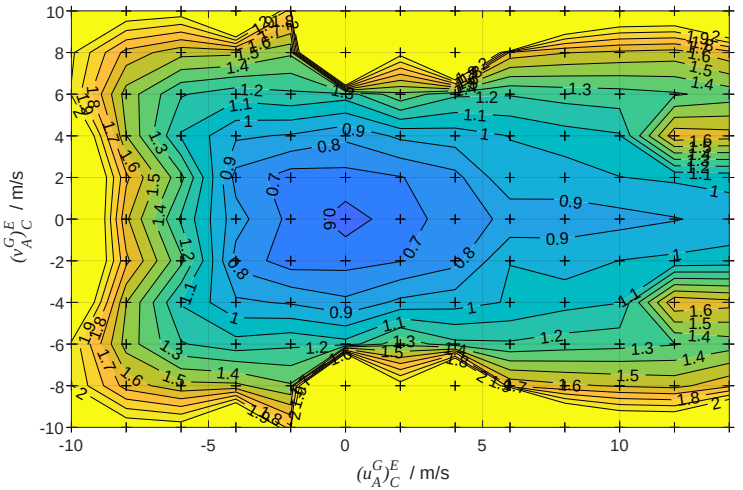


(b) μ_Γ boundary over the frequency under consideration of $[-3\sigma, 3\sigma]$ uncertainty interval for each real uncertainty Δ_i with $(v_A^G)_C^E$

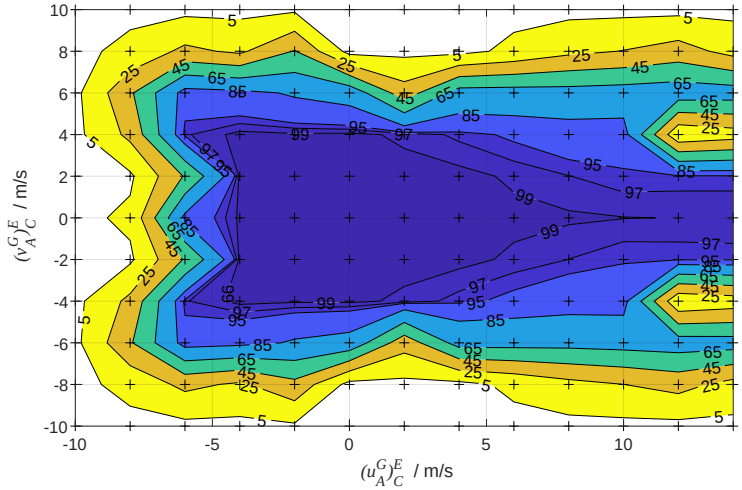
Figure 3.72: Results of advanced μ_Γ-analysis over the frequency under consideration of ADS-33 Handling Quality Level 2



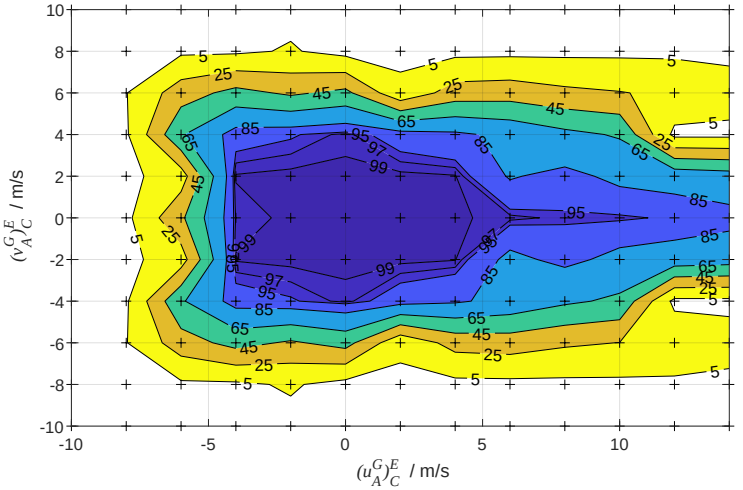
(a) Maximal μ_Γ-lower boundary under consideration of $[-3\sigma, 3\sigma]$ uncertainty interval for each real uncertainty Δ_i



(b) Maximal μ_Γ-upper boundary under consideration of $[-3\sigma, 3\sigma]$ uncertainty interval for each real uncertainty Δ_i



(c) Potential probability for conformity in % without probability maximization algorithm using lower μ-boundary



(d) Calculated probability in % for conformity without probability maximization algorithm using upper μ-boundary

Figure 3.73: Results of advanced μ_Γ-analysis and probabilistic μ-analysis without probability maximization over horizontal aerodynamic velocity envelope under consideration of ADS-33 Handling Quality Level 2

3.11 Comparison between SISO Metrics, MIMO Metrics and μ-analysis Results

In this section, the above elaborated results of the SISO Margins, the MIMO Margins and the μ_Γ results are compared. The contour plots in Fig. 3.74 are therefore created. The plots show the results of the three different robustness margins. After the figures are explained subsequently, the comparison is discussed.

SISO Nichols: The contour lines of SISO Nichols requirement are in the three sub-figures in Fig. 3.74 equivalent. The contour lines represent the unified set of the plots in Fig. 3.61, which shows the flight envelope of the horizontal velocity under consideration of the nichols diamond requirement at the three analysis points. Since the sets of the different analysis points distinguish, the unification of these sets (i.e. the unification of the green areas in Fig. 3.61(a)-(c)) is considered and represented as contour lines in the subsequent Fig. 3.74(a)-(c). The resulting contour lines show the horizontal velocity area, in which the nichols diamond is not cut in all channels at any analysis point.
MIMO robustness margins: Respective contour plots of the MIMO phase margins are shown only. Since the MIMO phase margins are more restrictive than the MIMO gain margins, concerning the horizontal velocity flight envelope, if the requirements PM$\geq$30 deg and |GM|$\geq$3 dB are considered, the MIMO gain margins are not shown in Fig. 3.74.
MIMO phase margin contour lines of different analysis points are drawn for each sub-figure. Note that three different contour levels are shown in each sub-figure. The results are directly derived from Fig. 3.74. Note that for the actuator contour lines lower contour line levels are chosen.
μ_Γ-analysis results: The results of the last section are plotted as contour lines in Fig. 3.74. Contour lines for the upper bound for $\mu^{ub} = 1$ and $\mu_{lb} = 1$ are drawn in each sub-figure. Note that in the three sub-figures the same μ-results lines are drawn. The contour lines are directly taken from Fig. 3.73(a)-(b).

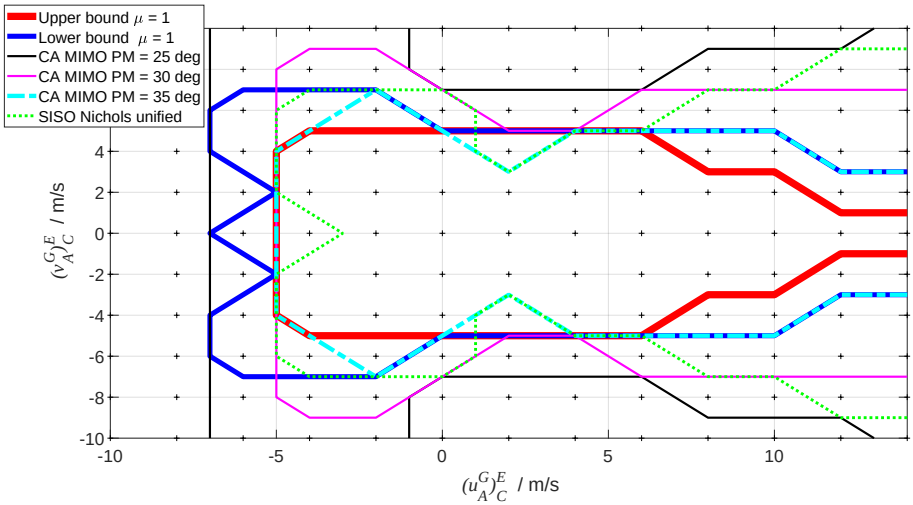
Discussion: As Fig. 3.74 shows, the μ_Γ upper bound is close to the μ_Γ lower bound in a tolerance of at least 2 m/s. The SISO Nichols contour line is always close to the μ_Γ lower bound within a tolerance of 2 m/s, except at higher forward velocities, i.e. $\left(u_A^G\right)_C^E >$10 m/s. In some cases, the SISO Nichols contour line match even with the μ_Γ-upper bound contour line (e.g. backward velocity $\left(u_A^G\right)_C^E = -5$ m/s).

The MIMO CA PM=30 deg contour line is also at least in a tolerance of 2 m/s to the μ_Γ-lower bound (again except at higher forward velocities). Apparently, the MIMO CA PM=35 deg is more restrictive, but the MIMO CA PM=35 deg contour line matches with the μ_Γ upper bound

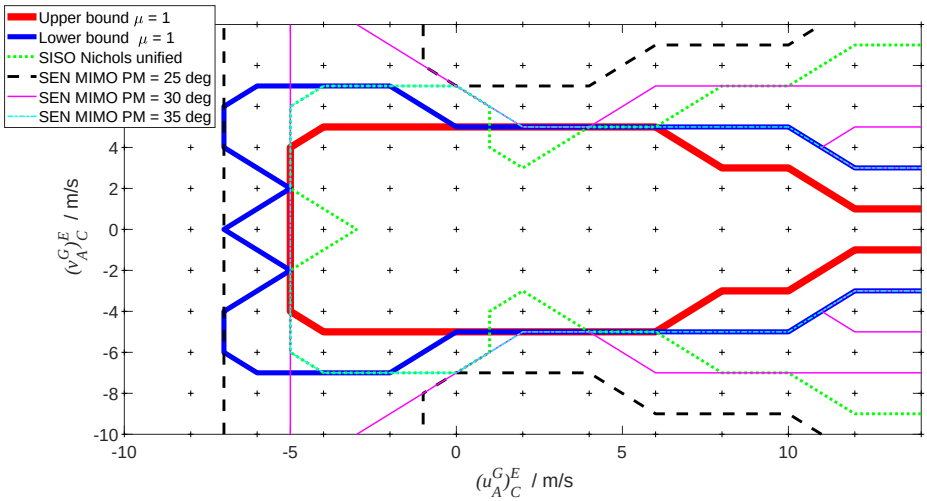
for $\left(u_A^G\right)_C^E \geq$4 m/s. At $\left(u_A^G\right)_C^E =$2 m/s the SISO Nichols and the MIMO CA PM=35 deg contour lines are more restrictive than the μ_Γ upper bound.
Below that velocity $\left(u_A^G\right)_C^E <$2 m/s the MIMO CA PM=35 deg contour line is at least between the μ_Γ lower and upper bound.
The same accounts for the SEN MIMO PM=30 deg and SEN MIMO PM=35 deg contour lines, but they differ at high sideward velocities $\left(v_A^G\right)_C^E \geq$|4| m/s, $\left(u_A^G\right)_C^E = -2$ m/s and they are not as restrictive as the MIMO CA contour lines at $\left(u_A^G\right)_C^E =$2 m/s.

The ACT MIMO PM=25 deg contour lines match with the μ_Γ-upper bound almost over the entire considered horizontal velocity envelope. Only at higher backward velocities it matches with the μ_Γ-lower bound contour line instead. The ACT MIMO PM=20 deg contour line is less restrictive but matches also over the entire considered horizontal velocity envelope. Even at higher forward velocities, it matches with the μ_Γ-upper bound contour line. At higher backward velocities the contour line is less restrictive compared to the other robustness margins. The ACT MIMO PM=35 deg contour line is too conservative since it covers an even more smaller area than the μ_Γ- upper bound.

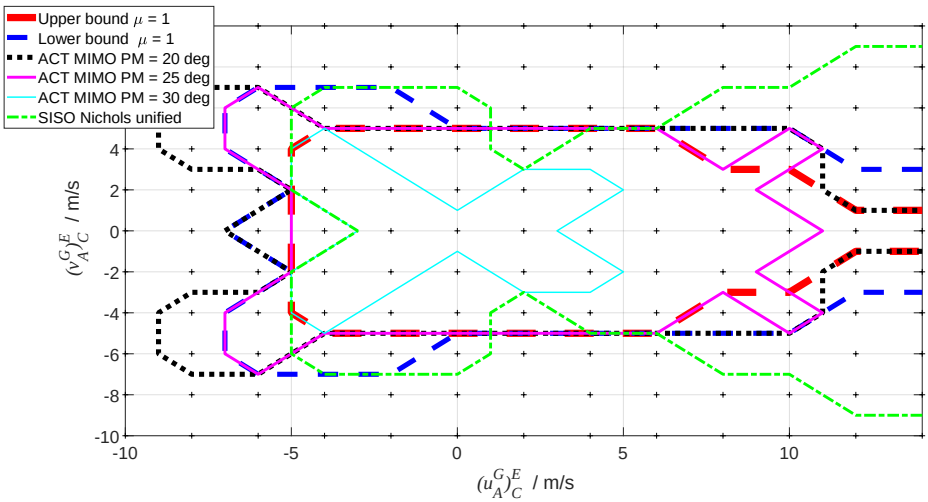
As conclusion, one can say that the horizontal velocity flight envelopes, in which discussed requirements (i.e. no Nichols diamond cut, MIMO PM$\geq$30 deg, MIMO |GM|≥ 3 dB, $\mu_\Gamma = 1$) are achieved, are qualitatively comparable for the most computed margins, but they don't match exactly.
Although, the three elaborated margins are based on different mathematical definitions and uncertainty considerations, this result is remarkable and verifies the validity of the three margin approaches. In terms of the MIMO margins, the number of channels at the analysis point need to be taken into account. For the present closed-loop system the MIMO phase margin requirement for the actuator analysis point PM$\geq$25 deg match with the μ_Γ contour lines, whereas MIMO PM=35 deg contour line of sensor and the control allocation analysis point match mainly with the μ_Γ contour line.



(a) MIMO phase margin contours at **control allocation (CA) AP** in comparison to SISO Nichols and μ_Γ-results



(b) MIMO phase margin contours at **sensor AP** in comparison to SISO Nichols and μ_Γ-results



(c) MIMO phase margin contours at **actuator AP** in comparison to SISO Nichols and μ_Γ-results

Figure 3.74: Comparison of nichols diamond requirement, possible MIMO margin requirements and the $\mu_\Gamma = 1$ boundary within the horizontal flight envelope of the considered eVTOL

3.12 Application of probabilistic-μ analysis

The proposed probabilistic-μ approach, introduced in Sec. 3.9, is applied to the presented closed-loop system of the eVTOL demonstrator. The analysis focuses on the full closed-loop system at selected trim points. In contrast to the classical and advanced μ-analysis, within the probabilistic μ-analysis, stochastic uncertainties are considered. The approach can give a lower bound value of the probability to achieve stability or a specified handling quality borderline. In the following, the set-up of the probabilistic-μ application is introduced first and finally the results are presented. The last subsection discusses further ideas and limiting factors to improve the algorithm in future work.

3.12.1 Algorithm Set-up

- **Uncertainties:** The uncertainties of the flight dynamic model and the actuator uncertainties as presented in Sec. 3.10.2 and Fig. 3.46-3.47 are considered as stochastic Gaussian-distributed uncertainties. Each uncertainty is embedded as scalar, multiplicative uncertainty. If the nominal value of the uncertain parameter is close to zero, additive uncertainties can be used instead. Following the proposed approach, the Gaussian-distributed uncertainties are assembled in a multivariate Gaussian distribution with the mean $\boldsymbol{\mu} = 0$ and a diagonal variance matrix $\underline{\Sigma}$. The diagonal entries of the matrix encompass the variances of each Gaussian distribution of the uncertainties presented in Fig. 3.46-3.47. Thus, the identified deviations to the nominal parameter are modeled as uncertainty such that the mean of the distribution is equal to zero (see Eq. (3.181)). Thus, the random variable vector $\boldsymbol{\Delta}$ encompasses in total 78 dimensions. It contains the 76 uncertainties of the flight dynamic model. It is assumed that each actuator has the same, but uncertain dynamic behavior. As result, in addition to 76 uncertainties of the flight dynamic model two actuator uncertainties are added, such that in total the random vector $\boldsymbol{\Delta}$ encompasses 78 dimensions. In consequence, the considered multivariate Gaussian distribution is defined.

- **Optimization variables:** As elaborated in Sec. 3.9, the transformation variables from the mapped uncertainty space to the true uncertainty space are used as optimization variables in the outer optimization of the probabilistic-μ approach. For the presented closed-loop system the following transformation is chosen (see Eq. (3.178)):

$$\boldsymbol{\Delta} = \boldsymbol{p}^0 + \underline{\boldsymbol{T}}\,\tilde{\boldsymbol{\Delta}} \tag{3.222}$$

with the shifting vector $\boldsymbol{p}^0 \in \mathbb{R}^{N\times 1}$ and the transformation matrix $\underline{\boldsymbol{T}} = \mathrm{diag}\{t_1, ...t_N\}$, $t_i \in \mathbb{R}$ and $N = 78$. Thus, the mapped uncertainties are only shifted and/or stretched. In total,

the outer optimization problem encompasses 156 optimization variables, which are accumulated in the vector $\boldsymbol{x} = \left[t_1, ..., t_N, p_1^0, ..., p_N^0\right]$. More information can be found in this regard in the result section in Sec. 3.12.2.

- **Solver:** The *surrogate* optimization solver from Ref. [104] is used for the presented closed-loop system. The solver is designed to solve global optimization problems. Note that in the case for the probabilistic-μ approach, it is not essential to find necessarily the global maximum (minimum), since the algorithm gives always a lower bound on the probability to achieve stability/conformity. In test trials for the presented optimization problem the solver *fmincon* from Ref. [111] has been also applied. However, the solver did not reach the results from the *surrogate* solver, and sometimes got stuck in local minima.

- **Solver settings:** The *default* settings of the solver settings are applied. The minimum number of random sample points to create at the start of a *surrogate* creation phase (*MinSurrogatePoints*, see Ref. [104]) is set to the lowest possible number, which is the number of optimization variables plus one. The maximum of function evaluations was set to 2000 function evaluations.

- **Initial points:** Initial points can be given to the *surrogate* optimizer, which evaluates the objective function at the initial points first after start. Some initial points are defined (otherwise the solver uses random samples). Therefor, a reference point is chosen first. The reference point is represented by the vector $\boldsymbol{x}_0 = \left[3\sigma_1, ..., 3\sigma_N, \boldsymbol{0}^{N\times 1}\right]^T$. The standard deviation $\sigma_i \in \mathbb{R}$ corresponds to the i-th uncertainty, shown in Fig. 3.46 and Fig. 3.47 and listed in Tab. 3.5 with $i = 1, ..., N$. In the surrounding space of the reference point, a grid of additional points is computed based on the reference point $\boldsymbol{x}_0$. One dimension of the reference point is multiplied by two and three individually, whereas the values for the remaining dimensions of the reference point are kept. This is done for every non-zero dimension of the reference point $\boldsymbol{x}_0$. Equivalently, 10^{-2} is added dimension-wise to the zero entries of the reference point by keeping the other entries. By that approach, suitable points are given to the solver at the start, since the reference point appeared to give a good initial probability.

- **Lower and upper bound of the optimization variables:** The *surrogate* optimizer requires also lower and upper bounds for the optimization variables. The lower bound is set for each stretching optimization variable (i.e. $t_1, ..., t_N$) to $10^{-5}\sigma_i$. The upper bound for each stretching variable is set to 1. The allowed interval for the vector entries $\boldsymbol{p}^0$ are set elementwise to $\left[-10^{-2},\ +10^{-2}\right]$.

- **Probability calculation:** As proposed in Sec. 3.9, the multivariate Gaussian distribution of the mapped uncertainties (see Eq. (3.182)) is used, to integrate over the

hypercube $\tilde{\mathbb{B}}'_{\Delta}$ which was stretched by the reciprocal of μ^{ub}. Note that in the present case the hypercube has 78 dimensions.

- **Conformity borderline:** For the presented calculation the ADS-33 HQ Level 2 line is used as conformity borderline. As a result, the probability is calculated, which is definitely a lower bound to achieve the HQ Level 2. Note that the ADS-33 HQ Level 2 conformity borderline matches at frequencies $\omega >$1 rad/s with the stability borderline.
- **Frequency range:** The frequency range for the μ-algorithm is set to $\omega = \left[0.1,\ \ 10^{1.5}\right]$. Thereby, the μ-upper bound is evaluated at 70 discrete and logarithmic-distributed frequencies. Within the probability maximization, the calculation of the lower bound was set to a very cheap and fast algorithm to save computational time. The number of points and the frequency range are compromises between calculation accuracy and calculation time.

3.12.2 Results

3.12.2.1 Probability over grid without maximization

Before the presented probability maximization algorithm is applied, the probability is calculated over the entire considered horizontal flight envelope without outer optimization of the presented probabilistic-μ approach. The lower and upper bound of μ is therefor calculated using the above selected reference point x_0. Multiples of the standard deviations as stretching variable values turned out to give high initial probability results. The μ-bounds are used to stretch the hypercube and calculate the probability to achieve conformity. The results are illustrated in Fig. 3.73(c)-(d) using the peak values of the lower and upper bound of μ. The probability results are related to the corresponding peak values of the μ-lower and μ-upper bound. The respective lower and upper bound values of μ are drawn in Fig. 3.73(a)-(b). Note that the definite lower bound is the probability, which is calculated using the μ-upper bound. The probability result using the μ-lower bound gives only an indication, what probability would be achieved, if the true structured singular value is equal to the possible, lowest conservative value. As visible in Fig. 3.73(d), for trim points near the hover trim point (i.e. $|(v_A^G)_C|$ =2 m/s, $(u_A^G)_C = -$2,...,4 m/s) the definite lower bound of the probability for conformity is greater than $>$99%. It can be thus said definitely, that the considered uncertain closed-loop system achieves conformity with a probability of more than $>$99% at these trim points. Considering the μ-lower bound the area of the contour line $>$99 % is even greater (see Fig. 3.73(c)). As it can be seen, the probability value decreases with increasing μ-value peak. This is reasonable, since with a higher μ-value, the hypercube, over which the multivariate Gaussian distribution is integrated, is less stretched and thus has less volume. This leads to reduced probability values at higher sideward and backward/forward velocities. The

$(v_A^G)_C$ / m/s at $(u_A^G)_C$ =0	-6	-4	-2	0
Probability without max. / %	52.235534	97.966891	99.952904	99.9995244
Probability with max. / %	72.416640	98.722140	99.972677	-

$(u_A^G)_C$ / m/s at $(v_A^G)_C = 0$	0	4	8
Probability without max. / %	99.9995244	99.498292	96.197408
Probability with max. / %	-	99.722699	98.071990

Table 3.6: Definite lower bound of the probability to achieve conformity without probability maximization and with probability maximization (i.e. with outer optimization of the bi-level optimization problem) for selected trim points.

exactly calculated lower bound values of the probability are given in Tab. 3.6 for some selected trim points. In the following, the probabilistic-μ approach is applied to increase the lower bound for conformity at selected trim points.

3.12.2.2 Selected trim points with maximization:

The lower bound for conformity is maximized by using the presented probabilistic μ-approach, which has been presented in Sec. 3.9. The selected trim points are given in the Tab. 3.6.
Using the set-up, explained above, the bi-level optimizations are conducted. Corresponding *surrogate* optimization plots are illustrated in Fig. 3.75. The plots are generated by the *surrogate* optimizer and show mainly the trend of the solver's best value over the number of function evaluations. Since the solver is implemented to search for a minimum the negative probability is used as objective function. Note that the maximum number of functions evaluation is set to 2000.
The *surrogate* optimization algorithm consists of a construction phase and a minimum search phase. The construct phase creates random points (together with the initial points), at which the objective function is evaluated. The number of these points is defined by the option *MinSurrogatePoints*. A *surrogate model* is created by using radial basis functions. In the search for minimum phase the solver generates hundreds or thousands of pseudorandom vectors around the so-called *incumbent* point, which represents the point which achieves the lowest objective function.

Under consideration of the incumbent point the solver generates several sample points. The adaptive point is then the point with the lowest merit function value. The objective function value is computed at the adaptive point, and the *surrogate* model is updated with the respective function value. If the adaptive point achieves a lower value as the incumbent value, the solver considers the adaptive point as incumbent. If this is not the case, the search

was not successful and the incumbent does not alter. Detailed information can be found in Ref. [112, Mathworks].

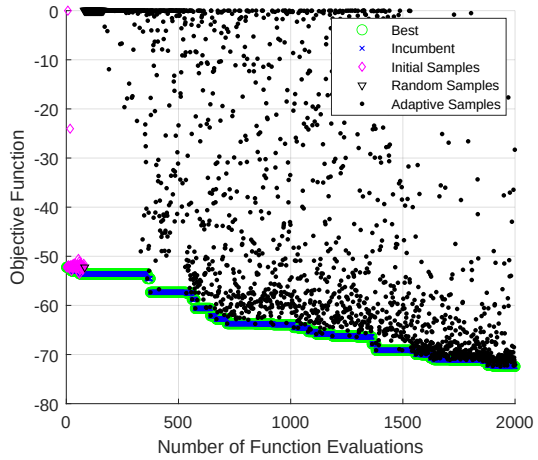
The random and the initial points are visible at the start of each optimization in Fig. 3.75 (marked as black triangle or magenta diamond). The adaptive points are shown as black dots. The incumbents are marked as blue crosses. Since *surrogate* reset is not conducted within the 2000 function evaluations, the best value (marked in green) is equivalent to the incumbent.

As it is illustrated in Fig. 3.75, at the start the solver is in need of hundreds of function evaluations to build the *surrogate* model. After about 300 function evaluations or more the adaptive samples find solutions which are below the current best value of objective function (i.e. the "negative" probability). With increasing number of function evaluations, the objective function is minimized. The lower bound of the probability to achieve conformity is increased in general at every analyzed trim point, however the magnitude of the increase depends on the respective trim point.
The achieved increase is visible in each sub-figure in Fig. 3.75 and in Tab. 3.6. At the trim point $(u_A^G)_C$ =0, $(v_A^G)_C = -6$ m/s the probability could be increased by 20 %, although the achieved 70 % might not be sufficient. As stated, these probabilities indicate definite lower bounds. With less sideward velocity, at the trim point $(u_A^G)_C$ =0, $(v_A^G)_C = -4$ m/s, an increase of approx. 0.7 % is reached within the number of function evaluations, such that the lower bound achieves $\approx$ 98.7 %. This increase might be small, however, the probability result marks an improved definite lower bound.
Similar results are achieved at other trim points, at which the initial probabilities are >95 %. For example, at the trim point $(u_A^G)_C$ =8 m/s, $(v_A^G)_C$ =0 m/s, the probability of the lower bound could be increased from approx. 96.2 % to approx. 98.1 %.
As the results indicate, the probabilistic-μ approach improves the lower bound of the probability to achieve conformity. This is demonstrated on a complex and real-live uncertain closed-loop system. The closer the optimizer gets to the 100 %, the more difficult it is to improve the lower bound of the probability. If the system achieves sufficiently small μ peak values, sophisticated lower bound for the probability are achieved, which can be used in practice. However, the author assumes, that much more potential lies within the approach, since a few conservatism factors have been not yet considered. In the next section, these reasons for conservatism are elaborated in more detail and suggestions for future work are given.

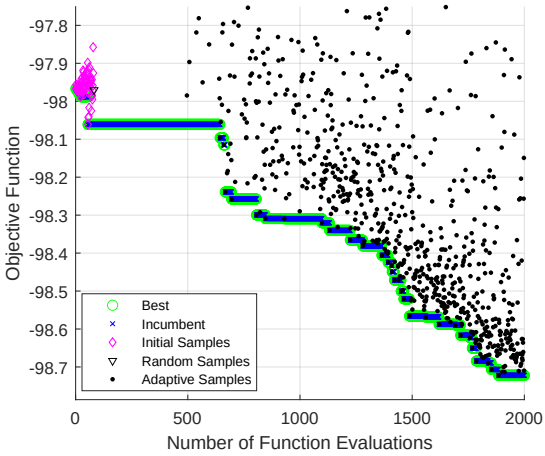
Calculation time: The optimizations are conducted on default desktop and laptop computers. The desktop computer is equipped with a *Intel Core i7-4790* processor with eight CPUs and 16 GB RAM. The evaluation of one cost function took thereby about 100 s. The

MATLAB Parallel Computing Toolbox Ref. [113] has been used to reduce the time for one function evaluation within the optimization process, such that one function evaluation took approx. 33 s in average of 1000 evaluations. If the function is evaluated 2000 times, this is equivalent to 18 h. The author agrees, that the algorithm has not been yet implemented from a code-efficient perspective rather than functional perspective. However, limiting functions are the evaluation of the μ-algorithm and the integration of the probability function.

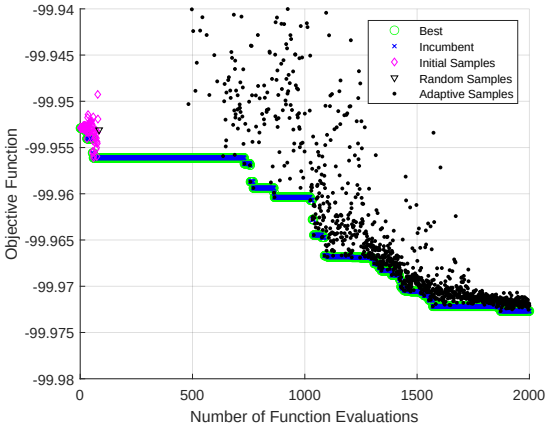
Number of optimization variables: As said, the number of possible optimization variables is limited within the thesis to the shifting vector $\boldsymbol{p}^0$ and the stretching variables of the uncertainties. However, this number can be increased in future work, since it limits also the shapes of possible hypercubes, over which the probability density function is integrated. The number of variables has been increased experimentally, but feasible solutions, which improve the lower bound could not be found within 2000 function evaluations.



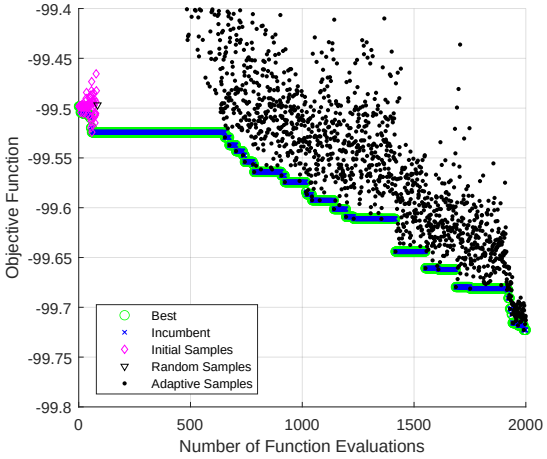
(a) $(u_A^G)_C$ =0 m/s, $(v_A^G)_C = -6$ m/s



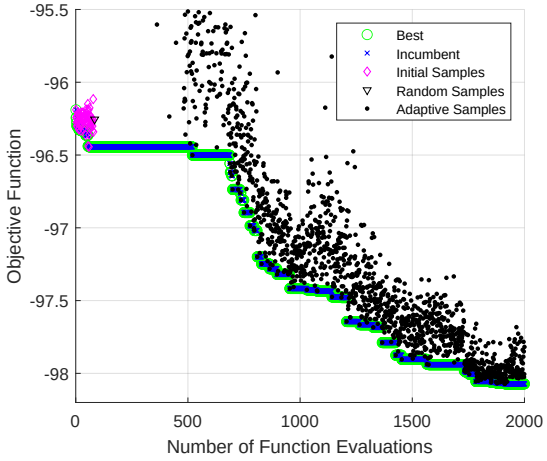
(b) $(u_A^G)_C$ =0 m/s, $(v_A^G)_C = -4$ m/s



(c) $(u_A^G)_C$ =0 m/s, $(v_A^G)_C = -2$ m/s



(d) $(u_A^G)_C$ =4 m/s, $(v_A^G)_C$ =0 m/s



(e) $(u_A^G)_C$ =8 m/s, $(v_A^G)_C$ =0 m/s

Figure 3.75: *Surrogate* optimization plots to maximize the lower bound of the probability to achieve ADS-33 HQ Level 2 conformity under consideration of the full uncertain linear closed-loop system with 78 stochastic, Gaussian-distributed uncertainties at selected trim points. Note that the optimization is set up as a minimization problem why the objective function is here equal to the negative probability to achieve conformity.

3.12.3 Conservatism and Ideas for Future Work

3.12.3.1 Conservatism

Three main factors are identified, which lead to conservatism. If the subsequent factors are taken into account in future work, the conservatism of the presented algorithm can be reduced. These main factors are elaborated next and can be addressed in future work to improve the algorithm.

Colored uncertainties: Uncertainties are called colored uncertainties, which are existing uncertainties in the system, but do not have any effect on the conformity or stability of the system. The uncertainties are called colored, since it can be literally compared to the color of the aircraft. The color of the aircraft has no effect on the stability/ conformity of the system. However, if an uncertainty is assigned to the color, an additional uncertain dimension is added to system.
Although the presented probabilistic-μ algorithm should account for that, by extending the stretching variable for that dimension, the algorithm will always calculate a conservative value in that dimension. The reason is, that the algorithm integrates over a hypercube in the current algorithm. The integral over the multivariate Gaussian distribution is always a limited integral and not unlimited in any uncertainty dimension. This leads to conservatism, also in the direction of the colored uncertainty. It is not a trivial task to identify a colored uncertainty. The μ-sensitivity might be used as an indicator. However, the sensitivity would need to be calculated over the whole uncertainty space, since correlation between uncertainties are also present (see e.g. Fig. 3.38-3.39), which is quite an effort for a high-dimensional uncertainty space. Future work can identify colored uncertainties such that they are not considered in the lower bound maximization.

Lower bound and upper bound: Conservatism arises also due to the upper bound calculation of the μ-value. As seen in the Fig. 3.73, this leads to noticeable differences in the probability calculation by comparing the probability calculation using the lower and upper bound of μ. The advantage is, that the μ-upper bound gives definite results. If the difference between lower and upper bound is reduced, the conservatism is decreased, too.

Two-sided interval versus one-sided interval: The integral over the multivariate Gaussian distribution is integrated over a hypercube. However, some uncertainties might affect the stability/ conformity only if they uncertainty is either decreased or increased, but not in both direction. This leads to conservatism, too.

3.12.3.2 Ideas for future work

Multiple ideas emerged during the development of the presented probabilistic-μ approach, how the algorithm can be further extended and augmented in future work. The most potential extensions are explained subsequently.

Utilization of found uncertainty set for Monte-Carlo simulations: The presented probabilistic μ-approach finds a definite uncertainty set, for which stability or conformity is guaranteed. The complementary of the found set can be used in future work for sampling based methods (monte-carlo methods) as initial set. This means, only generated samples outside of the found uncertainty set can be used to evaluate the objective function. In this way, the function evaluations of samples, which occur in the found set, are saved and less conservative probabilities for stability can be calculated.

Utilization of μ-lower bound for calculation of probability for definite instability: Instead of calculating the probability for stability using the μ-upper bound, the μ-lower bound can be used to compute the guaranteed probability for instability of the system. This is shown in Fig. 3.76. The plot illustrates exemplarily a determined two-dimensional uncertainty set (green) for definite conformity, which can be gained by the above elaborated probabilistic-μ approach. In red color, a possible uncertainty set is shown, which leads definitely to instability. This set can be calculated using the μ-lower bound results. By integration the uncertainty probability density function over such a set, a probability can be calculated, which would represent a lower bound of the probability leading definitely to instability. The computed probability represents again a lower bound since other uncertainty constellations might also exist, which lead to instability, and are not cover in the considered uncertainty set (e.g. shown in red color). Apparently, the set for definite stability (green) and the set for guaranteed instability (red) do not overlap. A gap between the two sets will occur, which results from the difference between the lower and the upper μ-bound. The results of such an algorithm, can be also beneficial, since it gives a definite lower bound for instability.

Further Ideas:

- The algorithm efficiency can be improved to enable a better applicability to real-live high-dimensional problems. By saving computation time, the number of optimization variables can be increased, such that the conservatism of the approach can be reduced.
- Uncertainties might not be modeled as Gaussian distributions. Non-Gaussian distributions could be approximated by representative weighted Gaussian mixture models.

Every weighted Gaussian distribution of the mixture model is then required to be integrated.

- New methods to integrate over arbitrary multi-dimensional probability distribution would lead to higher variability of the presented algorithm.
- Within the presented approach a specific mapping from the mapped uncertainty space to the true uncertainty space is chosen. Other transformation might lead to less conservative results.

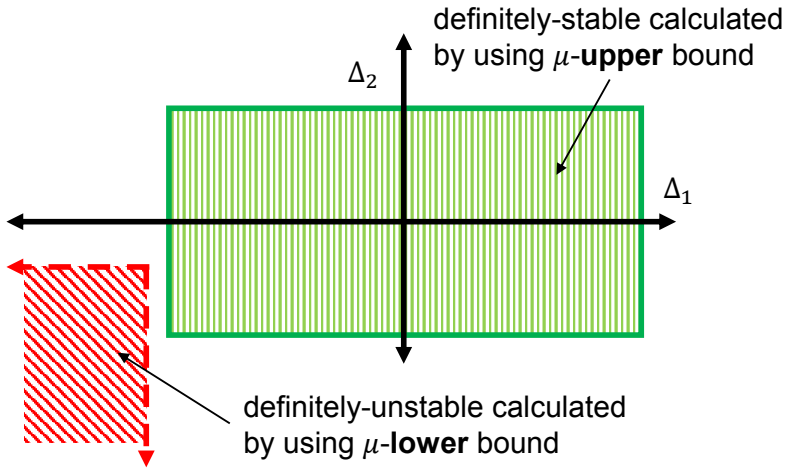


Figure 3.76: Under consideration of μ^{lb}-results, the probability can be calculated, which definitely brings the system to instability/ non-conformity.

Chapter 4

Development, Testing and Validation of Automatic Ground Control Functions of a Novel UAV for Automatic Take-Off and Landing

The last two chapters have introduced an automatic geofencing function as risk mitigation strategy for UAVs and robustness analysis methodologies for novel VTOL aerial vehicles. The following chapter comprises the design and development of automatic ground control functions for the novel diamond-shaped SAGITTA UAV demonstrator.
The chapter details the development of the ground controller for centerline tracking and runway alignment, the automatic braking functions for take-off and landing, the integration and verification of the navigation system including magnetometer and Global Navigation Satellite System (GNSS) antenna, and the planning and conduction of the on-ground tests from the first roll-out of the vehicle to successful first flight. The approach is presented by the application to the SAGITTA demonstrator with novel diamond-shaped configuration. The shown strategy can be applied to any other configuration of unmanned or optionally-piloted aerial vehicles, which take-off and land on runways automatically.
This chapter covers *Contribution III*, presented in Sec. 1.2.3, and encompasses the following main content:

- Complex on-ground aircraft modeling for the ground controller development
- Ground controller design for automatic centerline tracking and runway alignment
- Integration of sensor components of the flight control system and their related verification during on-ground tests
- Online gain set layout of the ground controller during taxi tests utilizing a specific ground testing software
- In-flight validation of the automatic ground control functions

In order to get familiar with the SAGITTA demonstrator and the corresponding flight control system, the next section introduces the conception of the UAV demonstrator, its system architecture and the developed flight control system. Sec. 4.2 explains the aircraft modeling with the focus on the on-ground dynamics and its interaction with the flight dynamics of the UAV. Subsequently, Sec. 4.3 introduces the ground controller design of the SAGITTA demonstrator. Sec. 4.4 elaborates the required sensor integration and testing by towing the aircraft and the online gain set synthesis of the ground controller based on real-live on-ground testing. After the sensor and ground controller testing is completed, Sec. 4.5 underlines the presented development by in-flight validation. Sections of the following work has been introduced by the author in Refs. [114, 115, 116].

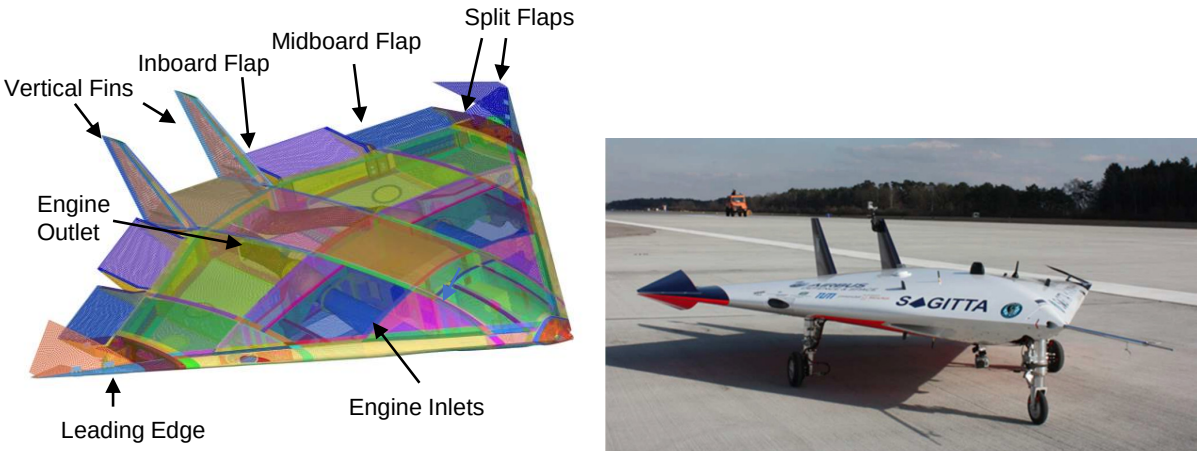
4.1 The SAGITTA Demonstrator

4.1.1 Configuration

The SAGITTA Open Innovation project has been launched by a joint cooperation between academic and industrial partners. The primary objective is the development of an UAV with a novel diamond-shaped configuration. The aircraft has a maximum take-off weight of 150 kg, a wingspan of 3.088 m and is illustrated in Fig. 4.1. The aircraft is powered by two integrated jet engines with 300 N thrust each. In-flight, the aircraft is controlled by inboard, midboard and outboard flaps commands. The inboard and midboard flaps are conventional control surfaces with their hinge line parallel to the trailing edge. The outboard flaps are novel split flaps at the wing tips with the hinge line parallel to the leading edge of the aircraft. The dirt and clean side of split flaps are deflected symmetrical on each side. The aircraft has a tripod landing gear with brakes on each main wheel. On ground, the aircraft is controlled by nose-wheel steering and outboard flaps deflections.

4.1.2 Concept of Operation

The nominal mission of the UAV demonstrator is to perform fully-automatic flights, including automatic take-off and landing. The nominal operation consists of waypoint navigation along pre-defined flight plans. The aircraft is monitored by the flight operator on ground during flight via the flight management control station. The flight operator is able to cease the waypoint-navigated reference mission and can guide the aircraft in various control modi with diverse authority levels. When the aircraft is in visual line of sight, so-called external pilots can take over and control the aircraft remotely.
In total, the aircraft is connected to different control stations via four data links. The aircraft receives commands and sends flight data via the *flight data link* and the *mission data link*



(a) Drawing (b) The SAGITTA demonstrator during taxi tests

Figure 4.1: Illustration of the SAGITTA demonstrator

from and to the ground station. If in line of sight, the external pilots can take over and command the aircraft via the *external pilot link*. In case of critical events, the aircraft can be terminated via the *termination link*. The flight operator can terminate the aircraft via a button within the ground control station. The termination command is send to the aircraft via the *termination link*, the engines are cut off and a parachute is triggered. The presented automatic safe area detection algorithm in Sec. 2 could be used for an automatic trigger of the termination system in future. The data links with its ground stations are depicted in Fig. 4.2. The flight operator commands the aircraft using the flight management control station and operates the aircraft on medium or high level modes. The basic operating functions of the flight management control stations are specified in the left column of Table 4.1. The following operation modes for the nominal case are available for the flight operator:

- Flight Operator (FO) Parking
- ATOL take-off
- ATOL take-off abort
- Medium Level Command
- High Level Command
- ATOL Landing
- ATOL Go-around
- Return to Base

By selecting the operation mode, the flight operator can command either on high level or medium level in flight. The aircraft performs waypoint navigation automatically on high-level command according to the flight plan, which is set by selecting a specific *Enroute List*. While the aircraft flies in high-level mode, the flight operator can activate the functions *Speed by*

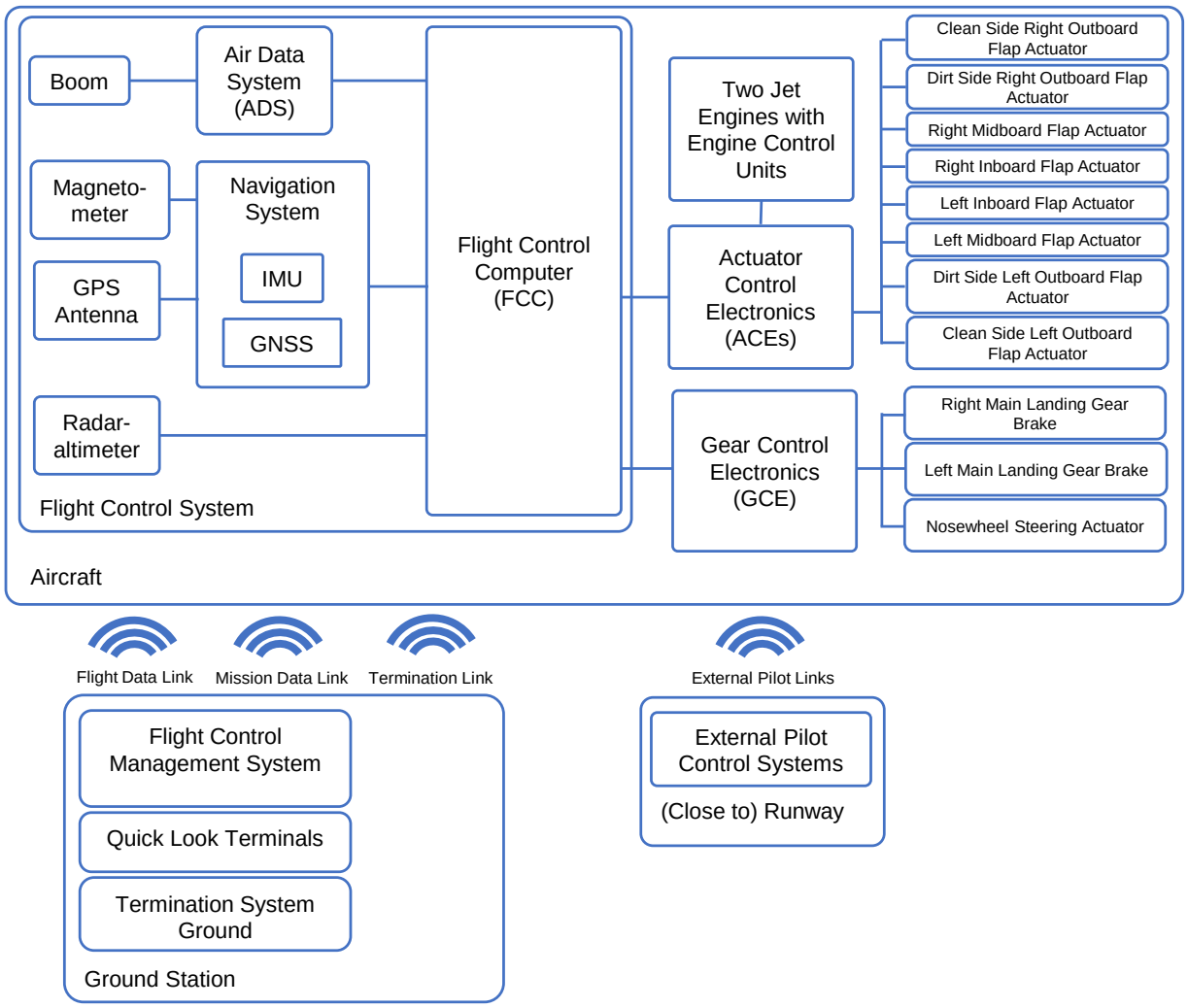


Figure 4.2: System Architecture of the SAGITTA demonstrator (based on Ref. [117])

FO or *Altitude by FO*, in order to command speed or altitude of the aircraft, which differ from the flight plan. If the flight operator switches to medium level mode, he commands the heading, the speed and the altitude of the aircraft equivalently to a conventional autopilot system. If the *Return-to-Base* mode is activated, the aircraft is guided directly to a defined home base waypoint. This mode is a combination of medium and high level commands. In order to initiate the automatic take-off or landing, various conditions need to be fulfilled. The required take-off and landing conditions for activation are detailed in Refs. [118, 119]. If the aircraft is in line of sight, the *external pilots* can take control.

It is distinguished, whether the external pilot takes over on-ground or in-air. If the aircraft is in-air, the external pilot controls primarily the aircraft by roll angle, pitch angle and thrust commands.

While the aircraft is on-ground, the command strategy change to a direct mode with rate damping. In Fig. 4.3 the EP command mapping for on-ground operation is illustrated. On-ground the external pilot commands the nose wheel steering angle by a direct mapping

Table 4.1: Operational Elements of the Flight Operator on the FMCS (Ref. [114])

Operating Functions		
Flight Software	Taxi Test Software	Data Type
High-Level		
Operation Mode	Operation Mode	Uint8
Speed by FO	Not Used	Boolean
Altitude by FO	Not Used	Boolean
Medium-Level		
Heading Control	Gain Set Selection	Double
Speed Control	Throttle Command	Double
Altitude Command	Not Used	Double
Flight Plan		
Enroute List	Setting NAV Alignment	Uint8
Link Loss List	Split Flaps Activation	Uint8
Control Switches		
Control Switch 1	Brake Activation	Boolean
Control Switch 2	Amplitude Trigger	Boolean
Control Switch 3	Lateral Offset Direction	Boolean
Control Switch 4	Offset Margin	Boolean
Control Switch 5	Runway Direction	Boolean
Control Switch 6	No GPS Structure	Boolean
Control Switch 7	NAV Alignment	Boolean
Control Switch 8	Activation Gain Set Selection	Boolean

between the EP yaw channel and the nose wheel steering command. The EP roll and pitch command channel are fed in the control stability augmentation system of the flight control system, where the flap deflections are calculated. During on-ground operations the EP roll command channel corresponds to a proportional deflection command of the midboard flaps. The EP pitch command channel is canceled out by the control stability augmentation system on-ground and thus has not an effect. The brake command, the parking brake command and the engines thrust command are according to the EP commands.

In case of data link losses, the flight operator modes and external pilot modes change to different contingency modes. A detailed overview about the available modes and their contingency procedures can be found in Ref. [120].

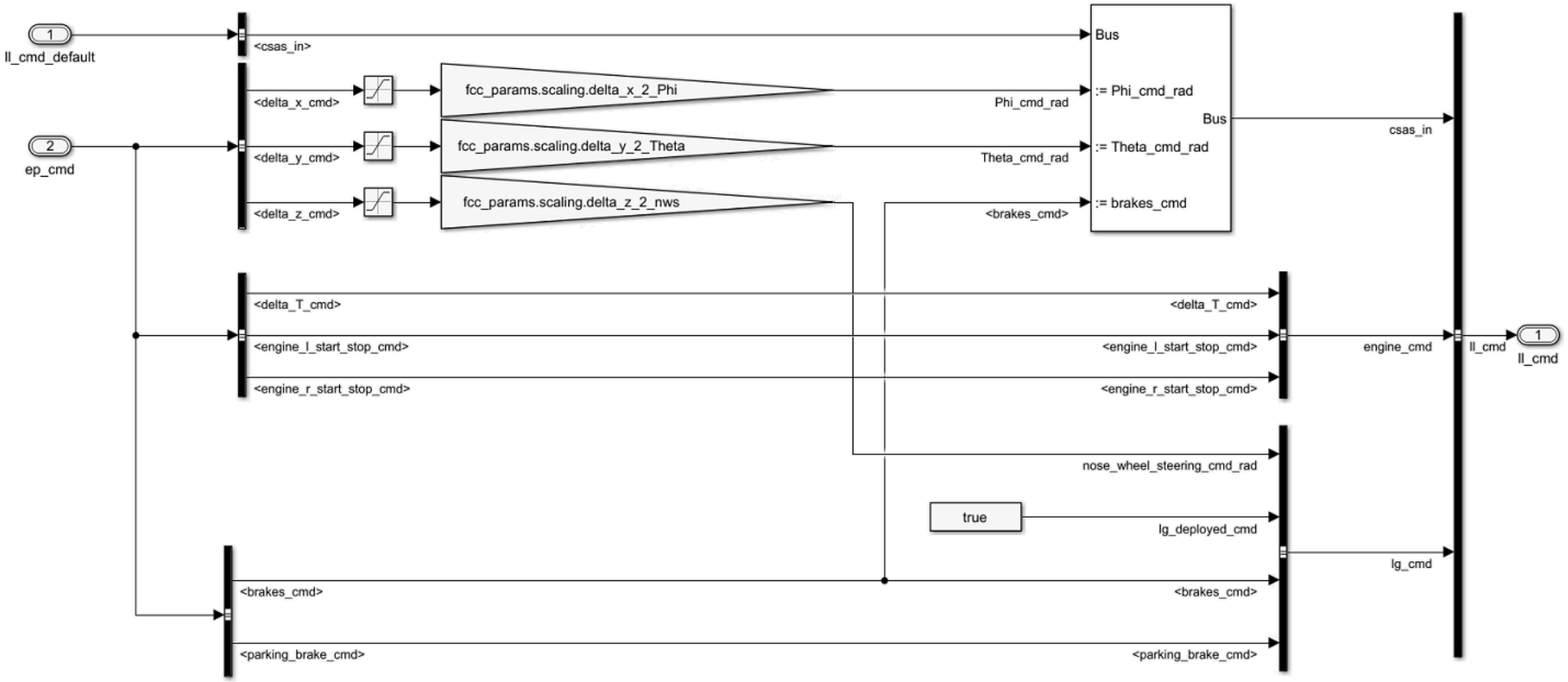


Figure 4.3: On-ground External Pilot Command Mapping implemented in *Mathworks MATLAB/ Simulink*

4.1.3 System Architecture

Fig. 4.2 depicts the overall system architecture of the demonstrator. The flight operator commands the aircraft via the flight management control station, which is integrated within the ground station. The commands are transmitted to the flight control computer in the aircraft via the above mentioned four data links. The communication between the control stations on ground and the aircraft is detailed in Ref. [121] by the system designers.
The flight control system is equipped with several sensors yielding the feedback signals to control the aircraft. The sensor equipment includes navigation system with inertial measurement unit, magnetometer and GNSS antenna, an air data computer with nose boom, and a radar altimeter for altitude measurements. Utilizing the commands of the flight operator and the sensor information, the flight control computer calculates the demanded control surface deflections, the engine thrust and the nose wheel steering angle and commands them to the actuator control electronics and the general control electronics.
The demonstrator is equipped with two actuator control electronics. The left/right actuator control electronic forwards the commands to the actuators of the control effectors on the left/right side. For the demonstrator the control effectors are the inboard, midboard and outboard flaps and the two jet engines. The actuator control electronics monitor their respective actuators and provide status information to the flight control computer. The two engines are controlled by two engine control units, which receive their spool speed commands from the their respective actuator control electronics. The gear control electronics transmit the brake command from the flight control computer to the main landing gear brakes, and passes the nose wheel steering angle command to the nose wheel steering actuator. The gear control electronics send the status information of weight-on-wheel sensors of the main and nose landing gear to the flight control system. More information to the general system architecture can be found in Ref. [122].
In the ground station, additional quick look terminals are installed to monitor the different systems. Any parameter, which is sent by the aircraft via the data links to the ground station, can be reviewed online during the operation by flight test engineers.

4.1.4 Flight Control System

4.1.4.1 Design and Structure

The core of the flight control system is the flight control software, which is explained in Ref. [114]. The most important auto flight functions are incorporated in the flight control software. The structure of the modular flight guidance and control system of the demonstrator is presented in Fig. 4.4. The inputs are the data links (C2), including the commands from the flight management control station, the sensor data and health stati, and the actuator and

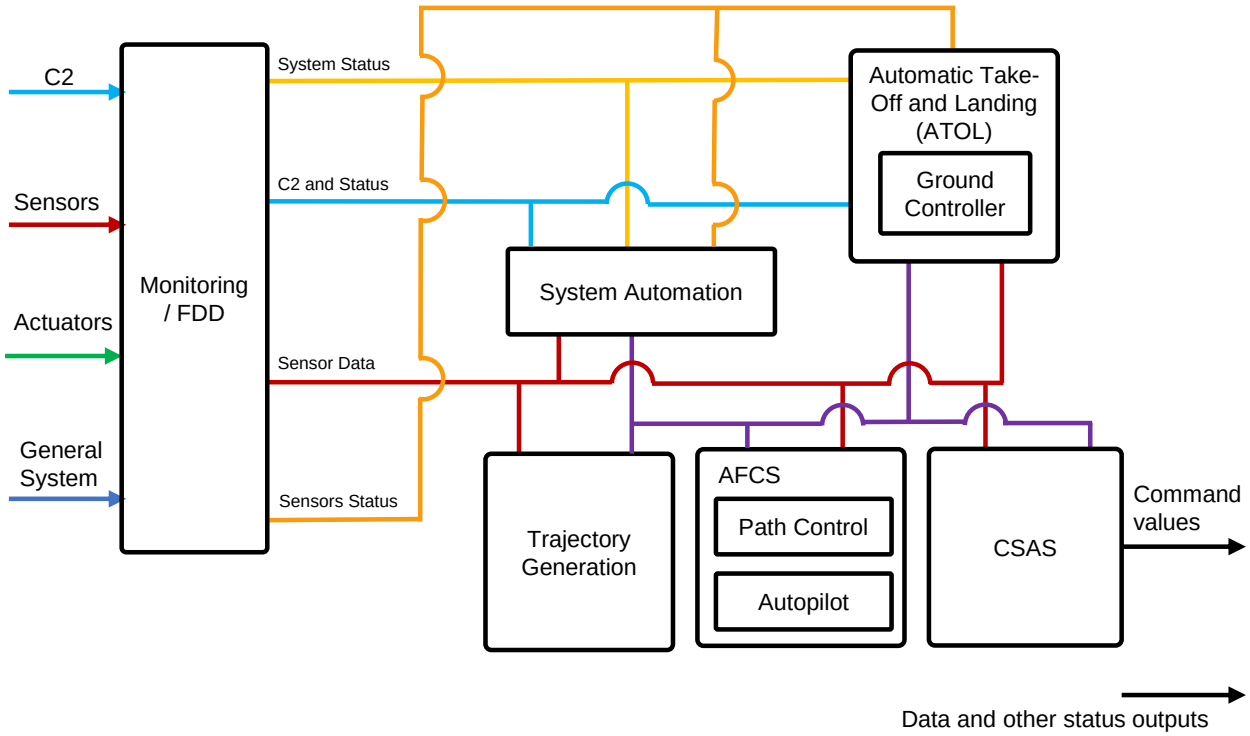


Figure 4.4: Architecture of Flight Control Software (reworked from Ref. [123])

general system information.

The monitoring module inspects the inputs by plausibility range checks, observing the health stati and quality evaluations and forwards the information to the other modules. Depending on the flight states and the stati of the monitoring system, the system automation module processes and decides on the flight operator commands which appropriate action has to be taken by enabling and/or disabling the remaining modules (see Ref. [120]). The trajectory generation (see Ref. [70]), the automatic flight control system (see Ref. [69]) (denoted as AFCS) and the control and stability augmentation system (CSAS) (see Refs. [124, 125]) contain the hierarchical control loops for flight management, flight guidance and flight control of the aircraft.

The automatic take-off and landing module is allocated at a similar level as the system automation module, but is only engaged during take-off and landing. It activates and deactivates different control loops for the take-off and landing maneuver (see Ref. [123]). The ground controller is enabled during the automatic take-off and landing maneuver. The ground controller is therefore integrated within the automatic take-off and landing module.

4.1.4.2 Software Development Process

The flight control software comprising the functional algorithms and the auto flight system is developed by a model-based approach using *MATLAB / Simulink / Stateflow* from *Mathworks*. Together with the simulation model, introduced in Sec. 4.2.1, which is also imple-

mented in *MATLAB / Simulink*, the closed-loop system in *MATLAB / Simulink* provides a model-in-the-loop simulation (MILS) environment, which is used as framework for testing and verification of the implemented flight control functions. Source code in *ANSI C* is generated from the implemented flight control software by *Mathworks Embedded Coder* and incorporated into a developed *C* framework, which enables the execution of the auto-generated code on the flight control computer. The framework processes the incoming and outgoing signals of the physical interfaces on the flight control computer to the inputs and outputs of the auto-generated code. More information about the *C* framework and the auto-code generation can be found in Ref. [126]. Hardware-in-the-loop tests are conducted to verify the auto-generated code and the respective *C* framework by executing the code on the flight control computer and comparing the results with the corresponding *MATLAB / Simulink* models. The design verification and system integration of the SAGITTA demonstrator is described in detail in Ref. [127].

4.2 Modeling & Simulation of the SAGITTA Demonstrator

4.2.1 SAGITTA Simulation Model

The core of the simulation model are the rigid-body equation of motions based on Newton's laws and the respective state integration. The forces and moments are calculated in corresponding models. The simulation model comprises also the motion kinematics of the aircraft, an aerodynamic model, a propulsion model, landing gear models of the main wheels and the nose wheel, actuator models of the control surfaces and sensor models of the navigation system and the air data system. In addition, the simulation model contains an atmospheric model, a magnetic field model, a gravitational model and a terrain model. Relevant for the development of the ground control functions are mainly the propulsion and the aerodynamic model, which are explained next, and the landing gear models for on-ground simulations, which are introduced in the next section in Sec. 4.2.2. Explanations of the remaining models and a comprehensive overview of the simulation model are given in Ref. [54, p.40ff.].
The propulsion model for the SAGITTA demonstrator has been developed by the *Institute of Aircraft Design of the Technical University of Munich*. The model calculates the propulsive forces and moments on the aircraft. Thereby, the model outputs the air mass flow, the fuel flow and the thrust force and moments in dependence of the revolutions per minute (RPM), the flight altitude, the *Mach* number, other aircraft states and the given thrust command. The propulsion model has been validated by dedicated engine runs with nozzle and inlet duct (see Ref. [128]). Straight acceleration runs during taxi tests introduced in Sec. 4.4.3 and Sec. 4.4.5 validated the model on-ground together with the landing gear model.
The model for aerodynamic characteristics was developed by the *University of Applied Sci-*

ences Ingolstadt and the *Institute of System Dynamics and Control of the German Aerospace Center (DLR)*. Wind-tunnel tests and computational-fluid-dynamic simulations from the *Institute of Aerodynamics and Fluid Mechanics of the Technical University of Munich* were utilized (see e.g. Refs. [129, 130]) for the development. Ref. [39] explains the used aerodynamic characteristics in the lateral motion in detail. The ground effect is considered by an empirical model, when the aircraft is less than 60 m above ground (see Ref. [54, p.48]).

4.2.2 SAGITTA Landing Gear Model[12]

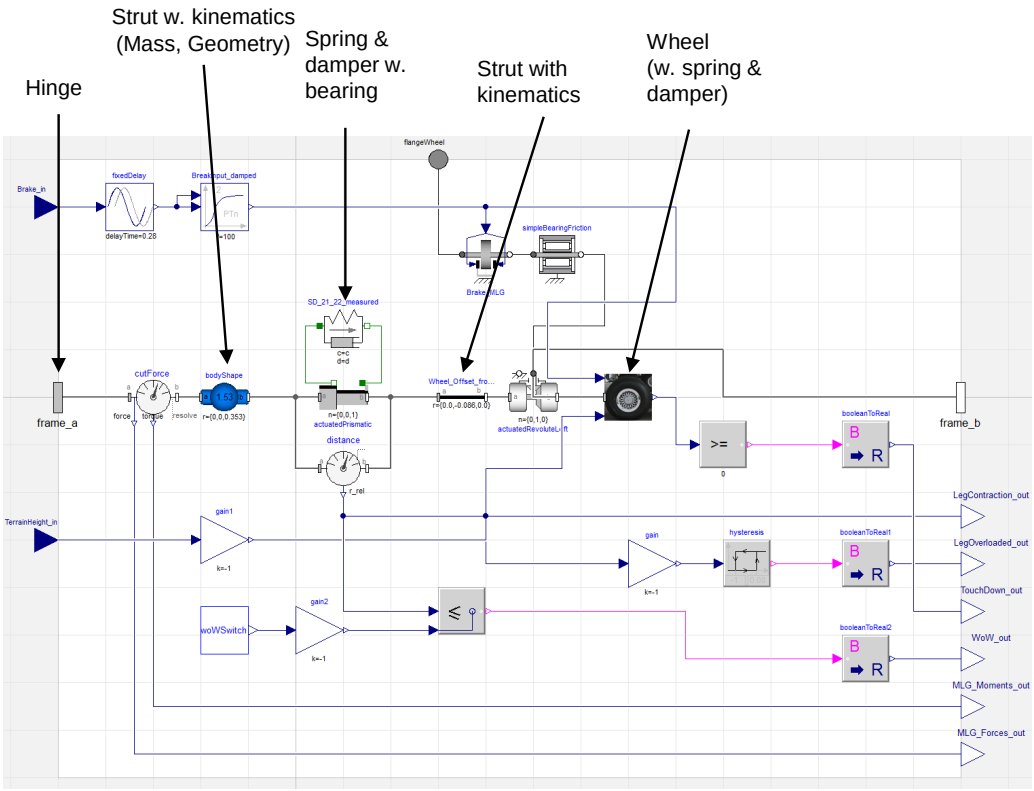
For the development and simulation of the ground controller an accurate model of the aircraft on ground and its physical properties is necessary. In combination with the aircraft flight dynamic model, including the aerodynamic characteristics, the landing gear model covers all physical effects of the aircraft during take-off, landing and ground operations. The landing gear of the real aircraft is shown in Fig. 4.5. The following section describes the respective physical dynamics and modeling of the used SAGITTA landing gear model, which has been developed by the *German Aerospace Center* (DLR). Sec. 4.2.2 has been published in Ref. [115], which introduced the landing gear model together with the ground control design. The corresponding landing gear section in Ref. [115] and thus also this section has been mainly created by R. Kuchar (second author) from the German Aerospace Center (*DLR*). The landing gear model is recaptured within this thesis to cover the full development of the ground control functions for the SAGITTA demonstrator in this chapter.
The developed landing gear model comprises the following physical properties:

- Longitudinal, lateral and vertical dynamics
- Braking (within the main gear models) and directional steering (within the gear model of the nose wheel)
- Elimination of inertial landing gear loads, since they are considered within the flight dynamic model
- Wheel slip approximation and Spin-up loads
- Weight-on-wheel signal indication
- Execution in a fixed-step and real-time simulation environment

The model comprises one wheel per landing gear leg, equivalent to the aircraft. Within the landing gear simulation model, the left main wheel is connected to a mechanical brake, whereas the nose wheel is attached to an actuator for directional aircraft steering. The modeling of the wheels is described first, before the vehicle dynamics are explained. Fig. 4.6 pictures the modeled vertical and lateral dynamics of the landing gear model.

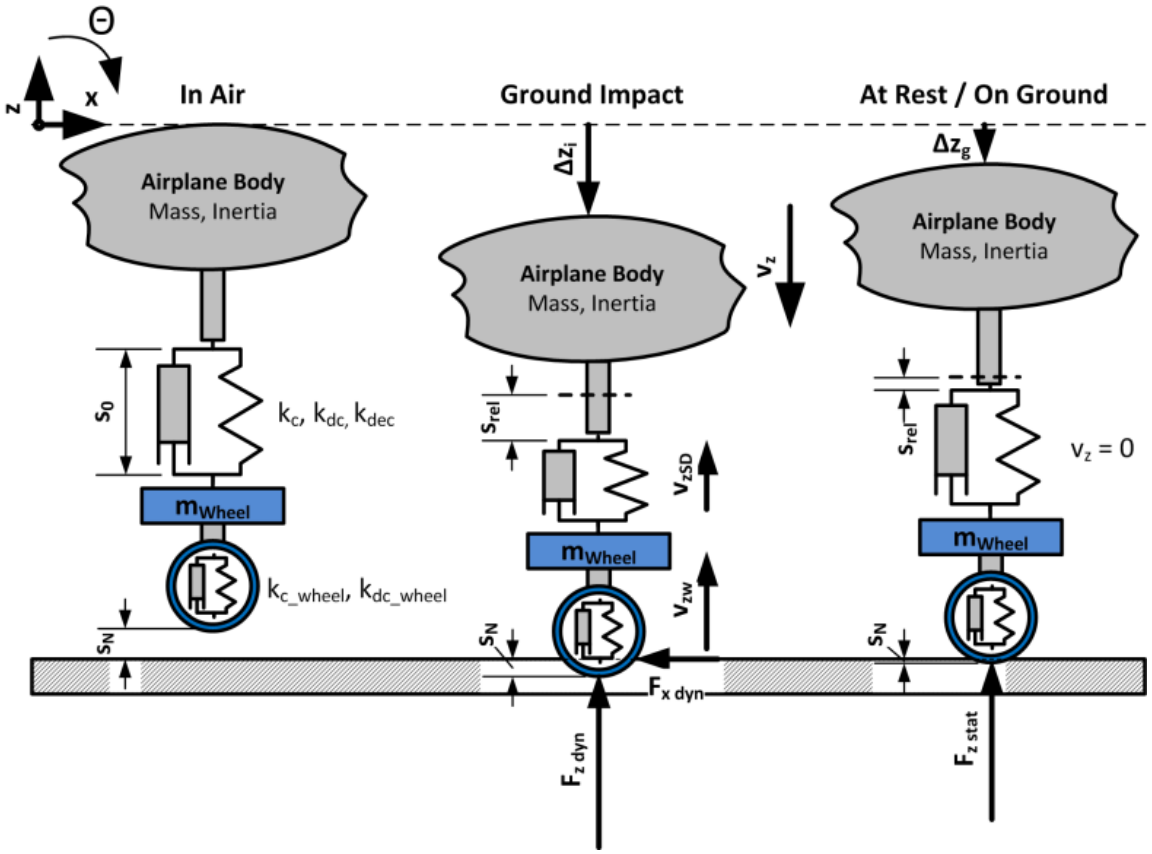
[12] This section represents a shortened version of the work in Ref. [115] and has been mainly created by R. Kuchar (second author) from the German Aerospace Center (*DLR*).

(a) The SAGITTA Landing Gear

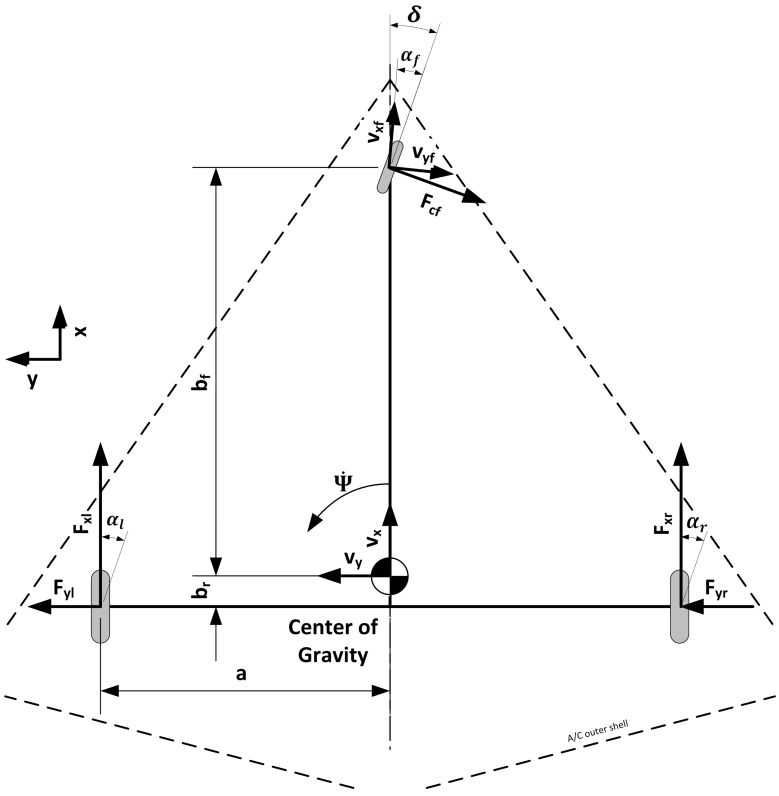


(b) Landing Gear with Modelica Multi-Body Elements (from R. Kuchar, DLR, in Ref. [115])

Figure 4.5: Illustration of the SAGITTA landing gear in reality and simulation



(a) Vertical landing gear modeling as multiple mass-spring damper



(b) Schematic tricycle landing gear of the SAGITTA demonstrator

Figure 4.6: Modeled vertical and lateral dynamics of the SAGITTA landing gear model (both figures from R. Kuchar, DLR, in Ref. [115])

4.2.2.1 Wheel Model

The wheel model covers the physical contact between the aircraft and the runway. The vertical wheel load forces $F_{z_W(f/l/r)}$ are calculated, utilizing the respective wheel deflection s_N and wheel spring-damper coefficients k_{cW} and k_{dW} (see Fig. 4.6(a)). Together with the normalized wheel slips of the longitudinal and the lateral directions and the vertical wheel load forces, the longitudinal and lateral fraction forces $F_{x_W(f/l/r)}$ and $F_{x_W(f/l/r)}$ are calculated using the *Pacejka Magic Formula*. These formulas can be reviewed in Ref. [131]. More details of the wheel model can be found in Ref. [115].

4.2.2.2 Vehicle Dynamics Model

The equations of motion of the demonstrator on ground are derived from the vehicle drawing illustrated in Fig. 4.6(b) and explained in more detail in Ref. [132]. The planar equations of motion in the center of gravity frame can be denoted:

$$\begin{aligned} m\,(\dot{v}_x - \dot{\psi}\,v_y) &= F_x = -F_{cf}\,\sin\delta + \frac{T_l}{R_R} - F_r \\ m\,(\dot{v}_y + \dot{\psi}\,v_x) &= F_y = F_{cf}\,\cos\delta + F_{y_W l} + F_{y_W r} \end{aligned} \tag{4.1}$$

$$J_z\,\ddot{\psi} = M_z = b_f\,F_{cf}\,\cos\delta + \frac{-a\,T_l}{R_R} - b_r(F_{y_W l} + F_{y_W r}) \tag{4.2}$$

where

- v_x is the longitudinal speed,
- v_y is the lateral speed,
- ψ is the yaw angle,
- m is the vehicle mass,
- J_z is the vehicle moment of inertia around the yaw axis,
- δ is the nose gear wheel steering angle,
- $F_{cf} = f(v_x, \dot{\psi}, \delta)$ is the cornering force of the nose gear wheel,
- R_R is the effective radius of the main gear wheels,
- T_l is the torque applied on the left wheel (brake),
- F_r is a term subsuming all the resistances (see Fig. 4.6(b))
- a, b_f and b_r are geometric parameters that characterize the position of the center of gravity as shown in Fig. 4.6(b).

The vertical equations of motion are based on the illustration in Fig. 4.6(a). The vertical dynamics of the aircraft on ground are mainly dominated by the spring-damper configuration of the main landing gear legs and the characteristics of their wheels:

$$m\,(\dot{v}_z + g) = F_z = F_{zf} + F_{zl} + F_{zr} \tag{4.3}$$

$$\begin{aligned} J_x \ddot{\phi} &= M_x = a\,(F_{zl} - F_{zr}) \\ J_y \ddot{\theta} &= M_y = b_r\,(F_{zl} + F_{zr}) - b_f\,F_{zf} \end{aligned} \tag{4.4}$$

with the respective leg forces (combined spring-damper and wheel forces) generally defined:

$$F_{z(f/l/r)} = k_{c\,SD}\,s_{rel} + k_{d\,SD}\,\dot{s}_{rel} + F_{zW} - m_W\,(\dot{v}_{zw} + g) \tag{4.5}$$

where

- v_z is the vertical speed,
- m_W is the mass of the wheel,
- $k_{c\,SD}$ denotes the spring coefficients of the respective leg spring and
- $k_{d\,SD} = f(v_{z\,SD})$ the damping coefficient respectively (see Fig. 4.6(b)),
- $s_{rel} = s_0 - s_{meas}$ is the compression/expansion distance as measured directly at the spring-damper location against s_0, the relaxed spring length (see Fig. 4.6(b)),
- J_x and J_y are the vehicle inertias around the roll and pitch axis,
- ϕ and θ are the respective roll and pitch (Euler-) angles of the vehicle,
- finally v_{zw} is the accumulated vertical wheel speed.

According to Ref. [114], dependent on the direction of the local movements, the main damping characteristics contain hysteresis, and the wheel damping is assumed to be constant.

4.2.2.3 Model Implementation in Modelica and Export

As the landing gear model employs a system with multiple mass-spring-damper components, the implementation with Modelica (see Ref. [133]) is suggested, as the embedded Multi-Body library provides a standardized approach to these problems.
The *Institute of Flight Systems of the German Aerospace Center (DLR)* has been in charge of the general implementation of the SAGITTA demonstrator landing gear and thus supplying the required database for the modeling of the dynamic landing gear model, which includes geometry, mass, spring-damper and wheel data.
Based thereon, the actual landing gear model has been implemented. Fig. 4.5(b) gives an impression of the main landing gear leg sub-model. Utilizing the object-oriented features within *Modelica*, the same model has been inherited for all three legs with only minor adaptations. The core elements of the model are fully parameterized for robustness analysis and simulation studies.
To extract static inertial content of the landing gear, that is already considered within the flight dynamic model, two inherited landing gear model instances are utilized: One instance being airborne at all times, whereas the second model may encounter ground contact depending on the actual flight conditions allowing to observe only ground-contact-triggered values.

As the ground control law is implemented and tested within a *MathWorks Simulink* environment, a direct integration method for the implemented *Modelica*-based model is required. As generic interface methods are available in both environments, the implemented model can be exported from *Modelica* using the Functional-Mockup interface (FMI for Model Exchange, see Ref. [134]) and directly integrated in *Simulink* via the S-Function mechanism.
Based on test-bench data (vertical dynamics) and live experiments during performed taxi tests in lateral and longitudinal motion (see section 4.4.3), the implemented landing gear model could be validated against existing hardware.

4.3 Design of the Ground Controller[13]

Parts of the following Sec. 4.3 and Sec. 4.4 have been already published by the author in Refs. [114, 115], which cover the design, the online gain set layout and testing of the ground controller and the development of the respective taxi test software.

4.3.1 Primary Objectives and derived Requirements

Generally, the ground controller is developed to keep the aircraft on the centerline of the runway during the take-off and landing maneuvers of the aircraft. The controller is able to counteract external environmental disturbances such as wind and uneven runway surfaces. For controlling the aircraft on ground the nose wheel angle and the control surface deflections of the midboard and the outboard flaps are available as inputs. Due to the present hardware conditions, the controller has to show robustness against a backlash between the steering actuator and the nose wheel angle itself and possible asymmetric braking behavior. For the centerline tracking GNSS availability is necessary. If a GNSS loss occurs during the landing maneuver a back-up control structure is designed.

4.3.2 Structure of the Ground Controller

The structure of the ground controller is depicted in Fig. 4.7. The cascaded controller is divided into an inner, middle and outer loop and comprises proportional elements only. The inputs to the inner loop are the yaw rate command r_{cmd} and the measured yaw rate r of the aircraft. The outputs are the nose wheel steering command δ_{cmd} and an aerodynamic crossfeed to the midboard and outboard flaps. The yaw rate command is calculated as output of the middle loop, which uses the course of the aircraft χ as feedback.
The reason for this is given by the available system characteristics, where the course signal χ features higher reliability and accuracy compared to the heading signal Ψ. The input of the middle loop is the course error between the current course of the aircraft and the given runway heading Ψ_{RWY}. The course command $\Delta\chi_{cmd}$ is determined as output of the outer loop, which uses the current lateral deviation of the aircraft y_{cmd} and the commanded lateral deviation y_{meas} with reference to the centerline as input.
In case of a GNSS loss during the take-off and landing maneuver, the aircraft still needs to decelerate safely until standstill on the runway. Therefore, a back-up control law is introduced since in this case the lateral deviation of the aircraft and the course information are not available anymore. When a GNSS loss is detected by the monitoring module of the flight control software, the back-up control law functionality is faded in. The back-up law is also

[13] This section has been published in Ref. [115] by the author and revised in this thesis.

shown in Fig. 4.7. The inner loop of the back-up law is equivalent to the nominal structure since the yaw rate measurement is still available. However, the outer loop and the middle loop are replaced by a heading controller. Hence, the back-up loop assures the alignment of the aircraft parallel to the centerline. Beside the control law for centerline tracking, the ground controller gives also the brake commands to the two main landing gears of the aircraft. Depending on the *ATOL mode* the ground controller gives different brake commands. The braking logic is explained in Sec. 4.3.4.

4.3.3 General Design Objectives

The cascaded control structure separates the different control tasks from in- to outside (fast to slow) and thus allows for a step-by-step design and implementation. The outer loop is almost isolated from disturbances because the faster middle and inner loop mainly compensates the disturbances. If the GNSS signal becomes unavailable, the cascaded control structure allows for a smooth transition to the heading controller by fading in the heading controller command signal. Integral elements are decisively not used in the control structure to avoid integral wind-up and to remain a simple and reliable structure. On the one hand, integral elements facilitate oscillation tendencies, on the other hand, if an integral wind-up occurred, one have to cope with a step-like change in the closed loop dynamics.
Although the driving dynamics change significantly with higher speeds (Ref. [57]), a gain scheduling with the ground speed is not utilized. The reason for this is the higher controller complexity in presence of gain scheduling, which leads to more parameters to be determined/tuned via simulation/taxi tests, and especially to the dependence of the ground controller functionality on proper available ground speed signal.
During taxi tests it turned out that proportional gains are completely sufficient for centerline tracking with adequate stability and robustness characteristics, which is also shown in Sec. 4.4.5. In addition, the cascaded control structure enables easy to introduce limitations of the commanded nose wheel steering angle and the yaw rate to prevent a hazardous tilting of the aircraft. The yaw rate command is limited as follows:

$$|r_{cmd}|_{max} = f_{y,max}/(V_K^R)^E \tag{4.6}$$

The maximum lateral acceleration in terms of $f_{y,max}$ is calculated by considering the maximum allowed centrifugal force, which causes an initiation of tilting the aircraft. Note, $(V_K^R)^E$ represents the absolute kinematic velocity, following the notation in Ref. [105]. The calculation of the tip-over velocity is not within the scope of this thesis. In addition to the yaw rate, the nose wheel steering command is limited to a predefined threshold, which is sufficient for centerline tracking. As Fig. 4.7 illustrates, in front of the nose wheel steering angle command limiter, a blending gain is introduced. A linear function from 0.2 m/s to 1.0 m/s ground

speed fades the nose wheel steering angle command in to prevent the ground controller from wearing the tires in standstill and at low ground speeds. Besides the easy limitation of the commands, the cascaded control structure also allows a yaw rate command crossfeed to the aerodynamic control surfaces of the aircraft. Since the control surface deflection commands are output signals of the CSAS passing the control allocation, the yaw rate command of the GC is an input to the CSAS. Inside the CSAS, the yaw rate control for flight is reconfigured for the yaw rate control on ground by the additional aero-crossfeed gain k_{rA}, which overwrites the yaw gains for flight. This mode is enabled by the trigger-signal *On-Ground Flag*, which is set by the ATOL module during the take-off and landing maneuvers.

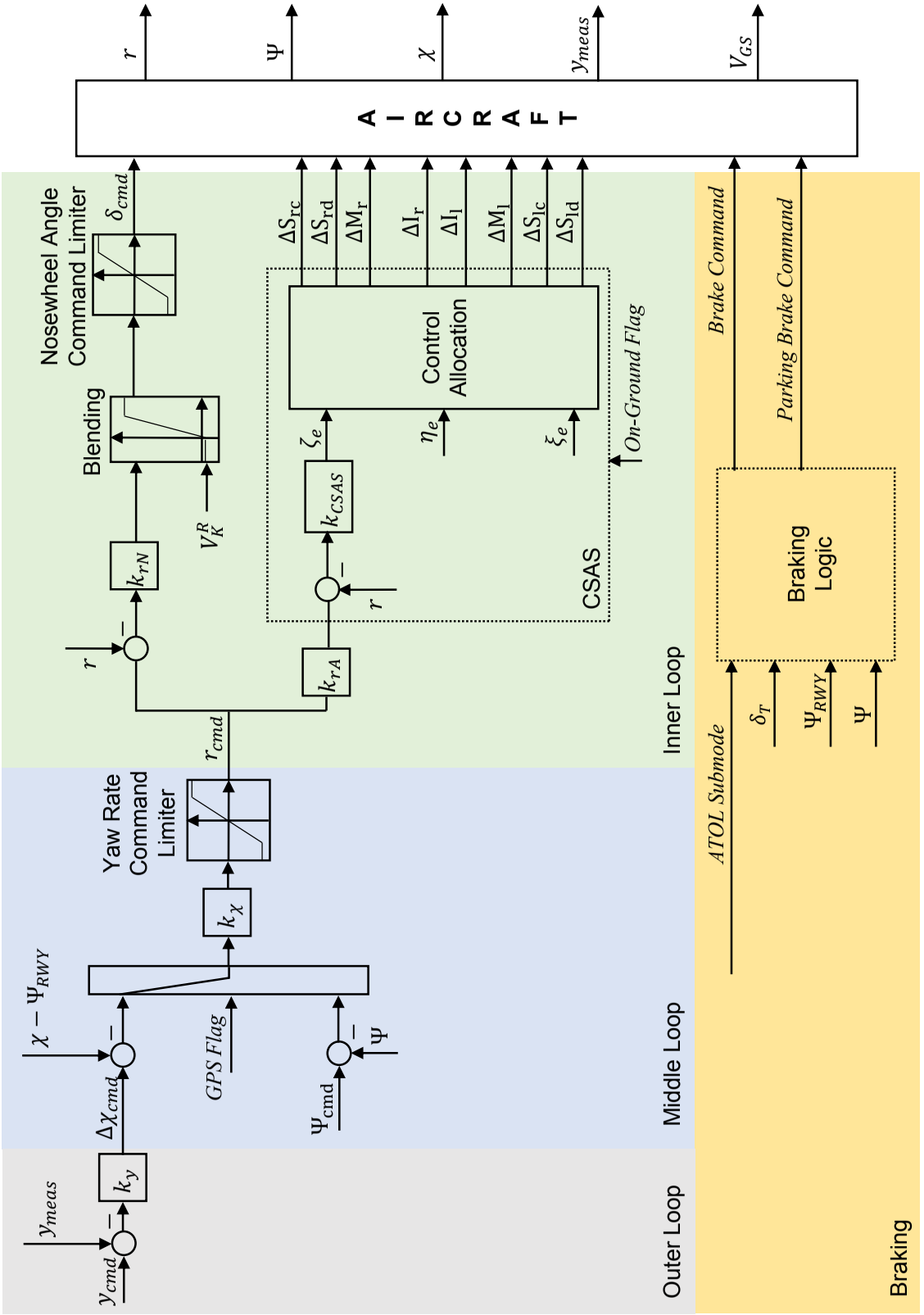


Figure 4.7: SAGITTA ground controller structure

4.3.4 Modes of the Ground Controller

The ground controller is integrated in the ATOL module of the flight control software. Depending on ATOL phases, different modes of the ground controller are enabled. The ground controller is enabled during the following ATOL phases:

- Parking
- Ground Acceleration
- Take-off Rotation
- Take-off Abort
- Landing Derotation
- Ground Deceleration

The full description of the available modes is provided in this subsection. The explained modes are illustrated in Fig. 4.8 for take-off and in Fig. 4.10 for landing. The switching from one ATOL phase to the next is executed by the ATOL module itself. The conditions for the phase transition are elaborated in Ref. [54].

Parking: The parking mode is commanded by the ATOL module when either the mode *ATOL take-off* is initialized by the Flight Operator or when the aircraft comes to a stop after take-off abort or ground deceleration. If the ground controller is in parking mode, the ground controller sends zero commands to the CSAS and to the nose wheel steering actuator. During parking no nose wheel command is given to avoid the turning of the nose wheel steering tire in standstill. The parking brake is closed and the brake is open as displayed in Fig. 4.8.

Ground Acceleration: After the take-off is initialized by the FO, the ATOL module changes its mode from parking to acceleration. The brakes are released and 100 % thrust command is given to the engines. The aircraft accelerates and the ground controller keeps the aircraft on the commanded trajectory by nose wheel steering commands. Since the flaps have no influence to the dynamics of the aircraft due to low dynamic pressure, the midboard and outboard flaps are deactivated for tracking until 10 kts. After the threshold is reached, the aero-crossfeed yaw rate command to the CSAS system is faded in. During the whole ground acceleration the yaw damper, the roll damper and the pitch damper integrated in the CSAS system are active. The pitch damper is part of the longitudinal module of the CSAS. There are two reasons why the pitch damper is active. On the one hand, it prevents the aircraft to become airborne unintentionally before its rotation speed is reached. If a positive pitch rate occurs due to an obstacle or bumpiness of the runway, the pitch damper works against pitching up by generating a negative pitch moment with inboard-flaps deflections. On the other hand, it prevents the aircraft to have unforeseen transition phases of different modes during the rotation of the aircraft. The pitch damper is necessary during the rotation phase. If the

pitch damper is activated at the beginning of the rotation phase, a transition phase is forced during a high-critical phase of the take-off. This risk can easily be avoided by an active pitch damper during the ground acceleration. Identical argumentation can be found for the roll damper. During the rotation of the aircraft a gust or the ground effect can disrupt the aircraft. For that reason the roll damper is activated during ground control. Since the roll damper is integrated in the lateral part of the CSAS module, the roll damper commands outboard and midboard surface deflections. The yaw damper is an essential part of the aerodynamic crossfeed of the ground controller. The yaw rate feedback of the aerodynamic crossfeed is integrated in the CSAS. This implementation enables an easy transition of the CSAS from the ground mode to the airborne mode. Together with the other reasons explained above for the roll and the pitch damper the yaw damper stays active during the ground acceleration. Both brakes are open during the ground acceleration as visualized in Fig. 4.8.

Take-off Rotation: The rotation is initialized when the aircraft exceeds the given rotation speed. A step-like ramp command is given to the inboard flaps by the CSAS to initiate the take-off rotation. During this phase, the ground controller is active equivalent to the ground acceleration sub-mode. Since the exact instant is unknown when the aircraft pitches up, the nose wheel steering command is active during this phase. By pitching up, the nose wheel influence decreases steadily and the aero-crossfeed command to the CSAS becomes essential to keep the aircraft aligned to the runway. Both brakes are open.

Take-off Abort: The Take-Off Abort can be commanded either by the Flight Operator or is triggered by the ATOL system when its ATOL availability conditions are violated. During the take-off abort, the aircraft starts to decelerate by brake commands. While the take-off abort sub-mode is active, the nose wheel steering as well as the aero-crossfeed yaw rate command of the ground controller are active.

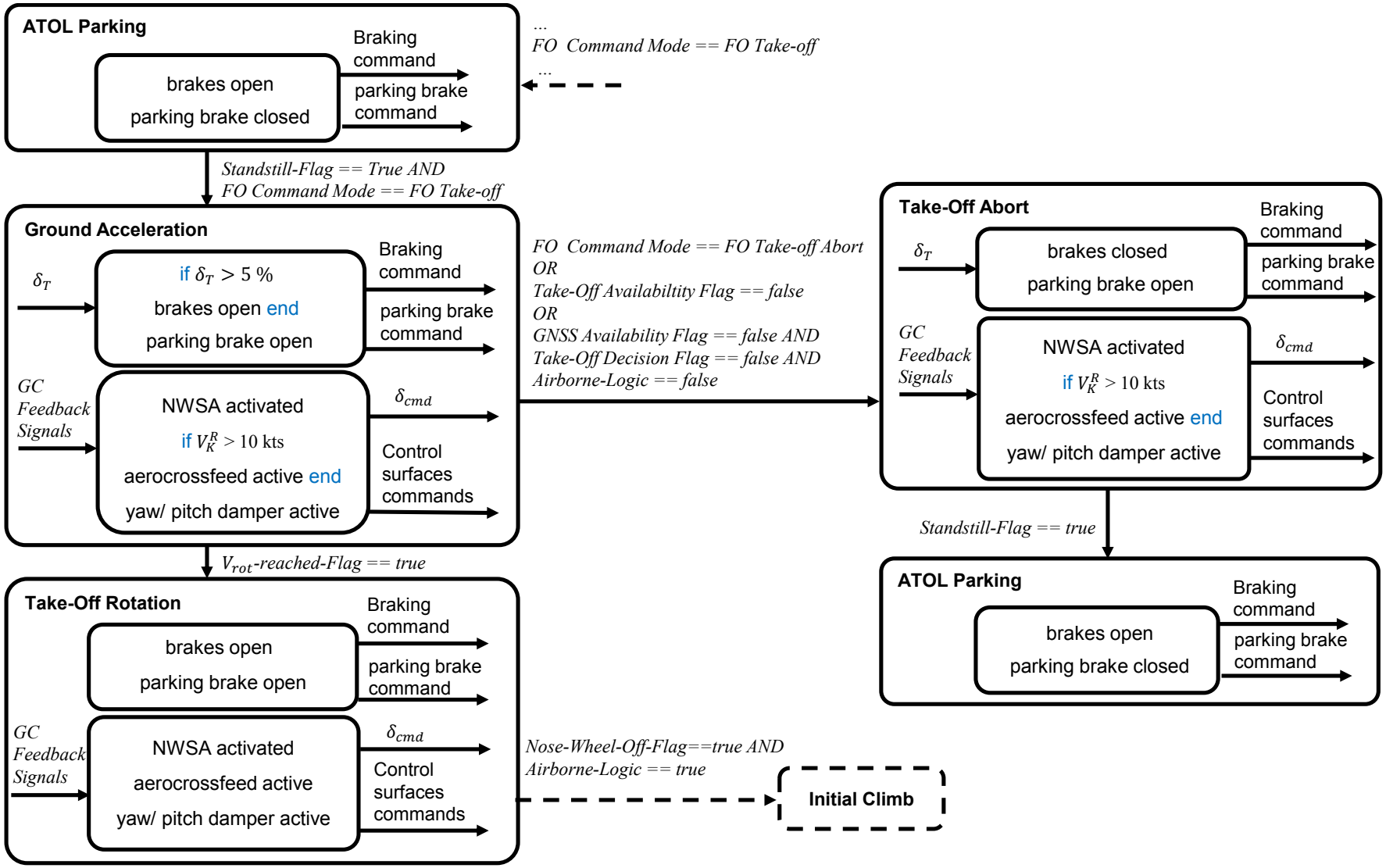


Figure 4.8: SAGITTA ground controller moding for take-off

Landing Derotation: The derotation phase lasts from the touchdown of both main wheels until a defined pitch angle by a negative pitch rate. The ATOL system triggers the ATOL sub-mode derotation when the sensors of both main wheels measure weight on wheel. During the derotation the aero-crossfeed of the ground controller is deactivated to avoid yaw inputs. At this stage of the landing maneuver it is not required to track the centerline immediately but to endure the touchdown with yaw tendency to the runway direction. Since the landing of the SAGITTA demonstrator does not consider a decrab maneuver, the aircraft can touch down with a crab angle. Due to the crab angle at the touchdown, a slip angle at both main tires can occur. The slip angle causes a lateral force on both main wheel tires, which result in a torque around the yaw axes of the aircraft. The torque yaws the aircraft into the runway direction. This physical effect enables landing with crab angle. Only the yaw damper is active. During the derotation the nose wheel angle δ points in the runway direction to avoid a lateral force on the nose wheel. The resulting torque by the slip angle is visualized by Fig. 4.9.

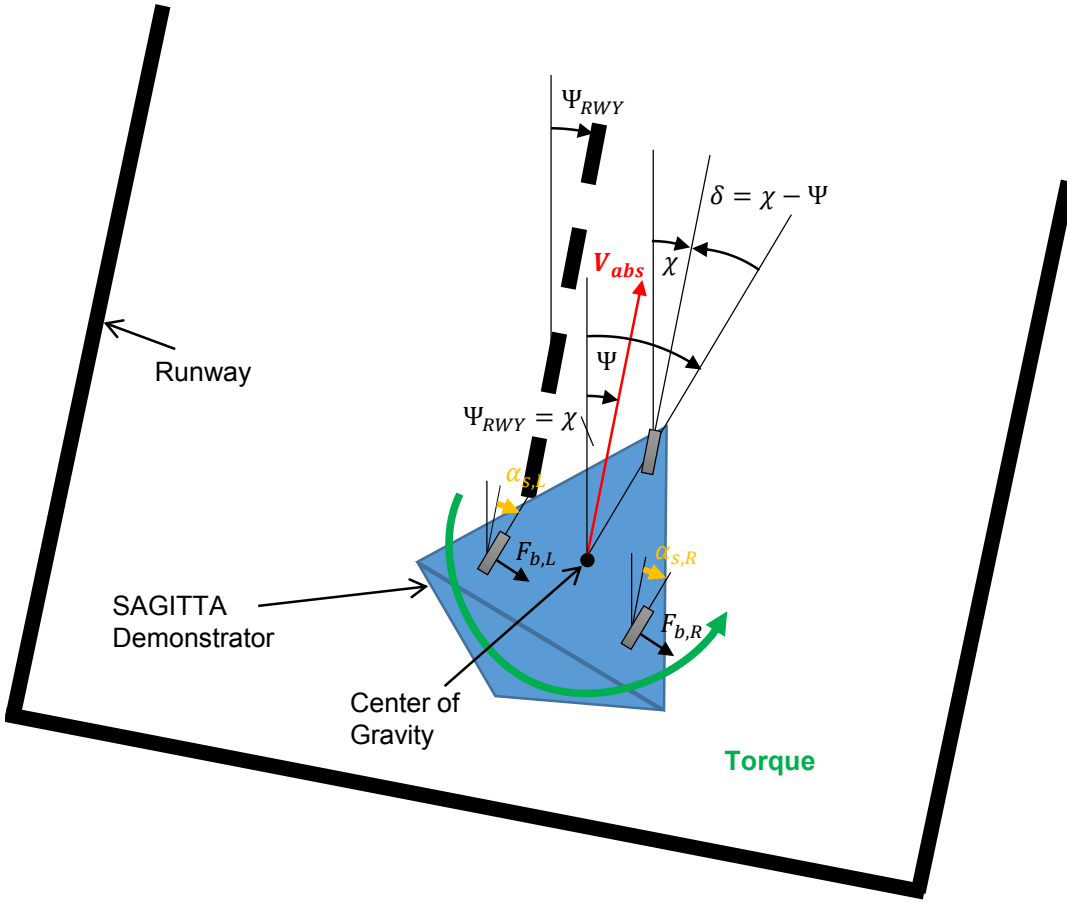


Figure 4.9: Ground controller derotation mode at touch down

Ground Deceleration: After the derotation phase the ATOL module changes into the ground deceleration sub-mode. During the ground deceleration the nose wheel steering, the aero-crossfeed and all dampers are active equivalent to the ground acceleration mode during take-off. The aircraft decelerates by the brake command. The brakes are commanded based on two conditions. The brakes are set if the ground deceleration mode is at least active for 2 s or if the angular displacement between the runway direction and the aircraft heading is higher than 2.5 deg.

Default: Open brakes and a open parking brake are the default brake settings. If the ground controller is active and the ATOL system is not in parking mode, ground acceleration mode, take-off abort mode or the ground deceleration mode, the default brake settings are given.

All of the branches of the ground controller structure in Fig. 4.7 are enabled in the nominal operation during the ground acceleration, the take-off abort and the ground deceleration. During these sub-modes the ATOL module commands a fixed and predefined throttle setting. During the take-off phase maximum throttle is driven. During the deceleration and the take-off abort the throttle command is cut to idle and the brakes are activated.

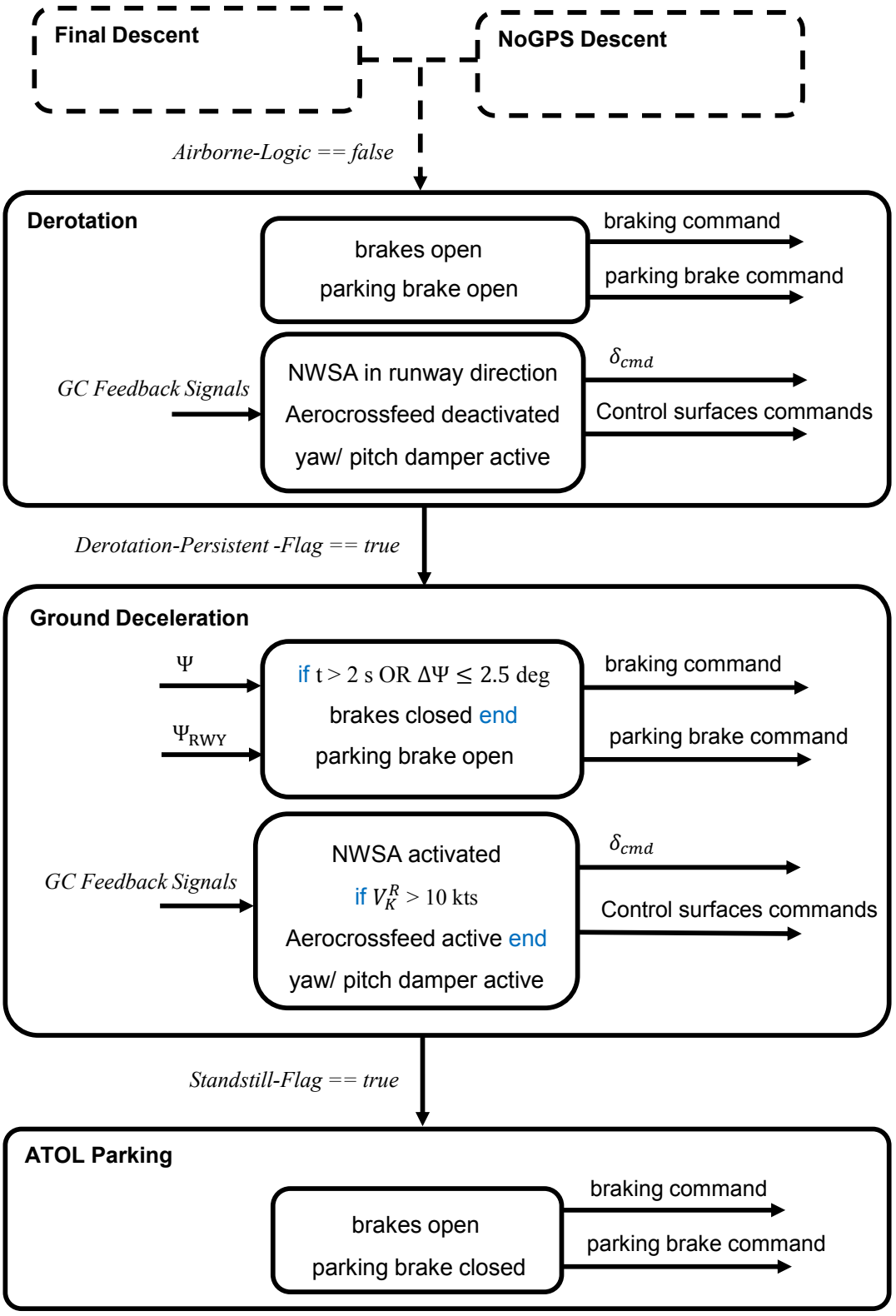


Figure 4.10: SAGITTA ground controller moding for landing

4.3.5 Development of the Taxi Test Software for Online Gain Set Selection[14]

The initial gain sets are preselected based on systematic model-in-the-loop simulation (see next section). Since the aircraft on ground model features limited fidelity due to unmodeled (high-order/ nonlinear) dynamics and unknown uncertainties, the final gain layout of the ground controller design is accomplished via systematic tuning in the course of several taxi test campaigns by use of the dedicated taxi test software detailed in the next section.

Since special features are required for testing, evaluation and gain tuning of the ground controller, which are not available in the final first flight software version, an additional taxi test software has to be designed. Unnecessary functionalities for on-ground operations within the flight control software are replaced by specific functions. In the following section, which has been published in Ref. [114] by the author, it is explained how the requirements and their compliance are realized by reconfiguring the operating functions of the FMCS to enable the features for ground controller testing.

Requirement 1: The taxi test software needs to be embedded into the designated hardware and software environment of the designated flight software, i.e. equivalent interfaces and the same flight management control stations need to be used.
Compliance 1: Therefore, the command signals in Table 4.1 are reconfigured for the taxi test software, which are not used during the ground control of the aircraft. In the right column of Table 4.1 it is indicated how the signals are replaced for the taxi test software. For simplicity and failure avoidance, parts of the flight software which are inactive and not necessary for on-ground operations are deleted. Thus, according to Fig. 4.4, the trajectory generation module and the AFCS are removed for the taxi test software. Equivalent to the flight software for take-off, the taxi test cases can be started by selecting the operation mode ATOL take-off. Then, the brakes are released and the aircraft is cleared for automatic taxiing. Therefore, it is required that all ATOL availability conditions for take-off are fulfilled. In order to fulfill the ATOL availability conditions the correct runway direction needs to be set according to the position and the direction of the next taxi run. This can be done by using the *Ctrl Switch 5* to choose the runway direction.

Requirement 2: The ground tests need to verify the tracking behavior at different speeds and in different deviation situations. Therefore it is necessary to excite the ground controller by external commands. One possibility for excitation of the ground controller is to change the lateral offset command such that the ground controller is forced to compensate the resulting

[14] This section has been published by the author in Ref. [114].

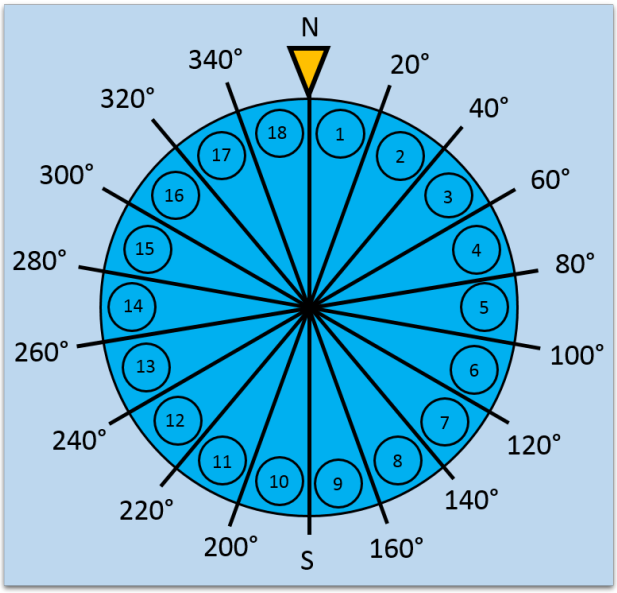


Figure 4.11: Gain set selection via recycling the heading control within the taxi test software

control deviation. For a stepwise approach the FO should be able to excite the ground controller with two different lateral offsets. White markings on the runway centerline of the test range appeared to be slippy under wet weather conditions which leads to different runway surface conditions. To avoid the latter influencing the taxi tests, it is required to perform the taxi test points with predefined lateral offset to the centerline.
Compliance 2: The offset command to the ground controller can be triggered by the FO by activating *Ctrl Switch 2*. After the *Ctrl Switch 2* is switched to ON, the offset command changes from its default lateral offset to the predetermined lateral offset amplitude, which can be set by the *Ctrl Switch 4*. Depending on the position of the aircraft regarding the centerline the FO needs to select the direction of the offset maneuver by *Ctrl Switch 3* such that the aircraft doesn't cross the white markings.

Requirement 3: Since the driving dynamics change rapidly at different velocities (Ref. [57]) the FO shall be able to control the velocity of the aircraft via the FMCS.
Compliance 3: The operating function *speed control* is recycled for the throttle setting. Thus, instead of controlling the velocity of the aircraft during flight, the FO can command the throttle setting to the engines directly in the taxi test software. This enables the conduction of test points with approximately constant speed by changing the throttle command appropriately.

Requirement 4: In order to improve the ground controller performance, the taxi test software requires the selection of a gain set from a range of preselected gain sets based on performed model-in-the-simulations.

Compliance 4: The FO can select 18 different gain sets by the heading control command, which is usually utilized for the AFCS. Depending on the range of the heading command the FO can select the gain sets. The test crew in the QLT can give confirmation to the FO, if a gain set is changed successfully. In the taxi test software a control observation parameter was recycled such that the FCC gives feedback to the ground station which gain set is selected. The realization of the gain set selection is depicted in Fig. 4.11. The gain sets can only be changed when the *Ctrl Switch 8* is activated and the aircraft remains in parking position to avoid setting changes of the ground controller during the test point.

Requirement 5: Within the flight software the take-off maneuver is fixed on a defined runway and direction. Since for testing it is more convenient that the aircraft can be commanded automatically back and forth on the runway, the FO should be able to perform automatic test-points in both runway directions.
Compliance 5: If the *Ctrl Switch 8* is ON and the aircraft is in the parking mode, the FO can set the runway direction with *Ctrl Switch 5*.

Requirement 6: Although the aerodynamic behavior of the control surfaces was analyzed in simulation and windtunnel (see Ref. [129]), the aerodynamic characteristics on ground are unknown and very uncertain. Therefore it is necessary to disable the control surfaces within the taxi test software to be able to analyze the influence of the nose wheel steering and the control surfaces separately.
Compliance 6: The selection of the *Link Loss List* is reconfigured for activation of the mid-board and outboard flaps for yawing moment augmentation on ground.

Requirement 7: It shall be ensured that only during parking of the aircraft internal settings of the ground controller and the runway direction can be changed by the flight operator to avoid unforeseen and unexpected aircraft behavior due to operating errors.
Compliance 7: The FO can change only the test settings, e.g. the runway direction, the gain set or the flaps activation, if the aircraft is in the parking mode.

Requirement 8: For the case of a GPS loss, it is required to test the back-up control structure, too. Therefore, the FO needs to be able to trigger the back-up control law of the ground controller.
Compliance 8: The FO is able to trigger the back-up control structure by switching *Ctrl Switch 6* to ON provoking an overwriting of the GPS status flag of the monitoring system. If the FO triggers a take-off abort due to an emergency, the taxi test software switches automatically back to the nominal ground control structure.

Requirement 9: Finally, it is required that the FO can execute the task without any high workload in order to keep track of the aircraft and to be able to perform emergency tasks if necessary.

Compliance 9: This requirement is met implicitly by the previous fulfillment of the requirements. Before starting the test point, the FO selects the predefined settings, i.e. the runway direction, the gain set and offset amplitude. Then the FO triggers *ATOL take-off*. By selecting the appropriate throttle setting the aircraft starts to accelerate. During rolling the FO only needs to adapt the appropriate throttle to achieve the dedicated speed and to trigger the offset commands. If the test point is conducted, he can either trigger *Ctrl Switch 1* to ON in order to activate the brake or take-off abort in emergency situations. Table 4.2 shows an example of a typical test procedure during the ground controller testing. The test procedure ensures the moderate workload for the FO.

Table 4.2: Typical test procedure for the conduction of a test case with offset command at dedicated velocity.

Test Card Procedure - Offset Command
1. EP turns the A/C and should position the A/C with a correct 3 m offset to centerline in EAST RWY direction. $\rightarrow$ FO changes to *FO parking mode* $\rightarrow$ FO switches *Ctrl Switch 5* to OFF (Runway direction West $\rightarrow$ East), in order to set the RWY direction $\rightarrow$ FO switches *Ctrl Switch 4* to ON (for 5m offset trigger) $\rightarrow$ It is checked, if ATOL is available $\rightarrow$ QLT checks, if correct Gain Set is selected $\rightarrow$ FO selects *ATOL take-off* mode
2. FO accelerates the A/C slowly via the Speed Control on the FMCS to the designated speed
3. When speed is stable, FO triggers offset via activation of *Ctrl Switch 2*
4. After A/C stabilized at commanded offset and the QLT gave a call, FO commands A/C back to original offset to RWY centerline
5. When A/C reached the commanded offset again and QLT gave the confirmation, FO brakes via *Ctrl Switch 1*.

4.3.6 Pre-Selection of Gain Sets[15]

Before the conduction of ground and taxi tests with the real aircraft, a certain number of initial gain sets is pre-selected based on systematic model-in-the-loop simulation at different speeds (i.e. 30 and 65 kts) using the landing gear model described in Sec. 4.2.2. Thereby, qualitative as well as quantitative aspects are considered with regard to step-like lateral offset excitations of the GC at different speeds. On the one hand, the amount of overshoot after a step command, the steady-state tracking error due to disturbances (e.g. crosswind, brake asymmetry, backlash), and rising and settling time are evaluated. On the other hand, the tracking behavior in terms of frequency and damping is considered qualitatively for all three control loops. Since in the nominal flight control software the ground controller cannot be excited by lateral offset commands, the specific taxi test software, explained in the previous section, is developed for simulation and real live testing on the runway. In Fig. 4.12 and 4.13 two typical simulated scenarios are plotted to underline the determination of appropriate gain sets for each control loop.
The step responses of three pre-selected gain sets are shown in blue, magenta and green color. Compared to the *Gain Set 1* (blue color), *Gain Set 2* (magenta color) comprises an increased gain k_{χ} and a decreased gain $k_{r,N}$. However, the green lines demonstrate *Gain Set 3* with just increased gain k_{χ}, compared to *Gain Set 1*.
Each plot consists of five subplots. In the first subplot, the ground speed is shown. The next three subplots display the three control loop signals of the GC. The second subplot shows the outer loop signals of the GC and thus the commanded and the current lateral deviation of the aircraft with regard to the centerline. The third subplot represents the middle loop of the GC. The commanded course and the current course of the aircraft are plotted. At low speeds (<1 m/s) the course is replaced by the heading signal since at these speeds no reliable course information exists. The black line represents the heading of the runway. The signals of the inner loop are shown in the fourth subplot. It can be seen that the yaw rate of the aircraft follows adequately the commanded yaw rate. The last subplot shows the commanded nose wheel steering angle.
Fig. 4.12 illustrates the simulated reaction of the GC with the three different gain sets due to the given lateral offset command at ground speed V_K^R=65 kts without any disturbances.
In contrast, Fig. 4.13 exhibits the GC behavior under 5 kts crosswind, which is initiated at simulation time t=4 s, plus additionally a lateral offset command at time t=25 s. The quantitative criteria of these two tests are compiled in Table 4.3. As the plots show, for the gain selection a trade-off needs to be considered between the quickness of the response, tracking performance, robustness and stability characteristics. If higher gain values are chosen such as the *Gain Set 2* and *Gain Set 3*, the quickness of the response and the robustness

[15]This section has been published by the author in Ref. [115].

Table 4.3: As shown in Fig. 4.12 and 4.13, regarding the gain layout a trade-off between response quickness, tracking performance, stability characteristics and robustness has to be considered.

Gain Set	1	2	3
Rise Time (in s)	4.00	3.24	3.17
Settling Time (in s)	15.61	9.62	9.62
Overshoot (in %)	33.0	13.5	17.5
Decay Ratio (in %)	27.3	40.7	57.1
Steady-State Offset (in m)			
without Disturbances	0	0	0
with 5 kts crosswind	2.48	2.25	1.54

against wind are increased due to the shorter rise and settling time and lower steady-state error. On the other hand, as it may be seen in the fourth and fifth subplot of Fig. 4.12, the yaw rate and the nose wheel steering angle reveal a high-frequency oscillation tendency, which indicates a degraded stability margin.

For instance, due to the increased gain k_χ, a higher initial overshoot of the yaw rate occurs, accompanied by clearly visible developed oscillations of the nose wheel steering angle command, nose wheel angle and yaw rate.

Such oscillations may be amplified in an unacceptable way by the backlash between steering actuator and the nose wheel steering angle, due to the dead zone, provoking increased control deviation.

In order to verify the robustness of the GC layout, a backlash between nose wheel actuator and nose wheel steering angle of 1 deg and a brake asymmetry of 10 % on the main wheels are implemented within the landing gear model.

Similar scenarios as described above have been simulated to identify gain sets, which reveal improper amplifications of the nose wheel steering angle oscillations.

The backlash represents a nonlinearity of the driving dynamics and causes a dead zone in the system provoking decreased stability margin. Although higher gains $k_{r,N}$ and k_χ enhance the quickness of the GC response verified by simulations, they lack due to their lower stability margin. In on-ground simulations with a backlash of 1 deg these increased gains have been identified as inappropriate. For this reason, the three gain sets in Fig. 4.12 are preselected for verification trials within taxi tests.

Since the aircraft on ground model features limited fidelity due to unmodeled higher-order dynamics and unknown uncertainties, the final gain layout of the GC has been accomplished via two taxi test campaigns by use of the developed taxi test software, explained in Sec. 4.3.5.

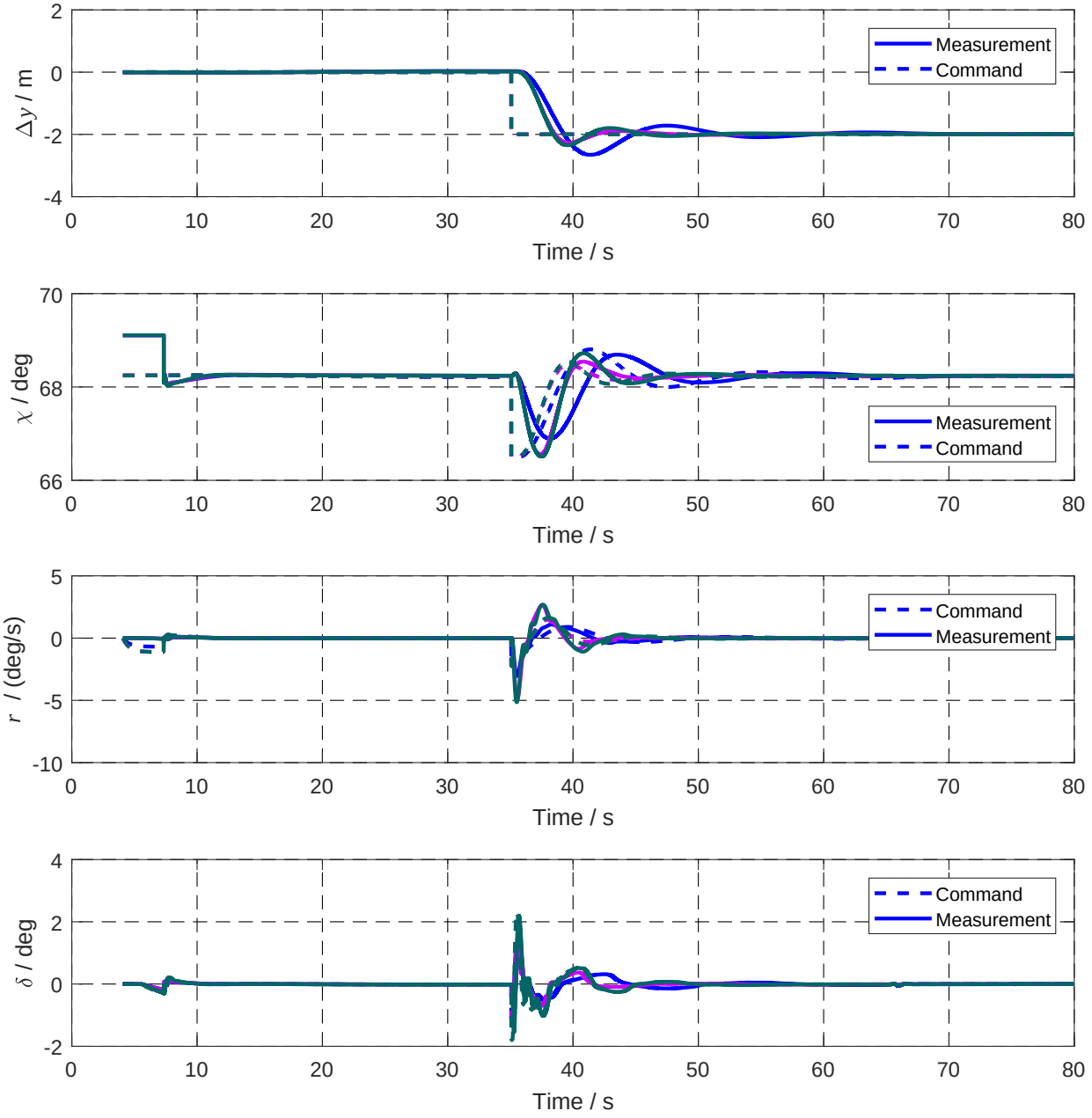


Figure 4.12: SAGITTA on-ground simulation without disturbances using three different gain sets (see Tab. 4.3)

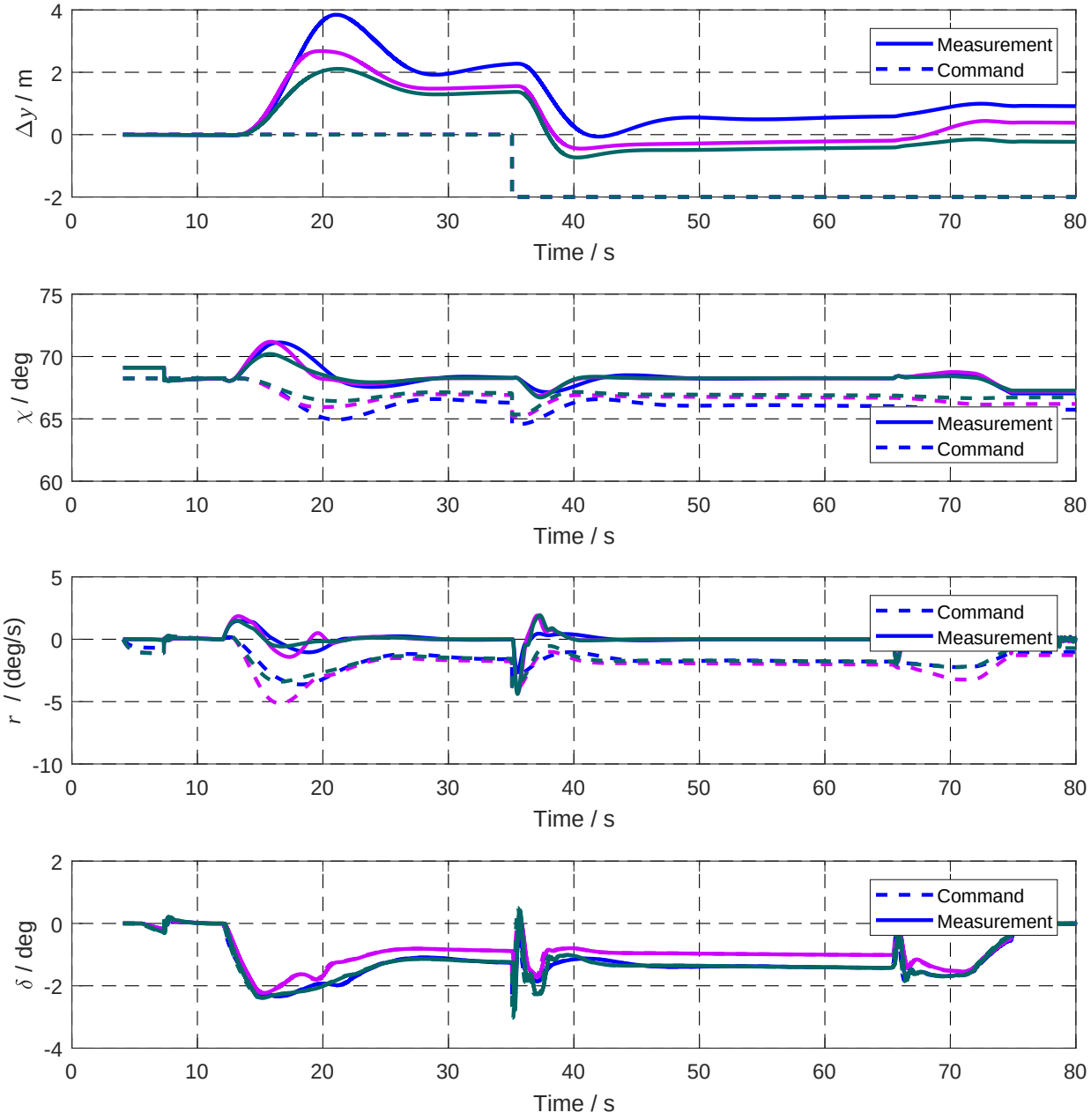
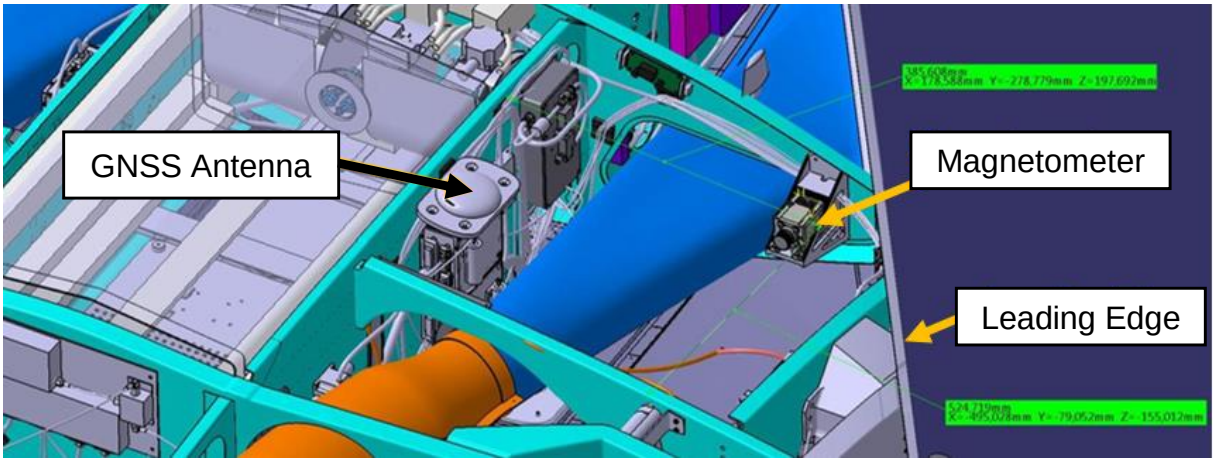


Figure 4.13: SAGITTA on-ground simulation with 5 kts crosswind using three different gain sets (see Tab. 4.3)

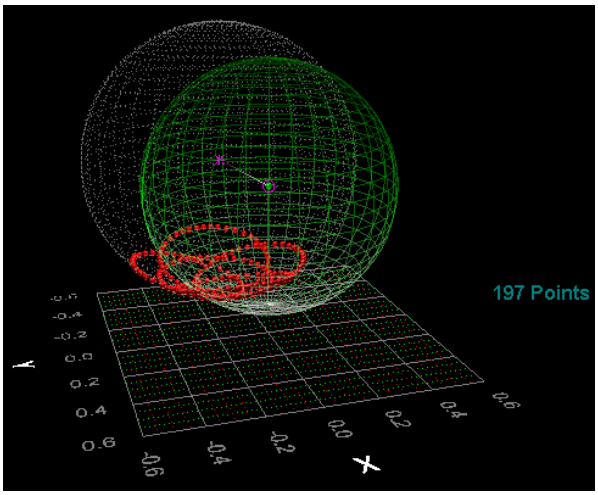
4.4 Real-Live On-Ground Testing For First Flight

This section comprises the necessary on-ground testing of the SAGITTA demonstrator to gain first flight maturity. Thereby, the focus is set on the verification of the flight control system considering the defined system boundaries in Sec. 4.1.3 and Fig. 4.2. Apparently, during the following explained towing and taxi tests, the other system components and the overall system are examined carefully and verified as well. This includes the propulsion system, the actuator and landing gear system group, the communication system including the data links and the ground station (with EPCS and FMCS). The reader is referred to Ref. [135, 122] for these system components.



(a) CAD visualization of the installed magnetometer (image created using Ref. [136])

(b) Setup for magnetometer calibration



(c) Illustration of the spherical fit of the magnetic field distortion (image created using Ref. [137])

Figure 4.14: Magnetometer calibration of the SAGITTA demonstrator

4.4.1 Sensor Integration and Testing

4.4.1.1 Sensor Selection and Integration

The major sensor component of the flight control system is the navigation system. As navigation device the *iTraceRT-M200-OLC* (see Ref. [138]) from the company *iMAR Navigation GmbH* is integrated into the aircraft. It contains an inertial measurement unit of microelectromechanical sensors (MEMS) and a GPS receiver for satellite- and ground based augmentation.
For improvement of position accuracy, ground-based Differential GPS (DGPS) is additionally applied. Therefore, a DGPS base station is assembled near the runway, which comprises a GPS antenna and a *NovAtel 'OEM628'* GPS receiver. The GPS solution is sent via the ground station and the mentioned data links to the aircraft and received by the navigation system. The navigation system was installed on the x-axis of the aircraft body-frame(roll axis) in the aircraft to avoid unnecessary centrifugal acceleration measurements by correct installation. The system provides all necessary kinematic signals such as angular rates, acceleration, attitude, velocity and position of the vehicle.
For valid heading information in standstill and in case of a GNSS loss, a low size magnetometer *iMAG-DMC-LS* from the company *iMAR Navigation GmbH* is installed near the right leading edge of the aircraft, as visualized in Fig. 4.14(a), and connected to the navigation system. The magnetometer provides three-dimensional measurements of the environmental earth's magnetic field to calculate the magnetic heading of the aircraft.
The air data system (see Ref. [139]) from the company *Stock Flight Systems* is installed in the aircraft including a nose boom from the company *SpaceAge Control Inc.* with two vanes. It measures static and dynamic pressure, pressure altitude, vertical speed, indicated and true airspeed. The two vanes of the noseboom are installed for angle of attack and angle of sideslip measurements.
In nominal operation, the navigation system uses the GNSS solution, the IMU measurements and the air data computer measurements (except the vanes) to run an extended Kalman filter for data fusion. If GNSS loss occurs, the navigation switches and utilizes the magnetometer measurements and the air data computer measurements for compensation.
Separated from the navigation system, a *Micro Radar Altimeter* from the company *Smart Microwave Sensors GmbH* is integrated for the initiation of the flare maneuver during automatic landing.

4.4.1.2 Laser Leveling of Navigation System

For a proper installation of the navigation system with the integrated gyros and accelerometers, a laser beam projector was used to compensate the attitude misalignment between the navigation system and the aircraft body frame. Therefore, three mirrors are attached

on the aircraft surface in a triangular shape at known positions of the aircraft's body frame. Using the laser beam projector the current attitude of aircraft can be determined by measuring the relative height of the three mirrors. At the same time attitude measurements of the navigation system are collected and an average is taken over a period of more than five minutes. Using the attitude measurements by the navigation system and the laser, the attitude misalignment is calculated. The misalignment is inserted into the software of the navigation system for leveling.

4.4.1.3 Magnetometer Calibration

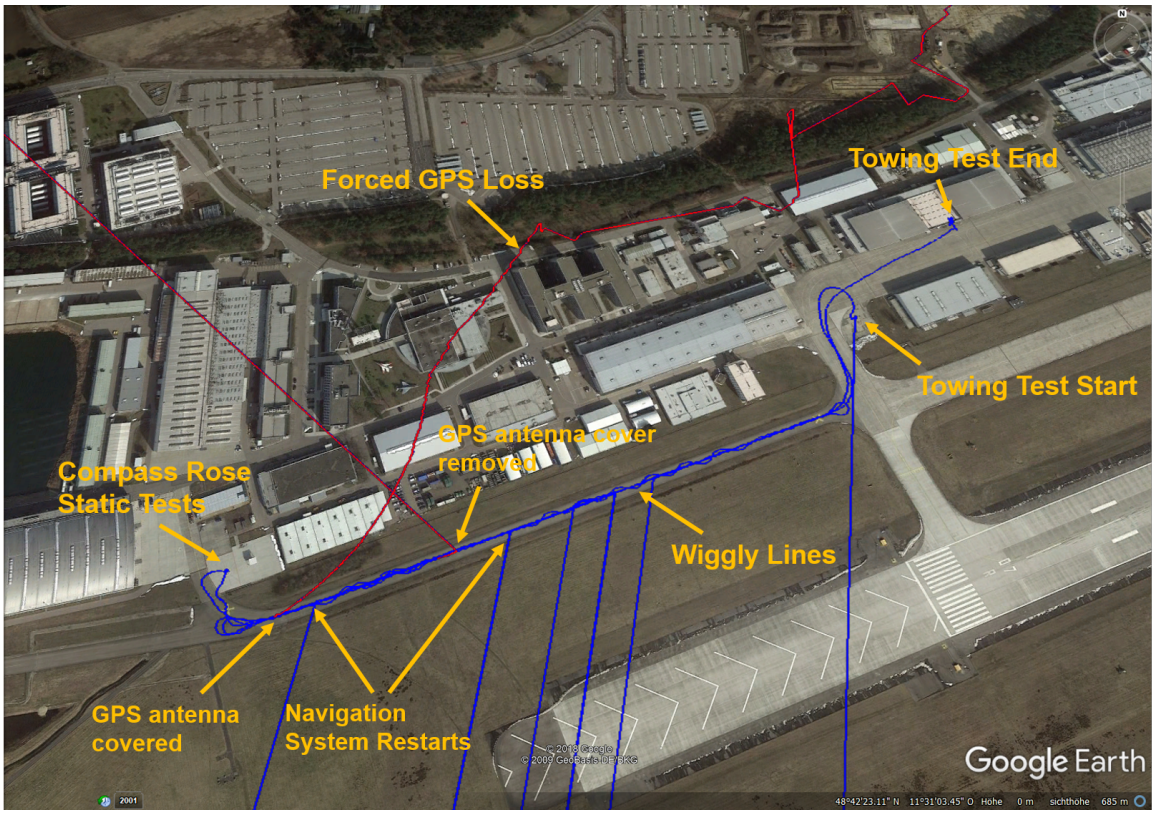
Valid heading information are essential for automatic take-off and landing maneuvers in case of a GNSS loss and in standstill. Nominally, the heading information is gained by a track alignment using the GNSS velocity after a velocity threshold is exceeded. In case of a GNSS loss, the track alignment cannot be used anymore. In standstill, before take-off, the track alignment is not a suitable option since no velocity has been established yet. Nevertheless, a valid heading information is required for automatic take-off, because otherwise the aircraft would accelerate without having the current direction available.
Therefore, a magnetometer is inevitable, for the operation of an UAV with automatic take-off and landing capabilities. Since the magnetic field is disturbed by the aircraft's avionics and frame, a magnetometer calibration is necessary for magnetic disturbance compensation. The magnetic field is influenced by hard-magnetic and/or soft-magnetic disturbances. According to Ref. [140], soft-magnetic disturbances are field distortions caused by magnetic materials with complex geometry. Hard-magnetic disturbances are induced by magnets and electric currents and require a rather simple geometry of compensation measurements. In order to compensate for these disturbances, a magnetometer calibration is mandatory. Apparently, a two-dimensional magnetometer calibration is not sufficient for an UAV since magnetic distortions in the third dimensions impact the magnetic field by changing bank and pitch angles in flight.
Usually, in piloted aircraft, identification flights are performed and the collected data of the magnetic field is used to calibrate the magnetometer. Since the mentioned UAV has never flown before and the magnetometer is required for automatic take-off at the first flight, the following approach is conducted for gaining first flight maturity. To enable a proper three-dimensional magnetometer calibration, the aircraft is lifted on a stack of pallets to take off the aircraft from the ground to avoid any ground disturbances.
For the calibration the aircraft is fixated on the stack and the aircraft's heading is changed stepwise 30 deg. At any step, the magnetic field is measured over a 5 s time period and the average point of the magnetic field is saved. After one turn, the attitude of the aircraft is changed on the stack. In total, the aircraft is turned five times with five different attitudes (i.e. wings level, negative/positive pitch angle, negative/positive bank angle), such that the

attitude envelope is covered as far as possible. Fig. 4.14(b) shows the aircraft during the magnetometer calibration with negative pitch angle.
In Fig. 4.14(c), the measurement points of the magnetic field during one magnetometer calibration of the demonstrator are illustrated. The figure shows a screenshot of the *Microstrain* magnetometer calibration software of the *Microstrain 3DM-GX3-25* Attitude Heading Reference System (AHRS) (see Ref. [141]), which is integrated as well in the aircraft for verification only. The AHRS sensor signals are not used in first flight in any feedback loop of the control laws. The purple circle illustrates the center of the sensor and the purple star represents the current center of a sphere/ellipsoid which the saved data is creating (Ref. [142]). The created sphere/ellipsoid represents the distortion of the magnetic field. Together with a three-dimensional offset vector of the magnetic field, the spherical/or ellipsoid fit is result of the magnetometer calibration. After completing, the data is considered as magnetic field disturbance in the heading estimation filter of the navigation device. The red circles in Fig. 4.14(c) represent measurement points of the magnetic field. The red circles show each turn of the aircraft during the calibration. As the following explained towing tests underline, five different attitudes with 30 deg heading steps and the shown procedure are sufficient for determination of the magnetic field distortion.

4.4.1.4 Towing Tests for Sensor Verification

For verification of the correct sensor integration and their functionalities within the aircraft, towing tests with several test points are conducted. Thereby, the aircraft is pulled by a towing vehicle as shown in Fig. 4.15(a). The towing tests are performed at an aerodrome from one of the industrial partners within the project. The runway direction are to the north-east Ψ_{RWY}=68.2 deg and to the south-west Ψ_{RWY}=248.2 deg.
The focus of the introduced towing tests within this thesis is set mainly on the verification of the navigation system solution in combination with the magnetometer based on qualitative assessment of the output signals. The air data system is primarily verified during the taxi test campaigns, where higher aircraft speeds are reached. The taxi tests are explained in the next section. The air data system and the navigation system are the primarily sensor source for the flight control system and are thus essential for first flight qualification.
The towing test covers test points with regard to the initialization procedure, the position/ velocity and magnetometer aiding, the acceleration and angular rates and the heading solution, and its behavior with and without GPS solution. Fig. 4.16 shows a selected part of the representative towing test with the most relevant data signals from the towing test start for the on-ground verification. During the test all systems within the aircraft are powered (except the engines) to mimic the closest environment of the navigation system to the first flight.
One representative towing test is shown in Fig. 4.15(b) and 4.16. Fig. 4.15(b) illustrates the track of the towed aircraft in blue color, which is measured by the integrated navigation

(a) The SAGITTA demonstrator with towing vehicle



(b) Track of aircraft during an exemplary towing test (created with Ref. [143])

Figure 4.15: SAGITTA demonstrator during towing tests

solution. The red color displays the measured position during a GNSS loss, which is forced by covering the GPS antenna with alumina foil to verify the behavior of the navigation system while no GNSS signals are received. Fig. 4.16 illustrates extracted data measured and filtered by the navigation system.
After initialization, the navigation system executes an attitude alignment. The bank and pitch angle are mainly estimated from the accelerometer measurements. The heading can be primarily calculated from the measured earth rotational rate by averaging over the measured angular rates using the bank and pitch angle estimations, if higher-budget sensors are available (e.g. fiber-optic or laser gyroscopic sensors) (Ref. [144]). The present navigation system uses MEMS sensors, which are not sufficient for estimation of the heading from the earth rotational rate. Thus, in standstill after initialization, the navigation system has invalid heading information.
The navigation system uses two different approaches to gain valid heading information. If the navigation system has GNSS available and the aircraft has established sufficient speed, the heading is initialized and calculated primarily by the covered track. If GNSS information are not available, the navigation system relies on the determined heading from the magnetometer.
Due to magnetic field disturbance during the start-up of the vehicle the heading is initialized by using the GNSS information after a certain velocity threshold (i.e. V> 2 m/s) is exceeded. Thereon, if the yaw standard deviation, which is calculated by the Kalman filter of the navigation system, exceeds a defined threshold, a magnetometer aiding is conducted. This is usually the case, if the aircraft is again in standstill after track alignment or at very low speeds.
At the navigation system initialization, valid heading information are not available, which can be seen by the high yaw standard deviation (see left bottom plot in Fig. 4.16). When the aircraft accelerates the yaw standard deviation drops and a valid heading information is available, which is forced by the track alignment. The velocity aiding and the barometric aiding are performed since GPS and air data are available. During low speed and standstill periods also the magnetometer aiding is performed.
As the track highlights, the aircraft moves towards the taxiway, which is parallel to the runway of the aerodrome and thus has the same direction as the runway. As the aircraft moves in a stop-and-go manner straight on the taxiway, the navigation solution outputs approximately the runway direction (see ground speed V and heading Ψ plot in the right column of Fig. 4.16). If the standard deviation is above the threshold, the magnetometer aiding is performed as desired. If the magnetometer calibration has been not conducted, the magnetic heading would eventually be incorrect, and the navigation solution would reject the aiding.
After the verified straight run, the aircraft is towed towards a compass rose of the aerodrome, where exact geographic direction are drawn on the ground. The aircraft is turned in each of

the four geographic directions until a magnetometer aiding is performed. This is visible in the heading plot in Fig. 4.16 by the step-like course graph. The navigation system shows the desired heading signal.
After the statics at the compass rose are completed, the aircraft is position on the taxiway in the north-east direction of the runway and the GPS antenna is covered. As the velocity aiding plot illustrates, the navigation system does not receive GNSS signals anymore and the velocity aiding is not performed anymore. The monitoring module of the flight control software identifies such a GPS loss by consistency checks. With the covered GPS antenna, the aircraft is towed straight on the taxiway. Inbetween the aircraft is towed in wiggly lines. Thereby, the position and the displayed ground speed drift away. In contrast to the position and the ground speed (as seen in the track plot in Fig. 4.15(b) and the ground speed plot in Fig. 4.16, the heading signal of the navigation system is still valid since it outputs the runway direction and the heading change by the wiggly lines correctly. Thereby, the magnetometer measurements are applied. During the conduction of the wiggly lines, the navigation system senses the yaw rate signal accordingly. Then, the aircraft is turned and towed in the opposite direction. The GPS antenna cover is removed, and the position and the ground speed is gained again by the position and velocity aiding. At this point, the time plot of Fig. 4.16 ends, but the towing test is proceeded as Fig. 4.15(b) shows. The test point above is repeated with and without GPS.
As the result of the representative towing test shows, by conduction of these towing tests the navigation system with magnetometer is verified by qualitative evaluation of gained towing test data.

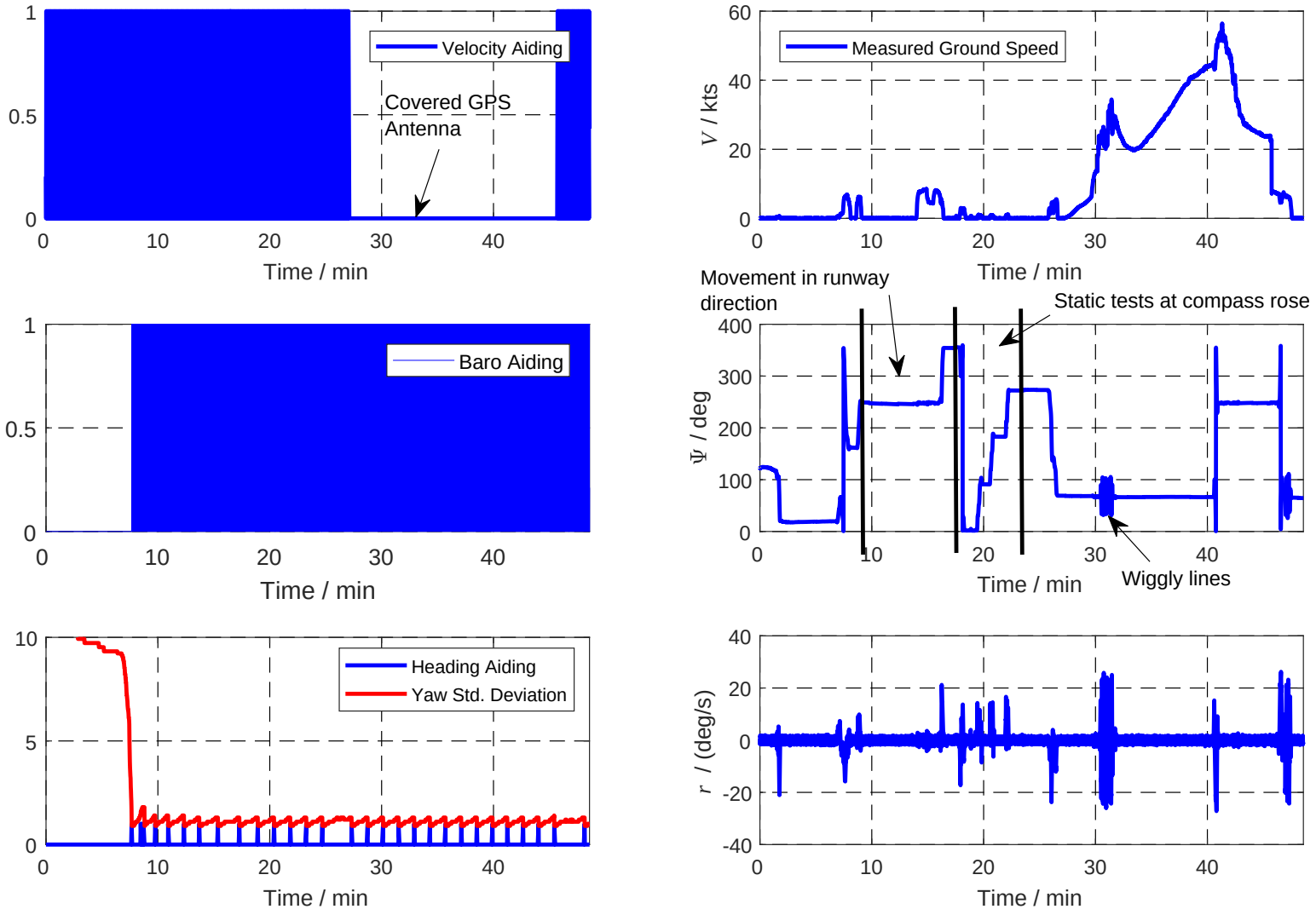


Figure 4.16: Navigation data of towing test for magnetometer verification

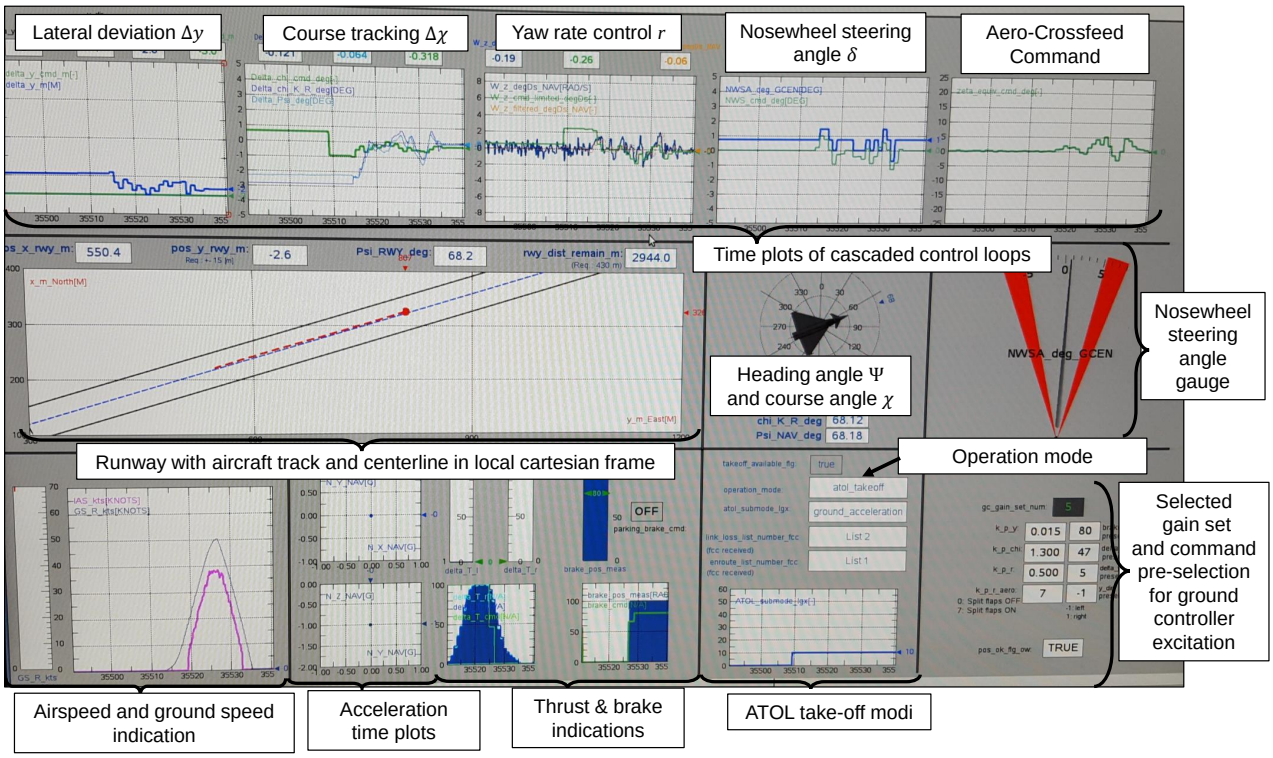


Figure 4.17: Monitoring display for online gain-set layout of the ground controller

4.4.2 Taxi Test Setup and Monitoring

As elaborated further in the following sections, EP controlled taxi tests and automatic taxi tests for the ground controller layout are conducted. The test crew is briefed on the taxi test procedure and the respective test objectives before the test. During the test two external pilots and one system engineer were in visual line of sight to the aircraft. The flight test conductor, who chaired the tests, the flight operator and system engineers were conducting and supvervising the tests in the ground station. The ground station consisted inter alia of the flight management control station (FMCS), the termination system and quick look terminals. The system engineers supervised the aircraft components by monitoring the received data from the down links of the aircraft to the ground station.

In order to analyze and monitor the driving dynamics of the aircraft during the EP controlled taxi tests and to select proper gains of the ground controller, a specific quick look terminal has been designed by the respective system engineer. Fig. 4.17 shows the designed and utilized monitoring display. The upper five signal plots show the different, cascaded control variables with command and measurement signal. These plots enable the analysis of the ground controller performance online during the test. The middle row shows mostly the aircraft position with covered track with respect to the runway. The row at the button shows the aircraft speed, accelerations, the commanded thrust and brake commands. At the lower, right corner the current operational mode (e.g. *EP-on-ground, FO Parking, ATOL take-off*)

and the current ATOL phase are illustrated. In addition, the presently selected gain set is indicated, which is important for the ground controller performance analysis. The gain set display is also important for the gain set confirmation to ensure that the proper gain set is selected by the flight operator and active.

4.4.3 External Pilot Controlled Taxi Tests and Landing Gear Model Validation

4.4.3.1 Overview and Conduction

To test the overall system of the UAV on-ground before first flight and to validate the landing gear model (introduced in Sec. 4.2.2), external pilot controlled taxi tests are conducted. In these tests, the external pilot takes command only and controls the aircraft on-ground. The available command channels are explained in Sec. 4.1.3. Thereby, the EP performs test points and maneuvers, which are predefined and briefed on flight test cards.
With regard to the first flight qualification of the flight control system, the aims of EP controlled taxi tests are the following:

- Verification of the EP on-ground command chain of the flight control system
- Further verification of integrated sensor systems
- Testing and verification of involved system components, which effect the vehicle on-ground performance (e.g. nose wheel steering, brake behavior, engine thrust)
- Landing gear model validation

For the system verification and the landing gear model validation the flight test data is recorded. The EP controlled taxi tests are the first tests, in which the complete system (including running engines) is enabled while the aircraft moves by its own power. Therefore, it is important to verify the entire system with all of its components within these tests (especially the sensor system with regard to the flight control system).
Particularly, the EP ground trials are the first tests, in which the flight control system is engaged actively in the control loop and thus, verify the whole chain from the EP data link receiver via the flight control computer to the control surfaces, the brakes, the nose wheel steering angle and the engine thrust.
For the ground controller layout these EP ground trials are important to analyze the on-ground behavior of the aircraft without an automatic function in-command. In that way, hidden faults of system components are easier to be isolated (i.e. brake asymmetry, backlash etc.) and the ground controller can be excluded from causing possible abnormal behavior.
The landing gear model validation is executed to avoid any unexpected behavior of the aircraft during first automatic test runs. The model accuracy of the landing gear including the

tire model is unknown since uncertainties such as the backlash between the steering actuator and nose wheel steering angle, braking asymmetry and the aircraft behavior against crosswind cannot be equally modeled.
Eventually, the EP controlled taxi tests are performed to gain confidence of the entire aircraft system for the conduction the first automatic taxi tests.
Four different test points at different speeds are executed, i.e. acceleration and deceleration, constant turns at various speeds, as well as lane changes and wiggly lines.
In order to validate the landing gear model the flight test data is extracted and the same commands of the flight operator and external pilots are commanded in the model-in-the-loop simulation. The aircraft is initialized at the same position as in reality. The commands of the test point and the initial position of the aircraft are equivalent between the simulation and the reality. External disturbances, such as wind, brake asymmetry and backlash, were not enabled in the validation process of the landing gear model.
The following subsection introduces conducted test maneuvers at different speeds and shows inter alia comparison plots of the different maneuvers between the real live tests on the runway and the model-in-the-loop simulation of the SAGITTA aircraft. These maneuvers are conducted in different taxi test campaigns and are controlled by the external pilot. The EP ground trials are divided in the following descriptions into first and second phase. In the second phase, the findings of the first phase and the respective improvements are incorporated into the system components and the overall aircraft, which are explained in Sec. 4.4.4

4.4.3.2 The EP Controlled Taxi Test Maneuvers

4.4.3.2.1 Straight Acceleration Runs

To validate the aircraft behavior on ground in the longitudinal motion straight acceleration runs are performed. The ground speed and the longitudinal position of the aircraft relative to the runway are analyzed to evaluate the acceleration and the deceleration behavior:

Description: The EP drives aircraft straight on the runway centerline with different velocities, then gives a steplike brake input.
Initial Specification: For the test, the aircraft is positioned on the runway, aligned with the runway centerline. The EP accelerates the aircraft by a constant throttle lever position, if possible. The velocities should be stabilized approximately at constant values, i.e. 10, 25, 45 kts. The aircraft shall move as far as possible on the runway centerline. The aircraft is trimmed during straight rolling prior to brake activation.
Input Specification: After desired constant velocity is reached, EP shall activate the brake steplike. The EP conducts the test for each velocity three times, starting with small brake command, up to medium and full brake command. The EP shall not try to compensate deviation to centerline, i.e. he shall give no steering angle input.

This maneuver mainly evaluates the acceleration performance, the necessary braking distance and the asymmetric braking behavior. The friction parameters of the landing gear model are changed stepwise to gain the best compliance of the model to the reality.
Fig. 4.18 illustrates an exemplary EP straight acceleration run up to 25 kts at an early stage of the ground tests. The acceleration in reality appears to be higher compared to the simulation, what leads to greater covered distance. The deceleration behavior matches between the simulation and the reality. After 15 s slight yaw rate oscillation occur in reality due to nose wheel steering inputs. This is possibly caused by the actuator backlash between steering actuator and the nose wheel. Since the equivalent EP inputs are given to the simulation, the nose wheel steering inputs are also visible in the simulation signals. When the EP brakes, higher yaw oscillations occur due to a yaw moment caused by a brake asymmetry in reality. In simulation the high yaw moment does not occur.

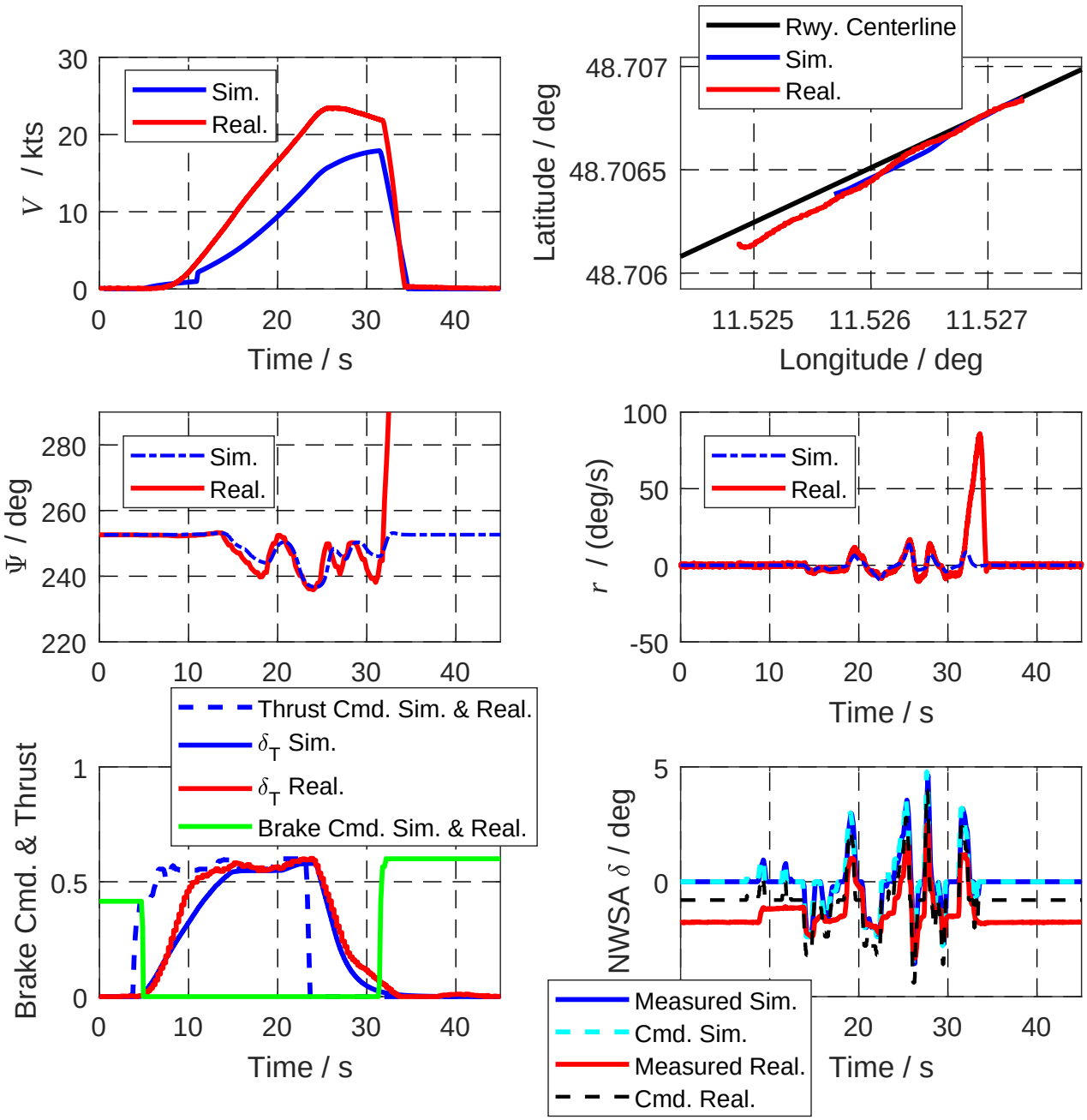


Figure 4.18: EP controlled straight acceleration run up to 25 kts

4.4.3.2.2 Single Lane Changes

The single lane change maneuver is performed by the EP to evaluate the velocity dependency of the aircraft behavior on ground and the aircraft behavior to nose wheel steering inputs. The parameters in focus for this maneuver are the yaw rate, the lateral acceleration, the nose wheel steering angle and the roll angle.

Description: EP commands a steering angle step input. After specific time, EP should steer A/C back to centerline by the same step steering command, but to the other side. Examined with different velocities.
Initial Specification: For the test, the aircraft is positioned on the runway, aligned with the runway centerline. The EP accelerates the aircraft by a constant throttle lever position, if possible. The velocities should be stabilized approximately at the constant values, i.e. 10 kts and 25 kts. The aircraft shall move as far as possible on the runway centerline.
Input Specification: EP shall steer A/C straight on runway at approximately constant speed, then EP gives step steer input on nose wheel. EP shall conduct test for three velocities with different values of steering angle commands. Different values of the steering angle commands for each velocities are selected. After approximately 2 s EP steers the aircraft back in direction of the runway, parallel to centerline.

For proper steering angle command EP should set up dual rate on the used remote control. By the dual rate the range and limit of the input command can be set. The maneuver is performed to determine the vehicle reaction to a leap of the nose wheel angle.
Fig. 4.19 shows a comparison plot between the simulation and the conducted single lane change maneuver in reality at 25 kts. This maneuver is conducted within the second phase of the EP controlled taxi tests, in which configuration improvements have been realized. The aircraft is accelerated by the EP after 20 s. As the plot shows, the accelerations of the simulation and reality matches. Later at 30 s, the EP performs the single lane change with steplike nose wheel steering angle commands up to 4 deg to the left and to the right. The lateral acceleration signal f_y and the yaw rate signal r are qualitatively almost equivalent between simulation and reality. The stationary velocity differs slightly, what causes the difference in the track plot. The deceleration phase coincides.

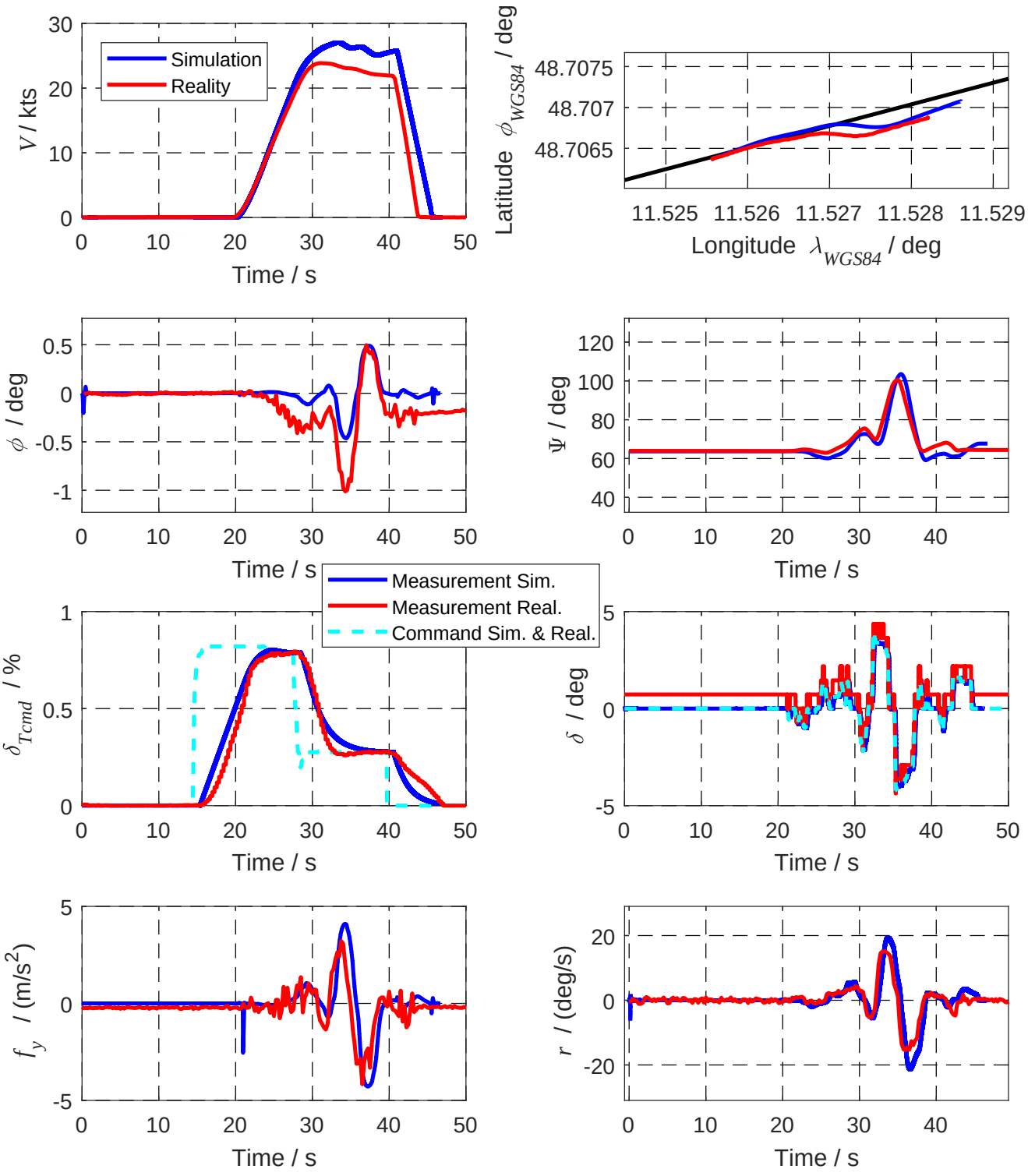


Figure 4.19: Manual single lane change maneuver by external pilot

4.4.3.2.3 Steady State Cornering

For this maneuver the EP commands a steady state cornering of the aircraft with different velocities. The maneuver is performed for evaluating the match between simulation model and nose wheel steering behavior. The yaw rate, the lateral acceleration and the diameter of the performed circles are the focus for this maneuver to validate the cornering stiffness, the slip and the cornering gradient of the landing gear model, including the tire model.

Description: EP commands a steady state cornering of the aircraft with different velocities.
Initial Specification: Throttle lever position shall be constant. Velocities should be stabilized approximately at constant values, i.e. 10, 15, 25 kts.
Input Specification: The EP commands a steady state step steering input on the nose wheel angle provoking steady state cornering. The EP shall apply 10 % of full steering angle command (provoking 2 deg nose wheel angle). The turning radius should approximately result in 40 m as precalculated via the Ackermann formula considering the nose wheel angle. After two proper circles, the EP can proceed with next test.

For proper steering angle command the EP sets up a dual rate on the used remote control. By the dual rate the range and limitation of the input command can be set. The EP shall pay attention to nose wheel angle and throttle lever position. A relevant parameter inter alia within this test maneuver is the achieved circle diameter. The reached stationary diameter quantifies, how much the cornering forces of the landing gear caused by established cornering stiffness angles differ between simulation and reality. Since the cornering forces are the essential forces of the on-ground lateral dynamics according to the landing gear model section, the circle diameters are good measures to prove the landing gear simulation model. Fig. 4.20 displays a conducted steady state cornering maneuver at 15 kts in reality and simulation. In the present test case the aircraft simulation reached a circle diameter of 31 m at a nose wheel steering angle input δ =5 deg. In reality the circles of the aircraft have a diameter of 47 m. The greater diameter in reality corresponds to the smaller measured yaw rate and lateral acceleration in reality.

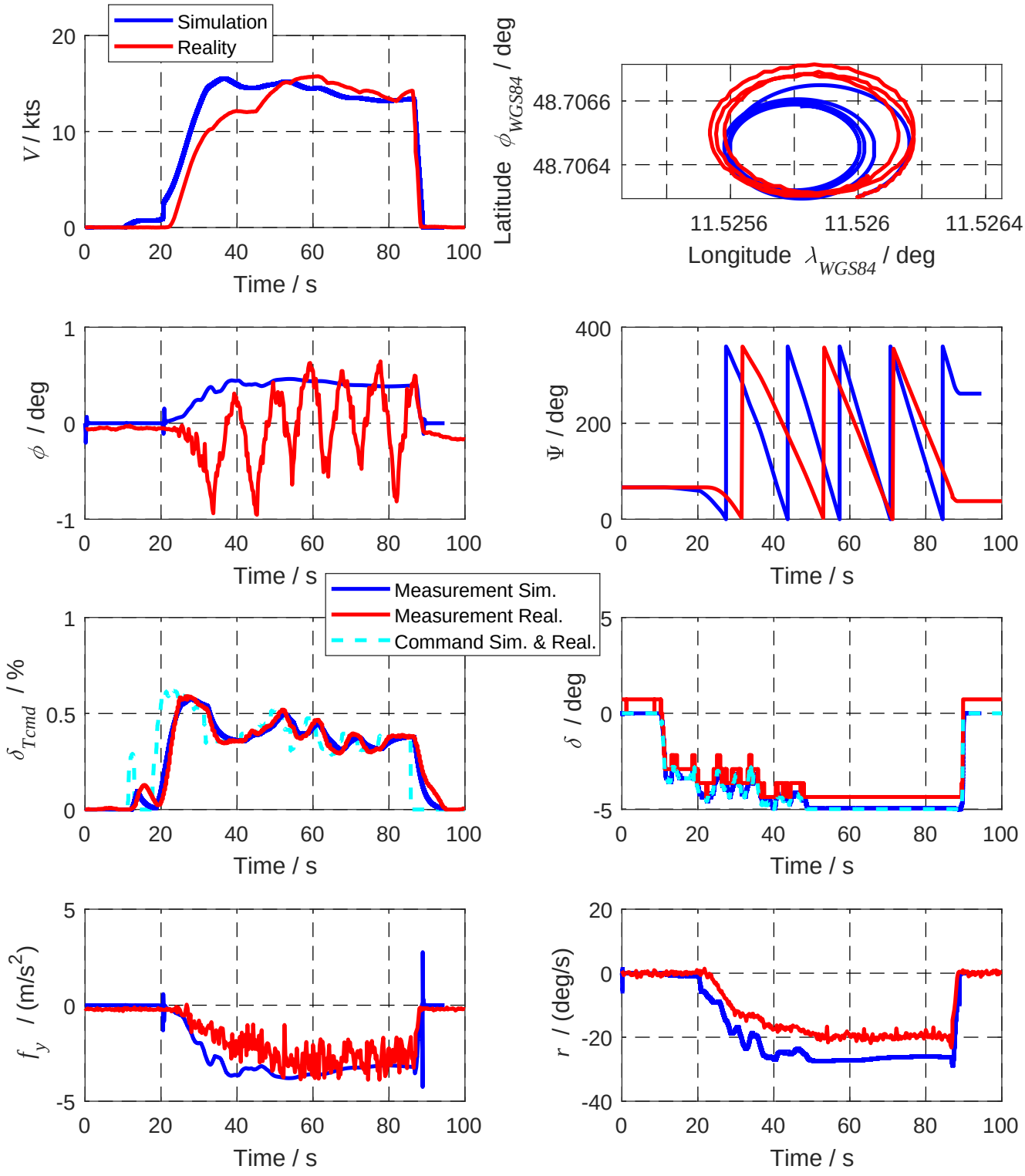


Figure 4.20: EP conducted steady state cornering maneuver at 15 kts

4.4.3.2.4 Wiggly Lines

The wiggly line maneuver is performed to analyze the lateral dynamics of the aircraft and the backlash between the steering actuator and the nose wheel. By the wiggly lines the nose wheel is excited in both direction. Thereby, the backlash asymmetry is investigated. This maneuver complements also handling quality training for the EP to predict the aircraft behavior on-ground.

Description: EP accelerates the aircraft to a predefined velocity, and then steers the aircraft in wiggly lines by nose wheel steering angle inputs.
Initial Specification: The aircraft is positioned on the runway, aligned with the runway centerline. The EP accelerates the aircraft by a constant throttle lever position, if possible. The velocities should be stabilized approximately at the constant values of 10 and 20 kts.
Input Specification: EP shall steer the aircraft straight on runway at approximately constant speed, then EP gives alternating steplike steer inputs to the left and right on the nose wheel. The EP applies a full steering angle command (provoking 20 deg nose wheel angle).

Fig. 4.21 shows relevant recorded sensor signals of a wiggly line maneuver up to 20 kts during the second phase of the EP controlled ground trials. After the EP accelerates to the defined speed, the EP starts to give nose wheel steering angle inputs up to 20 deg. In this test, an additional sensor signal was available, which measured the nose wheel steering angle independently from the actuator deflection (see signal *FTI meas.* in Fig. 4.21). In this way, the backlash could be analyzed by comparing the measured signal and the actuator deflection. As visible in the nose wheel steering angle plot, the backlash is approximately up 2 deg symmetrical in both sides.

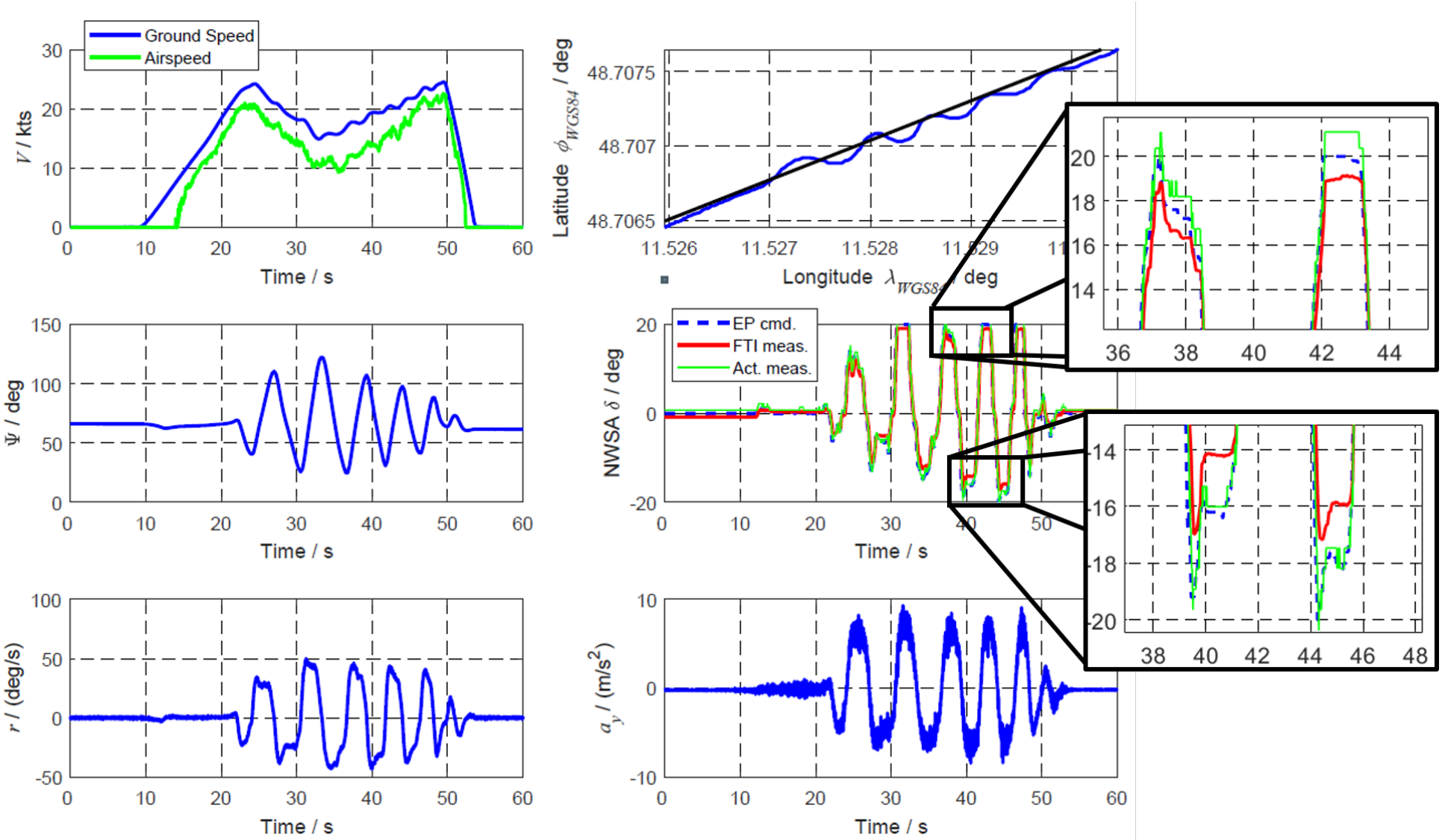


Figure 4.21: EP controlled wiggly lines

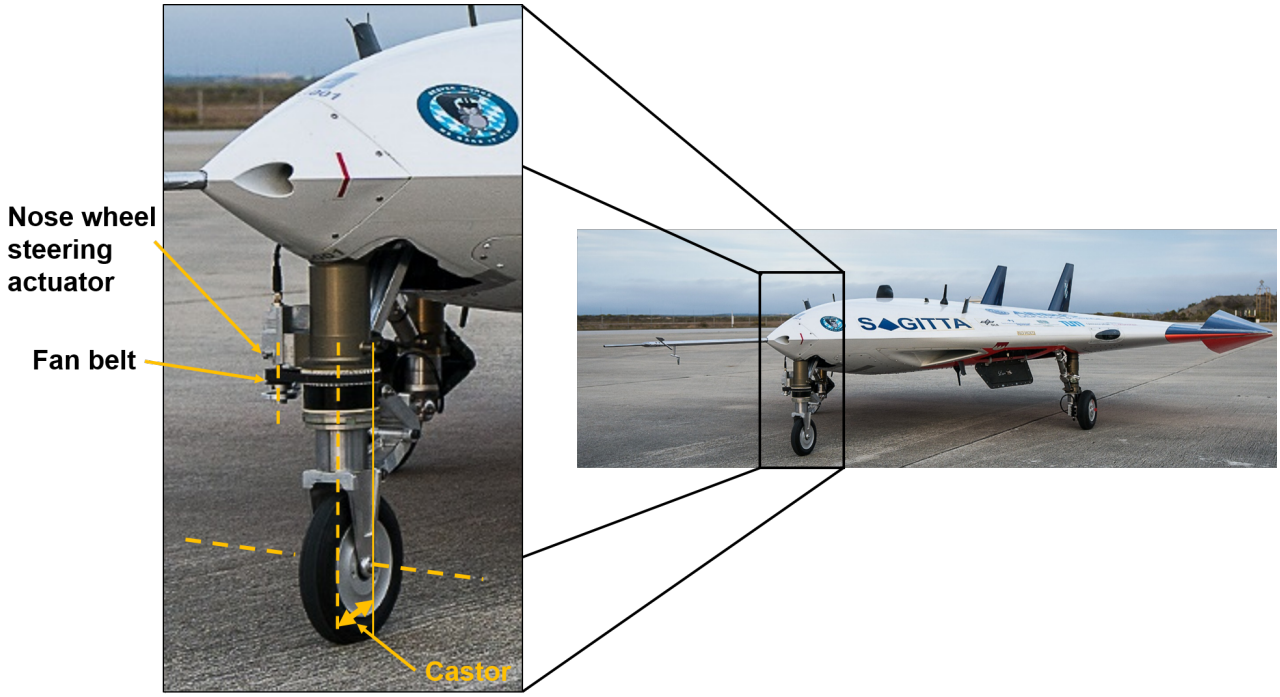


Figure 4.22: Illustration of the SAGITTA nose wheel castor and nose wheel steering actuator

4.4.4 Discoveries and Improvements

As mentioned above, after the first EP controlled taxi tests had been conducted, different system components were modified to improve the vehicle on-ground performance.
The main two findings with regard to the on-ground performance and the design of the ground controller are the brake asymmetry and the backlash between the actuator and nose wheel. These two findings are visible in the exemplary straight acceleration test case explained in Sec. 4.4.3.2.1.
In order to overcome the braking asymmetry, two main modifications are performed. Firstly, the length of the hydraulic brake pipes are adapted to ensure symmetric braking pressure on the left and right main landing gear. Secondly, the mapping between braking command (sent from the flight control computer to the GCEs) and braking pressure is changed. The braking pressure for full brake command is reduced by approximately 25 %. Additionally, the inclination of the mapping from brake command to braking pressure is reduced. That means, the braking pressure for 10 % brake command is increased, what results in an inclination reduction by using a linear interpolation between 10 % braking command and full brake command. In summary, the braking command is mapped now to a smaller range of braking pressure with lower braking pressure at the full brake command.
The backlash between the steering actuator and the nose wheel is identified as second finding for improvement within the first phase of the EP controlled ground trials with regard to the vehicle on-ground performance. In order to reduce the backlash behavior, two main modifications are conducted. First, the castor between the nose wheel axis and the steering

axis is increased to 10 mm. The castor is visualized in Fig. 4.21. The castor improves the directional stability by the restoring force. Secondly, the left main and right landing gear are adjusted to a wheel toe-in of 2 deg in order to achieve a higher directional stability.
The improvements by the modifications are visible by the introduced single lane change and wiggly line maneuver in the last section, which are conducted during the second phase of the EP controlled ground trials. As the plots show, the braking asymmetry and the backlash could be reduced. The modifications are embedded into the landing gear model. The derived results show a promising model behavior in all axes, taking the variate environmental conditions into account. Nevertheless, the ground controller needs to be designed to counteract appropriately these disturbances by dedicated nose wheel steering angle inputs.

4.4.5 Online Gain Set Layout of the Ground Controller[16]

After the successful performed EP controlled taxi tests within the second phase including the explained system modifications and the gained landing gear model confidence, the aircraft is prepared for the testing of the automatic on-ground functionalities. Using the taxi test software the FO is able to give lateral offset commands w.r.t. the centerline and throttle settings to the engines of the aircraft. This enables the evaluation of the ground controller tracking performance at different speeds and the assessment of the ground controller stability characteristics with regard to the backlash of the nose wheel steering and the brake asymmetry. The following chapter shows various test cases, which have been performed during taxi test campaigns. Straight acceleration runs, single and double lane changes, and maneuvers with triggered back-up control law at different speeds are typical test cases to verify, analyze and improve the ground controller performance regarding robustness, stability characteristics and tracking behavior.
First, straight acceleration runs and single lane changes are performed stepwise at 15, 35, and 50 kts with three pre-selected gain sets. After data analysis and control behavior assessment, a gain set could be identified, which yields adequate stability, robustness and control characteristics and was thus chosen as preliminary final gain set. For verification, acceleration runs up to 65 kts, featuring more exciting maneuvers such as double lane changes were performed and additional take-off test runs with abort before reaching the rotation speed are conducted. In the following each test case is explained exemplary by means of one conducted test and substantiated by diagrams shown in Fig. 4.23-4.26. Each plot consists of five subplots. In the first subplot the ground speed, indicated airspeed measured by the navigation system and the air data computer, respectively, are shown. The next three subplots display the main control loop signals of the ground controller. The second subplot shows the outer loop signals of the ground controller and thus, the commanded and the current lateral deviation of the aircraft to the centerline. The third subplot represents the middle loop of the ground controller. The commanded course and the current course of the aircraft are plotted. At low speeds (<1 m/s) the course is faded to the measured heading since at these speeds no reliable course information exists. The black line represents the heading of the runway. The test cases were performed in both runway directions. The signals of the inner loop are shown in the fourth subplot. It can be seen that the measured yaw rate follows adequately the commanded yaw rate. The last subplot shows the commanded and the measured nose wheel steering angle. For measuring the nose wheel steering angle an additional sensor was installed for the taxi tests. By comparison of these two signals, the backlash becomes visible.
Initially, the external pilot commands the aircraft remotely to its take-off position on the run-

[16] This section has been published by the author in Ref. [114].

way. If all ATOL availability conditions are fulfilled, the FO triggers the ATOL take-off mode. Then the aircraft starts to move and the test points are conducted. For the taxi tests the default commanded lateral deviation to the centerline was 3 m.

4.4.5.1 Single Lane Change

Fig. 4.23 displays the lane change maneuver, which has been performed with different gain sets at dedicated ground speeds. The present plot shows the test case at 40 kts with the initial pre-selected gain set. The FO accelerates the aircraft by the recycled speed control and triggers a 5 m offset via the *Ctrl Switch 2* after the briefed speed is reached. When the offset is compensated by the ground controller, the FO commands the aircraft back to its original lateral offset. The ground controller compensates the commanded offset by nose wheel steering inputs but reacts with overshoots regarding the lateral offset. The reason is visible in the third subplot of Fig. 4.23. After applying the lane change command, the course signal exhibits light oscillations, which result in the lateral offset overshoot. For that reason, the gain of the middle loop is adapted to decrease the overshoot of the lateral deviation and to reduce the light oscillations of the course signal.

4.4.5.2 Double Lane Change

The second test case depicted in Fig. 4.24 is an automatically performed double lane change with the dedicated final gain set for first flight. In this test the FO commands first a lateral offset to the right, and after a few seconds a larger lateral offset to the left. This testcase shows the ground controller behavior when its dynamics are excited and hence provides robustness confidence against abrupt changing environmental conditions, e.g. turning wind directions, gusts and turbulences. After the aircraft started to roll, the aircraft accelerates and captures the commanded lateral deviation. When the first lateral offset to the right is commanded by the FO the aircraft changes its course slightly by a low nose wheel steering input. At about 40 s the FO commands the aircraft to the other side of the centerline with a lateral deviation of 5 m by *Ctrl Switch 3*. Due to the higher control error between the commanded lateral deviation and the current lateral deviation the controller gives a high nose wheel steering command. As desired, the aircraft changes its direction to track the commanded trajectory. After initiating the brake command, the aircraft stops on the commanded lateral deviation.

4.4.5.3 Test of back-up law control structure

In the next test case shown in Fig. 4.25 the back-up control law is tested. Thereby, the back-up law is triggered externally by the FO according to Table 4.1 after a heading change is

performed. First the aircraft accelerates straight and the FO triggers an offset command with the nominal control structure. After the heading change of the aircraft is performed due to the offset command, the FO triggers the back-up control law. Then only the heading and the yaw rate is controlled. This becomes clear in the third subplot in Fig. 4.25. The heading deviation of the aircraft to the runway direction is minimized. The installation error of the navigation system becomes visible here. Since the navigation system is not installed properly with regard to the yaw axes a difference between the course and the heading occurs. For that reason, the aircraft doesn't move parallel to the centerline but moves with a course deviation of about 1 deg to the runway direction.

4.4.5.4 Straight Acceleration Run

The last test point represents a straight acceleration run up to 66 kts ground speed, which is equivalent to a take-off run during the first flight mission. The plot is displayed in Fig. 4.26. In this test point the initial aircraft position is exactly 3 m beside the centerline but the heading deviation to the runway is approximately 10 deg. After the FO gives the full throttle command, the aircraft speed increases while the ground controller corrects the aircraft into the runway direction by a nose wheel steering command. During the following acceleration phase the ground controller keeps the aircraft in the direction of the runway with a control error less than 0.5 deg. Further the aircraft stays on the commanded lateral offset during the straight run with an error less than 1 m. After the FO brakes by *Ctrl Switch 1* the ground controller compensates the yaw rate caused by the brake asymmetry by small nose wheel steering inputs.

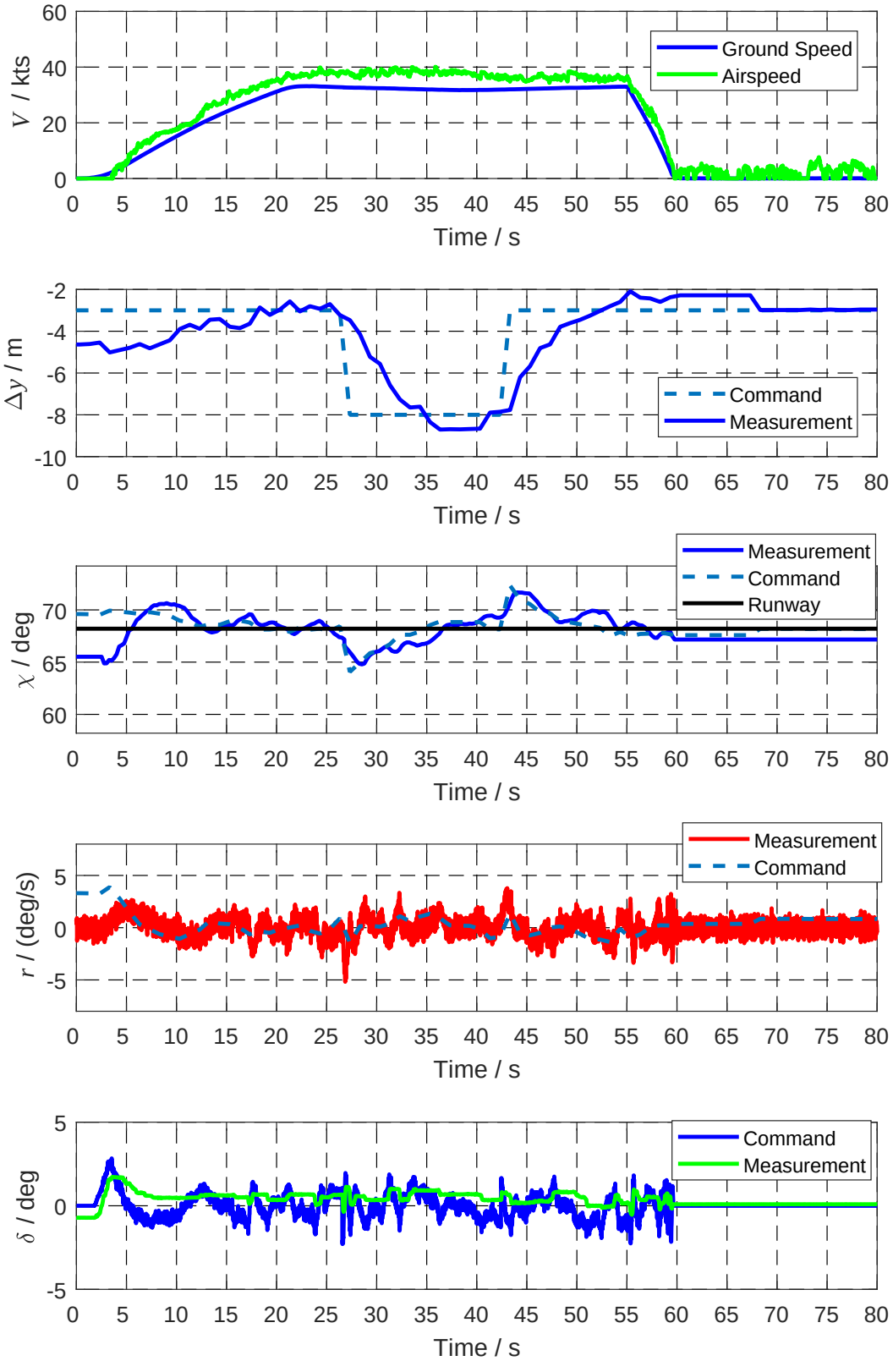


Figure 4.23: Offset command test case with gain set number 1 (initial) at 35 kts

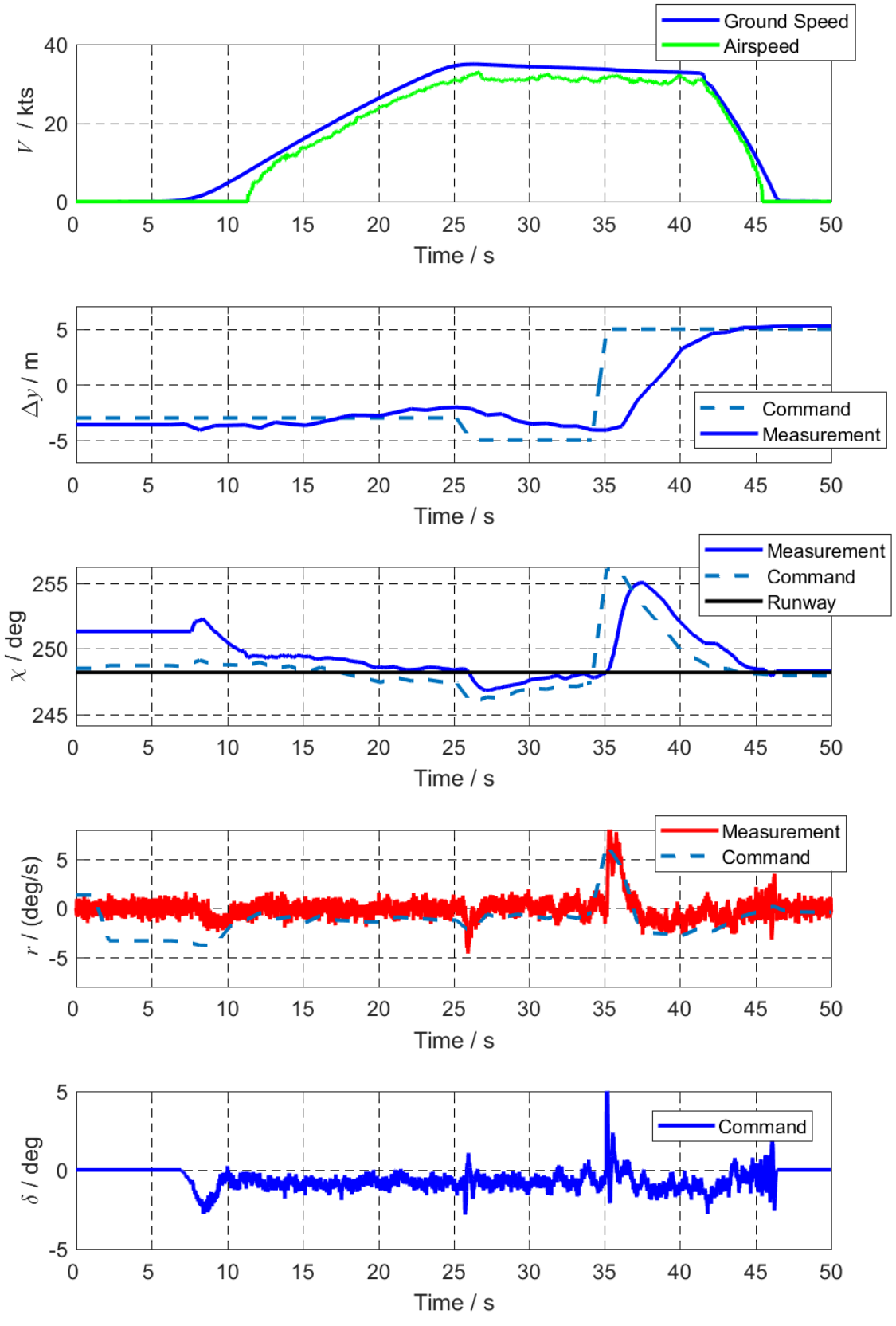


Figure 4.24: Double lane change test case with gain set number 2 (final gain set) at 35 kts

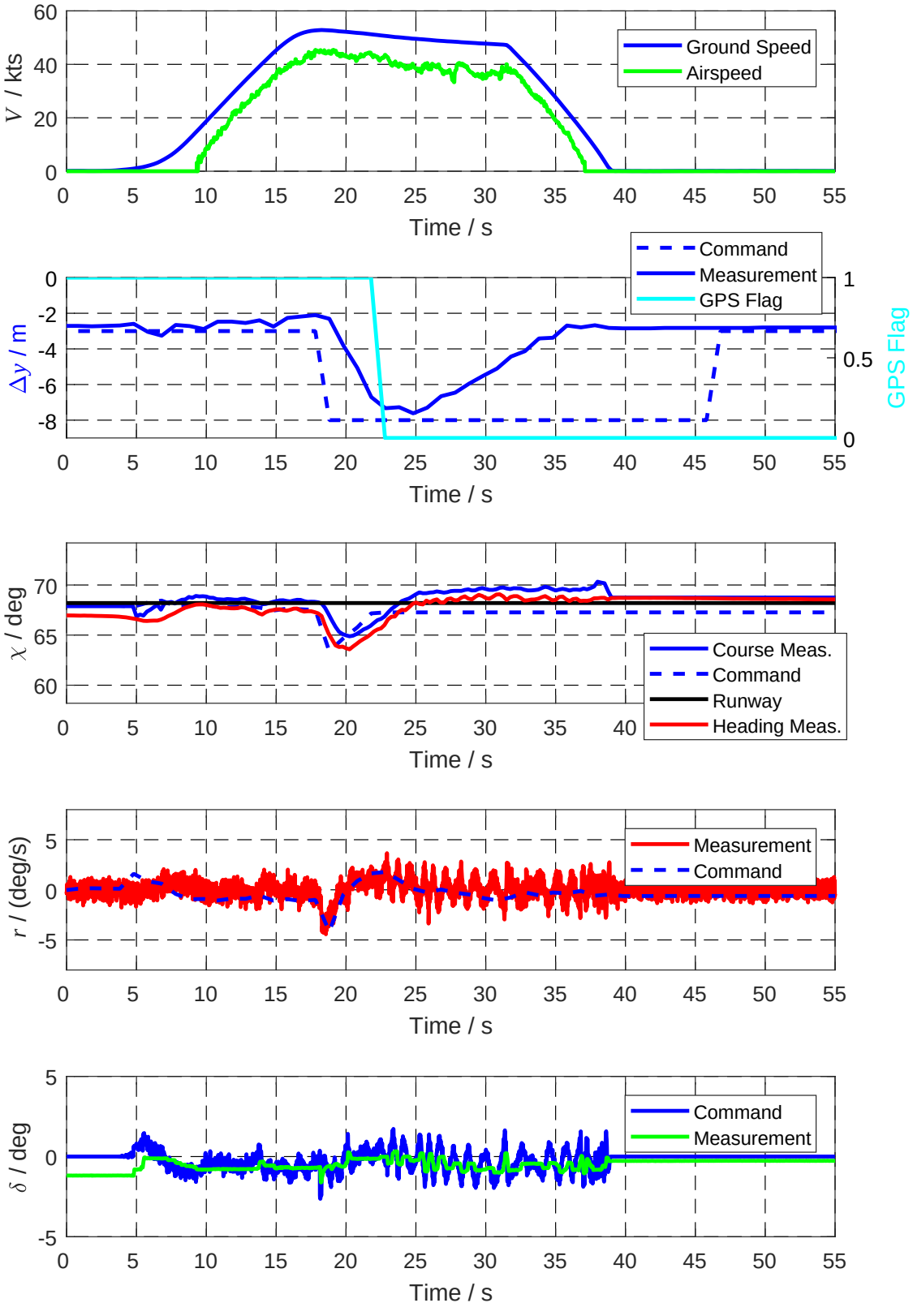


Figure 4.25: Single lane change test case with *No GPS* trigger using gain set number 2 (final) at 50 kts

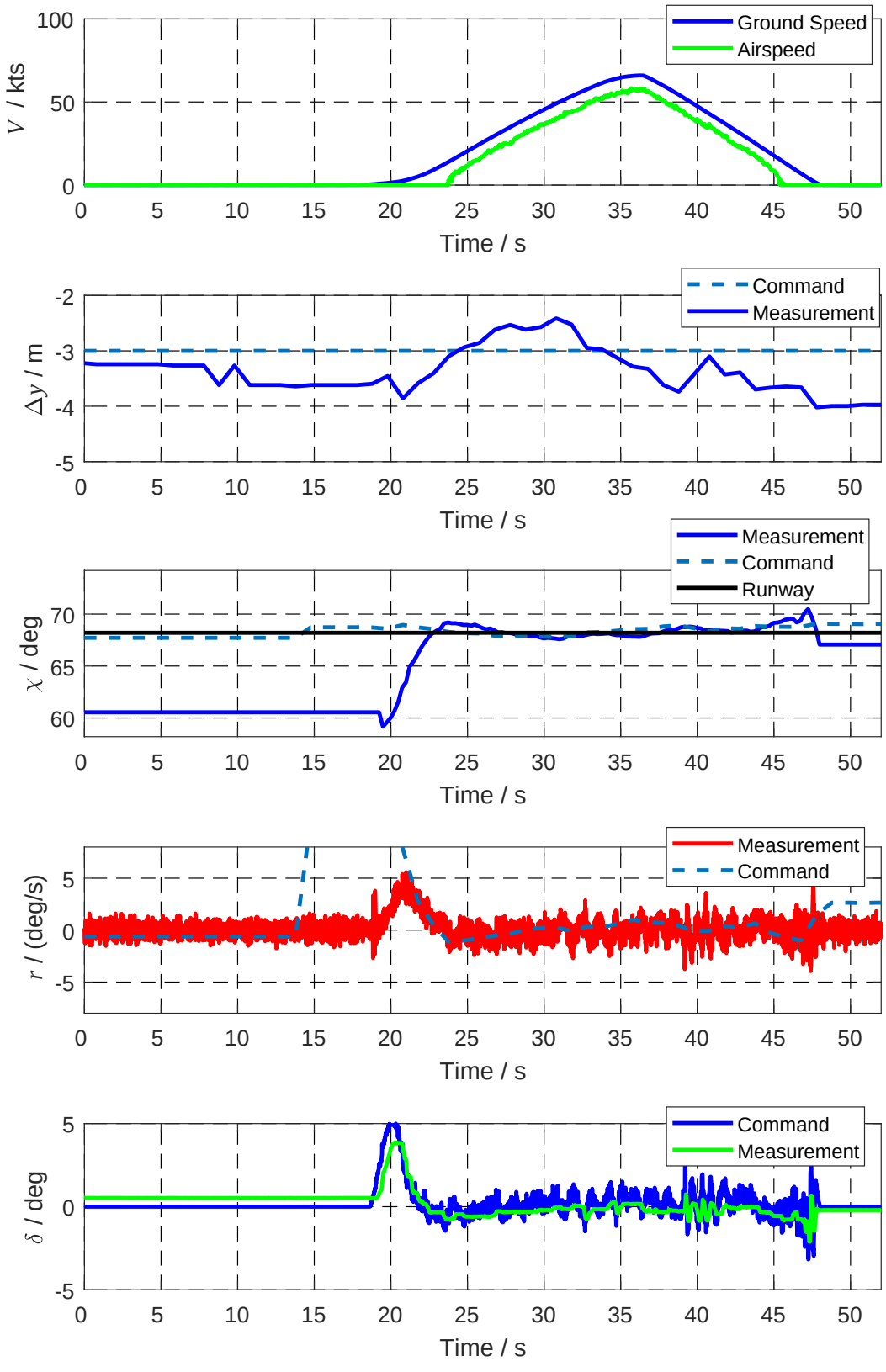


Figure 4.26: High speed taxi test with final gain set at 66 kts

4.4.6 Verification and Validation for First Flight

Final taxi test runs are performed with the first flight configuration of the overall system (including final landing gear and engine configuration) using the presented taxi test software and aircraft-in-the-loop simulations of the first flight mission with running engines are executed, in order to reach the final step for first flight qualification.

4.4.7 Final Taxi Tests at Ground Trial Facility[17]

Within the final taxi tests, the ground controller is qualified for first flight maturity. Thereby, a take-abort is tested by accelerating the aircraft up to 0.8 V_S stall speed and the ground controller is excited by FO offset commands at higher speeds. By succeeding these tests, the ground controller of the flight control system is set for first flight.

4.4.7.1 Double Lane Change

The first test case depicted in Fig. 4.27 is an automatically performed double lane change with the dedicated gain set for first flight at V_K^R=55 kts. In this test the FO commands at first a lateral offset to the right, and after a few seconds a larger lateral offset to the left. This test case shows the GC behavior when its dynamics are excited strongly and hence provides robustness confidence against abrupt changing environmental conditions, e.g. veering wind directions, gusts, and turbulences. After the aircraft starts to roll, the aircraft accelerates and captures the commanded lateral deviation. When the first lateral offset to the right is commanded by the FO, the aircraft changes its course slightly by small nose wheel steering deflections. At time about t=22 s the FO commands the aircraft to the other side of the centerline with a lateral deviation of Δy=-5 m. Due to the high deviation between the commanded lateral deviation and the current lateral deviation, the controller gives a relatively large nose wheel steering command. As desired, the aircraft changes its direction to track the commanded trajectory. After initiation of the brake command, the aircraft stops on the commanded lateral deviation.

4.4.7.2 Straight Acceleration Run

The second test point represents a straight acceleration run until ground speed V_K^R=64 kts, which is similar to a take-off run with abort during the first flight mission. The recorded data are plotted in Fig. 4.28. Within this test maneuver the aircraft was positioned 6 m beside the runway centerline and the heading deviation to the runway was approximately 5 deg. After the FO applies the full throttle command, the aircraft starts to roll while the GC corrects to

[17] This section has been published by the author in Ref. [115].

the commanded lateral offset and aligns the aircraft with the runway direction by nose wheel steering commands. During the following acceleration phase the GC keeps the aircraft in the direction of the runway with a control error of less than 0.5 deg. Further, the aircraft reaches and stays on the commanded lateral offset during the straight run with an error less than 1 m. After the FO initiates the brakes, the GC compensates the yaw rate caused by the brake asymmetry by small nose wheel steering deflections.

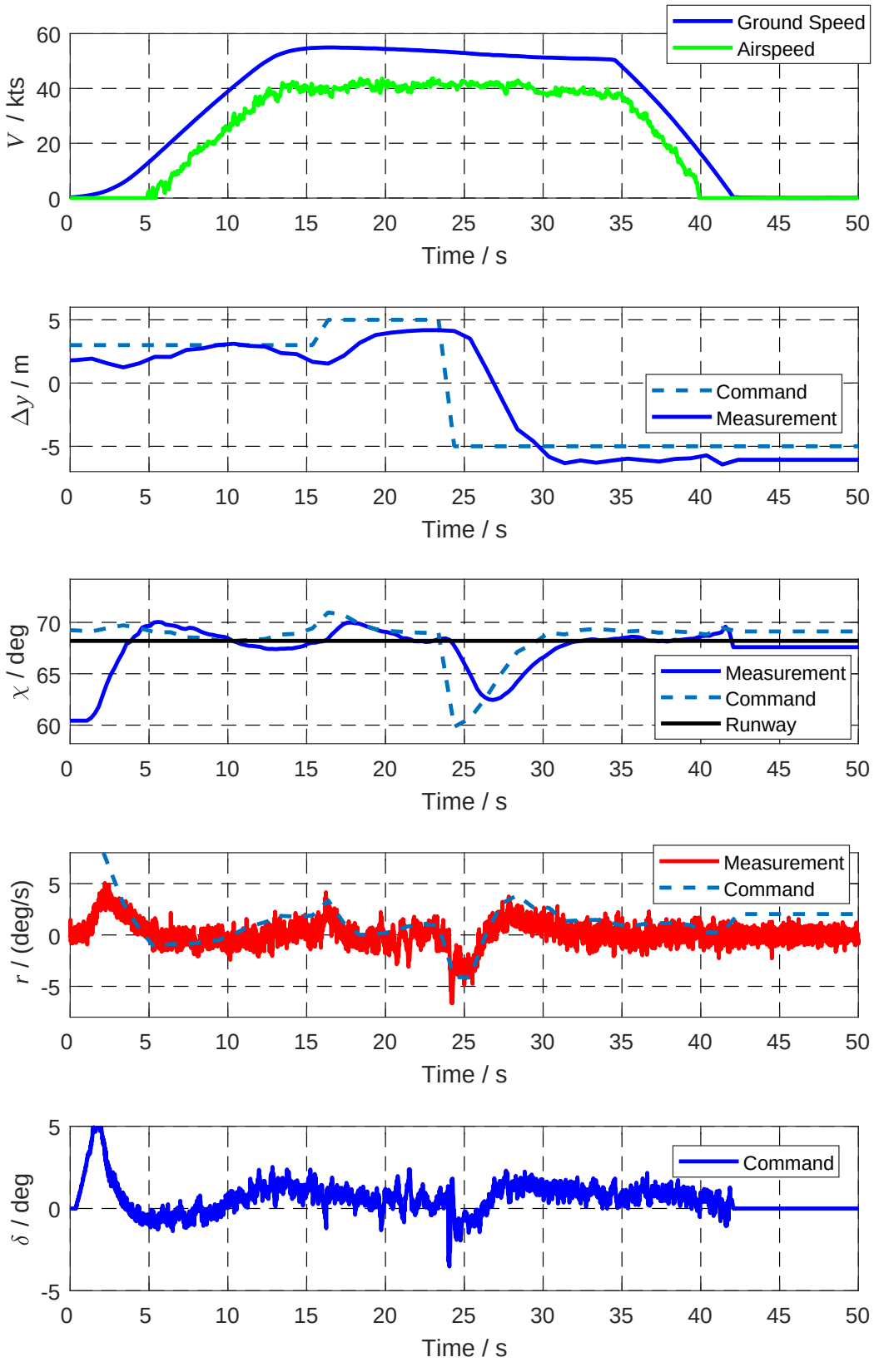


Figure 4.27: Double lane change maneuver test case with greater excitation

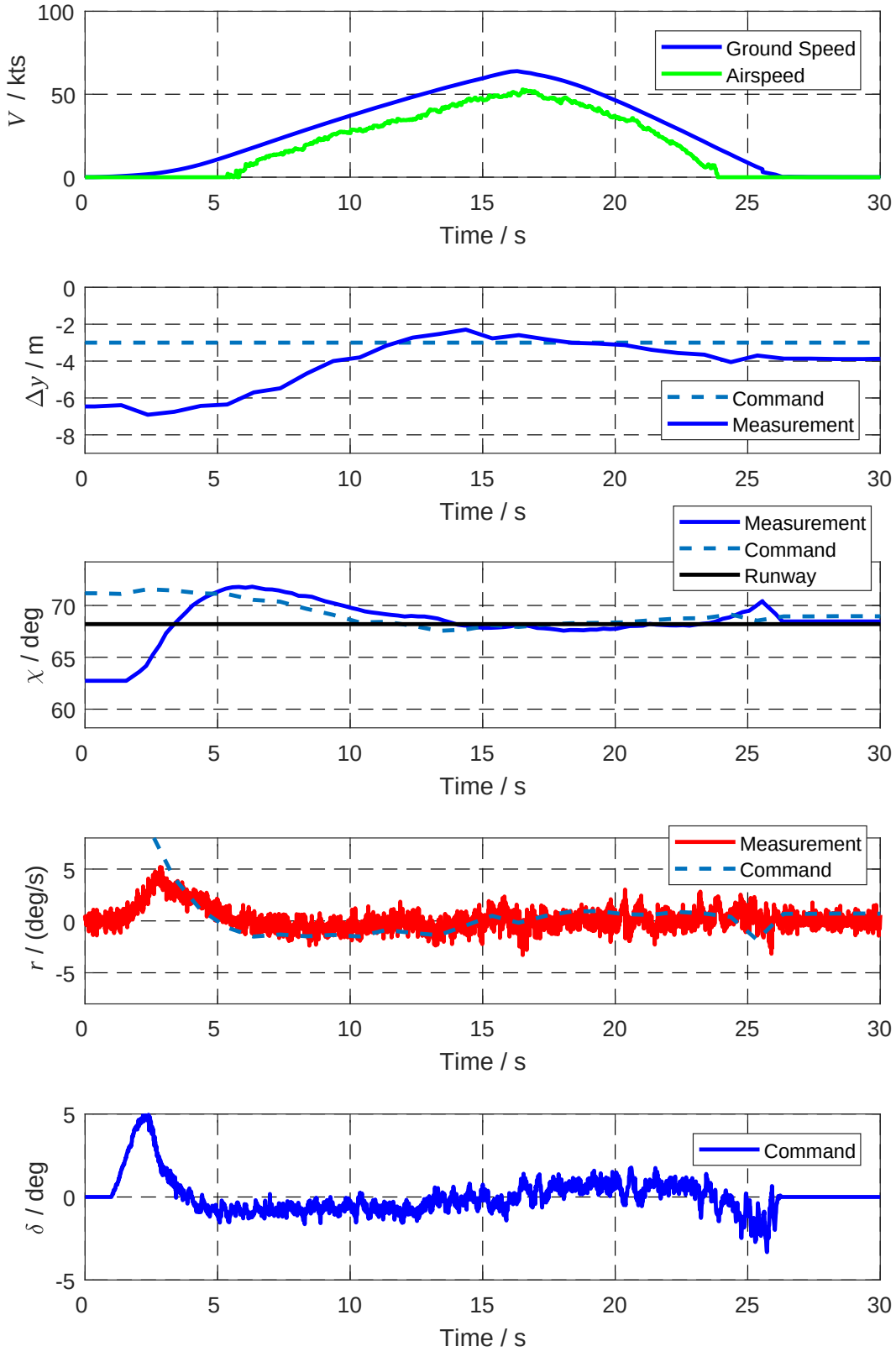


Figure 4.28: High speed straight acceleration run for first flight qualification

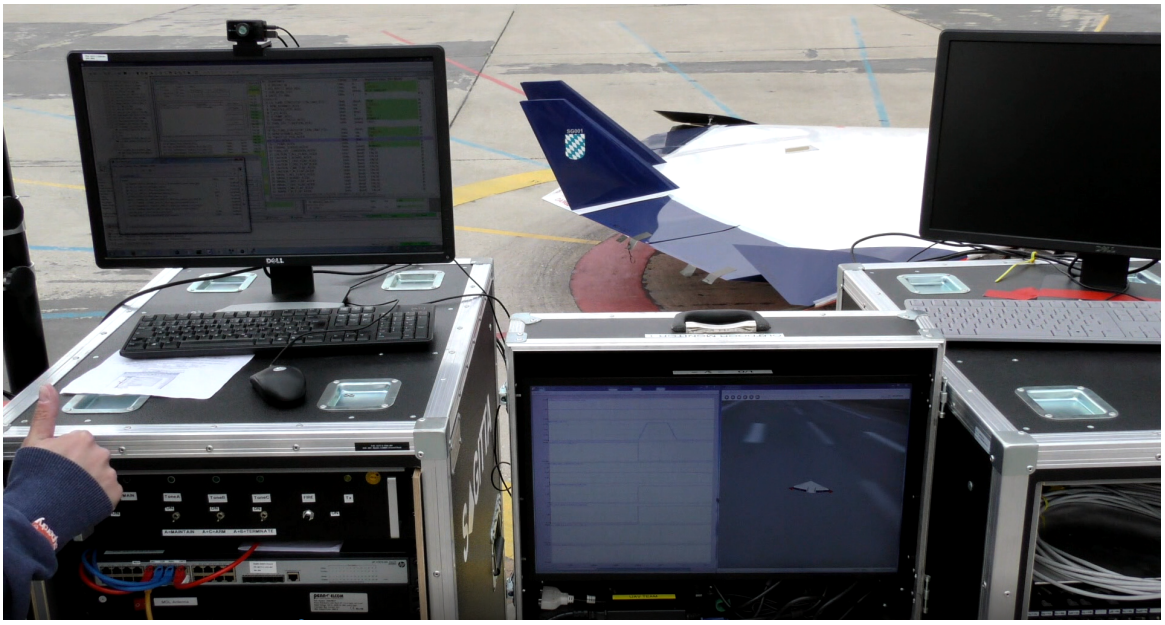


Figure 4.29: The SAGITTA demonstrator during aircraft-in-the-loop simulations

4.4.7.3 Full Flight Mission within Aircraft-in-the-Loop Simulation

Apart from the above explained towing and taxi tests, hardware- and aircraft-in-the-loop simulations of the flights planned in the first flight campaign are conducted with the final aircraft configuration and flight control software. In Fig. 4.29 the SAGITTA demonstrator within an aircraft-in-the-loop simulation is illustrated. During the aircraft-in-the-loop simulation, the aircraft is in standstill on-ground, and the aircraft dynamics are simulated by a real-time computer. In contrast to hardware-in-the-loop simulations, the most system components integrated in the aircraft are activated and are functioning within the various command and control loops. In addition to the flight control computer, the flight management control station, the data links, the control surface actuators and the engines are in operation and running. The simulated sensors signals are fed in the flight control computer. With the aircraft-in-the-loop simulations the system can be tested as close as possible to real flight. The system components and their loads (e.g. for the engines) can be tested within the closed-loop system on-ground. The aircraft-in-the-loop simulation enables the testing of the closed loop system in simulated flight. Thereby, the system components (except the sensor systems) are verified in conditions, which cannot be tested within taxi tests and will certainly occur in real flight. With respect to the ground controller, the aircraft-in-the-loop simulation enables the testing of the ground controller in the rotation and derotation phase, which could not be verified in ground trials, within the closed-loop hardware environment.

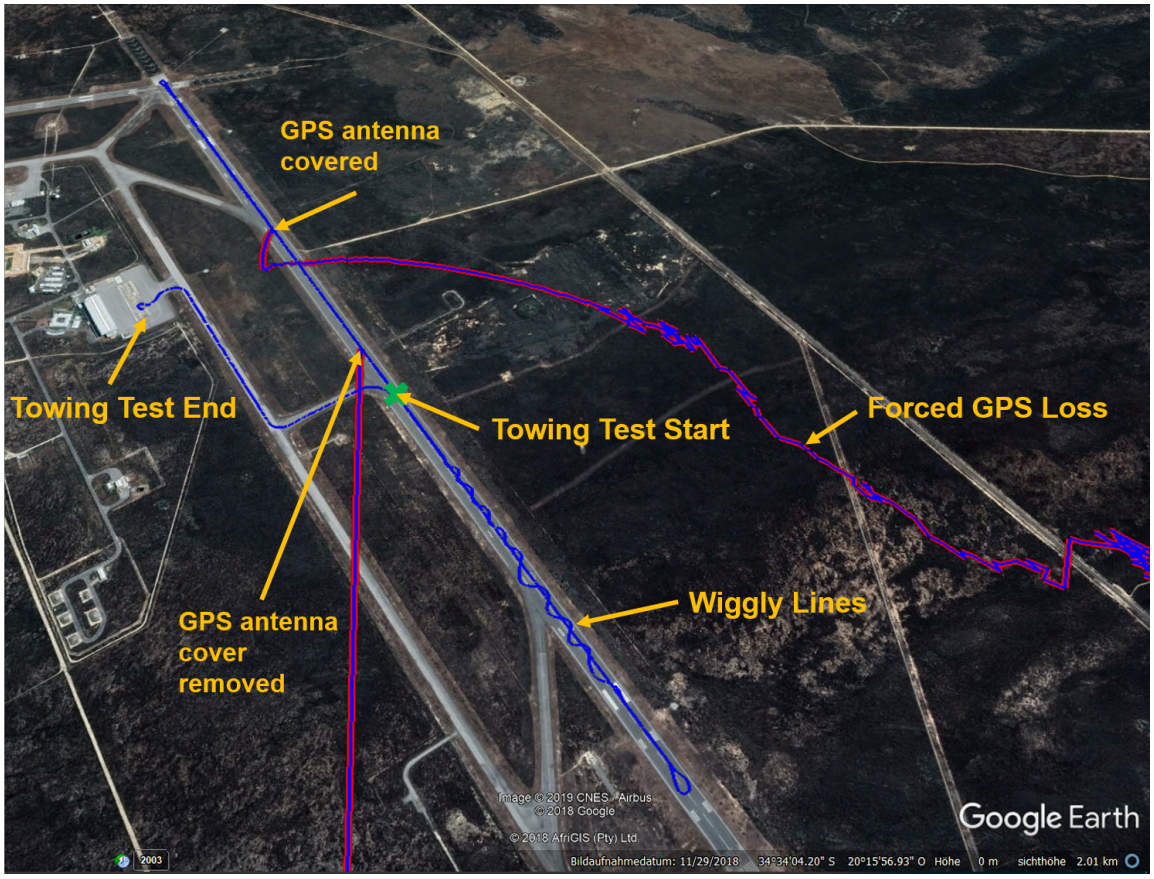


Figure 4.30: Track of the SAGITTA demonstrator during towing tests at the first flight aerodrome (created with Ref. [145])

4.5 In-Flight Validation

4.5.1 Pre-Flight Verification on-site

The first flight campaign is conducted at the Overberg Test Range in South Africa. Therefore, the whole equipment is packed and transported to the test facility. The whole ground infrastructure is re-installed and re-established at the test range. The system undergoes breakdown testing from each single system component to the overall system.
The flight control system is finally verified for first flight, analogously to the ground trials, within towing and taxi tests. No taxi test software is used at the test facility in Overberg in contrast to above presented taxi tests. The introduced pre-flight verifications are conducted using the flight control software developed for flight. Fig. 4.30 illustrates the measured ground track of the conducted towing test at the test facility. As indicated, the towing test consists of the same test points, as explained in Sec. 4.4.1.4, for sensor verification. The towing test includes inter alia straight runs, wiggly lines and forced GPS losses. The last test, in which the aircraft moves by its own power before first flight, is an automatic take-off run until 40 kts speed. The time plots of the conducted test are visualized in Fig. 4.31. With

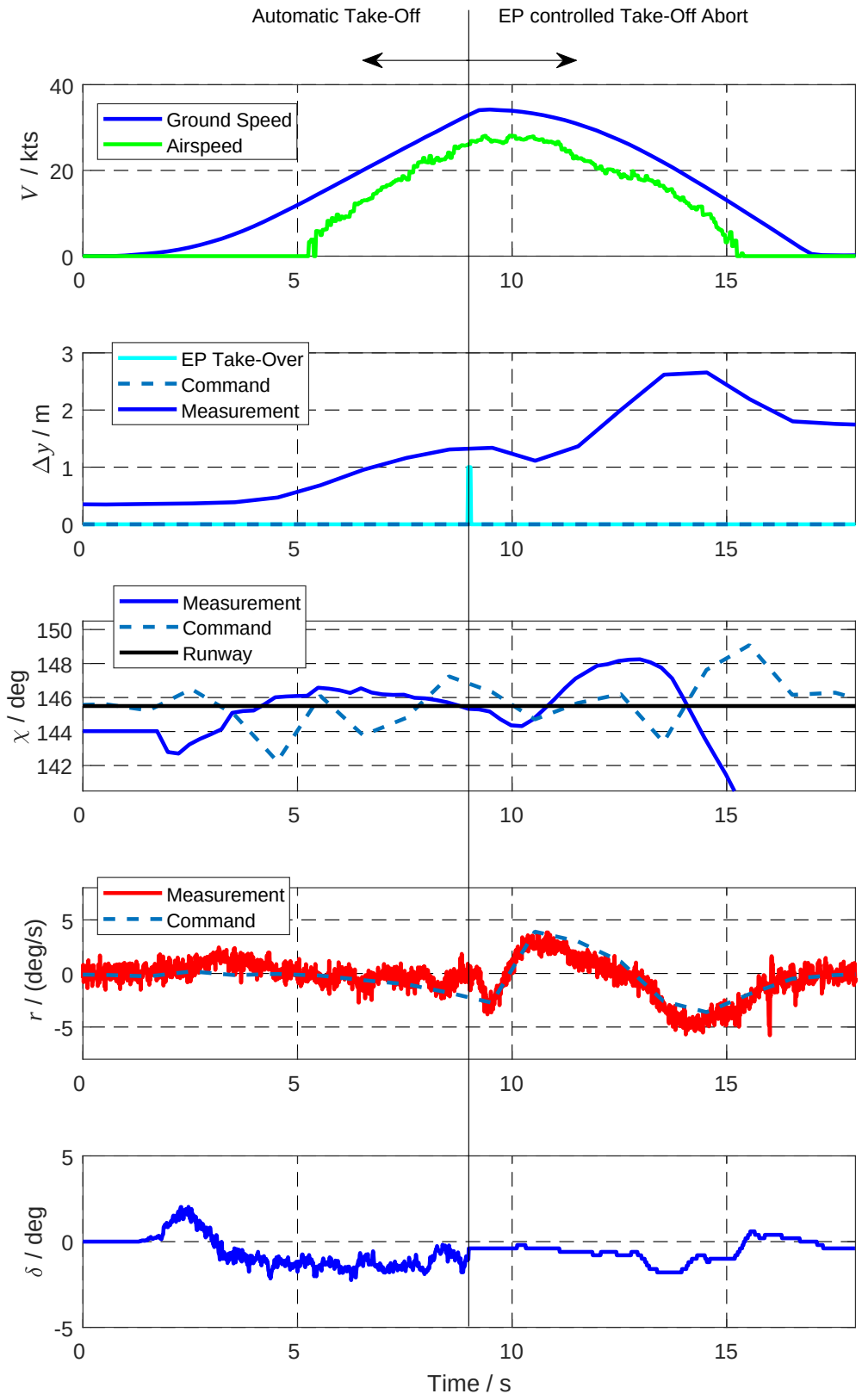


Figure 4.31: Automatic take-off with EP controlled take-off abort on test-site

the flight software, the ground controller cannot be excited anymore by lateral offset commands. The flight operator can only command automatic take-off, which is equivalent to a take-off run in first flight (see Sec. 4.3.4). The acceleration run could either be aborted by the flight operator by commanding *Take-Off Abort* or the external pilot takes command and decelerates the aircraft manually. Since the flight operator mode *Take-Off Abort* would cause an engine cut-off automatically, the external pilot stops the aircraft in this test by commanding idle thrust and full brake. This approach saves the engines abnormal conditions and represents an additional training for the external pilots, who should take-over in emergency situations. For the final taxi test, the flight operator commands automatic take-off after the aircraft is controlled by the external pilot to the defined take-off position. Then, equivalent to first flight, full thrust command is given to the engines and the brakes are released. The aircraft accelerates and the ground controller tracks the centerline (see e.g. lateral deviation Δy in Fig. 4.31). Within this test, before the ground speed 40 kts is reached, the external pilot takes control by switching the take-over toggle on the remote control. The external pilot tries to keep the aircraft close to the centerline, reduces the thrust to idle and brakes. After the external pilot takes command, the lateral deviation increases and the course error to the runway direction grows, as the yaw rate measurements underline. However, the external pilot is able to abort the aircraft manually. After this test has been successfully conducted, the aircraft and the overall system have been verified for first flight.
Beside the towing and taxi tests, engines runs and aircraft-in-the-loop simulations have been conducted, equivalent to Sec. 4.4.7.3.

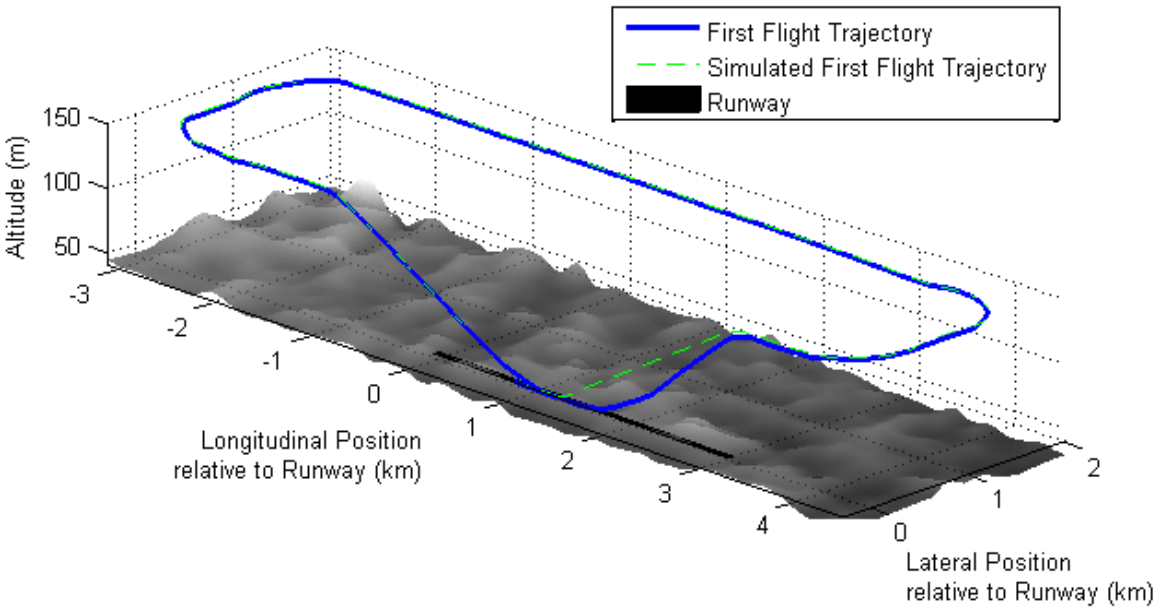


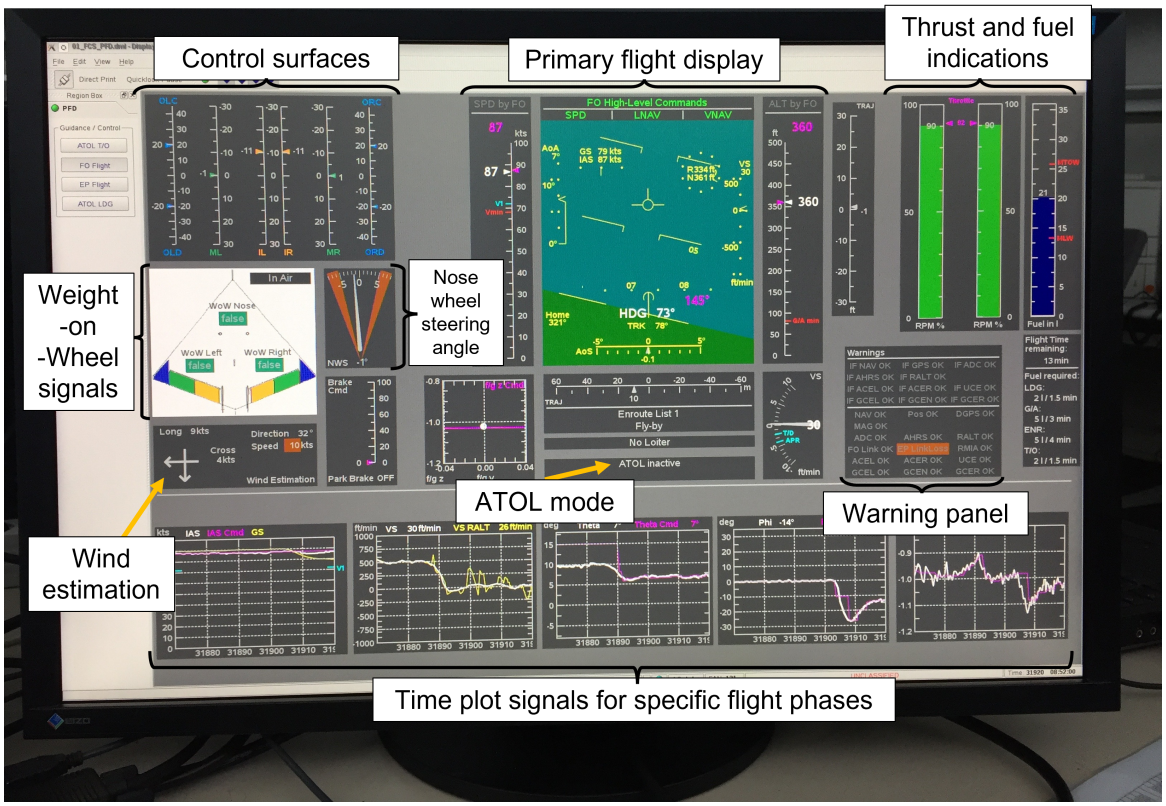
Figure 4.32: Simulated and real-live first flight mission of the SAGITTA demonstrator (Ref. [54]

4.5.2 First Flight Mission

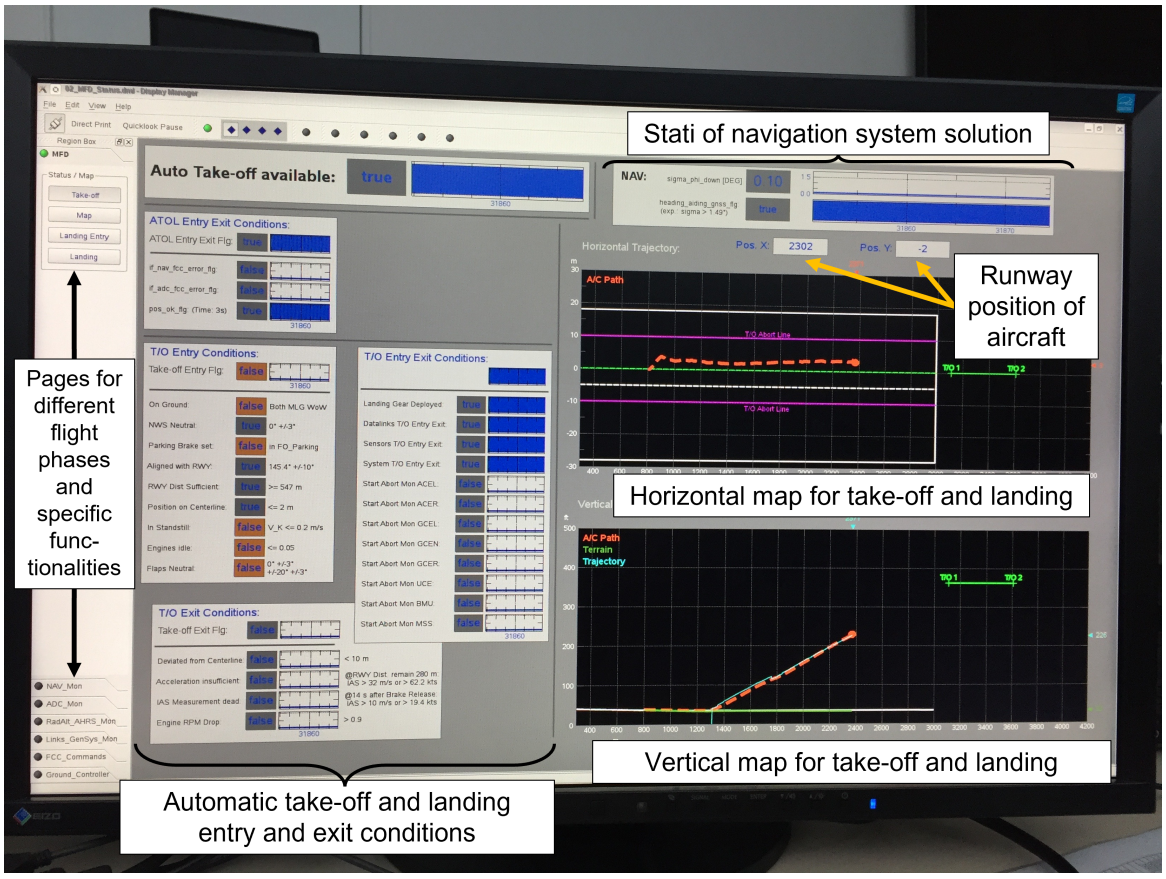
The nominal first flight mission of the SAGITTA demonstrator consists of an automatic take-off maneuver, a trajectory-controlled flight, which is specified by pre-defined waypoints, and an automatic landing maneuver. As introduced in Sec. 4.3.4, the ground controller is enabled during the automatic take-off and landing maneuver. The pre-defined waypoints are arranged in a rectangular order. As a result, the first flight mission has the shape of a so-called *racetrack pattern*. The two longer legs have a distance of 7.5 km and the two base legs cover a distance of 2 km. The take-off and landing is performed on the southern longer leg and in the direction of the runway to the south-east. The first flight mission is illustrated in Fig. 4.32.

4.5.3 First Flight Monitoring

In order to enable the monitoring of the flight control system during the first flight in the ground station by the system engineers, various displays are designed for the Quick Look Terminal. The concept of the displays should allow an immediate identification of possible failures within the complex flight control system as well as finding solutions for emerging issues in an appropriate and fast way. The whole concept is introduced in Ref. [146]. Two pages of the displays are shown in Fig. 4.33. The displays are implemented using an internal software of an industrial partner within the project.



(a) Primary flight display during trajectory-controlled flight



(b) Mission flight display during automatic take-off

Figure 4.33: Quick look terminals for monitoring of flight control system

4.5.4 Pre-Flight Checks For First Flight

Equivalent to the conducted taxi tests, pre-flight start-up procedures and check lists are complied to enable a smooth and professional preparation and operation of the first flight. In terms of the flight control system, the most important pre-flight checks are outlined in the following.

- GPS Satellite Constellation Prediction:
  The quality of the GPS signal is predicted for the time at the first flight . Therefore, it is checked, if the vertical and horizontal *Dilution of Precision* parameter is below a defined threshold, and if more than six satellites are available at the time of the first flight(see Ref. [54] for details).
- Quick Look Terminal Check:
  The Quick Look Terminals in the ground station are started up. It is checked, if the displays function.
- Plausibility checks for main signals of each system component:
  According to the start-up procedure, the system components are switched on step-by-step. After each activation, it is checked, if the main signals of each component are received and if their values are within a plausible range.
- Wind Conditions:
  The current wind conditions are read out from a weather panel and checked, if they are within the pre-defined limits for first flight.
- Air Data Computer:
  The current *QNH* is read from the weather panel and set for the air data computer.
- Vane Check:
  The vanes to measure the angle of attack and the angle of sideslip are moved by the engineer near the aircraft during the start-up. The signals are checked in the ground station by the system engineers
- Control Surface Check:
  The control surfaces are moved by the external pilots giving specific input commands. The system engineers check the displayed control surface signal behavior in the ground station.
- Weight-on Wheel Check:
  The signals of the weight-on-wheel sensors are checked, if they are functioning. Therefore, the aircraft is lifted up by the system engineer near the aircraft for each wheel.
- Engine Start Monitoring:
  The engine start-up is monitored and the engines values are checked (i.e. exhaust gas temperature, engine status, rotational speed).
- Navigation System Alignment:
  The alignment of the navigation system is conducted by the flight operator. After the

alignment, the signals of the navigation system are checked for plausibility (i.e. position, velocity, attitude).

- Brake Check:
  After the engines are running and the aircraft is ready for taxi, the external pilot accelerates the aircraft to 15 kts and brakes the aircraft.
- Fuel Check:
  While the aircraft moves to the take-off position, the fuel level is checked.
- Air Data Computer and Navigation System Check:
  While the aircraft is moving, the navigation system solution and the airspeed are checked for plausibility. It is monitored, if the magnetometer aiding is working properly.

The aircraft is controlled by the external pilot to the take-off position. Then, the flight operator takes over and commands *FO Parking*. The wind conditions are checked again. Finally, the aircraft is set for the first flight.

(a) Fully automatic final descent, shortly before touch-down

(b) Automatic braking and centerline tracking, shortly after touch-down

Figure 4.34: The SAGITTA demonstrator at the end of its first flight mission [Airbus D&S]

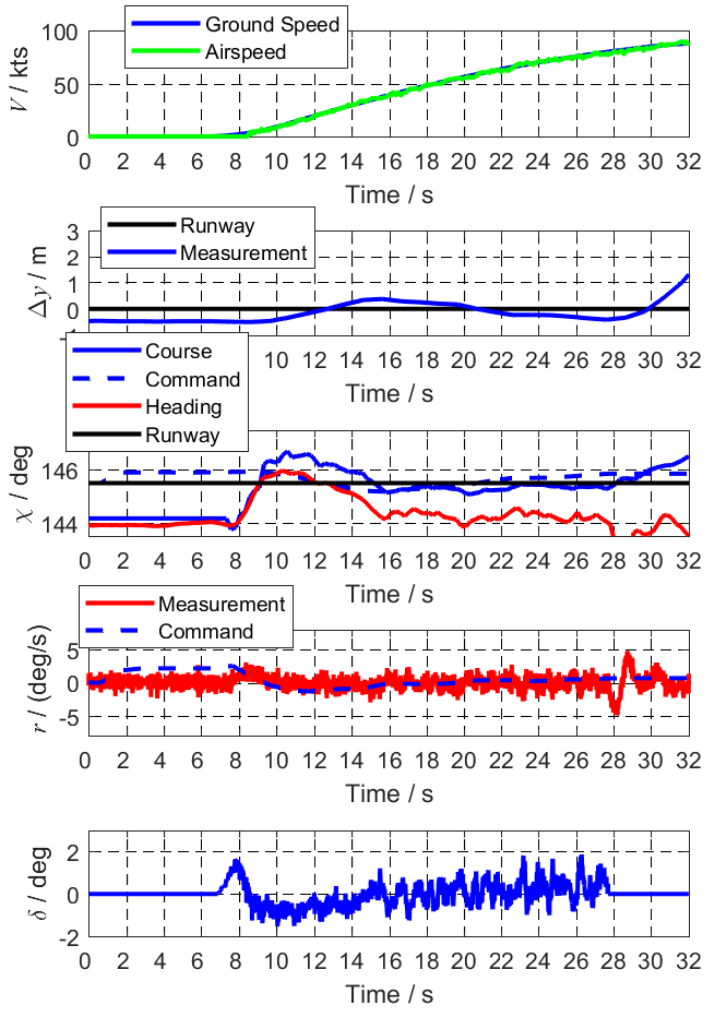
4.5.5 The Maiden Flight of the SAGITTA Demonstrator[18]

The first flight of the SAGITTA demonstrator was performed on *5th July 2017* at the Overberg Testrange in South Africa. After the pre-flight checks were completed, the flight operator triggered the automatic take-off at 16:15 local time. When the take-off maneuver was completed and the aircraft reached the altitude 100 m, the flight operator triggered the high-level command and activated the pre-programmed en-route waypoint list. The aircraft followed the pre-defined route, as illustrated in Fig. 4.32. When the aircraft entered the defined landing corridor, the flight operator activated automatic landing. The aircraft touched down and braked automatically until standstill at 16:22 local time. After a short inspection by the ground crew, the aircraft was controlled back to the apron by the external pilot.

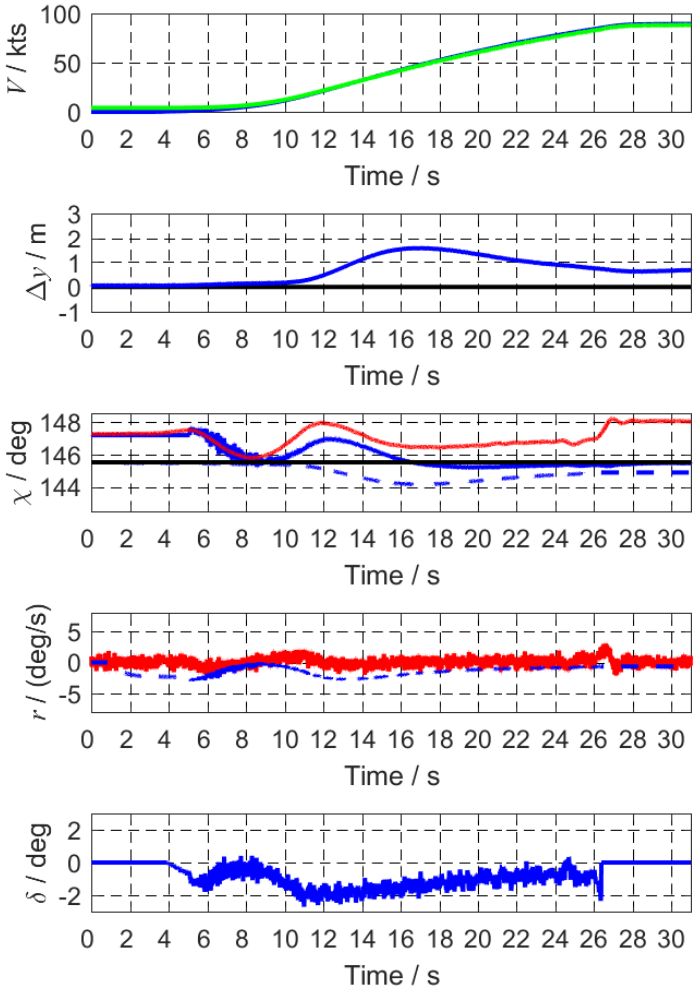
This section comprises the analysis and the verification of the described ground controller functionality in the course of the take-off and landing maneuver of the SAGITTA maiden flight and has been already published by the author in Ref. [116]. The recorded flight test data of the take-off and landing maneuver are processed to focus on the ground controller analysis by generating various specific diagrams and are compared with simulation results of analogous take-off and landing maneuver.
On the left hand side of Fig. 4.35 to 4.40 the flight test data of the take-off and landing maneuver are plotted, on the right hand side the data of the maneuvers from model-in-the-loop simulation.
Fig. 4.35(a), 4.35(b) and 4.38(a), 4.38(b) highlight the lateral motion of the aircraft. Each figure consists of five subplots. In the first subplot, the ground speed and the airspeed indicated by the ADS are shown. The next three subplots display the three control loop signals

[18] This section has been published by the author in Ref. [116].

of the GC. The second subplot shows the outer loop signals of the GC and thus the commanded and the current lateral deviation of the aircraft with regard to the centerline. The third subplot represents the middle loop of the GC. The commanded course and the current course of the aircraft are plotted. At low ground speeds (<1 m/s) the course is replaced by the heading signal since at these speeds no reliable course information exists. The black line represents the heading of the runway. The signals of the inner loop in terms of the measured and commanded yaw rate are shown in the fourth subplot. The last subplot shows the commanded nose wheel steering angle.
Further details of the maneuvers are shown in Fig. 4.36(a), 4.36(b) and 4.39(a),4.39(b). Each first subplot represents the deflection of the right and left split flaps. In order to understand, which mode of the automatic take-off and landing module is active, the second subplot shows the ATOL sub-mode. The weight-on-wheel sensor indications of the main-wheels and the nose wheel are plotted in the third subplot to detect when the aircraft is airborne. This is underlined by the load factor measurements of the navigation system plotted in the last subplot.
Fig. 4.37(a), 4.37(b) and Fig. 4.40(a), 4.40(b) show the aircraft signals of the longitudinal motion during the take-off and landing maneuver in reality and simulation. The first subplot shows the longitudinal position of the aircraft on the runway. The second subplot shows the ground speed and indicated airspeed, followed by the the signal of the longitudinal acceleration. The fourth subplot represents the pitch angle of the aircraft. The fifth subplot draws the throttle command signal, the actual thrust signal of the left and right engine and the brake command signal. To see the rotation and derotation command the inboard flaps deflections are plotted in the last subplot of Fig. 4.37 and 4.40.

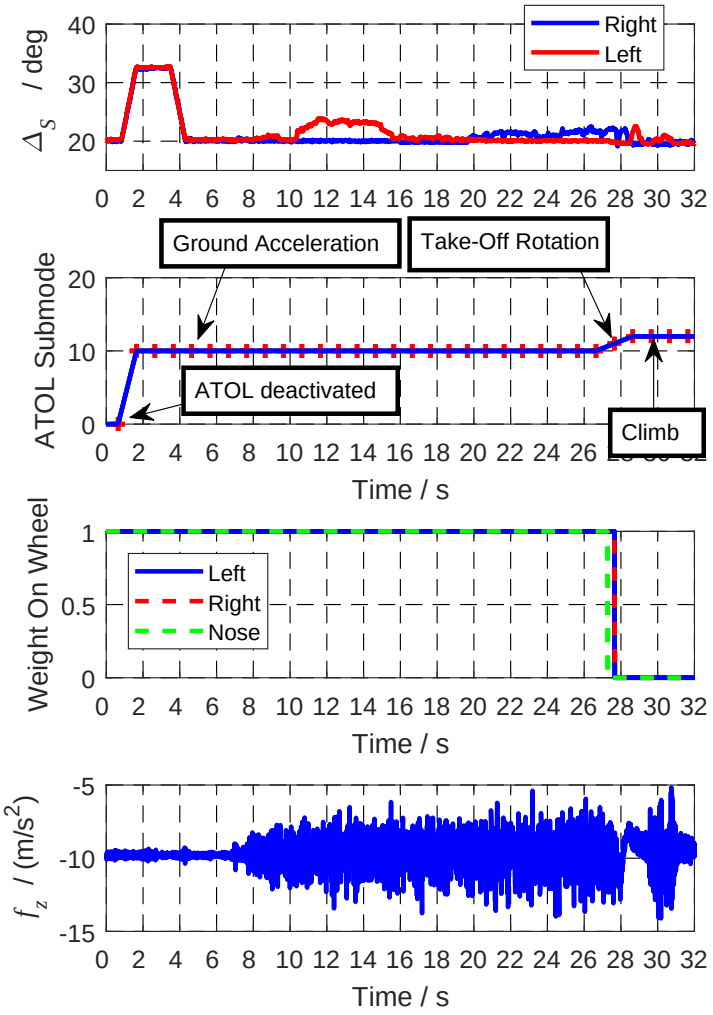


(a) Lateral motion signals of the take-off maneuver in first flight

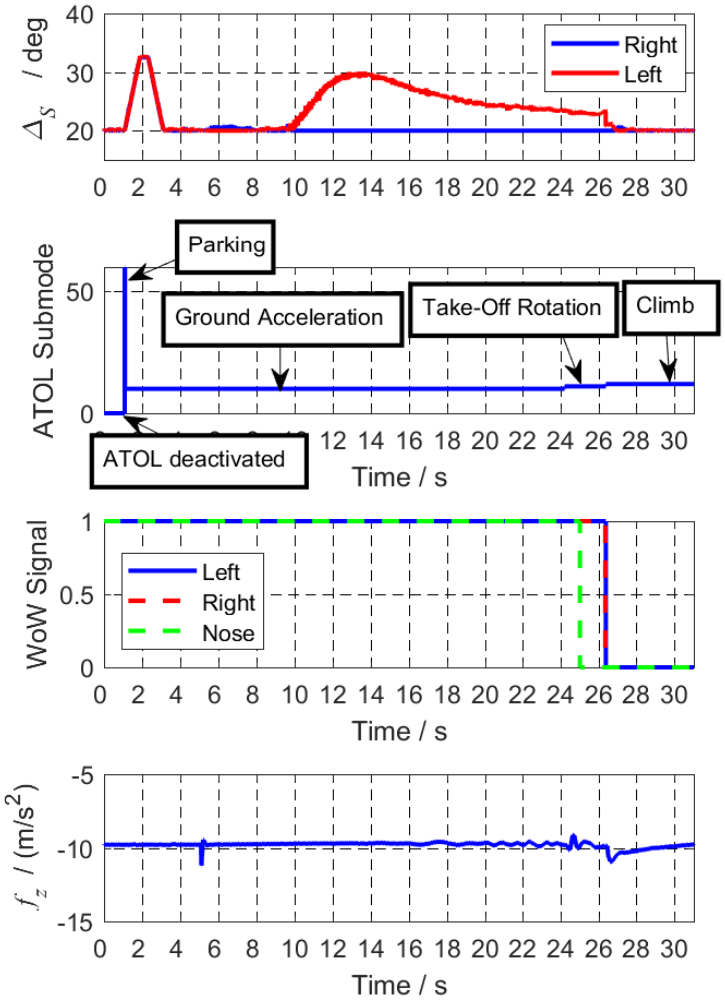


(b) Lateral motion signals of take-off maneuver with 4 kts crosswind in simulation

Figure 4.35: Take-off run at first flight of the SAGITTA demonstrator - Part 1

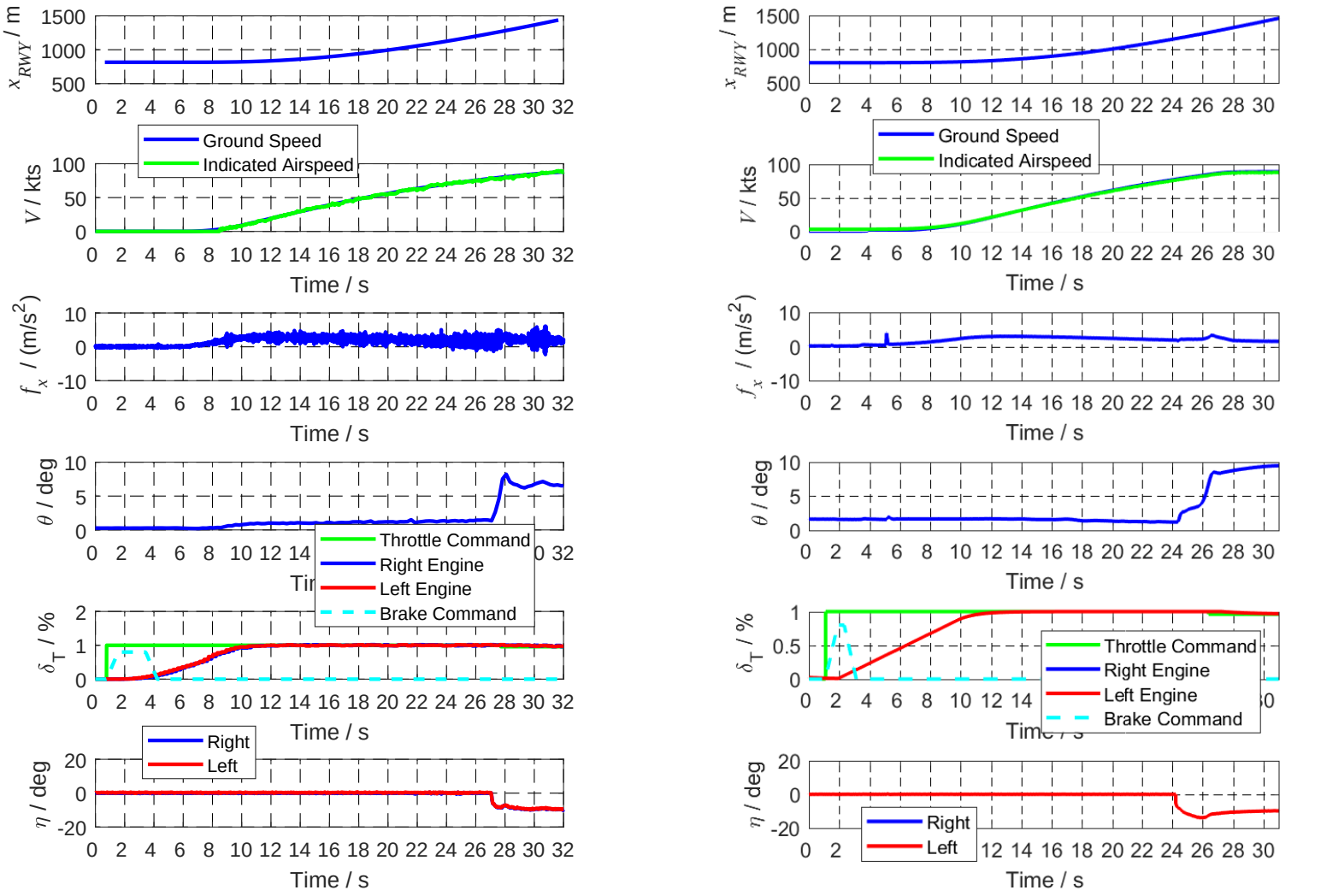


(a) Additional signals of take-off maneuver in first flight



(b) Additional signals of take-off maneuver in simulation

Figure 4.36: Take-off run at first flight of the SAGITTA demonstrator - Part 2



(a) Longitudinal motion signals of take-off maneuver in first flight

(b) Longitudinal motion signals of take-off maneuver in simulation

Figure 4.37: Take-off run at first flight of the SAGITTA demonstrator - Part 3

4.5.5.1 Automatic Take-Off Maneuver

4.5.5.1.1 Longitudinal Motion The FO activates the automatic take-off maneuver after 0.64 s. The ATOL module is enabled by the system automation module and the ATOL system activates the ground acceleration sub-mode (indicated by second subplot in Fig. 4.36(a)). The brakes are released after 3.2 s, when the thrust reaches 5 % of the maximum thrust. After about 7 s the SAGITTA aircraft starts to roll and subsequently accelerates with an averaged acceleration of 1.47 m/s^2 in reality until the take-off rotation command is triggered compared to an acceleration of 1.5 m/s^2 in simulation. After the aircraft reaches an indicated airspeed of 75.8 kts (39 m/s) the rotation command is initiated by the ATOL logic, which switches its sub-mode to "take-off rotation" (see Fig. 4.37). As in Ref. [118] described, a step plus ramp command is given to the in-board flaps to initiate the take-off rotation. In the course of the first flight the nose wheel lifted off first at 27.3 s, followed shortly by both main wheels simultaneously (see Fig.4.36(a)). The aircraft needs 23.6 s for the take-off run (from brake release to rotation command), compared to 21.9 s in simulation. Thereby, the aircraft covers a distance of 410 m (visible in longitudinal position subplot in Fig. 4.37). In simulation the aircraft exhibits 50 m shorter distance (360 m). Since the ground speed and indicated airspeed signals don't show any deviation within the whole take-off run, it can be assumed that head- or tailwind have not occur. Thus, the larger take-off run distance in reality can be related to different friction parameters between the tires of the aircraft and the runway surface. The rotation phase, which lasts from take-off rotation command (see inboard deflections) to airborne (i.e. main wheels lift-off and defined pitch angle reached) took 0.9 s at the first flight, compared to 2.2 s in simulation. In reality the aircraft rotates significantly faster than in simulation. For that reason the aircraft reaches the designed pitch angle 1.3 s earlier. This behavior is further investigated in Ref. [147]. Thus, the ground controller is enabled for a shorter time period than in simulation during the take-off rotation phase.

4.5.5.1.2 Lateral Motion As Fig. 4.35(a) shows, in reality the aircraft was positioned by the EP about 0.5 m left of the defined runway centerline with a heading deviation away from the runway direction of about 1 deg . After activation of the automatic take-off maneuver by the FO the ground controller compensates the lateral and heading deviation by a nose wheel steering command of maximum 1.6 deg. Since a light overshoot of the course command occurs, the aircraft passes over the runway centerline slightly at a speed of about 35 kts. This is compensated by a light nose wheel steering command to the left. During the take-off run a heading and course deviation appears of about 1 deg. This can be related to a light crosswind from the left, according to the written flight test cards of the first flight (Wind: 130 deg, 4 kts). For that reason the aircraft overpasses again the runway centerline slightly to the left up to a maximum of 0.4 m. As it can be seen in the fourth subplot of Fig.

4.35(a), the yaw rate of the aircraft follows adequately the commanded yaw rate in reality as well as in model-in-the-loop simulation. At 10 kts the outerboard flaps are activated for yaw rate aero-crossfeed. As a result, the flaps on the left side are deflected to compensate the course deviation of the aircraft. When the aircraft reached the rotation speed, the nose wheel lifted off first, while the main wheels were still on ground. During this short period the ground controller remains still active and calculates appropriate yaw rate commands in order to counteract deviations/ disturbances by corresponding outboard flaps deflections. As Fig. 4.35(a) underlines, the lateral deviation and course didn't change significantly during the take-off rotation. When the both main wheels indicate lift-off and the aircraft reaches a pitch angle of about 5 deg, the ATOL module switches off the ground controller and the nose wheel steering command is set to zero. This is visible in the last subplot plot in Fig. 4.35(a) at 27.7 s. After the aircraft is airborne, the lateral as well as the longitudinal CSAS are enabled for the initial climb phase by the ATOL module.
The take-off maneuver of the simulation is shown with 4 kts crosswind from the right (i.e. crosswind in opposite direction as in first flight). The aircraft is initialized in simulation with a heading deviation to the runway direction of about 1 deg. This is compensated by the ground controller with a maximum nose wheel steering deflection of 1.8 deg, analogously to the first flight. Similarly to the first flight data, the heading and course deviation increases to 1 deg. Due to the higher absolute crosswind within the applied simulation maneuver compared to the first flight, the maximum lateral deviation is about 1 m higher. Accordingly, the ground controller commands higher nose wheel steering angle deflections and outboard flaps deflections in simulation in order to compensate the crosswind disturbances and the lateral deviation error.

4.5.5.2 Automatic Landing Maneuver

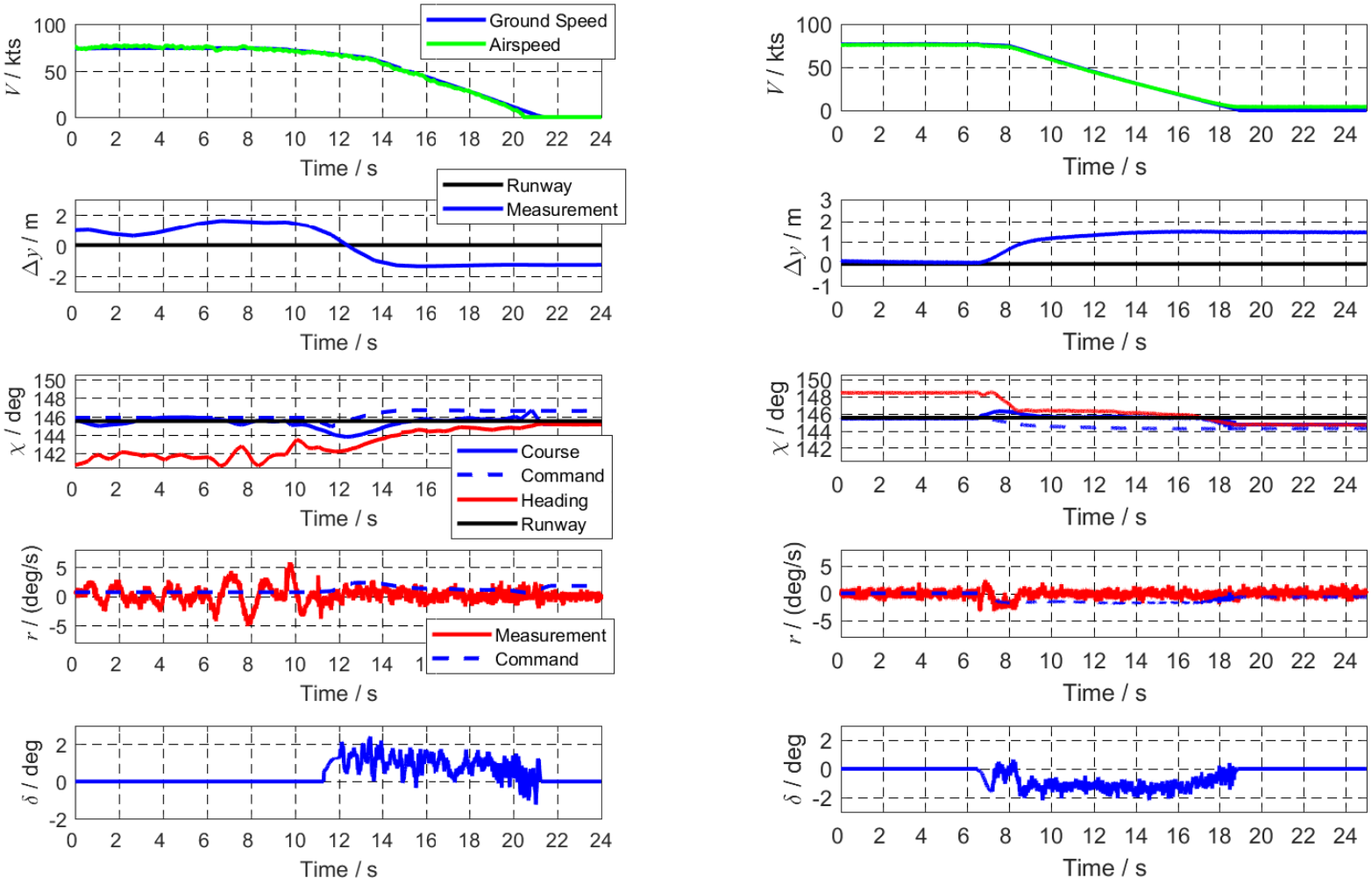
Analogously to the take-off maneuver, the landing maneuver will be discussed, analyzed and compared to the simulation in the next section. The plots of the landing maneuver are shown in Fig. 4.38 and Fig. 4.40 for longitudinal and lateral motion.

4.5.5.2.1 Longitudinal Motion When the aircraft is guided correctly into the respective area for landing after its mission and the conditions for automatic landing activation are fulfilled, the FO sets the mode *ATOL landing*. The automatic landing maneuver starts by longitudinal and lateral trajectory-controlled waypoint navigation. After a certain height is reached, the ATOL module commands a flare maneuver based on vertical speed commands. Below a specific height above ground the ATOL module switches into the final descent sub-mode. This is visible in the real-live data at 6.3 s in Fig. 4.40. Within this sub-mode, the throttle command is reduced linearly as shown in the fourth plot of Fig. 4.40. Finally, when the touch down is detected, the ATOL module shifts into the ATOL landing derotation sub-mode

and the ground controller described above becomes active. The touch down is detected, if the weight on wheels sensors of both main wheels indicate *on ground*. The weight on wheel signal of the landing maneuver is shown in the third subplot of Fig. 4.39(a) and 4.39(b). In reality during the first flight, the right weight on wheel sensor indicates *on ground* at 9.7 s, whereas the left and the nose wheel weight on wheel sensors show still *in air*. After the right weight on wheel sensor again has indicated *in air* at 9.9 s, the sensor signal displays and remains *on ground* from 10.9 s. At 11.3 s, the left weight on wheel sensor indicates also *on ground* for 0.1 s and jumps back to *in air*. At 11.6 s, the nose wheel and the left main weight on wheel sensor signal show *on ground* as the right weight on wheel sensor. One subplot below the weight on wheel signals, the vertical acceleration f_z is shown. The first peak of the vertical acceleration signal at 9.7 s coincides with the first *on ground* indication of the right weight on wheel sensor signal. The second peak concurs with the *on ground* indication of the nose wheel weight on wheel sensor signal. However, the plot and video material indicate that the main wheels almost touched the ground simultaneously. The switching is executed, when the system automation module gives the indication *aircraft on ground*. The system automation module states *aircraft on ground*, when both weight on wheel sensors of the main wheel show *on ground*. Taken into account the weight on wheel sensor characteristics mentioned above, this implies, that the switching to the landing derotation sub-mode was delayed. Further discussions can be found in Ref. [147]. In contrast, in the simulation the weight on wheel sensors indicate *on ground* simultaneously at 6.5 s, and 0.1 s later, the nose wheel weight on wheel sensor indicates *on ground* as well.
While derotating the ground controller alignes the nose wheel in runway direction by comparison of the actual aircraft heading and the known runway direction. This is visible in the nose wheel steering subplot in Fig. 4.38(a) and 4.38(b).
The landing derotation mode is activated at 11.3 s in the real-live flight data, visible by the nose wheel steering command step. When the aircraft's pitch angle persists below 2.8 deg for 0.5 s and both main wheels indicate *on ground*, the ground controller is switched to the landing deceleration mode. This switching occurs at 11.8 sec in reality and in simulation at 7.2 s. Then at 13.0 s in reality, the ground controller activates the brakes of the main wheels and the split flaps for braking. This is visible in the first subplot of Fig. 4.39(a),4.39(b) and the fifth subplot of Fig 4.40(a) and 4.40(b).
As the braking signal in Fig. 4.39(a) indicates, the braking is not commanded at the same time as the switching to the landing deceleration mode is performed. In the flight control software, the braking is commanded after touch down, if the deviation of the heading signal and the runway direction is below 2.5 deg or a maximum time of 2.5 s is reached after the landing deceleration mode is activated. This ensures on the one hand, that the aircraft has time to align with the centerline before the lateral tracking of the ground control is disrupted by the braking force. On the other hand, it is avoided, that the aircraft approaches the runway edge,

in case the heading signal deviation to the runway direction is still higher than 2.5 deg after the maximum time of 2.5 s. Since the heading deviation to the runway heading is more than 2.5 deg, when the deceleration is activated, the braking command is given 1.4 s in reality and 0.3 s later in simulation. Additional to the braking of the main wheels, the split flaps are deflected 12.5 deg symmetrically for braking (see first subplot Fig. 4.39(a) and 4.39(b)).
The rolling distance after touch down is about 200 m in reality, whereas the aircraft comes to a stop after 260 m in simulation. (see first subplot of Fig. 4.40(a) and 4.40(b)). Due to the deceleration, a negative pitch moment is produced in simulation as well as in reality. The pitching angle decreases visibly for that reason, when the brakes are activated (see Fig. 4.40).

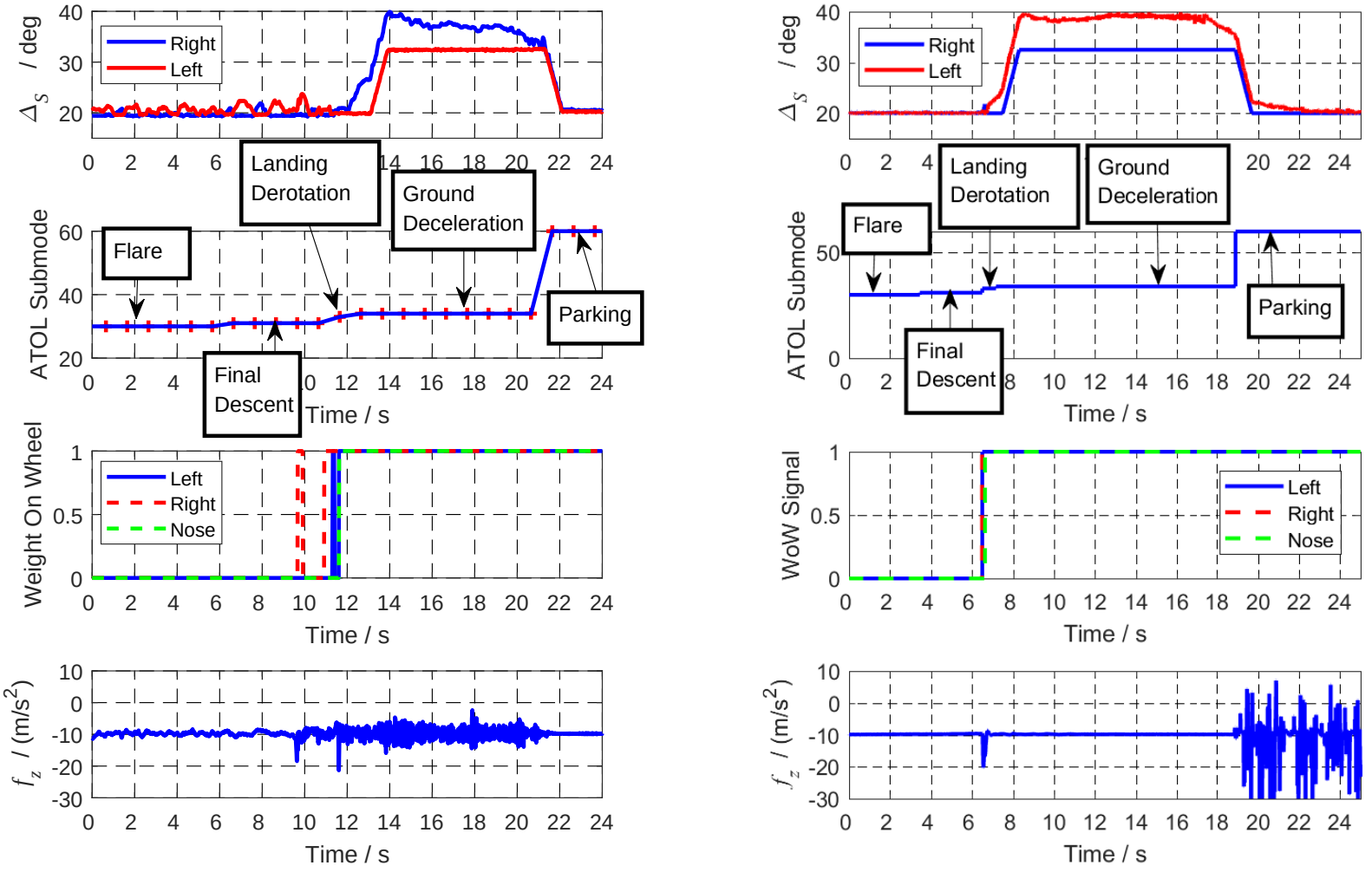
4.5.5.2.2 Lateral Motion In reality, due to the estimated 5 kts crosswind, the course signal and the heading signal deviate about 4.5 deg. While the descent in reality, the aircraft has a maximum deviation of 1.5 m to the right of the runway centerline (i.e. the aircraft flies on the right site of the centerline and the aircraft's nose shows to the left compared to the runway direction). (see first subplot of Fig. 4.38(a) and 4.38(b)). In the course of the landing derotation the split flaps are not used for lateral tracking of the runway centerline and the nose wheel is aligned with the runway. Only the yaw damper of the lateral CSAS controller is active. This shall avoid contrary commands by the tracking loop and the yaw damper. For that reason the aircraft crosses the centerline to the left site in reality. The touch down reduces the deviation between the heading and the course signal slightly due to a yaw moment in runway direction, visible in the plots of the simulation and reality. No decrab maneuver is implemented in the flight control software. Indeed, in the case of a crosswind landing maneuver it is relied on the stabilizing yaw moment, produced by the cornering stiffness angles of the main wheels. When the landing deceleration mode is activated, the split flaps and the nose wheel steering are released for the runway centerline tracking. In reality, the ground controller counteracts the lateral deviation to the left by positive (i.e. right) nose wheel steering deflections. For that reason, the lateral deviation does not increase further and the aircraft comes to a stop with a lateral deviation of less than 1.5 m. The course deviation to the runway direction is reduced almost to zero and yaw rate follows the yaw rate command quite acceptable. The lateral deviation is not further reduced due to the crosswind from the left site of the runway in reality. Similar behavior is visible in the simulation plots. In contrast, the crosswind comes from the right and the nose wheel steering command is to the left (i.e. negative).



(a) Lateral motion signals of landing maneuver in first flight

(b) Lateral motion signals of landing maneuver in simulation

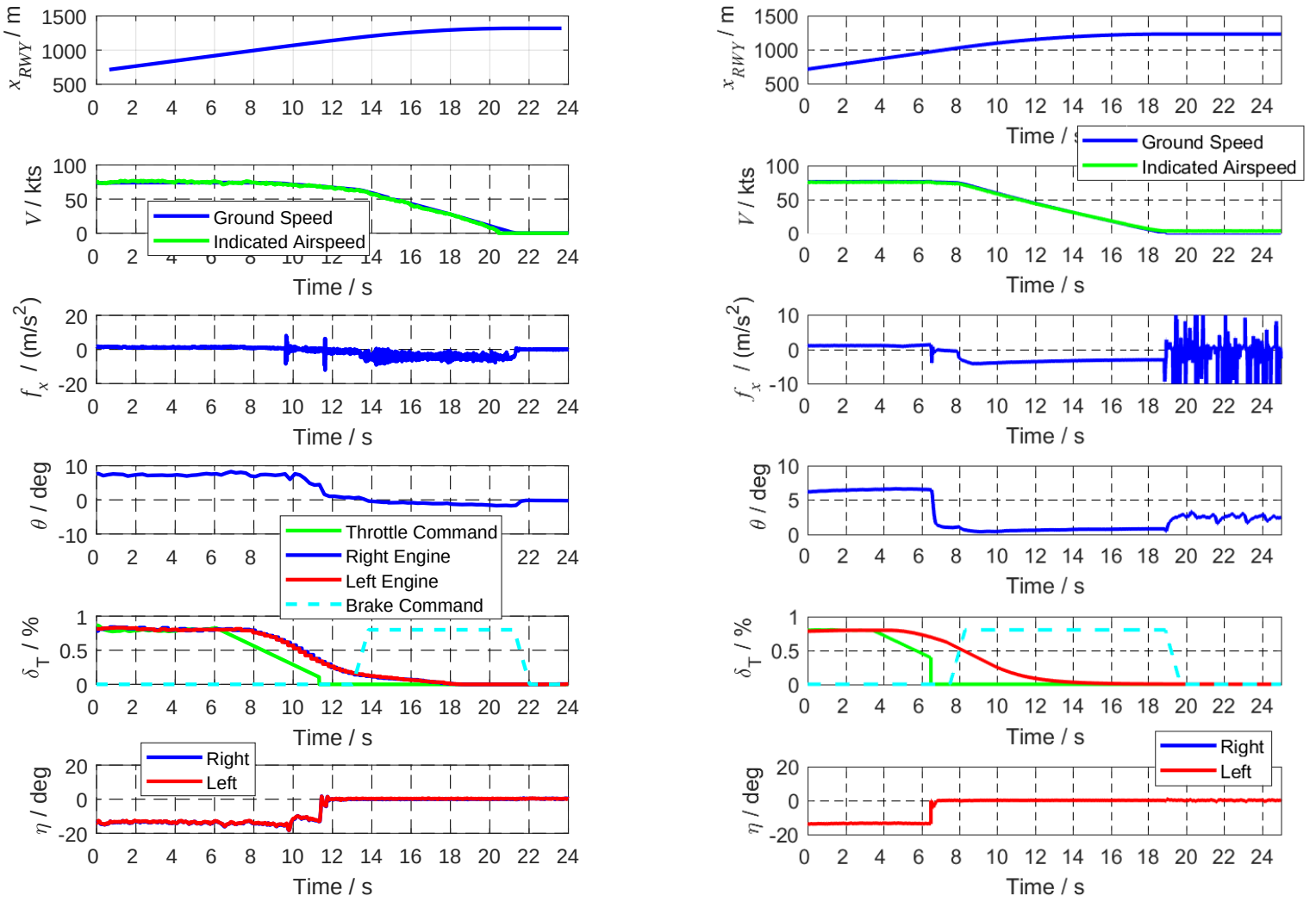
Figure 4.38: Automatic landing at first flight of the SAGITTA demonstrator - Part 1



(a) Additional signals of landing maneuver in first flight

(b) Additional signals of landing maneuver in simulation

Figure 4.39: Automatic landing at first flight of the SAGITTA demonstrator - Part 2



(a) Longitudinal motion signals of landing maneuver in first flight (b) Longitudinal motion signals of landing maneuver in simulation

Figure 4.40: Automatic landing at first flight of the SAGITTA demonstrator - Part 3

4.6 Summary

In July 2017 the SAGITTA demonstrator UAV accomplished two successful flight missions within its first flight campaign at the Denel Overberg Testrange, South Africa.
As the presented comparison of the recorded flight test data with corresponding simulation results highlight, the ground controller exhibited appropriate performance and functionalities and thus, satisfying overall capabilities for flight. Hence, a second flight in identical configuration was conducted successfully just after the maiden flight. The aircraft was controlled safely and adequately on ground during the whole take-off and landing maneuvers. Both flights during the first flight campaign were restricted to a crosswind limitation of 5 kts. However, for further flights, the crosswind limitation may be opened from the perspective of the ground controller performance assessment in order to demonstrate the robustness and performance at higher crosswind.

Chapter 5

Conclusion, Summary and Outlook

The recent aerospace evolution has driven start-ups and established companies to design and develop new types of aerial vehicles with new configurations and capabilities for exceptional applications. The thesis at hand contributes three functional enhancements for novel flight control systems with regard to such aerial vehicles. The subsequent section summarizes the achieved contributions and respective future work.

Contribution I:
Development of an Automatic Geofencing Algorithm For Three-Dimensional Space

Conclusion and Summary: The contribution proposes a novel geofencing concept for two- and three- dimensional space in order to reduce the risk in-air and on-ground for future aerial vehicles. The algorithm concept can be used for any unmanned or manned aerial vehicle in wing-borne flight to prevent entering prohibited airspace or leaving selected flight areas. To reduce the computational complexity within the flight control system, the algorithm has been designed such that complex computational tasks are executed offline before flight and only relevant tasks are executed online. The algorithm represents thus a risk mitigation strategy, which is relevant for the certification of future unmanned aerial vehicles, most importantly in the specific category. For the growing aerospace market and the increasing number of airspace participants, automatic geofencing algorithms on the aircraft flight control system will be inevitable in future. The presented concept considers thereby a flight area selected by the flight operator and the aerial vehicle limitations in wing-borne flight. The algorithm preprocesses the selected flight area and determines their validity with respect to the aircraft performance. A safety boundary is generated, such that the geofencing algorithm prevents the aerial vehicle from leaving the flight area. The flight area can be defined in two- or in three-dimensional space. Evasive maneuvers have been designed, which are initiated, if the aircraft hits the generated safety area boundary. The presented algorithm is embedded within a classical autopilot and has been demonstrated within nonlinear closed-loop simulation using a high-fidelity aircraft model.

Future Work: In future work, the implemented algorithm needs to be tested further in closed-loop simulation (mainly with wind and disturbances). Moreover, the simultaneous push-down and turn maneuver needs to be investigated and implemented. The application of the algorithm to a real flight control system is the next step, such that the algorithm can be verified in flight. Additional wind considerations can be also taken into account. The safety margin, for example, can be calculated online in dependence of wind data, what requires higher complexity on the aircraft's flight control system. The algorithm can be also applied as last-line of defense for future detect-and-avoid algorithms.

Contribution II:
Novel Stability & Robustness Analysis for Future Urbain Air Mobility

Conclusion and Summary: This part of the thesis contributes several important aspects with regard to the robustness analysis for future aerial vehicles and shows their successful application to a real-live eVTOL demonstrator.
First, this thesis exposes properties of the SISO gain and phase margin and the nichols diamond. It details the MIMO gain and phase margins of the sensitivity transfer function, of the complementary sensitivity transfer function and of the balanced sensitivity transfer function and supplements additional derivation steps.
Moreover, the thesis contributes the assessment of their conservatism. It has been proven successfully by using Fan's Theorem, that the upper gain of margin of the sensitivity transfer function is always the lowest conservative upper MIMO gain margin in comparison to the margins of the other two sensitivity transfer functions. Further, it has been proven, that the lower gain margin of the complementary sensitivity transfer function is always the lowest conservative lower MIMO gain margin. A sub-domain of the maximum singular values of the sensitivity transfer functions has been determined, in which the balanced MIMO phase margin shows lowest conservatism. To the author's knowledge, these are new mathematical proofs, which are essential for the correct application of the MIMO margins to real-live systems.
Furthermore, the thesis presents an approach utilizing the μ-lower bound to calculate a worst-case dynamic transfer function for scalar complex uncertainties. As a result, the approach enables to identify worst-case dynamics of a sub-system (e.g. actuators), which drive the system to the stability borderline or a specified conformity borderline. In consequence, the approach verifies the existence of dynamic uncertainties (in terms of a transfer function), which bring the system to the specified conformity borderline. To the author's knowledge, this is a novelty in the context of the μ-analysis.
Another contribution within this part is a new proposed probabilistic-μ approach which calculates a lower bound of the probability to achieve stability or a specified conformity borderline.

Thereby, unbounded stochastic uncertainties are considered. Note that uncertainties, considered in the classical μ-analysis, are defined by limited intervals and without specification of probability density functions. The result of the new proposed probabilistic-μ approach is a lower bound of the probability, which is in contrast to monte-carlo methods.
The presented theoretical contributions have been successfully applied to a real-live closed-loop system, which consists inter alia of an eVTOL flight dynamic model and an nonlinear INDI controller. The conservatism proof has been successfully verified by the calculated MIMO margins. SISO, MIMO and the structured singular value robustness margins have been contrasted over the horizontal velocity envelope of the aerial vehicle. The unification of the different elaborated margin thereby represent an allowed flight envelope for the first flights of the aerial vehicle.
Worst-case dynamics for the actuators have been identified, which bring the system to ADS-33 Level 2 handling quality borderline.

The probabilistic μ-analysis has been applied under consideration of 78 unbounded stochastic uncertainties. By using the bi-level optimization of the presented probabilistic μ-analysis the probability of failure could be reduced by approximately 50 $\%$.

Outlook and Future Work: Since the presented robustness metrics have been applied to a real-live aircraft system, the approach and the results need to be validated in flight. The calculated horizontal flight envelope can be used as initial allowed flight envelope for the first flights.
In addition, the presented probabilistic-μ approach has further potential to achieve a even less conservative lower bound of the probability for stability or conformity.
Multiple elaborated ideas can reduce the conservatism of the probabilistic-μ approach. For example, future work can focus on the identification of colored uncertainties. These are uncertainties, which occur as defined uncertainties in the system. However, they don't effect the stability or the conformity. If uncertainties are identified as colored uncertainties, it is not required to consider them for the probabilistic-μ approach. In that way, the conservatism of the probabilistic-μ approach can be further reduced.
Moreover, the complementary of the calculated uncertainty set, for which stability or conformity is guaranteed, can be also used as initial set for monte-carlo methods. This approach would combine monte-carlo samplings methods and the advantages of the probabilistic-μ analysis.

Contribution III:
Design, Development and Testing of Automatic Ground Control Functions of a Novel UAV for Automatic Take-Off and Landing:

Summary and Conclusion: This contribution shows any engineers, who want to fly a novel fixed-wing aerial vehicle automatically, how to develop the respective ground control functions of the flight control system for automatic take-off and landing including take-off rotation and derotation. The development is shown from the design, the implementation and ground tests up-to the first flight step-by-step.
The presented automatic ground control functions have been designed and developed for a novel diamond-shaped UAV with about 150 kg take-off weight. The automatic ground functions are active during the take-off and landing phase of the aerial vehicle and comprise the aircraft's control functions for runway centerline tracking and the automatic braking functions. They have been demonstrated in various taxi-tests of the real aircraft by using a specifically designed taxi-test software. The functions have been successfully demonstrated in flight during the first flight test campaign of the demonstrator in Overberg, South Africa. Thus, the functions have been validated on the real aircraft in flight. The presented approach is not limited to the demonstrator aircraft and can be applied to any other unmanned or optionally-piloted vehicle in future. As a result, this thesis contributes the development of automatic ground functions for novel fixed-wing aerial vehicles.

Future Work: The first and second flight of the diamond-shaped demonstrator has been conducted in calm-air with crosswind limits below 5 kts. The crosswind envelope can be extended in future test. The presented approach can be also applied to future fixed-wing aircraft to prove the generality of the presented ground controller approach for future aerial vehicles.

Appendix A

Stability and Robustness

A.1 MIMO Margins

A.1.1 Complementary Sensitivity Function

A.1.1.1 Derivation of transfer functions

Derivation of the following transfer functions, shown in Fig. 3.17(c):

$$
\begin{aligned}
\boldsymbol{du}_T \;\rightarrow\; \boldsymbol{u} \\
\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,(\boldsymbol{du}_T + \boldsymbol{u}) \\
\boldsymbol{u} &= \underline{\boldsymbol{K}}(s)\,(\boldsymbol{r} - \boldsymbol{y}) \\
\boldsymbol{u} &= -\,\underline{\boldsymbol{K}}(s)\,\boldsymbol{y} \quad \text{with} \quad \boldsymbol{r} = 0 \\
\boldsymbol{u} &= -\,\underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\,(\boldsymbol{du}_T + \boldsymbol{u}) \\
\boldsymbol{u} + \underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\,\boldsymbol{u} &= -\,\underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\,\boldsymbol{du}_T \\
\left(\underline{\boldsymbol{I}} + \underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\right)\boldsymbol{u} &= -\,\underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\,\boldsymbol{du}_T \\
\boldsymbol{u} &= -\underbrace{\left(\underline{\boldsymbol{I}} + \underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\right)^{-1}\underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)}_{\underline{\boldsymbol{T}}_I}\boldsymbol{du}_T
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{r} \;&\rightarrow\; \boldsymbol{y} \\
\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,\boldsymbol{u} \\
\boldsymbol{u} &= \underline{\boldsymbol{K}}(s)(\boldsymbol{r}-\boldsymbol{y}) \\
\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)(\boldsymbol{r}-\boldsymbol{y}) \\
\boldsymbol{y}+\underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)\boldsymbol{r} \\
\big(\underline{\boldsymbol{I}}+\underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)\big)\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)\boldsymbol{r} \\
\boldsymbol{y} &= \underbrace{\big(\underline{\boldsymbol{I}}+\underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)\big)^{-1}\,\underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)}_{\underline{\boldsymbol{T}}_O}\boldsymbol{r}
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{du}_T \rightarrow \boldsymbol{y} \\
\boldsymbol{u} &= \boldsymbol{u}+\boldsymbol{du}_T \\
\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,\boldsymbol{u} \\
\boldsymbol{u} &= \underline{\boldsymbol{K}}(s)\,(\boldsymbol{r}-\boldsymbol{y}) = -\underline{\boldsymbol{K}}(s)\,\boldsymbol{y} \\
\boldsymbol{u} &= -\,\underline{\boldsymbol{K}}(s)\,\boldsymbol{y}+\boldsymbol{du}_T \\
\boldsymbol{y} &= \underline{\boldsymbol{G}}\,\big(-\,\underline{\boldsymbol{K}}(s)\,\boldsymbol{y}+\boldsymbol{du}_T\big) \\
\big(\underline{\boldsymbol{I}}+\underline{\boldsymbol{G}}(s)\,\underline{\boldsymbol{K}}(s)\big)\,\boldsymbol{y} &= \underline{\boldsymbol{G}}\,\boldsymbol{du}_T \\
\boldsymbol{y} &= \underbrace{\big(\underline{\boldsymbol{I}}+\underline{\boldsymbol{G}}\,\underline{\boldsymbol{K}}(s)\big)^{-1}}_{\underline{\boldsymbol{S}}_O}\,\underline{\boldsymbol{G}}(s)\,\boldsymbol{du}_T
\end{aligned}
$$

$$
\begin{aligned}
\boldsymbol{r} \;&\rightarrow\; \boldsymbol{u} \\
\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,\boldsymbol{u} \\
\boldsymbol{u} &= \underline{\boldsymbol{K}}(s)\,(\boldsymbol{r}-\boldsymbol{y}) \\
\boldsymbol{u} &= \underline{\boldsymbol{K}}(s)\,(\boldsymbol{r}-\underline{\boldsymbol{G}}(s)\,\boldsymbol{u}) \\
\boldsymbol{u} &= \underline{\boldsymbol{K}}(s)\,\boldsymbol{r}-\underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\,\boldsymbol{u} \\
\big(\underline{\boldsymbol{I}}+\underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\big)\boldsymbol{u} &= \underline{\boldsymbol{K}}(s)\,\boldsymbol{r} \\
\boldsymbol{u} &= \underbrace{\big(\underline{\boldsymbol{I}}+\underline{\boldsymbol{K}}(s)\,\underline{\boldsymbol{G}}(s)\big)^{-1}}_{\underline{\boldsymbol{S}}_I}\,\underline{\boldsymbol{K}}(s)\,\boldsymbol{r}
\end{aligned}
$$

A.1.1.2 Derivation of the MIMO phase margin of the complementary sensitivity transfer function

From Eq. (3.74), the following holds:

$$1 + r_{T,min}\ e^{j\phi} = e^{j\theta_T} \tag{A.1}$$

With the complex number j and the Euler rule $e^{j\theta} = \cos(\theta) + j\sin(\theta)$, we gain:

$$1 + r_{T,min}\ \cos(\phi) + r_{T,min}\ j\sin(\phi) = \cos(\theta_T) + j\sin(\theta_T) \tag{A.2}$$

Separating the real and complex part of the equation, it results:

$$1 + r_{T,min}\ \cos(\phi) = \cos(\theta_T) \tag{A.3}$$

$$r_{T,min}\ \sin(\phi) = \sin(\theta_T) \tag{A.4}$$

From the complex part, we get:

$$\phi = \operatorname{asin}\left(\frac{\sin(\theta_T)}{r_{T,min}}\right) \tag{A.5}$$

With the trigonometric relation

$$\cos(\operatorname{asin}(x)) = \sqrt{1-x^2} \tag{A.6}$$

and by inserting the equation into the real part equation, we obtain:

$$1 + r_{T,min}\ \cos(\phi) = \cos(\theta_T) \tag{A.7}$$

$$1 + r_{T,min}\ \cos\left(\operatorname{asin}\left(\frac{\sin(\theta_T)}{r_{T,min}}\right)\right) = \cos(\theta_T) \tag{A.8}$$

$$1 + r_{T,min}\ \sqrt{1-\left(\frac{\sin(\theta_T)}{r_{T,min}}\right)^2} = \cos(\theta_T) \tag{A.9}$$

$$r_{T,min}^2\left(1-\left(\frac{\sin(\theta_T)}{r_{T,min}}\right)^2\right) = \Big(\cos(\theta_T)-1\Big)^2 \tag{A.10}$$

$$r_{T,min}^2 - \sin^2(\theta_T) = \cos^2(\theta_T) - 2\cos(\theta_T) + 1 \tag{A.11}$$

$$r_{T,min}^2 + 2\cos(\theta_T) - 1 = \cos^2(\theta_T) + \sin^2(\theta_T) = 1 \tag{A.12}$$

$$r_{T,min}^2 + 2\cos(\theta_T) = 2 \tag{A.13}$$

$$1 - \cos(\theta_T) = \frac{r_{T,min}^2}{2} \tag{A.14}$$

Using the following trigonometric relation and considering $y = x/2, \quad x, y \in \mathbb{R}$

$$\sin^2(y) = \frac{1}{2}(1 - \cos(2y)) \tag{A.15}$$

$$\sin^2\left(\frac{x}{2}\right) = \frac{1}{2}(1 - \cos(x)) \tag{A.16}$$

$$2\sin^2\left(\frac{x}{2}\right) = 1 - \cos(x) \tag{A.17}$$

$$\tag{A.18}$$

we yield:

$$2\sin^2\left(\frac{\theta_T}{2}\right) = \frac{r_{T,min}^2}{2} \tag{A.19}$$

$$\sin^2\left(\frac{\theta_T}{2}\right) = \frac{r_{T,min}^2}{4} \tag{A.20}$$

$$\sin\left(\frac{\theta_T}{2}\right) = \frac{r_{T,min}}{2} \tag{A.21}$$

This leads to the equation of the MIMO phase margin of the complementary sensitivity transfer function, which is equivalent to the result from the geometrical interpretation of Fig. 3.19:

$$\boxed{\theta_T = 2\,\mathrm{asin}\left(\frac{r_{T,min}}{2}\right)} \tag{A.22}$$

A.1.2 Sensitivity Function

A.1.2.1 Derivation of transfer function

Derivation of the following transfer function, shown in Fig. 3.20:

$$
\begin{aligned}
\boldsymbol{du_S} \;&\rightarrow\; \boldsymbol{u} \\
\boldsymbol{u} &= \boldsymbol{u^K} + \boldsymbol{du_S} \\
\boldsymbol{u^K} &= \underline{\boldsymbol{K}}(s)(\boldsymbol{r}-\boldsymbol{y}) = -\underline{\boldsymbol{K}}(s)\boldsymbol{y} \\
\boldsymbol{y} &= \underline{\boldsymbol{G}}(s)\,\boldsymbol{u} \\
\boldsymbol{u} &= -\underline{\boldsymbol{K}}(s)\boldsymbol{y} + \boldsymbol{du_S} \\
\boldsymbol{u} &= -\underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)\,\boldsymbol{u} + \boldsymbol{du_S} \\
\big(\underline{\boldsymbol{I}} + \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)\big)\boldsymbol{u} &= \boldsymbol{du_S} \\
\boldsymbol{u} &= \underbrace{\big(\underline{\boldsymbol{I}} + \underline{\boldsymbol{K}}(s)\underline{\boldsymbol{G}}(s)\big)^{-1}}_{\underline{\boldsymbol{S}}_I}\boldsymbol{du_S}
\end{aligned}
$$

A.1.2.2 Derivation of the MIMO Phase Margin of the sensitivity transfer function

From Eq. (3.86):

$$\frac{1}{1 - r_{S,min}\; e^{j\phi}} = e^{j\theta_S} \tag{A.23}$$

$$1 - r_{S,min}\; e^{j\phi} = e^{-j\theta_S} \tag{A.24}$$

With Euler rule and the complex number j, we get:

$$1 - r_{S,min}\cos(\phi) - r_{S,min}\, j\sin(\phi) = \cos(\theta_S) - j\sin(\theta_S) \tag{A.25}$$

With real and complex part equalization:

$$1 - r_{S,min}\cos(\phi) = \cos(\theta_S) \tag{A.26}$$

$$-r_{S,min}\sin(\phi) = \sin(\theta_S) \tag{A.27}$$

From the real part, we get:

$$\cos(\phi) = \frac{-\cos(\theta_S) + 1}{r_{S,min}} \tag{A.28}$$

$$\phi = \operatorname{acos}\Big(\frac{1 - \cos(\theta_S)}{r_{S,min}}\Big) \tag{A.29}$$

By inserting this equation into the complex part, the subsequent equation holds:

$$-r_{S,min}\sin(\phi) = \sin(\theta_S) \tag{A.30}$$

$$-r_{S,min}\sin\left(\operatorname{acos}\left(\frac{1-\cos(\theta_S)}{r_{S,min}}\right)\right) = \sin(\theta_S) \tag{A.31}$$

With the trigonometric function

$$\sin(\operatorname{acos}(x)) = \sqrt{1-x^2} \tag{A.32}$$

the equation becomes:

$$-r_{S,min}\sqrt{1-\left(\frac{1-\cos(\theta_S)}{r_{S,min}}\right)^2} = \sin(\theta_S) \tag{A.33}$$

$$r_{S,min}^2\left(1-\left(\frac{1-\cos(\theta_S)}{r_{S,min}}\right)^2\right) = \sin^2(\theta_S) \tag{A.34}$$

$$r_{S,min}^2 - \left(1-\cos(\theta_S)\right)^2 = \sin^2(\theta_S) \tag{A.35}$$

$$r_{S,min}^2 - (1-2\cos(\theta_S)+\cos^2(\theta_S)) = \sin^2(\theta_S) \tag{A.36}$$

$$r_{S,min}^2 - 1 + 2\cos(\theta_S) - \cos^2(\theta_S)) = \sin^2(\theta_S) \tag{A.37}$$

$$r_{S,min}^2 - 1 + 2\cos(\theta_S)) = \sin^2(\theta_S) + \cos^2(\theta_S) = 1 \tag{A.38}$$

$$r_{S,min}^2 + 2\cos(\theta_S) = 2 \tag{A.39}$$

$$1-\cos(\theta_S) = \frac{r_{S,min}^2}{2} \tag{A.40}$$

Using the following trigonometric function and considering $y = x/2, \;\; x,y \in \mathbb{R}$

$$\sin^2(y) = \frac{1}{2}(1-\cos(2y)) \tag{A.41}$$

$$\sin^2\left(\frac{x}{2}\right) = \frac{1}{2}(1-\cos(x)) \tag{A.42}$$

$$2\sin^2\left(\frac{x}{2}\right) = 1-\cos(x) \tag{A.43}$$

$$\tag{A.44}$$

we obtain:

$$2\sin^2\left(\frac{\theta_S}{2}\right) = \frac{r_{S,min}^2}{2} \tag{A.45}$$

$$\sin^2\left(\frac{\theta_S}{2}\right) = \frac{r_{S,min}^2}{4} \tag{A.46}$$

$$\sin\left(\frac{\theta_S}{2}\right) = \frac{r_{S,min}}{2} \tag{A.47}$$

$$\tag{A.48}$$

This leads to the presented phase margin of the sensitivity transfer function:

$$\boxed{\theta_S = 2\,\mathrm{asin}\left(\frac{r_{S,min}}{2}\right)} \tag{A.49}$$

A.1.3 Balanced Sensitivity Function

Derivation of the MIMO Phase Margin of the balanced sensitivity transfer function

Under consideration of $r_{ST,i} = r_{ST,min}$, if the uncertainty term from Eq. (3.94)

$$\frac{1-\Delta_i}{1+\Delta_i} \quad \text{with } \Delta_i = r_{ST,i}\, e^{j\phi} \tag{A.50}$$

adds only phase to the system and the gain of term stays equal to one, the two subsequent equations hold:

$$\frac{1 + r_{ST,min}\, e^{j\phi}}{1 - r_{ST,min}\, e^{j\phi}} = e^{j\theta} \tag{A.51}$$

$$\left|\left|\frac{1 + r_{ST,min}\, e^{j\phi}}{1 - r_{ST,min}\, e^{j\phi}}\right|\right|_2 = 1 \tag{A.52}$$

With Euler's rule

$$e^{j\phi} = \cos(\phi) + j\sin(\phi) \tag{A.53}$$

the imaginary and real term of the left side of Eq. (A.51) can be calculated with:

$$\frac{1+r_{ST,min}\ e^{j\phi}}{1-r_{ST,min}\ e^{j\phi}} = \tag{A.54}$$

$$= \frac{1+r_{ST,min}\ \cos(\phi)+jr_{ST,min}\ \sin(\phi)}{1-r_{ST,min}\ \cos(\phi)-jr_{ST,min}\ \sin(\phi)} = \tag{A.55}$$

$$= \frac{(1+r_{ST,min}\ \cos(\phi)+jr_{ST,min}\ \sin(\phi))(1-r_{ST,min}\ \cos(\phi)+jr_{ST,min}\ \sin(\phi)}{(1-r_{ST,min}\ \cos(\phi)-jr_{ST,min}\ \sin(\phi))(1-r_{ST,min}\ \cos(\phi)+jr_{ST,min}\ \sin(\phi))} = \tag{A.56}$$

$$= \frac{1-r_{ST,min}^2+2jr_{ST,min}\sin(\phi)}{(1-r_{ST,min}\ \cos(\phi))^2+(r_{ST,min}\ \sin(\phi))^2} = \tag{A.57}$$

$$= \underbrace{\frac{1-r_{ST,min}^2}{(1-r_{ST,min}\ \cos(\phi))^2+(r_{ST,min}\ \sin(\phi))^2}}_{Re} + j\underbrace{\frac{2r_{ST,min}\sin(\phi)}{(1-r_{ST,min}\ \cos(\phi))^2+(r_{ST,min}\ \sin(\phi))^2}}_{Im} \tag{A.58}$$

$$\tag{A.59}$$

From Eq. (A.52), it follows:

$$\underbrace{\sqrt{\underbrace{\left(\frac{1-r_{ST,min}^2}{(1-r_{ST,min}\ \cos(\phi))^2+(r_{ST,min}\ \sin(\phi))^2}\right)^2}_{Re^2}+\underbrace{\left(\frac{2r_{ST,min}\sin(\phi)}{(1-r_{ST,min}\ \cos(\phi))^2+(r_{ST,min}\ \sin(\phi))^2}\right)^2}_{Im^2}}}_{\sqrt{f(\phi)}} = 1 \tag{A.60}$$

This equation holds, if the phase ϕ is $\phi = \frac{\pi}{2}(2n-1)$ with $n \in \mathbb{Z}$. The solution is determined by plotting the left term of Eq. (A.60) as a function. The function is computed in Fig. A.1 for different discrete $r_{ST,min}$. The phase of the uncertainty term is computed by:

$$\tan\ \theta_{ST} = \left(\frac{Im\left(\frac{1+r_{ST,min}\ e^{j\phi}}{1-r_{ST,min}\ e^{j\phi}}\right)}{Re\left(\frac{1+r_{ST,min}\ e^{j\phi}}{1-r_{ST,min}\ e^{j\phi}}\right)}\right) \tag{A.61}$$

Using the above equations the phase can be computed by:

$$\tan\ \theta_{ST} = \left(\frac{2r_{ST,min}\sin(\phi)}{1-r_{ST,min}^2}\right) \tag{A.62}$$

Considering the solution $\phi = \frac{\pi}{2}(2n-1)$ of the above equation, the subsequent equation holds:

$$\tan\ \theta_{ST} = \left(\frac{2r_{ST,min}}{1-r_{ST,min}^2}\right) \tag{A.63}$$

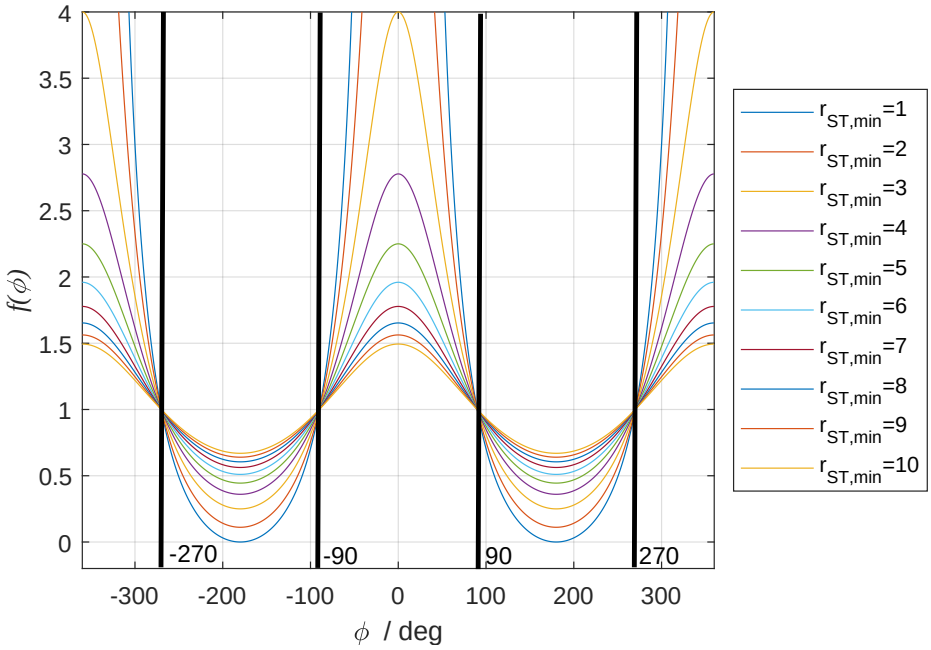


Figure A.1: Function plot $f(\phi)$: if $\phi = \frac{\pi}{2}(2n-1)$, then $f(\phi) = 1$.

Note that the following the trigonometric function holds:

$$\tan(x) = \frac{2\tan(\frac{x}{2})}{1-\tan^2(\frac{x}{2})} \tag{A.64}$$

By comparing Eq. (A.63) and Eq. (A.64), it follows for the phase:

$$\tan\left(\frac{\theta_{ST}}{2}\right) = r_{ST,min} \tag{A.65}$$

$$\boxed{\theta_{ST} = 2\ \text{atan}\left(r_{ST,min}\right)} \tag{A.66}$$

Appendix B

List of Scientific Publications

The subsequent references have been published by the author as research associate at the Institute of Flight System Dynamics at the Technical University of Munich:

- D. Seiferth, J. Diepolder, R. Afonso, F. Holzapfel, "Probabilistc-μ Analysis Using Mapped Uncertainties", in 2021 American Control Conference (ACC), 2021.
- G. García, D. Seiferth, V. Meidinger, D. Dollinger, P. Nagarajan, F. Holzapfel , "Conduction of Mission Task Elements within Simulator Flight Tests for Handling Quality Evaluation of an eVTOL Aircraft" , in Proceedings of the AIAA Scitech 2021 Forum , 2021.
- D. Seiferth, R. Afonso, F. Holzapfel, M. Heller, "Reduced Conservatism Proof of the Balanced MIMO Gain and Phase Margins", in 59th IEEE Conference on Decision and Control (CDC), 2020.
- D. Seiferth, F. Holzapfel, M. Heller, "Evasive Maneuvers of Optionally Piloted Air Vehicles For Three-Dimensional Geofencing", in Proceedings of the AIAA Scitech 2020 Forum, 2020.
- D. Seiferth, M. Landkammer, M. Heller, F. Holzapfel, "Automatic Safe Area Detection In Three-Dimensional Airspace", in Proceedings of the AIAA Aviation 2019 Forum, 2019.
- D. Seiferth, C. Blum, M. Heller, F. Holzapfel, "Distance Determination Between WGS84 Coordinates: An Assessment of Approximation Methods for Reduced Computational Complexity", in Proceedings of the ION 2019 Pacific PNT Meeting, 2019.
- D. Seiferth, B. Grüter, M. Heller, F. Holzapfel, "Fully-Automatic Geofencing Module For Unmanned Air Systems In Two Dimensional Space", in AIAA SciTech Forum, 2019.
- B. Grüter, D. Seiferth, M. Bittner, F. Holzapfel, "Emergency Flight Planning using Voronoi Diagrams", in AIAA SciTech Forum, 2019.

- D. Seiferth, M. E. Kügler, M. Heller, F. Holzapfel, "In-Flight Verification of a model-based designed Ground Controller for an innovative Unmanned Air Vehicle (UAV)", in 2018 Flight Testing Conference, AIAA AVIATION Forum, 2018.
- D. Seiferth, M. Heller, F. Holzapfel, "Automatic Safe Area Detection for novel Unmanned Air Vehicles", in IEEE International Conference on Control, Automation, and Robotics (ICCAR), 2018.
- M. E. Kügler, D. Seiferth, M. Heller, F. Holzapfel, "Real-Time Monitoring of Flight Tests with a Novel Fixed-Wing UAV by Automatic Flight Guidance and Control System Engineers", in IEEE International Conference on Control, Automation, and Robotics (ICCAR), 2018.
- D. Seiferth, R. Kuchar, M. Heller, "Model-based Design and Real Live On-Runway Testing of a Ground Controller for a novel diamond-shaped Unmanned Air Vehicle (UAV)", in 56th IEEE Conference on Decision and Control (CDC), 2017.
- D. Seiferth, M. Heller, "Testing and Performance Enhancement of a model-based designed Ground Controller for a diamond-shaped Umanned Air Vehicle (UAV)", in 2017 IEEE Conference on Control Technology and Applications (CCTA), 2017.
- D. Seiferth, G. Chowdhary, M. Mühlegg, F. Holzapfel, "Online Gaussian Process Regression with non-Gaussian Likelihood", in 2017 American Control Conference (ACC), 2017.

Appendix C

Acronyms

Acronym	Description
ACE	Actuator Control Electronics
ADS	Air Data System
AFCS	Automatic Flight Control System
AHRS	Attitude Heading Reference System
AMC	Acceptable Means of Compliance
ATOL	Automatic Take-Off and Landing
CAD	Computer-aided design
CSAS	Control and Stability Augmentation System
CS-UAS	Certification Specification Document of Unmanned Air Systems
Ctrl Switch	Control Switch on the Flight Management Control Station
DAA	Detect And Avoid
DGPS	Differential Global Positioning System
EASA	European Union Aviation Safety Agency
EP	External Pilot
EPCS	External Pilot Control Station
eVTOL	electrical Vertical Take-Off and Landing aircraft
FCC	Flight Control Computer
FDD	Fault Detection and Diagnosis
FTI	Flight Test Instrumentation
FMCS	Flight Management Control Station
FO	Flight Operator
GC	Ground Controller
GCE	Gear Control Electronics
GM	Gain Margin
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GuM	Guidance Material
HQ	Handling Qualities
IMU	Inertial Measurement Unit
JARUS	Joint Authorities for Rulemaking on Unmanned Systems

Acronym	Description
LFT	Linear Fractional Transformation
LTI	Linear Time-Invariant
MEMS	Microelectromechanical Sensors
MILS	Model-in-the-Loop Simulations
MIMO	Multiple-Input Multiple Output
MOC	Means of Compliance
MTOW	Maximum Take-Off Weight
NAV	Navigation System
NWSA	Nose Wheel Steering Angle
OSO	Operational Safety Objectives
QNH	Q-code for local atmosphere adapted to Mean Sea Level
RC	Resistor-capacitor filter
SAIL	Specific Assurance and Integrity Levels
SDE	Strap-down equation
SORA	Specific Operational Risk Assessment
SISO	Single-Input-Single-Output
PM	Phase Margin
TUM-FSD	Institute of Flight System Dynamics at the Technical University of Munich
UAS	Unmanned Aerial System
UAV	Unmanned Aerial Vehicle
VLOS	Visual Line of Sight
VTOL	Vertical Take-Off and Landing
WC	Worst-Case
WGS84	World Geodetic System 1984

Symbols

General Flight Dynamic Variables

Symbol	Description
B	Body-frame
α	Angle of attack
χ	Course angle of the aerial vehicle
E	Earth centered earth fixed frame
f_x	Measured translational acceleration in x-direction given in B-frame
f_y	Measured translational acceleration in y-direction given in B-frame
f_z	Measured translational acceleration in z-direction given in B-frame
$\boldsymbol{F}$	Three-dimensional force vector
g	Gravity constant
γ	Kinematic flight path angle
$\underline{\boldsymbol{J}}$	Plant inertia tensor
$\boldsymbol{M}$	Three-Dimensional moment vector
θ	Pitch angle
p	Roll rate
ϕ	Bank angle
$\boldsymbol{\Phi}$	Attitude angle vector: $\boldsymbol{\Phi} = \left[\phi, \theta, \psi\right]^T$
Ψ	Heading angle
q	Pitch rate
r	Yaw rate
$\left(u_A^G\right)$	Aerodynamic vel. in x-direction of body-frame B at center of gravity G
$\left(v_A^G\right)$	Aerodynamic vel. in y-direction of body-frame B at center of gravity G
V_A	Absolute aerodynamic velocity
V_K	Absolute kinematic velocity
V_W	Absolute wind velocity
$\left(w_A^G\right)$	Aerodynamic vel. in z-direction of body-frame B at center of gravity G

Geofencing

Symbol	Description
$\boldsymbol{A}$	Flight area array
D	Aerodynamic drag force
h	WGS84 altitude
L	Aerodynamic lift force
M	Number of flight zones
μ	Kinematic bank angle
n_z	Load factor
N_i	Number of vertices of ith flight zone
$\boldsymbol{P}$	Polygon matrix
R_L	Lateral turn radius
s_V	Vertical safety margin
T	Thrust force
$\boldsymbol{V}_i$	Vertex matrix of ith flight zone
$\vec{\boldsymbol{V}}_K$	Kinematic velocity vector
x_{pos}	Aircraft position in x-direction of local cartesian coordinate frame
y_p^*	y-coordinate of the p_yth y_p^*-plane
$\boldsymbol{y}^*$	vector of y_p^*-coordinates
y_{pos}	Aircraft position in y-direction of local cartesian coordinate frame
z_p^*	z-coordinate of the p_zth z_p^*-plane
$\boldsymbol{z}^*$	vector of z_p^*-coordinates
z_{pos}	Aircraft position in z-direction of local cartesian coordinate frame

Stability & Robustness Analysis

Symbol	Description
$\underline{\boldsymbol{A}}$	System matrix of state-space model
$\underline{\boldsymbol{A}}$	Affine transformation matrix in the context of probabilistic-μ
$\underline{\boldsymbol{B}}$	Input matrix of state-space model
$\underline{\boldsymbol{C}}$	Output matrix of state-space model
$\underline{\boldsymbol{C}}_S(s)$	Uncertainty transfer function cons. the sensitivity transfer function
$\underline{\boldsymbol{C}}_{ST}(s)$	Uncertainty transfer func. for der. of balanced MIMO margins (see Eq. (3.93))
c_μ	Reciprocal of μ-value
$c_{\mu,max}$	Reciprocal of μ-value
$\boldsymbol{du}_S(s)$	Disturbance input vector (see Fig. 3.3)
$\boldsymbol{du}_T(s)$	Disturbance input vector (see Fig. 3.4)
$\underline{\boldsymbol{D}}$	Feedthrough matrix of state-space model
Δ	Scalar uncertainty
Δ^R	Scalar real uncertainty
Δ^C	Scalar complex uncertainty
$\boldsymbol{\Delta}$	Vectorization of uncertainty matrix $\boldsymbol{\Delta} = \text{vec}(\underline{\boldsymbol{\Delta}})$
$\tilde{\boldsymbol{\Delta}}$	Vectorization of mapped uncertainty matrix $\tilde{\boldsymbol{\Delta}} = \text{vec}(\underline{\tilde{\boldsymbol{\Delta}}})$
$\underline{\boldsymbol{\Delta}}$	Unstructured or structured uncertainty matrix
$\underline{\tilde{\boldsymbol{\Delta}}}$	Mapped uncertainty matrix
$\underline{\boldsymbol{\Delta}}^{WC}$	Possible worst-case uncertainty matrix, which considers μ-lower bound
$\underline{\underline{\boldsymbol{\Delta}}}$	Set of allowed uncertainties $\underline{\boldsymbol{\Delta}}$
$\underline{\boldsymbol{D}}$	Scaling matrix for μ-upper bound minimization
$\underline{\underline{\boldsymbol{D}}}$	Set of scaling matrix $\underline{\boldsymbol{D}}$ for μ-upper bound minimization
$G(s)$	SISO plant transfer function
$\underline{\boldsymbol{G}}(s)$	MIMO transfer function of the plant
$\underline{\boldsymbol{G}}_P(s)$	MIMO transfer function of perturbed plant model
$K(s)$	SISO controller transfer function
K_Δ	Gain of worst-case transfer function candidate
$\underline{\boldsymbol{K}}(s)$	MIMO transfer function of the controller
$\boldsymbol{r}(s)$	Closed-loop system command vector
λ	Eigenvalue
$l(s)$	Maximum relative error

Symbol	Description
$L_I(s)$	SISO open loop transfer function (input)
$L_O(s)$	SISO open loop transfer function (output)
$\underline{\boldsymbol{L}}_I(s)$	MIMO open loop transfer function (input)
$\underline{\boldsymbol{L}}_O(s)$	MIMO open loop transfer function (output)
μ_Δ	μ-value with regard to stability borderline
μ_Δ^{lb}	μ-lower bound with regard to stability borderline
μ_Δ^{ub}	μ-upper bound with regard to stability borderline
$\mu_{\Delta\Gamma}$	μ-value with regard to conformity borderline
$\underline{\boldsymbol{M}}(s)$	MIMO transfer function resulting from LFT (see Fig. 3.16)
ω	Frequency
ω_{180}	Frequency, at which $L(j\omega)$ crosses the real axis
ω_c	Cross-over frequency
$\varphi(f(s))$	phase of the SISO transfer function $f(s)$
$\boldsymbol{p}^0$	Displacement vector in the context of the probabilistic-μ approach
$\underline{\underline{\boldsymbol{\Pi}}}$	Plant uncertainty set
r_S	Inverse of maximum singular value $\bar{\sigma}(\underline{\boldsymbol{S}})$
$r_{S,min}$	$= \inf_\omega r_S(\omega)$
r_{ST}	Inverse of maximum singular value $\bar{\sigma}(\underline{\boldsymbol{S}} - \underline{\boldsymbol{T}})$
$r_{S,min}$	$= \inf_\omega r_{ST}(\omega)$
r_T	Inverse of maximum singular value $\bar{\sigma}(\underline{\boldsymbol{T}})$
$r_{T,min}$	$= \inf_\omega r_T(\omega)$
ρ	Spectral radius
ρ_C	Complex spectral radius
ρ_R	Real spectral radius
s	Complex frequency parameter used for Laplace-transformation
$S_I(s)$	SISO input sensitivity transfer function
$S_O(s)$	SISO output sensitivity transfer function
$\underline{\boldsymbol{S}}_I(s)$	MIMO input sensitivity transfer function
$\underline{\boldsymbol{S}}_O(s)$	MIMO output sensitivity transfer function
$\underline{\boldsymbol{\Sigma}}$	Variance matrix of multivariate Gaussian distribution
θ_S	Phase margin of sensitivity transfer function
$\bar{\sigma}(\underline{\boldsymbol{f}}(s))$	Maximum singular value of MIMO transfer function $\underline{\boldsymbol{f}}(s)$
θ_{ST}	Phase margin of balanced sensitivity transfer function
θ_T	Phase margin of complementary sensitivity transfer function
$T_I(s)$	SISO input complementary sensitivity transfer function
$T_O(s)$	SISO output complementary sensitivity transfer function
$\underline{\boldsymbol{T}}(s)$	MIMO complementary sensitivity transfer function
$\underline{\boldsymbol{T}}(s)$	Transformation matrix in the context of the probabilistic-μ approach
$\underline{\boldsymbol{T}}_I(s)$	MIMO input complementary sensitivity transfer function
$\underline{\boldsymbol{T}}_O(s)$	MIMO output complementary sensitivity transfer function
$\underline{\boldsymbol{U}}$	Scaling matrix for μ-lower bound maximization
$\underline{\underline{\boldsymbol{U}}}$	Set of scaling matrix $\underline{\boldsymbol{U}}$ for μ-lower bound maximization
$\upsilon_{\Delta 1}$	SISO all-pass transfer function
$\upsilon_{\Delta 2}$	SISO *DT2* transfer function
Υ_Δ	SISO worst-case transfer function candidate w.r.t. stability borderline

Symbol	Description
Υ_Γ	SISO worst-case transfer function candidate w.r.t. stability borderline
$\boldsymbol{u}(s)$	Plant input vector
$w(s)$	SISO weighting transfer function
$\underline{\boldsymbol{W}}(s)$	MIMO weighting transfer function
$\boldsymbol{y}(s)$	Plant output vector

Application eVTOL

Symbol	Description
$\underline{\boldsymbol{C}}(s)$	MIMO transfer function of controller algorithm
$\underline{\boldsymbol{CA}}$	Linearized control allocation
$\underline{\boldsymbol{E}}(s)$	MIMO estimator transfer function
$\underline{\boldsymbol{F}}(s)$	MIMO transfer function of the command filter in control law
$\underline{\boldsymbol{G}}_{act}(s)$	MIMO actuator transfer function
$\underline{\boldsymbol{G}}_{SDE}$	Strap-down equation matrix: $\dot{\boldsymbol{\Phi}} = \underline{\boldsymbol{G}}_{SDE}(\boldsymbol{\Phi})\boldsymbol{\omega}_{pqr}$
$\underline{\boldsymbol{G}}_{sen}(s)$	MIMO sensor transfer function
$\underline{\boldsymbol{M}}_{XY}$	Transformation matrix from the coordinate frame X to Y
ω_{L01}	Rotational speed of left front hover propeller of considered eVTOL
$\boldsymbol{\omega}$	Hover-propeller turn rate vector
$\boldsymbol{\omega}_{fil}$	Estimated propeller turn rate vector
$\boldsymbol{\omega}_{pqr}$	Angular rate vector: $\boldsymbol{\omega}_{pqr} = \left[p,q,r\right]^T$
$\boldsymbol{u}$	Squared propeller turn rate vector
$\boldsymbol{u}_{fil}$	Estimated squared propeller turn rate vector

SAGITTA

Symbol	Description
δ_{cmd}	Nose wheel steering angle command
δ_T	Normalized engine thrust
Δ_S	Split flaps deflection
η	Elevator deflection
Ψ_{cmd}	Commanded heading direction
Ψ_{RWY}	Runway direction
r_{cmd}	Commanded yaw rate
x_{RWY}	Runway position in x-direction (runway direction) of local runway frame
y_{cmd}	Commanded lateral position with regard to the runway centerline
y_{meas}	Measured lateral position with regard to the runway centerline

List of Figures

1.1 Categories of UAS operations (based on Ref. [16]) 5
1.2 Thesis contributions in context of new technologies, regulations and commercial airspace 8
1.3 Overview and structure of the geofencing algorithm 9
1.4 Overview and structure of the stability and robustness analysis for an UAM vehicle 13
1.5 Comprehensive overview of the necessary steps for first flight qualification . . 18

2.1 Valid and invalid defined flight areas 26
2.2 Example of generated flight area and safety area 28
2.3 Issues of default offset polygon generation 31
2.4 Offset polygon generation 33
2.5 Vertex creation for offset polygon generation 34
2.6 Difficulties in safety area generation for angles $\alpha \leq$180 deg 37
2.7 Difficulties in safety area generation for angles $\alpha >$180 deg 37
2.8 Illustration of safety margin checking algorithm 38
2.9 Illustration of point-in-polygon algorithms: X_P, Y_P represent the coordinate of the aircraft position (red dot), X_1, Y_1 and X_2, Y_2 are coordinates of start and end point of a line. 41
2.10 Appropriate online selection of turn direction 42
2.11 Verification of geofencing algorithm in model-in-the-loop simulation 43
2.12 Exemplary air space structure of German Air Traffic Control Authority (image provided by Ref. [73]) 45
2.13 Conceptional flight area examples in three-dimensional space with current calculated xy-plane at current aircraft (red circle) altitude z 45
2.14 Exemplary flight area with two non-flight zones and with current computed xy- and xz-plane at current aircraft (red circle) position 46
2.15 Examples of flight zone geometries 46
2.16 Example of flight zone definition by the vertex matrix for z-level discretization 48
2.17 Example of flight zone definition by the vertex matrix for y-level discretization 49

2.18 Introduction of xz- and xy-planes for an exemplary zone with the shape of an octahedron . . . 51
2.19 Case illustration, which points are considered for intersection point calculation 52
2.20 Polgyon matrix for illustrated flight area . . . 53
2.21 Generated two-dimensional safety areas within the introduced xz- and xy-planes 55
2.22 Example of two-dimensional polygon definition at the aircraft position z_{pos} . . 57
2.23 Model-in-the-loop verification: outside (magenta), inside (cyan) . . . 58
2.24 Design of the lateral maneuver . . . 64
2.25 Design of the vertical maneuver . . . 67
2.26 Illustration of vertical turn radius R_V and necessary vertical margin s_V in dependence of γ under consideration of the velocity V_K =65 m/s and $|\Delta n_{des,V}| = 1$ 67
2.27 Illustration of combined pull-up and lateral maneuver . . . 69
2.28 Autopilot structure within model-in-the-loop environment: x_{WP}, y_{WP} represent waypoint coordinates in local cartesian frame . . . 70
2.29 Conducted lateral maneuver in model-in-the-loop simulations . . . 72
2.30 Conducted vertical maneuver in model-in-the-loop simulations . . . 74
2.31 Flight path of simulated combined vertical and lateral maneuver . . . 76
2.32 Simulation results of a combined maneuver . . . 77

3.1 Robustness and handling quality requirements . . . 82
3.2 Considered MIMO closed-loop system with signal dimensions n and m . . . 85
3.3 Block diagram for the derivation of the input and output sensitivity transfer functions $\underline{\boldsymbol{S}}_I$ and $\underline{\boldsymbol{S}}_O$ . . . 85
3.4 Block diagram for the derivation of the input and output complementary sensitivity transfer functions $\underline{\boldsymbol{T}}_I$ and $\underline{\boldsymbol{T}}_O$ . . . 85
3.5 Illustrative MIMO Closed-loop system . . . 87
3.6 Exemplary transformation procedure for the calculation of a SISO open-loop transfer function $L(s)$ after a *Bottleneck-Cut* at the first plant input channel . . 88
3.7 Real perturbations Δ^R and complex perturbation Δ^C with radius $|w_I(j\omega)| = 1$ and uniformly distributed samples . . . 93
3.8 Truncated and unbounded stochastic uncertainties in comparison . . . 94
3.9 Different forms of uncertainties with weighting matrices . . . 97
3.10 Nyquist plot of uncertain actuator model . . . 103
3.11 Design of weighting function $w_I(s)$ . . . 104
3.12 Exemplary probability density functions with limited and unlimited support . . 106
3.13 Illustrative SISO control configuration . . . 111
3.14 Qualitative illustration of Nichols diamond derivation based on Ref. [87, p.42ff.] 112
3.15 Small Gain Theorem (Ref. [42]) . . . 114
3.16 Linear Fractional Transformation . . . 116

3.17 MIMO gain and phase margin derivation of the complementary sensitivity function $\underline{\boldsymbol{T}}_I$ 120
3.18 Illustration of the term $1+\Delta_i$, if it changes in gain and/or phase; the dashed red circle represents the term $e^{j\theta_i}$, the blue circle illustrates the term $1+\Delta_i$, the angle θ_i shows the phase, which is added to the system, if the uncertainty causes phase only, the grey circles on the right show different-sized uncertainties, depending on $r_{T,i}$ 121
3.19 Geometrical interpretation for calculating the MIMO phase margin 121
3.20 MIMO gain and phase margin of the sensitivity function $\underline{\boldsymbol{S}}_I$ 125
3.21 MIMO gain and phase margin derivation of the balanced sensitivity function . 128
3.22 Illustrative results for closed-loop system with plant and PI-controller 133
3.23 Illustration of MIMO phase margin proof for reduced conservatism of balanced sensitivity transfer function 135
3.25 Normalization of uncertainty matrix $\underline{\boldsymbol{\Delta}}$ by matrix $\underline{\boldsymbol{W}}$ (same structure as $\underline{\boldsymbol{\Delta}}$), such that $\underline{\boldsymbol{\Delta}} = \underline{\boldsymbol{\Delta}}^* \underline{\boldsymbol{W}}$ with $\bar{\sigma}(\underline{\boldsymbol{\Delta}}^*) \leq 1$ 138
3.26 μ-upper bound minimization by introducing the scaling matrix $\underline{\boldsymbol{D}}$ 143
3.27 Exemplary conformity borderline $\Gamma(\omega)$ (reworked from (Ref. [90, p.92]) 145
3.28 Bode diagram of the *all-pass* (AP) transfer function $\upsilon_{\Delta 1}(s)$ and the *DT2*-element $\upsilon_{\Delta 2}(s)$ 151
3.29 Bode diagram of the worst-case transfer function candidate $\Gamma_\Delta(s)$ (red), which consists of the second-order *all-pass* (AP) $\upsilon_{\Delta 1}(s)$, the *DT2*-element $\upsilon_{\Delta 2}(s)$ and the gain K 151
3.30 Structure of exemplary closed-loop system with one-dimensional multiplicative uncertainty $\Delta(s)$ and its weight $W(s)$ 154
3.31 Comparison between closed-loop system including worst-case transfer function candidate with regard to the stability borderline (left, imaginary axis) and specified conformity borderline (right) 157
3.32 Verification plots of worst-case transfer function candidates with regard to the stability borderline (left) and the conformity borderline (right) 158
3.33 Uncertainties with limited intervals and probabilistic uncertainties 159
3.34 Upper bound μ^{ub} of an exemplary closed-loop system in dependence of weighting factors for the normalized uncertainties $\Delta_1, \Delta_2, \Delta_3$ 160
3.35 Illustration of the transformations in the proposed probabilistic-μ approach . . 161
3.36 Possible mapped uncertainty hyperspace transformations 164
3.37 Structure of the closed-loop system 167
3.38 Two-dimensional uncertainty space with three different guaranteed-stable uncertainty sets 169

3.39 Three-dimensional uncertainty space with three different guaranteed-stable uncertainty sets 169
3.40 Sketch of eVTOL demonstrator aircraft 171
3.41 Calculated trim angles over the sidewards velocity $(v_A^G)_C$ and forward velocity $(u_A^G)_C$ grid 176
3.42 Maximum eigen value of flight dynamics over the velocity envelope considering the full aerial vehicle's motion 181
3.43 Maximum eigen value of the longitudinal and lateral flight dynamics over the velocity envelope 181
3.44 Pole shift for **forward** (left column) and **backward** (right column) movements (see forward/backward velocity in colorbar, denoted in m/s) under consideration of the full, longitudinal and lateral motion in comparison to hover (red) . . 182
3.45 Pole shift for different **sidewards** movements (see sidewards velocity in colorbar, denoted in m/s) under consideration of the full, longitudinal and lateral motion in comparison to hover (red) 183
3.46 Fitted stochastic uncertainties from the nonlinear flight dynamic model within the plant system matrix $\underline{\boldsymbol{A}}$ at the trim point $(u_A^R)_C$ =0, $(v_A^R)_C = -4$ m/s 187
3.47 Fitted stochastic uncertainties from the nonlinear flight dynamic model within the plant input matrix $\underline{\boldsymbol{B}}$ at the trim point $(u_A^R)_C$ =0, $(v_A^R)_C = -4$ m/s 188
3.48 Linear closed-Loop system with detailed control law 190
3.49 INDI controller algorithm $\underline{\boldsymbol{C}}(s)$ without illustrated reference models 190
3.50 Command filter $\underline{\boldsymbol{F}}(s)$ with INDI approach 194
3.51 Verification of linear with nonlinear closed-loop system with three different step inputs in the **roll command channel** ϕ_{cmd}: linear model (red), nonlinear model (blue) 197
3.52 Verification of linear with nonlinear closed-loop system with three different step inputs in the **pitch command channel** θ_{cmd}: linear model (red), nonlinear model (blue) 198
3.53 Verification of linear with nonlinear closed-loop system with three different step inputs in the **yaw command channel** $\dot{\psi}_{cmd}$: linear model (red), nonlinear model (blue) 199
3.54 Selection of Analysis Points for SISO and MIMO margins 200
3.55 Nichols plots over envelope with sidewards velocity up to 4 m/s and forward/backward velocity between -4 m/s and 14 m/s considering *bottleneck-cut* before control allocation: Lines are marked, if nichols diamond is cut (red) . . . 203
3.56 Nichols plots over envelope with sidewards velocity up to 4 m/s and forward/backward velocity between -4 m/s and 14 m/s considering *bottleneck-cut* before sensors: Lines are marked, if nichols diamond is cut (red) 203

3.57 Nichols plots over envelope with sidewards velocity up to 4 m/s and forward/ backward velocity between -4 m/s and 14 m/s considering *bottleneck-cut* at control allocation. 204

3.58 SISO gain and phase margin before control allocation 205

3.59 SISO gain and phase margin at sensor AP 206

3.60 SISO gain (left side) and phase margin (right side) results at actuator AP over horizontal velocities 207

3.61 Horizontal flight envelope under consideration of nichols plots at different analysis points: Nichols diamond is cut in at least one respective channel of AP (red), Nichols diamond is not cut in all respective channels of AP (green), considered trim point (+) 210

3.62 Singular value plots at control allocation analysis point with varying sidewards velocity from $\left(v_A^G\right)_C^E = -6$ m/s (line with highest peak) to $\left(v_A^G\right)_C^E = 0$ m/s (line with lowest peak) 211

3.63 MIMO gain and phase margin over horizontal velocity grid at **control allocation analysis point**: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg 213

3.64 MIMO gain and phase margin over horizontal velocity grid at **sensor analysis point**: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg 214

3.65 MIMO gain and phase margin over horizontal velocity grid at **actuator analysis point**: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg 215

3.66 Lowest conservative MIMO gain and phase margins (i.e. maximal upper GM, minimal lower GM, maximal PM) at different analysis points (AP) over horizontal aerodynamic velocity envelope: green and blue areas fulfill |GM|>3 dB and |PM|>30 deg 216

3.67 Comparison of MIMO gain and phase margins at different analysis points: only contour lines are shown which are relevant in respect of the requirement |GM|>3 dB and |PM|>30 deg 217

3.68 MIMO phase margin of $\underline{\boldsymbol{T}}, \underline{\boldsymbol{S}}, \underline{\boldsymbol{S}} - \underline{\boldsymbol{T}}$ comparison at control allocation analysis point over the horizontal flight envelope 219

3.69 Weight approximation of eVTOL propeller actuator uncertainty 222

3.70 Closed-loop system behavior using computed worst-case uncertainties under consideration of real flight dynamic model uncertainties and complex actuator uncertainties and the **ADS-33 HQ Level 2 conformity borderline (see (b))** at the trim point $\left(u_A^R\right)_C = 0$, $\left(v_A^R\right)_C = -4$ m/s. 225

3.71 Closed-loop system behavior using computed worst-case uncertainties under consideration of real flight dynamic model uncertainties and complex actuator uncertainties and the author-defined **higher performance HQ conformity borderline (see (b))** at the trim point $(u_A^R)_C$ =0, $(v_A^R)_C = -4$ m/s. 226
3.72 Results of advanced μ_Γ-analysis over the frequency under consideration of ADS-33 Handling Quality Level 2 228
3.73 Results of advanced μ_Γ-analysis and probabilistic μ-analysis without probability maximization over horizontal aerodynamic velocity envelope under consideration of ADS-33 Handling Quality Level 2 229
3.74 Comparison of nichols diamond requirement, possible MIMO margin requirements and the $\mu_\Gamma = 1$ boundary within the horizontal flight envelope of the considered eVTOL 232
3.75 *Surrogate* optimization plots to maximize the lower bound of the probability to achieve ADS-33 HQ Level 2 conformity under consideration of the full uncertain linear closed-loop system with 78 stochastic, Gaussian-distributed uncertainties at selected trim points. Note that the optimization is set up as a minimization problem why the objective function is here equal to the negative probability to achieve conformity. 239
3.76 Under consideration of μ^{lb}-results, the probability can be calculated, which definitely brings the system to instability/ non-conformity. 242

4.1 Illustration of the SAGITTA demonstrator 245
4.2 System Architecture of the SAGITTA demonstrator (based on Ref. [117]) . . . 246
4.3 On-ground External Pilot Command Mapping implemented in *Mathworks MATLAB/ Simulink* 248
4.4 Architecture of Flight Control Software (reworked from Ref. [123]) 250
4.5 Illustration of the SAGITTA landing gear in reality and simulation 253
4.6 Modeled vertical and lateral dynamics of the SAGITTA landing gear model (both figures from R. Kuchar, DLR, in Ref. [115]) 254
4.7 SAGITTA ground controller structure 261
4.8 SAGITTA ground controller moding for take-off 264
4.9 Ground controller derotation mode at touch down 265
4.10 SAGITTA ground controller moding for landing 267
4.11 Gain set selection via recycling the heading control within the taxi test software 269
4.12 SAGITTA on-ground simulation without disturbances using three different gain sets (see Tab. 4.3) 274
4.13 SAGITTA on-ground simulation with 5 kts crosswind using three different gain sets (see Tab. 4.3) 275
4.14 Magnetometer calibration of the SAGITTA demonstrator 276

4.15 SAGITTA demonstrator during towing tests 280
4.16 Navigation data of towing test for magnetometer verification 283
4.17 Monitoring display for online gain-set layout of the ground controller 284
4.18 EP controlled straight acceleration run up to 25 kts 288
4.19 Manual single lane change maneuver by external pilot 290
4.20 EP conducted steady state cornering maneuver at 15 kts 292
4.21 EP controlled wiggly lines 294
4.22 Illustration of the SAGITTA nose wheel castor and nose wheel steering actuator 295
4.23 Offset command test case with gain set number 1 (initial) at 35 kts 300
4.24 Double lane change test case with gain set number 2 (final gain set) at 35 kts 301
4.25 Single lane change test case with *No GPS* trigger using gain set number 2 (final) at 50 kts 302
4.26 High speed taxi test with final gain set at 66 kts 303
4.27 Double lane change maneuver test case with greater excitation 306
4.28 High speed straight acceleration run for first flight qualification 307
4.29 The SAGITTA demonstrator during aircraft-in-the-loop simulations 308
4.30 Track of the SAGITTA demonstrator during towing tests at the first flight aerodrome (created with Ref. [145]) 309
4.31 Automatic take-off with EP controlled take-off abort on test-site 310
4.32 Simulated and real-live first flight mission of the SAGITTA demonstrator (Ref. [54] 312
4.33 Quick look terminals for monitoring of flight control system 313
4.34 The SAGITTA demonstrator at the end of its first flight mission [Airbus D&S] . 316
4.35 Take-off run at first flight of the SAGITTA demonstrator - Part 1 318
4.36 Take-off run at first flight of the SAGITTA demonstrator - Part 2 319
4.37 Take-off run at first flight of the SAGITTA demonstrator - Part 3 320
4.38 Automatic landing at first flight of the SAGITTA demonstrator - Part 1 325
4.39 Automatic landing at first flight of the SAGITTA demonstrator - Part 2 326
4.40 Automatic landing at first flight of the SAGITTA demonstrator - Part 3 327

A.1 Function plot $f(\phi)$: if $\phi = \frac{\pi}{2}(2n-1)$, then $f(\phi) = 1$. IX

List of Tables

3.1 Parametric uncertainties within actuator model 101
3.2 Parameter values of the open-loop plant, the controller, the actuator $G_{act}(s)$ and its uncertainty weight $W(s)$ (actuator and weight values are based on the work in Ref. [90, 89]) 155
3.3 Parameter values of the open-loop plant, the actuator and the controller . . . 166
3.4 Probability comparison between the novel probabilistic-μ approach and the direct Monte Carlo (MC) method 168
3.5 List of considered uncertainties in μ-analysis with real and complex uncertainties 224
3.6 Definite lower bound of the probability to achieve conformity without probability maximization and with probability maximization (i.e. with outer optimization of the bi-level optimization problem) for selected trim points. 236

4.1 Operational Elements of the Flight Operator on the FMCS (Ref. [114]) 247
4.2 Typical test procedure for the conduction of a test case with offset command at dedicated velocity. 271
4.3 As shown in Fig. 4.12 and 4.13, regarding the gain layout a trade-off between response quickness, tracking performance, stability characteristics and robustness has to be considered. 273

Bibliography

[1] SESAR Joint Undertaking. *European drones outlook study - Unlocking the value for Europe*. 2016. URL: https://www.sesarju.eu/sites/default/files/documents/reports/European_Drones_Outlook_Study_2016.pdf (visited on 10/05/2020).

[2] C. Reiche et al. *Urban Air Mobility Market Study*. National Aeronautics and Space Administration (NASA). 2018. URL: http://dx.doi.org/10.7922/G2ZS2TRG (visited on 01/29/2021).

[3] Blanca de Miguel Molina and Marival Segarra Oña. "The Drone Sector in Europe". In: *Ethics and Civil Drones: European Policies and Proposals for the Industry*. Ed. by María de Miguel Molina and Virginia Santamarina Campos. Cham: Springer International Publishing, 2018, pp. 7–33. ISBN: 978-3-319-71087-7. DOI: 10.1007/978-3-319-71087-7_2. URL: https://doi.org/10.1007/978-3-319-71087-7_2.

[4] NASA. *Urban Air Mobility (UAM) Market Study*. 2018. URL: https://www.nasa.gov/sites/default/files/atoms/files/uam-market-study-executive-summary-v2.pdf (visited on 10/05/2020).

[5] Morten Svihus from FLIR Systems. *Use of Nano UAV in non-military operations.* Paris, France: Presented at 6th Annual International Conference RPAS European Civil RPAS Operators & Operations Forum, 2018.

[6] Quantum-Systems GmbH. *Trinity F90+*. URL: https://www.quantum-systems.com/project/trinity-f90/ (visited on 01/24/2021).

[7] EMT Ingenieurgesellschaft. *LUNA - Luftaufklärungs- und Überwachungssystem*. URL: https://www.emt-penzberg.de/wp-content/uploads/2020/06/LUNA_de_02_EInzelseiten.pdf (visited on 01/24/2021).

[8] Wisk. *Transport trial to help Cora take off*. 2020. URL: https://wisk.aero/blog/transport-trial-to-help-cora-take-off/ (visited on 10/05/2020).

[9] G. Schmidt. "INS/GPS Technology Trends". In: NATO RTO Lecture Series, RTO-EN-SET-116, 2010.

[10] Christophe Adrados et al. "Global Positioning System (GPS) location accuracy improvement due to Selective Availability removal". In: *Comptes Rendus Biologies* 325.2 (2002), pp. 165 –170. ISSN: 1631-0691. DOI: https://doi.org/10.1016/S1631-0691(02)01414-2. URL: http://www.sciencedirect.com/science/article/pii/S1631069102014142.

[11] GIS Geography. *GPS Accuracy: HDOP, PDOP, GDOP, Multipath and the Atmosphere*. [Online; accessed 21-June-2020]. 2019. URL: https://gisgeography.com/gps-accuracy-hdop-pdop-gdop-multipath/.

[12] William J. Hughes Technical Center and WAAS T & E Team. *Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report*. [Online; accessed 21-June-2020]. 2016. URL: https://www.nstb.tc.faa.gov/reports/PAN96_0117.pdf#page=22.

[13] Intel Corporation. *The Evolution of a Revolution*. URL: https://download.intel.com/pressroom/kits/IntelProcessorHistory.pdf (visited on 01/25/2021).

[14] Cynthia A. Lundgren and Kang Xu and T. Richard Jow and Jan Allen and Sheng S. Zhang. "Part D Energy Conversion and Storage". In: *Handbook of Electrochemical Energy*. Berlin, Heidelberg: Springer, 2017, 451. DOI: {10.1007/978-3-662-46657-5}.

[15] Anubhav Datta. "Lecture notes for the workshop Principles of Electric-VTOL". In: University of Maryland. Dallas, USA: AIAA Aviation Forum 2019, 2019.

[16] European Aviation Safety Agency. *Notice of Proposed Amendment 2017-05 (B) - Introduction of a regulatory framework for the operation of drones*. 2017. URL: https://www.easa.europa.eu/sites/default/files/dfu/NPA%202017-05%20%28B%29.pdf (visited on 01/25/2021).

[17] European Aviation Safety Agency. *Advance Notice of Proposed Amendment (2015-10). Introduction of a regulatory framework for the operation of drones*. 2015. URL: https://www.easa.europa.eu/sites/default/files/dfu/A-NPA%202015-10.pdf (visited on 01/25/2021).

[18] European Aviation Safety Agency. *Acceptable Means of Compliance(AMC) and Guidance Material (GM) to Commission Implementing Regulation (EU) 2019/947*. 2019. URL: https://www.easa.europa.eu/sites/default/files/dfu/AMC%20%26%20GM%20to%20Commission%20Implementing%20Regulation%20%28EU%29%202019-947%20%E2%80%94%20Issue%201.pdf (visited on 01/25/2021).

[19] Joint Authorities for Rulemaking of Unmanned Systems. *JARUS guidelines on Specific Operatios Risk Assessment (SORA)*. [Online; accessed 25-Jan-2021]. 2019. URL: http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_06_jarus_sora_v2.0.pdf.

[20] Joint Authorities for Rulemaking of Unmanned Systems. *JARUS guidelines on SORA Annex C. Strategic Mitigation Collision Risk Assessment*. [Online; accessed 25-Jan-2021]. 2019. URL: http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_06_jarus_sora_annex_c_v1.0.pdf.

[21] Joint Authorities for Rulemaking of Unmanned Systems. *JARUS guidelines on SORA Annex D. Tactical Mitigation Collision Risk Assessment*. [Online; accessed 25-Jan-2021]. 2019. URL: http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_06_jarus_sora_annex_d_v1.0.pdf.

[22] European Aviation Safety Agency. *Proposed Special Condition for Light UAS*. 2019. URL: https://www.easa.europa.eu/sites/default/files/dfu/special_condition_light_uas.pdf (visited on 01/25/2021).

[23] European Aviation Safety Agency. *Special Condition for small-category VTOL aircraft*. 2019. URL: https://www.easa.europa.eu/sites/default/files/dfu/SC-VTOL-01.pdf (visited on 01/25/2021).

[24] European Aviation Safety Agency (EASA). *Proposed Means of Compliance with the Special Condition VTOL*. 2020. URL: https://www.easa.europa.eu/sites/default/files/dfu/proposed_moc_sc_vtol_issue_1.pdf (visited on 02/02/2021).

[25] European Aviation Safety Agency. *Concept of Operations for Drones – A risk based approach to regulation of unmanned aircraft*. [Online; accessed 1-May-2017]. 2017. URL: https://www.easa.europa.eu/system/files/dfu/204696_EASA_concept_drone_brochure_web.pdf.

[26] DJI. *DJI Phantom 4 User Manual*. Shenzhen, Guangdong, China, 2016. URL: https://dl.djicdn.com/downloads/phantom_4_pro/Phantom+4+Pro+Pro+Plus+User+Manual+v1.0.pdf (visited on 01/06/2020).

[27] Mia Stevens and Ella Atkins. “Multi-Mode Guidance for an Independent Multicopter Geofencing System”. In: June 2016. DOI: 10.2514/6.2016-3150.

[28] P.L. Pratyusha and V. P. S. Naidu. “Geo-Fencing for Unmanned Aerial Vehicle”. In: *International Journal of Computer Applications* (2015).

[29] Mia N. Stevens and Ella M. Atkins. “Layered Geofences in Complex Airspace Environments”. In: *2018 Aviation Technology, Integration, and Operations Conference*. 2018. DOI: 10.2514/6.2018-3348. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2018-3348.

[30] M. N. Stevens, H. Rastgoftar, and E. M. Atkins. “Specification and evaluation of geofence boundary violation detection algorithms”. In: *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. 2017, pp. 1588–1596. DOI: 10.1109/ICUAS.2017.7991472.

[31] Mia N. Stevens, Hossein Rastgoftar, and Ella M. Atkins. “Geofence Boundary Violation Detection in 3D Using Triangle Weight Characterization with Adjacency”. In: *Journal of Intelligent and Robotic Systems*. 2018. URL: https://doi.org/10.1007/s10846-018-0930-5.

[32] William Bollay. “Aerodynamic Stability and Automatic Control: The Fourteenth Wright Brothers Lecture”. In: *Journal of the Aeronautical Sciences* 18.9 (1951), pp. 569–617. DOI: 10.2514/8.2050. eprint: https://doi.org/10.2514/8.2050. URL: https://doi.org/10.2514/8.2050.

[33] R.C. Nelson. *Flight Stability and Automatic Control*. Aerospace Science & Technology. WCB/McGraw Hill, 1998. ISBN: 9780071158381.

[34] James D. Blight, R. Lane Dailey, and Dagfinn Gangsaas. “Practical control law design for aircraft using multivariable techniques”. In: *International Journal of Control* 59.1 (1994), pp. 93–137.

[35] Honeywell Technology Center et al. *Application of Multivariable Control Theory to Aircraft Control Laws: Final Report : Mulitvariable Control Design Guidelines*. NASA Center for AeroSpace Information. URL: https://books.google.de/books?id=MZ_eNQAACAAJ.

[36] E. Lavretsky and K. Wise. “Robust and Adaptive Control: With Aerospace Applications”. In: 2012.

[37] M. G. Michailidis et al. “Robust nonlinear control of the longitudinal flight dynamics of a circulation control fixed wing UAV”. In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 3920–3927.

[38] Matthias Heller. “Untersuchung zur Steuerung und Robusten Regelung der Seitenbewegung von Hyperschall-Flugzeugen”. Dissertation. München: Technische Universität München, 1999.

[39] Stanislav Braun. “Structurally Constrained Lateral Control of a Diamond-Shaped Aircraft”. Dissertation. München: Technische Universität München, 2019.

[40] Michael Haimerl. “Robust Flight Control Concept for a Hybrid Unmanned Aerial Vehicle”. Dissertation. Aachen: RWTH Aachen, 2019.

[41] Venkata Sravan Akkinapalli and Florian Holzapfel. “Incremental Dynamic Inversion based Velocity Tracking Controller for a Multicopter System”. In: *2018 AIAA Guidance, Navigation, and Control Conference*. DOI: 10.2514/6.2018-1345. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-1345. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2018-1345.

[42] R.L. Dailey. “Lecture notes for the workshop on H_∞ and μ for robust control”. In: *American Control Conference*. San Diego, USA, 1990.

[43] R.L. Dailey. “Lecture notes for the workshop on H_∞ and μ for robust control”. In: *IEEE Conference on Decision and Control*. Brighton, United Kingdom, 1991.

[44] Peter Seiler, Andrew Packard, and Pascal Gahinet. “An Introduction to Disk Margins”. In: *IEEE Control Systems Magazine (forthcoming)*. URL: https://arxiv.org/abs/2003.04771.

[45] P. M. Young, M. P. Newlin, and J. C. Doyle. “μ Analysis with real parametric uncertainty”. In: *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*. 1991, 1251–1256 vol.2. DOI: 10.1109/CDC.1991.261579.

[46] P. M. Young and J. C. Doyle. “Computation of μ with real and complex uncertainties”. In: *29th IEEE Conference on Decision and Control*. 1990, 1230–1235 vol.3. DOI: 10.1109/CDC.1990.203804.

[47] J. Doyle. “Analysis of feedback systems with structured uncertainties”. In: 129.6 (1982), pp. 242–250. ISSN: 0143-7054. DOI: 10.1049/ip-d.1982.0053.

[48] Christian David Heise. “Survivable Flight Control with Guaranteed Stability and Performance Characteristics”. Dissertation. München: Technische Universität München, 2017.

[49] Xiaoyun Zhu, Yun Huang, and J. Doyle. “Soft vs. hard bounds in probabilistic robustness analysis”. In: *Proceedings of 35th IEEE Conference on Decision and Control*. Vol. 3. 1996, 3412–3417 vol.3.

[50] S. Khatri and P. A. Parrilo. “Guaranteed bounds for probabilistic /spl mu/”. In: *Proceedings of the 37th IEEE Conference on Decision and Control (Cat. No.98CH36171)*. Vol. 3. 1998, 3349–3354 vol.3.

[51] Alexandre Falcoz, Daniel Alazard, and Christelle Pittet. “Probabilistic μ-analysis for system performances assessment”. In: *IFAC-PapersOnLine* 50.1 (2017). 20th IFAC World Congress, pp. 399 –404. ISSN: 2405-8963. DOI: https://doi.org/10.1016/j.ifacol.2017.08.181.

[52] S. Thai, C. Roos, and J. Biannic. “Probabilistic μ-Analysis for Stability and $\mathscr{H}_\infty$ Performance Verification”. In: *2019 American Control Conference (ACC)*. 2019, pp. 3099–3104.

[53] David Seiferth et al. “Probabilistic μ-Analysis using Mapped Uncertainties”. In: *American Control Conference*. New Orleans, USA, 2021.

[54] Martin E. Kügler. “Robust Automation of Take-off and Landing - Demonstrated on a Medium-Sized Unmanned Aircraft”. Dissertation. München: Technische Universität München, 2019.

[55] British Broadcasting Corporation (BBC). *US unmanned drone jet makes first carrier landing*. 2013. URL: https://www.bbc.com/news/23276968 (visited on 01/25/2021).

[56] Christoph Krause. “Safe and Robust Automation of Aircraft and System Operation”. Dissertation. München: Technische Universität München, 2021.

[57] M. Hiller D. Schramm and R. Bardini. *Vehicle Dynamics - Modeling and Simulation*. Berlin Heidelberg: Springer, 2014.

[58] E. Sadien et al. “Control design oriented modeling of an on-ground aircraft”. In: *2018 European Control Conference (ECC)*. 2018, pp. 2757–2762.

[59] D. Lemay et al. “Gain-Scheduled Yaw Control for Aircraft Ground Taxiing”. In: *IFAC Proceedings Volumes* 44.1 (2011). 18th IFAC World Congress, pp. 12970 –12975. ISSN: 1474-6670. DOI: https://doi.org/10.3182/20110828-6-IT-1002.00692.

[60] David Mark Lower. “Unique Aspects of Unmanned Aerial Vehicle Testing”. Master’s Thesis. University of Tennessee,Knoxville, 2004.

[61] David Seiferth et al. “Fully-Automatic Geofencing Module for Unmanned Air Systems In Two Dimensional Space”. In: *AIAA Scitech 2019 Forum*. 2019. DOI: 10.2514/6.2019-2078. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2019-2078.

[62] D. Seiferth et al. “Distance Determination Between WGS84 Coordinates- An Assessment Of Approximation Methods For Reduced Computational Complexity”. In: *ION’s Pacific PNT Conference (submitted)*. 2019.

[63] Joseph O’Rourke. *Computational Geometry in C*. 2nd ed. Cambridge University Press, 1998. DOI: 10.1017/CBO9780511804120.

[64] M. I. Shamos and D. Hoey. “Geometric Intersection Problems”. In: *17th Annual Symposium on Foundations of Computer Science*. 1976, pp. 208–215.

[65] Jon Bentley and Thomas Ottman. “Algorithms for Reporting and Countic Geometric Intersections”. In: *IEEE Trans. Computers*. 1979, pp. 643–647.

[66] I.J. Balaban. “An Optimal Algorithm for Finding Segment Intersections”. In: *Proc. 11-th Ann. ACM Sympos. Comp. Geom.* 1995, pp. 211–219.

[67] Joint Authorities for Rulemaking of Unmanned Systems. *JARUS CS-UAS Recommendations for Certification Specification for Unmanned Aircraft Systems*. [Online; accessed 07-Jan-2020]. 2019. URL: http://jarus-rpas.org/sites/jarus-rpas.org/files/jar_doc_16_cs_uas_edition1.0.pdf.

[68] B. Grüter et al. “Emergency Flight Planning using Voronoi Diagrams”. In: *2019 AIAA Scitech Forum*. 2019.

[69] S. P. Schatz and F. Holzapfel. “Modular trajectory / path following controller using nonlinear error dynamics”. In: *IEEE International Conference on Aerospace Electronics and Remote Sensing Technology*. 2014, pp. 157–163.

[70] V. Schneider, N. C. Mumm, and F. Holzapfel. “Trajectory generation for an integrated mission management system”. In: *IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*. 2015, pp. 1–7.

[71] V. Schneider et al. “Online trajectory generation using clothoid segments”. In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2016, pp. 1–6.

[72] David Seiferth et al. “Automatic Safe Area Detection In Three-Dimensional Airspace”. In: *AIAA Aviation 2019 Forum*. DOI: 10.2514/6.2019-3314. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2019-3314.

[73] R. Eisenschmidt GmbH, Deutsche Flugsicherung(DFS) Group. [Online; accessed 14-May-2019]. 2017. URL: https://www.eisenschmidt.aero/media/image/77/eb/f1/2017-11-29_Luftraumstruktur-3D_web5a1ec3cfa4262.jpg.

[74] Branko Grünbaum. “Polyhedra with Hollow Faces”. In: *Polytopes: Abstract, Convex and Computational*. Ed. by T. Bisztriczky et al. Dordrecht: Springer Netherlands, 1994, pp. 43–70. ISBN: 978-94-011-0924-6. DOI: 10.1007/978-94-011-0924-6_3. URL: https://doi.org/10.1007/978-94-011-0924-6_3.

[75] David Seiferth, Florian Holzapfel, and Matthias Heller. “Evasive Maneuvers of Optionally Piloted Air Vehicles For Three-Dimensional Geofencing”. In: *AIAA Scitech 2020 Forum*. DOI: 10.2514/6.2020-0490. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2020-0490.

[76] Florian Holzapfel. “Lecture notes for Flight System Dynamics I”. In: Institute of Flight System Dynamics, Technical University Munich, Garchig, 2016.

[77] S. P. Schatz et al. “Flightplan flight tests of an experimental DA42 general aviation aircraft”. In: *2016 14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2016, pp. 1–6. DOI: 10.1109/ICARCV.2016.7838646.

[78] Department of Defense. “Flight Control Systems - Design, Installation and Test of Piloted Aircraft, General Specification For”. In: *MIL-DTL-9490E*. 2008.

[79] SAE International. “Aerospace - Flight Control Systems - Design, Installation and Test of Piloted Military Aircraft, General Specification for”. In: *Aerospace Standard 94900 (SAE AS94900)*. 2007.

[80] United States Army Aviation and Missile Command Aviation Engineering Directorate Redstone Arsenal. “Aeronautical Design Stadnard Performance Specification Handling Qualities Requirements For Militiary Rotorcraft”. In: *ADS-33E-PRF*. Alabama,USA, 2000.

[81] NATO Standardization Agency. *STANAG 4671 - Unmanned Aerial Vehicles Systems Airworthiness Requirements*. Brussels, Belgium, 2009. URL: https://www.defense.gouv.fr/content/download/552731/9407958/file/4671eed01.pdf (visited on 02/01/2021).

[82] K. M. Sobel, J. C. Chung, and E. Y. Shapiro. “Application of MIMO Phase and Gain Margins to the Evaluation of a Flight Control System”. In: *1983 American Control Conference*. 1983, pp. 1286–1287. DOI: 10.23919/ACC.1983.4788322.

[83] Department of Defense. “Flying Qualities of Piloted Airplanes”. In: *Military Specification MIL-F-8785C*. Washington,USA, 1980.

[84] Department of Defense. “Flying Qualities of Piloted Airplanes”. In: *Military Handbook MIL-HDBK-1797*. Washington,USA, 1997.

[85] European Aviation Safety Agency. *Proposed Means of Compliance with the Special Condition VTOL*. 2020. URL: https://www.easa.europa.eu/sites/default/files/dfu/proposed_moc_sc_vtol_issue_1.pdf (visited on 03/25/2021).

[86] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: Analysis and Design*. 2nd ed. Hoboken, US-NJ: John Wiley, Aug. 2001.

[87] Matthias Heller. “Lecture notes on stability analysis”. In: *Flight dynamic challenges of highly-augmented aircraft*. Munich, Germany, 2016.

[88] *MATLAB Control System Toolbox*. The MathWorks, Natick, MA, USA. MATLAB Version: 9.8.0.1417392 (R2020a) Update 4.

[89] Felix Biertümpfel. “Analysis of the Robust Stability and Compliance of the Longitudinal Controller of Future Small Aircraft”. Master’s Thesis. München: Technische Universität München, 2016.

[90] Thaddäus Baier. “Entwicklung und Verifizierung eines Konzepts zur Robusten Regelung der Seitenbewegung zukünftiger Kleinflugzeuge”. Dissertation. München: Technische Universität München, 2016.

[91] Julian Theis. “Robust and Linear Parameter-Varying Control of Aeroservoelastic Systems”. Dissertation. München: Technische Universität Hamburg-Harburg, 2018.

[92] Simon Hecker. “Generation of low order LFT Representations for Robust Control Applications”. Dissertation. München: Technical University of Munich, 2006.

[93] Simon Hecker. “Guest lecture notes on μ-analysis”. In: *Flight dynamic challenges of highly-augmented aircraft*. Institute of Flight Systen Dynamics, Technical University Munich, Garchig, 2016.

[94] D. Seiferth et al. “Reduced Conservatism Proof of the Balanced MIMO Gain and Phase Margins”. In: *2020 59th IEEE Conference on Decision and Control (CDC)*. 2020, pp. 905–911. DOI: 10.1109/CDC42340.2020.9304342.

[95] Peter Seiler, Andrew Packard, and Pascal Gahinet. *An Introduction to Disk Margins*. 2020. arXiv: 2003.04771 [eess.SY].

[96] R.A. Horn and C.R. Johnson. *Topics in Matrix Analysis*. Cambridge University Press, 1991.

[97] Andreas Varga, Anders Hansson, and Guilhem Puyou. *Optimization Based Clearance of Flight Control Laws. A Civil Aircraft Application*. Berlin Heidelberg: Springer, 2012. DOI: 10.1007/978-3-642-22627-4.

[98] *MATLAB Robust Control Toolbox*. The MathWorks, Natick, MA, USA. MATLAB Version: 9.8.0.1417392 (R2020a) Update 4.

[99] A. Packard, M. K. H. Fan, and J. Doyle. “A power method for the structured singular value”. In: *Proceedings of the 27th IEEE Conference on Decision and Control*. 1988, 2132–2137 vol.3. DOI: 10.1109/CDC.1988.194710.

[100] A. Fabrizi, C. Roos, and Jean-Marc Biannic. “A detailed comparative analysis of μ lower bound algorithms”. In: *European Control Conference 2014*. Strasbourg, France, June 2014. URL: https://hal.archives-ouvertes.fr/hal-01098028.

[101] *MATLAB Statistics and Machine Learning Toolbox*. The MathWorks, Natick, MA, USA. MATLAB Version: 9.8.0.1417392 (R2020a) Update 4.

[102] Z. I. Botev. “The normal law under linear restrictions: simulation and estimation via minimax tilting”. In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 79.1 (2016), 125–148. ISSN: 1369-7412. DOI: 10.1111/rssb.12162. URL: http://dx.doi.org/10.1111/rssb.12162.

[103] H.-M. Gutmann. “A Radial Basis Function Method for Global Optimization”. In: *JOURNAL OF GLOBAL OPTIMIZATION* 19 (1999), p. 2001.

[104] *MATLAB Global Optimization Toolbox*. The MathWorks, Natick, MA, USA. MATLAB Version: 9.8.0.1417392 (R2020a) Update 4.

[105] Florian Holzapfel. “Lecture notes for Flight System Dynamics II”. In: Institute of Flight System Dynamics, Technical University Munich, Garchig, 2016.

[106] C. Blum et al. “Inertial laboratory simulation”. In: *2014 DGON Inertial Sensors and Systems (ISS)*. 2014, pp. 1–16. DOI: 10.1109/InertialSensors.2014.7049473.

[107] Niclas M. Bähr, Maximilian M. Soepper, and Florian Holzapfel. “Stochastic Lookup Tables - A Method for the Integration of Parametric Uncertainties in Non-Linear Simulation Models”. In: *AIAA Scitech 2021 Forum*. DOI: 10.2514/6.2021-0545. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2021-0545. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2021-0545.

[108] Ewoud Smeur. “Incremental Control of Hybrid Micro Air Vehicles”. Dissertation. München: Delft University of Technology, 2018.

[109] S. Sieberling, Q. P. Chu, and J. A. Mulder. “Robust Flight Control Using Incremental Nonlinear Dynamic Inversion and Angular Acceleration Prediction”. In: *Journal of Guidance, Control, and Dynamics* 33.6 (2010), pp. 1732–1742. DOI: 10.2514/1.49978.

[110] Tor A. Johansen and Thor I. Fossen. “Control allocation—A survey”. In: *Automatica* 49.5 (2013), pp. 1087–1103. ISSN: 0005-1098. DOI: https://doi.org/10.1016/j.automatica.2013.01.035.

[111] *MATLAB Optimization Toolbox*. The MathWorks, Natick, MA, USA. MATLAB Version: 9.8.0.1417392 (R2020a) Update 4.

[112] The MathWorks. *Surrogate Optimization Algorithm Documentation*. 2021. URL: https://de.mathworks.com/help/gads/surrogate-optimization-algorithm.html#mw_fd4feaed-717f-4e95-96f3-53890110908d (visited on 05/15/2021).

[113] *MATLAB Parallel Computing Toolbox*. The MathWorks, Natick, MA, USA. MATLAB Version: 9.8.0.1417392 (R2020a) Update 4.

[114] D. Seiferth and M. Heller. "Testing and performance enhancement of a model-based designed ground controller for a diamond-shaped unmanned air vehicle (UAV)". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. 2017, pp. 1988–1994. DOI: 10.1109/CCTA.2017.8062747.

[115] D. Seiferth, R. Kuchar, and M. Heller. "Model-based design and real live on-runway testing of a ground controller for a novel diamond-shaped Unmanned Air Vehicle (UAV)". In: *2017 IEEE 56th Annual Conference on Decision and Control (CDC)*. 2017, pp. 3934–3941. DOI: 10.1109/CDC.2017.8264238.

[116] David Seiferth, Martin E. Kügler, and Matthias Heller. "In-Flight Verification of a model-based designed Ground Controller for an innovative Unmanned Air Vehicle (UAV)". In: *2018 Flight Testing Conference*. DOI: 10.2514/6.2018-4276.

[117] M. E. Kügler and F. Holzapfel. "Enhancing the auto flight system of the SAGITTA Demonstrator UAV by fault detection and diagnosis". In: *IEEE International Conference on Aerospace Electronics and Remote Sensing Technology (ICARES)*. 2015, pp. 1–7.

[118] M. E. Kügler and F. Holzapfel. "Designing a safe and robust automatic take-off maneuver for a fixed-wing UAV". In: *14th International Conference on Control, Automation, Robotics and Vision (ICARCV)*. 2016, pp. 1–6.

[119] M. E. Kügler and F. Holzapfel. "Online self-monitoring of automatic take-off and landing control of a fixed-wing UAV". In: *2017 IEEE Conference on Control Technology and Applications (CCTA)*. 2017, pp. 2108–2113. DOI: 10.1109/CCTA.2017.8062764.

[120] Christoph Krause and Florian Holzapfel. "Designing a System Automation for a novel UAV Demonstrator". In: *14th International Conference on Control, Automation, Robotics & Vision (ICARCV)*. 2016.

[121] A. Zeitler et al. "A communication system approach for a small scale RPAS demonstrator". In: *2016 IEEE/AIAA 35th Digital Avionics Systems Conference (DASC)*. 2016, pp. 1–5. DOI: 10.1109/DASC.2016.7778111.

[122] Andreas Bierig et al. "Design of the General Systems for the SAGITTA Demonstrator UAV". In: *ICUAS 2017*. 2017. URL: https://elib.dlr.de/112801/.

[123] M. E. Kügler and F. Holzapfel. "Developing Automated Contingency Procedures for the ATOL System of a Fixed-Wing UAV through Online FDD". In: *AIAA Modeling and Simulation Technologies Conference, AIAA AVIATION Forum*. 2016, pp. 1–7.

[124] S. Braun et al. "Configuration assessment and preliminary control law design for a novel diamond-shaped UAV". In: *International Conference on Unmanned Aircraft Systems (ICUAS)*. 2014, pp. 1009–1020.

[125] M. Geiser and M. Heller. "Flight Dynamics Analysis and Basic Stabilization Study in Early Design Stages of the SAGITTA Demonstrator UAV". In: *Dt. Luft- und Raumfahrtkongress*. Berlin, 2012.

[126] Kajetan Nurnberger, Markus Hochstrasser, and F. Holzapfel. “Execution time analysis and optimisation techniques in the model-based development of a flight control software”. In: *IET Cyper-Phys. Syst.: Theory & Appl.* 2 (2017), pp. 57–64.

[127] Richard O. Kuchar. “A versatile Simulation environment for Design Verification, System Integration Testing and Pilot Training of a diamond-shaped Unmanned Aerial Vehicle”. In: *2018 AIAA Modeling and Simulation Technologies Conference*. DOI: 10.2514/6.2018-0126. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2018-0126.

[128] Lykourgos Bougas, Mirko Hornung, and Christian Rößler. “Design and Experimental Validation of a Propulsion Duct for a Jet Propelled Low Observable Scaled UAV Demonstrator”. In: *17th AIAA Aviation Technology, Integration, and Operations Conference*. DOI: 10.2514/6.2017-4378. URL: https://arc.aiaa.org/doi/10.2514/6.2017-4378.

[129] A. Hövelmann and C. Breitsamter. “Aerodynamic Characteristics of the SAGITTA Diamond Wing Demonstrator Configuration”. In: *Dt. Luft- und Raumfahrtkongress*. Berlin, 2012.

[130] Stefan Pfnür and Christian Breitsamter. “Unsteady aerodynamics of a diamond wing configuration”. In: *CEAS Aeronautical Journal*. Vol. 9. 2018, pp. 93–112. URL: https://doi.org/10.1007/s13272-018-0280-9.

[131] Hans B Pacejka and Egbert Bakker. “The magic formula tyre model”. In: *Vehicle system dynamics* 21.S1 (1992), pp. 1–18.

[132] Fabrizio Re and Ricardo de Castro. “Energetically Optimal Path Following for Electric Aircraft Taxi Systems Based on Convex Optimization”. In: *ESARS 2014 International Electric Vehicle Conference, Florence, Italy* (2014).

[133] Elmqvist Hilding, Martin Otter, and Sven Erik Mattsson. *Modelica Language Specification Version 3.3*. Tech. rep. Modelica Association, 2012.

[134] Torsten Blochwitz. *Functional Mock-up Interface for Model Exchange and Co-Simulation Specification Version 2.0*. Tech. rep. Modelica Association, 2014.

[135] A. Zeitler, A. Schwierz, and S. Hiergeist. “Datalink System Maturation and Flight Testing of the Sagitta UAS Demonstrator”. In: *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*. 2018, pp. 1–10. DOI: 10.1109/DASC.2018.8569693.

[136] *CATIA V5*. Dassault Systemes, France. CATIA Version: 5.20, Service Pack 7.

[137] *MIP Hard and Soft Iron Calibration Software*. LORD Corporation MicroStrain Sensing Systems, Williston, Vermont, USA. Version: 2.4.

[138] iMAR Navigation GmbH. *iTraceRT M-200 Datasheet*. St. Ingbert, Germany, 2014. URL: https://www.imar-navigation.de/downloads/TraceRT-M200-OLC.pdf (visited on 10/19/2019).

[139] Michael Stock. *Micro Air Data Computer (MADC) ARINC 825 Interface Control Document*. Farchach, Germany, 2015.

[140] Vectronix AG. *Vectronix Operator Manual DMC-SX Series*. Heerbrug, Switzerland, 2008.

[141] LORD Corporation MicroStrain Sensing Systems. *Lord Product Sheet 3DM-GX3-25*. Williston, Vermont, USA, 2016. URL: http://files.microstrain.com/3DM-GX3-25-Attitude-Heading-Reference-System-Data-Sheet.pdf (visited on 10/19/2019).

[142] LORD Corporation MicroStrain Sensing Systems. *LORD QUICK START GUIDE MIP Inertial Sensors with Magnetometers Hard and Soft Iron Calibration Software*. Williston, Vermont, USA, 2013. URL: http://files.microstrain.com/MIP%20Soft%20&%20Hard%20Iron%20Calibration%20Quick%20Start%20Guide.pdf (visited on 10/19/2019).

[143] Google Earth Pro 7.3.2.5776 (2019-03-05). *Manching, Germany. 48° 42′ 23.11″ N, 11° 31′ 03.45″ W, Eye alt 685 m*. 2018 Google, 2009 GeoBasis DE/BK, 2019 Google LLC. Mountain View, CA, USA, 2019. (Visited on 10/22/2019).

[144] J.L. Weston D.H. Titterton. *Strapdown inertial navigation technology*. Stevenage: Institution of Electrical Engineers, 2014.

[145] Google Earth Pro 7.3.2.5776 (2019-03-05). *Overberg, South Africa. 34° 34′ 04.20″ S, 20° 15′ 56.93″ O, Eye alt 2.01 km*. 2019 CNES / Airbus, 2018 Google, 2019 AfriGIS (Pty) Ltd. Mountain View, CA, USA, 2019. (Visited on 12/30/2019).

[146] M. E. Kügler et al. “Real-time monitoring of flight tests with a novel fixed-wing UAV by automatic flight guidance and control system engineers”. In: *2018 4th International Conference on Control, Automation and Robotics (ICCAR)*. 2018, pp. 141–146.

[147] Martin E. Kügler, Matthias Heller, and Florian Holzapfel. “Automatic Take-off and Landing on the Maiden Flight of a Novel Fixed-Wing UAV”. In: *2018 Flight Testing Conference*. DOI: 10.2514/6.2018-4275. eprint: https://arc.aiaa.org/doi/pdf/10.2514/6.2018-4275. URL: https://arc.aiaa.org/doi/abs/10.2514/6.2018-4275.